January 2014

Bayesian Modeling For Dealing With Uncertainty In Cognitive Radios

Hector Ivan Reyes Moncayo

BAYESIAN MODELING FOR DEALING WITH UNCERTAINTY IN COGNITIVE RADIOS

by

Héctor Iván Reyes Moncayo

A Dissertation

Submitted to the Graduate Faculty

of the

University of North Dakota

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Grand Forks, North Dakota
December
2014

This dissertation, submitted by Héctor Iván Reyes Moncayo in partial fulfillment of the requirements for the Degree of Doctor of Philosophy from the University of North Dakota, has been read by the Faculty Advisory Committee under whom the work has been done and is hereby approved.

Naima Kaabouch, PhD

Saleh Faruque, PhD

Prakash Ranganathan, PhD

Wen-Chen Hu, PhD

Ron Fevig, PhD

This dissertation is being submitted by the appointed advisory committee as having met all of the requirements of the School of Graduate Studies at the University of North Dakota and is hereby approved.

Wayne Swisher
Dean of the School of Graduate Studies

December 8, 2014
Date

PERMISSION

Title Bayesian Modeling for Dealing with Uncertainty in Cognitive Radios

Department Electrical Engineering

Degree Doctor of Philosophy

In presenting this dissertation in partial fulfillment of the requirements for a graduate degree from the University of North Dakota, I agree that the library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my dissertation work or, in her absence, by the Chairperson of the department or the dean of the School of Graduate Studies. It is understood that any copying or publication or other use of this dissertation or part thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of North Dakota in any scholarly use which may be made of any material in my dissertation.

Héctor Iván Reyes Moncayo
11/14/2014

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my gratitude to Dr. Naima Kaabouch, my advisor for her valuable guidance during the years I spent in my PhD program. I would also like to thank the other members of my advisory committee: Dr. Saleh Faruque, Dr. Prakash Ranganathan, Dr. Wen-Chen Hu, and Dr. Ron Fevig for their helpful feedback and support during this process. I also want to thank my colleague Sriram Subramaniam for his help during some of the experiments involved in my research.

The culmination of my doctoral program has been possible thanks to the financial support of Fulbright, Colciencias, ND EPSCOR through grant EPS-0184442, and Universidad of the Llanos in Villavicencio, Colombia.

Finally, I am grateful to my wife for these years of patience, support, and company.

To the memory of my father and to God, my heavenly father.

ABSTRACT

Wireless communication systems can be affected by several factors, including propagation losses, co-channel interference, and multipath fading. Uncertainty affects all of these factors making it even more difficult to model these systems. This dissertation proposes the use of probabilistic graphical models (PGM), such as Bayesian Networks and Influence Diagrams, as the core for reasoning and decision making in adaptive radios operating under uncertainty. PGM constitute a tool to understand and model complex relations among random variables. This dissertation explains how to build effective communication models that perform its functions under uncertainty. In addition, this work also presents a spectrum sensing technique based on the autocorrelation of samples to estimate the utilization level of wireless channels.

CHAPTER 1

INTRODUCTION

The environment conditions in Wireless networks change rapidly. Noise, co-channel interference, spectrum availability, multipath fading, propagation losses, Doppler phase, and Doppler frequency shift vary randomly impacting the overall performance of the wireless communication systems. Designers of conventional wireless communication systems assume that the conditions in which these systems are to operate will fall within certain intervals. For instance, when calculating the power budget of mobile and wireless links, a margin is included to account for the received power fading caused by the multipath propagation [1, 2]. Another typical assumption is that the radio system will tune in to channels allocated statically [3, 4]. In other words, conventional wireless communication systems are designed to work under predetermined conditions and lack adaptability to dynamic environments. Some problems of this conventional approach are: the conditions wherein radio systems operate are dynamic; and as the amount of wireless devices and services increases, a more optimal use of the radio spectrum becomes critical.

Cognitive radio (CR) technology has appeared as a potential alternative to conventional radio systems: one wherein the wireless system itself can learn from the environment and adapt to it. Although cognitive radio (CR) originally emerged as a technology for implementing

dynamic Spectrum access (DSA), its field of action can go further than the mere use of the spectrum. A CR can change not only its operating frequency but also its modulation scheme, spectral efficiency, transmission power, bandwidth, and other parameters in order to increase its performance in the midst of its surrounding conditions.

A cognitive radio (CR) is made of two main blocks: a software defined radio (SDR) block and an artificial intelligence (AI) block or cognitive engine (CE). The SDR performs the baseband signal processing operations by means of software, which provides flexibility when it comes to implementing radio systems, since the same hardware platform serves to realize different types of radios by changing the code in charge of the digital signal processing (DSP). The AI block, or CE, executes reasoning, learning, and decision making operations; and interacts with the SDR to control the DSP, so that the radio system adjusts its configuration according to the surrounding conditions and the performance objectives of the CR. The CR knows well its performance objectives; however, when it comes to knowing its surrounding conditions the CR has to face a problem: uncertainty. Therefore, a question arises: how can a cognitive radio learn, and make decisions in the midst of random conditions?

One possible answer to this question, explored in this dissertation is the use of probabilistic graphical models (PGMs) as a solution to represent wireless communication systems, while capturing the ever present uncertainty. In this dissertation, there are two types of PGM: Bayesian networks (BN) and influence diagrams (ID), which are an intuitive way to show how deterministic and random variables influence one to another. Bayesian networks are also known as belief networks, since their function is to constantly update their beliefs regarding the variables that compose them. In the context of PGM, those beliefs are on the states of the

variables. These beliefs are represented as probability distributions, so that calculations can be made in order to update those beliefs each time the system gets new evidence.

For instance, if we saw the sky totally blue, our belief about having no rain (NR) is stronger than the one about having rain (R); we can assign numbers to these beliefs: [p(R) = 0.1, p(NR) = 0.9]. Let us notice that all those numbers are positive and their sum is one; therefore, we can consider [p(R) = 0.1, p(NR) = 0.9], where p(R) and p(NR) represent the probability of having rain and no rain respectively. Suppose the sky is still blue but we start seeing some clouds; this is new evidence that makes us change our belief: now we think that is more probable that will see rain (R); so the numbers change to [p(R) = 0.4, p(NR) = 0.6]. Later on, dark clouds appear, new piece of evidence that reinforces the belief we have about having rain (R); therefore, the numbers change again: [p(R) = 0.8, p(NR) = 0.2]. Finally, it starts raining; now we are certain we have rain and the numbers tell so: [p(R) = 1, p(NR) = 0]. In this example, we can declare a variable, *rain*, with two states: rain (R) and no rain (NR). This variable is random; therefore, we need a probability distribution to characterize it. This probability distribution gets updated as new evidence comes. Bayesian networks are about declaring variables, their interaction, their states, and the probabilities of those states. This example has brought up something that as humans we do every day: we have beliefs about people, our environment, etc., we adjust our beliefs when we get new information (evidence), and we adapt to people and our environment. We do not assign numbers to those beliefs though; but we reason like in our example: in a Bayesian way.

Let us translate these ideas into the field of wireless communication systems. In those systems, we can find several random and deterministic variables in constant interaction. Some of these are noise, co-channel interference, propagation losses, transmission power, received power,

carrier frequency, bit error rate, etc. We can build the probabilistic graphical model (PGM) of a wireless communication system by defining the states of these variables, and characterizing qualitatively and quantitatively how they influence one another. Once built, this model can answer several queries; for instance, how likely it is that the co-channel interference in certain channel will be high (of course we have to define what high means)?, how likely it is that the bit error rate will be under certain level?, what is the configuration that fits the best the system user expectations about performance in a particular scenario?. This is the ultimate goal of this dissertation: having a model able to answer questions that a cognitive radio needs to learn and make decisions.

The main advantage of the PGM approach is that it can equip a CR with the means to learn about an uncertain environment, and rank decisions according to how satisfactory they are for the system. These tasks suit well the mission of cognitive radio systems: learn from the environment and adapt to it. With the increasing number of wireless systems and mobility services, this capability will become critical. Other benefits of PGMs: PGMs are modular; PGMs allow for the integration of knowledge, experimental data, and simulations; PGMs are intuitive.

This dissertation is organized as follows. Chapter 2 presents a background on cognitive radio (CR) technology and probabilistic graphical models (PGM). Chapter 3 describes the probabilistic models proposed to assist a cognitive radio in learning from the environment and making decisions; specifically, this work presents two models: one that takes the bit error rate (BER) as evidence to update the probability distributions of several variables that characterize the environment, such as co-channel interference, multipath fading, and noise; and other one that

calculates the utility of several configurations to find the ones with the highest utility for the wireless system. Chapter 4 presents several simulations that used to show the functionality of the proposed Bayesian network and Influence diagram. Chapter 5 explains the experiments performed on spectrum sensing, channel utilization estimation, and channel sounding. This chapter proposes a technique for spectrum sensing based on the autocorrelation of the received samples. This technique is applied to estimating the utilization level of several channels. In addition, this dissertation proposes to see the channel utilization level as a random variable and apply Bayesian probability to update its probability distribution each time the cognitive radio senses the state of the channel: busy or empty. Additionally, I present the results of experiments on channel sounding, the purpose of which is to estimate the channel impulse response of the wireless channel.

CHAPTER 2

BACKGROUND

2.1 Cognitive Radio Overview

Cognitive Radio is an emerging technology whose primary objective is the efficient utilization of the radio spectrum. A cognitive radio, built on a software-defined radio (SDR), is defined in [5] as an intelligent wireless communication system that is aware of the environment, learns from it and adapts to statistical variations in the input stimuli, with two main purposes:

Highly reliable communication whenever and wherever needed: since a cognitive radio can be aware of the conditions of the channel it can adapt to overcome propagation problems and increase the reliability.

Efficient use of radio spectrum: when primary users, incumbent licensees, are not using certain portions of the spectrum, secondary users, which are cognitive users, can use them opportunistically.

In this definition of cognitive radio, the term environment refers to the electromagnetic conditions surrounding the cognitive radio that fall within the frequencies of interest. Although the

input stimuli are mainly focused on the electromagnetic signals, the original idea of cognitive radio also contemplates other variables such as temperature and location [6].

To accomplish its mission, a cognitive radio executes a series of processes known as a cognitive cycle [6]. Figure 2.1.1 illustrates a simplified version of the cognitive cycle suggested by Doyle [3]. This cycle encompasses three parts: Observing, Decision Making and Taking Action. In each stage of the cycle, there are different processes taking place that involve techniques from different fields such as digital signal processing, estimation theory and artificial intelligence. A summary of the main aspects of each one follows.

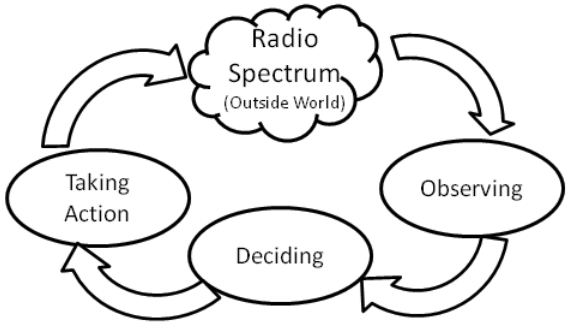


Figure 2.1.1: Cognitive Cycle (Modified from [16])

2.1.1 Observing

A cognitive radio is aware of the context where it operates. This awareness includes knowledge of the environment, the communication requirements of the users, the regulatory policies and its own capabilities [3]. Knowledge of the environment mainly includes understanding the surrounding radio spectrum scene. Spectrum sensing and channel estimation support the context awareness of a cognitive radio. Spectrum sensing is the process of obtaining awareness about the spectrum usage and existence of primary users in a determined area [7].

Channel estimation is the process of collecting the channel- state information (CSI) to assess the channel capacity [8] and other characteristics. In spectrum sensing we have to decide between two situations, primary user transmitting or primary user not transmitting. Therefore, the decision is between two discrete values. On the other hand, in channel estimation we have to obtain the approximate value of one or several parameters of a system that can take continuous values in a search domain [9].

Spectrum Sensing

Spectrum sensing falls in the domain of signal detection theory, in which we want to decide among some finite number of possible situations or "states of nature" [10]. In the context of cognitive radio, the decision is on whether or not there is a primary user present in the space and channel of interest for the secondary user. The more precise this decision is, the fewer the numbers of false alarms and misdetections will be. Each false alarm implies that the secondary user has wasted the opportunity of using an empty channel. Similarly, each misdetection can lead the secondary user to interfere with the primary user. Among the different tasks needed by cognitive radio, perhaps spectrum sensing is the one that has received the most attention from researchers. The main objective pursued by these researchers is to reduce the number of false alarms and misdetections, as well as the complexity and sensing time. The most common types of detectors mentioned in the literature are the energy detector, the cyclostationary detector, and the matched filter detector. Figure 2.1.2 compares these techniques in terms of complexity and accuracy.

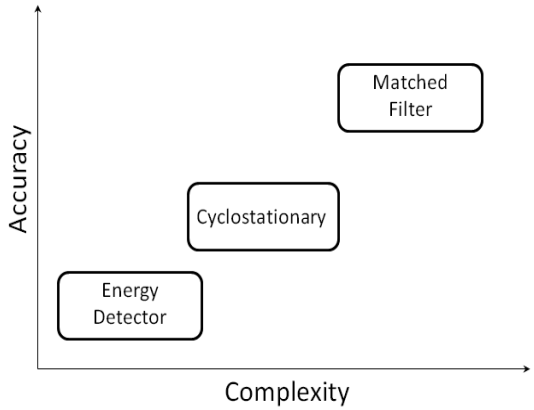


Figure 2.1.2: Comparison of spectrum sensing techniques (Modified from [17])

The energy detector is the simplest but also the least accurate .This sensor does not require any previous knowledge of the signal. The signal is detected by comparing the output of the energy detector with a threshold which depends on the noise floor [11]. The problem with this type of detector is that it does not distinguish between the signal to be detected and the noise. Additionally it performs poorly at low SNR.

The cyclostationarity detector takes advantage of the fact that the statistical parameters of practical communication signals vary periodically [12]. Examples of these signals include sinusoidal carriers in amplitude, phase, and frequency modulation systems, and periodic keying of the amplitude, phase, or frequency in digital modulation systems [13]. The cyclostationarity can be extracted by the spectral-correlation density (SCD) function [13-15]. The SCD function of modulated signals takes nonzero values at some nonzero cyclic frequencies. On the other hand, since noise is not cyclostationary its SCD has zero values at all non-zero cyclic frequencies. Therefore, by analyzing the SCD function it is possible to differentiate a particular signal from noise. In addition, the SCD provides a way to distinguish the signal type because different signals

have different non-zero cyclic frequencies [12]. Although cyclostationarity sensing performs well under low SNR conditions and uncertainty in the propagation channel, it has some drawbacks such as: (1) the sampling rate needs to be high, (2) the computation of SCD requires a large amount of samples, (3) the sampling time error and frequency offset could affect the cyclic frequencies [12] and (4) it is more complex compared with energy detection.

The matched filter detector is the most precise method for detecting primary users in cases where the transmitted signal is known. Its main advantage is that it takes less time than the other techniques to achieve a determined probability of false alarm or misdetection [16]. Nevertheless, matched filter sensing requires the cognitive radio to demodulate received signals. This condition means that the receiver needs to know signal features of the primary user such as bandwidth, operating frequency, modulation type, pulse shaping, and frame format [7]. That implies that the cognitive radio needs a different receiver for each type of signal it expects to detect.

Out of these three methods, ED is perhaps the simplest and popular method; however it requires knowledge of the noise power. MF and CSD need information about the signal prior to receive it. Inaccuracy in the knowledge of the noise power causes mistakes in the detection of signals. Similarly, the need of prior information about the signal increases complexity.

Covariance spectrum sensing can overcome these problems, since it capitalizes on the fact that the covariance matrix of noise and signals behave differently; therefore, it requires no information about the noise power or the signal [17]. The covariance matrix of the samples collected by the receiver contains information exploitable for the purpose of spectrum sensing. Zeng and Liang [17, 18] have proposed two methods that extract information out of the

covariance matrix of the samples. In [17] these authors introduce two statistics: the sum of the matrix elements that are not in the main diagonal, and the sum of the elements that are in the main diagonal. Comparing the ratio of these statistics with a threshold can tell the presence of either signal or noise. In [18] the authors use two metrics: the maximum to minimum eigenvalue (MME) ratio and the average received power to minimum eigenvalue ratio, also called energy with minimum eigenvalue (EME) detection. As in [17], the comparison of these ratios with a threshold can differentiate between noise and signal. The authors present simulations to evaluate the performance of their methods. Mate et al. [19] performed experiments with GNU Radio software and USRP (universal software radio peripheral) devices to evaluate the covariance and MME detection methods proposed in [17, 18]. They found that these two methods performed implausibly in the practice because the noise samples are not delta correlated as assumed in [17, 18]

To understand the principle behind the autocorrelation based sensing method, let us define $x(n) = s(n) + \eta(n)$ as the received samples, where $s(n)$ is the primary user signal, and $\eta(n)$ noise. Two hypotheses exist: 1) $\mathcal{H}_0$, i.e. absence of signal, and 2) $\mathcal{H}_1$, i.e. presence of signal. These hypotheses are given in [17, 18] and defined as

$$\mathcal{H}_0 : x(n) = \eta(n) \tag{2.1.1}$$

$$\mathcal{H}_1 : x(n) = s(n) + \eta(n) \ . \tag{2.1.2}$$

Let one define the vectors $\mathbf{x}$ and $\mathbf{s}$ as

$$\mathbf{x}(n) = [x(n) \ \ x(n-1) \ \cdots \ x(1)]^T , \tag{2.1.3}$$

and

$$\mathbf{s}(n) = [s(n) \;\; s(n-1) \;\; \cdots \;\; s(1)]^T \; . \tag{2.1.4}$$

The statistical covariance matrices of these vectors, defined in terms of the expectation E are

$$\mathbf{R}_{\mathbf{x}} = E[\mathbf{x}(n)\,\mathbf{x}^T(n)], \tag{2.1.5}$$

and

$$\mathbf{R}_{\mathbf{s}} = E[\mathbf{s}(n)\,\mathbf{s}^T(n)], \tag{2.1.6}$$

respectively.

According to [17], the matrix $\mathbf{R}_{\mathbf{x}}$ can be expressed as

$$\mathbf{R}_{\mathbf{x}} = \mathbf{R}_{\mathbf{s}} + \sigma_\eta^2 \mathbf{I} \; , \tag{2.1.7}$$

where σ_η^2 is the variance of the noise η, and $\mathbf{I}$ the identity matrix. Therefore, in absence of signal $\mathbf{R}_{\mathbf{s}}$ is zero as well as the non-diagonal elements of $\mathbf{R}_{\mathbf{x}}$. Based on this assumption, Zeng and Liang [17] have proposed the ratio between the sum of all the elements of $\mathbf{R}_{\mathbf{x}}$, $T_1 = \sum_n \sum_m |r_{nm}|$, and the sum its diagonal elements, $T_2 = \sum_n |r_{nn}|$, as metric to detect the absence or presence of signal. In absence of signal this ratio is supposed to be one, whereas with signal present this ratio is greater than one. It is clear that in the first case the ratio is $\frac{\sigma_\eta^2 \mathbf{I}}{\sigma_\eta^2 \mathbf{I}}$, whereas in the second case it is $\frac{\mathbf{R}_{\mathbf{s}} + \sigma_\eta^2 \mathbf{I}}{\sigma_\eta^2 \mathbf{I}}$. However, in the practice, even in absence of signal $\mathbf{R}_{\mathbf{x}}$ is a non-diagonal matrix and $T_1 / T2 > 1$[19].

The assumption about η made in [17] is that $E(\eta(n)\eta(n+\tau))=0$ for any $\tau \neq 0$. This condition holds when the noise is Gaussian, the problem is noise of the some receivers is colored and non-delta correlated, something to consider when analyzing the covariance or autocorrelation of the signal in order to decide between $\mathcal{H}_0$ and $\mathcal{H}_1$.

The method proposed in this dissertation calculates the autocorrelation of the samples defined as

$$\lambda(l) = \sum_{m=0}^{N_s-1} x(m)\, x^*(m-l), \tag{2.1.8}$$

where N_s is the number of samples and the symbol * represents the complex conjugate operation. Nevertheless, rather than using $\lambda(l)$ to build a covariance matrix as in [17-19], this work uses the Euclidean distance between $\lambda(l)$ and a reference line. Chapter 5 explains further details.

Channel Estimation

In addition to use spectrum sensing to know the availability of channels, their characterization becomes instrumental for the configuration and operation of wireless networks. Knowing the channel characteristics allows for the planning and adjustment of operation parameters of radio equipment, including transmission techniques, bandwidth, transmission power, bit rates and other parameters. Usually, these characteristics are gathered through exhaustive measurement campaigns. However, when looking for managing the radio spectrum dynamically, it is more useful to update the characteristics of the channel of interest with recent information.

During recent years and due to the increase of mobile wireless networks, the study of wireless channels that vary rapidly has attracted the attention of many researchers. Bello [20] introduced the assumption that the channel is *wide-sense stationary uncorrelated scattering* (WSSUS), which although relatively old remains in use today [21]. Recently in [22, 23] the authors show that the WSSUS assumption still holds although limited to a local context. In [21, 24] the author gives some guidance to approximate non-WSSUS channels to WSSUS channels. Literature presents several methods for estimating the response of the channel. The methods suggested in the literature can be divided into blind and non-blind methods. The blind methods obtain the channel response estimate without sending any pilot or training sequence. These methods estimate this response out of the samples the receiver takes. Maximum likelihood (ML) has been used to perform blind estimation [25, 26]. Even though ML has a good performance, it is computationally expensive. Some methods to reduce that complexity have appeared in the literature [27] including: Cyclic ML [26], Boolean Quadratic Program [28], and Expectation-Maximization [29, 30]. Other techniques such as subspaces [31] , second order statistics [32] and high order statistics [27] have also contributed to the improvement of blind estimation. On the other hand, non-blind estimation, which uses a pilot or training sequence has also been considered. As matter of fact, this approach is one of the most intensively studied methods for time-varying channels, which have short coherence time [33]. The pilots are previously known signals located in the time domain for single carrier systems and in the frequency domain for OFDM systems. Some of the proposed and most studied pilot aided estimators are linear minimum square error (LMMSE) estimator [34, 35], the least square (LS) estimator, and the best linear unbiased estimator (BLUE) [21]. What these methods have in common is that they use a

previously known sequence and the received signal to estimate the response of the channel. Another method is the correlation sounding technique, which takes advantage of the statistical properties of pseudo noise sequences and employs correlation to estimate the channel impulse response.

When propagating through a wireless channel, signals experience a variety of phenomena. These phenomena include multipath fading, shadowing, Doppler shift, and attenuation. Multipath is due to reflectors and scatters existing on the way of the signal; shadowing comes from obstacles that absorb energy from the signals; Doppler shift originates from the relative movement between transmitters and receivers; and attenuation is caused by the medium between nodes. Doppler shift also occurs when scatters and obstacles, located in the trajectory of the signal, move. Moreover, interferers affect the signal in such a way that can destroy it. Interferers are those devices transmitting at the same frequency as the system of interest. Sometimes this interference is temporary; therefore it is not always possible to detect it during the measurement campaign.

In order to quantify the magnitude of the aforementioned phenomena, the channel impulse response (CIR) has come into play. The Channel Impulse Response (CIR) of a wireless channel provides the information needed for the characterization of the channel. This information comprehends the number of paths used by the signal to propagate, the attenuation on each path and the relative delay between paths. Generally in the literature $h(\tau)$ symbolizes CIR as a function of the delay τ. However, since the behavior of the channel changes with time, the CIR also changes. In equation 2.1.9, $h(t,\tau)$ represents the time varying CIR (TV-CIR), t is the time,

τ is the delay, $a_i(t)$ is the time varying complex attenuation for the path i, $\delta(\tau-\tau_i)$ is the delta function representing the path with delay i and L is the number of paths [36].

$$h(t,\tau)=\sum_{i=1}^{L}a_i(t)\delta(\tau-\tau_i). \tag{2.1.9}$$

Different transformations of $h(t,\tau)$ produce diverse functions that characterize the channel in different domains, such as time, frequency, delay and Doppler [1, 21, 37].

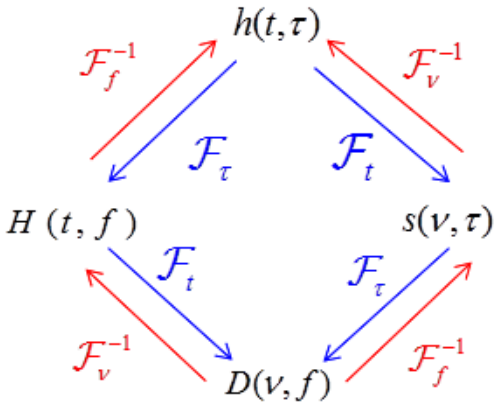


Figure 2.1.3: Channel System Functions Transformations. [1]

Figure 2.1.3 shows how $h(t,\tau)$ generates other functions by means of the Fourier transform, $\mathcal{F}$, and the inverse Fourier transform, $\mathcal{F}^{-1}$. These functions are the scattering function, $s(\nu,\tau)$, the Doppler variant transfer function, $D(\nu,f)$, and the time variant transfer function, $H(t,f)$. Although these functions depend on two variables, the Fourier transformations are with respect to one variable that can be t (time), f (frequency), τ (delay), or ν (Doppler frequency). For instance, $\mathcal{F}_\tau$ symbolizes Fourier transform with respect to the variable τ, $\mathcal{F}_\nu^{-1}$ inverse Fourier

transform with respect to the variable ν, and so on. The aforementioned functions are deterministic and characterize the channel in different but equivalent ways [1]. Specifically, the function $h(t,\tau)$ illustrates how the multiple propagation paths, their relative delays and attenuation levels change over time. Equivalently, the function $s(\nu,\tau)$ shows the multiple paths over time and the shift experienced by the central frequency due to the Doppler effect. The function $D(\nu, f)$ exhibits the effect of multipath in the frequency domain along with the Doppler effect. Finally, the function $H(t, f)$ is similar to $h(t,\tau)$; however, it shows how the multiple paths affect the channel in the frequency domain.

In the context of cognitive radio, the information provided by the functions explained previously needs to be expressed in a more condense way. The reason is that the reasoning, learning and decision making processes taking place in a CR can use more efficiently the information given in such a format than the whole information provided by the functions, which is helpful when graphically describing and analyzing the behavior of wireless channels. In this dissertation I am particularly interested in two condensed parameters: Delay spread and Doppler spread. These parameters lead to the *coherence bandwidth* and *coherence time* respectively. The coherence bandwidth, B_c, imposes restrictions over the bandwidth of the signal to transmit through the channel. Likewise, the coherence time, T_c limits the symbol time [1].

The Delay spread and the Doppler spread are the normalized second order central moments of the power delay profile (PDP) and the Doppler power spectrum (DPS) [1, 37, 38]. The PDP and the DPS come from considering the channel system functions as random processes. In fact this consideration is necessary since the original channel functions are deterministic and

unpredictable in the practice [21]. Therefore, statistical descriptions of the channel become necessary. These descriptions treat $h(t,\tau)$, $s(\nu,\tau)$, $D(\nu, f)$, and $H(t, f)$ as stochastic processes. To simplify these stochastic descriptions of the channel, the autocorrelation function along with the assumption that the channel is wide sense stationary-uncorrelated scattering (WSSUS) provide the necessary tools. For instance, by applying the autocorrelation function (ACF) to $h(t,\tau)$ we have [37]

$$R_h(t_1,t_2,\tau_1,\tau_2) = E[h(t_1,\tau_1)h^*(t_2,\tau_2)] \tag{2.1.10}$$

where * is the complex conjugate and $E[\cdot]$ is the expected value operation. The WSSUS, which is broadly accepted for mobile channels [39] model has two assumptions. The first assumption is that the stochastic process is wide sense stationary, WSS, which implies that the ACF depends only on $\Delta t = t_2 - t_1$, and not on the absolute time, t. Therefore, equation (2.1.10) becomes

$$R_h(\Delta t,\tau_1,\tau_2) = E[h(t,\tau_1)h^*(t+\Delta t,\tau_2)] \tag{2.1.11}$$

The second assumption is that the amplitudes and phases of the different paths are uncorrelated, which means the channel has uncorrelated scattering, US. Therefore, the ACF is zero when $\tau_1 \neq \tau_2$ and has a peak when $\tau_1 = \tau_2$. By applying this assumption to equation (2.1.11) we obtain

$$R_h(\Delta t,\tau) = E[h(t,\tau)h^*(t+\Delta t,\tau)], \tag{2.1.12}$$

which when calculated at $\Delta t = 0$ yields the function $p_h(\tau) = R_h(\tau) = R_h(\tau, 0)$, known in most of the literature as *power delay profile* [37]. The PDP represents the distribution of the power among the delayed paths of the signal arriving at the receiver. By normalizing the PDP, it turns into a probability density function, designated as $p(\tau)$. Equation (2.1.13) shows this normalization.

$$p(\tau) = \frac{R_h(\tau)}{\int_{-\infty}^{\infty} R_h(\tau) d\tau} = \frac{p_h(\tau)}{\int_{-\infty}^{\infty} p_h(\tau) d\tau} \tag{2.1.14}$$

The normalized second order central moment of $p(\tau)$ is

$$S_\tau = \sqrt{\int_{-\infty}^{\infty} (\tau - D_\tau)^2 p(\tau) d\tau} \tag{2.1.15},$$

known as the *delay spread*. In equation (2.1.15)

$$D_\tau = E[\tau] = \int_{-\infty}^{\infty} \tau p(\tau) d\tau \tag{2.1.16},$$

which is the *mean delay*.

Since in the practice, we will have only a limited number of discrete signals, we use the discrete versions of equations (2.1.15) and (2.1.16) as given by [37, 40]

$$S_\tau = \sqrt{\frac{\sum (\tau_i - D_\tau)^2 p_h(\tau_i)}{\sum p_h(\tau_i)}}\ , \tag{2.1.17}$$

where

$$D_\tau = \frac{\sum \tau_i p_h(\tau_i)}{\sum p_h(\tau_i)} \ . \tag{2.1.18}$$

A similar process works when calculating the Delay spread denoted as S_ν in this dissertation. In this case, we integrate the scattering function $s(\nu,\tau)$ with respect to τ. This integration yields $p_D(\nu)$, which is the Doppler spectrum. The equations (2.1.14) to (2.1.18) applied to $p_D(\nu)$ produce the parameter S_ν.

Matz and Hlawatsch [21] provide a definition of the coherence bandwidth B_c and the time bandwidth T_c in terms of the S_τ and S_ν respectively as given by equations (2.1.19) and (2.1.20).

$$B_c = \frac{1}{S_\tau}, \tag{2.1.19}$$

$$T_c = \frac{1}{S_\nu}. \tag{2.1.20}$$

The knowledge of the B_c, S_τ, T_c, and S_ν helps the communication system to adapt its operating configuration to fit better with the current conditions of the channel. Constraints on these parameters for different communication systems appear in the literature. For instance, in the case of an OFDM symbol with total symbol time $T_T = T_S + T_G$, where T_S is the symbol time and T_G is the guard time, Matolak [41] suggests

- $T_T >> S_\tau$ to avoid channel dispersion and get flat fading on each subcarrier,

- $T_G > S_\tau$ to avoid inter-block inter-symbol interference,
- $T_c > T_T$ to have coherent detection,
- $T_c > T_f$ to obtain slow fading, where T_f is the duration of a frame consisting in N symbols with duration T_S .

Similarly, Molisch [1] affirms that S_τ has strong impact on the bit error rate BER. Specifically, the error floor is proportional to S_τ. The delay spread depends on the scenery. Table 1 shows some typical values of S_τ for certain environments [1, 42].

Table 2.21: Typical values of S_τ for different environments [1, 42]

Environment	**Typical Delay Spread** S_τ
Indoor residential building	5-10 ns; but up to 30 ns
Indoor residential building	10-100 ns; but up to 300 ns
Factories and airport halls	50-200 ns
Microcells with Line of Sight (LOS)	5-100 ns
Microcells with no Line of Sight (non-LOS)	100-500 ns
Tunnels and Mines	Tunnels : 20 ns; Mines: up to 100 ns
Typical urban and suburban environments	100-800 ns; but up to 3 µs
Bad Urban and Hilly Terrain environment	S_τ up to 18 µs with clusters of 50 µs in cities and 100µs in mountainous terrain. A cluster is a group of signals from different trajectories arriving almost at the same time.

We have no similar measurements of S_ν, since this parameter depends on the carrier frequency, the relative speed between transmitter, and receiver and the angles between them.

The time varying- channel impulse response (TV-CIR) is then the basis for obtaining all the information needed for characterizing the channel. Therefore, the characterization of the channel starts with the estimation of $h(t,\tau)$. Rappaport [40, 43] classifies the methods for estimating $h(t,\tau)$ as follows:

- Pulse Envelope Measurement Systems: The transmitter sends a very short pulse approximate to the ideal pulse. Since the signal transmitted through the channel is an approximation of an impulse, the received signal is considered as an approximation of the TV-CIR or $h(t,\tau)$. The fact that these systems have to manage high power peaks creates a disadvantage since RF components working at high power are expensive and exhibit nonlinearities [1].

- PN Sequence Generator/Correlation System: The transmitter emits a pseudo-noise (PN) signal through the channel of interest. The receiver correlates the received signal with a stored copy of the PN sequence. The result of this correlation is an approximation of $h(\tau)$, which when calculated several times becomes an estimation of $h(t,\tau)$.

- Frequency Domain Channel Measurement: this method employs a vector network analyzer to measure the frequency response of the channel, which imposes a limitation, since both the transmitter and the receiver have to be connected to the network analyzer [43].

In cognitive radio systems, channel estimation is necessary for optimal adjustment of system parameters to changing conditions. In mobile communication systems such as UAS networks the received signal strength oscillates as the vehicle travels through interference patterns caused by

multipath, shadowing due to obstacles, and the change in distance between nodes. Generally, CR systems are designed to maximize their throughput and reliability for a given quality of service (QoS). This can be accomplished by adapting the system parameters to the fluctuations created by multipath and shadowing. This process requires estimation, prediction, and tracking of the received signal as accurate as possible [44].

Estimation is a statistical process that takes samples from the physical system to calculate approximate values of the parameters of the system. Normally, the parameter is identified as θ and its estimate as $\hat{\theta}$. The main idea behind estimation is to obtain an estimate as close as possible to the actual value of the parameter. There are different approaches to deal with θ and $\hat{\theta}$. Two of the most popular estimation approaches are Bayesian estimation and Maximum likelihood estimation.

The Bayesian estimation theory is based on the *Bayesian risk* and the *cost function* of the estimation error $\epsilon = \theta - \hat{\theta}$. The goal of a Bayesian estimator is to minimize the Bayesian risk $\mathcal{R}$, which is defined in (2.1.21) as the expected value E of the cost function $\mathcal{C}$ [10, 45] .

$$\mathcal{R} = E[\mathcal{C}(\epsilon)] \tag{2.1.21}$$

There are different kinds of cost functions. The most common ones are quadratic and step function [45]. When using the quadratic cost function, we have a minimum mean square error (MMSE) estimator. In the MMSE, we want to minimize $E[(\theta - \hat{\theta})^2]$, where the expectation is with respect to the power density function (PDF) $p(\mathbf{x};\theta)$. $p(\mathbf{x};\theta)$ is the PDF of $\mathbf{x}$, the vector of input samples, with θ as a parameter. On the other hand, when using the step function, also known as

"hit-or-miss" function, we have a maximum a posteriori (MAP) estimator. The "hit-or-miss" function assigns 1 when $\epsilon > \delta$ or $\epsilon < -\delta$, where $\delta > 0$ is the error threshold, and 0 otherwise. To minimize the Bayes risk with such cost function we have to maximize $p(\theta|\mathbf{x})$, which is the conditional, *a posteriori*, PDF of $\mathbf{x}$ conditioned on θ as explained in [45].

The maximum likelihood estimation (MLE) is the most popular approach to implementing practical estimators [45]. It is useful when the PDF of the parameter θ is unknown [9]. The objective of the MLE estimator is to choose θ so that the likelihood function is maximized [9, 10, 45].

2.1.2 Taking Action

Taking action for a cognitive radio (CR is to configure its transmission and reception parameters to obtain a desired behavior in order to accomplish a determined goal or set of goals [3]. The actions executed focus on two main activities. The first one is shaping the transmission profile and configuring any pertinent radio parameters to use efficiently the resources given to the CR and simultaneously not interfering with the resources of other radios. The second one is reshaping the transmission profile and reconfiguring the parameters when the resources change. The resources given to a CR are a set or frequencies, a set of time slots, and a set of antennas with beams pointed to different directions or any kind of combination of these. The cognitive engine (CE) is the entity that ultimately decides which actions the CR must take.

Any action over the physical level of the communication that is realizable by digital signal processing (DSP) can be implemented in a CR. The possibilities range from signal modulation to bean forming in smart antennas. The limitations are imposed mostly by the hardware, specifically

by the analog to digital converters and their sampling frequency. The DSP takes place in the SDR under the control of the CE. The CE decides the type of action to be taken. For instance, the CE can decide to increase the number of bits per symbol when the SNR allows it. This decision will increase the spectral efficiency. On the other hand, if the SNR is low because of attenuation, multipath fading, or other reasons; and the CE wants to maintain the wireless link, it can decide to reduce the number of bits per symbol in order to make the communication less prone to noise. Another example would be when the CR works with a smart antenna and the CE takes advantage of the steering capability of the antenna to exploit the space dimension. In conclusion, the actions taken by the CR affect the communication system in different aspects such as frequency, spatial fingerprint, robustness, and throughput.

2.1.3 Decision Making

The capability of making decisions is what distinguishes a cognitive radio from a conventional radio. This capability enables the cognitive radio to adapt itself to fulfill the specific requirements of a determined application. For instance, if the radio starts experiencing problems due to interference, the logical move is to switch to another channel. The CR needs to have a strategy to decide when it is going to switch the channel, which is the best channel to switch to, etc., always having in mind a goal. The goal could be maximizing throughput, reliability, or minimizing power consumption, and/or delay. It also can be a combination of these features or others. All these features need to be quantifiable in order to formulate a mathematical procedure to be executed in a computer or computation device.

Making decisions has associated other processes: o*rienting*, *planning*, and *learning* [6]. *Orienting* establishes priorities based on the observations. If the priority is normal, the next stage is *planning*, which implies to generate and/or evaluate the alternatives. If the priority is high the next stage is making a decision on the resources that would be allocated. *Learning* receives information from the other processes to build knowledge. This knowledge is fed back to the system to refine the deciding process.

The cognitive engine (CE) is the entity in the CR that executes the orienting, planning, deciding and learning tasks. The CE takes the stimuli, analyzes them and classifies the situation. It also determines the suitable response to the stimuli and decides how to reconfigure the system. The CE along with the SDR form the CR. Figure 2.1.4 shows the interaction of the CE with the SDR and the sensor [46]. In [3] the author compares a CR with a conventional radio having meters and knobs. Some examples of knobs are carrier frequency, bandwidth, signal duration, modulation scheme, transmission power, etc. She compares the meters to the sensing part of CR and the knobs with actuators in CR. In a conventional radio, it is a human operator who would decide when and how to move the knobs. In CR, the CE performs this task by using artificial intelligence (AI) techniques.

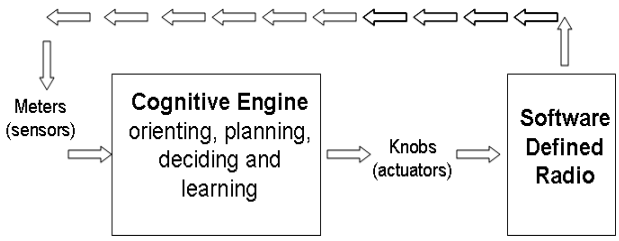


Figure 2.1.4: Cognitive engine and SDR (Modified from [30])

Diverse AI techniques have been proposed and are being investigated by researchers in an endeavor to have smarter cognitive radios [46-49]. Some of the proposed techniques are: Artificial Neural Networks (ANN), metaheuristic algorithms, and hidden Markov model (HMM). The literature reports the application of those techniques to different processes of CR including classification of signals for spectrum sensing [50, 51], radio parameter adaptation [52], [53], spectrum occupancy prediction [54], [55], and multi-objective optimization [56], [57].

Artificial neural networks (ANN) are a set of non-linear functions with configurable parameters that learn patterns from training data and can give outputs to new data according to the learned patterns. They provide a means of describing functions, processes or classes that are difficult to model analytically [47] . They have had different applications to cognitive radio, such as signal and modulation detection and classification [46]. For instance, for signal detection in low SNRs environments, ANNs have been combined with cyclostationarity-based spectrum sensing [58]. ANNs have shown their applicability in optimization of cognitive radios (CR) and performance characterization for different environments [48].

Metaheuristic algorithms have found place in CR implementations due to the lack of explicit relationships between the CR parameters and performance metrics. These algorithms scan through a set candidate configurations for suboptimal solutions. Genetic Algorithms (GA) fall under this category. They based their principles on genetic evolution and natural selection. The main idea is to combine the best candidates of chromosomes to produce better chromosomes, while eliminating the most deficient ones. The authors of [59] propose a multi-objective GA method for optimizing spectrum sensing to maximize the number of spectrum holes while keeping the

sensing overhead at certain limit. Rondeau and Warren [46] propose GA as a means for optimizing the configuration of CRs to satisfy specific performance goals. Huang et al. [60] propose the use of another heuristic algorithm called ant-colony-optimization (ACO) to improve the routing process in cognitive radio networks (CRN), which have to face variable spectrum availability and diverse quality of service (QoS) requirements.

Characterization of the activity of primary users (PUs) is the major field of application for hidden Markov models (HMM) to cognitive radio systems. Choi et al. [61] proposed the use of HMMs to estimate the duration of active and inactive periods for the PUs as well as the PU signal strength. Spectrum occupancy prediction has also benefited from HMMs [62]. Chen et al. [63] proposed a HMM-based channel state prediction model to reduce the negative impact of response delays created by hardware platforms. HMMs have also been considered for optimizing the handoff process in order for the secondary users (SUs) to maximize their transmission time, while minimizing collisions with PUs [64, 65]. For reducing such collisions, the SUs use past observations to estimate when the PUs will transmit again, so that they can vacate the channel before colliding with the PUs.

2.2 Probabilistic Graphical Models

2.2.1 The Bayesian Rule and probabilistic reasoning

Thomas Bayes and Pierre-Simon Laplace discovered independently the probabilistic theorem known today as the *Bayes' rule* [66] . Laplace, a renowned mathematician, gave this rule its mathematical form and scientific application [67]. Bayes' rule provides a tool to rigorously analyze evidence in the context of previous experience or knowledge [66]. The core of Bayes'

rule is the notion of *conditional probability*. When we have evidence that an event A has happened, we can adjust our belief about the occurrence of an event B, in other words, the probability of event B *conditioned* to event A, represented in the literature as $P(B \mid A)$. The conditional probability $P(B \mid A)$, also called the probability of B given A, is defined as

$$P(B \mid A) = \frac{P(A, B)}{P(A)}, \tag{2.2.1}$$

where $P(A, B)$ is the probability of events A and B occurring at the same time, also called *joint probability* of A and B, and $P(A)$ is the probability of event A occurs, also called prior probability of A [68, 69]. If $P(A) = 0$, $P(A \mid B)$ is undefined. By rearranging equation 2.2.1 we get

$$P(A, B) = P(A)P(B \mid A), \tag{2.2.2}$$

known as *chain rule* of conditional probabilities [69]. Equation 2.2.2 extended to more events becomes

$$P(A_1, \ldots, A_n) = P(A_1)P(A_2 \mid A_1) \cdots P(A_n \mid A_1 \ \ldots \ A_{n-1}); \tag{2.2.3}$$

therefore, we can factorize the joint probability of the combination of n events as the product of the probabilities of the first event, the second event conditioned to the first, the third event conditioned to the two first ones, and so on. Equation 2.2.3 holds for any order of events.

From 2.2.1 and considering that $P(A, B) = P(B, A)$ we can obtain the Bayes' rule [68];

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.2.4}$$

Bayes' rule can provide information about a hidden variable, or *hypothesis*, based on available data. Equation 2.2.4 written in a more general way becomes

$$P(hypothesis|data) = \frac{P(data|hypothesis)P(hypothesis)}{P(data)}, \tag{2.2.5}$$

where $P(hypothesis|data)$ is the *posterior* probability , $P(data|hypothesis)$ is the *likelihood*, $P(hypothesis)$ is the *prior* probability or *belief* about the hypothesis, and $P(data)$ is the *marginal likelihood* [66].

2.2.2 Bayesian Networks

A Bayesian network or belief network (BN) is a directed acyclic graph that expresses the beliefs about random variables along with their dependence relationships [70] . BNs make possible reasoning under uncertainty and combine the benefits of visual representation with the solid foundation of Bayesian probability [71]. They consist of two components: qualitative and qualitative components [72]. The qualitative component is the structure of the network that tells how the random variables relate one to another. The quantitative component contains the probability distributions that characterize the variables. The graphical representation makes easier the communication of domain knowledge among experts, users and systems. A Bayesian network can be built out of expert knowledge and/or from data collected from the process [72].

In a BN the vertices, or nodes, represent the random variables and the edges, or arcs, connecting them represent causal or dependence relationships among those variables. Child nodes

depend on parent nodes. The arcs, drawn as arrows, point from a parent to its children, indicating a causal dependence relationship. Each node has a probability distribution that quantifies the beliefs that the modeler has about the states of the variable. A root node is parentless and its probability distribution can come from expert knowledge and/or historical data. Children nodes have conditional probability distributions, the values of which depend on the different combinations of their parent states.

Bayesian networks are a tool for factorizing and performing operations over *joint distributions* [68, 69]. Joint distributions assign probabilities to events or states in terms of a set of random variables. Let $\mathcal{X} = \{X_1, \ldots, X_n\}$ be a set of random variables and $P(X_1, \ldots, X_n)$ the joint distribution that assigns probabilities to events specified in terms of the variables in $\mathcal{X}$. We can write $P(X_1, \ldots, X_n)$ as a product of conditional probabilities [68]. To give an illustration, from [68], let us consider the joint probability $P(T, J, R, S)$, where (T, J, R, S) are binary random variables. By applying Bayes' rule

$$P(X_1 \mid X_2) \equiv \frac{P(X_1, X_2)}{P(X_1)}, \tag{2.2.6}$$

several times to $P(T, J, R, S)$ we obtain the following factorization

$$\begin{aligned} P(T, J, R, S) &= P(T \mid J, R, S) P(J, R, S) \\ &= P(T \mid J, R, S) P(J \mid R, S) P(R, S) \\ &= P(T \mid J, R, S) P(J \mid R, S) P(R \mid S) P(S). \end{aligned} \tag{2.2.7}$$

To specify $P(T \mid J, R, S)$ we need $2^3 = 8$ values corresponding to the joint states of (J, R, S). Similarly, we need $4, 2, 1$ values for $P(J \mid R, S)$, $P(R \mid S)$, and $P(S)$, respectively, which

gives a total of $8+4+2+1=15$ values. Extended to a case with n binary random values this amounts to 2^n values. A Bayesian network helps to simplify the factorization of a joint distribution by describing conditional independencies graphically. It could happen that in this example T depends only directly on R and S, and J only on R, and R does not depend on S. The Bayesian network in figure 2.2.1 illustrates these conditional independence relations.

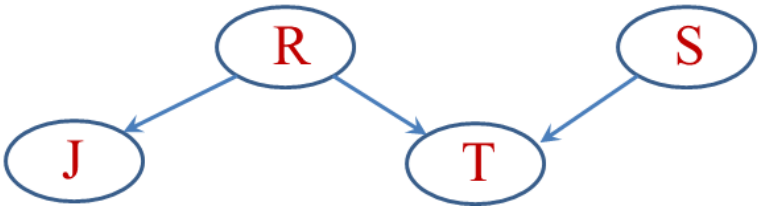


Figure 2.2.1: Bayesian Network

By considering these independence relations the factors in 2.2.7 now become

$$P(T \mid J, R, S) = P(T \mid R, S) \tag{2.2.8}$$

$$P(J \mid R, S) = P(J \mid R) \tag{2.2.9}$$

$$P(R \mid S) = P(R), \tag{2.2.10}$$

and the factorization of $P(T, J, R, S)$ now is

$$P(T, J, R, S) = P(T \mid R, S)P(J \mid R)P(R)P(S), \tag{2.2.11}$$

which reduces the amount of values needed to $4+2+1+1=8$. This is almost half the number of values needed when we ignore the conditional independences.

Equation 2.2.7 is an application of the chain rule for Bayesian networks defined in [69, 72] as

$$P(X_1,\ldots,X_n)=\prod_{i=1}^{n}P(X_i \mid Pa_{X_i}), \tag{2.2.12}$$

where Pa_{X_i} symbolizes the parents of the random variable X_i. Each factor $P(X_i \mid Pa_{X_i})$ is a conditional probability distribution (CPD).

The probabilistic influence not only does propagate from a parent to its children but also through trails that connect the nodes via intermediate nodes. Figure 2.2.2 shows how probabilistic influence propagates. In the four cases shown in figure 2.2.2 a trail between X and Y exists through Z. In the two first cases, a) and b), we consider X as indirect cause of Y. Figure 2.2.2.a shows the indirect *causal* effect of X over Y. Since X influences Z, Z influences Y. This effect propagates only when Z is not observed. If we observe Z, we no longer care about X, which means X loses influence over Y. Figure 2.2.2.b shows an indirect *evidential* effect of Y over X. Although, X causes Y, what we know about Y influences our belief about X. Nevertheless, as in the previous case, this holds only when Z is not observed. In figure 2.2.2.c, Z causes both X and Y. What we know about X impacts our belief about Y and vice versa. Likewise the previous cases, observing Z stops the probabilistic influence of X over Y, and Y over X. In the three previous cases $X \perp\!\!\!\perp Y \mid Z$, where $\perp\!\!\!\perp$ means conditionally independent. In figure 2.2.2.d, Z is a common effect of X and Y. When Z is unknown the trail between X and Y is inactive, which means $X \perp\!\!\!\perp Y$. Unlike the previous

cases, observing Z propagates the probabilistic influence between X and Y, i.e. $X \not\perp\!\!\!\perp Y \mid Z$, where $\not\perp\!\!\!\perp$ means conditionally dependent.

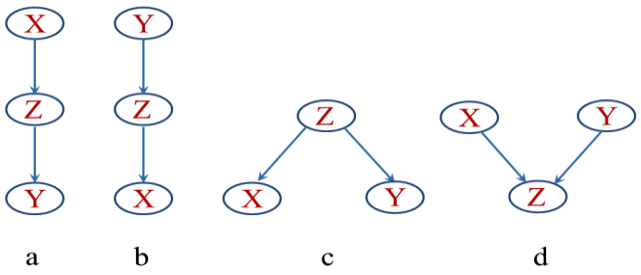


Figure 2.2.2: Flow of influence in Bayesian Networks

2.2.3 Influence Diagrams

When a Bayesian network is complemented with decision variables and utility nodes, it becomes an influence diagram (ID). An influence diagram facilitates decision making under uncertainty [68, 69, 72]. The random nodes of the diagram represent the uncertainty in the process. Some of the random nodes provide information to the decision nodes, which in turn affect other random variables. The utility nodes encode the preferences of the decision maker by assigning a particular value to each possible combination of choices and the states of the random variables. An ID permits the application of the maximum expected utility (MEU) principle to find the choice that favors the most to the decision maker in a particular situation. When building IDs is important to remember some simple rules: decision nodes can either be root nodes or depend on other decision nodes as well as chance nodes. They have to parent other decision nodes, chance nodes, or utility nodes. Utility nodes depend on decision and chance nodes, and have no children. An influence diagram needs at least one chance node, one decision node, and

one utility node. If the ID has several decision nodes, the designer of the ID must consider the order in which the decisions take place as well as the variables observed before making each decision.

Figure 2.2.3 illustrates a simple influence diagram, classical example from the literature [69]. The ovals represent chance or random variables, the rectangles decision variables, and the rhombuses utility nodes. The purpose of this ID is to help a business man decide whether or not to found a new business. The variable *Market* reflects how good the market is. It has three states: m^0, m^1, and m^2 that represent bad market, medium market, and good quality market respectively. Good quality market means that the market can return high profits. The node *Survey* represents the results of a study done on the quality of the market. It has four states: s^0, s^1, s^2, and s^{nc} that represent the results of the study estimate the market as bad, medium, and good quality respectively. The state s^{nc} means survey not conducted. The nodes *Test* and *Found* are binary nodes representing the decisions on doing the survey and founding the business. The states t^0, and f^0 indicate *action not taken*, whereas the states t^1 and f^1 *action taken*. The utility node *Cost* reflects the expenses of conducting the survey. We represent this variable as V_S. If the survey takes place, $V_S = -1$, otherwise $V_S = 0$. The node *Value* signifies the estimated return obtained depending on the situation of the market. V_R represents that return. $V_R = -7$, $V_R = 5$, and $V_R = 20$ when the market is bad, medium and good respectively. Notice we use negative values to indicate costs, or losses. For instance, when the market is bad, the businessman will lose money if he decides to found the business.

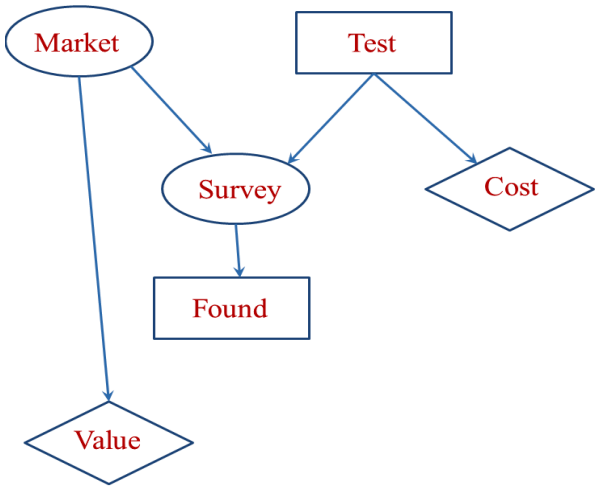


Figure 2.2.3: Example of Influence Diagram [69]

The nodes in this example have the following conditional probability tables (CPT). The table for *Market* is:

m^0	m^1	m^2
0.5	0.3	0.2

the table for *Survey* is:

	s^0	s^1	s^2
m^0	0.6	0.3	0.1
m^1	0.3	0.4	0.3
m^2	0.1	0.4	0.5

Notice that s^{nc} does not appear in the table; the reason is that this state only makes sense when the study is not conducted. We can express that fact with the conditional probability $P(s^{nc} \mid t^0, m) = 1$, which holds for all the states of the node *Market* represented as m .

The purpose of an influence diagram is to analyze the convenience of different courses of actions or strategies taken by an *agent* in an uncertain scenario. To accomplish that, each decision node D has a *decision rule* δ_D telling it what to do under all possible state combinations of its parent nodes [69]. A decision rule δ_D is a conditional probability $P(D \mid Pa_D)$ that acts as a function to map each instantiation pa_D of Pa_D to a probability distribution over the all different choices of D. A *decision rule* is deterministic if $\delta_D(D \mid pa_D) = 1$ for exactly one value of D. A *strategy* σ_D is the complete assignment of decision rules δ_D to each decision node of the influence diagram [69]. Changing a strategy means assigning other set of decision rules δ_D to the decision variables D of the influence diagram.

The influence diagram approach assumes a *single agent* making decisions one at a time in some specific order. Another assumption is that the agent remembers all its previous decisions and the information that it has acquired previously. Literature calls this *perfect recall* or *no forgetting* assumption [68, 69, 72]. Before making a decision the agents might or might not observe some random variables. This decision sometimes affects other random variables. Since an influence diagram is an acyclic graph, the agent cannot see the variables affected by the decision. To specify the order in the decisions, we partition the random variables $\mathcal{X}$ into disjoint

sets $\mathcal{J}_0,\ldots,\mathcal{J}_n$ so that the variables in set $\mathcal{J}_i$ contains the variables observed after decision D_i and before decision D_{i+1}. For instance, the agent observes $\mathcal{J}_0$ before D_1. The ordering of the influence diagram is expressed as

$$\mathcal{J}_0 \prec D_1 \prec \mathcal{J}_1 \prec D_2 \prec \cdots \prec \mathcal{J}_{n-1} \prec D_n \prec \mathcal{J}_n, \tag{2.2.13}$$

where the symbol $\prec$ represents precedence, $\mathcal{J}_0$ the set of variables observed before making the first decision (D_1), and $\mathcal{J}_n$ the set of variables observed after the last decision (D_n) is made. In some cases, the agent makes the first decision without observing any variable, i.e. $\mathcal{J}_0 = \varnothing$ and observes no variable after the last decision, i.e. $\mathcal{J}_n = \varnothing$.

2.2.4 Maximum Expected Utility

After specifying all the details of the influence diagram, we can proceed to calculate the expected utility for different strategies. We use $\mathcal{I}[\sigma]$ to represent the influence diagram with strategies σ [69]. Since the decision rules over decision nodes are probability distributions, we can think of them as conditional probability distributions and see the decision nodes as deterministic nodes. Therefore, $\mathcal{I}[\sigma]$ becomes to a Bayesian network that defines a probability distribution over the possible outcomes ζ [69, 73]. We represent this BN as $\mathcal{B}_{\mathcal{I}[\sigma]}$ following the format in [69]. The expected utility of $\mathcal{I}[\sigma]$ corresponds to [68, 69, 72]

$$\mathrm{EU}[\mathcal{I}[\sigma]] = \sum_{\zeta} P_{\mathcal{B}_{\mathcal{I}[\sigma]}}(\zeta) U(\zeta), \tag{2.2.14}$$

where $U(\zeta)$ is the utility of the outcome ζ corresponding to the sum of each utility variable in that outcome:

$$U(\zeta) = \sum_{V \in \mathcal{U}} \zeta\langle V \rangle . \tag{2.2.15}$$

An alternative to equation 2.2.14 that expresses more explicitly the factors that parameterize the network is:

$$\mathrm{EU}[\mathcal{I}[\sigma]] = \sum_{\mathcal{X} \cup \mathcal{D}} \left[\left(\prod_{X \in \mathcal{X}} P(X \mid P_{a_X}) \right) \left(\prod_{D \in \mathcal{D}} \delta_D \right) \left(\sum_{i \,:\, V_i \in \mathcal{U}} V_i \right) \right]. \tag{2.2.16}$$

The main purpose of calculating the expected utility (EU) is to find the strategy that the agent must follow to obtain the highest benefit, satisfaction or utility. That strategy, better known as maximum expected utility (MEU) strategy σ^*, maximizes the expected utility (EU). Therefore, the decision making progress assisted by ID comes down to find:

$$\arg\max_{\delta_{D_1}, \ldots, \delta_{D_k}} \mathrm{EU}[\mathcal{I}[\delta_{D_1}, \ldots, \delta_{D_k}]], \tag{2.2.17}$$

which the literature refers to as MEU principle [68, 69, 72].

In the case of the example shown in figure 2.2.3 when the MEU principle is applied the best strategy turns out to be to perform the study and found the business when the result of the study is either s^1 or s^2. The EU of this strategy is 3.22. Details can be found in chapter 23 of [69].

2.2.5 Probabilistic Inference in Bayesian Networks and Influence Diagrams

Bayesian networks and influence diagrams have the ability of answering different types of queries about the nodes they contain. Making queries to a BN or ID is a form of *probabilistic reasoning or probabilistic inference* [68]. Normally, the inference starts when we have *evidence*, which means we observe a variable or set of variables, and we want to know the probability of other variables given the evidence. For example, in the situation described by figure 2.2.4 we observe that $J=1$, and want to know the probability of $R=0$; we represent this query as $P(R=0 \mid J=1)$. This operation is known as *conditioning*. Another query is *marginalization,* in which we look for the probability of a variable no conditioned to the other variables. In the example shown in figure 2.2.4 we could *marginalize* to obtain $P(J)$, $P(T)$, $P(S)$, or $P(R)$. Other operations that BNs allow are: most probable explanation (MPE), maximum a posteriori probability (MAP), and sensitivity analysis [69, 72, 74].

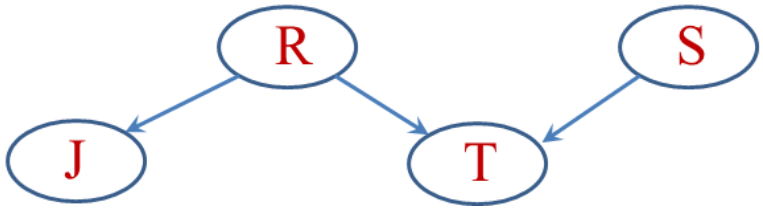


Figure 2.2.4: Example of Bayesian Network

Let us have a BN composed of the sets $\mathbf{E}$ and $\mathbf{Q}$. The set $\mathbf{E}$ contains the evidence variables, whereas $\mathbf{Q}$ contains the remainder of the variables of the BN that are not evidence. When $\mathbf{E}=\mathbf{e}$, MPE finds the instantiation $\mathbf{q}$ of $\mathbf{Q}$ that maximizes the probability $P(\mathbf{q} \mid \mathbf{e})$. If $\mathbf{E}=\mathbf{e}$, MAP finds the instantiation $\mathbf{q}_{sub}$ of $\mathbf{Q}_{sub} \subseteq \mathbf{Q}$ that maximizes the probability $P(\mathbf{q}_{sub} \mid \mathbf{e})$.

MPE looks for the instantiation of all the non-evidence variables, $\mathbf{Q}$, that explains $\mathbf{E} = \mathbf{e}$ with the highest probability. MAP does so but only for some of the non-evidence variables.

Out of the aforementioned types of queries, conditioning or *conditional probability query* is the most common [69]. For solving this type of queries there exist exact and approximate inference algorithms. For this dissertation I use an exact inference algorithm called variable elimination (VE). The main idea behind the VE algorithm is to eliminate the variables that are neither query nor evidence. The VE algorithm takes the factorized joint probability distribution and sums out the variables that it needs to eliminate. This operation is also called *factor marginalization*.

The factorized form of the joint probability distribution obtained with the chain rule for Bayesian networks - equation 2.2.7 - can be seen as the product of factors ϕ. A factor is a function that maps a set of variables $\mathbf{X}$ to a real number; $\phi: Val(\mathbf{X}) \mapsto \mathbb{R}$, where $Val(\mathbf{X})$ means value of the variable $\mathbf{X}$ and $\mapsto$ means map to. The set $\mathbf{X}$ is called *scope* of the factor; therefore, $Scope[\phi] = \mathbf{X}$. Equation 2.2.18 shows the chain rule for Bayesian networks as a product of factors:

$$P(X_1,\ldots,X_n) = \prod_{i=1}^{n} P(X_i \mid Pa_{X_i}) = \prod_{i=1}^{n} \phi_{X_i}, \tag{2.2.18}$$

where the conditional probability $P(X_i \mid Pa_{X_i})$ is represented by the factor ϕ_{X_i}. The scope of ϕ_{X_i} is the variable X_i and its parents Pa_{X_i}.

Let us consider the set of variables $\mathbf{X}$ and a variable $Y \notin \mathbf{X}$ being the scope of the factor ϕ: $\phi(\mathbf{X}, Y)$. *The factor marginalization* of Y in ϕ, denoted $\sum_Y \phi(\mathbf{X}, Y)$, is equivalent to a factor ψ over $\mathbf{X}$ such that [69] :

$$\psi(\mathbf{X}) = \sum_Y \phi(\mathbf{X}, Y). \tag{2.2.19}$$

Another name for the operation in 2.2.19 is summing out of Y in ψ. In this operation we only should sum up combinations where the states of $\mathbf{X}$ coincide.

The factor product and summation operations have properties equivalent to those of the product and summation over numbers [69]. Both operations are commutative: $\phi_1 \cdot \phi_2 = \phi_2 \cdot \phi_1$ and $\sum_X \sum_Y \phi = \sum_Y \sum_X \phi$; the product is associative: $(\phi_1 \cdot \phi_2) \cdot \phi_3 = \phi_1 \cdot (\phi_2 \cdot \phi_3)$; and they are interchangeable:

$$\sum_X (\phi_1 \cdot \phi_2) = \phi_1 \cdot \sum_X \phi_2, \tag{2.2.20}$$

if $X \notin Scope[\phi_1]$. This property allows to "push in" the summation, so that the summation is performed only on the subset of factors that contain the variable we want to eliminate. For instance, in 2.2.20 since we want to eliminate X, we push in the summation to sum only over ϕ_2 because ϕ_1 does not contain X.

The variable elimination (VE) algorithm takes advantage of the aforementioned properties. Figure 2.2.5 shows a very simple Bayesian network. The joint probability distribution for this

Bayesian network is: $P(A,B,C,D) = \phi_A \cdot \phi_B \cdot \phi_C \cdot \phi_D$. If for instance we want to know the marginal probability of D, $P(D)$, we apply factor marginalization: $P(D) = \sum_C \sum_B \sum_A P(A,B,C,D)$. By applying property 2.2.15 to it we get:

$$\begin{aligned} P(D) &= \sum_C \sum_B \sum_A \phi_A \cdot \phi_B \cdot \phi_C \cdot \phi_D \\ &= \sum_C \sum_B \phi_C \cdot \phi_D \cdot \left(\sum_A \phi_A \cdot \phi_B \right) \\ &= \sum_C \phi_D \cdot \left(\sum_B \phi_C \cdot \left(\sum_A \phi_A \cdot \phi_B \right) \right). \end{aligned} \tag{2.2.21}$$

The procedure in 2.2.21 can be summarized as:

$$\sum_Z \prod_{\phi \in \Phi} \phi \ . \tag{2.2.22}$$

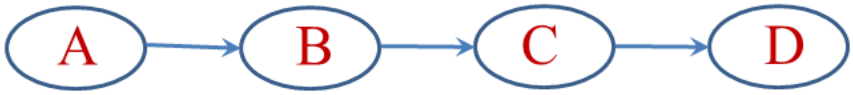


Figure 2.2.5: Simple BN for illustration

The expression 2.2.22 is also called *sum-product* inference task [69]. The variable elimination (VE) algorithm performs this inference task to sum out variables once at a time by using the property 2.2.20. When summing a variable, we multiply the factors that contain such variable to obtain a product factor. The next step is to sum out the variable from this product factor to generate a new factor, which will go to the next iteration as part of the new set of factors that the VE algorithm will be apply on. The VE algorithm will iterate until it removes all the variables it aim to eliminate. We can summarize the VE algorithm as follows [68, 69]:

The VE algorithm receives a set of factors Φ, a set of variables to eliminate $\mathbf{Z}$ and an ordering on $\mathbf{Z}$, $\prec$. If the set $\mathbf{Z}$ is $[Z_1,\ldots,Z_k]$, let the ordering $\prec$ be $Z_i \prec Z_j$ if and only if $i < j$. The set $\mathbf{Z}$ encompasses those variables that are neither query nor evidence. We refer to this algorithm as procedure Sum-Product-VE($\Phi, \mathbf{Z}, \prec$). The procedure Sum-Product-VE($\Phi, \mathbf{Z}, \prec$) follows these steps:

for $i = 1,\ldots,k$ do

$\Phi \leftarrow$ Sum-Product-Eliminate-Var(Φ, Z_i)

$\phi^* \leftarrow \prod_{\phi \in \Phi} \phi$ after completing the k_{th} iteration.

return ϕ^* at the end.

The procedure Sum-Product-Eliminate-Var(Φ, Z_i) is performed for each of the iterations $i = 1,\ldots,k$. This process receives the set of factors Φ, and the variable to be eliminated Z; then it performs the next operations:

1. Form a set Φ' with the factors that have Z in their scope:

$$\Phi' \leftarrow \{\phi \in \Phi \ : \ Z \in Scope[\phi]\}$$

2. Form a set Φ'' with the factors that do not have Z in their scope, which is the set Φ without Φ':

$$\Phi'' \leftarrow \Phi - \Phi'$$

3. Multiply the factors in the set Φ' and save the result in factor ψ :

$$\psi \leftarrow \prod_{\phi \in \Phi'} \phi$$

4. Add up the elements of the factor ψ where Z varies; this action eliminates Z from that factor. After eliminating Z from ψ save the result in the factor τ :

$$\tau \leftarrow \sum_{Z} \psi$$

5. Return the union between the set Φ'' and the factor τ

return $\Phi'' \cup \{\tau\}$

The VE algorithm also applies when introducing evidence. Let us have a Bayesian network K that parameterizes the set of variables $\mathcal{X}$, the set of query variables $\mathbf{Y}$, and evidence $\mathbf{E} = \mathbf{e}$. When introducing evidence the task is to compute $P(\mathbf{Y}, \mathbf{e})$. To execute this task, the factors are reduced by $\mathbf{E} = \mathbf{e}$ and eliminate the variables $\mathcal{X} - \mathbf{Y} - \mathbf{E}$ before applying the Sum-Product-VE $(\Phi, \mathbf{Z}, \prec)$ procedure to the network K. The factor ϕ^*,which comes from Sum-Product-VE $(\Phi, \mathbf{Z}, \prec)$,divided by α is $P(\mathbf{Y}, \mathbf{e})$. This whole procedure whereby $P(\mathbf{Y}, \mathbf{e})$ is obtained is called Cond-Prob-VE$(K, \mathbf{Y}, \mathbf{E})$ procedure. This procedure encompasses the next steps [69]:

1. $\Phi \leftarrow$ Factors parameterizing K

2. Replace each $\phi \in \Phi$ by $\phi[\mathbf{E} = \mathbf{e}]$

3. Set an elimination ordering $\prec$

4. $\mathbf{Z} \leftarrow = \mathcal{X} - \mathbf{Y} - \mathbf{E}$

5. $\phi^* \leftarrow$ Sum-Product-VE$(\Phi, \prec, \mathbf{Z})$

6. $\alpha \leftarrow \sum_{y \in Val(\mathbf{Y})} \phi^*(y)$

7. **return** α, ϕ^*

CHAPTER 3

BAYESIAN APPROACH FOR COGNITIVE RADIO

3.1 Introduction

Uncertainty affects all cognitive radio systems' processes. It appears everywhere in the cognitive radio cycle in terms of observing, decision-making, and taking action [6]. In the observing stage, uncertainty impacts the measurements taken by the CR, and concealed environmental factors that affect the measurements. In the next stage, the CR makes decisions based on what it senses from the environment and its own knowledge base. However, sometimes the CR knowledge base is built on models that leave out uncertainty. When making decisions and taking actions, the CR expects certain results, which in the practice are impacted by uncertainty.

This dissertation proposes to deal with uncertainty by using probabilistic graphical models (PGM), namely Bayesian networks (BN) and influence diagrams (ID). BN and ID intuitively define the interactions among deterministic and random variables utilizing directed graphs, where the cause variables point towards the effect variables. BN and ID deal with uncertainty by expressing random variables through probability distributions. Another important benefit of these techniques: they allow the integration of the knowledge about a problem with the data obtained from the scenario being modeled either through experiments or simulations. As an illustration, let us consider the free space loss function, FSL_{dB}:

$$FSL_{dB} = 32.45 + 20 \cdot \log(D_{Km}) + 20 \cdot \log(Fc_{MHz}), \tag{3.1.1}$$

where FSL_{dB} represents the free space loss in dB, D_{Km} , the distance between the transmitter and the receiver given in kilometers, and Fc_{MHz} , the carrier frequency in MHz. This expression not only provides a way to calculate FSL_{dB} but also describes how the variables D_{Km} and Fc_{MHz} influence the variable FSL_{dB}. In the field of wireless communications, expressions like 3.1.1 conform our preliminary knowledge, which combined with data and experience can be used the probabilistic model of the system.

Influence diagrams provide a viable means to make the decision, i.e. choose the action, which returns the highest utility for a particular application. To define the utility functions (UF) that calculate this utility, we will define criteria that the UF can weigh to calculate the utility. The method the utility function uses to weigh the criteria depends on the particular scenario. For instance, in one scenario, the highest data rate will weigh more than in another scenario, where saving energy weighs the most. In applications where spectrum auctioning is enabled, the cost of using a particular channel should then impact the selection of the most optimal configuration. Table 3.1.1 Summary of benefits of using BN and ID.

Table 3.1.1: Advantages of using Bayesian Networks and Influence Diagrams

Advantage	**Description**
Handling of uncertainty	BN incorporate uncertainty in the model as they base their parameters on the beliefs the modeler has about the problem. Several uncertain variables affect the wireless system, which in turn impacts the spectrum decision.
Modularity	Variables can be added or taken out of the models, as the modeler learns more about the problem.
Integration of data and expert knowledge	To build the model, we integrate our theoretical knowledge about wireless channels with real data obtained during the field experiments.
Graphical representation	The graphical representation makes it easier to represent and share our understanding about the interaction among the random variables.
Small number of samples	Even with a small number of samples, we can do probabilistic reasoning.

Table 3.1.1: Advantages of using Bayesian Networks and Influence Diagram (cont.)

Work with incomplete data	In situations where the system has incomplete data, for example, some of the sensors fail to provide measurements; expected maximization algorithms can still perform reasoning.
Decision- making under uncertainty	We decide which configuration is best to obtain the performance of a determined service profile.

3.2 Building the Model

Building a probabilistic graphical model (PGM) involves having three main elements: variables, causal relationships among variables, and the characterization of those relationships by means of conditional probability distributions (CPD). If we want to assess decisions we make regarding variables of the model, we implement those decisions by giving specific values to the states of the variables that we want to manipulate. Since we need to evaluate the outcome of such decisions to see which one fits the most with our goal, we must represent our preferences by means of utility functions.

This section will explain the process involved in constructing the Bayesian models that I present in this dissertation.

3.2.1 Identification of Variables and their Characterization

The starting point for this stage is knowing what is wanted from the model. In this dissertation, my main intention is to improve the performance of wireless links in the midst of uncertainty and limited conditions. Therefore, I should identify which variables can tell me something about the performance of the channel, called indicators in this dissertation, which variables influence these indicators, and which variables I can manipulate to increase the probability that the indicators will have values representing high performance.

One of the most important indicators in communication links is the bit error rate (BER); therefore, this dissertation focuses on analyzing and characterizing probabilistically how

different factors and parameters affect this indicator. Factors depend on the environment and the operational conditions of the communication system. Factors that affect BER are: noise, co-channel interference, multipath, and propagation losses. Parameters are configuration values or choices that tell the devices, such as transmitter and receivers, how they should operate. The user or designer of the communication system determines the values for the parameters. This dissertation considers the parameters: transmission power, modulation scheme, carrier frequency, and time of symbol.

After identifying the factors, parameters, and indicators, the variables encompassed by each factor and parameter are determined. Some of these variables will be deterministic and some of them random. Parameters are deterministic variables, since the user, the designer, or the system itself chooses their values; therefore, they have certainty about those values. Factors encompass random variables, since they depend upon the environment, and in realistic applications it is not possible to either control or have certainty about the scenarios in which wireless communication systems operate. However, some factors are affected by the parameters. For example, the propagation losses are affected among others by: the carrier frequency, which is a parameter; the distance, which is a random variable in mobile wireless networks; and multipath, which is another factor affected by the environment. Factors affected by parameters are still random. Variables that depend upon deterministic and random variables remain random. Hence, in this dissertation we consider three types of variables: deterministic, random, and influenced random variables.

Knowing which variables we want to incorporate into the probabilistic model we need to determine how to represent them. Representing those variables entails deciding between discrete

or continuous representation. When a variable is discrete, it takes finite states; whereas when the variable is continuous, it takes values from a set of real numbers. In this dissertation we use discrete representation due to the following reasons. First, most of the probabilistic inference algorithms work with discrete variables and the algorithms compatible with continuous variables only work with Gaussian variables or non-Gaussian variables either discretized or approximated as combinations of Gaussian variables. Second, the open source software libraries for probabilistic inference work mostly with discrete variables. Even commercial software packages discretize continuous variables to make them fit with algorithms devoted to discrete variables. Finally, in the practice, variables are not always Gaussian; therefore, in some cases it is necessary to approximate some variables. Additionally, discrete representation of variables fits with discrete probabilistic inference algorithms such as variable elimination, which have shown their maturity and stability.

The space or set of states that a discrete variable can take must be exhaustive and mutually exclusive [72]. *Exhaustiveness* means that the state space comprises all the possible states of the variable. *Mutual exclusiveness* means that no variable must take several states at the same time. For instance, a variable that represents the states of a machine must not have these states:{*working,not_working, working_or_not_working*}. The reason is that the machine taking the state *working_or_not_working* implies that it is simultaneously at the state *working* or the state *not_working*, which is impossible.

The following pages will describe the variables included in the Bayesian model proposed by this dissertation, the goal of which is to improve the performance of wireless communications links. This description includes two ways of sorting these variables: one sorts the variables as

parameters, factors, or indicators; the other one as deterministic, random or influenced random. The description also includes the explanation of the state space of each variable. By the name of each variable, it will appear a shorter name to be used in tables and graphs.

Parameters

As mentioned before, the user, designer, or system chooses the operating parameters; consequently those parameters are deterministic. This dissertation represents parameters as nodes or vertices of a Bayesian network. To represent the parameters as nodes, they are expressed in terms of probability distributions. Because of their deterministic character, the probability distributions only include 1s and 0s. For instance, if we want to choose 10dbm as the transmission power ("Tx") out of these options :{0dBm, 10dBm, 20dBm, 30dBm}; the probability distribution for "TX" is: [0,1,0,0], since we know for sure that "Tx" will be 10dBm, but not 0dbm, 20dBm, or 30dBm.

Transmission power ("Tx"): This dissertation considers four levels of transmission power: 0 dBm, 10 dBm, 20dBm, and 30 dBm. These values are based on the typical levels of power used by systems that operate in the ISM (Industrial, Scientific, and Medical) bands. The variable "Tx" has four states: [Tx_1, Tx_2, Tx_3, Tx_4]; which correspond to [0 dBm, 10 dBm, 20dBm, 30 dBm], respectively. This assignment can be adjusted according to the specific problem. The Bayesian network mostly cares about the probability of a variable being at certain state. The states can be single numbers, intervals of numbers, operating settings, etc. The probability distribution for "Tx" is:

Table 3.2.1: Probability distributions (PD) for Tx. *Use I, II, III, or IV for setting "Tx" at Tx_1,Tx_2,Tx_3, and Tx_4 respectively*.

PD / States	I	II	III	IV
Tx_1	1	0	0	0
Tx_2	0	1	0	0
Tx_3	0	0	1	0
Tx_4	0	0	0	1

Carrier Frequency ("Fc"): For illustration purposes, this dissertation considers three frequencies falling into different regions of the ISM bands: 915 MHz, 2400 MHz, and 5800 MHz. The states of "Fc" assigned to these frequencies in the same order are: [Fc_1, Fc_2, Fc_3]. The probability distribution for "Fc" is:

Table 3.2.2: Probability distributions (PD) for Fc. *Use I, II, or III for setting "Fc" at Fc_1,Fc_2, or Fc_3 respectively.*

PD / States	I	II	III
Fc_1	1	0	0
Fc_2	0	1	0
Fc_3	0	0	1

Modulation Scheme ("Mod"): Three different modulation schemes are used through this dissertation: differential BPSK, differential QPSK, and differential 8PSK, represented as states [2dpsk,4dpsk,8dpsk], respectively. The model proposed in this dissertation can work with other modulation schemes; however, for simplicity and demonstration purposes, only three are used. The probability distribution for the variable "Mod" is:

Table 3.2.3: Probability distributions (PD) for Mod. *Use I, II, or III for setting "Mod" at 2dpsk, 4dpsk, or 8dpsk respectively.*

PD / States	I	II	III	SE
2dpsk	1	0	0	1
4dpsk	0	1	0	2
8dpsk	0	0	1	3

A modulation scheme has a feature called *spectral efficiency* that indicates how many bits per symbol it transmits. The proposed model represents the *spectral efficiency* as "SE" and adds it in the model as a parameter depending on the modulation "Mod". Table 3.2.3 shows the value taken by "SE" for each modulation scheme.

Time of Symbol ("Ts"): The proposed model considers three different symbol times: [1.0μs, 4.0μs, 20.0μs]. The states assigned to those are: [Ts_1, Ts_2, Ts_3]. The symbol time tells the system which transmission bandwidth it should operate in. This document expresses the bandwidth in terms of the symbol time because it makes easier to introduce the concept of narrowband flat fading channel. This dissertation assumes that the wireless system being modeled uses narrowband flat fading channels. The section describing the multipath factor explains better the concept of narrowband flat fading channels. The probability distribution for the variable "Ts" is:

Table 3.2.4: Probability distributions (PD) for Ts. *Use I, II, or III for setting "Ts" at Ts_1,Ts_2 or Ts_3, respectively.*

PD / States	I	II	III
Ts_1	1	0	0
Ts_2	0	1	0
Ts_3	0	0	1

Factors

The factors considered in this dissertation are: *noise, co-channel interference, multipath, and propagation losses*. Each of this factors includes several variables. For instance, co-channel interference includes the variables *interference power* and *signal to interference power.*

1. Noise

Noise Density("N0"): This is the noise power in a bandwidth of 1 MHz expressed in dBm. This factor is mostly influenced by the environment in which the system operates. To determine the values on table 3.2.5, this model uses the equation that defines the noise power P_N in terms of the equivalent noise temperature T_N, and the noise equivalent bandwidth B_N [75]. PN is given by

$$P_N = k \cdot T_N \cdot B_N , \tag{3.2.1}$$

where, $k = 1.38 \times 10^{-23} J/K$ is the Boltzmann's constant. For defining the ranges for the variable "N0", it is considered that $B_N = 1 MHz = 10^6 Hz$. Similarly T_{SYS}, system equivalent noise temperature, replaces T_N. T_{SYS} represents the noise contribution of the receiver system, which includes the antenna, the low noise amplifier, the cable connecting the antenna to the receiver, and the receiver itself. Figure 3.2.1 shows a typical receiver system wherein the equivalent noise temperature of each component contributes to the total equivalent noise temperature of the system.

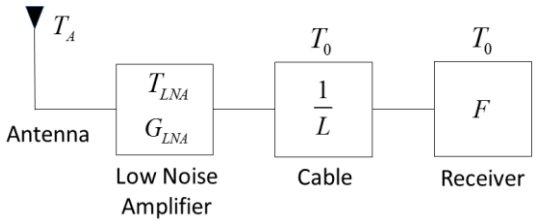


Figure 3.2.1: Typical Receiver System. [75]

The system noise temperature is [75]:

$$T_{SYS} = T_A + T_{LNA} + \frac{(L-1) \cdot T_0}{G_{LNA}} + \frac{L \cdot (F-1) \cdot T_0}{G_{LNA}} , \tag{3.2.2}$$

Where, T_A is the antenna noise temperature, T_{LNA} is the noise temperature of the low noise amplifier (LNA), L is the attenuation of the cable, T_0 is the environment temperature, and F is the noise factor of the receiver. We can consider that T_0 ranges between $290^{\circ}\mathrm{K}$ to $305^{\circ}\mathrm{K}$. T_A represents the noise captured by the antenna, which depends on the radiation of the objects surrounding the antenna and its radiation pattern. In land wireless communications T_A ranges between 50K and 300K; therefore, by using equation 3.2.2 the noise temperature system T_{SYS} is between 155K and 405K. With this T_{SYS} scope, the noise power $P_N = k \cdot T_{SYS} \cdot B_N$ with $B_N = 1\,MHz$ is within the interval $282.9 \times 10^{-14}\,mW$ to $580.2 \times 10^{-14}\,mW$. When expressing these numbers in dBm, the interval becomes [-116.7 dBm , -112.36 dBm], which after being approximated is [-117 dBm, -112 dBm). Table 3.2.5 shows the probability distribution for different subintervals of this range.

Table 3.2.5:Probability distribution for N0. *Spectral noise density relative to 1 MHz and 1 mW.*

State	Interval	Probability
N0_1	[-117,-115.75)	0.25
N0_2	[-115.75,-114.5)	0.25
N0_3	[-114.5,-113.25)	0.25
N0_4	[-113.25,-112)	0.25

Noise Power ("N"): This is noise power, which depends on the noise density "N0" and the bandwidth - expressed in this dissertation in terms of the time of symbol "Ts". The relation between the variables "N0" and "Ts" used to determine the intervals for the states of the variable is:

$$N = N0 - 10 \cdot \log(Ts), \tag{3.2.3}$$

Where, log represents the logarithm to base 10. Equation helps also in eliciting the probability distribution of "N". The details about elicitation of probability distribution will appear in section 3.2.3. This applies to the other variables for which analytical formulas that establish the relation among their parents exist.

Knowing the scope for "N0" and the values assigned to "Ts" allows for the calculation of the scope wherein the variable "N" will be. Then, that scope is divided into intervals. For "N" the number of intervals is 5, namely [N_1, N_2, N_3, N_4, N_5]. The ranges for each interval are: {N_1: [-130.1 , -124); N_2: [-124, -121) ; N_3: [-121,-118); N_4: [-118, - 115); N_5: [-115, -112) }. The criterion to define those intervals was to have smaller intervals in regions considered more critical; therefore, at the lowest level of noise the interval is 6 dB wide, while at the highest levels the intervals are 3 dB wide. Table 3.2.6 shows the CPD for the noise power "N".

Table 3.2.6: Conditional probability distribution (CPD) for N. *Noise power in dBm*

CPD / Parents		N_1	N_2	N_3	N_4	N_5	CPD / Parents		N_1	N_2	N_3	N_4	N_5
N0_1	Ts_1	0	0	0	1	0	N0_3	Ts_1	0	0	0	0	1
N0_1	Ts_2	0	1	0	0	0	N0_3	Ts_2	0	0	1	0	0
N0_1	Ts_3	1	0	0	0	0	N0_3	Ts_3	1	0	0	0	0
N0_2	Ts_1	0	0	0	0.565	0.435	N0_4	Ts_1	0	0	0	0	1
N0_2	Ts_2	0	0.595	0.405	0	0	N0_4	Ts_2	0	0	1	0	0
N0_2	Ts_3	1	0	0	0	0	N0_4	Ts_3	1	0	0	0	0

Energy of Bit to Noise Spectral Density("EbN0"): This variable has as parent the variables: spectral efficiency "SE", received power "Rx", and noise power "N". The relation among these variables is:

$$EbN0 = Rx - N - 10 \cdot \log(SE) . \quad (3.2.4)$$

Equation 3.2.4 is also used in the elicitation of the CPD for "EbN0". The scope of "EbN0" has been divided into 6 intervals, namely [EbN0_1, EbN0_2, EbN0_3, EbN0_4, EbN0_5, EnN0_6]. The ranges for each interval are: {EbN0_1: [-72.8db, 0db); EbN0_2: [0db, 10db); EbN0_3: [10db , 13db); EbN0_4: [13db, 16db); EbN0_5: [16db, 19db); EbN0_6: [19db, 109.1db)}. Notice that the intervals EbN0_2 to EbN0_5 are narrower than the intervals EbN0_1 and EbN0_6. The reason is that the proposed models considers this variable more critical at these intervals; therefore, the need of a better resolution at those regions. An "EbN0" greater than 19 dB is good enough for the receiver to have acceptable performance; therefore, we only need an interval, EbN0_6, to represent when "EbN0" is greater than 19 dB. The maximum EbN0 possible is 109.1 dB, which occurs when Rx is maximum, and N and SE minimum. Table 3.2.7 shows the conditional probability distribution for the variable EbN0.

Table 3.2.7: Conditional probability distribution (CPD) for EbN0. *Bit energy to spectral noise density ratio in dB. This table has 75 different combinations. We show only a portion of the table. The whole table can be found in appendix 1*

Parents			CPD EbN0_1	EbN0_2	EbN0_3	EbN0_4	EbN0_5	EbN0_6
SE_1	Rx_1	N_1	1	0	0	0	0	0
SE_1	Rx_1	N_5	1	0	0	0	0	0
SE_1	Rx_3	N_1	0	0	0	0.02	0.095	0.885
SE_1	Rx_3	N_5	0	0.465	0.165	0.175	0.17	0.025
SE_1	Rx_5	N_1	0	0	0	0	0	1
SE_1	Rx_5	N_5	0	0	0	0	0	1
SE_2	Rx_1	N_1	1	0	0	0	0	0
SE_2	Rx_1	N_5	1	0	0	0	0	0
SE_2	Rx_3	N_1	0	0	0.04	0.17	0.165	0.625
SE_2	Rx_3	N_5	0.015	0.645	0.17	0.155	0.015	0
SE_2	Rx_5	N_1	0	0	0	0	0	1
SE_2	Rx_5	N_5	0	0	0	0	0	1
SE_3	Rx_1	N_1	1	0	0	0	0	0
SE_3	Rx_1	N_5	1	0	0	0	0	0
SE_3	Rx_3	N_1	0	0.01	0.075	0.17	0.16	0.585
SE_3	Rx_3	N_5	0.12	0.655	0.17	0.055	0	0
SE_3	Rx_5	N_1	0	0	0	0	0	1
SE_3	Rx_5	N_5	0	0	0	0	0	1

2. Co-channel Interference

Interference power ("Co_Ch"): This variable represents the intensity of the co-channel interference affecting the receiver. This variable depends upon the environment wherein the receiver operates. The distance between the receiver and other systems operating at the same frequency affects this parameter. The power of the transmitters of co-channel systems also affects this variable. Table 3.2.8 shows the probability distribution for "Co_Ch". I set the intervals shown in this table to have better resolution in the last three intervals. The first interval, CC_1, is the broadest, because the levels of power that it contains are so small that the effect over the C/I is almost negligible; therefore, more precision for this interval is not necessary.

Table 3.2.8: Probability distribution for Co_Ch. *Co_Channel interference relative to 1 mW.*

State	Interval	Probability
CC_1	[-180 -130.1)	0.2
CC_2	[-130.1 -110)	0.2
CC_3	[-110 -95)	0.2
CC_4	[-95 -70)	0.2
CC_5	[-70 -21)	0.2

Signal to Interference Ratio ("C/I"): This variable is the relation between the received power "Rx" and the co-channel interference power "Co_Ch". As they are expressed in dBm their relation is given by:

$$C/I = Rx - Co\_Ch\,. \tag{3.2.5}$$

Equation 3.2.5 is also used in the elicitation of the CPD for "C/I". The scope of this variable splits into 6 intervals: [C/I_1, C/I _2, C/I _3, C/I _4, C/I _5, C/I _6]. The ranges for each interval are: {C/I _1: [-159, 0); C/I _2: [0, 20); C/I _3: [20, 30); C/I _4: [30, 40); C/I _5: [40, 50); C/I _6: [50, 159)} with values given in dB. During the simulations, it was observed that after 50 dB

further increments on "C/I" have low impact on the bit error rate BER; therefore C/I ratios bigger than 50 dB can be considered as part of the same interval. The maximum C/I ratio possible occurs when the Rx level is maximum and the co-channel interference is minimum, in this case, the maximum C/I possible is 159 dB. Table 3.2.9 gives the CPD for "C/I".

Table 3.2.9: Conditional probability distribution (CPD) for C/I. *Signal to co-channel interference ratio in dB.*

Parents \ CPD		C/I_1	C/I_2	C/I_3	C/I_4	C/I_5	C/I_6
Rx_1	CC_1	0.465	0.33	0.105	0.085	0.015	0
Rx_1	CC_2	1	0	0	0	0	0
Rx_1	CC_3	1	0	0	0	0	0
Rx_1	CC_4	1	0	0	0	0	0
Rx_1	CC_5	1	0	0	0	0	0
Rx_2	CC_1	0	0.22	0.195	0.17	0.205	0.21
Rx_2	CC_2	0.57	0.43	0	0	0	0
Rx_2	CC_3	1	0	0	0	0	0
Rx_2	CC_4	1	0	0	0	0	0
Rx_2	CC_5	1	0	0	0	0	0
Rx_3	CC_1	0	0	0.04	0.245	0.17	0.545
Rx_3	CC_2	0	0.59	0.36	0.05	0	0
Rx_3	CC_3	0.525	0.475	0.2	0.2	0.2	
Rx_3	CC_4	1	0	0	0	0	0
Rx_3	CC_5	1	0	0	0	0	0
Rx_4	CC_1	0	0	0	0	0.065	0.935
Rx_4	CC_2	0	0.025	0.2	0.43	0.245	0.1
Rx_4	CC_3	0	0.55	0.31	0.14	0	0
Rx_4	CC_4	0.47	0.5	0.03	0	0	0
Rx_4	CC_5	1	0	0	0	0	0
Rx_5	CC_1	0	0	0	0	0	1
Rx_5	CC_2	0	0	0	0	0.035	0.965
Rx_5	CC_3	0	0	0.01	0.125	0.28	0.585
Rx_5	CC_4	0	0.155	0.185	0.18	0.255	0.225
Rx_5	CC_5	0.51	0.33	0.1	0.06	0	0

3. Multipath

As mentioned in chapter 2, S_τ ,the time delay between the arrival of the first and last components of the same symbol, tells how much a wireless channel stretches a symbol in time [2]. The relation between the delay spread and the time of symbol *Ts* will determine if the channel behaves as flat fading or selective fading channel. A *flat fading channel* responds uniformly over the bandwidth of the transmitted signal, whereas a *selective channel* behaves differently depending on the frequency even within the bandwidth of the signal.

Figure 3.2.2 shows the behavior of a flat fading channel, also called narrowband channel. This behavior happens when $Ts >> S_\tau$. Notice that under this condition the multipath components of the same symbol arrive within the same delay bin making impossible for the

receiver to distinguish them. Hence, the receiver sees those components as one. However, the constructive and destructive interference among the multipath components affect the amplitude of the received symbol.

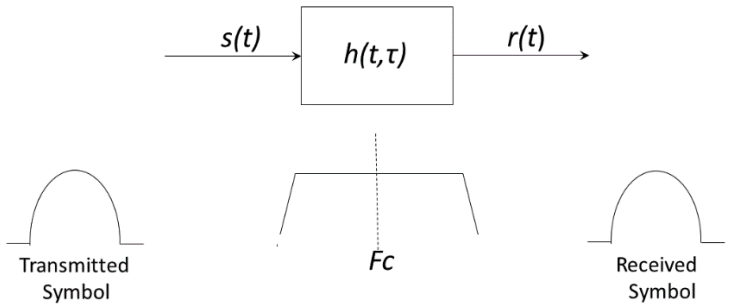


Figure 3.2.2: Flat Fading Channel.

In contrast, if $Ts < S_\tau$, the channel exhibits a frequency selective behavior as shown in figure 3.2.3. In a selective fading channel, also called wideband channel, the multipath components arrive at different delay bins. This causes components of a symbol to interfere with adjacent symbols, something called inter symbol interference (ISI). In order to counteract the ISI caused by multipath propagation, one common technique is the use of multiple narrowband channels instead of a single wideband channel. This dissertation assumes that the symbol time *Ts* has a such value that the channel is narrowband; therefore, the delay spread is not considered, since the channel satisfies the condition $Ts >> S_\tau$. Nevertheless, the other variables related with multipath, such as Doppler frequency and Doppler phase shift should be considered because they affect flat as well as selective channels.

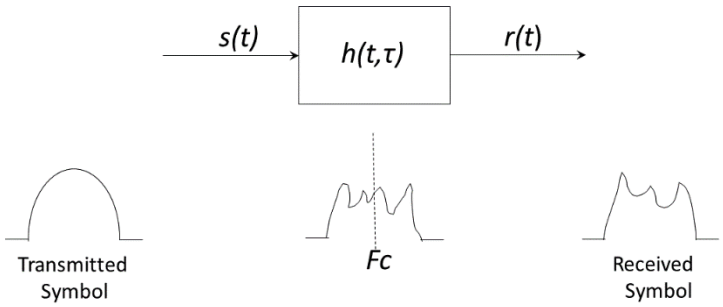


Figure 3.2.3: Selective Fading Channel

Doppler Frequency Shift ("Fd"): This variable has three different states: [Fd_1, Fd_2, Fd_3], the ranges of which are: [0- 169.5), [169.5, 444.5), and [444.5, 1075), with values given in Hz. Those ranges depend upon the carrier frequency "Fc", the speed and the angle between the trajectories of transmitter and receiver. For simplicity, this work assumes a constant speed of 100 Km, and uniform distribution of the angle within the range [0, 2π). The expression to obtain the ranges and CPD for "Fd" is:

$$Fd = \frac{\Delta v \cdot F_c \cdot \cos(\theta)}{c}, \tag{3.2.6}$$

Where, Δv is the speed difference between the transmitter and the receiver, F_c is the carrier frequency, θ is the angle formed by the trajectories of the transmitter and receiver, and c is the speed of light. The ranges for "Fd" indicated previously resulted from evaluating equation 3.2.6 at each carrier frequency, assuming that the speed difference is maximum; $\Delta = 200 Km/h$ - and θ is either 0 or π, so that $\cos(\theta) = 1$. This assumption yields the maximum Doppler frequency shift possible at each carrier frequency. The conditional probability table (CPT) for "Fd" is:

Table 3.2.10: Conditional probability distribution (CPD) for Fd. *Doppler frequency shift in Hz.*

CPD / Parents	Fd_1	Fd_2	Fd_3
Fc_1	1	0	0
Fc_2	0.5	0.5	0
Fc_3	0.33	0.33	0.33

To obtain table 3.2.10 we need to notice that if Fc=Fc_1, Fd falls for sure into the interval Fd_1; therefore the probability distribution is [1, 0, 0]. In case Fc=Fc_2, Fd is in either in Fd_1 or Fd_2 with equal probability, since the angle θ is uniformly distributed; therefore, the probabilities are: [0.5, 0.5, 0]. For the same reason, if Fc=Fc_3 the probabilities are: [0.33, 0.33, 0.33].

Doppler Phase Shift ("Dop_Phi"): The Doppler phase shift is the change in phase accumulated during the transmission of one symbol with duration "Ts". The Doppler frequency shift represents the rate of change of the phase. Therefore, the Doppler phase shift is given by [2]

$$Dop\_Phi = \phi_D = \int_{Ts} 2\pi Fd \cdot dt \,. \tag{3.2.7}$$

Since we consider "Fd" constant during the symbol, equation 3.2.7 becomes:

$$Dop\_Phi = \phi_D = 2\pi \cdot Fd \cdot Ts \,. \tag{3.2.8}$$

Equation 3.2.8 determines the ranges for "Dop_Phi" as well as its CPD. Table 3.2.11 shows the CPD for "Dop_Phi". The intervals Phi_1, Phi_2, and Phi_3 corresponds to [0, 0.05), [0.05, 0.1), and [0.1, 0.136) respectively. The numbers in these intervals are in radians.

Table 3.2.11: Probability distribution for Dop_Phi. *Doppler phase shift in radians.*

CPD / Parents		Phi_1	Phi_2	Phi_3
Fd_1	Ts_1	1	0	0
Fd_1	Ts_2	1	0	0
Fd_1	Ts_3	1	0	0
Fd_2	Ts_1	1	0	0
Fd_2	Ts_2	1	0	0
Fd_2	Ts_3	0.855	0.145	0
Fd_3	Ts_1	1	0	0
Fd_3	Ts_2	1	0	0
Fd_3	Ts_3	0	0.595	0.405

4. Propagation Losses

Distance ("Dist"): Distance is one of the parameters that influences the propagation losses. For this variable it is assumed that the distance between the transmitter and the receiver falls within some of the four intervals shown in table 3.2.12. We can imagine that the transmitter (TX) is at the center of the innermost circle shown in figure 3.2.4, and that the receiver (RX) can be within any of the other regions circumscribed by the outer circles. The probability distribution in table 3.2.12 depends upon the pattern of movement of transmitter and receiver.

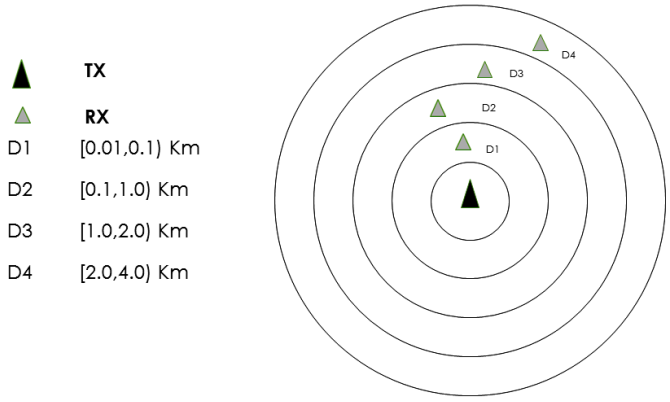


Figure 3.2.4: Distance ranges. *Distances are in Kilometers.*

Table 3.2.12: Probability distribution for Dist. *Distance in Kilometers.*

State	Interval	Probability
D_1	[0.01 0.1)	0.25
D_2	[0.1 1.0)	0.25
D_3	[1.0 2.0)	0.25
D_4	[2.0 4.0)	0.25

Table 3.2.12 shows a uniform probability distribution; however, this distribution will change depending on the specific pattern of movement of the wireless nodes. There are different ways to update the probability distribution of this variable: From GPS information transmitted in each packet by the communicating nodes; from probabilistic inference performed over the Bayesian network explained in this chapter; or from a combination of both. Nowadays, location information provided by GPS systems is being exploited in many fields, such as transportation [76], commerce, and tracking [77] . Similarly, in the field of wireless networks, location information exchanged among the nodes can serve to calculate the distance separating those nodes, and estimate the movement patterns.

Free Space Losses ("FSL"): This variable represents the power loss in free space. The parents of this variable are the distance "Dist" and the carrier frequency "Fc". I divided the scope of this variables into four intervals defined as: {FSL _1: [51, 82); FSL _2: [82, 107); FSL _3: [107, 114); FSL _4: [114,120)}. The expression how "Dist" and "Fc" influence "FSL" is:

$$FSL = 32.45 + 20 \cdot \log(Dist) + 20 \cdot \log(Fc) . \qquad (3.2.9)$$

Equation 3.2.9 also helps in determining the scope of "FSL" and the CPD. The intervals FSL_3 and FSL_4 where made narrower, since at these levels FSL is more critical; we require more resolution. Table 3.2.13 shows the CPD for "FSL" with values in dB.

Table 3.2.13: Probability distribution for FSL. *Free space losses in dB.*

Parents / CPD		FSL_1	FSL_2	FSL_3	FSL_4	Parents / CPD		FSL_1	FSL_2	FSL_3	FSL_4
D_1	Fc_1	1	0	0	0	D_3	Fc_1	0	1	0	0
D_1	Fc_2	1	0	0	0	D_3	Fc_2	0	1	0	0
D_1	Fc_3	0.47	0.53	0	0	D_3	Fc_3	0	0	1	0
D_2	Fc_1	0.235	0.765	0	0	D_4	Fc_1	0	1	0	0
D_2	Fc_2	0.025	0.975	0	0	D_4	Fc_2	0	0.125	0.875	0
D_2	Fc_3	0	0.9	0.1	0	D_4	Fc_3	0	0	0.02	0.98

Multipath Fading ("Fade"): This variable represents the fading of the power caused by the combination of multiple versions of the same signal that arrive at the receiver with different phases and amplitudes. These multiple versions of the signal come from reflectors and scatterers present in the environment where the signal propagates. This variable also includes the shadowing due to obstacles. Therefore, this variable depends on the environment. The scope for "Fade" has been defined between 0db to 60 dB, which can cover the range that this variable could have in a realistic scenario. This scope is divided into three intervals, namely {Sh _1: [0, 5); Sh _2: [5, 15); Sh _3: [15, 60)}, with values in dB. The letter "Sh" in the names of the intervals come from *shadowing*, included in this variable.

Table 3.2.14: Probability distribution for Fade. *Multipath fading or shadowing due to obstacles expressed in dB.*

State	Interval	Probability
Sh_1	[0 5)	0.33
Sh_2	[5 15)	0.33
Sh_3	[15 60)	0.33

Total Losses ("Attn"): This variable represents the total propagation losses in the system, which are the summation of two main components: free space losses "FSL" and multipath fading, represented here as "Fade". The scope of this variable is divided into four intervals: {Attn _1:

[51, 88.25); Attn _2: [88.25, 119.25); Attn _3: [119.25, 150.75); Attn _4: [150.75, 180.0)}, with values given in dB. The influence of "FSL" and "Fade" over "Attn" is determined by

$$Attn = FSL + Fade \tag{3.2.10}$$

Table 3.2.15 shows the CPD for this variable.

Table 3.2.15: Conditional probability distribution (CPD) for Attn. *Total attenuation expressed in dB.*

CPD / Parents		Attn_1	Attn_2	Attn_3	Attn_4	CPD / Parents		Attn_1	Attn_2	Attn_3	Attn_4
FSL_1	Sh_1	1	0	0	0	FSL_3	Sh_1	0	1	0	0
FSL_1	Sh_2	0.865	0.135	0	0	FSL_3	Sh_2	0	0.375	0.625	0
FSL_1	Sh_3	0.185	0.585	0.23	0	FSL_3	Sh_3	0	0	0.54	0.46
FSL_2	Sh_1	0.14	0.86	0	0	FSL_4	Sh_1	0	0.405	0.595	0
FSL_2	Sh_2	0.005	0.98	0.015	0	FSL_4	Sh_2	0	0.005	0.995	0
FSL_2	Sh_3	0	0.2	0.665	0.135	FSL_4	Sh_3	0	0	0.41	0.59

Received Power (Rx): This variable represents the power at the receiver, which depends upon the transmission power "Tx" and the total propagation losses "Attn" according to this expression:

$$Rx = Tx - Attn \tag{3.2.11}$$

We can calculate the minimum and maximum value of *Rx* with the expression 3.2.11 and minimum and maximum values of *Tx* and *Attn*. For instance, to obtain the minimum value of *Rx* we evaluate that expression with the minimum value of *Tx* and the maximum value of *Attn*, which yields -180.0 dBm. Similarly, the expression 3.2.11 evaluated at the maximum *Tx* and the minimum *Attn* produces the maximum value of *Rx*, -21.0 dBm.

The extent of this variable, -180 dBm to -21 dBm, has been divided into five intervals: {Rx _1: [-180.0, -130.1); Rx _2: [-130.1, -121.0); Rx _3: [-121.0, -100.0); RX _4: [-100.0, -80.0); RX _5: [-80.0, -21.0)}, with values in dBm. Table 3.2.16 shows the CPD for the variable "Attn", elicited with the assistance of the expression 3.2.11.

Table 3.2.16: Conditional probability distribution (CPD) for Rx. *Received power expressed dBm.*

Parents \ CPD		RX_1	RX_2	RX_3	RX_4	RX_5	Parents \ CPD		RX_1	RX_2	RX_3	RX_4	RX_5
Attn_1	Tx_1	0	0	0	0.215	0.785	Attn_3	Tx_1	0.645	0.285	0.07	0	0
Attn_1	Tx_2	0	0	0	0	1	Attn_3	Tx_2	0.275	0.31	0.415	0.2	0.2
Attn_1	Tx_3	0	0	0	0	1	Attn_3	Tx_3	0.03	0.315	0.605	0.05	0.2
Attn_1	Tx_4	0	0	0	0	1	Attn_3	Tx_4	0	0	0.65	0.35	0
Attn_2	Tx_1	0	0	0.65	0.35	0	Attn_4	Tx_1	1	0	0	0	0
Attn_2	Tx_2	0	0	0.355	0.6	0.045	Attn_4	Tx_2	1	0	0	0	0
Attn_2	Tx_3	0	0	0	0.66	0.34	Attn_4	Tx_3	1	0	0	0	0
Attn_2	Tx_4	0	0	0	0.355	0.645	Attn_4	Tx_4	0.68	0.3	0.02	0	0

Indicator

An indicator is a variable that tells how the system is performing. It is also an influenced random variable. In the communication systems arena, the bit error rate (BER) is the most prevalent indicator of wireless, wired and optical channels. BER is a random variable; however, by manipulating some of the factors that influence it, it is possible to increase the probability of this variable going or staying into the state that the systems needs it to be.

Bit Error Rate("BER"): The bit error rate is defined as the number of erroneous bits divided by the total number of transmitted bits. This variable has as parents: "EbN0", "C/I", "Dop_Phi", and "Mod". For simplicity, this dissertation uses a different conditional probability distribution (CPD) per each modulation scheme. The Bayesian network model will load the CPD corresponding to the modulation scheme it is using. Therefore, these CPDs will have only three parents: "EbN0", "C/I", and "Dop_Phi". "BER" extents from 0 (0% erroneous bits) to 1 (100% erroneous bits). This scope contains five intervals:{BER_1: $[0, 10^{-5})$; BER_2: $[10^{-5},10^{-3})$; BER_3: $[10^{-3}, 10^{-2})$; BER_4: $[10^{-2}, 10^{-1})$; BER_5: $[10^{-1},1)$}, where the numbers have no units.

Due to the lack of an analytical or empirical expression describing how the variables "EbN0", "C/I", and "Dop_Phi" impact "BER" for different modulation schemes, simulation of a

wireless communication system with different modulation schemes is necessary in order to elicit the CPDs of this variable. For demonstration purposes, this dissertation considers three different modulation schemes: differential binary shift keying (DBPSK), differential quaternary shift keying (DQPSK), and differential eight phase shift keying (D8PSK). This method applies to other modulation schemes as well. Tables 3.2.16 to 3.2.18 show fragments of the CPDs for DBPSK, DQPSK, and D8PSK modulation schemes. These tables show only fragments, since each CPD has 108 different combinations (6 states for "EbN0" times 6 states for "C/I" times 3 states for "Dop_Phi"). Appendix 1 contains the whole CPDs of "BER" for the aforementioned modulation schemes; section 3.2.3 describes the details of the elicitation of these CPDs.

Table 3.2.17: Conditional probability distribution (CPD) for BER in DBPSK. *The CPD for BER has 108 different combinations. This tables only shows the first and the last set of combinations. The whole table can be found in appendix 1 as Table A.1.1.*

Parents			CPD BER_1	BER_2	BER_3	BER_4	BER_5
EbN0_1	C/I_1	Phi_1	0	0	0	0	1
EbN0_1	C/I_5	Phi_2	0	0	0	0	1
EbN0_1	C/I_6	Phi_3	0	0	0	0	1
EbN0_2	C/I_1	Phi_1	0	0	0	0	1
EbN0_2	C/I_5	Phi_2	0	0.09	0.17	0.46	0.28
EbN0_2	C/I_6	Phi_3	0	0.105	0.19	0.445	0.26
EbN0_3	C/I_1	Phi_1	0	0	0	0	1
EbN0_3	C/I_5	Phi_2	0.645	0.355	0	0	0
EbN0_3	C/I_6	Phi_3	0.635	0.365	0	0	0
EbN0_4	C/I_1	Phi_1	0	0	0	0	1
EbN0_4	C/I_5	Phi_2	0.995	0.005	0	0	0
EbN0_4	C/I_6	Phi_3	1	0	0	0	0
EbN0_5	C/I_1	Phi_1	0	0	0	0	1
EbN0_5	C/I_5	Phi_2	1	0	0	0	0
EbN0_5	C/I_6	Phi_3	1	0	0	0	0
EbN0_6	C/I_1	Phi_1	0	0	0	0	1
EbN0_6	C/I_5	Phi_2	1	0	0	0	0
EbN0_6	C/I_6	Phi_3	1	0	0	0	0

Table 3.2.18: Probability distribution for BER in DQPSK . *The CPD for BER has 108 different combinations. This tables only shows the first and the last set of combinations. The whole table can be found in appendix 1 as Table A.1.2.*

Parents			CPD BER_1	BER_2	BER_3	BER_4	BER_5
EbN0_1	C/I_1	Phi_1	0	0	0	0	1
EbN0_1	C/I_5	Phi_2	0	0	0	0	1
EbN0_1	C/I_6	Phi_3	0	0	0	0	1
EbN0_2	C/I_1	Phi_1	0	0	0	0	1
EbN0_2	C/I_5	Phi_2	0	0	0	0.46	0.54
EbN0_2	C/I_6	Phi_3	0	0	0	0.465	0.535
EbN0_3	C/I_1	Phi_1	0	0	0	0	1
EbN0_3	C/I_5	Phi_2	0	0.03	0.86	0.11	0
EbN0_3	C/I_6	Phi_3	0	0.005	0.82	0.175	0
EbN0_4	C/I_1	Phi_1	0	0	0	0	1
EbN0_4	C/I_5	Phi_2	0.125	0.84	0.035	0	0
EbN0_4	C/I_6	Phi_3	0.055	0.83	0.115	0	0
EbN0_5	C/I_1	Phi_1	0	0	0	0	1
EbN0_5	C/I_5	Phi_2	0.875	0.125	0	0	0
EbN0_5	C/I_6	Phi_3	0.735	0.265	0	0	0
EbN0_6	C/I_1	Phi_1	0	0	0	0	1
EbN0_6	C/I_5	Phi_2	1	0	0	0	0
EbN0_6	C/I_6	Phi_3	1	0	0	0	0

Table 3.2.19: Probability distribution for BER in D8PSK. *The CPD for BER has 108 different combinations. This tables only shows the first and the last set of combinations. The whole table can be found in appendix 1 1 as Table A.1.3.*

Parents			CPD BER_1	BER_2	BER_3	BER_4	BER_5
EbN0_1	C/I_1	Phi_1	0	0	0	0	1
EbN0_1	C/I_5	Phi_2	0	0	0	0	1
EbN0_1	C/I_6	Phi_3	0	0	0	0	1
EbN0_2	C/I_1	Phi_1	0	0	0	0	1
EbN0_2	C/I_5	Phi_2	0	0	0	0.135	0.865
EbN0_2	C/I_6	Phi_3	0	0	0	0.095	0.905
EbN0_3	C/I_1	Phi_1	0	0	0	0	1
EbN0_3	C/I_5	Phi_2	0	0	0	1	0
EbN0_3	C/I_6	Phi_3	0	0	0	1	0
EbN0_4	C/I_1	Phi_1	0	0	0	0	1
EbN0_4	C/I_5	Phi_2	0	0	0.145	0.855	0
EbN0_4	C/I_6	Phi_3	0	0	0	1	0
EbN0_5	C/I_1	Phi_1	0	0	0	0	1
EbN0_5	C/I_5	Phi_2	0	0.08	0.92	0	0
EbN0_5	C/I_6	Phi_3	0	0	0.855	0.145	0
EbN0_6	C/I_1	Phi_1	0	0	0	0	1
EbN0_6	C/I_5	Phi_2	0.96	0.04	0	0	0
EbN0_6	C/I_6	Phi_3	0.955	0.045	0	0	0

3.2.2 Identification of the Structure of the Model

After identifying the variables of the model, the next step is to establish how they influence one another. In the case of wireless communication systems, there is a set of analytical and

empirical expressions that relate different variables. I have already used some of these expressions through the last section to determine the ranges of some variables. I can use the same expressions to establish causal relations among the variables of my proposed model. However, in some cases there are not analytical expressions that consider how a variable depends upon others. For instance, there are not analytical expressions telling how the bit error rate ("BER") depends on the combination of: "Mod" ,"EbN0", "C/I", and "Dop_Phi". Although, I lack such expressions, I know that "Mod", "EbN0", "C/I", and "Dop_Phi" impact "BER". Therefore, I have the information to determine the structure of a local Bayesian network containing "BER" and its parents: "Mod", "EbN0", "C/I", and "Dop_Phi". Figure 3.2.5 shows that Bayesian network.

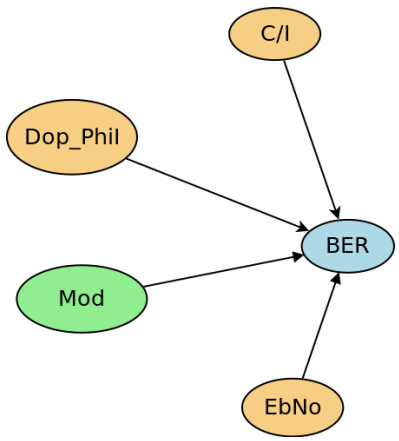


Figure 3.2.5: Local Bayesian network for "BER". *Green color means the variable is a parameter, brownish color the variable is an influenced random variable, and blue color the variable is an indicator.*

I have started with the "BER" because this is the indicator for the Bayesian model under construction. That model is intended for the system to perform probabilistic inference on the variables that impact "BER", so that it can manipulate its parameters to reduce "BER". The

manipulation of the parameters will be based on the principle of maximum expected utility (MEU), explained in chapter 2.

After determining which the parents of "BER" are, I need to find the parents of the parents of "BER". Then I find the parents of these parents and so on. In summary, to figure out the structure of the Bayesian I start by finding the parents of the variable of interest, "BER" in this case", and then proceed to do the same with the parent variables all the way until I find the root variables. Figures 3.2.5 through 3.2.8 show how the Bayesian model was built gradually.

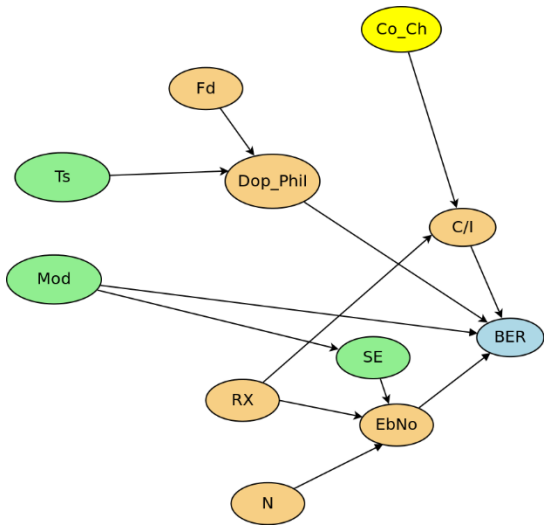


Figure 3.2.6: Bayesian network for "BER" extended one generation back. *Green color means the variable is a parameter, yellow color the variable is a random variable, brownish color the variable is an influenced random variable, and blue color the variable is an indicator.*

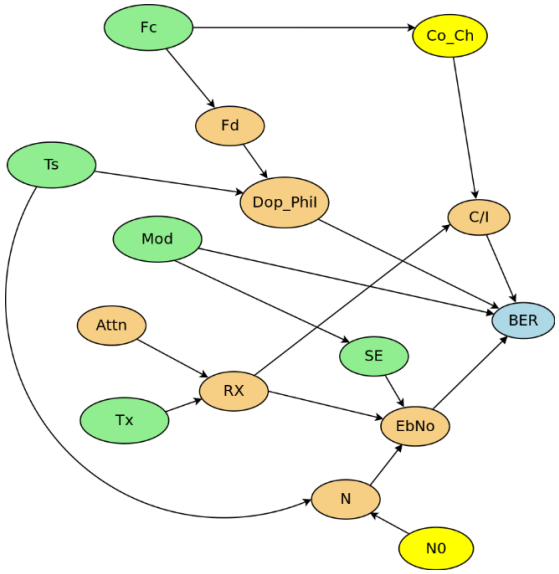


Figure 3.2.7: Bayesian network for "BER" extended two generations back. *Green color means the variable is a parameter, yellow color the variable is a random variable, brownish color the variable is an influenced random variable, and blue color the variable is an indicator.*

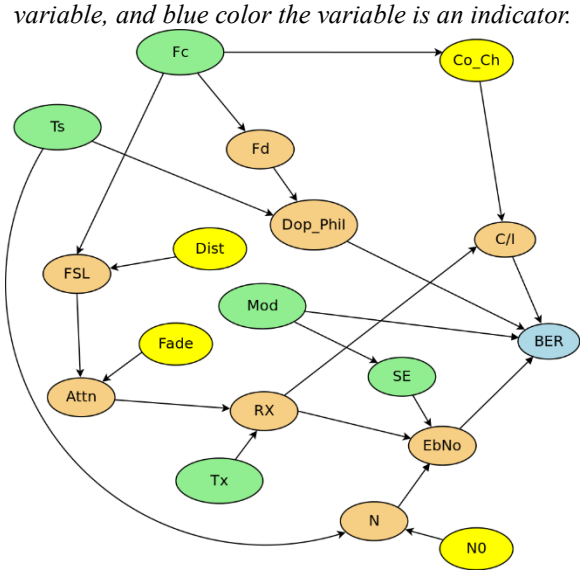


Figure 3.2.8: Bayesian network for "BER" extended three generations back. *Green color means the variable is a parameter, yellow color the variable is a random variable, brownish color the variable is an influenced random variable, and blue color the variable is an indicator.*

3.2.3 Elicitation of the Conditional Probability Distributions

To elicit the conditional probability distributions (CPD), I take each variable and analyze how its probability distribution behaves per each different combination of its parents. If the

variable is parentless, I analyze how its behavior could look like in a particular scenario. Such analysis can be based on experience, experiments, or simulations. In some cases, rough estimates of the probability distributions are good enough, since the Bayesian model will update these distributions as it gets evidence and learns through experience. Therefore, in the case of the variables "Co_Ch", "Dist", "Fade", and "N0" I start the model with estimates of their probability distributions (PD). When the system receives evidence, in this case the state of "BER", the Bayesian model performs probabilistic inference on these variables to update their PD (probability distribution).

When the variable has parents, I take each combination of the states of its parents and find the probability of that variable being in each state. To find those probabilities, I need an expression or function telling which state the variable will take depending on the combination of the parents. If the states of the variables were single numbers, this process would come down to simply evaluating the functions. However, in the practice, variables Bayesian models have variables with states defined either as single numbers or as intervals. Therefore, the process of eliciting the numbers entails more than evaluating functions: it requires to take multiple samples per each interval, for each possible combination of parent variables and using the function to find in which of its interval the child variable falls. The steps to estimate the CPDs of the Bayesian model random variables are the following:

1. Build a local Bayesian network with the variable for which I want to elicit the CPD. Figure 3.2.6 e) shows the local BN for Rx.
2. Enumerate all the possible combinations of parent variable states. The total number of combinations amounts to the product of number of states of the parent variables. For

example, “Tx” and “Attn”, the parents of “Rx” have 5 and 4 states, respectively; hence the number of total combinations is 20, as seen in table 3.2.16.

3. Find a function that expresses the child variable in terms of its parents. Equations 3.2.3 to 3.2.11 from the previous section are used in this step. When I lack such kind of expression, like in the case of the variable “BER”, I must create a simulation able to take samples of the parent variables per each possible combination and yield a sample of the child variable.
4. Obtain random samples of all the intervals that define the states of the variables. The functions or simulations mentioned in step 3 will take samples, i.e. single numbers, per each interval. The more samples per interval I take, the more precise the CPDs will be. Continuing with the variable “Rx”, I need to take N samples from each of the 5 intervals of “Tx” and from each of the 4 intervals of “Attn”. Then, I evaluate the function or run the simulation per each possible combination of samples of “Tx” and “Attn”. In other words, I evaluate the function or run the simulation N times the total amount of combinations of the parent states. In the case of “Tx” and “Attn” I evaluate function 3.2.2 N X 5 X 4 times.
5. Out of the previous step I obtained a matrix of samples: one row per each time I evaluated the function or ran the simulation; one column per each parent and one for the child variable. In the case of “Rx”, I have three columns: two for “Tx” and “Attn” and one for “Rx”.
6. The matrix of data from step 5 must be discretized, since the CPDs of the Bayesian network are in terms of the states of the variables not in terms of single numbers. To

discretize the data means to replace each sample by the name of the state to which it belongs. The discretization process uses the definition of ranges shown in section 3.2.1.

7. Finally, I use maximum likelihood estimation (MLE) [69] to learn the CPD of the child variable using the data obtained in step 6 and the structure (BN) from step 1.

Figure 3.2.9 shows the local Bayesian networks used to elicit the CPDs for the variables of the model. Appendix 2 contains the python routines used for the elicitation of the CPDs. Chapter 4 explains the simulation performed to obtain the CPD for "BER".

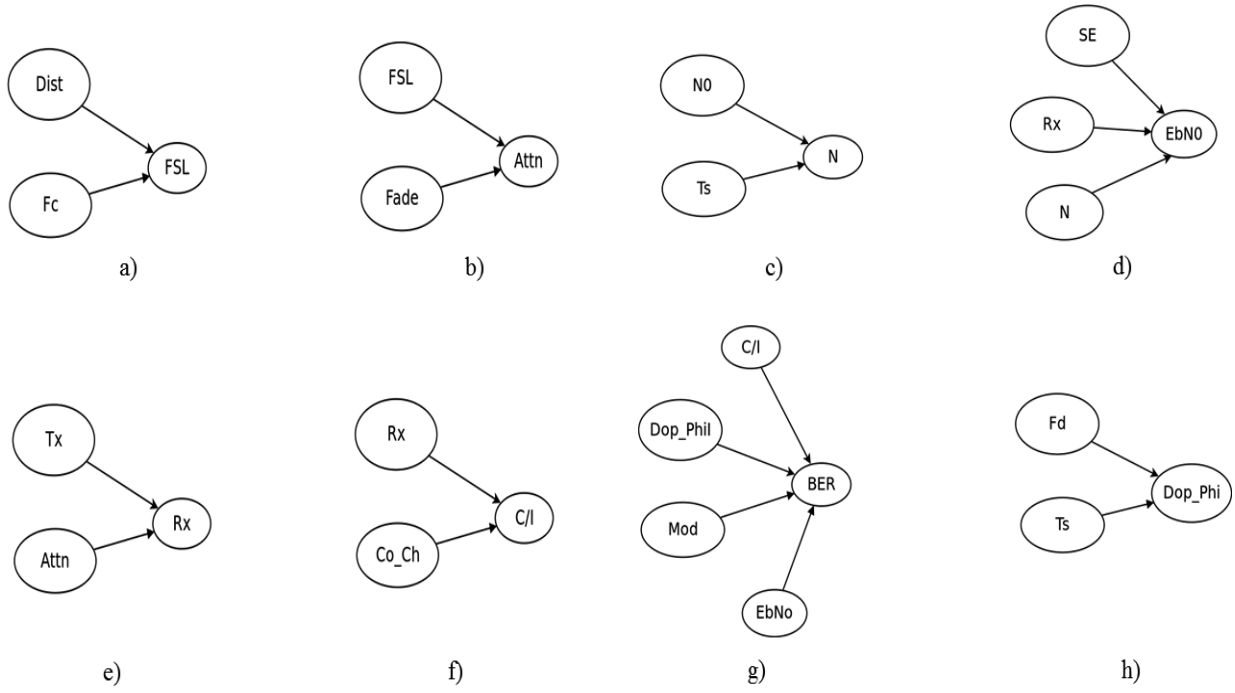


Figure 3.2.9: Local Bayesian Networks used for eliciting the conditional probability distributions (CPD). *a) BN for "FSL", b)BN for "Attn", c)BN for "N", d)BN for EbN0, e) BN for "Rx", f) BN for "C/I", g) BN for "BER", and h) BN for "Dop_Phi".*

3.2.4 The Utility Function

The main goal of the Bayesian model being presented in this dissertation is to assist the cognitive radio in making decisions that satisfy the best the requirements of the wireless

communication system. A utility function ("UF") provides a means to represent the preferences of the system making decisions. The utility function takes as arguments the variables over which I want to express my preference. The utility function "UF" has as its arguments the indicators, "BER" in this case, parameter variables, and other random or influenced random variables that can help in assessing the convenience of certain configuration for the system given the probabilistic distribution of the indicator. I can express a "UF" either as a function or as a table that assigns a number to each possible combination of the states of its arguments: it must assign higher numbers to the more convenient combinations, and lower or negative numbers to the less convenient or inconvenient combinations. In my model, the arguments of "UF" are: the indicator "BER", and the parameters "Tx" and "SE". Table 3.2.20 shows the utility function.

Table 3.2.20: Utility function. *The numbers are dimensionless.*

Arguments of UF			Utility	Arguments of UF			Utility
Tx_1	SE_1	BER_1	60	Tx_3	SE_1	BER_1	80
Tx_1	SE_1	BER_2	10	Tx_3	SE_1	BER_2	70
Tx_1	SE_1	BER_3	-10	Tx_3	SE_1	BER_3	-20
Tx_1	SE_1	BER_4	-15	Tx_3	SE_1	BER_4	-30
Tx_1	SE_1	BER_5	-20	Tx_3	SE_1	BER_5	-40
Tx_1	SE_2	BER_1	80	Tx_3	SE_2	BER_1	75
Tx_1	SE_2	BER_2	20	Tx_3	SE_2	BER_2	70
Tx_1	SE_2	BER_3	-5	Tx_3	SE_2	BER_3	-25
Tx_1	SE_2	BER_4	-10	Tx_3	SE_2	BER_4	-35
Tx_1	SE_2	BER_5	-15	Tx_3	SE_2	BER_5	-35
Tx_1	SE_3	BER_1	100	Tx_3	SE_3	BER_1	80
Tx_1	SE_3	BER_2	30	Tx_3	SE_3	BER_2	75
Tx_1	SE_3	BER_3	-3	Tx_3	SE_3	BER_3	-20
Tx_1	SE_3	BER_4	-6	Tx_3	SE_3	BER_4	-30
Tx_1	SE_3	BER_5	-10	Tx_3	SE_3	BER_5	-40
Tx_2	SE_1	BER_1	50	Tx_4	SE_1	BER_1	60
Tx_2	SE_1	BER_2	5	Tx_4	SE_1	BER_2	55
Tx_2	SE_1	BER_3	-15	Tx_4	SE_1	BER_3	-35
Tx_2	SE_1	BER_4	-20	Tx_4	SE_1	BER_4	-40
Tx_2	SE_1	BER_5	-25	Tx_4	SE_1	BER_5	-60

Table 3.2.20: Utility function. *The numbers are dimensionless*. (Cont.)

Tx_2	SE_2	BER_1	75	Tx_4	SE_2	BER_1	70
Tx_2	SE_2	BER_2	15	Tx_4	SE_2	BER_2	65
Tx_2	SE_2	BER_3	-10	Tx_4	SE_2	BER_3	-20
Tx_2	SE_2	BER_4	--15	Tx_4	SE_2	BER_4	-30
Tx_2	SE_2	BER_5	-20	Tx_4	SE_2	BER_5	-40
Tx_2	SE_3	BER_1	90	Tx_4	SE_3	BER_1	50
Tx_2	SE_3	BER_2	85	Tx_4	SE_3	BER_2	40
Tx_2	SE_3	BER_3	0	Tx_4	SE_3	BER_3	-40
Tx_2	SE_3	BER_4	-5	Tx_4	SE_3	BER_4	-45
Tx_2	SE_3	BER_5	-15	Tx_4	SE_3	BER_5	-55

In this case the cognitive radio (CR) makes decisions as of how to adjust its parameters to increase the probability that the bit error rate "BER" is at lower levels, states BER_1 and BER_2. Accomplishing this with low power, Tx_1, or Tx_2, and high spectral efficiency, SE_3, is preferable to accomplishing it with higher power and low spectral efficiency; for instance, the combination [Tx_1, SE_3,BER_1] has bigger utility than the combination [Tx_4, SE_1, BER_1], table 3.2.20. Table 3.2.20 also shows that combinations with high values of BER, BER_4, BER_5, have negative utility. I can incorporate the "UF" into the model by adding it as a node to the Bayesian network, which turn the BN into an influence diagram (ID). Figure 3.2.10 shows the ID proposed to assist the CR in adjusting its parameters to improve the performance of the wireless communication system.

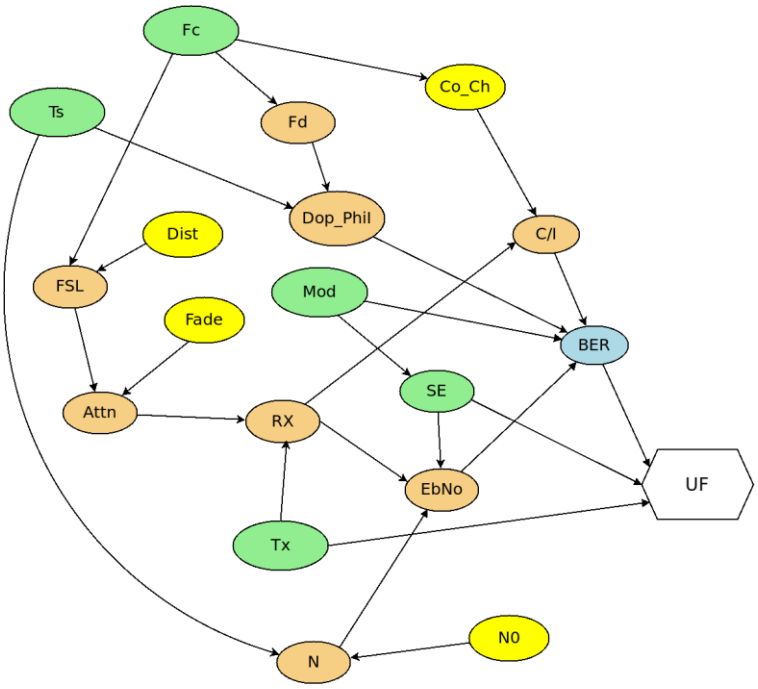


Figure 3.2.10: Influence diagram for decision making. *Green color means the variable is a parameter, yellow color the variable is a random variable, brownish color the variable is an influenced random variable, and blue color the variable is an indicator. The hexagon represents the utility function "UF".*

CHAPTER 4

SIMULATIONS OF THE BAYESIAN MODELS FOR WIRELESS COMMUNICATION SYSTEMS

This chapter describes the simulations of the proposed Bayesian model. The first section describes the simulation performed to obtain the CPD of different modulation schemes. The second section presents a simulation of how the Bayesian network probabilistically reasons on random variables of a wireless system. Finally, the third section explains how the Bayesian model assists the wireless communication system in selecting the configuration with the highest expected utility.

4.1 Simulation of Digital Modulation Schemes to Obtain the CPD of BER

As mentioned in chapter 3, I lack an empirical or analytical function able to express the bit error rate BER in terms of EbN0, Dop_Phi, and C/I, which I could use to generate data to elicit its conditional probability distribution. Hence, I need to simulate a modulator-demodulator set for each of the modulation scheme I want to consider. For demonstration purposes this dissertation includes the modulation schemes: differential BPSK (2dpsk); differential QPSK (4dpsk), and differential 8PSK (8dpsk).

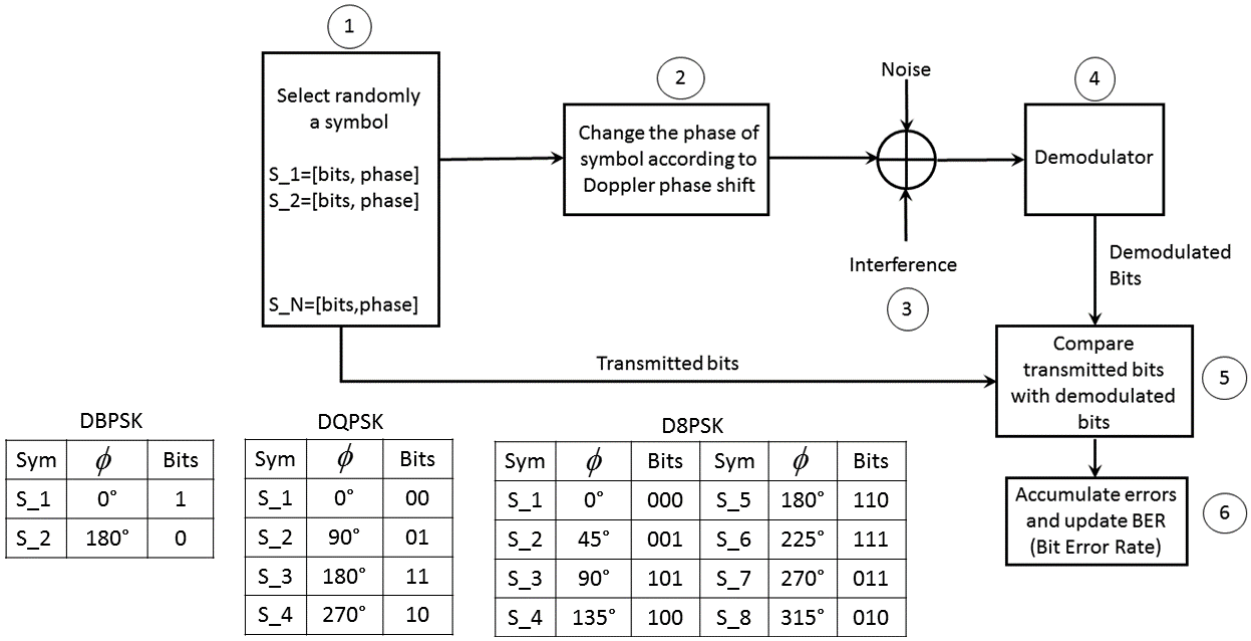
4.1.1 Methodology

In this dissertation, I consider that the bit error rate BER is affected by EbN0, C/I, and Dop_Phi. In the previous chapter, I defined the state domain for each of the parent variables

EbN0, C/I, and Dop_Phi. Because these variables affect BER simultaneously, I must simulate the modulator-demodulator link at each possible combination of the parent variables to observe the state taken by BER under that combination. For every possible combination of EbN0, C/I, and Dop_Phi I generate N samples from the intervals represented by the states of the variables. For instance, if the combination is {EbN0_4:[16, 19); C/I_4:[20, 30); Phi_2:[0, 0.005)}, I generate N uniformly distributed random samples within the intervals [16, 19), [20, 30), and [0, 0.005). Then, I run the simulation N times: one time per each combination of samples from those intervals. The bigger the number N, the more precise the results are. For every iteration of the simulation, I generate, modulate, and send K random symbols. At the receiver I demodulate the symbols, compare them with the transmitted symbols and accumulate the number of errors. At the end of the K iteration, I divide the number of errors by the total amount of transmitted bits to obtain the bit error rate "BER".

Figure 4.1.1 summarizes the process of simulating the transmission and demodulation of each of the K symbols mentioned in the previous paragraph, the purpose of which is to estimate the bit error rate, BER, for different levels of noise, interference, and Doppler phase shift. In ① the transmitter block selects randomly a symbol, which maps to one or several bits according to the modulation scheme. The phase ϕ of the carrier is modified to represent the selected symbol; the tables in figure 4.1.1 show the phases and bits per each symbol for modulations DBPSK, DQPSK, and D8QPSK. In ② the phase of the symbol is modified to simulate the Doppler effect. In ③ noise and interference are added to the symbol. In ④ the symbol is demodulated as explained in figures 4.1.2 and 4.1.3. In ⑤ the demodulated bits are compared with the transmitted bits, and in ⑥ the errors are accumulated to update the BER. The next paragraphs

explain in more details the demodulation process.



DBPSK

Sym	ϕ	Bits
S_1	0°	1
S_2	180°	0

DQPSK

Sym	ϕ	Bits
S_1	0°	00
S_2	90°	01
S_3	180°	11
S_4	270°	10

D8PSK

Sym	ϕ	Bits	Sym	ϕ	Bits
S_1	0°	000	S_5	180°	110
S_2	45°	001	S_6	225°	111
S_3	90°	101	S_7	270°	011
S_4	135°	100	S_8	315°	010

Figure 4.1.1: Sending symbols to estimate the bit error rate (BER) under different conditions of EbN0, noise, and Doppler phase shift for modulations DBPSK, DQPSK, and D8PSK.

Generally, simulations of digital modulation schemes consider the impact of additive white Gaussian noise (AWGN) at the receiver. Figure 4.1.2 shows the structure of a receiver operating with M-ary differential phase shift keying (DPSK) modulation; $M = 2^b$, where b is the number of bits per symbol [78]. The received signal $r(t)$ is the summation of the incoming signal $s_i(t)$ and the Gaussian noise $n(t)$. In order to consider the effect of co-channel interference, represented by C/I, and Doppler phase, represented by Dop_Phi, I have modified the system shown in figure 4.1.1. In figure 4.1.2 the phase of the symbols $\sqrt{E_s}s_{ui}(t)$ is modified by the Doppler phase shift ϕ_D, then the co-channel interference $ci(t)$, and the noise $n(t)$ are added at the receiver.

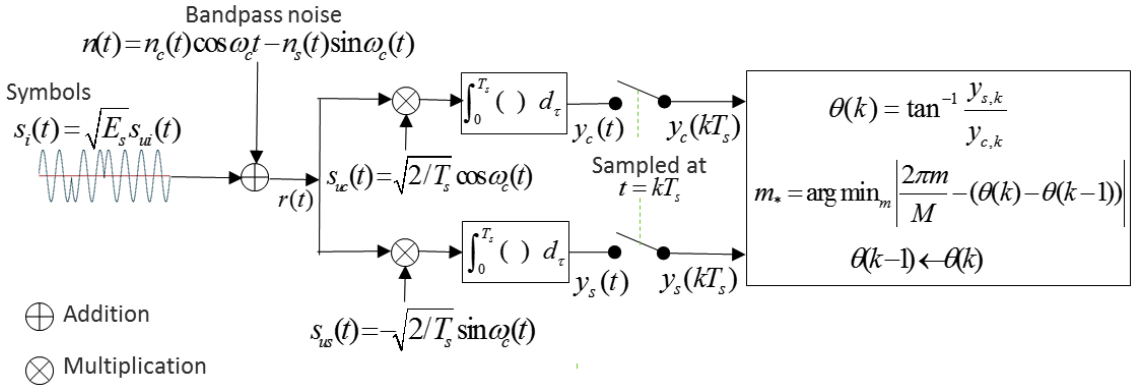


Figure 4.1.2: M-ary DPSK receiver with only noise [78]

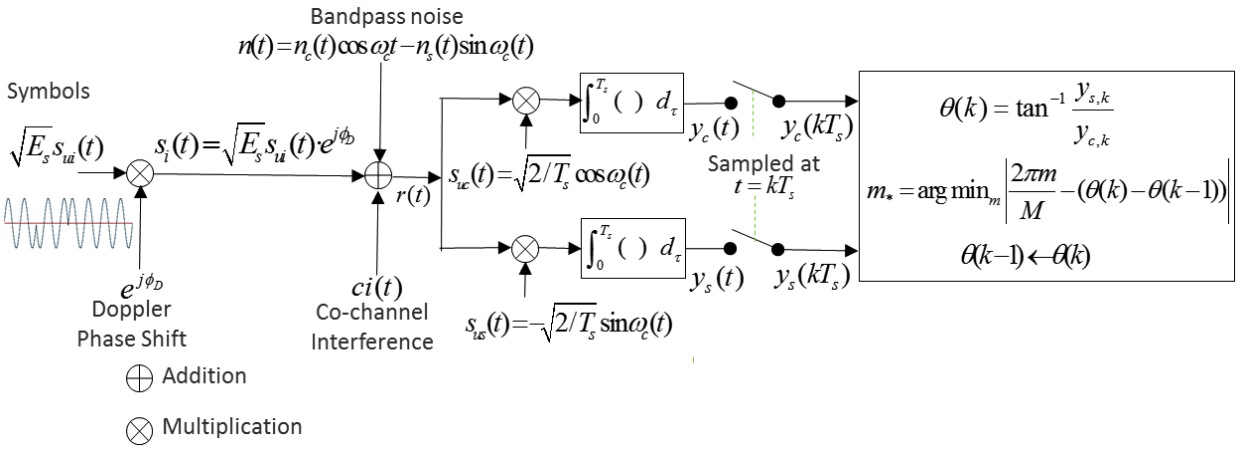


Figure 4.1.3: M-ary DPSK receiver with noise, co-channel inference and Doppler phase shift

To simulate the effect of the Doppler phase shift, I first represent the symbols as exponential numbers with phases corresponding to the M-ary constellation being simulated, then I multiply those numbers by an exponential number with a phase equal to the Doppler phase shift, so that I alter the phase of the received symbol accordingly. Below are the python code lines that perform this process:

```
phases= (2*pi/M)*np.arange(M) #Generate phases according to constellation
      exp_symbols= np.exp(phases*1j) #Represent symbols as exponential numbers

tx_signal=exp_symbols[is1] #random symbol as exponential number

channel=np.exp(FD*1j) #Doppler effect represented exponential number

rx_symbol=tx_signal*channel  #Alter the tx_signal according to Doppler phase shift

rx_phase=cmath.phase(rx_symbol)  #Phase of the symbol at the receiver
```

With the modified phase, I generate a sinusoidal signal using this python code line:

```
sws=np.concatenate((sws[Ns:],sqrt(2*Es/Ts)*np.cos(wc*t+rx_phase)),axis=1)) #Represent the symbol
                                                                            #as sinusoidal signal
```

This sinusoidal signal represents the received signal $s_i(t)$, to which I add Gaussian noise and co-channel interference as explained in the next paragraphs.

To simulate the effect of Gaussian noise on the communication system, I generate Gaussian random numbers with standard deviation $\sigma = 1$ and scale them by a factor according to the $\frac{E_b}{N_0}$ level being simulated. The portion of the python code that performs that operation is:

```
      SN_EB=round(10*np.log10(b),1)   # This term is to convert EbN0 to SNR in dB
                                      # according to b, bits per symbol.
SNRdB=EbN0 +SN_EB; # convert to SNR in dB
SNR=10**(SNRdB/10) # convert SBRdB to linear
N0=b*(Es/b)/SNR; sigma2=N0/2; sgmsT=sqrt(sigma2/T)  #sgmsT is to scale the
                                                    #noise according to Eb/N0

    #The next line represents the noise as a sinusoidal signal
     bp_noise=  np.cos(wct)*np.random.normal(0,1,Ns)+ np.sin(wct)*np.random.normal(0,1,Ns)
     bp_noise = sgmsT* bp_noise  #Scaling the noise according to EbN0
```

To simulate the effect of co_channel interference I add a sinusoidal signal scaled according to the signal to co-channel interference ratio. This is the python code that performs that operation

```
#The next line converts C/I to linear and multiplies the sinusoidal representing the
    #co-channel interference, so that it matches C/I
Co_Channel= CI_lin(C_I,Es,Ts)* sqrt(2*Es/Ts)*np.cos(wc*t+np.random.uniform(0,2*pi))
```

After I have the signal affected by Doppler effect, the noise, and the co-channel interference, I add them to obtain the received signal $r(t)$. Then I correlate $r(t)$ with the in-phase and in-quadrature components stored in the receiver to obtain the phase of the incoming signal, as shown by the formulas in figures 4.1.2 and 4.1.3.

For convenience, I grouped these processes in a function called BER (EbB0, C_I, Doppler, b, MaxIter), which takes as arguments: EbN0; C_I; Doppler; b, the number of bits per symbol; and MaxIter, the number of iterations. The arguments EbN0, C_I, and Doppler are random numbers within the intervals corresponding to the state of these variables. MaxIter represents the number of symbols transmitted and used to calculate the bit error rate every time I execute the function BER. The whole python code can be found in the appendix portion of this dissertation. I executed the function BER 200 times for every combination of the states of the variables EbN0, C/I and Dop_Phi. Since these states represent intervals wherein each variable falls, I generated random samples from these intervals each time I ran the function. For instance, if I were working with the combination: {EbN0_5, C/I_3, Phi_1}, where the states represent the intervals [16, 19), [20, 30), and [0, 0.05) respectively, I would take a random sample within each interval when I ran the function, ① in figure 4.1.3. After generating these raw data, I discretize them to represent states of the variables, since the algorithm for learning the probability distributions works with the states of the variables but not with actual numbers, ② in figure 4.1.3. Finally, in ③ I pass the discretized data, obtained in ②, through a portion of code that learns the conditional probability distribution (CPD) from the discretized data and the local Bayesian

network for the variable BER. As mentioned in chapter 3, I use local Bayesian networks as a tool to elicit conditional probability distributions, CPDs.

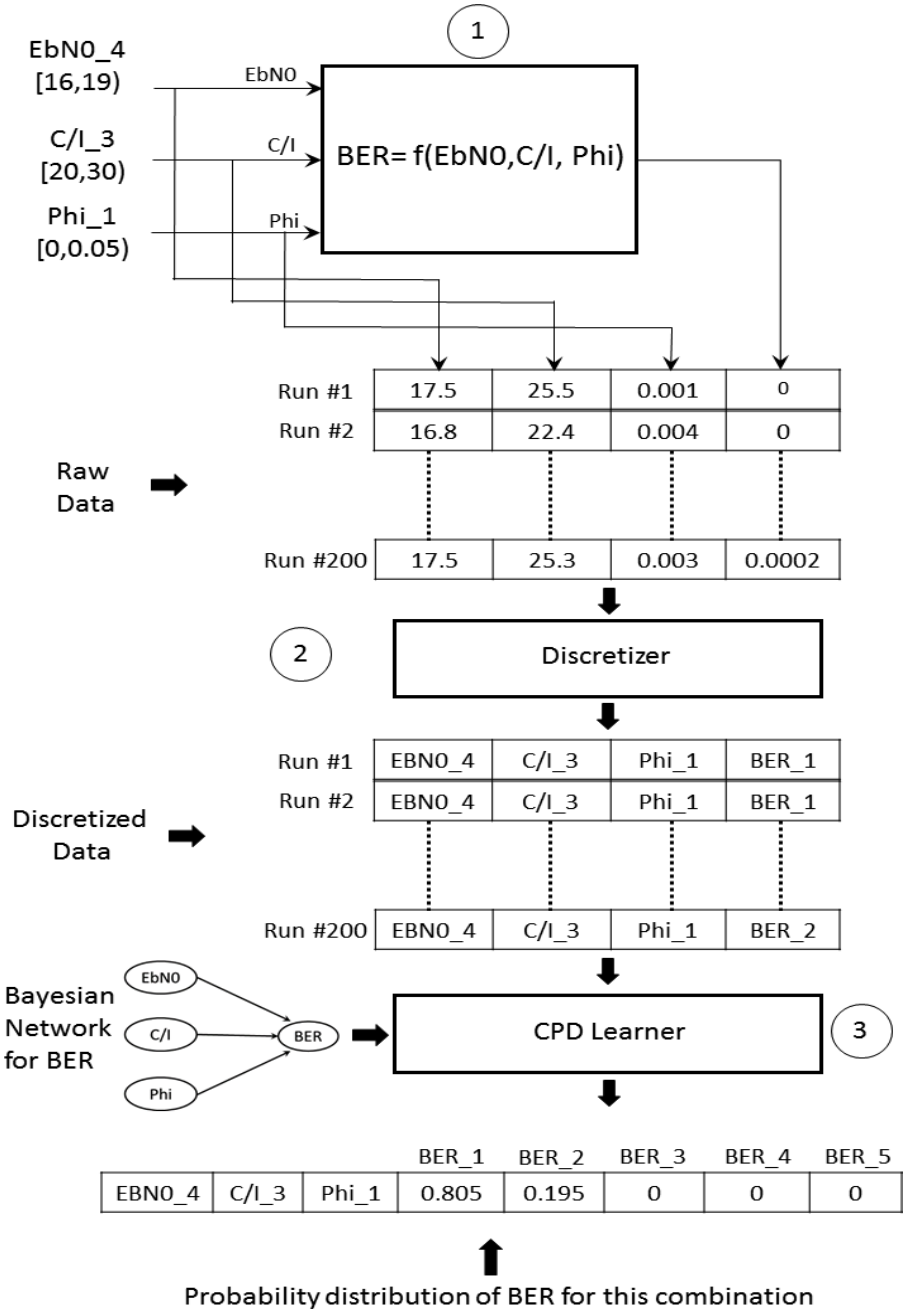


Figure 4.1.4: Process for learning the conditional probability distribution (CPD) for BER. In this example, the modulation system is DQPSK.

4.1.2 Results and Discussion

Tables 4.1.1 and 4.1.2 show the conditional probability distributions (CPD) for BER obtained after performing the simulation for different modulation schemes: differential binary phase shift keying (DBPSK), differential quaternary phase shift keying (DQPSK), and differential 8 phase shift keying (D8PSK). A CPD describes how the probability distribution of a random variable changes according to its influencing variables. When a random variable depends on discrete variables, its CPD must map each combination of the states of these variables, its parents, to its probability distribution. The CPD of the bit error rate, BER has 108 combinations due to the different states of the parents: 6 states for EbN0 times 6 states for C/I times 3 states for Dop_Phi. The probability distribution of BER falls over 5 states: BER_1, BER_2, BER_3, BER_4, and BER_5. BER_1 is the preferable state for the communication system, since it corresponds to the lowest error rate. Therefore, the parameters of the system must be manipulated so that the probability of the BER being low (states BER_1, or BER_2) increases and the probability of being high (states BER_3 to BER_5) decreases.

Since these tables include a high number of combinations, I will analyze only some of the more interesting combinations; the whole tables can be found in the appendix 1. Let us start with the combinations where the probability of BER being high is the greatest. This situation happens when either EbN0 or C/I is low and gets more critical as I increase the number of bits per symbol. When EbN0=EbN0_1, no matter what the states of C/I and Dop_Phi are, BER= BER_5 with probability 1. This happens for all the modulation schemes used in this dissertation: DBPSK, DQPSK, and D8PSK. One can see that in table 4.1.1 that the probability distribution of BER is [0, 0, 0, 0, 1], which corresponds to the probabilities of this variable being in the states

BER_1, BER_2, BER_3, BER_4, and BER_5 respectively, in other words: [p(BER=BER_1), p(BER=BER_2), p(BER=BER_3), p(BER=BER_4), p(BER=BER_5)].

Table 4.1.1: Probability distribution of BER for DBPSK, DQPSK, and D8PSK when EbN0 is EbN0_1.

States of the Parents			Probability distribution of BER				
EbN0_1	C/I_1 to C/I_6	Phi_1 to Phi_3	0	0	0	0	1

The probability distribution of BER is also [0, 0, 0, 0, 1] when C/I is at its lowest values, state C/I_1, regardless of the states of the other variables and the modulation scheme.

Table 4.1.2: Probability distribution of BER for DBPSK, DQPSK, and D8PSK when C/I is C/I_1.

States of the Parents			Probability distribution of BER				
EbN0_1to EbN0_6	C/I_1	Phi_1 to Phi_3	0	0	0	0	1

When EbN0 increases, the probability distribution of BER is no longer [0, 0, 0, 0, 1]. One can see in Tables 4.1.3 through 4.1.5 that when EbN0=EbN0_2, the p(BER_1) to p(BER_4) increase, whereas the p(BER_5) decreases. This is more noticeable at high values of C/I. In table 4.1.3, let us compare the row identified with I with the one identified with II. I can observe that in I, C/I is C/I_2, the probability distribution of BER is [0, 0, 0.1, 0.31, 0.59], whereas in II, C/I is C/I_6 the probability distribution is [0.01, 0.07, 0.235, 0.455, 0.23]. It makes sense that if C/I is higher the error rate diminishes. I can also see that p(BER=BER_5) drops from 0.59 to 0.23 when C/I goes from C/I_2 to C/I_6. This pattern holds in Tables 4.1.4 and 4.1.5, although p(BER=BER_5) drops by a smaller amount. These tables correspond to the DQPSK and D8PSK

modulations respectively, which are more susceptible to noise and interference, or in other words exhibit high error probability for a given EbN0 and C/I.

Table 4.1.3: Probability distribution of BER for DBPSK EbN0 is EbN0_2.

States of the parents			Probability distribution of BER					
EbN0_2	C/I_2	Phi_1	0	0	0.1	0.31	0.59	I
EbN0_2	C/I_2	Phi_2	0	0	0.075	0.39	0.535	
EbN0_2	C/I_2	Phi_3	0	0	0.1	0.27	0.63	
EbN0_2	C/I_3	Phi_1	0	0.09	0.28	0.405	0.225	
EbN0_2	C/I_3	Phi_2	0.005	0.06	0.295	0.36	0.28	
EbN0_2	C/I_3	Phi_3	0	0.065	0.255	0.415	0.265	
EbN0_2	C/I_4	Phi_1	0	0.08	0.22	0.44	0.26	
EbN0_2	C/I_4	Phi_2	0.005	0.095	0.175	0.47	0.255	
EbN0_2	C/I_4	Phi_3	0	0.12	0.19	0.405	0.285	
EbN0_2	C/I_5	Phi_1	0	0.11	0.25	0.385	0.255	
EbN0_2	C/I_5	Phi_2	0	0.09	0.17	0.46	0.28	
EbN0_2	C/I_5	Phi_3	0	0.08	0.22	0.41	0.29	
EbN0_2	C/I_6	Phi_1	0.01	0.07	0.235	0.455	0.23	II
EbN0_2	C/I_6	Phi_2	0	0.06	0.245	0.435	0.26	
EbN0_2	C/I_6	Phi_3	0	0.105	0.19	0.445	0.26	

Table 4.1.4: Probability distribution of BER for DQPSK EbN0 is EbN0_2.

States of the parents			Probability distribution of BER					
EbN0_2	C/I_2	Phi_1	0	0	0	0.2	0.8	I
EbN0_2	C/I_2	Phi_2	0	0	0	0.165	0.835	
EbN0_2	C/I_2	Phi_3	0	0	0	0.155	0.845	
EbN0_2	C/I_3	Phi_1	0	0	0	0.47	0.53	
EbN0_2	C/I_3	Phi_2	0	0	0	0.47	0.53	
EbN0_2	C/I_3	Phi_3	0	0	0	0.47	0.53	
EbN0_2	C/I_4	Phi_1	0	0	0	0.465	0.535	
EbN0_2	C/I_4	Phi_2	0	0	0	0.56	0.44	
EbN0_2	C/I_4	Phi_3	0	0	0	0.505	0.495	
EbN0_2	C/I_5	Phi_1	0	0	0.005	0.405	0.59	
EbN0_2	C/I_5	Phi_2	0	0	0	0.46	0.54	
EbN0_2	C/I_5	Phi_3	0	0	0	0.455	0.545	
EbN0_2	C/I_6	Phi_1	0	0	0.005	0.465	0.53	II
EbN0_2	C/I_6	Phi_2	0	0	0	0.52	0.48	

Table 4.1.5: Probability distribution of BER for D8PSK EbN0 is EbN0_2.

States of the parents			Probability distribution of BER					
EbN0_2	C/I_2	Phi_1	0	0	0	0	1	I
EbN0_2	C/I_2	Phi_2	0	0	0	0	1	
EbN0_2	C/I_2	Phi_3	0	0	0	0	1	
EbN0_2	C/I_3	Phi_1	0	0	0	0.085	0.915	
EbN0_2	C/I_3	Phi_2	0	0	0	0.08	0.92	
EbN0_2	C/I_3	Phi_3	0	0	0	0.065	0.935	
EbN0_2	C/I_4	Phi_1	0	0	0	0.085	0.915	
EbN0_2	C/I_4	Phi_2	0	0	0	0.135	0.865	
EbN0_2	C/I_4	Phi_3	0	0	0	0.115	0.885	
EbN0_2	C/I_5	Phi_1	0	0	0	0.12	0.88	
EbN0_2	C/I_5	Phi_2	0	0	0	0.135	0.865	
EbN0_2	C/I_5	Phi_3	0	0	0	0.105	0.895	
EbN0_2	C/I_6	Phi_1	0	0	0	0.105	0.895	II
EbN0_2	C/I_6	Phi_2	0	0	0	0.145	0.855	
EbN0_2	C/I_6	Phi_3	0	0	0	0.095	0.905	
EbN0_2	C/I_6	Phi_3	0	0	0	0.465	0.535	

Tables 4.1.6 through 4.1.8 show that when EbN0 reaches its maximum, EbN0=EbN0_6, p(BER=BER_1) gets its highest values and even reaches 1. It is important to mention that if the variable C/I is at a low value, it also affects the probability distribution of BER. This influence becomes stronger as I move from DBPSK to D8PSK modulation: the bits per symbol augment. Table 4.1.6 shows that with DBPSK modulation p(BER=BER_1)= 1 when C/I $\geq$ C/I_3. Remember that I want p(BER=BER_1) to be as high as possible, so that I can have fewer errors and retransmissions in the network. What table 4.1.6 tells is that with DBPSK modulation this is more probable to obtain than with DQPSK and D8PSK modulation. Indeed, this is something that the literature suggests. In this dissertation I are expressing this knowledge by means of probability tables such as tables 4.1.6 through 4.1.8. Bayesian networks and influence diagrams require this type of tables to perform probabilistic reasoning.

Table 4.1.6: Probability distribution of BER for DBPSK when EbN0 is EbN0_6.

States of the parents			Probability distribution of BER					
EbN0_6	C/I_2	Phi_1	0.65	0	0.025	0.07	0.255	I
EbN0_6	C/I_2	Phi_2	0.63	0.03	0.015	0.08	0.245	
EbN0_6	C/I_2	Phi_3	0.605	0.04	0.025	0.07	0.26	
EbN0_6	C/I_3	Phi_1	1	0	0	0	0	
EbN0_6	C/I_3	Phi_2	1	0	0	0	0	
EbN0_6	C/I_3	Phi_3	1	0	0	0	0	
EbN0_6	C/I_4	Phi_1	1	0	0	0	0	
EbN0_6	C/I_4	Phi_2	1	0	0	0	0	
EbN0_6	C/I_4	Phi_3	1	0	0	0	0	
EbN0_6	C/I_5	Phi_1	1	0	0	0	0	
EbN0_6	C/I_5	Phi_2	1	0	0	0	0	
EbN0_6	C/I_5	Phi_3	1	0	0	0	0	
EbN0_6	C/I_6	Phi_1	1	0	0	0	0	II
EbN0_6	C/I_6	Phi_2	1	0	0	0	0	
EbN0_6	C/I_6	Phi_3	1	0	0	0	0	

Table 4.1.7: Probability distribution of BER for DQPSK when EbN0 is EbN0_6.

States of the parents			Probability distribution of BER					
EbN0_6	C/I_2	Phi_1	0.375	0.03	0.015	0.135	0.445	I
EbN0_6	C/I_2	Phi_2	0.355	0.06	0.07	0.12	0.395	
EbN0_6	C/I_2	Phi_3	0.25	0.07	0.1	0.165	0.415	
EbN0_6	C/I_3	Phi_1	1	0	0	0	0	
EbN0_6	C/I_3	Phi_2	1	0	0	0	0	
EbN0_6	C/I_3	Phi_3	0.995	0.005	0	0	0	
EbN0_6	C/I_4	Phi_1	1	0	0	0	0	
EbN0_6	C/I_4	Phi_2	1	0	0	0	0	
EbN0_6	C/I_4	Phi_3	1	0	0	0	0	
EbN0_6	C/I_5	Phi_1	1	0	0	0	0	
EbN0_6	C/I_5	Phi_2	1	0	0	0	0	
EbN0_6	C/I_5	Phi_3	0.995	0.005	0	0	0	
EbN0_6	C/I_6	Phi_1	1	0	0	0	0	II
EbN0_6	C/I_6	Phi_2	1	0	0	0	0	
EbN0_6	C/I_6	Phi_3	1	0	0	0	0	

Table 4.1.8: Probability distribution of BER for D8PSK when EbN0 is EbN0_6.

States of the parents			Probability distribution of BER					
EbN0_6	C/I_2	Phi_1	0.025	0.015	0.035	0.25	0.675	I
EbN0_6	C/I_2	Phi_2	0	0.01	0.07	0.235	0.685	
EbN0_6	C/I_2	Phi_3	0	0	0.025	0.355	0.62	
EbN0_6	C/I_3	Phi_1	0.96	0.035	0.005	0	0	
EbN0_6	C/I_3	Phi_2	0.725	0.225	0.05	0	0	
EbN0_6	C/I_3	Phi_3	0.36	0.34	0.295	0.005	0	
EbN0_6	C/I_4	Phi_1	0.96	0.04	0	0	0	
EbN0_6	C/I_4	Phi_2	0.945	0.055	0	0	0	
EbN0_6	C/I_4	Phi_3	0.935	0.06	0.005	0	0	
EbN0_6	C/I_5	Phi_1	0.985	0.015	0	0	0	
EbN0_6	C/I_5	Phi_2	0.96	0.04	0	0	0	
EbN0_6	C/I_5	Phi_3	0.915	0.06	0.025	0	0	
EbN0_6	C/I_6	Phi_1	0.99	0.01	0	0	0	II
EbN0_6	C/I_6	Phi_2	0.945	0.05	0.005	0	0	
EbN0_6	C/I_6	Phi_3	0.955	0.045	0	0	0	

4.2 Simulation of the Bayesian Network

In chapter 3, I presented the Bayesian model to perform probabilistic inference on random variables that affect the performance of the wireless communication system. This section presents how the Bayesian network can answer some queries about specific variables based on evidence. I will analyze several cases that show how the wireless system Bayesian network (WBN) updates the conditional probability tables (CPD) of some random variables as it gets new evidence. The variables of interest in this case are: N0 (Noise Density), Co_Ch (Interference Power), Fade (Multipath Fading), and Dist (Distance).

4.2.1 Methodology

To see how the WBN responds to queries I first assume that I have no knowledge about the probability distribution of the variables of interest: N0, Co_Ch, Fade, and Dist. In the Bayesian

model described in chapter 3, these are random variables that affect the operation of the wireless communication system. Because of their random nature, the way I have to estimate how they can impact the wireless system is through their probability distributions. Since at the beginning I ignore the probability distribution of the aforementioned variables, I start with uniform probability distributions. Then, I can update the probability distributions of these variables by observing the state of one of the variables they influence. In this case, I observe the state of the variable bit error rate, BER, ① in figure 4.2.1. I can call this process getting evidence. This evidence is used to reduce the factor that represents the conditional probability distribution (CPD) of BER, ② in figure 4.2.1; I explained the factor reduction process in chapter 2. After this CPD is reduced, the variable elimination, VE, algorithm is used to get rid of all the other variables except for the ones, whose CPDs I want to infer, ③ in figure 4.2.1. I already explained the VE algorithm in the second part of chapter 2. From that explanation, remember that eliminating a variable means to summing out all the occurrences of that variable in the factors that represent the Bayesian network.

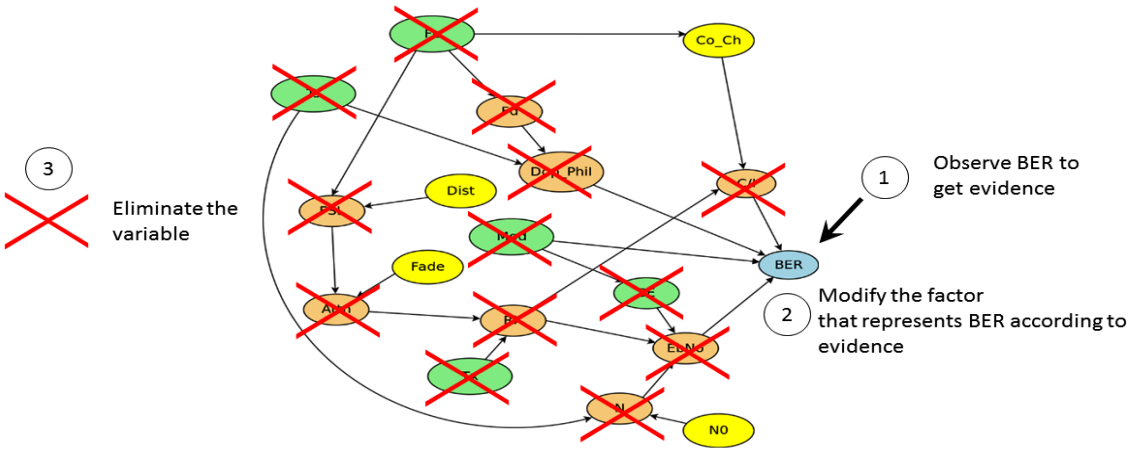


Figure 4.2.1: Variable elimination (VE) based on evidence. The evidence is the state of BER that the receiver observes.

Once I have the CPDs, I use them the next time I get new evidence; therefore in each

iteration I perform probabilistic inference with CPDs updated in the previous step. Figure 4.2.2 illustrates the wireless system Bayesian network (WBN) as an entity that receives as inputs: *Setup_n*, the values of the parameters at the iteration *n*; *Evidence_n*, the observed value of BER at the iteration *n*; *Query_n*, set of variables for which I want to infer the CPDs; *CPD_n*, CPDs of the query variables at the iteration *n*. With those inputs the WBN yields *CPD_n+1*, the updated CPDs of the query variables to be used in the next iteration, *n+1*. If in the iteration 0 I have no previous knowledge about the query variables, CPD_0 is uniform for all the variables. In case I have some preliminary knowledge about the query variables that comes from former experiments and measurements, the CPD_0 are no longer uniform but contain probabilistic information won through experience. For instance, the radio device could measure the interference power (*Co_Ch)* regularly, and from these measurements estimate the CPD of this variable.

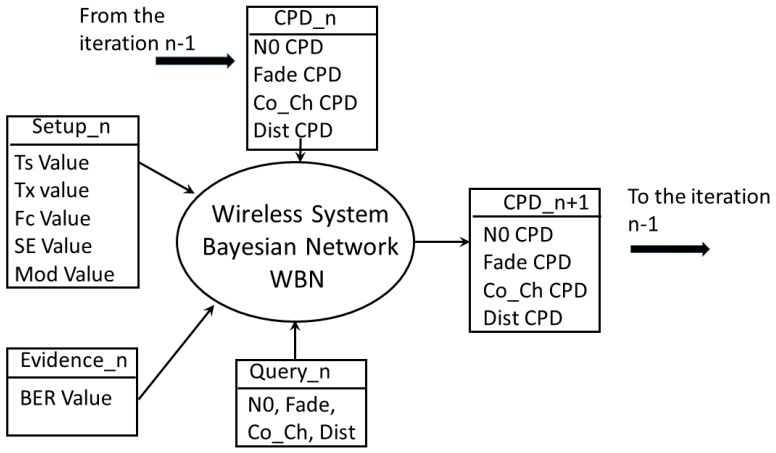


Figure 4.2.2: Updating the conditional probability tables based on the evidence obtained in each iteration.

To show how the Bayesian network learns from new evidence I perform simulations, wherein I observe BER several times, and show how the CPDs evolve with each observation. Each time I run these simulations, I assume initial uniform probability distributions for the

variables N0 (Noise Density), Co_Ch (Interference Power), Fade (Multipath Fading), and Dist (Distance). I follow the procedure described in figure 4.2.2: each time I get new evidence, I make a query, and use the CPDs obtained in the query in the next iteration. In the set of experiments that assume uniform probabilities at the beginning, the initial CPDs are: N0:{0.25,0.25,0.25,0.25}; Co_Ch:{0.2,0.2,0.2,0.2,0.2}; Dist:{0.25,0.25,0.25,0.25}; Fade:{0.333,0.333,0.333}.

I analyze different cases:

1. The BER is mostly low, the transmission power Tx is low, Tx_1, the spectral efficiency is high, SE_3, and modulation D8PSK.
2. The BER is mostly low, the transmission power Tx is high, Tx_4, the spectral efficiency is low, SE_1, and modulation DBPSK.
3. The BER is mostly high, the transmission power Tx is low, Tx_1, the spectral efficiency is high, SE_3 and modulation D8PSK.
4. The BER is mostly high, the transmission power Tx is high, Tx_4, the spectral efficiency is low, SE_1, and modulation DBPSK.

In all the four cases, I kept the carrier frequency Fc, and the time of symbol Ts at Fc_1, and Ts_1 respectively.

4.2.2 Results and Discussion

As I present the probability distributions in this section, let us remember that in chapter 3, I defined these probability distributions as follows:

N0: [p(N0=N0_1), p(N0=N0_2), p(N0=N0_3), p(N0=N0_4)], where N0_1 < N0_2 < N0_3 < N0_4.

Co_Ch: [p(Co_Ch=CC_1), p(Co_Ch=CC_2), p(Co_Ch=CC_3), p(Co_Ch=CC_4), p(Co_Ch=CC_5)], where CC_1 < CC_2 < CC_3 < CC_4 < CC_5.

Dist: [p(Dist=D_1), p(Dist=D_2), p(Dist=D_3), p(Dist=D_4)], where D_1 < D_2 < D_3 < D_4.

Fade: [p(Fade=Sh_1), p(Fade=Sh_2), p(Fade=Sh_3)], where Sh_1 < Sh_2 < Sh_3.

Let us remember also that p(x=x_1) means the probability that the variable x is in the state x_1. For sake of brevity, in some cases I use p(x_1) to represent p(x=x_1). For instance, either p(Co_Ch=CC_1) or p(CC_1) represents the probability that the co-channel interference power Co_Ch is at its lowest values, state CC_1.

Case 1: BER is mostly low, the transmission power Tx is low, Tx_1, the spectral efficiency is high, SE_3 and modulation D8PSK.

The setup is: {'Fc': 'Fc_1', 'Ts': 'Ts_1', 'Tx': 'Tx_1', 'SE': 'SE_3'}

The initial probability distributions, before observing BER, are:

N0: [0.25, 0.25, 0.25, 0.25]

Co_Ch: [0.2, 0.2, 0.2, 0.2, 0.2]

Dist: [0.25, 0.25, 0.25, 0.25]

Fade: [0.333, 0.333, 0.333]

After observing BER, the probability distributions become:

Observation 1: BER= BER_1

N0: [0.267, 0.267, 0.267, 0.199]

Co_Ch: [0.306, 0.296, 0.235, 0.137, 0.025]

Dist: [0.529, 0.220, 0.125, 0.125]

Fade: [0.499, 0.389, 0.112]

Before observation 1, it is equally probable for the variables to be at low, medium, or high levels. Notice that after observing that BER is low, the network yields CPDs with higher probabilities for states that cause BER=BER_1. This is because lower values of noise power density N0, distance Dist, and multipath fading Fade increase the probability of EbN0 being at high values, which reduces the probability of error or BER; similarly, lower values of co-channel interference power Co_Ch, increase the probability of C/I being at high values, which along with a high EbN0 also contribute to reduce the BER.

Observation 2: BER =BER_2

N0: [0.273, 0.273, 0.273, 0.180]

Co_Ch: [0.398, 0.316, 0.187, 0.093, 0.007]

Dist: [0.565, 0.208, 0.113, 0.113]

Fade: [0.532, 0.407, 0.061]

After observation 2, the trend continues. The probabilities of lower values of N0, Co_Ch, Dist, and Fade decrease, whereas those of higher values decrease.

Observation 3: BER = BER_1

N0: [0.282, 0.282, 0.282, 0.153]

Co_Ch: [0.431, 0.335, 0.173, 0.059, 0.000]

Dist: [0.770, 0.134, 0.048, 0.048]

Fade: [0.591, 0.389, 0.02]

Observation 4: BER = BER_1

N0: [0.289, 0.289, 0.289, 0.134]

Co_Ch: [0.452, 0.347, 0.160, 0.040, 0.000]

Dist: [0.888, 0.075, 0.018, 0.018]

Fade: [0.628, 0.365, 0.007]

Observation 5: BER=BER_1

N0: [0.294, 0.294, 0.294, 0.118]

Co_Ch': [0.468, 0.354, 0.149, 0.027, 0],

Dist: [0.948, 0.039, 0.007, 0.007]

Fade: [0.656, 0.341, 0.002]

Notice in observations 3 through 5 that the Bayesian network reinforces its beliefs that N0, Co_Ch, Dist, and Fade are at their lowest values. For instance, the initial probabilities for CC_1 and CC_5 are both 0.2; after observation 1, p(Co_Ch = CC_1) rises to 0.306 and p(Co_Ch = CC_5) drops to 0.025. This trend continues through all the observations; by observation 5, p(Co_Ch = CC_1) has risen to 0.468, whereas p(Co_Ch = CC_5) has dropped to zero.

Case 2: BER is mostly low, the transmission power Tx is high, Tx_4, the spectral efficiency is low, SE_1 and modulation DBPSK.

The setup is: {'Fc': 'Fc_1', 'Ts': 'Ts_1', 'Tx': 'Tx_4', 'SE': 'SE_1'}

The initial probability distributions, before I observe BER, are:

N0: [0.25, 0.25, 0.25, 0.25]

Co_Ch: [0.2, 0.2, 0.2, 0.2, 0.2]

Dist: [0.25, 0.25, 0.25, 0.25]

Fade: [0.333, 0.333, 0.333]

After observing BER, the probability distributions become:

Observation 1: BER= BER_1

N0: [0.255, 0.255, 0.255, 0.233]

Co_Ch: [0.26, 0.254, 0.2344, 0.186, 0.066]

Dist: [0.292, 0.245, 0.231, 0.231]

Fade: [0.392, 0.384, 0.223]

As in the previous case, BER is mostly low. However, the transmission power Tx is set at a high value and the modulation scheme is DBPSK, which is more tolerant to noise and interference than the modulation used in case 1, D8PSK. With such a configuration, and if the environment remains with the same conditions as in case 1, p(BER=BER_1) should go higher, since the signal power has increased and a modulation scheme is less prone to error. Like in case 1, the BN starts with uniform probability distributions. After observation 1, p(Co_Ch=CC_1) goes up to 0.26 and p(Co_Ch=CC_5) falls to 0.066. The same behavior took place in case 1; however, in this case the change in the probability distribution of Co_Ch is smaller. Since Tx is high, Tx=Tx_4, the fact that BER=BER_1 could be caused more by the high level of Tx than by the low value of Co_Ch. Therefore, the belief the Bayesian network has about Co_Ch being low is less strong than in case 1.

Observation 2: BER = BER_2

N0: [0.203, 0.203, 0.203, 0.391]

Co_Ch: [0.382, 0.382, 0.19, 0.045, 0.0]

Dist: [0.117, 0.271, 0.306, 0.306]

Fade: [0.112, 0.156, 0.732]

The setting of the transmitter is such that p(BER=BER_1) should be high. The transmission power is high and the modulation scheme is less prone to noise. Therefore, if the evidence tells BER=BER_2 or higher, the Bayesian network (BN) modifies its beliefs about the environment. The BN starts to believe the environment has such conditions that causes more transmission

errors; this change in beliefs is represented through a change in the probability distributions of the variables N0, Co_Ch, Fade, and Dist. Let us remember that a Bayesian network (BN) is also called Belief network because it represents its beliefs about the state of variables as probabilities, which it adjusts as it gets new evidence; the stronger the belief, the higher the probability, and the weaker the belief, the lower the probability.

Let us notice the changes in the probability distributions of N0, Fade, and Dist after observation 2; the probabilities of higher values have increased, this is due to the fact that higher values of noise density (N0) and multipath fading (Fade), and larger distances (Dist) increase the probability of error (BER). Therefore, the probability distributions, beliefs of the Bayesian network, evolved accordingly: p(N0_4)=0.391, p(D_4)= 0.306, and p(Sh_3) = 0.732. When in case 1 the evidence told BER=BER_2, those probability distributions did not suffer such a dramatic change; the probabilities for N0, Fade and Dist being at their lowest values remained high. The explanation: in case 1, the transmission power was low, Tx=Tx_1, while the spectral efficiency high, modulation D8PSK with SE=SE_3. This setup makes the communication system more prone to error; therefore, the fact that BER increased to BER_2 is due more to the configuration than to the environment.

Observation 3: BER = BER_1

N0: [0.221, 0.221, 0.221, 0.337]

Co_Ch: [0.415, 0.390, 0.166, 0.028, 0.0]

Dist: [0.156, 0.277, 0.283, 0.283]

Fade: [0.162, 0.223, 0.615]

Observation 4:BER= BER_1

N0: [0.235, 0.235, 0.235, 0.295]

Co_Ch: [0.440, 0.394, 0.148, 0.018, 0.0]

Dist: [0.193, 0.279, 0.264, 0.264]

Fade: [0.214, 0.291, 0.495]

Observation 5: BER =BER_1

N0: [0.246, 0.246, 0.246, 0.263]

Co_Ch: [0.459, 0.395, 0.133, 0.012, 0.0]

Dist: [0.226, 0.277, 0.248, 0.248]

Fade: [0.262, 0.353, 0.385]

In observations 3 to 5, the evidence telling repeatedly that BER=BER_1 has made p(N0_4), p(D_4), and p(Sh_3) start dropping and p(N0_1), p(D_1); and p(Sh_1) start increasing. Which means that shorter distances, lower noise, and lower multipath fading have started to be more probable.

Case 3: The BER is mostly high, the transmission power Tx is high, Tx_4, the spectral efficiency is low, SE_1 and modulation DBPSK.

The setup is: {'Fc': 'Fc_1', 'Ts': 'Ts_1', 'Tx': 'Tx_4', 'SE': 'SE_1'}

The initial probability distributions, before I observe BER, are:

N0: [0.25, 0.25, 0.25, 0.25]

Co_Ch: [0.2, 0.2, 0.2, 0.2, 0.2]

Dist: [0.25, 0.25, 0.25, 0.25]

Fade: [0.333, 0.333, 0.333]

When I start observing BER, the probability distribution become:

Observation 1: BER= BER_4

N0: [0.219, 0.219, 0.219, 0.344]

Co_Ch: [0.242, 0.246, 0.207, 0.165, 0.140]

Dist: [0.156, 0.260, 0.292, 0.292]

Fade: [0.179, 0.202, 0.618]

After observation 1, the probability distributions show that high values of noise density N0, multipath fading Fade, and distance Dist are more probable than lower values: p(N0_4) = 0.344, p(D_4) = 0.292, p(Sh_3) = 0.618, whereas p(N0_1) = 0.219, p(D_1) = 0.156, and p(Sh_1) = 0.179. An environment with such conditions increases the error probability BER. Since BER is high, BER=BER_4, the Bayesian network, BN, has modified its beliefs accordingly.

Observation 2: BER= BER_2

N0: [0.212, 0.212, 0.212, 0.364]

Co_Ch: [0.404, 0.412, 0.156, 0.028, 0.0]

Dist: [0.071, 0.257, 0.335, 0.335]

Fade: [0.036, 0.053, 0.910]

In this observation, BER=BER_2, which is lower than in the previous observation; however, high values of noise density N0, multipath fading Fade, and distance Dist are even more probable than before. If I consider that with this setup, Tx=Tx_4, and modulation DBPSK, I should get low BER. BER=BER_2 implies that the conditions of the environment are such that BER is higher than expected.

Observation 3: BER =BER_3

N0: [0.202, 0.202, 0.202, 0.393]

Co_Ch: [0.431, 0.467, 0.096, 0.005, 0.0]

Dist: [0.032, 0.24, 0.364, 0.364]

Fade: [0.005, 0.009, 0.986]

Observation 4: BER= BER_4

N0: [0.191, 0.191, 0.191, 0.425]

Co_Ch: [0.455, 0.484, 0.059, 0.001, 0.0]

Dist: [0.014, 0.218, 0.383, 0.383]

Fade: [0.000, 0.002, 0.998]

In observations 3 and 4 I have the same trend that I have in the two first observations.

Observation 5: BER = BER_1

N0: [0.224, 0.224, 0.224, 0.328]

Co_Ch: [0.488, 0.467, 0.045, 0.000, 0.0]

Dist: [0.022, 0.241, 0.368, 0.368]

Fade: [0.001, 0.003, 0.996]

When BER is low, BER=BER_1, I see a trend slightly different from the one of the first four observations: the probabilities of high values start decreasing, whereas the probabilities of low values start increasing. This indicates that the environment has become more favorable for obtaining lower error probabilities; therefore, the probability distributions change accordingly.

Case 4: The BER is mostly high, the transmission power Tx is low, Tx_1, the spectral efficiency is high, SE_3 and modulation D8PSK.

The setup is: {'Fc': 'Fc_1', 'Ts': 'Ts_1', 'Tx': 'Tx_1', 'SE': 'SE_3'}

The initial probability distributions, before observing BER, are:

N0: [0.25, 0.25, 0.25, 0.25]

Co_Ch: [0.2, 0.2, 0.2, 0.2, 0.2]

Dist: [0.25, 0.25, 0.25, 0.25]

Fade: [0.333, 0.333, 0.333]

After observing BER, the probability distribution become:

Observation 1: BER = BER_5

N0: [0.242, 0.242, 0.242, 0.275]

Co_Ch: [0.135, 0.153, 0.192, 0.236, 0.283]

Dist: [0.154, 0.260, 0.292, 0.292]

Fade: [0.266, 0.301, 0.433]

In this case, the configuration is less favorable for getting low error probabilities than the one used in case 3. Hence, it is natural seeing high values of BER, which in turn causes that the changes observed in the probability distributions are smaller and less extreme than the ones observed in case 1. In case 3, the probabilities tended to be either very low or very high. On the other hand, in this case, (case 4) the probability distributions look more evenly distributed. An almost evenly distributed probability distribution provides less information than a biased probability distribution: the more bias a probability distribution has, the more information it provides about the random variable characterized by that distribution; for instance, if the probability distribution for the variable Distance is Dist:[0.948, 0.039, 0.007, 0.007], since p (D_1) is so high, I am almost sure that this variable is in state D_1, which represents distances between 10 and 100 meters. On the other hand, if the probability distribution is Dist: [0.154, 0.260, 0.293, 0.293] I cannot be as certain about the state of the variable as I was with the previous probability distribution. Let us look at the numbers; none of them is much bigger than the others, which means I have less information about the variable Distance. The reason: with the configuration used in this case, low power and modulation prone to error (D8PSK), the error probability is high; hence, if the evidence tells the BER is high, there is no surprise because I was expecting that; therefore, I get less information out of this evidence. The less surprising the

evidence is, the less information it provides. Despite of that, the Bayesian network still learns as it gets new evidence; however, it needs to collect more pieces of evidence.

Observation 2: BER = BER_4

N0: [0.263, 0.263, 0.263, 0.211]

Co_Ch: [0.329, 0.269, 0.194, 0.144, 0.062]

Dist: [0.156, 0.260, 0.292, 0.292]

Fade: [0.353, 0.419, 0.228]

Observation 3: BER= BER_5

N0: [0.249, 0.249, 0.249, 0.252]

Co_Ch: [0.243, 0.234, 0.218, 0.203, 0.101]

Dist: [0.068, 0.256, 0.337, 0.337]

Fade: [0.279, 0.386, 0.334]

Observation 4: BER= BER_4

N0: [0.274, 0.274, 0.274, 0.178]

Co_Ch: [0.443, 0.303, 0.158, 0.084, 0.011]

Dist: [0.047, 0.247, 0.353, 0.353]

Fade: [0.338, 0.513, 0.148]

Observation 5: BER =BER_5

N0: [0.256, 0.256, 0.256, 0.231]

Co_Ch: [0.352, 0.291, 0.203, 0.134, 0.019]

Dist: [0.014, 0.222, 0.382, 0.382]

Fade: [0.275, 0.488, 0.237]

4.3 Simulation of the Influence Diagram

In the previous section, I presented simulations showing how the wireless system Bayesian network (WBN) uses evidence to update its beliefs about some of the random variables of the

network. In this section, I perform some simulations to demonstrate how the influence diagram (ID), explained in chapter 3, can assist the wireless communication system in selecting the configuration that suits it the best in the midst of the random conditions of the environment, characterized by the conditional probability distributions (CPD). Remember from chapter 3 that I can build an influence diagram by adding one or several utility function nodes to a Bayesian network. In figure 4.3.1, I have added a utility function to the BN I used in the previous section.

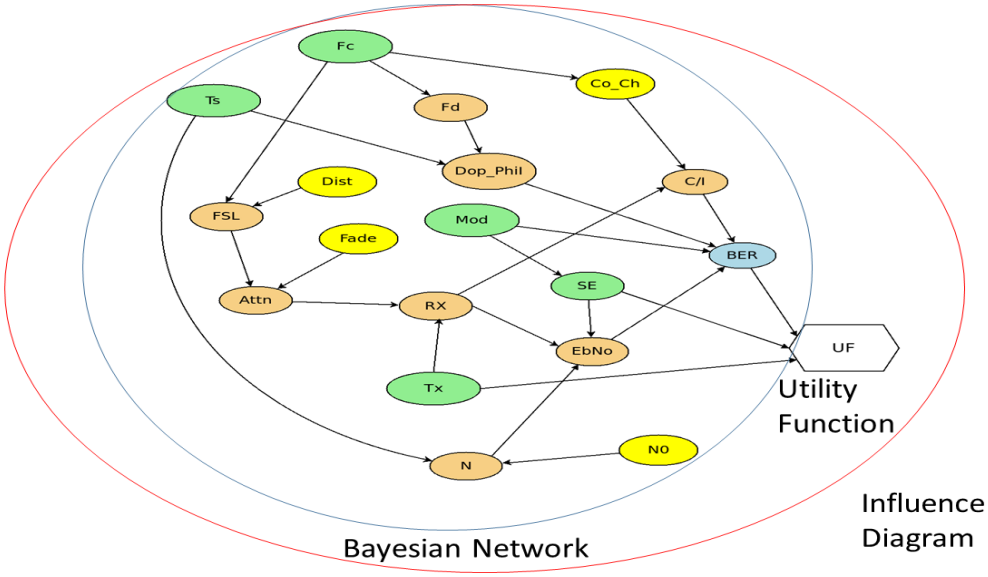


Figure 4.3.1. Influence Diagram for Decision Making in Wireless Communication System. *A Bayesian network plus one or several utility functions turns into an Influence Diagram.*

4.3.1 Methodology

To show the functionality of the influence diagram (ID), I first have to briefly explain how the wireless communication system Bayesian network (WBN) and the ID work together. The WBN updates the CPD as it gets new evidence, and passes them along to the ID, which calculates the utilities for different policies in order to select the one with the maximum utility. Figure 4.3.2 illustrates this process.

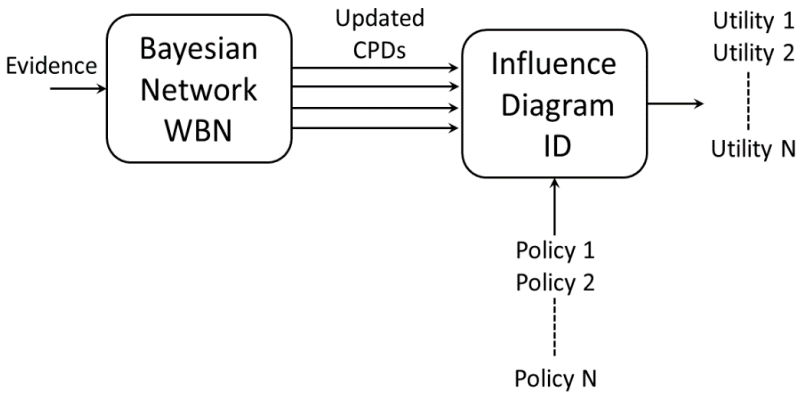


Figure 4.3.2. Bayesian Network interacting with the Influence Diagram. *The influence diagram calculates the utility for each policy based on the CPDs passed by the Bayesian network.*

For demonstration purposes, I suppose different scenarios wherein the wireless system operates and have the influence diagram calculate the utility for different policies applied in those scenarios. In this context, a policy is a set of decisions as of how to configure the wireless communication system. In other words, a policy is a specific configuration, which contains the values for the parameters: transmission power (Tx), carrier frequency (Fc), time of symbol (Ts), modulation scheme (Mod), and spectral efficiency (SE). I first load the conditional probability distributions (CPD) into the influence diagram (ID), then pass the policy of which I want to calculate the utility, and finally sort the policies descendently according to their utilities.

To represent different scenarios, I manipulate the probability distributions of the variables N0 (Noise Density), Co_Ch (Interference Power), Fade (Multipath Fading), and Dist (Distance) accordingly; I assign high probabilities to states that I would expect to see more in that scenario. For example, the probability distributions for a scenario wherein the distances tend to be short, the multipath fading low, and the noise density medium could look like: N0:[0.05, 0.4, 0.5, 0.05]; Fade:[0.8, 0.2, 0]; Dist:[0.7, 0.2, 0.08, 0.02]. On the other hand, if the scenario is characterized

by long distances, high multipath fading, and high noise density, the probability distributions would be N0:[0, 0.2, 0.15, 0.65]; Fade:[0, 0.2, 0.8]; Dist:[0.05, 0.15, 0.3, 0.5]. In each radio frequency channel, I find different levels of co-channel interference. Consequently, the probability distribution of the co-channel interference power, Co_Ch, changes according to the carrier frequency of the channel used by the communication system. This does not mean that the carrier frequency controls the level of interference; what it means is that the random behavior of the co-channel interference found in a channel differs from the one encountered in another channel. In the following simulations, as I change the value of the carrier frequency Fc, I assign different probability distributions to the variable Co_Ch.

I represent the policies through the probability distributions of the parameters. Although I consider the parameters as deterministic variables, I must assign probability distributions to them, so that they can make part of the influence diagram and Bayesian network. I simply assign probability 1 to the state that represents the current setup value of the parameter. For instance, let use this configuration or policy: transmission power (Tx) = 0dBm; carrier frequency (Fc) = 2400 MHz; time of symbol (Ts) = 1 microsecond; spectral efficiency (SE) = 1bps/Hz. The probability distributions for the parameters would be: Tx:[1, 0, 0, 0]; Fc:[0, 1, 0]; Ts:[1, 0, 0]; and SE:[1, 0, 0].

4.3.2 Results and Discussion

In this section, I present and discuss the results from simulations in different scenarios, wherein I calculated the utilities of several policies. In all these cases, I supposed that the co-channel interference power behaved according to these probability distributions: if Fc=Fc_1,

then Co_Ch:[0.2, 0.2, 0.3, 0.3, 0] ; if Fc=Fc_2, then Co_Ch: [0.1, 0.4, 0.4, 0.1, 0]; and if Fc=Fc_3, then Co_Ch:[0.4, 0.3, 0.2, 0.1, 0]. For each scenario, I show the ten policies with the highest utilities in descending order. I specify each policy by listing the values I assign to each parameter, and by the side of the policy I write down the utility. For instance, in the line

`[{'Fc': 'Fc_2', 'Ts': 'Ts_3', 'Tx': 'Tx_2', 'SE': 'SE_3'}, 72.98],`

`{'Fc': 'Fc_2', 'Ts': 'Ts_3', 'Tx': 'Tx_2', 'SE': 'SE_3'}` is the policy and `72.98` is the utility that the influence diagram has calculated for that policy.

Let us remember from chapter 3 what the values of the parameters represent: [Fc_1, Fc_2, Fc_3] = [915 MHz, 2400 MHz, 5800 MHz]; [Ts_1, Ts_2, Ts_3] = [1.0μs, 4.0μs, 20.0μs]; [Tx_1, Tx_2, Tx_3, Tx_4] = [0 dBm, 10 dBm, 20dBm, 30 dBm]; and [SE_1, SE_2, SE_3] = [1bps/Hz, 2bps/Hz, 3bps/Hz]

Scenario 1: In this scenario the distances tend to be short, the multipath fading low, and the noise power density medium. The probability distributions in this case are: N0:[0.05, 0.4, 0.5, 0.05]; Fade:[0.8, 0.2, 0]; Dist:[0.7, 0.2, 0.08, 0.02]; Co_Ch(Fc_1):[0.2, 0.2, 0.3, 0.3, 0]; Co_Ch(Fc_2): [0.1, 0.4, 0.4, 0.1, 0]; and Co_Ch(Fc_3):[0.4, 0.3, 0.2, 0.1, 0].

```
[{'Fc': 'Fc_2', 'Ts': 'Ts_3', 'Tx': 'Tx_2', 'SE': 'SE_3'}, 72.98]
[{'Fc': 'Fc_2', 'Ts': 'Ts_3', 'Tx': 'Tx_1', 'SE': 'SE_3'}, 72.59]
[{'Fc': 'Fc_2', 'Ts': 'Ts_2', 'Tx': 'Tx_2', 'SE': 'SE_3'}, 71.49]
[{'Fc': 'Fc_2', 'Ts': 'Ts_3', 'Tx': 'Tx_3', 'SE': 'SE_3'}, 70.99]
[{'Fc': 'Fc_1', 'Ts': 'Ts_3', 'Tx': 'Tx_2', 'SE': 'SE_3'}, 70.52]
[{'Fc': 'Fc_2', 'Ts': 'Ts_2', 'Tx': 'Tx_3', 'SE': 'SE_3'}, 70.12]
[{'Fc': 'Fc_2', 'Ts': 'Ts_2', 'Tx': 'Tx_1', 'SE': 'SE_3'}, 70.03]
[{'Fc': 'Fc_2', 'Ts': 'Ts_1', 'Tx': 'Tx_2', 'SE': 'SE_3'}, 69.81]
[{'Fc': 'Fc_1', 'Ts': 'Ts_3', 'Tx': 'Tx_1', 'SE': 'SE_3'}, 69.69]
[{'Fc': 'Fc_1', 'Ts': 'Ts_2', 'Tx': 'Tx_2', 'SE': 'SE_3'}, 69.2]
```

In this case the configuration with the highest utility, 72.98, is {'Fc': 'Fc_2', 'Ts': 'Ts_3', 'Tx': 'Tx_2', 'SE': 'SE_3'}, which means tuning the radio at 2400 MHz, using a time of symbol of 20

μsec, transmission power of 10 dBm, and D8PSK modulation, whose spectral efficiency is 3bps/Hz. Let us remember that this configuration is the best according to the preferences I have expressed through the utility function (UF) explained in chapter 3. This utility function has as arguments BER, SE, and Tx. BER is an indicator, which depends on the conditions of the environment and the parameters, whereas SE and Tx are parameters chosen by the user or the system. Since I have no preferences regarding the carrier frequency (Fc) and time of symbol (Ts), I did not include them as parameters of the UF. However, by calculating the utilities of different policies, the influence diagram can suggest which values of Fc and Ts can fit the best my preferences. The trend in this scenario is that the distances are short and the power fading due to multipath propagation is low, which causes the total attenuation to be low. This explain the fact that the policies that make the top of the list include low levels of power, Tx_1, and Tx_2, and medium frequency, Fc_2.

Scenario 2: In this scenario the distances are long, the multipath fading is high, and the noise power density is high. The probability distributions in this case are: N0:[0, 0.2, 0.15, 0.65]; Fade:[0, 0.2, 0.8]; Dist:[0.05, 0.15, 0.3, 0.5] ; Co_Ch(Fc_1):[0.2, 0.2, 0.3, 0.3, 0]; Co_Ch(Fc_2): [0.1, 0.4, 0.4, 0.1, 0]; and Co_Ch(Fc_3):[0.4, 0.3, 0.2, 0.1, 0].

```
[{'Fc': 'Fc_1', 'Ts': 'Ts_3', 'Tx': 'Tx_3', 'SE': 'SE_2'}, 3.99]
[{'Fc': 'Fc_1', 'Ts': 'Ts_3', 'Tx': 'Tx_2', 'SE': 'SE_2'}, 3.22]
[{'Fc': 'Fc_1', 'Ts': 'Ts_3', 'Tx': 'Tx_1', 'SE': 'SE_3'}, 2.03]
[{'Fc': 'Fc_1', 'Ts': 'Ts_3', 'Tx': 'Tx_1', 'SE': 'SE_2'}, -0.08]
[{'Fc': 'Fc_1', 'Ts': 'Ts_3', 'Tx': 'Tx_2', 'SE': 'SE_3'}, -0.22]
[{'Fc': 'Fc_1', 'Ts': 'Ts_2', 'Tx': 'Tx_3', 'SE': 'SE_2'}, -0.34]
[{'Fc': 'Fc_1', 'Ts': 'Ts_2', 'Tx': 'Tx_1', 'SE': 'SE_3'}, -0.54]
[{'Fc': 'Fc_1', 'Ts': 'Ts_2', 'Tx': 'Tx_2', 'SE': 'SE_2'}, -0.65]
[{'Fc': 'Fc_2', 'Ts': 'Ts_3', 'Tx': 'Tx_1', 'SE': 'SE_3'}, -0.97]
[{'Fc': 'Fc_1', 'Ts': 'Ts_3', 'Tx': 'Tx_2', 'SE': 'SE_1'}, -1.25]
```

This scenario has such difficult conditions that the probability of error (BER) tends to be high. This fact causes the utility of all the policies to be relatively low compared with the ones obtained in other scenarios, scenario 1 and scenario 3. I even got negative utilities. This result suggests that the configurations available to set up the communication system are unfit for the scenario represented by the probability distributions enunciated in the previous paragraph. Because of the long distances and high multipath fading, the total attenuation is high. This, combined with the high value noise density decreases EbN0, which in turn makes BER go up. One solution to this situation would be to use higher transmission power and lower carrier frequency Fc.

Scenario 3: In this scenario, the distances are medium, the multipath fading is medium, and the noise power density is low. The probability distributions in this case are: N0:[0.7, 0.2, 0.1, 0]; Fade:[0.1, 0.8, 0.1]; Dist:[0.1, 0.4, 0.5, 0] ; Co_Ch(Fc_1):[0.2, 0.2, 0.3, 0.3, 0]; Co_Ch(Fc_2): [0.1, 0.4, 0.4, 0.1, 0]; and Co_Ch(Fc_3):[0.4, 0.3, 0.2, 0.1, 0].

```
[{'Fc': 'Fc_2', 'Ts': 'Ts_3', 'Tx': 'Tx_3', 'SE': 'SE_3'}, 51.23]
[{'Fc': 'Fc_2', 'Ts': 'Ts_2', 'Tx': 'Tx_3', 'SE': 'SE_2'}, 51.03]
[{'Fc': 'Fc_2', 'Ts': 'Ts_2', 'Tx': 'Tx_3', 'SE': 'SE_3'}, 50.87]
[{'Fc': 'Fc_2', 'Ts': 'Ts_1', 'Tx': 'Tx_3', 'SE': 'SE_2'}, 49.13]
[{'Fc': 'Fc_2', 'Ts': 'Ts_1', 'Tx': 'Tx_3', 'SE': 'SE_3'}, 45.3]
[{'Fc': 'Fc_1', 'Ts': 'Ts_3', 'Tx': 'Tx_3', 'SE': 'SE_2'}, 44.95]
[{'Fc': 'Fc_1', 'Ts': 'Ts_2', 'Tx': 'Tx_3', 'SE': 'SE_2'}, 44.78]
[{'Fc': 'Fc_1', 'Ts': 'Ts_3', 'Tx': 'Tx_3', 'SE': 'SE_3'}, 43.66]
[{'Fc': 'Fc_1', 'Ts': 'Ts_2', 'Tx': 'Tx_3', 'SE': 'SE_3'}, 43.21]
[{'Fc': 'Fc_1', 'Ts': 'Ts_1', 'Tx': 'Tx_3', 'SE': 'SE_2'}, 43.17]
```

In this scenario, the conditions are not as good as in scenario 1, and not as bad as in scenario 2. In this case the configuration with the highest utility, 51.23, is {'Fc': 'Fc_2', 'Ts': 'Ts_3', 'Tx': 'Tx_3', 'SE': 'SE_3'}, which means tuning the radio at 2400 MHz, using a time of symbol of 20 μsec, transmission power of 20 dBm, and D8PSK modulation, whose spectral efficiency is

3bps/Hz. This utility is smaller than the maximum utility I got in scenario 1, 72.98. Since the conditions of the environment have changed, the marginal probability distribution of BER changed as well. This change makes the utility function UF yield a different number, which means the UF depends not only on the conditions of the environment but also on the policy. Hence, in the same scenario, the utility function (fed by the influence diagram) can tell which configurations or policies satisfy the best the preferences of the system.

CHAPTER 5

EXPERIMENTS ON SPECTRUM SENSING AND CHANNEL ESTIMATION

This chapter describes experiments performed on channel sensing and channel sounding, which are processes that a cognitive radio (CR) needs to acquire knowledge about the environment wherein it operates. First section describes experiments on spectrum sensing using an algorithm based on the autocorrelation of the received samples. Second section presents this method applied to scanning different channels in order to obtain their utilization levels. Third section describes method that uses Bayesian probability to learn the utilization of the channels based on previous and current observations. Finally, fourth section describes experiments on channel sounding.

5.1 Experiments on Spectrum Sensing.

5.1.1 Methodology

Let us define $x(n) = s(n) + \eta(n)$ as the received samples, where $s(n)$, is the primary user signal and $\eta(n)$ is the noise. Two hypotheses exist: 1) $\mathcal{H}_0$, i.e. absence of signal, and 2) $\mathcal{H}_1$, i.e. presence of signal. These hypotheses are given in [17, 18] and defined as

$$\mathcal{H}_0 : x(n) = \eta(n) \tag{5.1.1}$$

$$\mathcal{H}_1 : x(n) = s(n) + \eta(n) \ . \tag{5.1.2}$$

Let one define the vectors $\mathbf{x}$ and $\mathbf{s}$ as

$$\mathbf{x}(n) = [x(n) \;\; x(n-1) \;\; \cdots \;\; x(1)]^T , \tag{5.1.3}$$

and

$$\mathbf{s}(n) = [s(n) \;\; s(n-1) \;\; \cdots \;\; s(1)]^T . \tag{5.14}$$

The statistical covariance matrices of these vectors, defined in terms of the expectation E are

$$\mathbf{R}_{\mathbf{x}} = E[\mathbf{x}(n)\,\mathbf{x}^T(n)], \tag{5.1.5}$$

and

$$\mathbf{R}_{\mathbf{s}} = E[\mathbf{s}(n)\,\mathbf{s}^T(n)], \tag{5.1.6}$$

respectively.

According to [17], the matrix $\mathbf{R}_{\mathbf{x}}$ can be expressed as

$$\mathbf{R}_{\mathbf{x}} = \mathbf{R}_{\mathbf{s}} + \sigma_\eta^2 \mathbf{I} , \tag{5.1.7}$$

where σ_η^2 is the variance of the noise η, and $\mathbf{I}$ the identity matrix. Therefore, in absence of signal $\mathbf{R}_{\mathbf{s}}$ is zero as well as the non-diagonal elements of $\mathbf{R}_{\mathbf{x}}$. Based on this assumption, Zeng and Liang [17] have proposed the ratio between the sum of all the elements of $\mathbf{R}_{\mathbf{x}}$, $T_1 = \sum_n \sum_m |r_{nm}|$ and the sum of its diagonal elements, $T_2 = \sum_n |r_{nn}|$, as metric to detect the absence or presence of signal. In absence of signal this ratio is supposed to be one whereas with signal present this ratio is greater than one. It is clear that in the first case the ratio is $\frac{\sigma_\eta^2 \mathbf{I}}{\sigma_\eta^2 \mathbf{I}}$,

whereas in the second case it is $\frac{\mathbf{R}_s + \sigma_\eta^2 \mathbf{I}}{\sigma_\eta^2 \mathbf{I}}$; however, in the practice, even in absence of signal $\mathbf{R}_x$ is a non-diagonal matrix and $T_1 / T2 > 1$[19].

The assumption about η made in [17] is that $E(\eta(n)\eta(n+\tau)) = 0$ for any $\tau \neq 0$. This condition holds when the noise is Gaussian; the problem is that the noise of the USRP receiver is colored and non-delta correlated, something to consider when analyzing the covariance or autocorrelation of the signal in order to decide between $\mathcal{H}_0$ and $\mathcal{H}_1$.

In the proposed method I calculate the autocorrelation of the samples defined as

$$\lambda(l) = \sum_{m=0}^{N_s - 1} x(m)\, x^*(m-l), \tag{5.1.8}$$

where, N_s is the number of samples and the symbol * represents the complex conjugate operation. Nevertheless, rather than using $\lambda(l)$ to build a covariance matrix as in [17-19], I use the Euclidean distance between $\lambda(l)$ and a reference line.

This reference line corresponds to the ACF of the samples when the signal is strong enough to have certainty about its presence. Ideally this reference line is a slope. Figure 5.1.1 shows the ACF of the samples in the absence of signal. As one can see ACF is not delta function. This figure also shows that when the signal is strong enough, it practically coincides with the reference line. Therefore, comparing the ACF with the reference line can provide information about the presence or absence of signal. Since the reference line is a slope, its equation is given by $y = mx + b$ (equation of a line). I generate N_s points of this line and store them in a vector called ACF_{ref}. To perform the spectrum sensing I take N_s samples and store them in a vector

called ACF_{in}. The Euclidean distance between ACF_{in} and ACF_{ref}, called $D_{Euclidean}$, is the metric that I compare with a threshold γ_1 to decide between $\mathcal{H}_0$ and $\mathcal{H}_1$.

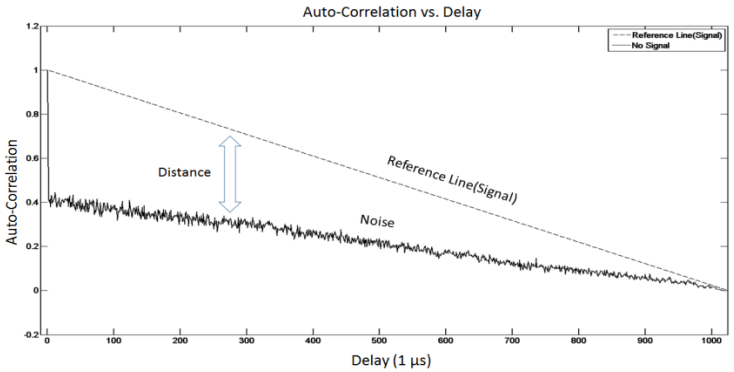


Figure 5.1.1: Distance from th ACF to a reference line as a metric for signal detection

The proposed method is compared with the autocorrelation at the first lag, $ACF(1)$. Ideally, $ACF(1) = 0$ when only noise exists, and $ACF(1) \neq 0$ when a signal and noise are present. In [10], $ACF(1)$ has been proposed as a metric for spectrum sensing [79]. However, my experiments show that $ACF(1) \neq 0$ in presence of only noise with low standard deviation. The reason for that is that the USRP generates colored noise with is not delta correlated; therefore, when this noise is stronger than the white noise received from the environment, the ACF is not delta correlated either. This fact can wrongly indicate that there is a signal present, when indeed it is the colored noise generated by the USRP the one that makes the ACF look like there is a signal. I attempted to solve this situation by subtracting the noise caused by the USRP receiver. To do this, I stored in a matrix several vectors containing N_s samples taken by the USRP with the signal generator turned off. Then, I calculated the average of all the vectors, stored it in η_{USRP} and subtracted it from the signal being received by the USRP during the experiments. After

subtracting, in absence of signal (only noise) I obtained an ACF that looked more like a delta function. Next section, figure 5.1.3, shows some results in which I can observe that this method helped in counteracting the effect of the USRP noise on the ACF.

Figure 5.1.2 shows the experimental setup used to perform the experiments. A signal generator was connected to a USRP N200 unit by means of a cable. The purpose of connecting them by a cable was to reduce the random external noise as much as possible. Reducing the external noise allowed me to generate noise samples, whose standard deviation I could control. For the simulations, I simulated standard normal noise samples, η_{Sim}, with different standard deviations, which were added to the signal samples collected by the USRP. I estimated the noise produced by the receiver, η_{USRP}, and subtracted it from the received samples as explained in the previous section. I also performed experiments without subtracting η_{USRP}. The processing of the signal was done in GNU Radio software using Python language and the USRP units served as interface between the analog RF domain and the digital software domain.

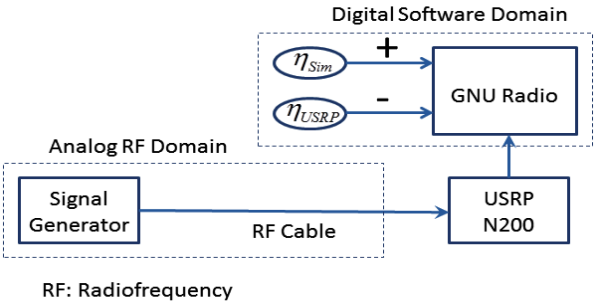


Figure 5.1.2: Experimental Setup.

5.1.2 Results and Discussion

Examples of some of the results obtained during the experiments are illustrated in figure 5.1.3. This figure represents the change in the behavior of ACF with respect to delay at different

levels of Signal-to-Noise Ratio (SNR) ranging from 0dB to +25 dB and when no signal is provided by the signal generator. With no signal the ACF should be 1 at 0 and zero, or almost zero, at the other values. However, at the left of figure 5.1.3, which shows the results obtained without subtracting the USRP noise, I observe that with no signal ACF is not zero at values of delay where it should be. The right of figure 5.1.3 shows that after subtracting η_{USRP}, the ACF obtained with no signal is as it should be: 1 at 0 and almost zero at the other values.

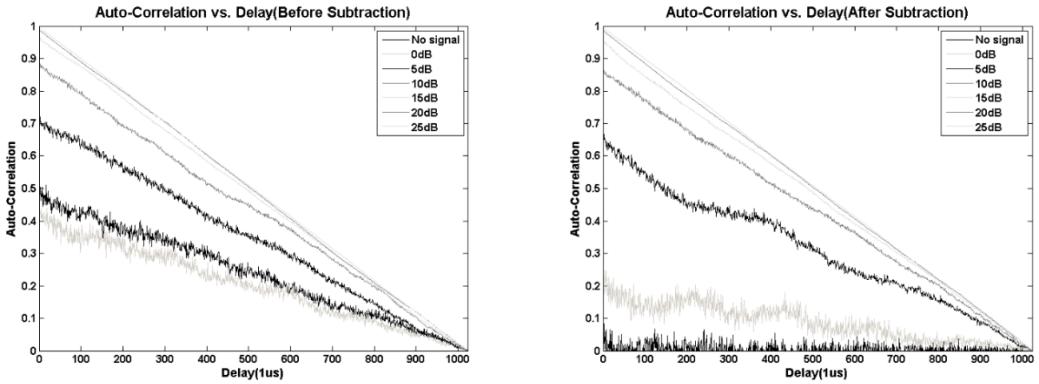


Figure 5.1.3: Behavior of the ACF with No signal and different values of SNR. Notice that the Euclidean distance from ACF to the reference line decreases as SNR increases. The standard deviation of η_{Sim} is 0.01.

Figure 5.1.4 represents the plots Normalized values of Euclidean Distance and ACF (1) with respect to SNR for different values of standard deviations of the simulated noise η_{Sim}. As one can see with no signal present, the ACF is the farthest from the reference line. Hence, at certain value of $D_{Euclidean}$, a threshold differentiates two situations: signal absent ($\mathcal{H}_0$) and signal present ($\mathcal{H}_1$). The left of figure 4 shows that the Euclidean distance exceeding certain threshold means absence of signal ($\mathcal{H}_0$), whereas when it is below that threshold means presence of signal ($\mathcal{H}_1$). On the other hand, the right of figure 4 shows that ACF (1) exceeding a threshold indicates presence of signal ($\mathcal{H}_1$), whereas when it is under the threshold means absence of signal ($\mathcal{H}_0$).

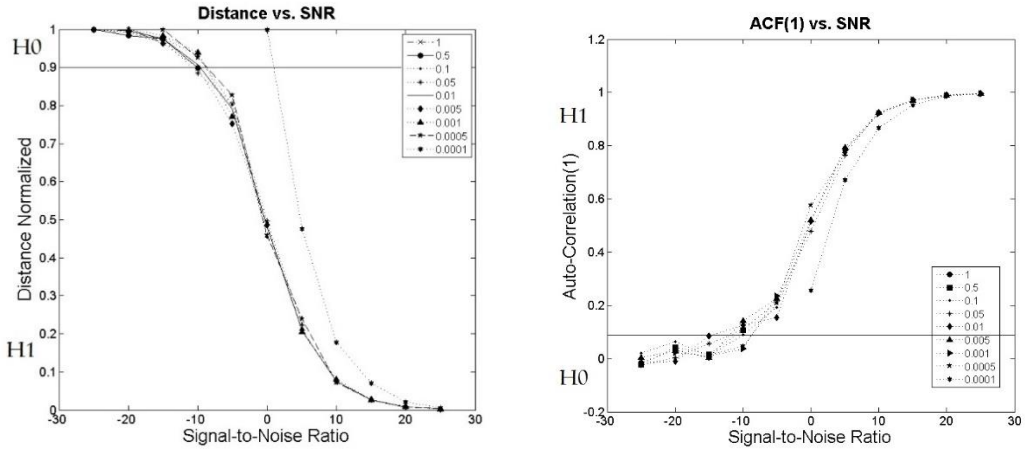


Figure 5.1.4: Behavior of $D_{Euclidean}$ as SNR changes with different standard deviations of η_{Sim} (Left). Behavior of ACF(1) as SNR changes with different standard deviations of η_{Sim} (Right).

To evaluate the impact of the threshold selection on the performance of the method, I performed experiments with different thresholds for both methods and calculated the probability of detection (P_d) for several signal to noise ratios. Examples of results are given in Figure 5.1.5. This figure represents the change of probability of detection with respect to SNR at different levels of threshold for both Euclidean distance method and ACF (1) method. As one can see the best threshold for the Euclidean distance method is 0.95 and for the ACF (1) method is 0.1, since they yield the highest P_d at the lowest SNR.

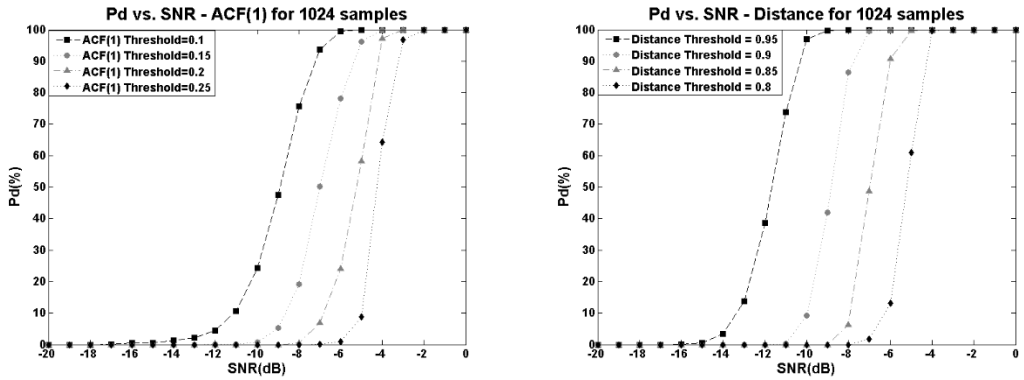


Figure 5.1.5: Behavior of Probability of Detection as SNR changes with different threshold levels of (left) ACF(1) and (right) $D_{Euclidean}$.

Finally, Figure 5.1.6 compares both methods set up at their best thresholds for 1024 samples. In this figure, the results clearly indicate that the Euclidean distance method outperforms the ACF (1) method, since it achieves a probability of detection $P_d = 100\%$ at a lower SNR. The difference in SNR is about 4 dB.

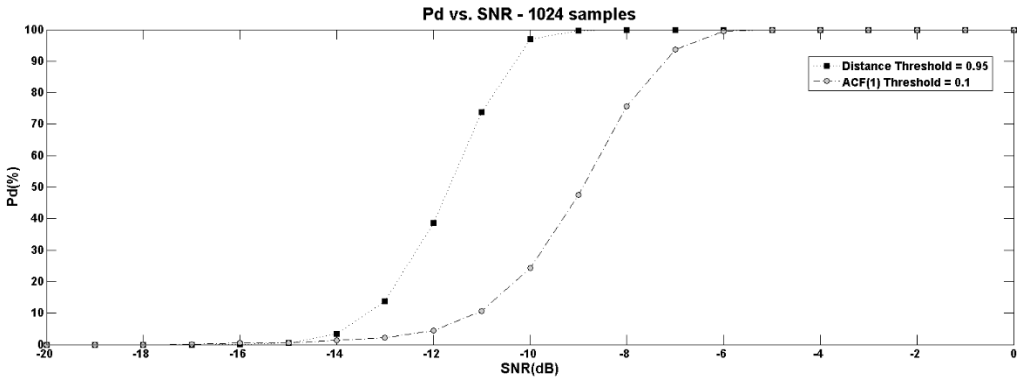


Figure 5.1.6: Behavior of Probability of Detection as SNR changes for 1024 samples and threshold levels of $D_{Euclidean}$ = 0.95 and ACF(1) = 0.1

The results have shown that the Euclidean distance method takes into consideration the inherent noise of the USRP receiver not assuming that the ACF in absence of signal is a delta function, which is the assumption of the covariance and eigenvalue based methods. The results of the experiments showed that the Euclidean distance method performs better than the ACF (1) method having a SNR gain of about 4 dB for a given SNR.

5.2 Channel Scanning Experiment

5.2.1 Methodology

The autocorrelation based spectrum sensing technique explained in the previous section can be extended to scan different channels in order to establish their utilization level. The objective behind determining the utilization level of channels is for the cognitive radio (CR) to have a

selection of the least used channels to choose from, so in case it needs to share the radio spectrum with other devices, it has some knowledge to pick a channel wherein it will experience fewer collisions. Cognitive Radio technology contemplates dynamic spectrum access (DSA) and spectrum sharing not only in unlicensed bands but also in low utilization licensed bands. The motivation for this is to exploit the spectrum as much as possible. Since the utilization of channels varies according to location, time, and frequency, the CR needs updated information about this variable, which it can get by scanning the channels of interest. Conventional spectrum analyzers have the channel scanning utility, which works by measuring the power in each channel and comparing it with a threshold. This approach follows the same principle of the energy detection method mentioned in chapter 2. One drawback that the energy detection method has is that it makes no difference between signal and noise power, which can cause the spectrum sensor to detect a signal when indeed only noise exists. I propose scan the channels of interest using the autocorrelation based technique exposed in the previous section. As explained before this technique analyzes the autocorrelation function of received samples to determine if they contain only noise or signal plus noise. I think this technique serves well the purpose of determining the utilization level of a determined channel, since I need to know if there are signals being transmitted more than I need to know their power.

I scanned channels in four bands: 850MHz, 1910MHz, 2400MHz and 5800 MHz. The first two bands are licensed and have been allocated to mobile communications, whereas the last two bands are unlicensed and used for different wireless communication systems such as WiFi, Bluetooth, and Zigbee. These systems share this bands. In the 850 MHz band, the central frequencies in MHz for the scanned channels were: 824, 827, 829, 832, 834, 837, 839, 842, 844,

847, 849, 869, 872, 874, 877, 879, 882, 884, 887, 889, 892, and 894. In the 1910 MHz band the channels were: 1850, 1856, 1862, 1868, 1874, 1880, 1886, 1892, 1898, 1904, 1910, 1930, 1936, 1942, 1948, 1954, 1960, 1966, 1972, 1978, 1984, 1990. In the 2400MHz band, the central frequencies in MHz for the scanned channels were: 2400, 2402, 2407, 2412*, 2417*, 2422*, 2427*, 2432*, 2437*, 2442*, 2447*, 2452*, 2457*, 2462*, 2467*, 2472*, 2477, 2482, 2484, 2487, 2492, and 2497; where the asterisks indicate that the frequency is a standard Wi-Fi channel. In the 5800 MHz band, the scanned channels were: 5725, 5730, 5735, 5740, 5745, 5750, 5755, 5760, 5765, 5770, 5775, 5780, 5785, 5790, 5795, 5800, 5805, 5810, 5815, 5820, 5825, 5830, 5835, 5840, 5845, 5850, 5855, 5860, 5865, 5870, and 5875.

I used two USRP N200 units: one for the 850MHz and 1910MHz bands, and the other one for the bands 2400 MHz and 5800 MHz. Due to limitations in the hardware each USRP unit scanned only one channel at a time. The following are the steps followed during this channel scanning process:

1. Tune the USRP unit to the central frequency of the first channel of the list.
2. Take 1024 samples.
3. Execute the Euclidean distance autocorrelation based algorithm.
4. Make the decision as whether or not the channel is busy or empty. The decision has two values: 0 represents the channel is empty, and 1 represents the channel is occupied.
5. Store the decision taken along with the current time and central frequency of the channel.
6. Go to the next channel in the list and repeat the procedure.
7. After scanning all the channels in the list go back to the first channel and repeat until the experiment is over.

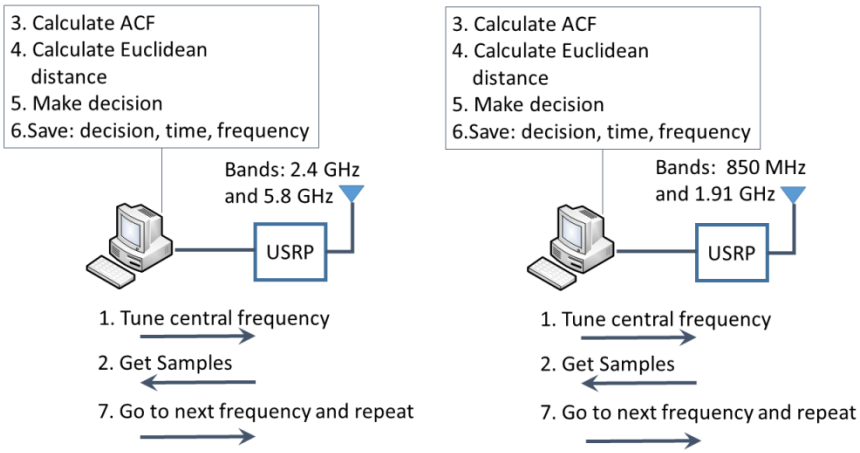


Figure 5.2.1 Scanning several channels with the Euclidean distance autocorrelation based sensing method. The arrows in steps 1,2,and 7 indicate the exchange of information between the computer and the USRP. Steps 3 to 4 take place in the computer

After finishing the experiment, I processed the data. This process is summarized in the following steps:

1. Filter the data according the day of the week, the interval of time, and the central frequency. I divided the day into five intervals: midnight: [12:00 AM to 6 AM); morning: [6 AM to 12 PM); afternoon: [12 PM to 4 PM); evening: [4PM to 8 PM), and night: [8 PM to 12 AM). A bracket indicates that the interval includes the value following it, whereas a parenthesis indicates that the interval excludes the value preceding it. For instance, the interval midnight includes 12:00 AM but not 6 AM. After filtering, I will have the results yielded by the sensor (ones or zeros) grouped by time interval, and carrier frequency.
2. For each time interval and frequency I store the results (1 and 0) in an array. At the end of this step I will have an array of length L, where L equals the number of times the sensor has observed that particular channel within the time interval.
3. For each time interval and frequency I sum the elements of the corresponding L long array and divide by L to get an estimate of the utilization level at that particular time interval and

frequency.

4. At the end of this process, I will have an estimate of the utilization level for each frequency and time interval.

5.2.2 Results and Discussion

Figures 5.2.2 and 5.2.3 summarize the results obtained during one week of scanning several channels at the 850 MHz, 1910 MHz, 2400 MHz, and 5800 MHz bands. What I observe for each time interval is the average of the channel utilization levels estimated for the time interval per each day of the week. I observed a marked contrast between the utilization level of the channels in the 850 MHz and 1910 MHz and the one of the 2400 MHz and 5800 MHz band channels. In the first two bands I observed occupancy levels ranging between 25% and 100 %; whereas, in the last two bands the occupancy level falls in the range 0.5 % to around 7 %. The 850 MHz and 1910 MHz band are licensed and have been allocated to the cell phone communication system. According to this in the area where performed the experiment, the 850 MHz and 1910 MHz band channels are not good candidates for neither dynamic spectrum access (DSA) nor spectrum sharing. Conversely, the low utilization of the 2.4 GHz and 5.8 GHz channels make them good candidates for spectrum sharing and DSA.

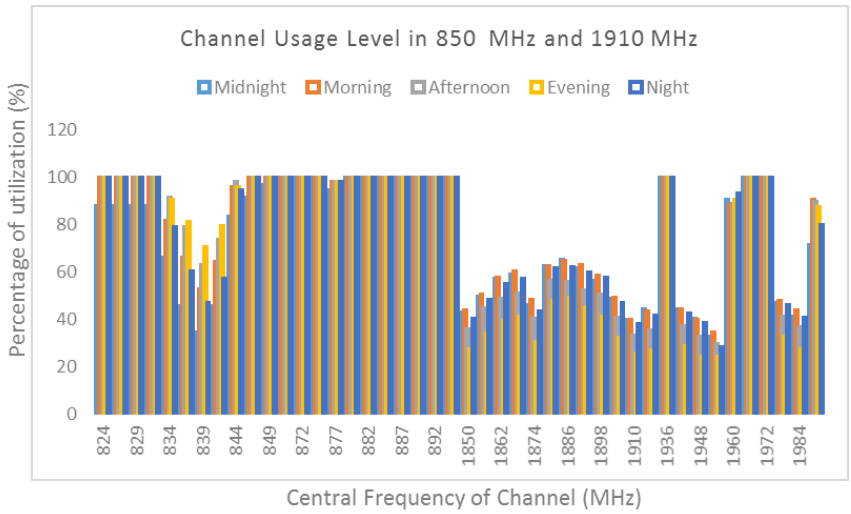


Figure 5.2.2 Results of the scanning of channels in the band of 2.4 GHz and 5.8 GHz obtained during a one week period in the Signal processing laboratory of the Electrical Engineering Department at the University of North Dakota.

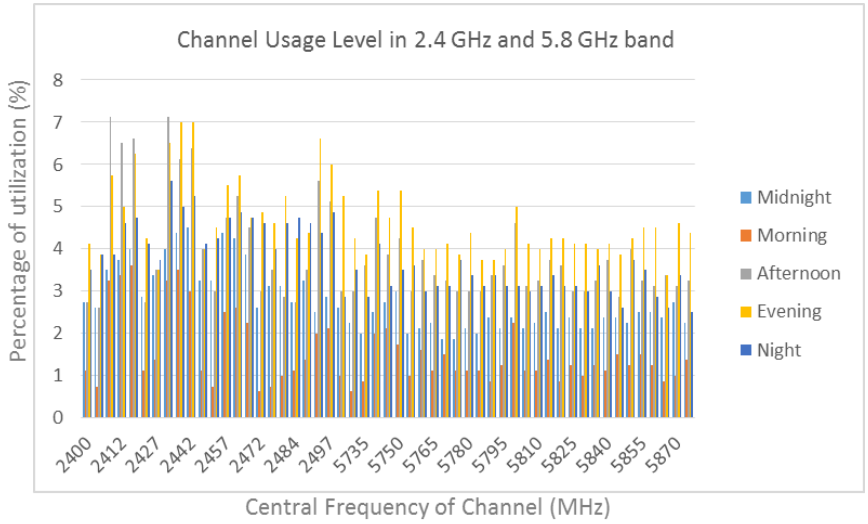


Figure 5.2.3 Results of the scanning of channels in the band of 2.4 GHz and 5.8 GHz obtained during a one week period in the Signal processing laboratory of the Electrical Engineering Department at the University of North Dakota.

Notice in figure 5.2.3 that the utilization level is higher in the 2.4 GHz band than in the 5.8 GHz band, something that makes sense considering the higher number of devices that operate at range of frequency compared to the ones that operate in the 5.8 GHz range. I also can see that the occupancy is the lowest during the morning and the highest during the evening and afternoon.

5.3 Bayesian Approach for Learning Spectrum Utilization

5.3.1 Methodology

In the previous experiment I estimated the use of several channels by sampling several times a counting the number of occurrences at the end to calculate the percentage of utilization of the channels. The numbers obtained can be seen as a survey of the channel usage during a period of time. In such experiment, there was no learning involved. In the experiment described in this section, the radio starts ignorant of the channel usage, and as it scans the channel it learns about its utilization level. I can see the utilization level as a number between 0 and 1 meaning 0% and 100% usage respectively. The channel utilization level is a random variable that follows a probability distribution. The objective of this experiment is to learn that probability distribution for the channels of interest. I start the experiment with a prior probability distribution for the channel utilization level. I use a uniform distribution, since I am just starting the experiment and have no knowledge about this variable. To obtain evidence, I observe the channel with the Euclidean distance autocorrelation based sensor explained in section 5.1. This sensor yields 1 when it detects a signal being transmitted in the channel (channel occupied), and 0 when it sees only noise (channel not occupied). The cognitive radio adjusts the probability distributions of the channels as it receives new evidence from the sensor.

The steps followed for estimating the channel utilization level using the Bayesian approach are:

1. Start with a uniform probability distribution for the channel utilization level.
2. Tune the USRP unit to the central frequency of the channel of interest.
3. Take 1024 samples.

4. Execute the Euclidean distance autocorrelation based algorithm.
5. Make the decision as whether or not the channel is busy or empty. The decision has two values: 0 represents the channel is empty, and 1 represents the channel is occupied.
6. Use the observation from step 5 as evidence for updating the prior probability distribution to obtain the posterior probability distribution of the channel utilization level.
7. Save the updated probability distribution periodically, and at the end of each time interval, so that the next opportunity the cognitive radio (CR) observes the channel at the beginning of that time interval, it starts with a prior knowledge of the channel usage level corresponding to that particular time interval. For instance, the CR observes the channel during the time interval I have called morning, [6 AM to 12 PM); the first day at 6 AM it starts with a uniform prior probability. At 12 PM the CR stores the posterior probability that will be used the next day at 6 AM as prior probability.
8. Start again in step 1. But this time use the posterior probability from step 6 as prior probability.

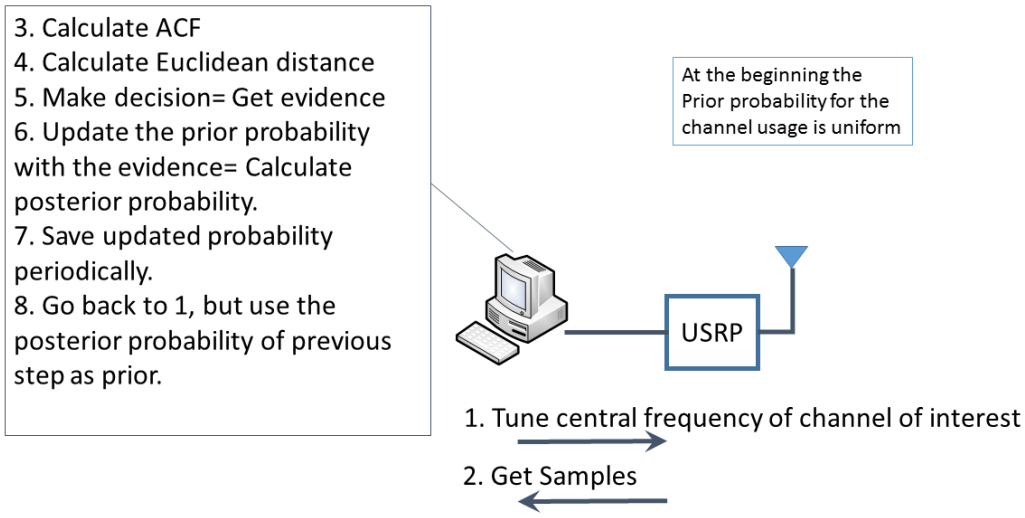


Figure 5.3.1 Using Bayesian probability for estimating the usage level of one or several wireless channels. The arrows in steps 1,2 indicate the exchange of information between the computer and the USRP. Steps 3 to 8 take place in the computer

5.3.2 Results and Discussion

I think the best way to present the results from this experiment is to illustrate how the probability distribution of the channel utilization evolves. For simplicity I monitored only one channel, 2410 MHz; however, the method can scan several channels like in the process described in section 5.2. Figure 5.3.2.a shows that I start with a uniform distribution. Figure 5.3.2.b shows the posterior probability obtained after observing the channel once. Notice that utilization 0% has the highest probability and the probability for the other utilization levels descend linearly. Figure 5.3.2.c shows that with two samples the probability for utilization 0% is still the highest; however the probabilities for the other levels reduce faster. After 10 observations, figure 5.3.2.d, the maximum is no longer at 0% but has shifted to a value around 10%, 0.1 in the horizontal scale. With 100 observations, figure 5.3.2.e, the peak moves to around 5% (0.05 in the horizontal scale) and the curve narrowed. Figure 5.3.2.f shows that with 1000 observations the peak stayed at 5% and the curve narrowed even more. This example has shown the CR using Bayesian

probability to learn how much a radio frequency channel is being utilized.

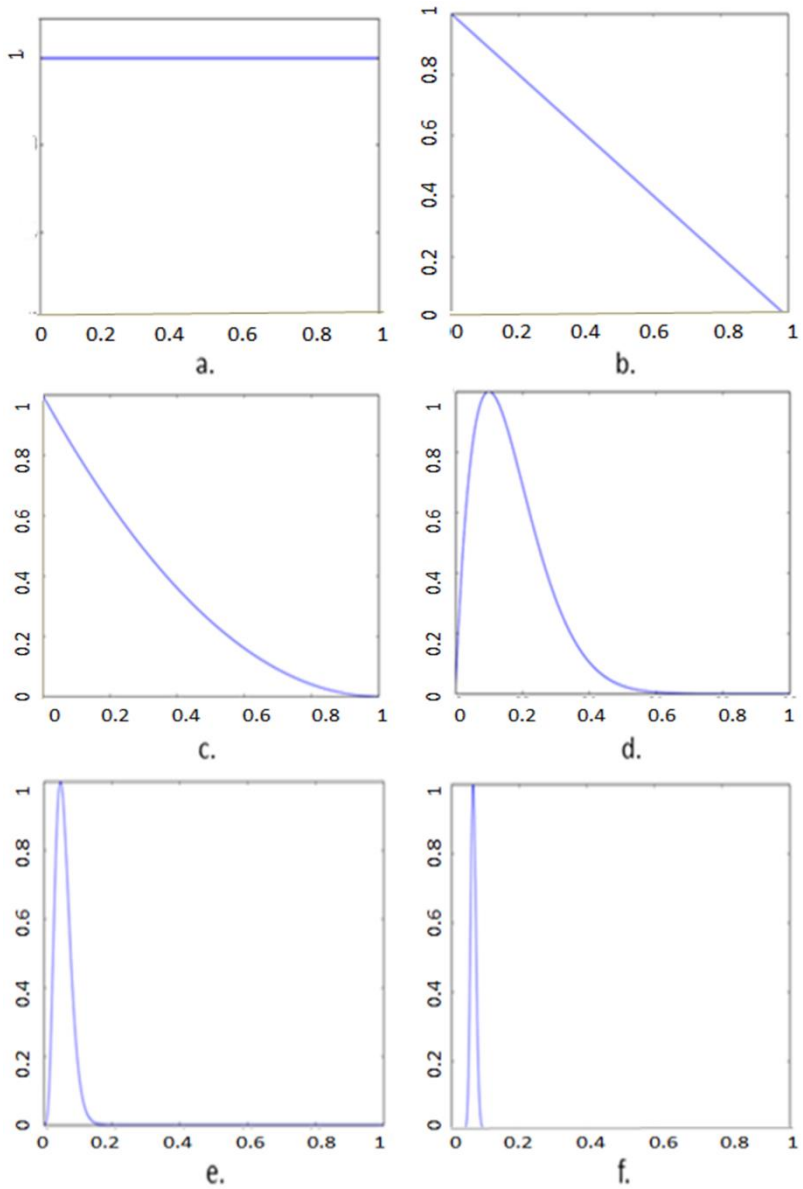


Figure 5.3.2 Evolution of the probability distribution for the channel utilization level. The horizontal axis represents the utilization level, where 0 is 0% and 1 is 100%. The vertical axis is the probability, which goes from 0 to 1. a.) Prior probability distribution; b.)Posterior probability distribution after 1 observation; c.) Posterior probability distribution after 2 observations; d.) Posterior probability distribution after 10 observations; e.) Posterior probability distribution after 100 observations; f.) Posterior probability distribution after 1000 observations. The curves are normalized to set up the maximum at 1.

5.4 Channel Sounding Experiment

5.4.1 Methodology

Channel Impulse Response

The channel is characterized by estimating the channel impulse response (CIR) and calculating two parameters from it: Doppler Spread and Delay Spread, which condense the information provided by the CIR. These parameters lead to the *coherence bandwidth* and *coherence time*. The coherence bandwidth, B_c, imposes restrictions over the bandwidth of the signal to transmit through the channel. Likewise, the coherence time, T_c limits the symbol time [1]. Matz and Hlawatsch [21] provide a definition of the coherence bandwidth and the coherence time as shown in equation (5.4.1).

$$B_c = \frac{1}{S_\tau}, \qquad T_c = \frac{1}{S_\nu}, \tag{5.4.1}$$

where S_τ and S_ν represents the Delay Spread and Doppler Spread respectively.

Knowing $B_c, S_\tau, T_c,$ and S_ν helps the communication system -for instance, a cognitive radio- to adapt its operating configuration to fit better with the current conditions of the channel.

The channel impulse response (CIR) – represented as $h(t,\tau)$ in equation 5.4.2 – indicates the number of paths used by the signal to propagate, the attenuation on each path, and the relative delay between paths.

$$h(t,\tau) = \sum_{i=1}^{L} a_i(t)\delta(\tau - \tau_i), \tag{5.4.2}$$

where t is the time, τ is the delay, $a_i(t)$ is the time varying complex attenuation for the path i, the delta function $\delta(\tau-\tau_i)$ represents the path i with delay τ_i, and L is the number of paths [36]. The Fourier transform of $h(t,\tau)$ with respect to t yields $s(\nu,\tau)$ – the scattering function – which shows the change of the paths and the shift of the central frequency due to the Doppler effect.

Condense Parameters of the Channel

The Delay spread and the Doppler spread are the normalized second order central moments of the power delay profile (PDP) and the Doppler power spectrum (DPS) [1, 37, 38] . To calculate PDP and DPS $h(t,\tau)$ and $s(\nu,\tau)$ are considered stochastic processes, which is necessary, since they are unpredictable in the practice [21]. To simplify $h(t,\tau)$ and $s(\nu,\tau)$ I use the autocorrelation function (ACF) and assume that the channel is wide sense stationary – uncorrelated scattering (WSSUS) [21, 39]. For instance, by applying the ACF to $h(t,\tau)$ I have [37]

$$R_h(t_1,t_2,\tau_1,\tau_2)=E[h(t_1,\tau_1)h^*(t_2,\tau_2)] \tag{5.4.3}$$

where h^* is the complex conjugate of h and $E[\cdot]$ is the expected value operation. The WSSUS model, which is broadly accepted for mobile channels [39], has two assumptions. The first assumption is that the stochastic process is wide sense stationary, WSS, which implies that the ACF depends only on $\Delta t=t_2-t_1$, and not on the absolute time, t. Therefore, equation (5.4.3) becomes

$$R_h(\Delta t,\tau_1,\tau_2)=E[h(t,\tau_1)h^*(t+\Delta t,\tau_2)]. \tag{5.4.4}$$

The second assumption is that the amplitudes and phases of the different paths are uncorrelated, which means the channel has uncorrelated scattering, US. Therefore, the ACF is zero when $\tau_1 \neq \tau_2$ and has a peak when $\tau_1 = \tau_2$. By applying this assumption to equation (5.4.4) it becomes

$$R_h(\Delta t, \tau) = E[h(t,\tau)h^*(t+\Delta t,\tau)] \ , \tag{5.4.5}$$

which calculated at $\Delta t = 0$ yields the function $p_h(\tau) = R_h(\tau) = R_h(\tau, 0)$ or *power delay profile* – PDP [37]. The PDP represents the distribution of the power among the delayed paths of the signal arriving at the receiver. By normalizing the PDP, it turns into a probability density function, designated as $p(\tau)$. Equation (5.4.6) shows this normalization.

$$p(\tau) = \frac{R_h(\tau)}{\int_{-\infty}^{\infty} R_h(\tau)d\tau} = \frac{p_h(\tau)}{\int_{-\infty}^{\infty} p_h(\tau)d\tau} \ . \tag{5.4.6}$$

The normalized second order central moment of $p(\tau)$ is

$$S_\tau = \sqrt{\int_{-\infty}^{\infty} (\tau - D_\tau)^2 p(\tau)d\tau} \ , \tag{5.4.7}$$

the *delay spread*. In equation (5.4.7)

$$D_\tau = E[\tau] = \int_{-\infty}^{\infty} \tau p(\tau)d\tau \ , \tag{5.4.8}$$

the *mean delay*.

Since in the practice, only a limited number of discrete signals avail, I use the discrete versions of equations (5.4.7) and (5.4.8) as given by [37, 40]

$$S_\tau = \sqrt{\frac{\sum (\tau_i - D_\tau)^2 p_h(\tau_i)}{\sum p_h(\tau_i)}} \ , \tag{5.4.9}$$

where

$$D_\tau = \frac{\sum \tau_i p_h(\tau_i)}{\sum p_h(\tau_i)} \ . \tag{5.4.10}$$

A similar process works when calculating the Delay Spread – S_ν; the integral of the scattering function $s(\nu,\tau)$ with respect to τ yields $p_D(\nu)$, known as the Doppler spectrum. The equations (5.4.6) to (5.4.10) applied to $p_D(\nu)$ return S_ν.

Estimation of the Channel Impulse Response and the Condense Parameters of the Channel

To estimate $h(t,\tau)$, S_τ, and S_ν a pseudo-random (PN) sequence is transmitted through the channel and its autocorrelation calculated at the receiver. This method bases on the fact that the autocorrelation of white noise is an impulse [80]. Since sending white noise through a channel is impractical, I used a signal PN sequence because its autocorrelation resembles the autocorrelation of white noise [81]. Figure 5.4.1 shows that the transmitter and receiver combine of GNU Radio software along with a USRP (Universal Software Radio Peripheral) unit. The USRP TX sends a PN sequence through a channel with response $h(t,\tau)$. The USRP RX takes the signal from the channel, processes and delivers it to the PN correlator block that calculates the autocorrelation to obtain $\hat{h}(t,\tau)$ – an estimate of $h(t,\tau)$. The next block takes this estimate to calculate S_τ and S_ν using the aforementioned equations. The PN sequence originates from a Galois linear feedback shift register (GLFSR) generator [82]. A GLFSR generator has a polynomial, whose degree n determines the length sequence L according to $L = 2^n - 1$. The

GLFSR and PN correlator blocks have both two parameters: mask and degree, which must agree to calculate a autocorrelation, otherwise the PN correlator would calculate the cross-correlation. The configuration used for the experiments was: degree 9 and mask 0 to get a 511 bits long autocorrelation sequence. Each autocorrelation sequence represents the channel impulse response at certain instant t_i. Several of these sequences arranged one after another form $h(t,\tau)$. Figure 5.4.2 in the next section illustrates examples of $h(t,\tau)$ and $s(\nu,\tau)$.

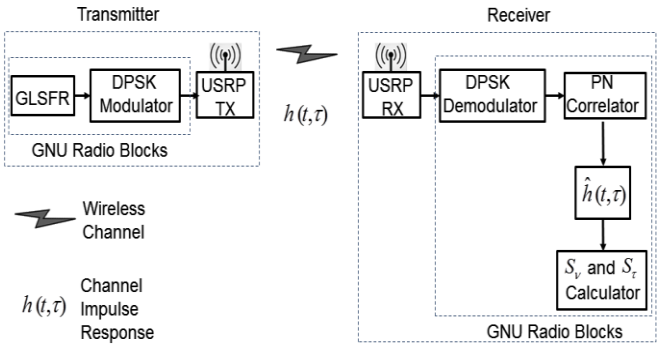


Figure 5.4.1: Block Diagram of the Channel Sounder

5.4.2 Results and Discussion

The experiments were performed at different environments – an anechoic chamber, a parking lot surrounded by buildings, and a street located between two parking lots – and at the frequencies: 850MHz, 1910MHz, 2410MHz and 5850MHz. The two first frequencies are commonly used in cellphone networks and two last ones belong to the group of ISM (Industrial, Scientific, and Medical) bands, which are unlicensed and prevalently used in wireless networks, such as Wi-Fi, Bluetooth and Zigbee. The Delay and Doppler spread were calculated every 8 seconds, 220 times per each experiment. The experiments in the anechoic chamber were performed with and without interference; a signal generator was adjusted at 5 and 10 dBm to

create two levels of interference. The outdoor experiments were made in the morning and in the afternoon observing the surrounding activity, such as movement of cars and people (their portable devices), in order to see how the results were affected.

Figure 5.4.2 provides examples of the channel impulse response $h(t,\tau)$ and scattering function $s(\nu,\tau)$ obtained during the experiments performed in the parking lot. As one can see Figures 5.4.2a and 5.4.2b show only one path and no Doppler shift, whereas figures 5.4.2c and 5.4.2d show multiple paths and Doppler shifts. The sampling rate was configured to one million of samples per second, which set the bit rate at 1 Mbps and the time resolution at 1 μs. Figures 5.4.2a and 5.4.2c show how $h(t,\tau)$ forms from putting each autocorrelation one after another along the axis, *"Time (us)"*. Since each autocorrelation sequence is 511 bits long and the time of bit is 1 μsec, its duration is 511 μsec. Therefore, $h(t,\tau)$ is sampled every 511 μsec, which corresponds to the sampling time, T_{samp}.The inverse of T_{samp} is the sampling rate, F_s, which is 1956 Hz. Figures 5.4.2b and 5.3.2d illustrate $s(\nu,\tau)$, the Fourier transform of $h(t,\tau)$ with respect to time. Since F_s is 1956 Hz, the axis *"Doppler frequency (Hz)"* in figures 5.4.2b and 5.4.2d ranges between $-\frac{F_s}{2} = -978Hz$ and $\frac{F_s}{2} = 978Hz$ [83].

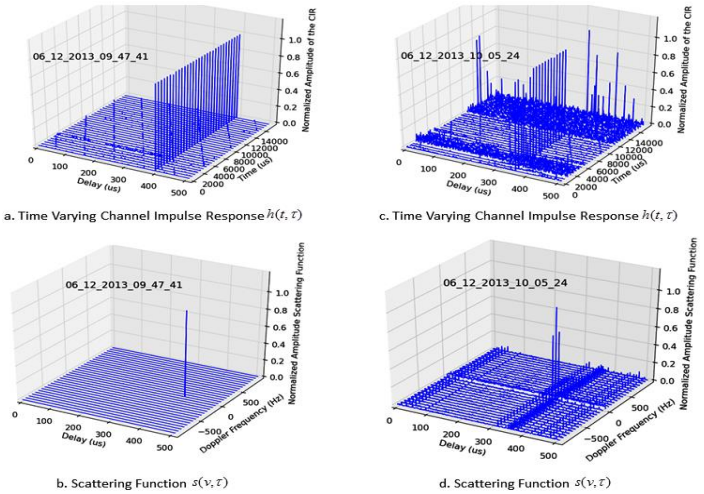


Figure 5.4.2: Examples of time varying impulse response $h(t,\tau)$ and scattering function $s(\nu,\tau)$ obtained during experiments performed in a parking lot.

Experiments in Controlled Environment

Figure 5.4.3 shows the Delay and Doppler Spread functions in the anechoic chamber. As expected in such environment free of reflections and movement, the Delay and Doppler spreads were zero. Figures 5.4.4 and 5.4.5 show examples of results obtained using a continuous signal as a source of interference. As expected, the Delay and Doppler functions consistently differed from zero and concentrated around the average value. Experiments performed at other frequencies - 5850 MHz, 1910 MHz, and 850 MHz - yielded similar results, Delay and Doppler spread functions were equal to zero in absence of interference, whereas with interference they always differed from zero and their averages increased as the power of the interference increased.

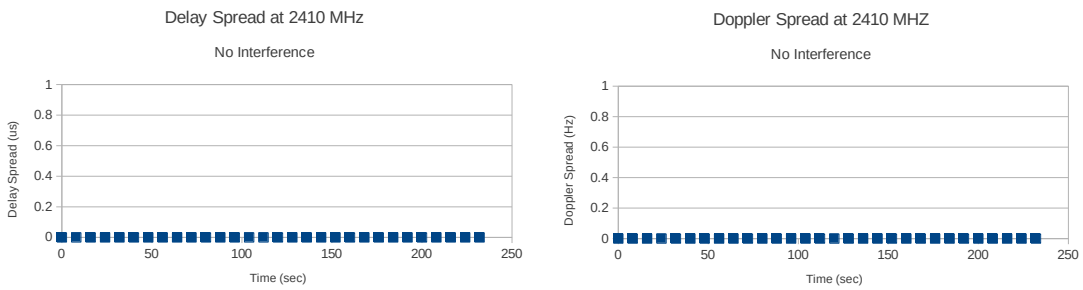


Figure 5.4.3: Example of results at the anechoic chamber with no signal generator.

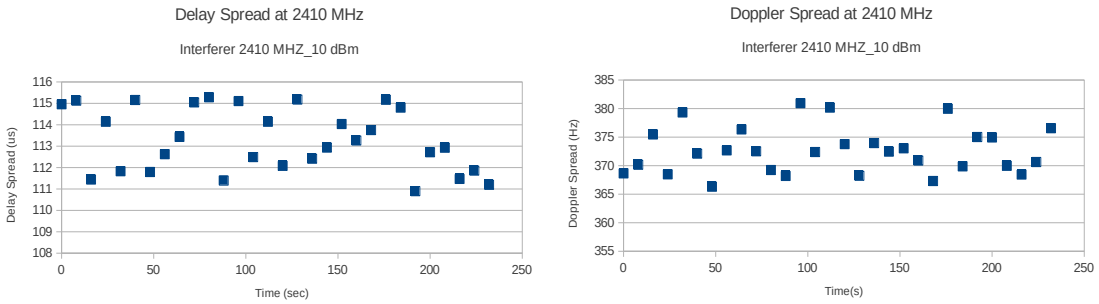


Figure 5.4.4: Example of results at the anechoic chamber with signal generator at 10 dBm.

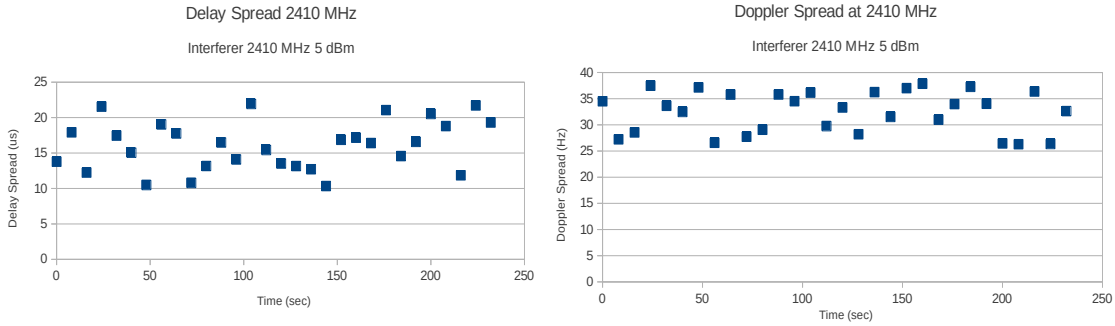


Figure 5.4.5: Example of results at the anechoic chamber with signal generator at 5 dBm.

Experiments in Outdoor Environment

Figures 5.4.6 through 5.4.9 give examples of results obtained in a parking lot at 5850 MHz, 2410 MHz, 1910 MHz and 850 MHz during the afternoon. Experiments performed during the morning produced similar behavior. As one can see, most of the results were zero and the few

non-null values ranged in a wide scope. Figures 5.4.10 through 5.4.13 give examples of the results obtained in a street located between two parking lots at the same frequencies comport alike.

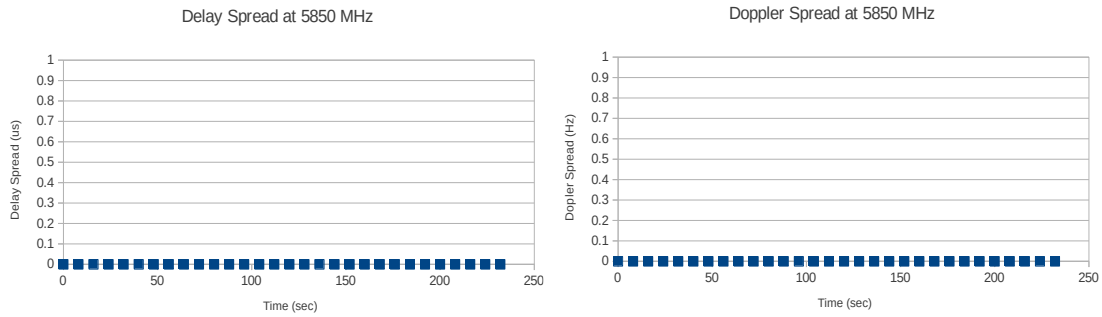


Figure 5.4.6: Example of results obtained in the parking lot surrounded by buildings at 5850 MHz.

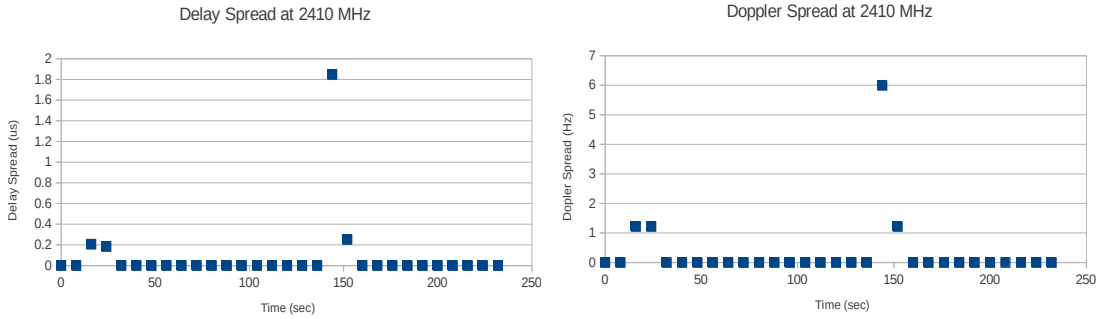


Figure 5.4.7: Example of results obtained in the parking lot surrounded by buildings at 2410 MHz.

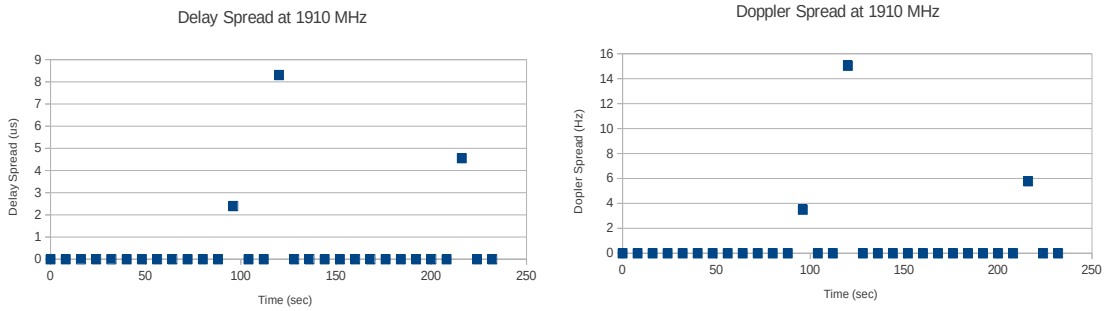


Figure 5.4.8: Example of results obtained in the parking lot surrounded by buildings at 1910 MHz.

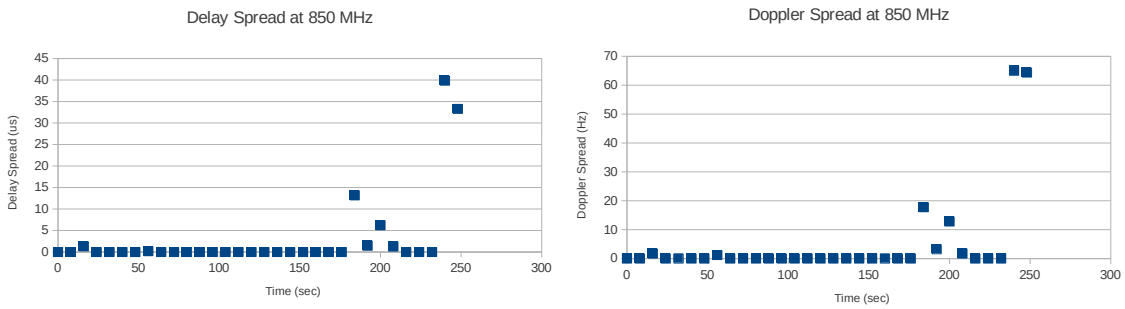


Figure 5.4.9: Example of results obtained in the parking lot surrounded by buildings at 850 MHz.

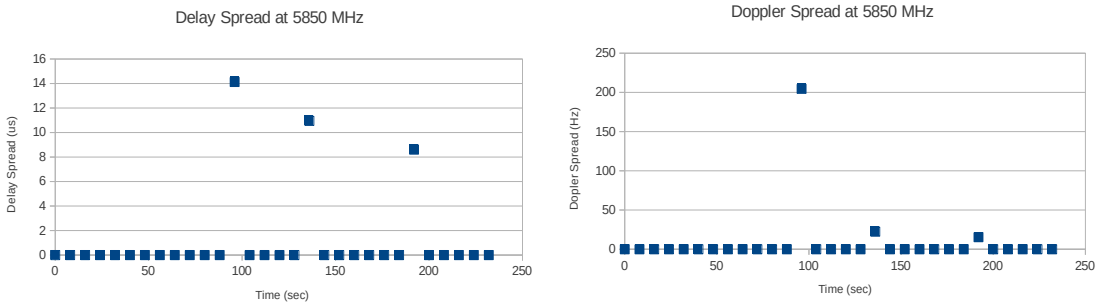


Figure 5.4.10: Example of results for experiments obtained in a street between two parking lots at 5850 MHz.

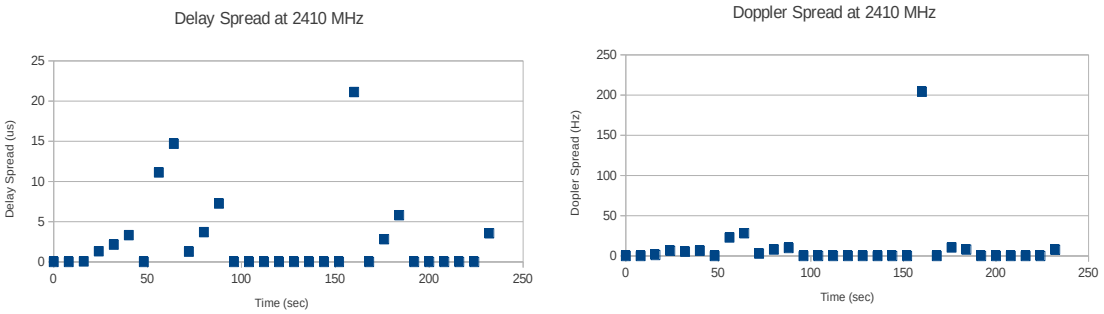


Figure 5.4.11: Example of results for experiments obtained in a street between two parking lots at 2410 MHz.

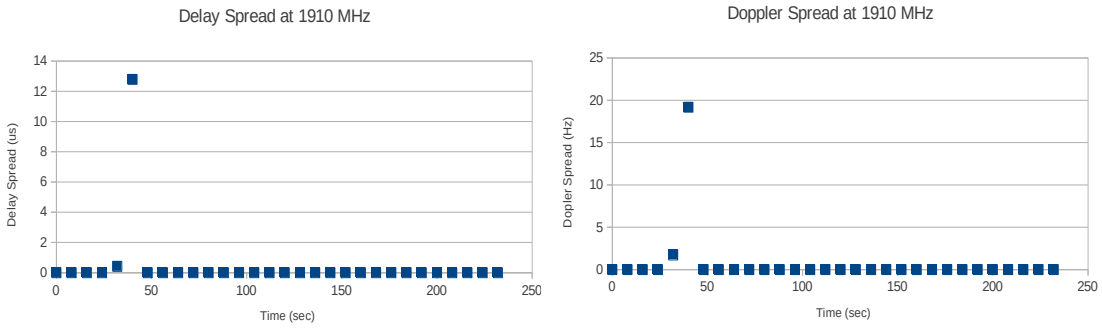


Figure 5.4.12: Example of results for experiments obtained in a street between two parking lots at 1910 MHz.

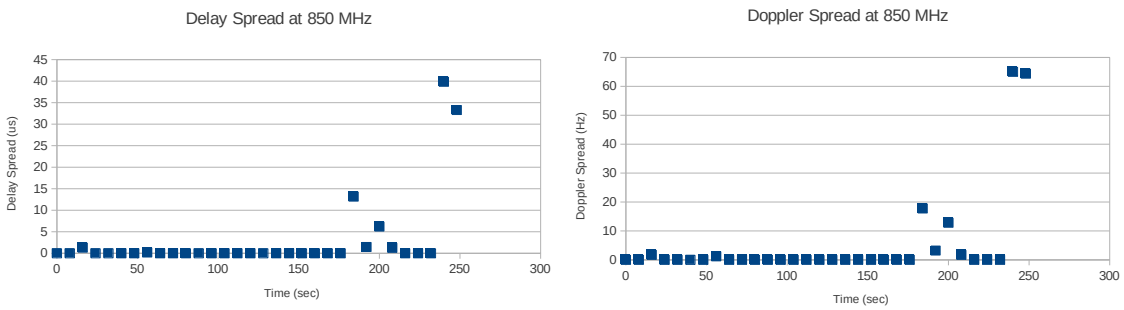


Figure 5.4.13: Example of results for experiments obtained in a street between two parking lots at 850 MHz.

Table 5.4.1 shows that in the anechoic chamber the results for Delay and Doppler spread rose around eight times when the power of the signal generator increased by 5 dB. In the other scenarios – outdoors without controlled interference – the numbers were much lower and equaled zero in most of the cases. In most cases, the numbers obtained in the street exceeded the numbers obtained in the parking lot. 2410 MHz registered the highest values, whereas 5850 MHz registered the lowest. For the outdoor scenarios the values changed from morning to afternoon, although in some cases the change was small, such as at 5850 MHz in the parking lot and at 2410 MHz in the street. Table 1 shows that in the anechoic chamber the results for Delay and Doppler spread rose around eight times when the power of the signal generator increased by 5 dB. In the other scenarios – outdoors without controlled interference – the numbers were much lower and equaled zero in most of the cases. In most cases, the numbers obtained in the street exceeded the numbers obtained in the parking lot. 2410 MHz registered the highest values, whereas 5850 MHz registered the lowest. For the outdoor scenarios the values changed from morning to afternoon, although in some cases the change was small, such as at 5850 MHz in the parking lot and at 2410 MHz in the street.

In presence of interference, the results obtained at the anechoic chamber for Delay and Doppler spread differed consistently from zero. This means that interference influenced the results given by the channel sounder, since the anechoic chamber is an environment free of reflections and scatterers. The impact of the interference on the results depends on its intensity as shown in table 5.4.1. In the outdoor experiments the outcomes also agreed with what I expected; the results mostly equated zero, which makes sense, considering that the distances between the channel sounder and sources of multipath or Doppler shift, such as reflectors and scatterers, were less than 300 meters, distance needed to have delays of 1 μsec – the time resolution of my channel sounder. However, some results differed from zero. For instance, some Doppler Spread results differed from zero even in absence of movement, which means they must have come from interference.

Table 5.4.1: Average delay spread, Doppler spread and percentage of non-null results

Frequency (MHz.)	Scenario and Conditions*	AVG_Del** (μsec.)	AVG_Dopp** (Hz.)	Non-null ** (%)
5810	Parking Lot_Morning	0.00	0.07	0.9
	Parking Lot_Afternoon	0.00	0.00	0
	Street_Morning	0.42	1.98	4.5
	Street_Afternoon	0.52	2.15	5.4
2410	Anechoic Chamber- No interference	0.00	0.00	0.00
	Anechoic Chamber_5 dBm	16.27	31.58	100
	Anechoic Chamber_10 dBm	113.14	373.27	100
	Parking Lot_Morning	0.39	0.30	5
	Parking Lot_Afternoon	0.08	0.29	7.2
	Street_Morning	1.00	3.45	8.2
	Street_Afternoon	0.96	3.59	9
1910	Parking Lot_Morning	0.04	0.08	0.9
	Parking Lot_Afternoon	0.15	0.25	3.2
	Street_Morning	0.23	0.40	4.1
	Street_Afternoon	0.13	0.23	2.3
850	Parking Lot_Morning	0.29	0.89	4
	Parking Lot_Afternoon	0.43	1.94	5.4
	Street_Morning	1.43	2.50	7.2
	Street_Afternoon	0.89	1.57	6.4

***Conditions** can be the presence and absense of interference; or morning and afternoon

****AV_Del:** Average of Delay Spread ; **AV_Dopp**: Average of Delay Spread; **Non-Null:** Percentage of non-null numbers

** Averages and percentages calculated out of 220 numbers

During the outdoor experiments I observed an association between the frequency and the non-null results. For example, the percentages of non-null results at 2410 MHz exceeded those obtained at 5850 MHz. This observation coincides with the fact that 2410 MHz is more common than 5850 MHz. At 1910 MHz the results were smaller compared to those at 850 MHz. A possible explanation is that the higher the frequency, the higher the propagation losses; therefore, at 1910 MHz the signal and the interference are more attenuated, which can explain the results shown in table 5.4.1 and figures 5.4.6 to 5.4.13. The non-null results obtained in the outdoor experiments were fewer and sparser than those obtained in the anechoic chamber, because in the anechoic chamber the interference was constant, whereas in the outdoor scenarios it was intermittent and changed its intensity randomly. Another association observed during these experiments was between the transit of cars and non-null Delay and Doppler spread results; in the experiments performed in the street, when the cars passed between transmitter and receiver the results differed from zero. A possible explanation for this is that the cars attenuated the signal and/or created multipath signals as they interrupted the line of sight between the transmitter and receiver affecting the results.

CONCLUSIONS

In this dissertation, I have presented a probabilistic approach to cognitive radio (CR) technology. I have identified the variables that interact in a wireless communication system, and seen them either deterministic or random variables. This variable identification has included the specification of the state domain of each variable. I have also established how these variables influence one to another by applying the knowledge on communication system available to me, as well simulations to generate data that allow me to establish the causal relationships among variables. For instance, from the literature I know that the received power (Rx) equals the transmitted power (Tx) minus the total propagation losses (Attn), therefore I used this knowledge to generate samples that the MLE (maximum likelihood estimation) algorithm used the learn the conditional probability distributions (CPD) that represent the influence that the variables Tx and Attn have over the variable Rx.

With the outcome from the process described in the previous paragraph, I built a Bayesian network (BN) and an influence diagram (ID) to learn from evidence, bit error rate (BER), and make better informed decisions in the midst of uncertainty, the random conditions of the environment wherein the systems functions. To show the functionality of these probabilistic graphical models, I assumed different scenarios and different setups of the radio. I also supposed the bit error rate (BER) falling at different states according to some trend: sometimes

tending to be low, and others high. Through several examples, I demonstrated how the BN modified its beliefs (probability distributions) as it got new evidence. As for the influence diagram (ID) I assumed scenarios with distinct conditions, which I represented through probability distributions in agreement with the scenarios under consideration. I showed and explained how the ID yielded different utility values for the same policy, configuration, being implemented under different conditions, scenarios.

As I discussed earlier in this document, a CR gets information from the environment (observing stage), reasons, learns from it, decides how to act (decision making stage), and acts (taking action stage). Since one important aspect of the observing stage is spectrum sensing, in this work I proposed and tested a technique for spectrum sensing that exploits the autocorrelation of the received samples. Let us remember that the purpose of spectrum sensing is to determine when a wireless channel is empty or being used by other device's radio transmission. After testing the proposed spectrum sensing technique in one channel, I used it to scan several channels in order to estimate their utilization level. To perform such estimation I used two methods: frequentist and Bayesian. In the frequentist method, I got the channel utilization estimate after having observed the channel several times. On the other hand, in the Bayesian method, I considered the channel utilization as a random variable having uniform initial probability distribution. This probability distribution got adjusted with each observation of the channel, which provides a constantly updated estimation of the channel utilization.

To the best of my knowledge, this is the first work that proposes a probabilistic graphical model (Bayesian Network and Influence Diagram) to assist a cognitive radio (CR) in dealing with uncertainty in wireless communication systems. I have proposed how to analyze and

characterize several random variables that affect communication systems, which has included the definition of ranges for those variables and their division in intervals represented through states. Specifically, I have shown the causal relations among variables and proposed methods for eliciting the conditional probability distributions that describe quantitatively those causal relations.

I have shown through some simulations that the proposed Bayesian network is able to update its beliefs (represented as probability distributions) about the environment as it gets new evidence. In my model, the state of the bit error rate (BER) serves as evidence. I can think of this ability as a capability that the wireless communication system has for learning from experience (evidence), although because of the uncertainty present in the environment, this learning is probabilistic.

The idea of influence diagrams for decision making in wireless communication is another contribution of this dissertation. By means of an influence diagram (ID), a cognitive radio can evaluate quantitatively how well a determined set of parameters fits with its goals and preferences. To do so, the ID includes a node called utility function, which assigns numbers to the combinations of the parameters and the value of BER. I showed that the utility of a policy differs from one scenario to another. I can think of the utility as a measure of the level of convenience or satisfaction. A policy that is convenient (high utility) for the system in one scenario could be less convenient or even inconvenient (low or negative utility) in other scenario. In this dissertation, I presented an example wherein I have high preference for setups with which I obtain low BER with low transmission power and high spectral efficiency; hence the utility function of the proposed ID takes as arguments: BER, spectral efficiency, and transmission

power. However, I can also extend the ID to applications where I have preferences about frequency, bandwidth, and other parameters. For instance, it has been foreseen that wireless channels will be auctioned or rented; therefore, I would like to consider the cost of renting when deciding which channel to use. Influence diagrams are a tool I could consider in such application.

This dissertation is the starting point of a long term research on probabilistic graphical modelling applied to cognitive radio and wireless communication system. In this work, I have used simulations to demonstrate the functionality of the proposed Bayesian models. I consider that future work should include the integration of these models with a software defined radio (SDR) platform such as USRP radios. After integrating these parts I will be able to perform experiments in controlled as well as in realistic scenarios.

Future work includes adding to the model other random variables such as packet delivery ratio (PDR), bad packet ratio (BPR), clear channel assessment (CCA), and received signal strength indicator (RSSI). These random variables are performance metrics that are commonly available in communication systems, but currently unexploited. By describing in my model how the environment influences these metrics, I can use them to enrich my evidence to update the beliefs (probability distributions) the Bayesian network has about the factors that affect the performance of the communication system.

In this work, I have represented the utility function as a table; however, as I increase the number of variables and the number of discrete states taken for each variable, this table format might become tedious. Therefore, future work should include searching for ways to generate

utility functions in a more friendly way, such as a piece of code that maps my preferences to numbers creating a table compatible with the format used by the influence diagram.

In this dissertation, I have shown how to use probabilistic graphical models (PGM) to assist a cognitive radio (CR) in learning from evidence and deciding what the most optimal configuration is. However, wireless communication systems have other needs that I can explore how to satisfy with PGM. For instance, one need in cognitive radio systems using dynamical spectrum access is to select the most convenient radio frequency channel to tune in. This particular area of cognitive radio is called spectrum decision. Upcoming work derived from this dissertation include building probabilistic graphical models to support cognitive radio in this task, spectrum decision. areas of future work include: investigate how fade depth and fade duration affect the bit error rate (BER), and incorporate this knowledge into the Bayesian model; include methods in the model to take into consideration the intermodulation products and the adjacent channel interference (ACI) that might arise when selecting and combining radio frequency channels.

APPENDIX 1

CONDITIONAL PROBABILITY TABLES FOR BIT ERROR RATE (BER)

In this appendix we present the conditional probability distributions for the variable bit error rate (BER) for different modulation schemes: DBPSK, DQPSK, and D8PSK. The tables are divided into two sections: one titled *states of the parents*, which contains the combinations of the states of the variables EbN0, C/I, and Dop_Phi; and the other one titled *probability distribution*, which contains the probability for each state of BER.

Table A.1.1: Conditional probability distribution (CPD) for BER in DBPSK.

States of the parents			Probability Distribution of BER				
EbN0_1	C/I_1	Phi_1	0	0	0	0	1
EbN0_1	C/I_1	Phi_2	0	0	0	0	1
EbN0_1	C/I_1	Phi_3	0	0	0	0	1
EbN0_1	C/I_2	Phi_1	0	0	0	0	1
EbN0_1	C/I_2	Phi_2	0	0	0	0	1
EbN0_1	C/I_2	Phi_3	0	0	0	0	1
EbN0_1	C/I_3	Phi_1	0	0	0	0	1
EbN0_1	C/I_3	Phi_2	0	0	0	0	1
EbN0_1	C/I_3	Phi_3	0	0	0	0	1
EbN0_1	C/I_4	Phi_1	0	0	0	0	1
EbN0_1	C/I_4	Phi_2	0	0	0	0	1
EbN0_1	C/I_4	Phi_3	0	0	0	0	1
EbN0_1	C/I_5	Phi_1	0	0	0	0	1
EbN0_1	C/I_5	Phi_2	0	0	0	0	1

Table A.1.1: Conditional probability distribution (CPD) for BER in DBPSK. (Cont.)

EbN0_1	C/I_5	Phi_3	0	0	0	0	1
EbN0_1	C/I_6	Phi_1	0	0	0	0	1
EbN0_1	C/I_6	Phi_2	0	0	0	0	1
EbN0_1	C/I_6	Phi_3	0	0	0	0	1
EbN0_2	C/I_1	Phi_1	0	0	0	0	1
EbN0_2	C/I_1	Phi_2	0	0	0	0	1
EbN0_2	C/I_1	Phi_3	0	0	0	0	1
EbN0_2	C/I_2	Phi_1	0	0	0.1	0.31	0.59
EbN0_2	C/I_2	Phi_2	0	0	0.075	0.39	0.535
EbN0_2	C/I_2	Phi_3	0	0	0.1	0.27	0.63
EbN0_2	C/I_3	Phi_1	0	0.09	0.28	0.405	0.225
EbN0_2	C/I_3	Phi_2	0.005	0.06	0.295	0.36	0.28
EbN0_2	C/I_3	Phi_3	0	0.065	0.255	0.415	0.265
EbN0_2	C/I_4	Phi_1	0	0.08	0.22	0.44	0.26
EbN0_2	C/I_4	Phi_2	0.005	0.095	0.175	0.47	0.255
EbN0_2	C/I_4	Phi_3	0	0.12	0.19	0.405	0.285
EbN0_2	C/I_5	Phi_1	0	0.11	0.25	0.385	0.255
EbN0_2	C/I_5	Phi_2	0	0.09	0.17	0.46	0.28
EbN0_2	C/I_5	Phi_3	0	0.08	0.22	0.41	0.29
EbN0_2	C/I_6	Phi_1	0.01	0.07	0.235	0.455	0.23
EbN0_2	C/I_6	Phi_2	0	0.06	0.245	0.435	0.26
EbN0_2	C/I_6	Phi_3	0	0.105	0.19	0.445	0.26
EbN0_3	C/I_1	Phi_1	0	0	0	0	1
EbN0_3	C/I_1	Phi_2	0	0	0	0	1
EbN0_3	C/I_1	Phi_3	0	0	0	0	1
EbN0_3	C/I_2	Phi_1	0.065	0.22	0.245	0.165	0.305
EbN0_3	C/I_2	Phi_2	0.04	0.29	0.205	0.18	0.285
EbN0_3	C/I_2	Phi_3	0.045	0.265	0.19	0.17	0.33
EbN0_3	C/I_3	Phi_1	0.505	0.49	0.005	0	0
EbN0_3	C/I_3	Phi_2	0.605	0.39	0.005	0	0
EbN0_3	C/I_3	Phi_3	0.425	0.575	0	0	0
EbN0_3	C/I_4	Phi_1	0.69	0.31	0	0	0
EbN0_3	C/I_4	Phi_2	0.66	0.34	0	0	0
EbN0_3	C/I_4	Phi_3	0.62	0.38	0	0	0
EbN0_3	C/I_5	Phi_1	0.675	0.325	0	0	0
EbN0_3	C/I_5	Phi_2	0.645	0.355	0	0	0
EbN0_3	C/I_5	Phi_3	0.595	0.405	0	0	0

Table A.1.1: Conditional probability distribution (CPD) for BER in DBPSK. (Cont.)

EbN0_3	C/I_6	Phi_1	0.68	0.32	0	0	0
EbN0_3	C/I_6	Phi_2	0.7	0.3	0	0	0
EbN0_3	C/I_6	Phi_3	0.635	0.365	0	0	0
EbN0_4	C/I_1	Phi_1	0	0	0	0	1
EbN0_4	C/I_1	Phi_2	0	0	0	0	1
EbN0_4	C/I_1	Phi_3	0	0	0	0	1
EbN0_4	C/I_2	Phi_1	0.31	0.165	0.085	0.12	0.32
EbN0_4	C/I_2	Phi_2	0.33	0.185	0.08	0.125	0.28
EbN0_4	C/I_2	Phi_3	0.32	0.145	0.09	0.135	0.31
EbN0_4	C/I_3	Phi_1	1	0	0	0	0
EbN0_4	C/I_3	Phi_2	1	0	0	0	0
EbN0_4	C/I_3	Phi_3	0.975	0.025	0	0	0
EbN0_4	C/I_4	Phi_1	1	0	0	0	0
EbN0_4	C/I_4	Phi_2	1	0	0	0	0
EbN0_4	C/I_4	Phi_3	1	0	0	0	0
EbN0_4	C/I_5	Phi_1	1	0	0	0	0
EbN0_4	C/I_5	Phi_2	0.995	0.005	0	0	0
EbN0_4	C/I_5	Phi_3	0.995	0.005	0	0	0
EbN0_4	C/I_6	Phi_1	1	0	0	0	0
EbN0_4	C/I_6	Phi_2	1	0	0	0	0
EbN0_4	C/I_6	Phi_3	1	0	0	0	0
EbN0_5	C/I_1	Phi_1	0	0	0	0	1
EbN0_5	C/I_1	Phi_2	0	0	0	0	1
EbN0_5	C/I_1	Phi_3	0	0	0	0	1
EbN0_5	C/I_2	Phi_1	0.475	0.085	0.06	0.13	0.25
EbN0_5	C/I_2	Phi_2	0.405	0.07	0.08	0.085	0.36
EbN0_5	C/I_2	Phi_3	0.46	0.1	0.105	0.085	0.25
EbN0_5	C/I_3	Phi_1	1	0	0	0	0
EbN0_5	C/I_3	Phi_2	1	0	0	0	0
EbN0_5	C/I_3	Phi_3	1	0	0	0	0
EbN0_5	C/I_4	Phi_1	1	0	0	0	0
EbN0_5	C/I_4	Phi_2	1	0	0	0	0
EbN0_5	C/I_4	Phi_3	1	0	0	0	0
EbN0_5	C/I_5	Phi_1	1	0	0	0	0
EbN0_5	C/I_5	Phi_2	1	0	0	0	0
EbN0_5	C/I_5	Phi_3	1	0	0	0	0
EbN0_5	C/I_6	Phi_1	1	0	0	0	0

Table A.1.1: Conditional probability distribution (CPD) for BER in DBPSK. (Cont.)

EbN0_5	C/I_6	Phi_2	1	0	0	0	0
EbN0_5	C/I_6	Phi_3	1	0	0	0	0
EbN0_6	C/I_1	Phi_1	0	0	0	0	1
EbN0_6	C/I_1	Phi_2	0	0	0	0	1
EbN0_6	C/I_1	Phi_3	0	0	0	0	1
EbN0_6	C/I_2	Phi_1	0.65	0	0.025	0.07	0.255
EbN0_6	C/I_2	Phi_2	0.63	0.03	0.015	0.08	0.245
EbN0_6	C/I_2	Phi_3	0.605	0.04	0.025	0.07	0.26
EbN0_6	C/I_3	Phi_1	1	0	0	0	0
EbN0_6	C/I_3	Phi_2	1	0	0	0	0
EbN0_6	C/I_3	Phi_3	1	0	0	0	0
EbN0_6	C/I_4	Phi_1	1	0	0	0	0
EbN0_6	C/I_4	Phi_2	1	0	0	0	0
EbN0_6	C/I_4	Phi_3	1	0	0	0	0
EbN0_6	C/I_5	Phi_1	1	0	0	0	0
EbN0_6	C/I_5	Phi_2	1	0	0	0	0
EbN0_6	C/I_5	Phi_3	1	0	0	0	0
EbN0_6	C/I_6	Phi_1	1	0	0	0	0
EbN0_6	C/I_6	Phi_2	1	0	0	0	0
EbN0_6	C/I_6	Phi_3	1	0	0	0	0

Table A.1.2: Conditional probability distribution (CPD) for BER in DQPSK

States of the parents			Probability Distribution of BER				
EbN0_1	C/I_1	Phi_1	0	0	0	0	1
EbN0_1	C/I_1	Phi_2	0	0	0	0	1
EbN0_1	C/I_1	Phi_3	0	0	0	0	1
EbN0_1	C/I_2	Phi_1	0	0	0	0	1
EbN0_1	C/I_2	Phi_2	0	0	0	0	1
EbN0_1	C/I_2	Phi_3	0	0	0	0	1
EbN0_1	C/I_3	Phi_1	0	0	0	0	1
EbN0_1	C/I_3	Phi_2	0	0	0	0	1
EbN0_1	C/I_3	Phi_3	0	0	0	0	1
EbN0_1	C/I_4	Phi_1	0	0	0	0	1
EbN0_1	C/I_4	Phi_2	0	0	0	0	1
EbN0_1	C/I_4	Phi_3	0	0	0	0	1
EbN0_1	C/I_5	Phi_1	0	0	0	0	1
EbN0_1	C/I_5	Phi_2	0	0	0	0	1
EbN0_1	C/I_5	Phi_3	0	0	0	0	1
EbN0_1	C/I_6	Phi_1	0	0	0	0	1
EbN0_1	C/I_6	Phi_2	0	0	0	0	1
EbN0_1	C/I_6	Phi_3	0	0	0	0	1
EbN0_2	C/I_1	Phi_1	0	0	0	0	1
EbN0_2	C/I_1	Phi_2	0	0	0	0	1
EbN0_2	C/I_1	Phi_3	0	0	0	0	1
EbN0_2	C/I_2	Phi_1	0	0	0	0.2	0.8
EbN0_2	C/I_2	Phi_2	0	0	0	0.165	0.835
EbN0_2	C/I_2	Phi_3	0	0	0	0.155	0.845
EbN0_2	C/I_3	Phi_1	0	0	0	0.47	0.53
EbN0_2	C/I_3	Phi_2	0	0	0	0.47	0.53
EbN0_2	C/I_3	Phi_3	0	0	0	0.47	0.53
EbN0_2	C/I_4	Phi_1	0	0	0	0.465	0.535
EbN0_2	C/I_4	Phi_2	0	0	0	0.56	0.44
EbN0_2	C/I_4	Phi_3	0	0	0	0.505	0.495
EbN0_2	C/I_5	Phi_1	0	0	0.005	0.405	0.59
EbN0_2	C/I_5	Phi_2	0	0	0	0.46	0.54
EbN0_2	C/I_5	Phi_3	0	0	0	0.455	0.545
EbN0_2	C/I_6	Phi_1	0	0	0.005	0.465	0.53
EbN0_2	C/I_6	Phi_2	0	0	0	0.52	0.48
EbN0_2	C/I_6	Phi_3	0	0	0	0.465	0.535

Table A.1.2: Conditional probability distribution (CPD) for BER in DQPSK. (Cont.)

EbN0_3	C/I_1	Phi_1	0	0	0	0	1
EbN0_3	C/I_1	Phi_2	0	0	0	0	1
EbN0_3	C/I_1	Phi_3	0	0	0	0	1
EbN0_3	C/I_2	Phi_1	0	0	0.085	0.43	0.485
EbN0_3	C/I_2	Phi_2	0	0	0.095	0.445	0.46
EbN0_3	C/I_2	Phi_3	0	0	0.075	0.4	0.525
EbN0_3	C/I_3	Phi_1	0	0.025	0.795	0.18	0
EbN0_3	C/I_3	Phi_2	0	0	0.75	0.25	0
EbN0_3	C/I_3	Phi_3	0	0	0.725	0.275	0
EbN0_3	C/I_4	Phi_1	0	0.09	0.865	0.045	0
EbN0_3	C/I_4	Phi_2	0	0.025	0.88	0.095	0
EbN0_3	C/I_4	Phi_3	0	0	0.825	0.175	0
EbN0_3	C/I_5	Phi_1	0	0.14	0.82	0.04	0
EbN0_3	C/I_5	Phi_2	0	0.03	0.86	0.11	0
EbN0_3	C/I_5	Phi_3	0	0	0.825	0.175	0
EbN0_3	C/I_6	Phi_1	0	0.09	0.845	0.065	0
EbN0_3	C/I_6	Phi_2	0	0.045	0.9	0.055	0
EbN0_3	C/I_6	Phi_3	0	0.005	0.82	0.175	0
EbN0_4	C/I_1	Phi_1	0	0	0	0	1
EbN0_4	C/I_1	Phi_2	0	0	0	0	1
EbN0_4	C/I_1	Phi_3	0	0	0	0	1
EbN0_4	C/I_2	Phi_1	0	0.065	0.205	0.22	0.51
EbN0_4	C/I_2	Phi_2	0	0.01	0.23	0.275	0.485
EbN0_4	C/I_2	Phi_3	0	0.01	0.21	0.335	0.445
EbN0_4	C/I_3	Phi_1	0.08	0.785	0.135	0	0
EbN0_4	C/I_3	Phi_2	0.04	0.77	0.19	0	0
EbN0_4	C/I_3	Phi_3	0.015	0.64	0.345	0	0
EbN0_4	C/I_4	Phi_1	0.185	0.81	0.005	0	0
EbN0_4	C/I_4	Phi_2	0.13	0.825	0.045	0	0
EbN0_4	C/I_4	Phi_3	0.05	0.805	0.145	0	0
EbN0_4	C/I_5	Phi_1	0.295	0.69	0.015	0	0
EbN0_4	C/I_5	Phi_2	0.125	0.84	0.035	0	0
EbN0_4	C/I_5	Phi_3	0.07	0.815	0.115	0	0
EbN0_4	C/I_6	Phi_1	0.235	0.76	0.005	0	0
EbN0_4	C/I_6	Phi_2	0.155	0.83	0.015	0	0
EbN0_4	C/I_6	Phi_3	0.055	0.83	0.115	0	0
EbN0_5	C/I_1	Phi_1	0	0	0	0	1

Table A.1.2: Conditional probability distribution (CPD) for BER in DQPSK. (Cont.)

EbN0_5	C/I_1	Phi_2	0	0	0	0	1
EbN0_5	C/I_1	Phi_3	0	0	0	0	1
EbN0_5	C/I_2	Phi_1	0.015	0.23	0.14	0.26	0.355
EbN0_5	C/I_2	Phi_2	0.01	0.175	0.13	0.215	0.47
EbN0_5	C/I_2	Phi_3	0.005	0.145	0.15	0.26	0.44
EbN0_5	C/I_3	Phi_1	0.805	0.195	0	0	0
EbN0_5	C/I_3	Phi_2	0.63	0.37	0	0	0
EbN0_5	C/I_3	Phi_3	0.47	0.53	0	0	0
EbN0_5	C/I_4	Phi_1	0.94	0.06	0	0	0
EbN0_5	C/I_4	Phi_2	0.915	0.085	0	0	0
EbN0_5	C/I_4	Phi_3	0.755	0.245	0	0	0
EbN0_5	C/I_5	Phi_1	0.93	0.07	0	0	0
EbN0_5	C/I_5	Phi_2	0.875	0.125	0	0	0
EbN0_5	C/I_5	Phi_3	0.745	0.255	0	0	0
EbN0_5	C/I_6	Phi_1	0.945	0.055	0	0	0
EbN0_5	C/I_6	Phi_2	0.885	0.115	0	0	0
EbN0_5	C/I_6	Phi_3	0.735	0.265	0	0	0
EbN0_6	C/I_1	Phi_1	0	0	0	0	1
EbN0_6	C/I_1	Phi_2	0	0	0	0	1
EbN0_6	C/I_1	Phi_3	0	0	0	0	1
EbN0_6	C/I_2	Phi_1	0.375	0.03	0.015	0.135	0.445
EbN0_6	C/I_2	Phi_2	0.355	0.06	0.07	0.12	0.395
EbN0_6	C/I_2	Phi_3	0.25	0.07	0.1	0.165	0.415
EbN0_6	C/I_3	Phi_1	1	0	0	0	0
EbN0_6	C/I_3	Phi_2	1	0	0	0	0
EbN0_6	C/I_3	Phi_3	0.995	0.005	0	0	0
EbN0_6	C/I_4	Phi_1	1	0	0	0	0
EbN0_6	C/I_4	Phi_2	1	0	0	0	0
EbN0_6	C/I_4	Phi_3	1	0	0	0	0
EbN0_6	C/I_5	Phi_1	1	0	0	0	0
EbN0_6	C/I_5	Phi_2	1	0	0	0	0
EbN0_6	C/I_5	Phi_3	0.995	0.005	0	0	0
EbN0_6	C/I_6	Phi_1	1	0	0	0	0
EbN0_6	C/I_6	Phi_2	1	0	0	0	0
EbN0_6	C/I_6	Phi_3	1	0	0	0	0

Table A.1.3: Conditional probability distribution (CPD) for BER in D8PSK.

States of the parents			Probability Distribution of BER				
EbN0_1	C/I_1	Phi_1	0	0	0	0	1
EbN0_1	C/I_1	Phi_2	0	0	0	0	1
EbN0_1	C/I_1	Phi_3	0	0	0	0	1
EbN0_1	C/I_2	Phi_1	0	0	0	0	1
EbN0_1	C/I_2	Phi_2	0	0	0	0	1
EbN0_1	C/I_2	Phi_3	0	0	0	0	1
EbN0_1	C/I_3	Phi_1	0	0	0	0	1
EbN0_1	C/I_3	Phi_2	0	0	0	0	1
EbN0_1	C/I_3	Phi_3	0	0	0	0	1
EbN0_1	C/I_4	Phi_1	0	0	0	0	1
EbN0_1	C/I_4	Phi_2	0	0	0	0	1
EbN0_1	C/I_4	Phi_3	0	0	0	0	1
EbN0_1	C/I_5	Phi_1	0	0	0	0	1
EbN0_1	C/I_5	Phi_2	0	0	0	0	1
EbN0_1	C/I_5	Phi_3	0	0	0	0	1
EbN0_1	C/I_6	Phi_1	0	0	0	0	1
EbN0_1	C/I_6	Phi_2	0	0	0	0	1
EbN0_1	C/I_6	Phi_3	0	0	0	0	1
EbN0_2	C/I_1	Phi_1	0	0	0	0	1
EbN0_2	C/I_1	Phi_2	0	0	0	0	1
EbN0_2	C/I_1	Phi_3	0	0	0	0	1
EbN0_2	C/I_2	Phi_1	0	0	0	0	1
EbN0_2	C/I_2	Phi_2	0	0	0	0	1
EbN0_2	C/I_2	Phi_3	0	0	0	0	1
EbN0_2	C/I_3	Phi_1	0	0	0	0.085	0.915
EbN0_2	C/I_3	Phi_2	0	0	0	0.08	0.92
EbN0_2	C/I_3	Phi_3	0	0	0	0.065	0.935
EbN0_2	C/I_4	Phi_1	0	0	0	0.085	0.915
EbN0_2	C/I_4	Phi_2	0	0	0	0.135	0.865
EbN0_2	C/I_4	Phi_3	0	0	0	0.115	0.885
EbN0_2	C/I_5	Phi_1	0	0	0	0.12	0.88
EbN0_2	C/I_5	Phi_2	0	0	0	0.135	0.865
EbN0_2	C/I_5	Phi_3	0	0	0	0.105	0.895
EbN0_2	C/I_6	Phi_1	0	0	0	0.105	0.895
EbN0_2	C/I_6	Phi_2	0	0	0	0.145	0.855
EbN0_2	C/I_6	Phi_3	0	0	0	0.095	0.905

Table A.1.3: Conditional probability distribution (CPD) for BER in D8PSK.

EbN0_3	C/I_1	Phi_1	0	0	0	0	1
EbN0_3	C/I_1	Phi_2	0	0	0	0	1
EbN0_3	C/I_1	Phi_3	0	0	0	0	1
EbN0_3	C/I_2	Phi_1	0	0	0	0.2	0.8
EbN0_3	C/I_2	Phi_2	0	0	0	0.135	0.865
EbN0_3	C/I_2	Phi_3	0	0	0	0.175	0.825
EbN0_3	C/I_3	Phi_1	0	0	0	1	0
EbN0_3	C/I_3	Phi_2	0	0	0	1	0
EbN0_3	C/I_3	Phi_3	0	0	0	0.99	0.01
EbN0_3	C/I_4	Phi_1	0	0	0	1	0
EbN0_3	C/I_4	Phi_2	0	0	0	1	0
EbN0_3	C/I_4	Phi_3	0	0	0	1	0
EbN0_3	C/I_5	Phi_1	0	0	0	1	0
EbN0_3	C/I_5	Phi_2	0	0	0	1	0
EbN0_3	C/I_5	Phi_3	0	0	0	0.995	0.005
EbN0_3	C/I_6	Phi_1	0	0	0	1	0
EbN0_3	C/I_6	Phi_2	0	0	0	0.995	0.005
EbN0_3	C/I_6	Phi_3	0	0	0	1	0
EbN0_4	C/I_1	Phi_1	0	0	0	0	1
EbN0_4	C/I_1	Phi_2	0	0	0	0	1
EbN0_4	C/I_1	Phi_3	0	0	0	0	1
EbN0_4	C/I_2	Phi_1	0	0	0	0.315	0.685
EbN0_4	C/I_2	Phi_2	0	0	0	0.38	0.62
EbN0_4	C/I_2	Phi_3	0	0	0	0.305	0.695
EbN0_4	C/I_3	Phi_1	0	0	0.06	0.94	0
EbN0_4	C/I_3	Phi_2	0	0	0.015	0.985	0
EbN0_4	C/I_3	Phi_3	0	0	0	1	0
EbN0_4	C/I_4	Phi_1	0	0	0.27	0.73	0
EbN0_4	C/I_4	Phi_2	0	0	0.085	0.915	0
EbN0_4	C/I_4	Phi_3	0	0	0	1	0
EbN0_4	C/I_5	Phi_1	0	0	0.31	0.69	0
EbN0_4	C/I_5	Phi_2	0	0	0.145	0.855	0
EbN0_4	C/I_5	Phi_3	0	0	0.015	0.985	0
EbN0_4	C/I_6	Phi_1	0	0	0.25	0.75	0
EbN0_4	C/I_6	Phi_2	0	0	0.12	0.88	0
EbN0_4	C/I_6	Phi_3	0	0	0	1	0
EbN0_5	C/I_1	Phi_1	0	0	0	0	1

Table A.1.3: Conditional probability distribution (CPD) for BER in D8PSK

EbN0_5	C/I_1	Phi_2	0	0	0	0	1
EbN0_5	C/I_1	Phi_3	0	0	0	0	1
EbN0_5	C/I_2	Phi_1	0	0	0.005	0.385	0.61
EbN0_5	C/I_2	Phi_2	0	0	0	0.4	0.6
EbN0_5	C/I_2	Phi_3	0	0	0	0.355	0.645
EbN0_5	C/I_3	Phi_1	0	0.03	0.74	0.23	0
EbN0_5	C/I_3	Phi_2	0	0	0.615	0.385	0
EbN0_5	C/I_3	Phi_3	0	0	0.345	0.655	0
EbN0_5	C/I_4	Phi_1	0	0.26	0.74	0	0
EbN0_5	C/I_4	Phi_2	0	0.05	0.945	0.005	0
EbN0_5	C/I_4	Phi_3	0	0	0.79	0.21	0
EbN0_5	C/I_5	Phi_1	0	0.375	0.625	0	0
EbN0_5	C/I_5	Phi_2	0	0.08	0.92	0	0
EbN0_5	C/I_5	Phi_3	0	0	0.82	0.18	0
EbN0_5	C/I_6	Phi_1	0	0.37	0.63	0	0
EbN0_5	C/I_6	Phi_2	0	0.045	0.945	0.01	0
EbN0_5	C/I_6	Phi_3	0	0	0.855	0.145	0
EbN0_6	C/I_1	Phi_1	0	0	0	0	1
EbN0_6	C/I_1	Phi_2	0	0	0	0	1
EbN0_6	C/I_1	Phi_3	0	0	0	0	1
EbN0_6	C/I_2	Phi_1	0.025	0.015	0.035	0.25	0.675
EbN0_6	C/I_2	Phi_2	0	0.01	0.07	0.235	0.685
EbN0_6	C/I_2	Phi_3	0	0	0.025	0.355	0.62
EbN0_6	C/I_3	Phi_1	0.96	0.035	0.005	0	0
EbN0_6	C/I_3	Phi_2	0.725	0.225	0.05	0	0
EbN0_6	C/I_3	Phi_3	0.36	0.34	0.295	0.005	0
EbN0_6	C/I_4	Phi_1	0.96	0.04	0	0	0
EbN0_6	C/I_4	Phi_2	0.945	0.055	0	0	0
EbN0_6	C/I_4	Phi_3	0.935	0.06	0.005	0	0
EbN0_6	C/I_5	Phi_1	0.985	0.015	0	0	0
EbN0_6	C/I_5	Phi_2	0.96	0.04	0	0	0
EbN0_6	C/I_5	Phi_3	0.915	0.06	0.025	0	0
EbN0_6	C/I_6	Phi_1	0.99	0.01	0	0	0
EbN0_6	C/I_6	Phi_2	0.945	0.05	0.005	0	0
EbN0_6	C/I_6	Phi_3	0.955	0.045	0	0	0

APPENDIX 2

PYTHON CODE

Code for simulating DBPSK, DQPSK, and D8PSK

This code implements the function BER. This function takes as arguments EbN0,C_I,Doppler,b, MaxIter. EbN0 is the ratio between the energy of bit and the spectral noise density in dB. C_I is the ratio between the carrier and the co-channel interference in dB. Doppler represents the Doppler phase shift in radians. The argument b is the number of bits per symbol. MaxIter is the number of symbols sent for each combination of EbN0,C_I,Doppler, and b. After completing a number of iterations equal to MaxIter the function returns the bit error rate (BER) for the symbols transmitted, MaxIter. This is the function used for eliciting the conditional probability distribution (CPD) of the variable BER, which has as parents the variables EbN0,C_I,Doppler. The part of the code that sends symbols to estimate BER was modified from a simulation program in the book [78]. The code in this book was written in matlab and only considers noise as a factor affecting the BER. I have written the code in python and modified to include co-channel interference and Doppler phase shift.

```
from __future__ import division
import math
import cmath
import numpy as np
from pylab import *
RAND_SEED = 42
pi=math.pi
```

```
def CI_lin(C_I,Es,Ts):
    return sqrt(2*Es/Ts)*10**(-1*(C_I/20))
def BER(EbN0,C_I,Doppler,b, MaxIter):

    M=2**b  #b is the number of bits per symbol
    Tb=1; Ts= b*Tb
    Nb=32; Ns= b*Nb
    nd=1
    T=Ts/Ns; LB=4*Ns ; LBN1= LB-Ns; scope=range(LBN1,LB) ;scope0=range(LBN1-nd,LB-nd)
    Es=b;sqEs=sqrt(Es)
    ss_1=np.array([[0],[1]]) #1 bit per symbol
    ss_2=np.array([[0,0],[0,1],[1,1],[1,0]]) # 2 bits per symbol
    ss_3=np.array([[0,0,0],[0,0,1],[1,0,1],[1,0,0],[1,1,0],[1,1,1],[0,1,1],[0,1,0]]) #3 bits per symbol
    ss= [[], ss_1, ss_2, ss_3]
    phases= (2*pi/M)*np.arange(M) #Generate phases accoding to constellation
    exp_symbols= np.exp(phases*1j) #Represent symbols as exponential numbers
    wc=8*pi/Ts; t=np.arange(Ns)*T; wcT=wc*T
    wct=wcT*np.arange(Ns)
    symbols=np.zeros([M,len(t)])
    su=np.zeros([2,len(t)])
    for i in np.arange(M):symbols[i,:]=sqrt(2*Es/Ts)*np.cos(wc*t+phases[i])
    su[0,:]=np.cos(wc*t);su[1,:]=-np.sin(wc*t);su=su*sqrt(2/Ts);suT=su*T  #Symbols stored in the receiver for comparison
    SN_EB=round(10*np.log10(b),1)  #This term is to convert EbN0 to SNR in dB according to b, bits per symbol.
    SNRdB=EbN0  # convert to SNR in dB
    SNR=10**(SNRdB/10) # convert SBRdB to linear
    N0=b*(Es/b)/SNR; sigma2=N0/2; sgmsT=sqrt(sigma2/T) #sgmsT is to scale the noise according to Eb/N0
    sws=np.zeros(LB); yr=np.zeros([2,LB])
    nobe=0
    is0=0  #Initial Signal
    th0=0  #Initial guess of signal phase
    for k in range(1,MaxIter+1):
        i=np.random.randint(M);s=ss[b][i,:]
        is_k=(is0+i)%M; is1=is_k
        tx_signal=exp_symbols[is1]#random symbol as exponential number
```

```
        FD= np.random.uniform(-Doppler,Doppler) # Generate the random Doppler phase shift
        channel=np.exp(FD*1j) #Doppler effect represented exponential number
        rx_symbol=tx_signal*channel # Alter the tx_signal according to Doppler phase shift
        rx_phase=cmath.phase(rx_symbol) #Phase of the symbol at the receiver
        sws=np.concatenate((sws[Ns:],sqrt(2*Es/Ts)*np.cos(wc*t+rx_phase)),axis=1)  #Represent the symbol as sinusoidal signal
#The next line represents the noise as a sinusoidal signal
        bp_noise=  np.cos(wct)*np.random.normal(0,1,Ns)+ np.sin(wct)*np.random.normal(0,1,Ns)
        bp_noise = sgmsT* bp_noise  #Scaling the noise according to EbN0
 #The next line converts C/I to linear and multiplies the sinusoidal representing the  co-channel interference, so that it matches C/I
        Co_Channel= CI_lin(C_I,Es,Ts)* sqrt(2*Es/Ts)*np.cos(wc*t+np.random.uniform(0,2*pi))
        rn=sws[scope0]+ bp_noise + Co_Channel
        yr=np.concatenate((yr[:,Ns:],suT*rn),axis=1)
        ycsk= np.sum(yr[:,scope],axis=1)
        th=np.arctan2(ycsk[1],ycsk[0]);dth=th-th0
        if dth<-pi/M: dth=dth+2*pi
        mmin=np.argmin(abs(dth-phases))
        d=ss[b][mmin,:]
        nobe=nobe+sum(s!=d)
        is0=is_k; th0=th
        if nobe >100: break
    return (nobe/(k*b))
```

Code for generating data to elicit the conditional probability distribution (CPD) of the variable BER

This piece of code calls the function BER for different combinations of the states of the parents of the variable BER to generate data that will be used for eliciting the CPD of BER.

```
from __future__ import division
import math
import cmath
#import random
import numpy as np
from pylab import *
import itertools
```

```
from BER import BER
#bins= list with the intervals for each state
#samples= Number of samples per interval
#function= It is the function that tells the relation among the variables.
#datafile= name of the file where we want to save the output
def discretizer(in_data,bins,states,variables,out_data):
#This function discretizes the data generated by the function cpddata.
#It receives: bins(Definition of the intervals for the variables), samples(number of samples we want to generate for interval ),
#function(the relation among the parent variables to generate the children variable), out_file(the file where we want to save the result.
    mydata=np.load(in_data)
   discrete=np.zeros(len(mydata)).reshape(len(mydata),1)
   newbins=[]
   for i in range(len(bins)):
      newbins.append(sorted(list(set(list(itertools.chain(*bins[i]))))))
      bins=newbins
   for i in range(len(bins)):
            inds=np.digitize(mydata[:,i],bins[i])
            inds=inds-1
            inds=list(inds)
            discretized=states[i][inds]
            discretized2=discretized.reshape(len(mydata),1)
            discrete=np.hstack((discrete,discretized2))
   discrete=discrete[:,range(1,len(bins)+1)]
   dlist=[]
   for i in range(len(discrete)):
      sample=zip(variables,discrete[i,:])
      sample.reverse()
      sample=dict(sample)
      dlist.append(sample)
   np.save(out_data,dlist)
   return
def cpddata(bins,samples,function,out_file):
#This function creates the data for getting the CPD of the variables
#It receives: bins(Definition of the intervals for the variables), samples(number of samples we want to generate for interval ),
```

```
#function(the relation among the parent variables to generate the children variable), out_file(the file where we want to save the result.
    bins.pop()
    data=[]
    arg=[]
    combined= list(itertools.product(*bins))
    combined=combined*samples
    for i in range(len(combined)):
        for j in range(len(combined[i])):
            argnew= np.random.uniform(combined[i][j][0],combined[i][j][1])
            arg=np.hstack((arg,argnew))
    rows=int(len(arg)/len(combined[0]))
    col= len(combined[0])
    arg=np.reshape(arg,(rows,col))
    for i in range(rows):
        data_new=np.hstack((arg[i],function(arg[i])))
        data=np.hstack((data,data_new))
    data=np.reshape(data,(rows,col+1))
    np.save(out_file,data)
    return
#########################################################################################################
# This section creates the data for generating the CPDs of BER: Bit Error Rate
#########################################################################################################
#Parents: EbN0,C_I, and Dop_Phi
#We import the function BER(EbN0,C_I,Doppler,b, MaxIter), which is defined in a
#separeted file, since it is long.
#The next functions define the relation between parents and the child variable.
#These functions receive the arguments in the array x. x[0]=EbNO, X[1]=C_I,
# and X[2]=Dop_Phi
def BER_2d(x):                  #This function if for 2DPSK modulation
    b=1    #Bits per symbol
    MaxIter=10000
    ber=BER(x[0],x[1],x[2],b,MaxIter)
    return ber
def BER_4d(x):                  #This function if for 4DPSK modulation
```

```
    b=2   #Bits per symbol
    MaxIter=10000
    ber=BER(x[0],x[1],x[2],b,MaxIter)
    return ber
def BER_8d(x):                          #This function if for 8DPSK modulation
    b=3    #Bits per symbol
    MaxIter=10000
    ber=BER(x[0],x[1],x[2],b,MaxIter)
    return ber
binsEbN0= [[0.0, 3.0],[3.0, 6.0],[6.0, 10.0],[10.0, 14.0],[14.0,1000.0]]
binsC_I= [[0.0, 3.0],[3.0,6.0],[6.0, 12.0],[12.0 , 20.0],[20.0, 1000.0]]
binsPhi= [[0.001,0.08],[0.08,0.136]] #These are given in radians
binsBER= [[0.0, 1e-5],[1e-5, 1e-3],[1e-3, 1e-1],[1e-1, 1.0 ]]
bins=[binsEbN0,binsC_I,binsPhi,binsBER]
states_EBN0=np.array(["EbN0_1","EbN0_2","EbN0_3","EbN0_4","EbN0_5"])
states_C_I=np.array(["C/I_1","C/I_2","C/I_3","C/I_4","C/I_5"])
states_Phi=np.array(["Phi_1","Phi_2"])
states_BER=np.array(["BER_1","BER_2","BER_3","BER_4"])
states=[states_EBN0,states_C_I,states_Phi,states_BER]
var=["EbN0","C/I","Dop_Phi","BER"]
samples=1
#This section is for 2dpsk modulation
#################################################################################################
BER2_data= 'BER2_data.npy'
cpddata(bins,samples,BER_2d,BER2_data) #cdpdata(bins,samples,function,out_data)
bins=[binsEbN0,binsC_I,binsPhi,binsBER] #Redefinitions of bins for the next function
discretizer(BER2_data,bins,states,var,BER2_data) #discretizer(in_data,bins,states,variables,out_data)
#This section is for 4dpsk modulation
#################################################################################################
BER4_data= 'BER4_data.npy'
cpddata(bins,samples,BER_4d,BER4_data) #cdpdata(bins,samples,function,out_data)
bins=[binsEbN0,binsC_I,binsPhi,binsBER] #Redefinitions of bins for the next function
discretizer(BER4_data,bins,states,var,BER4_data) #discretizer(in_data,bins,states,variables,out_data)
#This section is for 8dpsk modulation
```

```
#############################################################################################

BER8_data= 'BER8_data.npy'

cpddata(bins,samples,BER_8d,BER8_data) #cdpdata(bins,samples,function,out_data)

bins=[binsEbN0,binsC_I,binsPhi,binsBER] #Redefinitions of bins for the next function

discretizer(BER8_data,bins,states,var,BER8_data) #discretizer(in_data,bins,states,variables,out_data)
```

REFERENCES

[1] A. F. Molisch. *Wireless Communications.* 2010.

[2] A. Goldsmith. *Wireless Communications*. 2005.

[3] L. Doyle. *Essentials of Cognitive Radio*. 2009.

[4] M. Cave, C. Doyle and W. Webb. *Essentials of Modern Spectrum Management.* 2007.

[5] S. Haykin. Cognitive radio: Brain-empowered wireless communications. *IEEE J. Select. Areas Commun. 23(2),* pp. 201-220. 2005.

[6] J. Mitola III and G. Q. Maguire Jr. Cognitive radio: Making software radios more personal. *Personal Communications, IEEE 6(4),* pp. 13-18. 1999.

[7] T. Yucek and H. Arslan. A survey of spectrum sensing algorithms for cognitive radio applications. *Communications Surveys & Tutorials, IEEE 11(1),* pp. 116-130. 2009.

[8] P. Demestichas, A. Katidiotis, K. A. Tsagkaris, E. F. Adamopoulou and K. P. Demestichas. Enhancing channel estimation in cognitive radio systems by means of bayesian networks. *Wireless Personal Communications 49(1),* pp. 87-105. 2009.

[9] A. P. Sage and J. L. Melsa. *Estimation Theory with Applications to Communications and Control* 1971.

[10] H. V. Poor. *An introduction to signal detection and estimation*. *New York, Springer-Verlag,* 1988.

[11] H. Urkowitz. Energy detection of unknown deterministic signals. *Proc IEEE 55(4),* pp. 523-531. 1967.

[12] Y. Zeng, Y. C. Liang, A. T. Hoang and R. Zhang. A review on spectrum sensing for cognitive radio: Challenges and solutions. *EURASIP Journal on Advances in Signal Processing* 2010.

[13] W. A. Gardner. Exploitation of spectral redundancy in cyclostationary signals. *Signal Processing Magazine, IEEE 8(2),* pp. 14-36. 1991.

[14] W. Gardner, W. Brown and C. K. Chen. Spectral correlation of modulated signals: Part II--digital modulation. *Communications, IEEE Transactions On 35(6),* pp. 595-601. 1987.

[15] W. Gardner. Spectral correlation of modulated signals: Part I--analog modulation. *Communications, IEEE Transactions On 35(6),* pp. 584-594. 1987.

[16] R. Tandra and A. Sahai. Fundamental limits on detection in low SNR under noise uncertainty. Presented at Wireless Networks, Communications and Mobile Computing, 2005 International Conference On. 2005.

[17] Y. Zeng and Y. Liang. Spectrum-sensing algorithms for cognitive radio based on statistical covariances. *Vehicular Technology, IEEE Transactions On 58(4),* pp. 1804-1815. 2009.

[18] Y. Zeng and Y. Liang. Eigenvalue-based spectrum sensing algorithms for cognitive radio. *Communications, IEEE Transactions On 57(6),* pp. 1784-1793. 2009.

[19] A. Mate, K. Lee and I. Lu. Spectrum sensing based on time covariance matrix using GNU radio and USRP for cognitive radio. Presented at Systems, Applications and Technology Conference (LISAT), 2011 IEEE Long Island. 2011.

[20] P. Bello. Characterization of randomly time-variant linear channels. *Communications Systems, IEEE Transactions On 11(4),* pp. 360-393. 1963.

[21] G. Matz and F. Hlawatsch. Fundamentals of time-varying communication channels. *Wireless Communications Over Rapidly Time-Varying Channels, F.Hlawatsch and G.Matz, Eds.Academic Press* pp. 1-63. 2011.

[22] L. Bernadó, T. Zemen, F. Tufvesson, A. F. Molisch and C. F. Mecklenbrauker. The (in-) validity of the WSSUS assumption in vehicular radio channels. Presented at Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium On. 2012.

[23] L. Bernadó, T. Zemen, F. Tufvesson, A. F. Molisch and C. F. Mecklenbräuker. Delay and doppler spreads of non-stationary vehicular channels for safety relevant scenarios. *arXiv Preprint arXiv:1305.3376* 2013.

[24] G. Matz. On non-WSSUS wireless fading channels. *Wireless Communications, IEEE Transactions On 4(5),* pp. 2465-2478. 2005.

[25] E. G. Larsson, P. Stoica and J. Li. On maximum-likelihood detection and decoding for space-time coding systems. *Signal Processing, IEEE Transactions On 50(4),* pp. 937-944. 2002.

[26] E. G. Larsson, P. Stoica and J. Li. Orthogonal space-time block codes: Maximum likelihood detection for unknown channels and unstructured interferences. *Signal Processing, IEEE Transactions On 51(2),* pp. 362-372. 2003.

[27] V. Choqueuse, A. Mansour, G. Burel, L. Collin and K. Yao. Blind channel estimation for STBC systems using higher-order statistics. *Wireless Communications, IEEE Transactions On 10(2),* pp. 495-505. 2011.

[28] W. Ma, B. Vo, T. N. Davidson and P. Ching. Blind ML detection of orthogonal space-time block codes: Efficient high-performance implementations. *Signal Processing, IEEE Transactions On 54(2),* pp. 738-751. 2006.

[29] A. Gallo, E. Chiavaccini, F. Muratori and G. M. Vitetta. BEM-based SISO detection of orthogonal space-time block codes over frequency flat-fading channels. *Wireless Communications, IEEE Transactions On 3(6),* pp. 1885-1889. 2004.

[30] Y. Li, C. N. Georghiades and G. Huang. Iterative maximum-likelihood sequence estimation for space-time coded systems. *Communications, IEEE Transactions On 49(6),* pp. 948-951. 2001.

[31] N. Ammar and Z. Ding. Blind channel identifiability for generic linear space-time block codes. *Signal Processing, IEEE Transactions On 55(1),* pp. 202-217. 2007.

[32] J. Vía, I. Santamaría and J. Pérez. Code combination for blind channel estimation in general MIMO-STBC systems. *EURASIP Journal on Advances in Signal Processing 2009*pp. 3. 2009.

[33] G. Leus, Z. Tang and P. Banelli. Estimation of time-varying Channels–A block approach. 2011.

[34] N. Geng, X. Yuan and L. Ping. Dual-diagonal LMMSE channel estimation for OFDM systems. *Signal Processing, IEEE Transactions On 60(9),* pp. 4734-4746. 2012.

[35] K. Hung and D. W. Lin. Pilot-based LMMSE channel estimation for OFDM systems with Power–Delay profile approximation. *Vehicular Technology, IEEE Transactions On 59(1),* pp. 150-159. 2010.

[36] A. F. Molisch, L. J. Greenstein and M. Shafi. Propagation issues for cognitive radio. *Proc IEEE 97(5),* pp. 787-804. 2009.

[37] F. P. Fontán and P. M. Espieira. *Modelling the Wireless Propagation Channel: A Simulation Approach with MATLAB* 2008.

[38] R. Narasimhan and D. C. Cox. A generalized doppler power spectrum for wireless environments. *Communications Letters, IEEE 3(6),* pp. 164-165. 1999.

[39] S. Saunders and A. Aragón-Zavala. *Antennas and Propagation for Wireless Communication Systems* 2007.

[40] T. S. Rappaport. *Wireless Communications: Principles and Practice.* 2004.

[41] D. W. Matolak. Air-ground channels & models: Comprehensive review and considerations for unmanned aircraft systems. Presented at Aerospace Conference, 2012 IEEE. 2012.

[42] A. F. Molisch and F. Tufvesson. *Multipath Propagation Models for Broadband Wireless Systems.* 2004.

[43] D. Maas, M. H. Firooz, J. Zhang, N. Patwari and S. K. Kasera. Channel sounding for the masses: Low complexity GNU 802.11 b channel impulse response estimation. *Wireless Communications, IEEE Transactions On 11(1),* pp. 1-8. 2012.

[44] C. Tepedelenlioğlu, A. Abdi, G. B. Giannakis and M. Kaveh. Estimation of doppler spread and signal strength in mobile communications with applications to handoff and adaptive transmission. *Wireless Communications and Mobile Computing 1(2),* pp. 221-242. 2001.

[45] S. M. Kay. Fundamentals of statistical signal processing, volume I: Estimation theory (v. 1). 1993.

[46] T. W. Rondeau and C. W. Bostian. *Artificial Intelligence in Wireless Communications.* 2009.

[47] A. He, Kyung Kyoon Bae, T. R. Newman, J. Gaeddert, Kyouwoong Kim, R. Menon, L. Morales-Tirado, J. J. Neel, Youping Zhao, J. H. Reed and W. H. Tranter. A survey of artificial intelligence for cognitive radios. *Vehicular Technology, IEEE Transactions On 59(4),* pp. 1578-1592. 2010.

[48] M. Bkassiny, Y. Li and S. K. Jayaweera. A survey on machine-learning techniques in cognitive radios. *Communications Surveys & Tutorials, IEEE 15(3),* pp. 1136-1159. 2013.

[49] W. Jouini, C. Moy and J. Palicot. Decision making for cognitive radio equipment: Analysis of the first 10years of exploration. *EURASIP Journal on Wireless Communications and Networking 2012(1),* pp. 26. 2012.

[50] A. Fehske, J. Gaeddert and J. Reed. A new approach to signal classification using spectral correlation and neural networks. Presented at New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium On. 2005.

[51] A. F. Cattoni, M. Ottonello, M. Raffetto and C. S. Regazzoni. Neural networks mode classification based on frequency distribution features. Presented at Cognitive Radio Oriented Wireless Networks and Communications, 2007. CrownCom 2007. 2nd International Conference On. 2007.

[52] Y. Zhao, J. Gaeddert, L. Morales, K. Bae, J. S. Um and J. H. Reed. Development of radio environment map enabled case-and knowledge-based learning algorithms for IEEE 802.22 WRAN cognitive engines. Presented at Cognitive Radio Oriented Wireless Networks and Communications, 2007. CrownCom 2007. 2nd International Conference On. 2007.

[53] Z. Zhang and X. Xie. Intelligent cognitive radio: Research on learning and evaluation of CR based on neural network. Presented at Information and Communications Technology, 2007. ICICT 2007. ITI 5th International Conference On. 2007.

[54] I. A. Akbar and W. H. Tranter. Dynamic spectrum allocation in cognitive radio using hidden markov models: Poisson distributed case. Presented at SoutheastCon, 2007. Proceedings. IEEE. 2007.

[55] C. H. Park, S. W. Kim, S. M. Lim and M. S. Song. HMM based channel status predictor for cognitive radio. Presented at Microwave Conference, 2007. APMC 2007. Asia-Pacific. 2007.

[56] T. R. Newman, B. A. Barker, A. M. Wyglinski, A. Agah, J. B. Evans and G. J. Minden. Cognitive engine implementation for wireless multicarrier transceivers. *Wireless Communications and Mobile Computing 7(9),* pp. 1129-1142. 2007.

[57] L. Yong, J. Hong and H. Y. Qing. Design of cognitive radio wiereless parameters based on multi-objective immune genetic algorithm. Presented at Communications and Mobile Computing, 2009. CMC'09. WRI International Conference On. 2009.

[58] Y. Tang, Q. Zhang and W. Lin. Artificial neural network based spectrum sensing method for cognitive radio. Presented at Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference On. 2010.

[59] A. Balieiro, P. Yoshioka, K. Dias, D. Cavalcanti and C. Cordeiro. A multi-objective genetic optimization for spectrum sensing in cognitive radio. *Expert Syst. Appl. 41(8),* pp. 3640-3650. 2014.

[60] P. Huang, F. Lin and J. Zhou. ACO-based routing algorithm for cognitive radio networks. *Mathematical Problems in Engineering.* 2013.

[61] K. W. Choi and E. Hossain. Estimation of primary user parameters in cognitive radio systems via hidden markov model. *Signal Processing, IEEE Transactions On 61(3),* pp. 782-795. 2013.

[62] X. Xing, T. Jing, W. Cheng, Y. Huo and X. Cheng. Spectrum prediction in cognitive radio networks. *Wireless Communications, IEEE 20(2)*, pp. 90-96. 2013.

[63] Z. Chen, N. Guo, Z. Hu and R. C. Qiu. Channel state prediction in cognitive radio, part II: Single-user prediction. Presented at Proceedings of IEEE SoutheastCon. 2011.

[64] C. Pham, N. H. Tran, C. T. Do, S. I. Moon and C. S. Hong. Spectrum handoff model based on hidden markov model in cognitive radio networks. Presented at Information Networking (ICOIN), 2014 International Conference On. 2014.

[65] M. T. Soleimani, M. Kahvand and R. Sarikhani. Handoff reduction based on prediction approach in cognitive radio networks. Presented at Communication Technology (ICCT), 2013 15th IEEE International Conference On. 2013.

[66] J. V. Stone. *Bayes' Rule: A Tutorial Introduction to Bayesian Analysis*. 2013.

[67] S. B. McGrayne. *The Theory that would Not Die: How Bayes' Rule Cracked the Enigma Code, Hunted Down Russian Submarines, & Emerged Triumphant from Two Centuries of Controversy*. 2011.

[68] D. Barber. *Bayesian Reasoning and Machine Learning*. 2012.

[69] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. 2009.

[70] E. Charniak. Bayesian networks without tears. *AI Magazine 12(4)*, pp. 50. 1991.

[71] M. Neil, N. Fenton and L. Nielson. Building large-scale bayesian networks. *The Knowledge Engineering Review 15(03)*, pp. 257-284. 2000.

[72] U. B. Kjræulff and A. L. Madsen. *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis: A Guide to Construction and Analysis*. 2012.

[73] N. Fenton and M. Neil. Making decisions: Using bayesian nets and MCDA. *Knowledge-Based Syst. 14(7)*, pp. 307-325. 2001.

[74] A. Darwiche. Bayesian networks. *Commun ACM 53(12)*, pp. 80-90. 2010.

[75] D. Roddy. Satellite communications. 2001.

[76] L. Allulli, G. F. Italiano and F. Santaroni. "Exploiting GPS data in public transport journey planners," in *Experimental Algorithms*Anonymous. 2014.

[77] S. Srivastava, S. Ahuja and A. Mittal. Determining most visited locations based on temporal grouping of GPS data. Presented at Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20-22, 2011.

[78] W. Yang, Y. S. Cho, W. G. Jeon, J. W. Lee, J. H. Paik, J. K. Kim, M. Lee, K. I. Lee, K. W. Park and K. S. Woo. *Matlab/Simulink for Digital Communication.* 2009.

[79] R. K. Sharma and J. W. Wallace. Improved autocorrelation-based sensing using correlation distribution information. Presented at Smart Antennas (WSA), 2010 International ITG Workshop On. 2010.

[80] H. Stark. *Probability and Random Processes with Application to Signal Processing, 3/E* 2002.

[81] R. J. Pirkl and G. D. Durgin. Optimal sliding correlator channel sounder design. *Wireless Communications, IEEE Transactions On 7(9),* pp. 3488-3497. 2008.

[82] M. Goresky and A. M. Klapper. Fibonacci and galois representations of feedback-with-carry shift registers. *Information Theory, IEEE Transactions On 48(11),* pp. 2826-2836. 2002.

[83] S. Salous. *Radio Propagation Measurement and Channel Modelling.* 2013.