PhD-FSTC-2017-42
The Faculty of Sciences, Technology and Communication

DISSERTATION

Presented on 18/07/2017 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG
EN INFORMATIQUE

by

Anestis TSAKMALIS
Born on 16 October 1984 in Thessaloniki, Greece

ACTIVE LEARNING IN COGNITIVE RADIO NETWORKS

Dissertation defense committee

Dr. Symeon Chatzinotas, dissertation supervisor
Deputy Head of the SIGCOM research group in SnT, University of Luxembourg

Dr. Björn Ottersten
Professor and Director of SnT, University of Luxembourg

Dr. Ana Isabel Pérez-Neira, Vice-chairman
Professor, Technical University of Catalonia

Dr. Antonio G. Marques
Associate Professor, King Juan Carlos University

Dr. Radu State, Chairman
Head of the SEDAN research group in SnT, University of Luxembourg

Abstract

In this thesis, numerous Machine Learning (ML) applications for Cognitive Radios Networks (CRNs) are developed and presented which facilitate the efficient spectral coexistence of a legacy system, the Primary Users (PUs), and a CRN, the Secondary Users (SUs). One way to better exploit the capacity of the legacy system frequency band is to consider a coexistence scenario using underlay Cognitive Radio (CR) techniques, where SUs may transmit in the frequency band of the PU system as long as the induced to the PU interference is under a certain limit and thus does not harmfully affect the legacy system operability. This thesis starts with an overview of the ML literature for CRNs in Chapter 2 and continues with the contributions which are divided into the three following chapters.

In Chapter 3, we propose a ML approach for detecting the Modulation and Coding scheme (MCS) of a PU. This Spectrum Sensing (SS) task considers Higher Order Statistical (HOS) features of the sensed PU signal and an efficient ML classifier, the Support Vector Machine (SVM), to identify the modulation scheme used by the PU. The outcome of this identification is combined with the log-likelihood ratios (LLRs) of the PU signal code syndromes to find the PU MCS. This process is called Modulation and Coding Classification (MCC) and it will play an important role in the next part of this thesis.

In Chapter 4, we take advantage of the MCC process in order to bypass the absence of communication between the PU and the SU systems. Due to lack of cooperation between the two systems, the CRN may exploit this multilevel MCC sensing *feedback* as implicit channel state information (CSI) of the PU link in order to constantly observe the impact of the aggregated interference it causes. Changes in the PU MCS because of the CRN induced interference are considered as PU reactions following the PU Adaptive Coding and Modulation (ACM) protocol. In the examined case study, this MCC feedback allows the CRN to sequentially adjust the SU transmit power levels in order to jointly tackle maximizing the CRN throughput, a Power Control (PC) optimization objective, and learning the interference channel gains which basically constitute the PU interference constraint of the aforementioned optimization problem. Ideal candidate learning approaches for this problem setting with high convergence rate are the Cutting Plane Methods (CPMs). The work of this part laid the foundation of the *Active Learning* (AL) thesis perspective enabled by PU pieces of feedback.

In Chapter 5, we aim solely at learning the interference channel gains by sequentially probing the PU system. Here, we no longer consider the MCC feedback but the ACK-/NACK binary packet which is acquired by eavesdropping the reverse PU link and indicates whether the probing-induced interference is harmful or not. This rudimentary

piece of feedback is chosen in order to focus on developing sophisticated probing design techniques for learning the PU interference constraint, since we have already demonstrated the benefits of using the multilevel MCC feedback. We adopt an approach related with AL, a ML field where a learning algorithm sequentially chooses unlabelled data and requests its label in order to reach to a learning solution with as less as possible label queries. This means that the unlabelled data chosen in each step must be selected or designed intelligently so that it delivers more information about the learning solution. This process is clearly correlated with designing the SU transmit power levels, the probing, in order to render it as more informative as possible. Additionally, we incorporate the probability of each binary feedback being correct into this intelligent probing mechanism by developing multivariate Bayesian AL methods inspired by the Probabilistic Bisection Algorithm (PBA) and the CPMs.

In Chapter 6, the AL rationale of the previous chapters is further enhanced by adding a constraint on the number of the harmful probing attempts during this learning process. Specifically, the CRN aims to learn the PU interference constraint with as few as possible probing attempts while limiting under a threshold the number of harmful probing-induced interference events or equivalently of NACK packet observations. No uncertainty of the ACK/NACK binary packet is taken into account in this scenario. A provenly optimal solution for this constrained AL problem is obtained and implemented using a fast and accurate Bayesian Learning method, the Expectation Propagation (EP).

Finally, in Chapter 7 we summarize the conclusions and discuss the promising research applications of AL in the CR framework.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor Dr. Symeon Chatzinotas for advising, supporting and trusting me during the last four years. All the fruitful results of our work exist only because of his advice and patience, even when I repeatedly failed to deliver them on time. I will never forget the research freedom he provided me with and the confidence he placed in me while exploring so many new research areas. Moreover, I would like to thank Prof. Björn Ottersten, a man of unique insight, for his supervision during my PhD research in the Interdisciplinary Centre for Security, Reliability and Trust (SnT). Without his constructive support and inspiring ideas, a lot of perspectives of our work would not have been revealed. I sincerely thank him for giving me the opportunity to work with him in SnT, a place full of chances and opportunities which has flourished under his management and vision. I would also like to thank Prof. Ana Isabel Pérez-Neira for her kindness and insightful advise which were important for advancing my research and exploring new paths in science.

My warmest thanks go to my friends, officemates, colleagues and fellow PhD students who have made the last four years the most enjoyable period of my life. Thank you for reminding me that sometimes you need to forget about work and enjoy yourself. Thank you for always being there and for proving to me that people from all over the world have so many things in common. Your presence in this journey has been a true blessing and you all hold a special place in my heart. But above all, thank you for withstanding the countless, endless, never-ending hours of music coming from my room or my office. To Estelle, I want to express my sincere love for all her kindness and love especially during the last months of my PhD. She has been my main source of motivation and strength even when things appeared impossible to be done.

Finally, I would like to thank my family for everything. I would not be the one I am today without them. The love, the encouragement, the support, the human values I carry with me, my first Math knowledge and so many other wonderful things are all because of them, my parents, Giorgos and Fotini, two remarkable Math teachers, and my brother, Kiriakos. Thank you for always being by my side. My love and gratitude cannot be expressed in words.

Anestis Tsakmalis
Luxembourg, July 2017

Contents

List of Figures

List of Tables

Abbreviations

ACM	**A**daptive **C**oding and **M**odulation
AL	**A**ctive **L**earning
AWGN	**A**dditive **W**hite **G**aussian **N**oise
BF	**B**eam**F**orming
CBS	**C**ognitive **B**ase **S**tation
CPM	**C**utting **P**lane **M**ethod
CR	**C**ognitive **R**adio
CRN	**C**ognitive **R**adio **N**etwork
CSI	**C**hannel **S**tate **I**nformation
DSA	**D**ynamic **S**pectrum **A**ccess
DP	**D**ynamic **P**rogramming
EP	**E**xpectation **P**ropagation
FDMA	**F**requency **D**ivision **M**ultiple **A**ccess
HOS	**H**igher **O**rder **S**tatistics
LDPC	**L**ow **D**ensity **P**arity **C**heck
MCC	**M**odulation and **C**oding **C**lassification
MCMC	**M**arkov **C**hain **M**onte **C**arlo
MCS	**M**odulation and **C**oding **S**cheme
PBA	**P**robabilistic **B**isection **A**lgorithm
PC	**P**ower **C**ontrol
PU	**P**rimary **U**ser
QoS	**Q**uality **o**f **S**ervice
SINR	**S**ignal to **N**oise plus **I**nterference **R**atio
SNR	**S**ignal to **N**oise **R**atio
SS	**S**pectrum **S**ensing

SU	**S**econdary **U**ser
SVM	**S**upport **V**ector **M**achine

To my parents, Giorgos and Fotini, for everything

"Learning never exhausts the mind"
Leonardo da Vinci

Chapter 1

Introduction

1.1 Motivation and Scope

Radio spectrum is well known to be a limited resource. Ever since its first commercial usage, regulations for limiting services to specific frequency bands have been enforced. This rulemaking process assumes that a static assignment of services to frequency bands not only facilitates the financial exploitation of the Radio Spectrum, but also limits interference and supports the construction of cheap and less complicated transceivers, a major technological restraint.

Within the last years though, wireless communications have faced a steadily growing demand of multimedia and other bandwidth consuming interactive services. At the same time, measurements of the spectrum usage have shown that even if some segments are congested because of this service demand burst, many others are being underutilized [1] as they still rely on old communication technologies. This has led us to rethink the inefficient static nature of this architecture which limits accessibility and creates the false perception of spectrum saturation.

Towards this direction, the research community proposed a breakthrough architecture design called Dynamic Spectrum Access (DSA) [2] according to which radio spectrum should be distributed to users and services based not on rigid regulations, but flexible restraints which take into consideration overall spectrum availability, access needs, service priorities, market perspectives, network optimization objectives and QoS requirements. The first step of this evolution is to retain the costly infrastructure and spectrum access protocols of licensed services based on outdated technology, the so called legacy systems, which are not fully utilizing their assigned bands or which are resilient to interference, and allow unlicensed users access the frequency bands assigned to these licensed services.

These unlicensed users must be flexible and intelligent radio devices with DSA abilities which will detect access opportunities in these bands and exploit them to serve their own service demands.

This new kind of radio is the catalytic agent of DSA and it is called in the literature Cognitive Radio (CR) [3, 4]. It was first introduced by J. Mitola who actually borrowed the term "Cognitive" from the Computer Science world in order to signify a radio device able to sense, understand, adapt and interact with its surroundings based on the user's demands and the environment's limitations. In the CR regime, the licensed and the unlicensed users are called respectively Primary Users (PUs) and Secondary Users (SUs) basically in order to distinguish their service priority. To better define the CR, we describe its basic abilities which are usually classified in two major groups, Spectrum Sensing (SS) and Decision Making (DM). A new trend in CR technology which is adopted in this thesis as well and it is closely related to its origins is the application of Machine Learning (ML) in both SS and DM.

The main enabler of the CR is the SS part which gives "eyes" and "ears" to the "body" of this intelligent radio. Like any cognitive entity, the CR must first examine its environment in order to learn from it and then interact with it. SS applies advanced signal processing and other methods and enables the CR to "observe" the spectrum. One way of making the CR aware of its environment is PU signal detection which mainly focuses on the classic binary hypothesis testing of PU existence. Another way of enhancing the CR's senses is PU signal classification. This new radio must be able not only to detect whether a PU signal exists but also to identify its kind, a direction which we consider in the beginning of this thesis. In Chapter 3, we develop SS techniques based on ML tools in order to identify the Modulation and Coding Scheme (MCS) of the PU, a process that we call Modulation and Coding Classification (MCC). This SS piece of information can play an important role for the second function of the CR, the DM.

The second skill set of the cognition cycle, the DM, represents the interacting abilities of the CR with its environment which actually perform an adaptive configuration of the radio operational parameters, like beamformer, transmit power, modulation, coding, frequency, time schedules and others, and consequently reach some goals, such as system throughput, SINR maximization or interference mitigation. The DM process is a broad scientific area and usually employs optimization tools or other mathematical mechanisms to enhance spectrum usage. In this thesis, the configurable CR parameters we are interested in are the transmit power levels of the unlicensed cognitive users. This CR power level adaptation is called in the literature Power Control (PC) and in Chapters 4 and 5 we demonstrate its applications for achieving optimization and learning goals.

In the CR regime, there are three main application categories which are also characterized as DSA examples, the underlay, the overlay and the interweave mechanisms [5]. In the underlay one, a CR Network (CRN), the SUs, may transmit in a PU frequency band as long as the CRN generated interference to the PUs is under a certain threshold. As far as the overlay methods are concerned, PUs share knowledge of their codebooks and possibly messages with the SUs, so as to reduce the interference on the PU receiver side or even relaying the PU message in order to enhance the PU communication link. In the interweave approach, SUs identify holes in space, time or frequency where PUs are absent and they transmit only in the case of these vacancies. In this thesis, we focus on underlay CRN paradigms, which enable a simultaneous coexistence in frequency of PUs and SUs.

This vast topic has been thoroughly investigated from many aspects depending on the system model, the optimization variables, the objective functions and the constraints. An interesting approach of the underlay problem tackled by the research community is to consider a centralized CRN where a central decision maker, the Cognitive Base Station (CBS), elaborates an intelligent selection of the SU operational parameters and communicates them through a control channel [6]. In this context, network optimization problems have been formulated to achieve common or different SINR requirements for each SU, maximum SU system throughput, maximum weighted throughput, maximum worst SU throughput or minimum transmit power, subject to PU QoS constraints, like SINR, data rate or outage probability.

A major challenge though in all these underlay problems is the knowledge of the interference channel gains from the CR transmitters to the PU receivers and the PU receiver interference threshold which define the PU QoS constraints. This arises because PU systems are usually legacy systems which were not foreseen to coexist and interact with other systems. Therefore, a CRN cannot rely on an access protocol that cooperates with the PUs in order to utilize their frequency bands. This lack of cooperation forbids passing information for inferring the interference channel gains and the PU interference threshold. Yet, the CRN must acquire some kind of knowledge about the CR-to-PU channel gains and the interference limit.

A common approach to tackle this issue in the CR regime is the CRN to exploit a PU link state feedback, monitor how this changes because of the CRN operation and thus estimate these interference parameters. This idea, called proactive SS, was first proposed in [7] where the SU probes the PU and senses its effect from the PU power fluctuation, a rather informative piece of information. The proposed DSA application concerns only the SU system without adding any complexity in the infrastructure or a control channel between the PU system and the SU one in order to exchange information about the

channel gains or the induced interference. Later work retained this probing scheme, but it focused on more rudimentary feedback such as the binary ACK/NACK packet of the reverse PU link which requires the implementation of the complete PU receiver on the CRN side to decode the PU feedback message. The main problems of this approach are the hardware complexity of the complete PU receiver, security issues risen from the exploitation of the PU message and a minimum required SINR of the sensed PU signal to decode the PU message. In Chapter 4, we demonstrate how the MCC feedback can be used instead of the ACK/NACK packet in an underlay PC scenario for both optimizing and learning purposes. This application bridges our work in SS, the MCC, with an essential DM function, the PC. Subsequently, in Chapter 5, we focus on the binary ACK/NACK packet to develop PC techniques whose only goal is the *Active Learning* (AL) of the interference channel gains. Finally, in Chapter 6, the same AL perspective is kept with the constraint of limiting under a threshold the number of NACK packet observations which actually correspond to harmful probing-induced interference events.

1.2 Thesis Organization

In this section, we include the outline of the thesis. We begin with a literature review about ML applications for CRNs in Chapter 2. Next, we introduce a ML approach for signal classification and specifically for MCC purposes in Chapter 3. Chapter 4 deals with an underlay PC scenario where the CRN throughput is maximized subject to an uknown PU interference constraint using the MCC feedback obtained from the PU signal. In Chapter 5, we study an AL PC scheme aiming at learning the PU interference constraint with uncertain ACK/NACK feedback. Finally, in Chapter 6, an constrained AL probing technique is examined with accurate ACK/NACK feedback. For each topic, a brief background is provided along with the considered problem statements and the main contributions are listed.

1.2.1 Chapter 2: Machine Learning for Cognitive Radio Networks

In this chapter, we mention the state of the art ML applications for CRNs by dividing them into two major groups. The first group consists of the works which study ML applications for SS, specifically for signal classification purposes. This includes the signal feature extraction and the classifiers used to process these features. Telecommunication signals have a great variety of characteristics. Among them, there are many distinguishable statistical features which are unique for each signal type and which can be combined with sophisticated classifiers to easily identify the kind of the sensed signal.

The second group includes the works concerning ML applications for DM. This scientific area is extremely vast and it is better explained by separating it into two categories, the centralized and the decentralized ones. Learning how to make decisions aiming at a specific objective with or without other decision-makers in the environment is related to topics like Classic Optimization Theory, Dynamic Programming (DP), Game Theory and Control Theory.

1.2.2 Chapter 3: Modulation and Coding Classification

In this chapter, we study signal feature design, which is mostly based on statistical signal processing, and ML tools to classify the sensed PU signal MCS. The extracted features are the signal Higher Order Statistics (HOS) cumulants and the log-likelihood ratios (LLRs) of the signal code syndromes. The proposed classifier is one of the most powerful ML techniques, the Support Vector Machine (SVM). This overall process is called Modulation and Coding Classification (MCC) and it will be proven helpful for enhancing the PC mechanism of the CRN. The content of this chapter is published in [8–10].

1.2.2.1 Contributions

The contributions of this chapter are as follows:

1. The combination of HOS cumulants, code syndrome LLRs, SVMs and likelihood based code rate classifiers for identifying the MCS of a sensed signal.
2. Comprehensive simulation results are provided to demonstrate the classification accuracy of the suggested scheme even in a low Signal-to-Noise Ratio (SNR) environment.
3. A cooperative MCC procedure is introduced based on plurality voting, which is simple and delivers better detection results than other methods in the multiple hypothesis testing and sensor fusion literature.

1.2.3 Chapter 4: Power Control Using Modulation and Coding Classification Feedback in Cognitive Radio Networks

In this chapter, a case study is considered with a PU and multiple SUs where the PU link is changing its MCS based on an Adaptive Coding and Modulation (ACM) protocol and is operating in its assigned band together with a CRN accessing this band and having

knowledge of this ACM protocol. Our idea is to detect the PU MCS in a cooperative way in the CBS which gathers the sensed MCC feedback from all the SUs through a control channel and combines them using a hard decision fusion rule and subsequently to exploit this multilevel feedback, instead of the binary ACK/NACK packet that is hard to obtain, in order to learn the CR-to-PU channel gains. This channel knowledge is acquired by having the SUs constantly changing their transmit power under the CBS instructions and checking whether the CRN caused the PU MCS to change, a clearly probing procedure. First, a simple AL technique is demonstrated for probing the PU system and learning the interference channel gains of a CRN using the observed MCS degradation of the PU. Next, a PC method aided by interference channel gain estimation is presented which maximizes the total SU throughput subject to maintaining the PU QoS. Furthermore, a novel technique is developed so that the probing/learning method can be performed concurrently with the pursuit of the CRN maximum throughput and without this affecting the learning convergence time. The content of this chapter is published in [10, 11].

1.2.3.1 Contributions

The contributions of this chapter are as follows:

1. For the first time, the MCS degradation is used as a multilevel feedback of the induced interference.
2. A simple AL probing scheme for learning static interference channels is developed.
3. A PC mechanism for static interference channels is proposed where maximizing the total SU throughput subject to an unknown PU interference constraint is taking place simultaneously with an interference channel gain learning process.
4. A dynamic adaptation of this mechanism is proposed for slow fading channels which takes into account a window of the most recently observed feedback.
5. Simulations show faster convergence rate of the latter solution compared to the literature benchmark method and the first one we developed.

1.2.4 Chapter 5: Active Learning of the Interference Constraint with Uncertain ACK/NACK Feedback in Cognitive Radio Networks

In this chapter, we study solely AL probing methods suitable for rapidly estimating the interference channel gains from multiple SU transmitters to a PU receiver and we no

longer consider the MCC feedback, but the ACK/NACK packet. The probing design novelty here is that the uncertainty of the SS feedback, the binary ACK/NACK, is taken into account to design the probing power levels of the CRN. Due to low SNR conditions of the sensed PU signal by the CRN sensing antenna, the ACK/NACK packet may be decoded imperfectly. Therefore, the feedback likelihood must be considered not only for updating our knowledge about the interference channel gains, but also for designing the probing power levels. Initially, a simple Bayesian method based on the univariate Probabilistic Bisection Algorithm (PBA) is considered for uncertain binary feedback and subsequently multivariate Bayesian AL methods are developed using this uncertainty. The content of this chapter is published in [12, 13].

1.2.4.1 Contributions

The contributions of this chapter are as follows:

1. A simple and computationally cheap univariate Bayesian AL technique is presented.
2. The novel construction of multivariate Bayesian AL methods designed for probing the PU and learning fast interference channel gains.
3. An optimality proof is provided for one of these multivariate Bayesian AL methods.
4. A computationally cheap and analytical CPM adaptation is given as a Bayesian AL technique suitable for high dimensional problems.
5. Simulations show the performance superiority of the AL methods developed in the second part of this chapter compared to the literature benchmark method and the one we suggested in the first part.

1.2.5 Chapter 6: Constrained Bayesian Active Learning of the Interference Constraint in Cognitive Radio Networks

In this chapter, we examine an AL probing method using accurate ACK/NACK packet observations. The enhancement introduced in this part of the thesis is that the SU power levels are sequentially design in order to learn the PU interference constraint with as few as possible probing attempts while limiting under a threshold the number of NACK packet observations. This constraint is of great practical significance since it represents specific technical operation constraints of the PU system. A provenly optimal solution for this constrained AL problem is obtained and implemented with a sophisticated,

accurate and fast Bayesian Learning method, the Expectation Propagation (EP). The content of this chapter will be submitted to *IEEE Trans. Signal Process.*.

1.2.5.1 Contributions

The contributions of this chapter are as follows:

1. The novel construction of a provenly optimal Constrained Bayesian AL method designed for probing the PU and learning fast interference channel gains while maintaining the ratio of harmful probing attempts under a limit.
2. A new computationally cheap, fast and analytical implementation of a sophisticated and accurate Bayesian Learning technique, the Expectation Propagation, suitable for the sequential probing design nature of our problem.
3. Simulations show fast learning convergence rates for our Constrained Bayesian AL method and most importantly adequate satisfaction of the harmful interference constraint.

1.3 Publications

The author has published his PhD research in *IEEE* journals, international conferences and one book chapter. The publications are listed below with the acronyms "J", "C" and "BC" defining the journal, conference and book chapter publications respectively.

- J1: A. Tsakmalis, S. Chatzinotas and B. Ottersten, "Centralized Power Control in Cognitive Radio Networks Using Modulation and Coding Classification Feedback", *IEEE Trans. Cognitive Commun. and Networking*, vol. 2, no. 3, pp. 223–237, Sept. 2016
- J2: A. Tsakmalis, S. Chatzinotas and B. Ottersten, "Interference Constraint Active Learning with Uncertain Feedback for Cognitive Radio Networks", *IEEE Trans. Wireless Commun.*, vol. XX, no. XX, pp. XX–XX, XX 2017
- C1: A. Tsakmalis, S. Chatzinotas and B. Ottersten, "Automatic Modulation Classification for Adaptive Power Control in Cognitive Satellite Communications", *Proc. 7th ASMS Conf. and 13th SPSC Workshop*, Sept. 2014
- C2: A. Tsakmalis, S. Chatzinotas and B. Ottersten, "Modulation and Coding Classification for Adaptive Power Control in 5G Cognitive Communications", in *Proc.*

IEEE 14th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC), Jun. 2014

- C3: A. Tsakmalis, S. Chatzinotas and B. Ottersten, "Power Control in Cognitive Radio Networks Using Cooperative Modulation and Coding Classification", in *Proc. 10th Int. Conf. on Cognitive Radio Oriented Wireless Netw. (CROWNCOM)*, Apr. 2015

- C4: A. Tsakmalis, S. Chatzinotas and B. Ottersten, "Active Interference Constraint Learning with Uncertain Feedback for Cognitive Radio Networks", in *Proc. IEEE International Conf. on Commun. (ICC)*, May 2016

- BC1: A. Tsakmalis, S. Chatzinotas and B. Ottersten, "Power Control in Heterogeneous Networks using Modulation and Coding Classification", in *5G Networks: A Research and Development Perspective*, 2016

Chapter 2

Machine Learning for Cognitive Radio Networks

In this chapter, we review the ML literature in CR applications which relates to this thesis. Broadly speaking, we divide the related literature into two parts, ML tools for signal classification in the first part and for DM in the second one.

In the first section, we mention the works which use ML tools for classifying the PU signal, a SS task. We elaborate on signal feature extraction and learning techniques. Feature extraction in the CR regime derives mostly from statistical signal processing. The learning part is comprised of supervised and unsupervised classifiers from ML and also likelihood based classifiers or simple decision tree approaches which were the first approaches in signal classification. In this thesis, we categorize all these classifiers as Bayesian or non-Bayesian.

In the second section, we summarize the main contributions of learning in DM for telecommunication networks and CRNs. We divide this category into two subgroups, the centralized and the decentralized. Subsequently, we attempt to explain the basics of the most noteworthy learning techniques within the DM framework.

2.1 Machine Learning applications for Signal Classification in Cognitive Radio Networks

The enabler of the CR is the sensing part. Throughout most of the existing literature, SS focuses on one hypothesis testing, the existence or absence of PU in a frequency band. Another way of making the CR aware of its environment is to detect the type of

the PU signal. This new radio must be able to identify all kinds of signals depending on the modulation scheme, the code or its rate, the protocol, the waveform and many other signature factors and it has been realized by extracting features of the signal and classifying it based on tools from the ML literature or other simpler ones.

2.1.1 Features

Communication signals contain many characteristics that give us information about their nature. Some of them are extracted in a straightforward way and others in a more complex one. Now, let us consider that the sensed PU signal sample by the CR receiver is:

$$r_{SU}[i] = h_S * s_{PU}[i] + n_{SU}[i] \tag{2.1}$$

where h_S is the sensing channel gain, $s_{PU}[i]$ is the transmitted symbol from the PU and $n_{SU} \sim \mathcal{N}(0, N_{SU})$ is the AWGN. The first and simplest signal feature, which has been widely used [14–17], is its energy:

$$E_r = \sum_{i=1}^{N_s} |r_{SU}[i]|^2 \tag{2.2}$$

where N_s is the number of the samples. Furthermore, if we assume that $A[i]$ is each sample's amplitude, $|r_{SU}[i]|$, then another feature which can be derived is the maximum value of the spectral power density of the normalized centered instantaneous amplitude γ_{max} and it is defined as [18–20]:

$$\gamma_{max} = \frac{max|FFT[A_{cn}[i]]|^2}{N_s} \tag{2.3}$$

where $FFT[.]$ is the Fast Fourier Transform operator and A_{cn} is the zero centered unitary instantaneous amplitude which is defined as:

$$A_{cn}[i] = \frac{A[i]}{E\{A[i]\}} - 1. \tag{2.4}$$

This parameter is considered to be efficient for recognizing different amplitude modulated signals. Additionally, another characteristic of the PU sensed signal is the standard deviation of the absolute value of the nonlinear component of the normalized instantaneous amplitude $\sigma_{\alpha\alpha}$ which is expressed as [18–20]:

$$\sigma_{\alpha\alpha} = \sqrt{E\{A_{cn}^2[i]\} - E^2\{|A_{cn}[i]|\}} \tag{2.5}$$

and it is used to distinguish M-ASK, M-QAM and other amplitude modulated signals. Furthermore, a fourth easy to extract PU signal feature is the standard deviation of the

absolute value of the nonlinear component of the sample phase in the non-weak signal samples $\sigma_{\alpha p}$ which is calculated as [18–20]:

$$\sigma_{\alpha p} = \sqrt{E\{\phi_{NN}^2[i]\} - E^2\{|\phi_{NN}[i]|\}} \tag{2.6}$$

where ϕ_{NN} is expressed as:

$$\phi_{NN}[i] = \phi_N[i] - E\{\phi_N[i]\} \tag{2.7}$$

and $\phi_N[i]$ corresponds only to the phase of the non weak signal samples. These samples are the ones whose amplitude is above a certain amplitude threshold. A very similar to the previous but nevertheless useful feature is also the standard deviation of the nonlinear component of the sample phase in the non weak signal samples σ_{dp} computed as [18–20]:

$$\sigma_{dp} = \sqrt{E\{\phi_{NN}^2[i]\} - E^2\{\phi_{NN}[i]\}}. \tag{2.8}$$

The fifth practical feature taken advantage of for MFSK signal classification is the standard deviation of the absolute value of the nonlinear component of the sample frequency in the non weak signal samples $\sigma_{\alpha f}$ estimated as [18–21]:

$$\sigma_{\alpha f} = \sqrt{E\{f_{NN}^2[i]\} - E^2\{f_{NN}[i]\}} \tag{2.9}$$

where f_{NN} is expressed as:

$$f_{NN}[i] = \frac{f_N[i] - E\{f_N[i]\}}{R_s} \tag{2.10}$$

and $f_N[i]$ corresponds only to the frequency sample of the non weak signal samples, while R_s is the PU signal symbol rate.

Also, numerous metrics of the normalized Power Spectral Density (PSD) of the received signal can provide us useful information about the nature of the signal:

$$S_{r_{SU}}^n(f) = \frac{|R_{SU}(f)|^2}{\int_{-\infty}^{\infty} |R_{SU}(f)|^2 \, \mathrm{d}f} \tag{2.11}$$

where $R_{SU}(f)$ is the Fourier Transform of $r_{SU}[i]$.

Other features are extracted from Time-Frequency (TF) distributions [22, 23] (Short Time Fourier Transform, Wigner-Ville distribution, Choi–Williams distribution etc.). The two most used TF distributions are the STFT and the Wigner-Ville distribution. The first one is known for being the easiest TF distribution to calculate, while the second one for its smooth and continuous characteristics. These functions provide us a view of the signal in both time and frequency domain.

This computationally demanding category delivers more specifically the center frequency C and its spread S. To obtain these features, a threshold is applied to the TF components at each sampled time instant. Let f be the frequencies in the processed data and $\tau(f)$ be the TF components at a certain instant. The threshold at any time instant is the average value of the TF components $\overline{\tau}$ obtained at that time instant. The center frequency C and its spread S are defined as

$$C = \frac{\sum_f fI(f)}{\sum_f I(f)} \tag{2.12}$$

and

$$S = \frac{\sum_f (f - C)^2 I(f)}{\sum_f I(f)} \tag{2.13}$$

where

$$I(f) = \begin{cases} 1, & \tau(f) \geq \overline{\tau} \\ 0, & otherwise \end{cases} \tag{2.14}$$

If only one system transmits at a certain time instant, C and S are good indicators of the center frequency and bandwidth of that system. When multiple systems transmit at the same time instant, these features may technically not mean anything, but do still give a hint of the systems that are involved.

The hardest to extract signal metrics are the cyclostationary (CS) ones. CS signals possess statistical features which vary periodically with time, specifically the cumulants of the signal. This sophisticated and very interesting signal processing approach was first introduced by Gardner and is capable of detecting underlying periodicities [24, 25]. Most researchers who focus on modulation recognition consider signals which exhibit cyclostationarity in second-order statistics, like the autocorrelation function. Still, CS signal processing of second order statistics is not adequate for distinguishing QAM and PSK schemes among themselves, but a CS processing of higher order statistics is. Notable studies which incorporate second-order CS processing to detect modulation schemes and specific codes are [14, 15, 26–32]. Higher order CS has been studied less mostly because of its very high computational complexity. Examples of its use for signal classification can be found in [33–36]. Initially, we present the general definitions from Higher Order CS processing and next we proceed with more specific second order CS features.

Now, let us see how all these metrics are defined for a cyclostationary signal which for annotation simplicity will be referred to as $x[i]$. Initially, we specify the n_{th} order temporal moment function with q conjugated terms, $R_x(i, \boldsymbol{\tau}; n, q)$, as:

$$R_x(i, \boldsymbol{\tau}; n, q) = \hat{E}\left[\prod_{j=1}^{n} x^{(*)_j}(u + \tau_j)\right][i] \tag{2.15}$$

where $\boldsymbol{\tau}$ is the lag vector $[\tau_1, ..., \tau_n]$, $(*)_j$ denotes optional conjugacy (and there are in total q conjugated terms in this product) and $\hat{E}[.]$ is the general sine-wave extraction operator (sometimes denoted as $E^{\alpha}[.]$). Basically, when the operator $\hat{E}[.]$ is applied on a time series $y[i]$, it reconstructs it by using only its periodic components:

$$\hat{E}[y[k]][i] = \sum_{\alpha \in A(y[k])} \langle y[k]\, e^{-j2\pi\alpha k} \rangle_k\, e^{j2\pi\alpha i} \tag{2.16}$$

where the set $A(y[k])$ contains all the nonzero Fourier Series coefficients of $y[k]$ and $\langle . \rangle_k$ is the averaging operator w.r.t. the variable k. Next, we define the n_{th} order cyclic temporal moment function, $R_x^c(\alpha, \boldsymbol{\tau}; n, q)$, as the Fourier Series coefficients in the discrete time of the (poly)periodic moment $R_x(i, \boldsymbol{\tau}; n, q)$:

$$R_x^c(\alpha, \boldsymbol{\tau}; n, q) = \langle R_x(i, \boldsymbol{\tau}; n, q)\, e^{-j2\pi\alpha i} \rangle_i. \tag{2.17}$$

Respectively, we can specify the n_{th} order temporal cumulant function, $C_x(i, \boldsymbol{\tau}; n, q)$, and the n_{th} order cyclic temporal cumulant function, $C_x^c(\alpha, \boldsymbol{\tau}; n, q)$, using the moment-to-cumulant conversion formula. First, $C_x(i, \boldsymbol{\tau}; n, q)$ can be expressed using the temporal moments of equal and lower order as:

$$C_x(i, \boldsymbol{\tau}; n, q) = \sum_p (|p| - 1)!(-1)^{|p|-1} \prod_b R_x(i, \boldsymbol{\tau}; n_b, q_b) \tag{2.18}$$

where p runs through the list of all partitions of $\{1, ..., n\}$, b runs through the list of all blocks of the partition p, $|p|$ is the number of blocks in the partition p, n_b is the number of elements in the block b and q_b is the number of conjugations in the block b. Consequently, $C_x^c(\alpha, \boldsymbol{\tau}; n, q)$ which is described as the Fourier Series coefficients of $C_x(i, \boldsymbol{\tau}; n, q)$:

$$C_x^c(\alpha, \boldsymbol{\tau}; n, q) = \langle C_x(i, \boldsymbol{\tau}; n, q)\, e^{-j2\pi\alpha i} \rangle_i \tag{2.19}$$

can also be written using (2.18) as:

$$C_x^c(\alpha, \boldsymbol{\tau}; n, q) = \sum_p (|p| - 1)!(-1)^{|p|-1} \sum_{\boldsymbol{\beta}\mathbf{1}=\alpha} \prod_{b=1}^{N_b} R_x^c(\beta_b, \boldsymbol{\tau}; n_b, q_b) \tag{2.20}$$

where N_b is the number of blocks in the partition b and $\boldsymbol{\beta}$ is the row vector $[\beta_1, ..., \beta_{N_b}]$.

The multidimensional Fourier transforms of cyclic temporal cumulants are called cyclic polyspectra, or else spectral cumulants. Finally, we must underline that in the vast literature concerning higher order CS processing there is an essential simplification which enables the computations in (2.17): signals are almost always considered cycloergodic.

As far as the second order CS features are concerned, these are derived from all the previous equations where we set $n = 2$ and $q = 1$. Initially, we define $R_x^c(\alpha, \tau; 2, 1)$ as the Cyclic Autocorrelation Function (CAF) which basically measures the correlation of different frequency-shifted versions of a signal. This second order underlying periodicity becomes clear when $R_x^c(\alpha, \tau; 2, 1)$ is different from zero for some nonzero frequency α. The frequency α is called cyclic frequency and the set of cyclic frequencies α for which $R_x^c(\alpha, \tau; 2, 1) \neq 0$ is called the cyclic spectrum. Now, in order to detect the cyclic frequencies easier, there must be a transfer from the time domain to the frequency domain.

For this reason, another useful function in second order CS signal processing is the Fourier Transform of CAF. This is called the Spectral Correlation Density (SCD) function and it is defined as:

$$S_x(\alpha, f) = \sum_{\tau=-\infty}^{\infty} R_x^c(\alpha, \tau; 2, 1) e^{-j2\pi f \tau}. \tag{2.21}$$

Furthermore, it is beneficial to measure the degree of local spectral redundancy derived from spectral correlation. A metric to compute this is the Spectral Coherence function, $C_x(\alpha, f)$, a normalized version of $S_x(\alpha, f)$, which is determined as:

$$C_x(\alpha, f) = \frac{S_x(\alpha, f)}{\sqrt{S_x(0, f + \alpha/2) S_x(0, f - \alpha/2)}}. \tag{2.22}$$

The final step of the second order CS processing is acquiring the α-domain profile or Cyclic Domain Profile (CDP). This is calculated as:

$$I(\alpha) = \max_f |C_r^\alpha(f)| \tag{2.23}$$

and exhibits the peak values of the Spectral Coherence function which are more convenient to handle for signal classification, especially modulation classification.

Additionally, we must mention one of the easiest to extract and most widely used statistical feature set, the signal temporal cumulants [18, 19, 26, 37]. These have distinctive theoretical values among different modulation schemes and even though they demand a great amount of samples, they are easy to be calculated. Their definition will be given in Chapter 3.

Finally, of particular interest are code features like the code length and the number of encoded symbols per block which have been exploited in [38] for recognizing Linear Space–Time Block Codes and the code syndromes which are based on the unique parity-check matrix that each code has and have been used in [39] for Low Density Parity Check (LDPC) code identification. A detailed description of the last code features will be provided in Chapter 3.

2.1.2 Classifiers

As far as the ML methods used in the CR literature are concerned, there are two general categories. The first one is the non-Bayesian group where signal features are computed and using techniques mostly based on optimization, these features are processed and exploited to classify or cluster signal samples as noise or any other kind of signal. The second one is the Bayesian group or likelihood based group which takes advantage of the likelihood functions of statistical signal features, which are most of the times well defined, or of a Bayesian based learning structure and distinguishes the different signal classes by usually comparing likelihood ratios.

Within the non-Bayesian literature, there has been significant progress using ML to identify correctly the kind of the signal. These ML algorithms are divided into two categories, the supervised and the unsupervised classifiers, and their difference is that the first one requires training with labelled data, while the second one does not. Assuming that one of the feature extraction procedures mentioned before is chosen by the CR to process the sensed signal samples and the feature vectors are $\boldsymbol{x}_i \in \boldsymbol{X}$, then the supervised methods also need the corresponding label $y_i \in Y$ in order to be trained, whereas the unsupervised methods can learn without y_i by clustering similar feature vectors.

A popular supervised ML option is the Artificial Neural Networks (ANNs). ANNs have been successfully used for modulation detection [18, 21, 22, 26, 28, 30] and they were one of the early applications of ML in this field. They are biology inspired computational machines which imitate the function of a set of interconnected neurons, the basic processing units of the brain. They have been proven able to store experiential knowledge through a learning process and therefore they became a powerful classification technique. In the ANN model, each of these neurons applies a transformation function on the weighted sum of its inputs to produce an output. A lot of these units can create a sequence of neuron layers, where the first one is the input layer and the last one is the output layer. Between these two layers, there are the intermediate or hidden ones, which are not directly connected to the outside world.

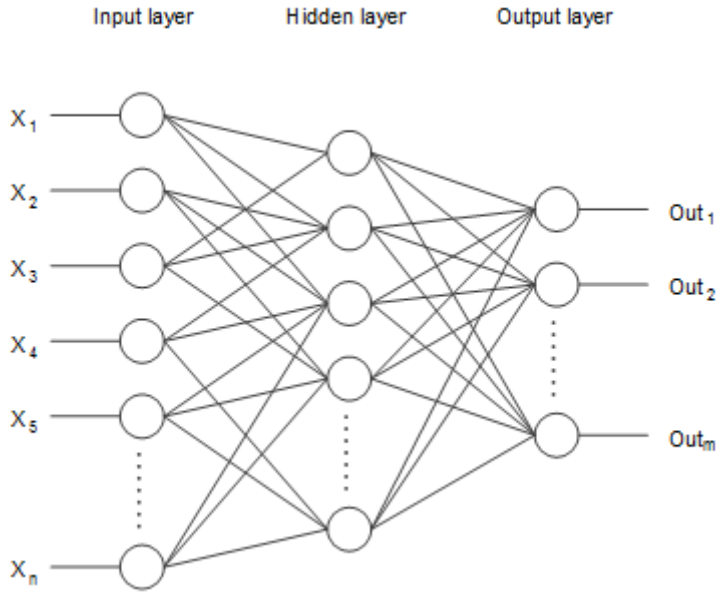

FIGURE 2.1: A Feedforward Neural Network

The kinds of ANNs used in this research area are the Feedforward Neural Networks [18, 21, 22, 26, 28, 30] which have been successful for identifying BPSK, QPSK, FSK, MSK, and AM signals in [28], CCK and OFDM signals in [21], 2ASK, 4ASK, 2FSK, BPSK, QPSK, AM, DSB, SSB, FM, OFDM, 16QAM and 64QAM signals in [18], Pulse compression waveforms in [22] and BPSK, QAM, FSK and MSK signals in [26]. A simple structure of a Feedforward Neural Network is shown in Fig. 2.1. In this network the information moves only forward. From the input nodes, data goes through the hidden nodes and on to the output nodes. There are no cycles or loops in the network.

Another powerful supervised classification tool used in modulation classification literature [19, 29] is the Support Vector Machines (SVMs). Its mathematical foundation is statistical learning theory and it has been developed by Vapnik [40]. It has been successfully used to distinguish BPSK, QPSK, GMSK, 16-QAM, 64QAM, FM and AM signals in [19], AM, ASK, FSK, PSK, MSK and QPSK signals in [29] and BPSK signal and AWGN in [16]. An extensive description of the SVMs will be given in Chapter 3.

An additional classification technique which is simple and moderately effective is the decision trees. By appropriately designing feature thresholds based on feature statistical properties, researchers have managed to discriminate modulation schemes [26, 33, 34], Space-Time Block Codes [32], Spatial Multiplexing and Alamouti Space-Time Block Code Signals [41] and WiMAX and LTE OFDM signals [31]. Furthermore, distance criteria between sensed signal features and their possible ideal no-noise values have been of great importance with satisfactory results. Most of the distance based research work for signal classification concerns modulation scheme classification [35–37].

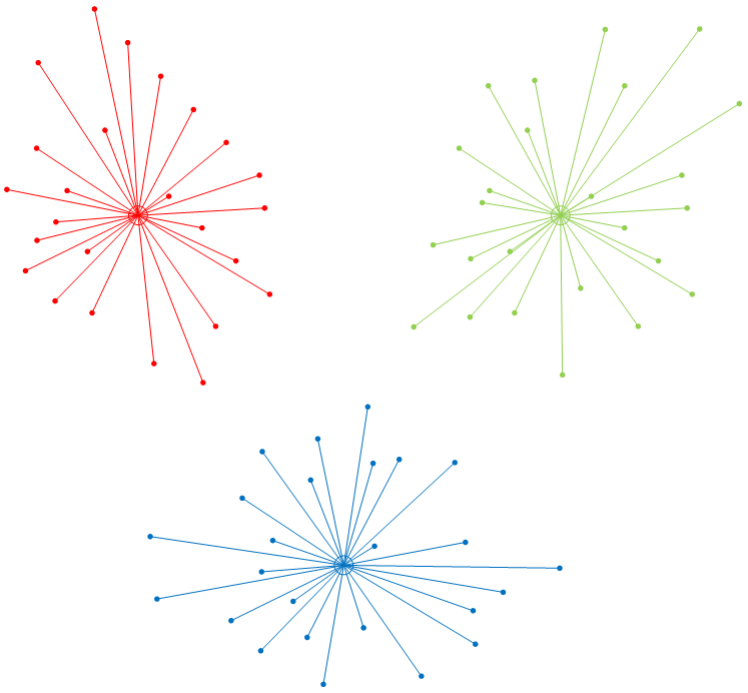

FIGURE 2.2: The k-means clustering

In the second subgroup of this category, the unsupervised one, the signal recognition problem is being tackled in a more autonomous manner. Since this kind of learning methods do not need labelling of the training feature vectors, that means that the CRs equipped with such learning modules can detect signal types without someone indicating the class of the training signal features to the CR.

One of the initial attempts of utilizing such algorithms is the work introduced in [20], where three algorithms are employed, the K-means, the X-means and the Self-Organizing Maps (SOMs), to distinguish 8VSB, OFDM and 16QAM signals. The first technique is clustering the observed feature vectors into a certain given number of classes so as to minimize the sum of the squared distances of all samples from their class centroids as shown in Fig. 2.2. The second one is a variation of the K-means method capable of training without any knowledge of the class number. The SOMs are a special kind of ANN which represent training samples in a low dimensional space, usually on a plane, where the ANN's neurons organize themselves through an neuron weight updating process.

Within the Bayesian literature, signal classification contributions have been made in both modulation scheme and coding recognition. A moment using likelihood based method is presented in [42] for identifying BPSK and QPSK signals. An overview of these techniques concerning the modulation scheme detection can be found in [43], where basically three variations of this likelihood approach are demonstrated, the average likelihood ratio test (ALRT), the generalized likelihood ratio test (GLRT) and the hybrid

likelihood ratio test (HLRT). Likelihood based classifiers have also achieved great results in coding recognition whether they deal with specific coding systems [38, 39] or any other coding schemes with parity check relations [44].

Additionally, a less attractive learning machine the Hidden Markov Models (HMMs) has been applied to recognize AM, BPSK, FSK, MSK and QAM sensed signals in [26, 27]. As a classifier, HMMs are used to process probabilistically a sequence of observations, symbols in this case, and identify from which process, specifically here from which modulation scheme, this sequence was produced. Even though HMMs are powerful classifiers, they require a huge memory space to store a large number of past observations and they are also computationally very complex, which makes them unsuitable for a CR device to embed. Nevertheless, they are modified versions of popular HMM solving algorithms (e.g. Baum–Welch algorithm) which can be used for fast learning but with less accuracy.

Furthermore, non-parametric unsupervised learning procedures have been utilized to recognize signals with notable results. The authors of [14, 15] recommended the Dirichlet Process Mixture Model (DPMM) clustering algorithm and use the Gibbs sampling to sample from the posterior distribution and to update the DPMM hyperparameters. Also, they proposed a simplified and a sequential DPMM classifier in order to reduce the computationally demanding DPMM classifier by exploiting the Chinese Restaurant Process property of the Dirichlet process and improve the selection strategy of the Gibbs sampler. Applying this classifier has been successful for distinguishing WiFi and Bluetooth signals with great accuracy in [15] and Zigbee and WLAN signals in [23].

2.2 Machine Learning applications for Decision Making in Cognitive Radio Networks

Learning procedures have played an important role in DM processes and have shown remarkable theoretical and practical results within the CR framework. A great variety of algorithms has been employed for learning action strategies, or action policies, with many different objectives, such as spectrum vacancy access or spectrum vacancy bidding, for learning equilibria in game modelled CRNs or for learning other player, or agent, strategies. The main characteristic of a DM process is that resources are allocated sequentially subject to some unknown problem parameters and after every allocation there is some piece of information about how successful that allocation was. This instantaneous cost or reward can be exploited to learn strategies achieving some long term objectives. We divide this category into two groups, first the learning methods for centrally organized CRNs and second the learning methods for decentralized CRNs.

In each group, the contributions of each ML technique for DM in CRNs is discussed and their basic structure is explained.

2.2.1 Decision Making in Centralized Cognitive Radio Networks

In centrally organized CRNs, learning how to allocate resources with unknown environment parameters or reactions is basically the same as the DM problem with a single CR, meaning a single agent. This happens because the CBS knows every piece of information that the CRs have and dictates their every action. The research community has formulated this problem basically under the umbrella of DP using techniques from two groups, model-based and model-free network evolution knowledge. This concerns the interactions between the CRN and its RF environment. The CRN receives an observation about the environment O_t, usually this is the PU system, which is indicative of the environment state S_t. This observation may be reliable or unreliable and may be a complete description of the environment state or incomplete. Based on this observation, the CRN, or the single CR/agent, must somehow make a decision. Now, assuming that the CRN has knowledge of the way it interacts with the environment, e.g. the PU spectrum activity and its collision protocol, it may incorporate this into its DM process. After making a decision or as most commonly said in the DM literature taking an action A_t, the CRN receives some kind of reward R_t and observes again the environment to check what kind of effect it caused. This new observation O_{t+1} of the new environment state S_{t+1} follows some dynamics equation or transition probability matrix which describes the interaction of the CRN with the RF environment.

Model-based DM is very closely related to Control Theory which always assumes having knowledge of these dynamics. If this network evolution model is not known or it is too complex, then model-free DM approaches must be adopted. This subject has been studied within the Reinforcement Learning (RL) framework which is basically an Approximate DP formulation. Strictly speaking, learning a strategy or policy, meaning the way an agent takes actions, is a true learning problem only for the model-free case. Nevertheless, we include the model-based approach as well because the agent still needs to learn how to deal with rewards in this multistage problem. Even though this may appear as an optimization or a Control Theory problem, the truth is that the separating lines are very thin.

As far as the model-based approach is concerned, the problem is frequently formed into a Markov Decision Process (MDP) or its probabilistic version, the Partially Observable Markov Decision Process (POMDP), and strategy or policy learning is designed using this framework. In [45], admission control and channel allocation decisions in cognitive

overlay networks are designed to support delay sensitive communications of SUs. The rollout algorithm is employed to derive a suboptimal policy which performs very close to the optimal one. This algorithm is basically a Policy Iteration (PI) method but with a single policy improvement step. The authors of [46], assuming periodic channel sensing, design a MAC protocol using a constrained MDP and learn the randomized access policy by solving a Linear Program (LP).

POMDP formulations are more common in the CR literature, since they tackle the uncertain sensing issue. In [47], a myopic sensing and access policy is implemented which is close to the optimal POMDP solution and performs very close to it. The authors of [48] have tackled a joint sensing and Resource Allocation (RA) problem for interweave scenarios in a more efficient way. By appropriately decomposing the sensing and the RA parts in this multistage discounted problem, they managed to find a stationary policy using either Value Iteration (VI) or PI. This scheme is designed to minimize the cost of sensing, maximize the weighted sum rate of the SUs and limit the probability of interfering with the PUs. In [49], a decentralized cognitive access scheme is proposed but assuming each agent's action and policy is independent of the others and they only way they interact is by a collision constraint where they consider that a collision with a SU is basically a collision with the PU. Basically, the authors based on this assumption they form the decentralized problem into a single agent problem, abstractly exploiting the water-filling rationale. The last case study in the model-based POMDP category is a learning aided method for accessing spectrum holes between PU packet bursts [50]. The proposed mechanism learns the PU traffic pattern using an HMM and utilizes the spectrum holes with a simple stationary suboptimal policy.

In the model-free DM group, RL is the tool seeking domain. The main strategy learning mechanism in this case is the Q-learning, an approximate DP technique which instead of exact gain matrices, it takes decisions based on an approximate gain matrix, which is being updated sequentially through simulation instances. In [51, 52], the problem of dynamic channel assignment has been addressed with Q-learning. A popular trend in Q-Learning, and in DP, is replacing the approximate gain matrix with some adaptive approximation, like an ANN, which is the method of [52] and basically combines the Neuro-DP with RL. Another application of Q-learning has been suggested in [53] for MIMO transmission control where a modification of Q-learning is developed for solving a constrained MDP problem. Other model-free applications have been proposed for channel access in [54] where a locally optimal policy for a constrained POMDP is found based on policy gradient. Finally, MDPs with unknown dynamics have been addressed by learning on-the-fly these dynamics [55, 56].

2.2.2 Decision Making in Decentralized Cognitive Radio Networks

In decentralized CRNs, the employed strategy learning schemes do not differ so much from the centralized case. Usually, learning schemes from the centralized, or single-agent, case are exploited to tackle the decentralized strategy learning scheme. Most of the times, their performance or convergence is not guaranteed, but empirical studies show that with the proper adaptations a decentralized DM method can achieve strategy learning objectives with satisfactory speed. Here, we make a distinction about the model-based and model-free strategy learning as previously and explain why we focus only on model-free strategy learning and more specifically on strategy learning schemes within the Game Theory (GT) framework.

Similarly with optimization, strategy learning may be applied in three ways: centralized, or single agent, distributed and decentralized, or GT based. The first we already described it in the previous subsection. The distributed fashion deals with multiple agents which make decisions on their own and may also share information with other agents. The main characteristic of this scheme is that, just like in distributed optimization, all the agents have the same objective. So, they allocate resources sequentially and on their own and at each time step they receive some kind of feedback which shapes their strategies. This DM problem has been well investigated in distributed adaptive resource allocation and distributed control problems. Distributed versions of water-filling for example are basically such a problem where the threshold violation plays the role of the feedback. In the decentralized case, each agent is on its own, but still coupled with the rest. This means that their objectives are different, but coupled, and information sharing may or may not be happening. This vast category is described in terms of GT and has shown a lot of growth in the CR regime. The dynamics of such a problem are most commonly difficult to be described and therefore a black box approach is adopted, the model-free strategy learning case.

First, we describe applications of Learning Automata (LA) techniques which allow an agent to learn the stationary randomized policy by directly updating the probabilities of state-actions based on the utility feedback. This is a kind of approximate PI in terms of DP. In [57], the joint relay selection and discrete power control problem is investigated in a cognitive relay network subject to the interference power constraint at the PUs and the total available power constraint for the secondary relays. The authors of [58] utilize a special case of the general LA updating rule, the Linear-Reward-Inaction (L_{R-I}) learning, for enabling SUs to access PU channels. In [59], the Bush-Mosteller LA is adopted for learning the Nash Equilibrium (NE) of the repeated power control game in a CRN with a set of aggregated PU interference constraints. The constrained game is transformed into an unconstrained game with the help of Lagrange multipliers.

Moreover, strategy learning schemes based on Q-learning have been used for network selection [60] and controlling aggregated interference in CRNs [61]. Another decentralized strategy learning tool is best-response learning which has been successfully applied in CR auctions for resources modelled as Stochastic Games [62]. Additionally, a strategy learning option is Fictitious play or modifications of it in repeated games for RA [63–65]. Regret based methods are popular in the GT framework and have been exploited for spectrum access [66] and interference mitigation [67]. Interesting approaches have also integrated unsupervised learning mechanisms in strategy learning for estimating other agents' actions. In [68], spectrum access in a CRN is modelled as a repeated auction game subject to monitoring and entry costs. Knowledge about other SU actions is gained using a Bayesian nonparametric belief update scheme based on the Dirichlet process. Finally, simultaneous policy and value learning has been proposed [69] and heuristic RL techniques have been developed for spectrum seasing via auctions in CRNs [70].

Chapter 3

Modulation and Coding Classification

In this chapter, we study signal feature design, which is based on statistical signal processing, and ML tools to classify the sensed PU signal MCS. The extracted features are the signal Higher Order Statistics (HOS) cumulants and the log-likelihood ratios (LLRs) of the signal code syndromes. The proposed classifier for the modulation scheme recognition is one of the most powerful ML techniques, the Support Vector Machine (SVM), and we also employ a likelihood based classifier for the code rate identification. This overall process is called Modulation and Coding Classification (MCC) and it will be proven helpful for enhancing the RF environment awareness of the CRN. The content of this chapter is published in [8–10].

3.1 Introduction

Signal classification has proven to be both in theory and in practice a more enhanced SS task. Throughout most of the existing literature, SS focuses on one hypothesis testing, the existence or absence of PU in a frequency band. Signal classification though goes one step beyond, it identifies the kind of signals in the RF environment which brings more awareness to the CR.

In this chapter, we develop an MCS identifier using specific statistical signal features and advanced learning techniques. Specifically, we compute the signal HOS cumulants and the LLRs of the signal code syndromes and feed them respectively to an SVM for

modulation scheme recognition and to a likelihood based classifier for coding rate identification. The overall recognition process is called Modulation and Coding Classification (MCC) and delivers specifically the following contributions:

- For the first time, a combined modulation scheme and coding rate recognition process is designed which is easy to be implemented and performs efficiently even for sensed signals of low SNR.
- A cooperative MCC procedure is introduced based on plurality voting, which is simple and delivers better detection results than other methods in the multiple hypothesis testing and sensor fusion literature.
- Comprehensive simulation results are provided to demonstrate the classification accuracy of the suggested scheme even in a low Signal-to-Noise Ratio (SNR) environment.

3.2 Modulation Scheme Classification

The chosen signal classification technique extracts HOS cumulants of the signal and classifies them with a reliable and sophisticated ML detector, the SVM. Although a plethora of features exists, only HOS cumulants can be used to discriminate the PSK and QAM modulation schemes and are easily computed. An additional reason for choosing these specific features is the low complexity of their computation, an advantage which is not met in other statistical features like the CS ones. As noted in Chapter 2, higher order CS processing is actually very effective for modulation scheme classification, but it is a very computationally expensive process. Moreover, the SVM is chosen as a very sophisticated and advanced supervised classifier with reasonable implementation complexity.

3.2.1 The Higher Order Statistics Cumulants

In Chapter 2, we presented a vast collection of signal features exploited for signal classification. Here, we examine a new feature set exploited for modulation classification, the sensed signal cumulants which have distinctive theoretical values among different modulation schemes and even though they demand a great amount of samples, they are easy to calculate. Assuming the signal model described in (2.1), the 2nd, 4th, 6th and 8th order mixed cumulants of r_{PU}, $C^r_{2,0}$, $C^r_{2,1}$, $C^r_{4,0}$, $C^r_{4,1}$, $C^r_{4,2}$, $C^r_{6,0}$, $C^r_{6,1}$, $C^r_{6,2}$, $C^r_{6,3}$, $C^r_{8,0}$, $C^r_{8,1}$, $C^r_{8,2}$, $C^r_{8,3}$, $C^r_{8,4}$, have been used successfully by the research community [20, 26, 35] and have delivered results with high probability of detection.

Cumulants are best expressed in terms of raw moments. A generic formula for the joint cumulants of several random variables $X_1, ..., X_n$ is

$$C_{X_1,...,X_n} = \sum_{\pi} (|\pi| - 1)!(-1)^{|\pi|-1} \prod_{B \in \pi} E\left\{ \prod_{i \in \pi} X_i \right\} \tag{3.1}$$

where π runs through the list of all partitions of $1, ..., n$, B runs through the list of all blocks of the partition π and $|\pi|$ is the number of parts in the partition. For example:

$$\begin{aligned} C_{X_1,X_2,X_3} =& E\{X_1X_2X_3\} - E\{X_1X_2\}E\{X_3\} \\ & - E\{X_1X_3\}E\{X_2\} - E\{X_2X_3\}E\{X_1\} \\ & + 2E\{X_1\}E\{X_2\}E\{X_3\}. \end{aligned} \tag{3.2}$$

Consequently, the p-order mixed cumulant $C^r_{p,q}$ of the complex received signal can de derived from the joint cumulant formula in (3.1) as:

$$C^r_{p,q} = C_{\underbrace{r, ..., r}_{\text{(p-q) times}}, \underbrace{r^*, ..., r^*}_{\text{(q) times}}}. \tag{3.3}$$

Because of the symmetry of the considered signal constellations, pth-order mixed cumulant for p odd are equal to zero and also it can be easily proven that for p even $C^r_{p,q} = C^r_{p,p-q}$. The estimates of the previous statistical characteristics are going to be the features fed into a pattern recognition structure which will decide the modulation scheme the signal belongs to.

3.2.2 The Support Vector Machines

Another effective supervised classification tool used in modulation classification literature [19, 29] is the Support Vector Machines (SVMs). Its mathematical foundation is statistical learning theory and it has been developed by Vapnik [40]. A major drawback of the SVMs is that initially they require a lot of computations to train themselves offline but they can become very accurate. The SVMs operate by finding a hyperplane in a high dimensional space which divides the training samples in two classes. This hyperplane is chosen so that the distance from it to the nearest data points on each side is maximized as shown in Fig. 3.1. This is called the maximum-margin hyperplane and basically it is computed by formulating the aforementioned problem into a Quadratic Programming (QP) form and solving it with interior point based solvers.

Nevertheless, the most interesting contribution of the SVMs is in the non-linear separation of data. This machinery with some small adaptations and using the so-called kernel trick can be used to map indirectly input feature vectors into a high dimensional

space in which they become linearly separable [40]. The impressive part of this high dimensional approach is that it happens without any extra computational effort. The

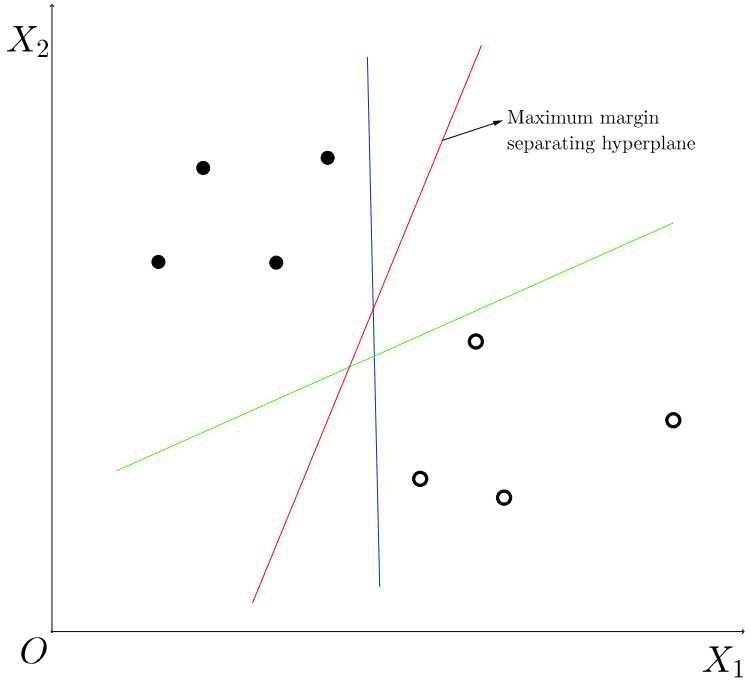

FIGURE 3.1: Maximum-margin separating hyperplane

reason this non-linear mapping Θ does not add any extra computational burden lies on the way the SVM operates. For a simple linear separation in the initial feature space, the SVM training has to solve a QP problem which considers only the dot products of the training feature vectors. Extending this idea to a higher dimension space where the feature vector "images" are linearly separable, the SVM again needs only to know the dot products of the dimensional expansions of the training feature vectors. This enables us to surpass the obstacle of knowing this non-linear mapping Θ and just calculate the dot products of the training feature vector mappings.

This is the point where the kernel trick is used. Given two vectors from the training feature space $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, the dot product of their mappings in some high dimensional feature space is:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Theta(\boldsymbol{x}_i) \cdot \Theta(\boldsymbol{x}_j) \tag{3.4}$$

where $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ denotes the kernel function. In most classification applications, the polynomial function kernel (3.5) and the Gaussian Radial Basis Function (GRBF) kernel (3.6) are used:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (1 + \boldsymbol{x}_i \cdot \boldsymbol{x}_j)^d \tag{3.5}$$

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}). \tag{3.6}$$

In previous work, the most commonly used kernel and also chosen here is the GRBF which is actually a polynomial kernel of infinite degree. Originally, the SVM is a binary classifier, but can also be used for multi-class classification into one of the available classes, here modulation schemes, if we consider a combination of binary classifiers to find to which class it most likely belongs compared to every other one. In this one-against-one approach, the most typical strategy for labeling a test signal feature vector is to cast a vote to the resulting class of each binary classifier. After repeating the process for every pair of classes, the test signal is assigned to the class with the maximum number of votes.

3.2.3 Results for Modulation Scheme Classification

In this section, the performance of the Modulation Scheme Classification method is presented. First, it must be mentioned that the received PU signal through the sensing link can be of 6 types, QPSK, 8PSK, 16QAM, 32QAM, 64QAM and 128QAM, and of lower SNR level than the one in the receiver of the PU link. Also, 2 numbers of symbol samples are tested in the simulations, $Ns_1 = 2048$ and $Ns_2 = 65536$ and the performance of the SVM binary classifier network is examined in the SNR ranges of $[-5, 10]$ and $[-11, 5]$ respectively. Moreover, for each case of Ns and SNR, the training and testing procedures were performed using number of the signals $N_{train} = 10000$ and $N_{test} = 1000$ from each modulation scheme. The metric used to measure the detection performance of the AMC method for a class j is the probability of correct classification (P_{cc}), which is defined as:

$$P_{cc} = \frac{N_{cc}}{N_{test}} \tag{3.7}$$

where N_{cc} is the number of correctly classified by the SVM signals of class j.

In Fig. 3.2 and 3.3, the P_{cc} of the simulations is shown. Initially, an obvious remark is that the higher the SNR of the test signal, the higher the P_{cc}. Furthermore, for a specific SNR the P_{cc} is higher, if the number of symbol samples is increased. Additionally, one can notice that the lower the order of the constellation to be classified, the easier it is for the SVM to recognize it. Another interesting result derived from Fig. 3.2 and 3.3 is that the P_{cc} vs SNR curves form 3 groups. This indicates that some classes have similar P_{cc} behaviour, because some modulation schemes have the same constellation pattern. The 3 groups formed are the QPSK-8PSK, the 16QAM-64QAM and the 32QAM-128QAM.

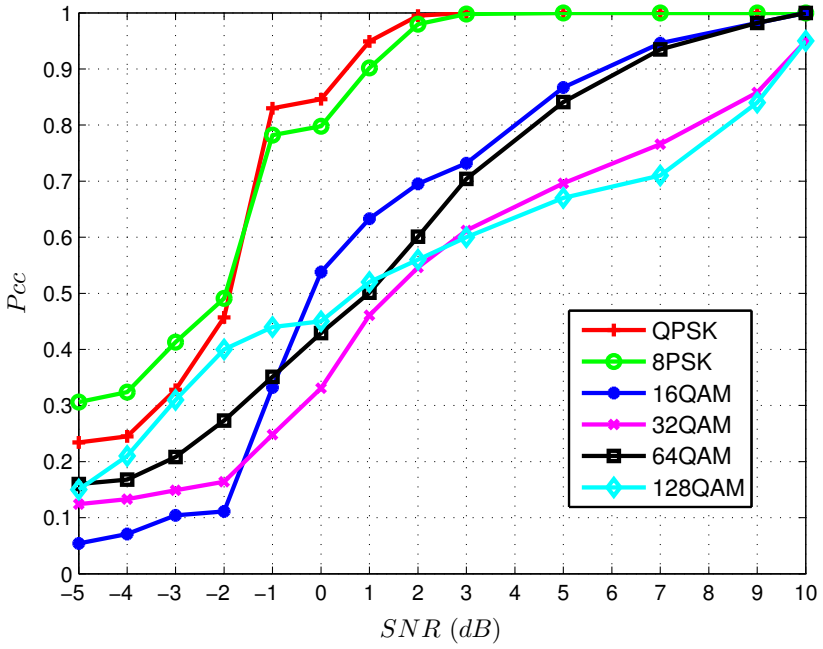

FIGURE 3.2: P_{cc} vs SNR for $Ns_1 = 2048$

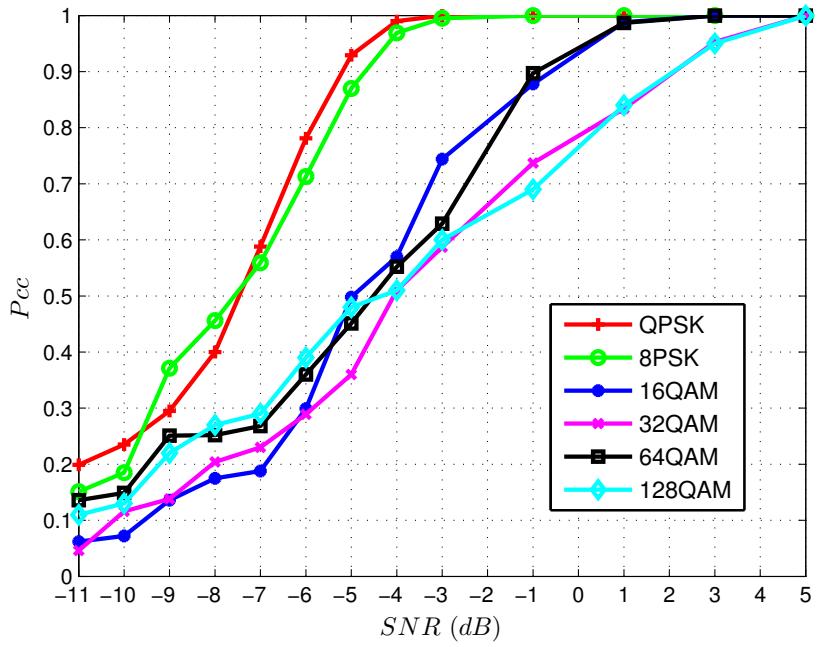

FIGURE 3.3: P_{cc} vs SNR for $Ns_2 = 65536$

Apparently, the SVM classifier has similar performance for modulation schemes of almost identical pattern, such as the rectangular one for the 16QAM-64QAM pair and the cross-like one for the 32QAM-128QAM pair. One more conclusion which has to be noted is that for $P_{cc} = 1$ in all classes, the minimum required SNR for $Ns_1 = 2048$ and $Ns_2 = 65536$ is $10dB$ and $4dB$ respectively.

3.3 Code Rate Classification

In this section, coding classification likelihood based classifiers is considered. An overview of these techniques concerning the modulation scheme detection can be found in [43], where basically three variations of this likelihood approach are demonstrated, the average likelihood ratio test (ALRT), the generalized likelihood ratio test (GLRT) and the hybrid likelihood ratio test (HLRT). Nevertheless, significant contributions with the help of these tools have been achieved for identifying codes. All previous researchers in this area have taken advantage of the log-likelihood ratios to identify codes like Space-Time Block Codes (STBC) [38], Low Density Parity Check (LDPC) code rates [39] or any other coding schemes with parity check relations [44].

Of particular interest mostly because of their practicality are the classification methods based on the unique parity-check matrix that each code has [39, 44]. Assuming that a CR intents to recognize the encoder of a PU transmitter and has a priori information of all the possible PU encoders, a reasonable piece of knowledge about the PU system, then the CR must detect the most likely encoder being used. Each candidate encoder θ' has an exclusive parity-check matrix $\mathbf{H}_{\theta'} \in \mathbb{Z}_2^{N_{\theta'} \times N_c}$, where $N_{\theta'}$ is the number of parity check relations of the candidate encoder and N_c is the length of the produced by the encoder θ' codeword. Given a codeword $\mathbf{c}_\theta \in \mathbb{Z}_2^{N_c \times 1}$ from encoder θ, in a noiseless environment the following

$$\mathbf{H}_{\theta'}\mathbf{c}_\theta = \mathbf{0} \tag{3.8}$$

holds over the Galois field $\mathbb{GF}(2)$ if and only if $\theta' = \theta$. Due to noise in the codeword though, some errors occur in (3.8) even when choosing the correct encoder θ. These errors are called code syndromes e^k and for a candidate encoder θ' in vector form they are defined as

$$\mathbf{e}_{\theta'} = \mathbf{H}_{\theta'}\mathbf{c}_\theta \tag{3.9}$$

where $\mathbf{e}_{\theta'} \in \mathbb{Z}_2^{N_{\theta'} \times 1}$ and every line represents a parity-check relation.

In order to use the code syndromes $\mathbf{e}_{\theta'}$ for code identification, one needs to calculate the LLR of each bit of the codeword $\mathbf{c}_\theta$, which after some processes in the log-likelihood domain is obtained as

$$LLR(c[m]|r_{SU}[n]) = LLR(r_{SU}[n]|c[m]) \tag{3.10}$$

where $c[m]$ is the considered bit and $r_{SU}[n]$ is the corresponding received symbol sample. This is the result of the log-likelihood soft decision demodulation. Subsequently, if $e^k_{\theta'}$

is the syndrome derived from the k^{th} parity check relation of the candidate encoder θ'

$$e_{\theta'}^{k} = c[k_1] \oplus c[k_2] \oplus ... \oplus c[k_{N_k}] \tag{3.11}$$

where N_k is the number of codeword bits taking part in the XOR operations of the parity check relation, then the LLR of $e_{\theta'}^{k}$ is given by

$$LLR(e_{\theta'}^{k}) = 2tanh^{-1}\left(\prod_{q=1}^{N_k} tanh\left(LLR(c[k_q])/2\right)\right) \tag{3.12}$$

a log likelihood property of the XOR operation shown in [71]. Based on these LLRs of the code syndromes, two different approaches have been proposed to identify the right encoder. The first one suggests a GLRT test which assumes a priori information on the distributions of the GLRT nuisance parameters [44]. This is actually the first piece of work taking advantage of the (3.12) lemma for coding identification. The second one proposes as a soft decision metric the average LLR of the code syndromes. This is calculated as

$$\Gamma_{\theta'} = \frac{\sum_{k=1}^{N_{\theta'}} LLR(e_{\theta'}^{k})}{N_{\theta'}}. \tag{3.13}$$

Once, the average syndrome LLRs of all the candidate encoders are calculated, the estimated encoder can be identified as

$$\hat{\theta} = \arg\max_{\theta' \in \Theta} \Gamma_{\theta'} \tag{3.14}$$

where Θ is the set of the encoder candidates.

3.3.1 Results for Modulation and Coding Classification

In this section, a combination of the two aforementioned techniques is proposed capable of performing MCC. The idea is to sequentially combine the modulation and coding rate classifiers into an MCC module to detect the MCS of the sensed signal. So, first the signal cumulants are estimated and fed into supervised classifier, the SVM, in order to identify the modulation scheme with high accuracy and then the average code syndrome LLRs are computed to calculate the soft decision metric of (3.13) and detect the signal source encoder. The presented MCC module is tested in the SNR range of $[-11, 14]$ for a sensed signal of a system which utilizes MCSs of QPSK 1/2, QPSK 3/4, 16QAM 1/2, 16QAM 3/4, 64QAM 2/3, 64QAM 3/4 and 64QAM 5/6 with LDPC coding. Also, the number of symbol samples considered to be sensed in the simulations is $Ns = 64800$ in order to obtain all the necessary statistical features presented in the previous techniques.

The metric used to measure the detection performance of the MCC method for a class j is the probability of correct classification P_{cc} as defined in (3.7). In Fig. 3.4, the

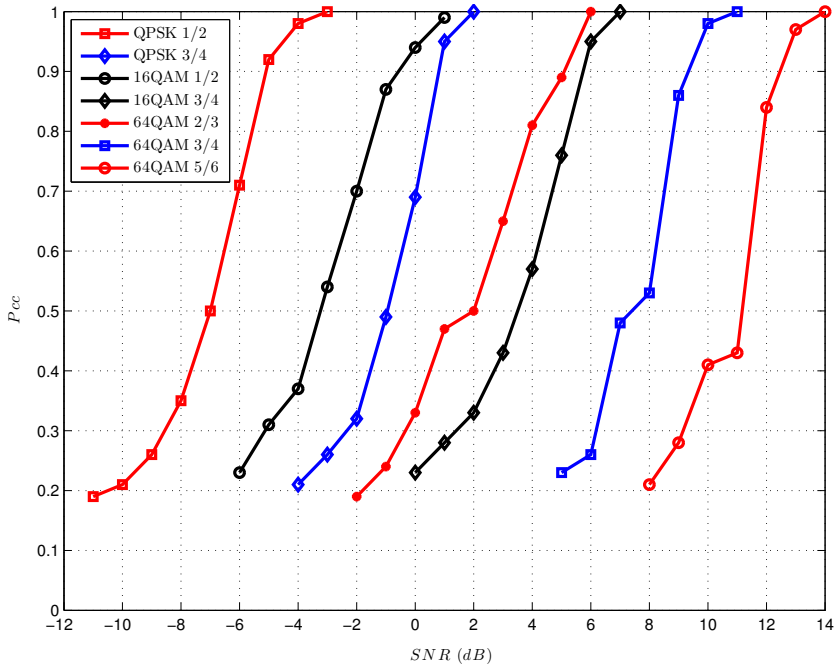

FIGURE 3.4: P_{cc} vs SNR for $Ns = 64800$ symbol samples

P_{cc} performance from the simulations is shown. Initially, an obvious remark is that the higher the SNR of the test signal, the higher the P_{cc}. Furthermore, one can notice that the lower the order of the constellation or the code rate to be classified, the easier it is to recognize it. Other conclusions which can be drawn are that the P_{cc} curves are very steep, mostly due to the performance of the coding classifier, and that they actually resemble the decoding curves for each MCS.

3.4 A Simple Cooperative Modulation and Coding Classification Scheme

In this section, we provide a simple cooperative MCC process which combines MCS estimations from different sources. A general description of cooperative SS in the CR literature is that each SU performs a SS technique independently, forwards its observation to the Cognitive Base Station (CBS) via a control channel and finally the CBS using a fusion rule combines this information to get to a decision. Here, a hard decision fusion of observations obtained by different users equipped with a secondary omnidirectional antenna only for sensing the PU signal and an MCC module is considered using a plurality voting system [72]. Based on this voting system, the CBS collects all the MCC pieces of feedback over a sensing period and decides the MCS of the PU. Let

$C = \{c_1, .., c_K\}$ denote the set of the MCS candidates of the ACM protocol, which are considered to be equiprobable, K is the number of elements of this set and V_{c_j} is the vote tally associated with the class c_j.

During the voting procedure, the CBS first gathers the votes from N SUs, basically the MCS estimates of each SU, of a sensing period denoted as $MCS_1, ..., MCS_N$ and which support elements of the class set C. All the votes are of same importance and no use of weight factors is made. With every vote MCS_i, the CBS increases by one the vote tally V_{c_j} of the c_j class supported by this vote. After casting every vote of the n_{th} period to the corresponding vote tally, the CBS identifies the MCS^n as:

$$MCS^n = \arg\max_{c_j \in C} V_{c_j}. \tag{3.15}$$

Even though plurality voting is a simple and not sophisticated method which elects the MCS value that appears more often than all of the others, it produces the correct voting output under the condition that some SUs have sensing channels of moderate quality. Its equivalent voting system for binary data fusion, the majority one, has been used by the research community to improve the detection and false alarm probabilities with satisfactory results. Additionally, it is appropriate in multiple hypothesis tests where the statistics of the classification metric are not easy to handle, as in our case.

Chapter 4

Power Control Using Modulation and Coding Classification Feedback in Cognitive Radio Networks

In this chapter, an underlay scenario of one PU and multiple SUs is considered where the PU link is changing its MCS based on an Adaptive Coding and Modulation (ACM) protocol and the CRN is assumed to have knowledge of this ACM protocol while attempting to access the PU frequency band. Our idea is to detect the PU MCS in a cooperative way in the CBS which gathers the sensed MCC feedback from all the SUs through a control channel and combines them using the cooperative MCC technique from Chapter 3 and subsequently to exploit this multilevel feedback, instead of the binary ACK/NACK packet that is hard to obtain, in order to learn the interference channel gains. This channel knowledge is obtained by having the SUs sequentially adjusting their transmit power levels dictated by the CBS and checking if the CRN caused the PU MCS to change, a clearly probing procedure. Essentially these gains constitute the PU interference constraint. First, a simple AL technique is demonstrated for probing the PU system and aiming solely at learning the interference channel gains using the observed MCS degradation of the PU. After this probing/learning stage, a CRN performance optimization stage follows where the CRN may maximize its total SU throughput subject to the now known PU constraint. Next, a sophisticated PC method is presented which maximizes the total SU throughput subject to the unknown PU constraint and at the same time learns this unknown PU constraint without this affecting the learning convergence time. The content of this chapter is published in [10, 11].

4.1 Introduction

As shown in 2, SS and specifically signal classification is an important part of the cognition cycle of the CR which may further enhance the SS abilities of the CR. Another essential skill set of the CR is its interacting techniques with the RF environment. In this chapter, the latter abilities concern the transmit power of the unlicensed cognitive users which coexist in the same frequency band with the PUs and they are described as PC. One major category of cognitive PC techniques accomplishing this coexistence is the underlay one [5]. In the underlay CR scenarios, on which we focus here, SUs may transmit in the PU frequency bands as long as the interference induced to the PU is under a certain limit. Therefore, the CRN should learn how to manage properly the transmit powers of its users. As mentioned before, the first stage of the DSA evolution will be the deployment of CRs, the SUs, capable of using their acute senses in order to access frequency bands already used by older communication technologies, the PUs, also referred to as legacy systems. Therefore, the transmit power strategy under which the SUs will access the frequency band of the PUs cannot rely on an access protocol that cooperates with the one adopted by the PU to enter the frequency band, simply because the PUs' infrastructure or protocols cannot be easily changed. A practical approach for the CRN would be the SUs to be coordinated by a Cognitive Base Station (CBS) using a dedicated control channel, which signifies a centralized PC scheme [6]. Still, the CRN must acquire some kind of knowledge about the SU-to-PU channel gains and hence the induced interference to the PU.

Since no cooperation between the PU and SU systems is expected, accurate Channel State Information (CSI) about the interference channels cannot be obtained. In the CR context though, a common approach is the CR individual user or network to exploit a PU link state feedback, monitor how this changes because of the CRN operation and thus estimate the SU-to-PU channel gains. In previous work, this was extracted from the binary ACK/NACK feedback of the reverse PU link [59, 73–75] for PC or beamforming purposes. Here, we must mention that acquiring this binary feedback would require the implementation of the complete PU receiver on the CR side to decode the PU message and retrieve its ACK/NACK feedback. In addition to the hardware complexity issue, this rises security issues for the exploitation of the PU message. Also, to decode the PU message the sensed PU signal on the CR side must have a minimum required SINR, which might not always be the case.

4.1.1 Contributions

In this chapter, an underlay cognitive case study is considered where the PU link is changing its MCS based on an ACM protocol and operating in its assigned band together with a CRN accessing this band and *having knowledge of this ACM protocol*. Our idea is to detect the PU MCS in a cooperative way in the CBS which gathers the sensed MCC feedback from all the SUs through a control channel and combines them using a hard decision fusion rule and subsequently to exploit this multilevel feedback, instead of the binary ACK/NACK packet that is hard to obtain, in order to learn the SU-to-PU channel gains. Our goal in this chapter is to maximize the CRN throughput subject to an unknown PU QoS constraint, the PU interference constraint. The solution of this problem is the well known waterfilling scheme. Our first approach is to first learn the unknown constraint, which basically means learning the SU-to-PU channel gains, by using a simple cognitive probing method where we consecutively adapt the SU power levels and detecting whether an MCS degradation happened or not. This is basically our first AL approach in interference channel gain learning. After this AL stage, the CRN may proceed with maximizing its performance metric, the CRN throughput, subject to the estimated PU interference constraint.

Next, a sequential PC method is demonstrated which maximizes the total SU throughput subject to the unknown PU interference constraint and at the same learns this unknown constraint again by using the MCC feedback. This novel technique is developed so that the probing/learning method can be performed concurrently with the pursuit of the CRN maximum throughput and without this affecting the learning convergence time. Ideal learning approaches for this problem setting are the CPMs, whose high learning rate is not affected severely by the sampling procedure, i.e. the CRN power allocation. In our problem, the sampling procedure is choosing sequentially training data (the SU transmit power levels) which satisfy the optimization objective subject to the estimated interference constraint of each learning step. Here, we focus on two of the fastest CPMs, the analytic center cutting plane method (ACCPM) and the center of gravity cutting plane method (CGCPM). The ACCPM has been used by the research community for enhancing the speed of various learning methods and the CGCPM has attracted attention mostly due to its theoretically fastest convergence rate.

These two design novelties of exploiting the MCC feedback and combining them with an AL procedure in such a way delivers specifically the following contributions:

- For the first time, the MCS degradation is used as a multilevel feedback of the induced interference. As marked in the Chapter 3, MCC is a combined procedure of extracting HOS cummulants and feeding them to an SVM classifier and then

using the code syndrome LLRs with a likelihood based classifier. Therefore, the complexity of the MCC module is much simpler than that of an actual decoder which is used in underlay CR scenarios of other papers to obtain the ACK/NACK packets of the PU reverse link or even of a PU packet preamble decoder. In addition, the MCC feedback provides more information than the binary feedback and therefore improves the learning rate of the interference constraint.

- A simple probing/learning process for learning static interference channel gains is developed with low implementation complexity for designing the CRN probing power vectors and which is published in [11].

- A PC mechanism for static interference channels is proposed where maximizing the total SU throughput subject to an unknown PU interference constraint is taking place simultaneously with an interference channel gain learning process. The optimization part focuses on SU power allocation and assumes that sub-bands of equal bandwidth are allocated to each SU. This mechanism is an enhanced variation of the scheme proposed in [75]. Specifically, in this work a theoretically faster CPM is implemented and used, the CGCPM rather than the ACCPM, and a modification in the sample diversity or exploration process is also introduced based on the proximity to the true learning solution. This work has been published in [10].

- A dynamic adaptation of this mechanism is proposed for slow fading channels which takes into account a window of the most recently observed feedback and which is published in [10].

- Simulations show a convergence rate for the CPM based methods faster than the method developed in [11] which we consider as one of our benchmarks and furthermore a learning speed superiority of the CGCPM based method compared to the ACCPM based technique [75].

4.1.2 Structure

The remainder of this chapter is structured as follows: Section 2 reviews in detail prior work related to cognitive scenarios using a PU link feedback. Section 3 provides the system model and the problem formulation. Section 4 presents the benchmark AL scheme we developed. Section 5 analyses the simultaneous PC and interference channel learning algorithm. Section 6 shows the simulation results obtained from the application of the proposed techniques and compares them with the method of [75]. Finally, Section 7 gives the concluding remarks.

4.2 Related work

Previous work in the field of cognitive underlay PC has considered a great variety of assumptions, protocols, system models, optimization variables, objective functions, constraints and other known or unknown parameters. The general form of the underlay CR scenarios is the optimization of a SU system metric, such as total throughput, worst user throughput or SINR of every SU, subject to QoS constraints for PUs, like SINR, data rate or outage probability [5]. Moreover, the research community has formed combinations of the aforementioned PC problems with beamforming patterns, base station assignment, bandwidth or channel allocation and time schedules, which led to more complicated joint problems, but with the same basic form. Based on the coordination or cooperation of the CR network, PC is separated in two categories, the centralized and the decentralized.

The most important issue arising from cognitive scenarios is the knowledge of the interference channel gains. In prior work, this piece of information was either assumed known [76] or within some uncertainty limits [77, 78]. Although, this presumption helped to devise sophisticated optimization problems, it is not applicable in most cases. Here, we describe scenarios with one common characteristic, no prior knowledge of the CR transmitter to PU receiver channel gain. This assumes that a learning mechanism of the interference channel gains is implemented by a central decision maker or each SU individually. A necessary condition for the learning process is the availability of a feedback which is usually acquired by a SS technique, assuming no cooperation between the CRN and the PU system. An interesting idea was proposed in [7] called proactive SS, where the SU probes the PU and senses its effect from the PU power fluctuation. Further, the exploitation of the MCC feedback, which is used in our work, is suggested briefly by the authors of [7] in a footnote and also thoroughly investigated in [79], a quite recent admission in the CR literature, proving the applicability of such an approach. Primarily though, the most common piece of information being used to estimate the interference channel gains is the binary feedback, which is often obtained by eavesdropping the PU feedback channel and detecting the ACK/NACK packet.

In the decentralized or distributed underlay scenarios, the binary feedback has been used to enable CRs apply Reinforcement Learning procedures, like Q-Learning and Bush-Mosteller Learning, to regulate the aggregated interference to the PU [61] and additionally reach a throughput optimization objective [59]. Formulating this problem as a repeated PC game and employing Game Theory to analyse it [59] has been a critical contribution to explain the behaviour of such a system and prove the convergence of decentralized learning methods. Also, pricing distributed PC schemes have been developed under outage probability constraints [73].

As far as the centralized underlay research work is concerned, a central decision maker, the CBS, must learn the interference channel gains, elaborate an intelligent selection of the operational parameters of the SUs, such as their transmit power, and communicate it to them. Even though distributed PC underlay scenarios have been investigated thoroughly, the centralized PC problem combined with interference channel gain learning is still an unexplored area. Remarkably, the most sophisticated and fast methods suitable for the CBS learning the interference channel gains of multiple SUs with the use of feedback come from multiple antenna underlay cognitive scenarios. In this point, we need to explain how channel learning in beamforming problems can easily be translated as channel learning in centralized CRN PC problems. If you assume that each one of the multiple antennas corresponds to a SU in a CRN, then coordinating the beamforming vectors in order to estimate the CR to PU channel gains is no different than a CBS coordinating the transmit powers of a CRN for the same purpose. In fact, designing the transmit powers is actually much simpler than composing each antenna's complex coefficient in the beamforming scenarios, since in PC no phase parameters are incorporated.

Previous researchers in this field have exploited slow stochastic approximation algorithms [80, 81], the one-bit null space learning algorithm (OBNSLA) [74] and an ACCPM based learning algorithm [82]. The last two approaches were introduced as channel correlation matrix learning methods with the ACCPM based technique outperforming the OBNSLA. All these learning techniques are based on a simple iterative scheme of probing the PU system and getting a feedback indicating how the PU operation is changed. One other thing in common of the aforementioned work is the discrimination of the channel learning phase and the transmission phase which is optimum to an objective, like the maximum total throughput or maximum SINR transmission. Thus, the optimization objective is achieved only after the learning process is terminated, a principle followed in our first AL work [11] and demonstrated in Section 4 of this chapter. Nonetheless, the ideal would be to tackle them jointly and learn the interference channel gains while at the same time pursuing the optimization objective without that affecting the learning convergence time. On this rationale, the authors of [75] proposed an ACCPM based learning algorithm where probing the PU system targets to both learning channel correlation matrices and maximizing the SNR at the SU receiver side.

In this chapter, we exploit this idea in the underlay PC problem by using the MCC sensing feedback instead of the binary ACK/NACK packet captured from the PU feedback channel. In this problem formulation, first a purely learning approach is adopted and a simple AL solution is provided which is also used later on as a benchmark method. Next, we demonstrate an enhanced adaptation of [75] where learning the interference channel gains from each SU to the PU receiver is performed concurrently with maximizing the

total SU throughput under an interference constraint which depends on these channel gains. At the end, we provide results to compare the last method with the benchmark techniques of [11, 75].

4.3 System Model and Problem Formulation

Consider a PU link and N SU links existing in the same frequency band as shown in Fig. 4.1. Furthermore, a Frequency Division Multiple Access (FDMA) method allows SU links to operate in separate sub-bands of the PU frequency band and not to interfere with each other, but still aggregately cause interference to the PU system. In addition, all these PU sub-bands are assumed to have equal bandwidth. The structure of the CRN is a centralized one where the SUs are coordinated by the CBS using a dedicated control channel. The formulation of the problem and the system model is compatible with real world settings such as the cognitive satellite scenarios described in [83, 84]. In one of these case studies, satellite terminals, the SUs, transmit to their appointed satellite and coexist in the same satellite covered area with a microwave link, the PU, which they interfere. Additionally, the satellite terminal operation is being dictated by the gateway and in principle this CRN and the microwave link are not communicating with each other. Concerning the technical details of the problem, the examined scenarios in this chapter are considering the PU channel gain to be static and the unknown interference channel gains static and slow fading. Here we focus on channel power gains g, which in general are defined as $g = \|g^c\|^2$, where g^c is the complex channel gain. From this point on, we will refer to channel power gains as channel gains.

Further, interference to the PU link is caused by the transmitter part of each SU link to the receiver of the PU link. Taking into account that the SU links transmit solely in the PU frequency band, the aggregated interference on the PU side is defined as:

$$I_{PU} = \mathbf{g}\ \mathbf{p}^{\intercal} \tag{4.1}$$

where $\mathbf{g}$ is the interference channel gain vector $[g_1, ..., g_N]$ with g_i being the $\mathrm{SU_i}$-to-PU interference channel gain and $\mathbf{p}$ is the SU power vector $[p_1, ..., p_N]$ with p_i being the $\mathrm{SU_i}$ transmit power. Additionally, the SINR of the PU is defined as:

$$SINR_{PU} = 10 \log \left(\frac{g_{PU} p_{PU}}{I_{PU} + N_{PU}} \right) \mathrm{dB} \tag{4.2}$$

where g_{PU} is the PU link channel gain, p_{PU} is the PU transmit power and N_{PU} is the PU receiver noise power.

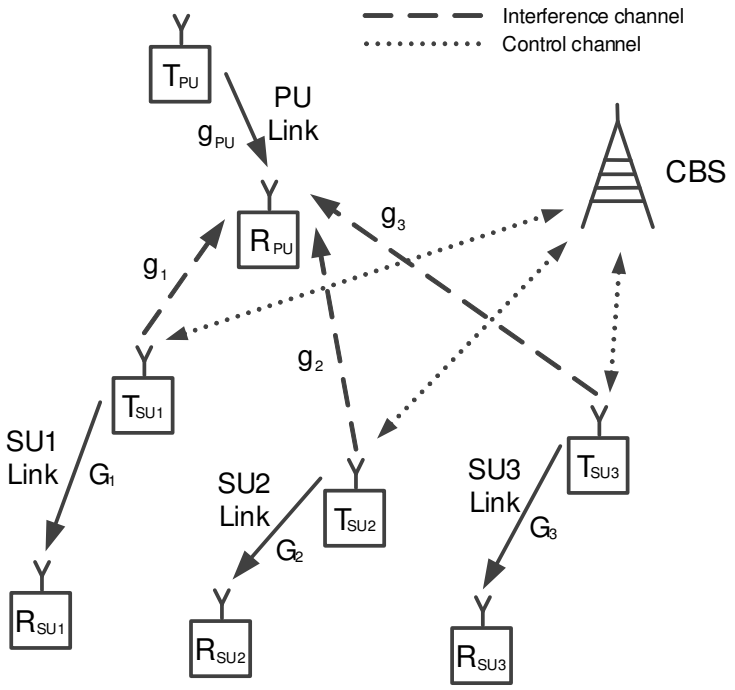

FIGURE 4.1: The PU system and the CR network

In this chapter, we address the problem of total SU throughput (U_{SU}^{tot}) maximization without causing harmful interference to the PU system, which can be written as:

$$\underset{\mathbf{p}}{\text{maximize}} \qquad U_{SU}^{tot}(\mathbf{p}) = \sum_{i=1}^{N} W_i \log\left(1 + \frac{G_i p_i}{N_i}\right) \tag{4.3a}$$

$$\text{subject to} \qquad \mathbf{g}\,\mathbf{p}^{\mathsf{T}} \leq I_{th} \tag{4.3b}$$

$$\mathbf{0} \leq \mathbf{p} \leq \mathbf{p_{max}} \tag{4.3c}$$

where $\mathbf{p_{max}} = [p_{max_1}, ..., p_{max_N}]$ with p_{max_i} being the maximum transmit power level of the SU_i transmitter, G_i is the channel gain of the SU_i link, N_i is the noise power level of the SU_i receiver and W_i is bandwidth the SU_i link. Assuming that the SUs are assigned by the CBS to PU sub-bands of equal size, W_i is equal to $W_{SU} = \frac{W_{PU}}{N}$, where W_{PU} is the PU bandwidth. The channel gain parameters G_i and the noise power levels N_i are considered to be known to the CRN and not to change over time. An observation necessary for tackling this problem is that the g_i gains normalized to I_{th} are adequate for defining the interference constraint. Therefore, the new version of (4.3b) will be:

$$\mathbf{h}\,\mathbf{p}^{\mathsf{T}} \leq 1 \tag{4.4}$$

where $\mathbf{h} = \frac{\mathbf{g}}{I_{th}}$.

This optimization problem is convex and using the Karush-Kuhn-Tucker (KKT) approach a capped multilevel waterfilling (CMP) solution is obtained [76] for each $\mathrm{SU_i}$ of the closed form:

$$p_i^* = \begin{cases} p_{max_i} & \text{if } \frac{1}{\lambda h_i} - \frac{N_i}{G_i} \geq p_{max_i} \\ 0 & \text{if } \frac{1}{\lambda h_i} - \frac{N_i}{G_i} \leq 0 \\ \frac{1}{\lambda h_i} - \frac{N_i}{G_i} & \text{otherwise} \end{cases}, \; i = 1, \ldots, N \tag{4.5}$$

where λ is the KKT multiplier of the interference constraint (4.4) and which can be determined as presented in [76].

Even though this problem setting is well known and already investigated, in the next sections we will demonstrate how to cope with it without knowing the interference constraint (4.4). In the next sections, two algorithms will be described which combine learning the normalized interference channel gain vector $\mathbf{h}$ of (4.4) with the use of an implicit PU CSI feedback.

4.3.1 The Multilevel Modulation and Coding Classification Feedback

In this section, we deal with the MCC feedback, which is the enabler of the interference constraint learning defined by the unknown h_i parameters. Initially, the outputs of the cooperative MCC procedure have to be noted. In Chapter 3, a cooperative MCC method is described where all the SUs are equipped with a secondary omnidirectional antenna only for sensing the PU signal and an MCC module which enables them to identify the MCS of the PU. Specifically, each SU collects PU signal samples, estimates the current MCS, forwards it through a control channel to the CBS and finally the CBS using a hard decision fusion rule combines all this information to get to a decision based on a plurality voting system. After casting every vote, the CBS identifies the PU MCS.

Strong interference links may have a severe effect on the MCS chosen by the PU link, which changes to more robust modulation constellations and coding rates depending on the level of the $SINR_{PU}$. Let $\{MCS_1, .., MCS_J\}$ denote the set of the MCS candidates of the ACM protocol and $\{\gamma_1, .., \gamma_J\}$ the corresponding minimum required $SINR_{PU}$ values, which whenever violated, an MCS adaptation happens. Furthermore, consider these sets arranged such that γ's appear in an ascending order. Here, it has to be pointed out that it is reasonable to assume that the CRN has some a priori knowledge of the standard of the legacy PU system whose frequency band attempts to enter and therefore the CRN can be aware of the PU system ACM protocol and of its γ_j values. Assuming that N_{PU} and the received power remain the same at the PU receiver side, the $\{\gamma_1, .., \gamma_J\}$ values correspond to particular maximum allowed I_{PU} values, designated

as $\{I_{th_1}, .., I_{th_J}\}$. Hence, whenever the PU is active, for every MCS_j it can be inferred that I_{PU} lies within the interval $(I_{th_{j+1}}, I_{th_j}]$, where I_{th_j} is the interference threshold over which the PU is obliged to change its transmission scheme to a lower order modulation constellation or a lower code rate and $I_{th_{j+1}}$ is the interference lower limit below which the PU can change its transmission scheme to a higher order modulation constellation or a higher code rate. Still, the actual values of these thresholds are unknown to the CRN, since the CRN cannot be aware of the N_{PU} and the received power at the PU receiver side.

This groundwork predisposes us how to transform the MCS feedback into a multilevel piece of information. Nevertheless, in our interference channel learning problem we have to encounter the fact that the CRN has no knowledge of $\{I_{th_1}, .., I_{th_J}\}$. To this direction, the observation that learning the interference channel gain vector $\mathbf{g}$ is equivalent to learning the normalized interference channel gain vector $\mathbf{h}$ of (4.4) is essential. Now, taking as reference the PU MCS when the SU system is not transmitting at all, $MCS_{ref} = MCS_k$, and the corresponding $\gamma_{ref} = \gamma_k$, where $k \in \{1, .., J\}$, the following γ ratios can be defined:

$$c_j = \frac{\gamma_j}{\gamma_{ref}} \tag{4.6}$$

where $j \neq k$ and $j \in \{1, .., J\}$. Supposing a high SNR_{PU} regime, $g_{PU}p_{PU} \gg N_{PU}$, the I_{th_j} ratios can also be determined as:

$$\frac{I_{th_j}}{I_{th_{ref}}} = \frac{\gamma_{ref}}{\gamma_j} = \frac{1}{c_j} \tag{4.7}$$

where $I_{th_{ref}}$ is the interference threshold of MCS_{ref}.

The knowledge of these ratios has a great significance for our normalization process which has two steps. Now, let MCS_{ref} be the sensed MCS when the CRN is silent and no interference occurs, $\mathbf{p} = \mathbf{0}$, and MCS_j be the deteriorated MCS after the SU system interfered the PU using an arbitrary SU power vector $\mathbf{p}$. The information gained by the CBS as mentioned before is that:

$$I_{th_{j+1}} < \mathbf{g}\ \mathbf{p}^{\intercal} \leq I_{th_j}. \tag{4.8}$$

These inequalities can be rewritten using the I_{th} ratios as:

$$\frac{I_{th_{ref}}}{c_{j+1}} < \mathbf{g}\ \mathbf{p}^{\intercal} \leq \frac{I_{th_{ref}}}{c_j} \iff \frac{1}{c_{j+1}} < \mathbf{h}\ \mathbf{p}^{\intercal} \leq \frac{1}{c_j} \tag{4.9}$$

where the first step of the normalization process takes place and normalizes $\mathbf{g}$ like in (4.4) with $I_{th} = I_{th_{ref}}$ as $\mathbf{h} = \frac{\mathbf{g}}{I_{th_{ref}}}$.

In the second normalization step, the former inequalities (4.9) are formulated as:

$$\begin{aligned} \mathbf{h}\,\tilde{\mathbf{p}}_{\mathbf{u}}^{\intercal} &> 1 \\ \mathbf{h}\,\tilde{\mathbf{p}}_{\mathbf{l}}^{\intercal} &\leq 1 \end{aligned} \tag{4.10}$$

where $\tilde{\mathbf{p}}_{\mathbf{l}} = c_{j+1}\mathbf{p}$ and $\tilde{\mathbf{p}}_{\mathbf{u}} = c_j\mathbf{p}$. Thus, when interference is introduced to the PU system, the MCC feedback allows us to detect where the interfering SU power vector lies within the feasible region more accurately without searching uselessly the power vector feasible region by using the I_{th} ratios, c. This second normalization step is the advantage of using the multilevel MCC feedback instead of a simple binary indicator, such as the ACK/NACK packet of the PU link, and it will be employed by the learning technique described in the latter section in order to estimate the unknown interference channel gain vector, $\mathbf{h}$, and reach the optimization objective defined by (4.5).

4.4 A Simple Active Learning Algorithm for Interference Channel Gain Learning

From here on, the equality extreme of the constraint (4.4) will be referred to as the interference hyperplane. In this section, a probing method is described for estimating the interference hyperplane where the CRN probes the PU and subsequently applies the MCC technique to monitor the PU MCS and collect the information described earlier. The target of this sequential probing and sensing process is to select SU power vectors which aim to minimize the number of probing attempts. To this direction, an adaptation of the bisection algorithm for higher dimensions is implemented. We suggest performing bisection-like searches on N linear segments in order to find N intersection points with the interference hyperplane and hence the hyperplane itself. For the 2D example in Fig. 4.2, the searches are performed on the line segments OA and OB. The method proposed in this section is to manipulate better the MCC information as a multilevel feedback.

First, the feasible set of this problem is defined as $\Omega^N = \{\mathbf{p}|0 \leq p_i \leq p_{max_i}, i = 1,\ldots,N\}$, an N-dimensional rectangle with 2^N corners. To locate N linear segments crossing the interference hyperplane, a number of end points needs to be known with some of them below the N-dimensional plane and the rest above it. Given that the interference hyperplane crosses the feasible region Ω^N, there is always a known point below this N-dimensional plane, the $[0,...,0]$, and one above it, the $[p_{max_1},...,p_{max_N}]$.

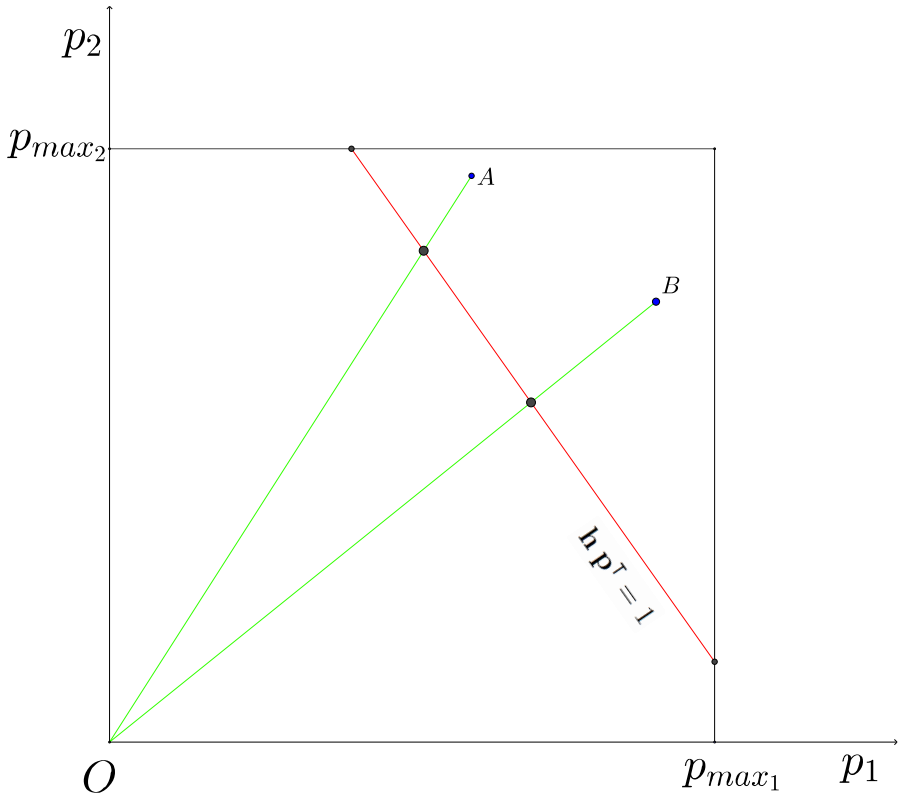

FIGURE 4.2: A 2D graphical example of the algorithm

So, in the worst case scenario, $N-1$ more points are needed to define N linear segments crossing the hyperplane. Now, if we consider that the initial sensing MCC feedback by the CRN when no probing occurs is MCS_{ref}, a simple end point search is to examine random power vectors within Ω^N and check whether they cause or not PU MCS deterioration which indicates whether they are above or below the interference hyperplane respectively. After the end points, $\mathbf{p_i^{end}}$, $i = 1, ..., N$, are found and the line segments are defined, line searches can be performed on each one of them so as to detect the intersection points, $\mathbf{p_i^{in}}$, $i = 1, \ldots, N$ and hence the interference hyperplane. Once the intersection points of the line segments, $\mathbf{p_i^{in}}$, $i = 1, \ldots, N$, are estimated and the interference hyperplane is defined, the h_i gains can be found as the solution of an $N \times N$ system using the equality of the constraint (4.9):

$$\begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_N \end{bmatrix} = \begin{bmatrix} \mathbf{p_1^{in}} \\ \mathbf{p_2^{in}} \\ \vdots \\ \mathbf{p_N^{in}} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{4.11}$$

where $\mathbf{p_i^{in}}$, $i = 1, \ldots, N$, are represented as row vectors.

A detailed description of the search method on a line segment with arbitrary end points should also be given. Assuming 2 points, $\mathbf{Q_1}$ and $\mathbf{Q_2}$, in the N-dimensional space, every point $\mathbf{Q}(\theta)$ lying on the line segment defined by them is expressed using the parametric equation $\mathbf{Q}(\theta) = \theta\mathbf{Q_1} + (1-\theta)\mathbf{Q_2}$, where $\theta \in [0,1]$. So, basically the line search is performed within the θ region $[0,1]$ to find a θ^*. Now, specifically in our scenario, the line searches are performed using feedback which has the form of (4.10). This means that at time step k of the line search, the applied probing power vector $\mathbf{p}(k)$, or testing point, corresponds to $\tilde{\mathbf{p}}_\mathbf{u}(k)$ and $\tilde{\mathbf{p}}_\mathbf{l}(k)$ which on their turn define $\theta_u(k)$ and $\theta_l(k)$ respectively in terms of the aforementioned parametric form. These pieces of information help us to update the uncertainty θ region $[\theta_{low}, \theta_{up}]$ in the following way:

$$[\theta_{low}, \theta_{up}] = \begin{cases} [\theta_{low}, \theta_u(k)] & \text{if } \theta_l(k) < \theta_{low} \text{ and } \theta_u(k) < \theta_{up} \\ [\theta_l(k), \theta_{up}] & \text{if } \theta_l(k) > \theta_{low} \text{ and } \theta_u(k) > \theta_{up} \\ [\theta_l(k), \theta_u(k)] & \text{if } \theta_l(k) > \theta_{low} \text{ and } \theta_u(k) < \theta_{up} \\ [\theta_{low}, \theta_{up}] & \text{if } \theta_l(k) < \theta_{low} \text{ and } \theta_u(k) > \theta_{up} \end{cases} . \tag{4.12}$$

Moreover, in order to probe as less times as possible and give an AL sense in this probing process, we choose to design $\mathbf{p}(k)$ in a bisection-like way where $\mathbf{p}(k)$ corresponds to the midpoint of the uncertainty interval at each step. An overview of the entire process described in this section is presented in Algo. 1.

Algorithm 1 Interference hyperplane estimation algorithm

Sense MCS_{ref}
Search Ω^N for endpoints, $\mathbf{p_i^{end}}$, $i = 1, ..., N$, to define N linear segments
$t = 1$
for $i = 1, \ldots, N$ **do**
  Assume the interval $[0,1]$ for the i_{th} linear segment, choose its midpoint, design $\mathbf{p}(t)$ as $\mathbf{p}(t) = 1/2\ \mathbf{p_i^{end}}$ and probe the PU system
  Sense $MCS(t)$
  Update $[\theta_{low}, \theta_{up}]$ using (4.12)
  repeat
    $t = t + 1$
    Choose the midpoint θ_m of $[\theta_{low}, \theta_{up}]$, design $\mathbf{p}(t) = \theta_m\ \mathbf{p_i^{end}}$ and probe the PU
    Sense $MCS(t)$
    Update $[\theta_{low}, \theta_{up}]$ using (4.12)
  until $\theta_{up} - \theta_{low} \leq \epsilon$
  Choose the midpoint θ_m of $[\theta_{low}, \theta_{up}]$ and define $\mathbf{p_i^{in}} = \theta_m\ \mathbf{p_i^{end}}$
end for
Calculate $\mathbf{h}$ using (4.11)

4.5 The Simultaneous Power Control and Interference Channel Learning Algorithm

Initially, we need to describe the basic rationale of the suggested algorithm. In this work, a proactive approach is adopted where iteratively the PU is probed with some interference and the CRN senses the effect of this interference by detecting the PU MCS as illustrated in Fig. 4.3. The steps of this recurrent algorithm are:

***Step 1*: Design probing and probe the PU**

***Step 2*: Sense feedback and infer the probing impact**

Specifically, in this probing process the CRN designs the probing power vector $\mathbf{p}$, communicates $\mathbf{p}$ to all SUs and probes the PU system, *Step 1* of Fig. 4.3, and next the SUs collect PU signal samples, extract their estimates of PU MCS, send them to the CBS and fuse them to make the final MCS decision, *Step 2* of Fig. 4.3.

Subsequently, the main problem tackled in this section is to find a fast learning method aided by feedback and whose training samples can be chosen by an intervening process without that affecting the convergence time of the learning part. This idea was first explored as a cognitive beamforming problem by the authors of [75] who managed by properly probing the PU system and using only ACK/NACK packets of the PU feedback channel to simultaneously learn channel correlation matrices and maximize the SNR at the SU receiver side by applying a CPM, the ACCPM. CPMs are iterative techniques which cut an uncertainty set in a sequential way using inequalities in order to localize a search point [85].

In each CPM iteration, two pieces of information are needed to define a cut:

- **the center of the uncertainty set**

- **a hyperplane passing through this center**

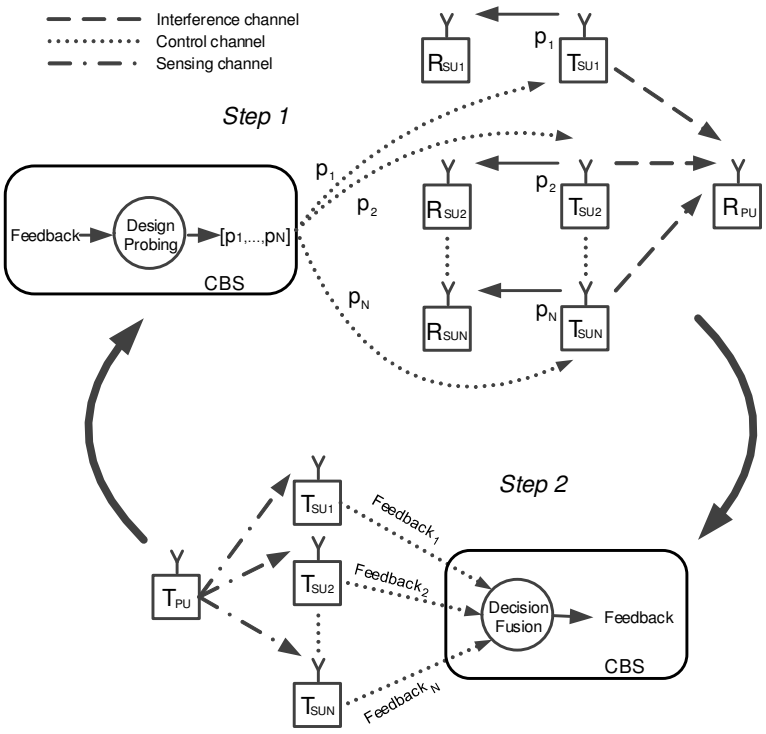

FIGURE 4.3: The algorithm: Probe (*Step 1*) and Sense (*Step 2*)

In our problem, the goal of this learning procedure is to estimate the parameter vector $\mathbf{h}$ of the interference constraint as represented in (4.4) using the SU system probing power vectors as training samples. In this probing procedure, the SU system has the freedom of intelligently choosing the training samples in order to learn and not just receive them from a teaching process. This kind of learning where the learner actually chooses training samples that are more informative so that he can reach the learning solution faster is precisely an AL rationale. The learning speed, and thus the smaller number of probing power vectors, is an essential part of the suggested idea, because of two main reasons. The SU system must learn the interference constraint fast so that first it will not interfere the PU and reduce the PU QoS for a long time and secondly it can apply this learning method in a fading channel environment. Ideal AL methods for this task are the newly introduced to this field CPMs. Still, the CPMs that we have chosen are used to localize points in a search space. For this purpose, a conceptual trick must be used which in Machine Learning literature was introduced by Vapnik [40] and is called the "version space duality". According to that, points in the training sample or feature space are hyperplanes in the parameter or version space and vice versa. Hence, when a learning procedure tries to estimate the parameters of a hyperplane (the version) it actually tries to localize a point in the parameter or version space. In our problem,

the feature space corresponds to the training sample space or the power vector space and the version space to the parameter **h** space, where the point being sought is the endpoint of the interference channel gain vector. In addition, the inequalities obtained by feedbacks (the labels of our training) are meaningful also in the parameter **h** space since they are linear inequalities with respect to h_i's.

One main advantage of CPMs is that the training sample, **p** in this case, can be chosen based on any rationale without that affecting the decrease of the uncertainty region in the parameter **h** space. This rationale can be in our problem the solution of the optimization problem defined in (4.5). Hence, approaching the actual endpoint of the parameter vector **h** can happen in parallel with maximizing the SU system throughout, the optimization objective. More specifically, at each learning step the CPM only dictates the center of the uncertainty set, an estimation of **h**, and the hyperplane/cutting plane passing through this center, which is actually determined by **p**, can be the solution of (4.3). Since the chosen cutting plane passes through it, the SU system power allocation vector is considered to satisfy the equality of the so far estimated interference constraint.

4.5.1 Details of the CPM application to our problem

This section examines the CGCPM and the ACCPM and their corresponding centers, the center of gravity and the analytic center. Now, consider that the initial sensing MCC feedback by the CRN when no probing occurs, $\mathbf{p}(0) = \mathbf{0}$, is MCS_{ref}. Following t probing attempts, the CBS has collected t MCC pieces of feedback which correspond to t pairs of inequalities:

$$\begin{matrix} \mathbf{h}\,\tilde{\mathbf{p}}_{\mathbf{u}}^{\intercal}(k) > 1 \\ \\ \mathbf{h}\,\tilde{\mathbf{p}}_{\mathbf{l}}^{\intercal}(k) \leq 1 \end{matrix} \quad , \; k = 1, \ldots, t. \tag{4.13}$$

The (4.13) inequalities are derived as described in previous section in the form of (4.10) and additionally consider inequalities coming from probing power vectors which do not cause MCS deterioration. In order to keep a single notation in (4.13) even for power vectors not degrading the PU MCS, the first inequality does not hold and $\tilde{\mathbf{p}}_{\mathbf{l}}$ is regarded equal to **p** in this special case. An additional constraint for the h_i parameters is that h_i's have to be positive as channel gains:

$$h_i \geq 0, \quad i = 1, \ldots, N \tag{4.14}$$

The inequalities (4.13) and (4.14) define a convex polyhedron $\mathcal{P}_t$, the uncertainty set of the search problem:

$$\mathcal{P}_t = \left\{ \mathbf{h} \mid \mathbf{h} \geq \mathbf{0}, \mathbf{h}\, \tilde{\mathbf{p}}_{\mathbf{u}}^{\intercal}(k) > 1, \mathbf{h}\, \tilde{\mathbf{p}}_{\mathbf{l}}^{\intercal}(k) \leq 1, k = 1, \ldots, t \right\} \tag{4.15}$$

In the CGCPM, the center of gravity CG of the convex polyhedron $\mathcal{P}_t$ is calculated in vector form as:

$$\mathbf{h_{CG}}(t) = \frac{\int_{\mathcal{P}_t} \mathbf{h}\, dV_h}{\int_{\mathcal{P}_t} dV_h} \tag{4.16}$$

where V_h represents volume in the parameter $\mathbf{h}$ space. The advantages of the CGCPM are that its convergence to the point in search is guaranteed and that the number of the uncertainty set cuts or inequalities needed are of $\mathcal{O}(N \log_2(\frac{R}{r}))$ complexity, where R is the ball radius including the initial uncertainty region and r is the ball radius centered around the true interference channel gain vector endpoint [85]. This convergence rate is ensured by the fact that any cutting plane passing through the CG reduces the polyhedron volume by at least 37% at each step. The main disadvantage of using the CG is its calculation, a computationally expensive integration process in multiple dimensions known to be a #P-hard problem. A way of bypassing this issue is the randomization solution proposed by the author of [86] which computes an approximation of the CG. The general idea is to generate many random sample points within $\mathcal{P}_t$ by taking a random walk, the so called *Hit and Run* method, and average them to find the CG. The computational complexity of the Hit and Run CG approximation to retain the $\mathcal{O}(N \log_2(\frac{R}{r}))$ convergence rate is $\mathcal{O}(N^6)$ [87], since $\mathcal{O}(N^4)$ random walk steps are required and $\mathcal{O}(N^2)$ arithmetic operations need to be implemented for each step.

In the ACCPM, the analytic center AC of the convex polyhedron $\mathcal{P}_t$ is calculated in vector form as:

$$\begin{aligned} \mathbf{h_{AC}}(t) = \arg\min_{\mathbf{h}} \Bigg(& -\sum_{k=1}^{t} \log(\mathbf{h}\, \tilde{\mathbf{p}}_{\mathbf{u}}^{\intercal}(k) - 1) \\ & -\sum_{k=1}^{t} \log(1 - \mathbf{h}\, \tilde{\mathbf{p}}_{\mathbf{l}}^{\intercal}(k)) - \sum_{i=1}^{N} \log(h_i) \Bigg). \end{aligned} \tag{4.17}$$

Interior point methods can be used to efficiently solve the optimization problem described in (4.17) with a computational complexity of $\mathcal{O}(\sqrt{t})$ and estimate the AC which make this center a tractable choice for CPMs [88]. Furthermore, an upper bound for the number of inequalities needed to approach the sought point has been evaluated to prove

the convergence of the ACCPM which is of $\mathcal{O}(\frac{N^2}{r^2})$ complexity, also referred to as iteration complexity.

4.5.2 The Necessity of Exploration

Even though this framework seems ideal for learning the interference constraint and at the same time pursuing the optimization objective, there is still a problem arising. The optimization part, which is responsible for choosing the training power vectors, focuses on cutting planes of specific direction as illustrated in Fig. 4.4. These training power vectors basically correspond to the power level ratios which maximize $U_{SU}^{tot}(\mathbf{p})$ and are subject to the *initial* interference hyperplane estimation. Thus, they focus on specific power level ratios and contribute only in reducing uncertainty in this direction.

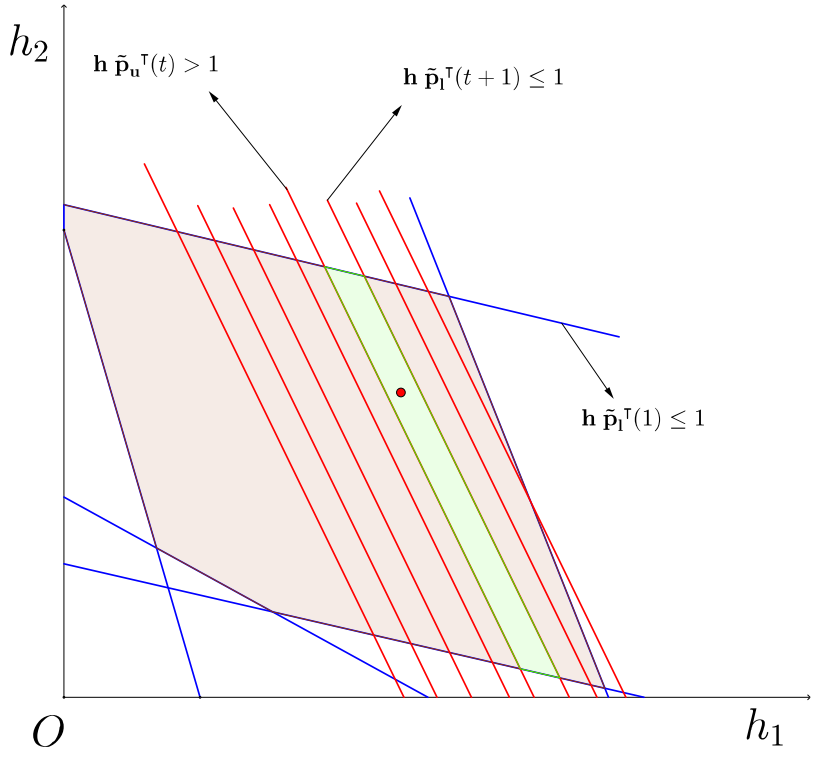

FIGURE 4.4: The CPM in 2D when no exploration occurs

This indicates that choosing the training power vectors based solely on the optimization problem is not a good strategy. Instead, the SU system should start probing the PU system in an exploratory manner by diversifying initially the training power vectors and gradually, when enough knowledge of the interference constraint is obtained, shift to an exploitive behaviour which allocates power levels to the SUs specified by the optimization problem solution (4.5).

The authors of [75] proposed to make this shift from *exploration* to *exploitation* by mixing the optimization objective, the maximization of the SU received SNR, with a similarity metric of the beamforming vectors. The influence of this similarity metric in the design of these probing vectors was determined to be a decreasing function of time, so that the desirable transition could happen. This is a combination of two tactics known in the ML community as the ϵ-decreasing and contextual-ϵ-greedy strategies [89] and according to which the choice of the training samples is performed using an exploration or else randomization factor, ϵ. In these strategies, this factor decreases as time passes or depending on the similarity of the training samples, resulting in explorative behaviour at the beginning and exploitative behaviour at the end. Nevertheless, this logic not only requires tuning of the exploration factor time dependency according to performance results, but it also does not guarantee that enough diversification has occurred to reach the learning goal, which in the case of [75] is the channel correlation matrix, since time on its own cannot be an indicator of approaching the exact values of the sought parameters.

The enhancement introduced in this section is to relate the exploration factor, ϵ, to the proximity of $\mathbf{h}(t)$ to $\mathbf{h}$, where $\mathbf{h}(t) = \mathbf{h_{CG}}(t)$ or $\mathbf{h}(t) = \mathbf{h_{AC}}(t)$ depending on the CPM. Clearly this depends on the geometry of $\mathcal{P}_t$, the region where we search. Towards this goal, a simple approximation of this convex polyhedron, the minimum bounding box containing it, is adopted. The minimum bounding box, $\mathcal{B}_t$, indicates how large the uncertainty region, $\mathcal{P}_t$, is and in order to compute this, we first need to solve the following $2N$ Linear Programs:

$$h_{max_i}(t) = \max_{h \in \mathcal{P}_t} h_i, i = 1, \ldots, N \tag{4.18}$$

$$h_{min_i}(t) = \min_{h \in \mathcal{P}_t} h_i, i = 1, \ldots, N \tag{4.19}$$

which provide us the boundaries for the values of h_i at each step t. Now, let $\mathbf{V}(t) = \{\mathbf{v}_1(t), .., \mathbf{v}_{N_v}(t)\}$, where $N_v = 2^N$, denote the set of the minimum bounding box vertices which are defined straightforward from the boundaries of h_i. A proximity metric of $\mathbf{h}(t)$ to $\mathbf{h}$ could be the euclidean distance of these points $d(\mathbf{h}(t), \mathbf{h}) = \|\mathbf{h}(t) - \mathbf{h}\|$, but the problem is that $\mathbf{h}$ is unkonwn. To fix this, the proximity metric is chosen as the maximum distance of $\mathbf{h}(t)$ from a $\mathcal{B}_t$ vertex:

$$d_{max}(t) = \max_{\mathbf{v}_j(t) \in \mathbf{V}(t)} d(\mathbf{h}(t), \mathbf{v}_j(t)) \tag{4.20}$$

which is an upper bound of $d(\mathbf{h}(t), \mathbf{h})$. The proposed error driven solution is to relate ϵ to this proximity metric, a variation of the tactic known as adaptive ϵ-greedy strategy. According to this, the closer the learning algorithm gets to the exact value $\mathbf{h}$, the less

exploration occurs and training power vectors are more relative to the optimization problem solution (4.5). A simple design to adapt ϵ is:

$$\epsilon(t) = \begin{cases} 1 - \frac{d_{th}}{d_{max}(t)} & \text{if } d_{max}(t) > d_{th} \\ 0 & \text{if } d_{max}(t) \leq d_{th} \end{cases} \tag{4.21}$$

where the threshold d_{th} is linked with the precision limit that the learning algorithm has. That signifies that once $d_{max}(t)$ passes below this threshold, the algorithm has reached the exact solution within an error bound and thus there is no need to explore, but to exploit and choose power vectors according to (4.5).

Moreover, the usage of $\epsilon(t)$ has to be specified and the way the training power vectors are chosen in case of $\epsilon(t) > 0$. As mentioned before, $\epsilon(t)$ is a randomization factor which imposes that the power vector must be chosen randomly with $\epsilon(t)$ probability and the reason for that is to differentiate the cutting hyperplanes passing through the AC or CG of the CPM procedure. This random selection of power vectors is better explained in the power vector space, i.e. the variable space. The random power vector has to satisfy first the equality version of the so far estimated interference constraint (4.4):

$$\mathbf{h}(t)\ \mathbf{p}^{\intercal} = 1 \tag{4.22}$$

and second the constraints (4.3c). Consequently, this random selection is translated into a uniform sampling on the simplex piece $\mathcal{S}(t)$ defined by (4.22) and (4.3c).

4.5.3 The Static and Slow Fading Channel Formulation of the Algorithm

To clarify all this process described thoroughly in the previous section, we present it in Algo. 2. Specifically, in the t_{th} iteration of this process the CRN designs the probing vector $\mathbf{p}(t)$ and probes the PU system, which requires a T_p period for the CBS to calculate and communicate $\mathbf{p}(t)$ to all SUs and for the CRN to actually probe the PU (*Step 1* of Fig. 4.3), and the CBS detects the PU MCS, $MCS(t)$, which demands a T_s period for all SUs to collect PU signal samples, extract their estimates of PU MCS, send them to the CBS and amass them to make the final MCS decision (*Step 2* of Fig. 4.3). It also must be mentioned that Algo. 2 has no stopping criterion. This is actually a consequence of the exploration factor design, because as time passes by, the interference channel gains are better estimated and thus the probing design process switches from power vectors which are more *informative* about the interference channel gains to power vectors which maximize the CRN capacity. Therefore, the learning and the optimization

parts, which depend on the exploration/exploitation strategy, are actually intertwined which means that there is no need for the algorithm to terminate after some time, since it will naturally switch to designing power vectors for CRN capacity maximization.

Algorithm 2 The Simultaneous Power Control and Interference Channel Learning Algorithm

$t = 0$
$\mathbf{p}(t) = \mathbf{0}$
Sense $MCS(t)$
Assume an initial $\mathbf{h}(t)$
loop
  $t = t + 1$
  Compute $\epsilon(t)$
  Generate $rand \in (0, 1)$
  if $rand \geq \epsilon(t)$ **then**
    Exploit: $\tilde{\mathbf{p}}(t) = \arg\max U_{SU}^{tot}$ s.t. $\mathbf{h}(t)\ \tilde{\mathbf{p}}^{\intercal} = 1$
  else
    Explore: $\tilde{\mathbf{p}}(t) =$ random point $\in \mathcal{S}(t)$
  end if
  Sense $MCS(t)$
  Create new pair of inequalities (4.13)
  Compute $\mathbf{h}(t)$ using a CPM
end loop

Here, we must emphasize on two practical considerations related to the algorithm operation. First, the PU cannot instantly change its MCS once interference is caused. In reality, the PU needs time to detect this interference and adapt to a new MCS. In case the CRN probes and estimates faster than the PU can adapt itself, then the PU will not have adequate time to adjust its transmission to interference caused by a specific SU power vector. But even if the PU does adapt its transmission and change its MCS, on the next step the CRN will falsely know that the cause of this MCS change was the last SU power vector. Therefore, the CRN must be aware of the PU adaptation period in order to probe the PU at least for that period of time and then detect the PU MCS. Secondly, the messaging overhead has to be analysed which defines the CRN control channel. The first kind of messages being passed through the control channel are the PU MCS estimates from the SUs to the CBS which require $\lceil \log_2(J) \rceil$ bits considering there are J MCS candidates of the PU ACM protocol. The second kind of messages are the transmit power commands from the CBS to the SUs which demand $\lceil \log_2(N_{pl}) \rceil$ bits if we assume that the SU power range is discretized to N_{pl} power levels. It is also assumed that all the previous messages are being communicated correctly and no errors occur.

A formulation for slow fading interference channels is also given with some modifications of Algo. 2. The solution proposed in this section is window-based in contrast with the

maximum likelihood concept suggested in [75] which considered a probit modelling of each inequality age. To approach the case of slow fading interference channels, first we must take into account the grade of channel variation over time. For this purpose, a quasi static block fading modelling of the interference channels is chosen, according to which the interference channel gains remain constant within a block period, also called coherence time. Assuming that the coherence time T_c of the interference channels is known and the same for all interference channels, the crucial problems we need to tackle is the asynchronous change of the interference channel gains and the lack of knowledge about the exact time an interference channel change occurs. In order to handle these issues, first we calculate how many probing and sensing time periods fit in the coherence time, approximately $t_c = \frac{T_c}{T_p+T_s}$. From these t_c iteration periods which correspond to an equal number of probing power vectors and sensing inequality pairs, we recommend to use for the slow fading algorithm formulation the last $t_w = \lfloor \frac{t_c}{N} \rfloor$ inequality pairs to construct a time window from the $(t-t_w)_{th}$ to the t_{th} probing and sensing period. This actually changes the set of inequalities taken into account to compute the $\mathbf{h}(t)$ using a CPM in order to include only the latest t_w inequality pairs:

$$\begin{matrix} \mathbf{h}\,\tilde{\mathbf{p}}_{\mathbf{u}}^{\intercal}(k) > 1 \\ \\ \mathbf{h}\,\tilde{\mathbf{p}}_{\mathbf{I}}^{\intercal}(k) \leq 1 \end{matrix} \quad , \; k = t-t_w, \ldots, t. \tag{4.23}$$

More precisely, the convex polyhedron is no longer defined by (4.13) and (4.14), but by (4.23) and (4.14).

In this overall description of the proposed algorithm, we must also mention a simple practical adaptation of the algorithm which can tackle fading PU channels. In this case, the normal operation PU MCS may change because of the dynamic PU link nature. This can have a severe effect in the algorithm operation, since MCS_{ref} will no longer be static. In order to confront this, the CRN may adopt a duty cycle operation where it can periodically stop transmitting and solely sense the current normal operation PU MCS.

4.5.4 Multiple PU interference constraint learning

Now, let us consider the multiple PU interference constraint learning scenario. Here, we assume a PU system with M users where each PU is assigned to a separate frequency band. In this section, we will show how to tackle this multiple constraint problem by decoupling it. An important piece of information the CRN must have to achieve this decoupling is the way the PUs occupy the PU system bandwidth which is determined by

the number of the PU channels and their bandwidth. Once this is known, a CRN may partition the N SU set to M subsets and spread them over the PU system bandwidth in an FDMA fashion again as shown in Fig. 4.5 so that no SU interferes to more than one PU. Each SU assigned to subset m occupies a sub-band of length $W_{SU_m} = \frac{W_{PU_m}}{N_m}$. Each SU subset is defined as $\{SU_{1,m}, .., SU_{N_m,m}\}$ where $m = 1, \ldots, M$ and N_m is the number of elements of the m_{th} subset.

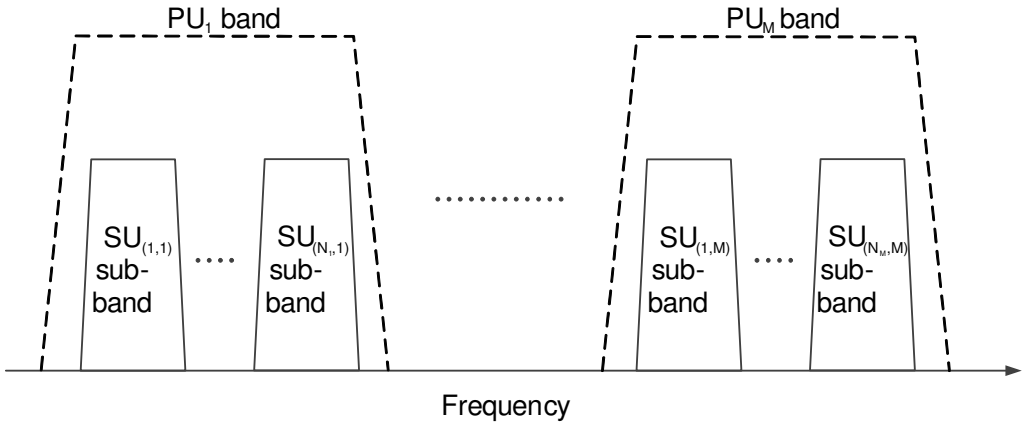

FIGURE 4.5: The SU FDMA scheme in the multiple PU scenario

This decomposition allows the CBS to separate the multiple interference constraint AL to multiple AL sub-problems and thus execute simultaneously our proposed method for each PU and SU subset. Hence, the original problem can be expressed into the following M constraint learning sub-problems:

$$\mathbf{g}_m \, \mathbf{p}_m^{\intercal} \leq I_{th,m}, \; m = 1, \ldots, M \tag{4.24}$$

where $\mathbf{g}_m$ are the interference channel gain vectors $[g_{1,m}, ..., g_{N_m,m}]$ with $g_{i,m}$ being the $\mathrm{SU_{i,m}}$-to-PU interference channel gain, $\mathbf{p}_m$ are the SU power vectors $[p_{1,m}, ..., p_{N_m,m}]$ with $p_{i,m}$ being the $\mathrm{SU_{i,m}}$ transmit power and $I_{th,m}$ are the $\mathrm{PU_m}$ interference thresholds.

In order for this approach to work, each SU must sense only within the PU band it is assigned. Otherwise it may detect the MCC feedback of a PU which it does not interfere and therefore contribute incorrectly to its corresponding cooperative MCC process. Thus, extracting the MCC feedback for each PU is also a decoupled procedure which provides in every sensing period the following inequalities:

$$\begin{aligned} &\mathbf{h}_m \, \tilde{\mathbf{p}}_{\mathbf{u,m}}^{\intercal} > 1 \\ &\qquad\qquad , \; m = 1, \ldots, M. \\ &\mathbf{h}_m \, \tilde{\mathbf{p}}_{\mathbf{l,m}}^{\intercal} \leq 1 \end{aligned} \tag{4.25}$$

4.6 Results

In this section, we provide simulation results to compare the performance of the benchmark and the CPM based method proposed in this chapter and of the benchmark solution given in [75]. Our benchmark method is a computationally cheap AL method which performs consecutive 1-D line bisection-like searches in the SU power vector feasible region in order to find the interference hyperplane and it is expected to have worse learning performance than the CPM based techniques which actually perform high dimensional bisections in the version space. The CPM based methods we developed are an enhancement of the ACCPM based simultaneous channel correlation matrix learning and beamforming solution provided in [75]. Furthermore, the CGCPM is tested to validate its theoretically faster convergence compared to that of the ACCPM. Additionally, the benefit of utilizing the multilevel MCC feedback instead of the binary ACK/NACK packet is demonstrated for all the aforementioned techniques. To prove the MCC feedback superiority, we have chosen the legacy PU system to be operating using an ACM protocol close to the outdated technical specifications of 802.11a/g with LDPC coding [90, 91]. The selected MCS set and the corresponding γ values are:

TABLE 4.1: The PU ACM protocol

MCS	$\boldsymbol{\gamma}$
$BPSK$ 1/2	5dB
$BPSK$ 3/4	6dB
$QPSK$ 1/2	7dB
$QPSK$ 3/4	9dB
16QAM 1/2	13dB

Also, the PU receiver is chosen to normally operate at $SINR_{PU} = 20$dB with no interference and $N_{PU} = -103$dBm resulting in $MCS_{ref} = 16QAM$ 1/2. The I_{th}, which corresponds to 16QAM 1/2 and over which a PU MCS adaptation occurs resulting in PU QoS deterioration, is -97dBm and it is unknown to the CRN. Given the information in Table 4.1, the formulation of the γ ratios can easily be written using (4.6) in order to construct the normalized inequality pairs (4.10). Additionally, the threshold d_{th}, which is related to the precision limit of the learning algorithm and to the exploration factor design, is chosen at 5% signifying that once the learning error upper bound, $d_{max}(t)$, is below 5% the algorithm no longer explores but solely exploits to achieve the CRN throughput maximization.

Initially, the static interference channel scenario is examined with $N = 5$ SUs which are dispersed uniformly within a 3km range around the PU receiver. The interference channel gains that are unknown to the CRN are assumed to follow an exponential path loss model $g_i = \frac{1}{d_i^4}$, where d_i is the distance of the SU_i from the PU receiver in metres.

The last SU operational parameter is the maximum transmit power, p_{max_i}, which is set to 23dBm for all SUs. The aforementioned simulation parameters are also collected in Table 4.2.

TABLE 4.2: Simulation Parameters

PU Parameters for $\mathbf{I_{PU}} = \mathbf{0}$	**Value**
MCS_{ref}	$16QAM$ 1/2
$SINR_{PU}$	20dB
N_{PU}	-103dBm
I_{th}	-97dBm
CRN Parameters	**Value**
N for static channel scenario	5 and 10
N for slow fading channel scenario	5
p_{max_i}	23dBm
d_{th}	5%
T_c	$250(T_p + T_s)$
t_w	50

Fig. 4.6 shows the channel estimation error diagrams for the benchmark, ACCPM-based and CGCPM-based methods depending on the number of time flops where each time flop is the time period $T_p + T_s$ necessary to coordinate the CRN, probe the PU system, sense the MCC feedback and decide collectively the PU MCS. The interference channel gain vector estimation error metric at each time flop is defined as the normalized root-square error $\frac{\|\mathbf{h}(t)-\mathbf{h}\|}{\|\mathbf{h}\|}$. The error figure results are obtained as the average of the error metric defined earlier over 100 SU random topologies, which deliver 100 random draws of interference channel gain vectors $\mathbf{g}$.

It can be clearly seen in Fig. 4.6 that the CPM-based methods outperform the benchmark AL method we first developed and which is based on 1-D bisection-like searches. This occurs because our benchmark method may be the fastest AL method in the training sample space, but the proposed CPM-based methods are performed in the version space, which appears to be more efficient. More specifically as far as the method comparison is concerned, for an estimation error approximately 1%, our benchmark method achieves convergence in 78 and 65 time flops for binary and MCC feedback respectively, whereas the corresponding numbers of time flops for the ACCPM-based benchmark technique are 61 and 55 and for the CGCPM-based one are 55 and 50. For the binary feedback, a gain of at least 17 time flops is accomplished and for the MCC feedback the gain is at least 10 time flops.

Another outcome is that the utilization of the MCC feedback instead of the binary ACK/NACK packet reduces the convergence time significantly in our benchmark method and noticeably in the CPM-based AL methods. Specifically, for an estimation error of

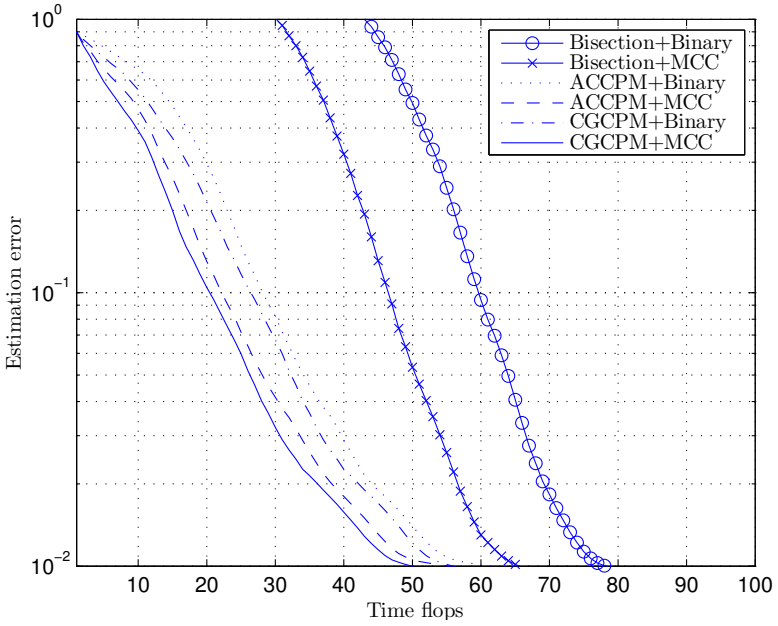

FIGURE 4.6: Interference channel gain vector estimation error progress vs time of all method and feedback combinations for 5 SUs

1%, in our benchmark technique this gain of time flops is almost 13 and in the CPM-based techniques it is nearly 6. Even though the convergence time reduction is small in the CPM case, it is considered a notable enhancement considering that CPM-based techniques are already fast enough. The final conclusion derived from Fig. 4.6 is about the comparison of the two CPM-based learning mechanisms. It is observed that the CGCPM-based scheme surpasses the benchmark ACCPM-based one and particularly for an estimation error of 1% the CGCPM-based procedure outperforms the ACCPM-based one in the binary feedback case by 6 time flops and in the MCC feedback case by 5 time flops.

In the next diagrams, we investigate an important aspect of the methods presented so far, the aggregated interference caused to the PU during the simultaneous learning and CRN capacity maximization process. As all these probing methods progress in time, it is essential to examine the degradation of the PU link quality which can be quantified as the induced harmful interference. To this direction, we designed a metric which measures the PU interference exceeding I_{th} averaged over the 100 SU random topologies, the scenarios of our simulations. This parameter of average harmful interference over the 100 SU random topologies is expressed as:

$$I_{harm,av}(t) = E\left[H\left(I_{PU}(t) - I_{th}\right) * I_{PU}(t)\right] \tag{4.26}$$

where E is the expectation operator and H is the Heaviside step function. In Fig. 4.7 and 4.8, we may see for the simple AL benchmark, the ACCPM-based benchmark and

the CGCPM-based methods the $I_{harm,av}$ progress in time for binary feedback and MCC feedback respectively. Originally, it is clear by comparing Fig. 4.7 and 4.8 that taking

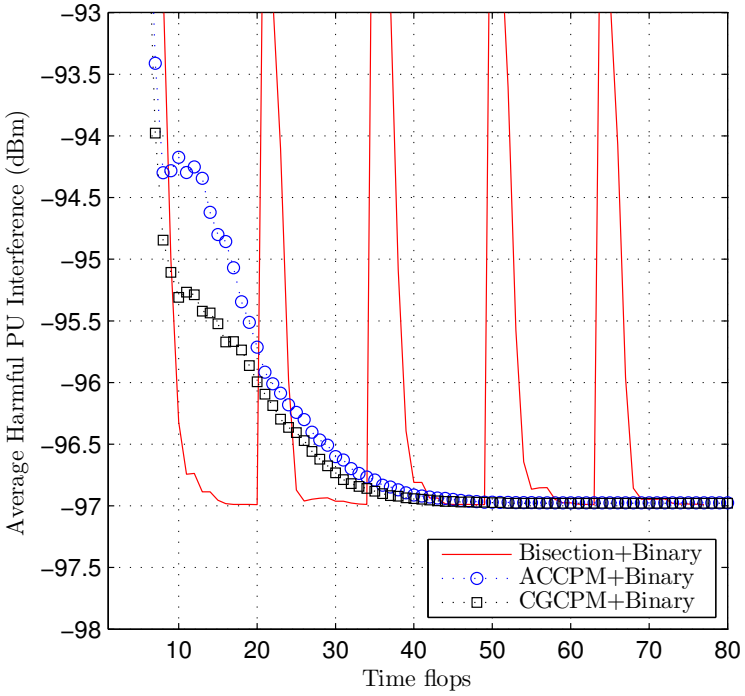

FIGURE 4.7: $I_{harm,av}$ progress vs time using binary feedback

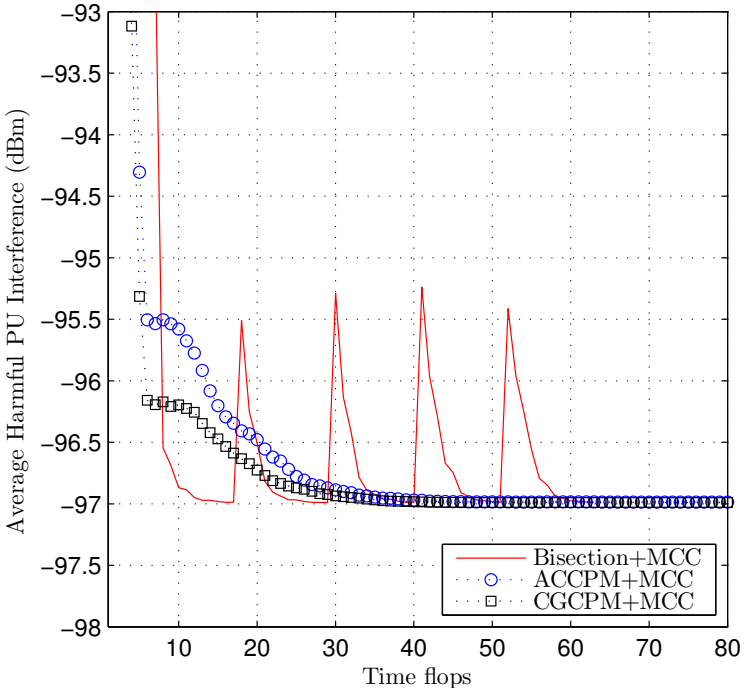

FIGURE 4.8: $I_{harm,av}$ progress vs time using MCC feedback

advantage of the MCC feedback instead of the binary one causes less interference and conduces to faster convergence. Secondly, it is observed that the CPM-based methods

reach the learning objective faster than our benchmark AL method and that in the cases of both binary and MCC feedback the CGCPM-based scheme converges to the PU interference threshold limit faster than the ACCPM-based and induces less harmful interference to the PU. Lastly, the combination of probing method and feedback which is optimal in terms of protecting the PU is the CGCPM-based method with MCC feedback.

Additionally, we need to examine how well all the methods maximize the CRN capacity while learning the interference channel gain vector, $\mathbf{h}$. Similarly with the previous metric, we define the average CRN capacity over the 100 random SU topologies as:

$$U_{SU,av}^{tot}(t) = E\left[U_{SU}^{tot}\right] \tag{4.27}$$

and study its progress in time for binary feedback in Fig. 4.9 and for MCC feedback in Fig. 4.10. The last diagrams of the 5 SU static scenario depict this parameter. The

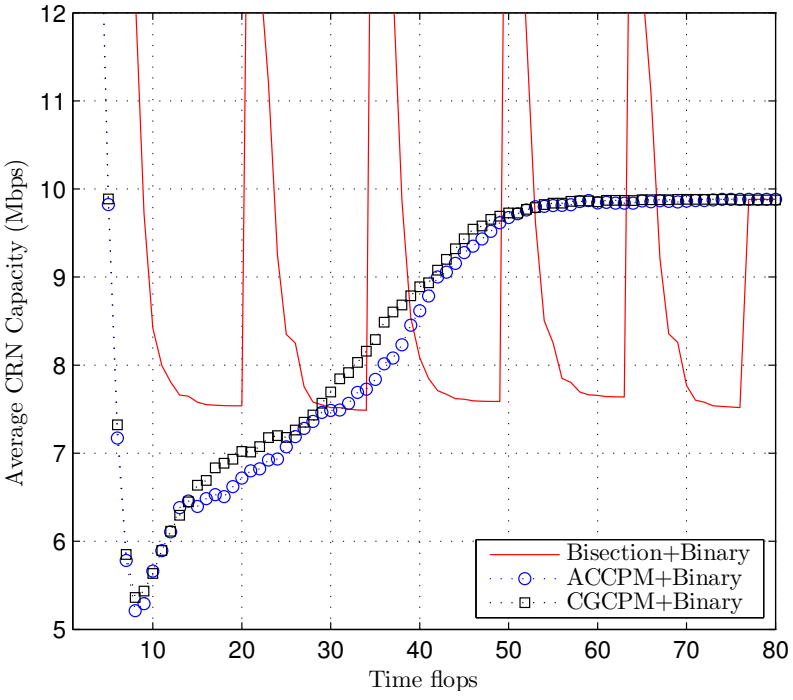

FIGURE 4.9: $U_{SU,av}^{tot}$ vs time using binary feedback

results of the average CRN capacity in Fig. 4.9 and Fig. 4.10 initially show, as stated before, the benefit of using the MCC feedback. Specifically, it can be clearly observed that the maximum value of $U_{SU,av}^{tot}$ is achieved faster in the MCC feedback case by 10 time flops. Again, the CGCPM-based method because of its better learning rate, switches earlier to the capacity maximization problem and therefore performs marginally better the the ACCPM-based one both in Fig. 4.9 and Fig. 4.10. Finally, we need to comment that ous simple AL benchmark method, which only focuses on learning $\mathbf{h}$, pursues the

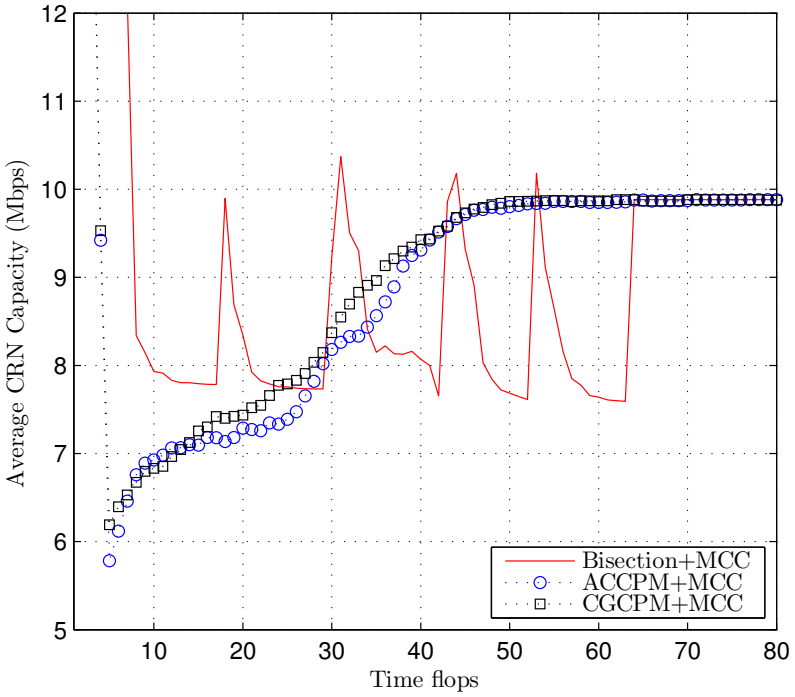

FIGURE 4.10: $U^{tot}_{SU,av}$ vs time using MCC feedback

CRN capacity maximization target only after it reaches the learning solution and not simultaneously.

To clearly show that the CGCPM based method is faster than the ACCPM based one, a fact indicated by the CPM theory about their iteration complexities and mentioned in subsection 5.1, we need to increase the problem dimensions, the number of the SUs. Particularly, these theoretical convergence properties of the CPMs indicate that for an estimation absolute error r the ACCPM-based method needs $\mathcal{O}(\frac{N^2}{r^2})$ probing attempts to learn an interference channel gain vector, $\mathbf{h}$, of N dimensions, while the CGCPM-based method requires $\mathcal{O}(N \log_2(\frac{R}{r}))$ probing attempts for the same purpose. This difference between the necessary probing attempts of the two methods is increased as the CRN grows. The next diagram in Fig. 4.11 is about a static interference channel scenario with $N = 10$ SUs and exhibits the channel estimation error metric for the ACCPM-based and CGCPM-based methods with MCC feedback. Furthermore, the error performances of the same method and feedback combinations for $N = 5$ SUs are shown in the same diagram to validate experimentally that the convergence gain between the ACCPM-based and CGCPM-based methods is increased as the size of the CRN, namely the number of the SUs, N, is increased from $N = 5$ SUs to $N = 10$ SUs.

Specifically, as seen in Fig. 4.11, our variation of the ACCPM, which was used in [75] to enhance the channel correlation matrix learning speed, achieves an estimation error 1% at 95 time flops, while the corresponding CGCPM based algorithm obtains the same

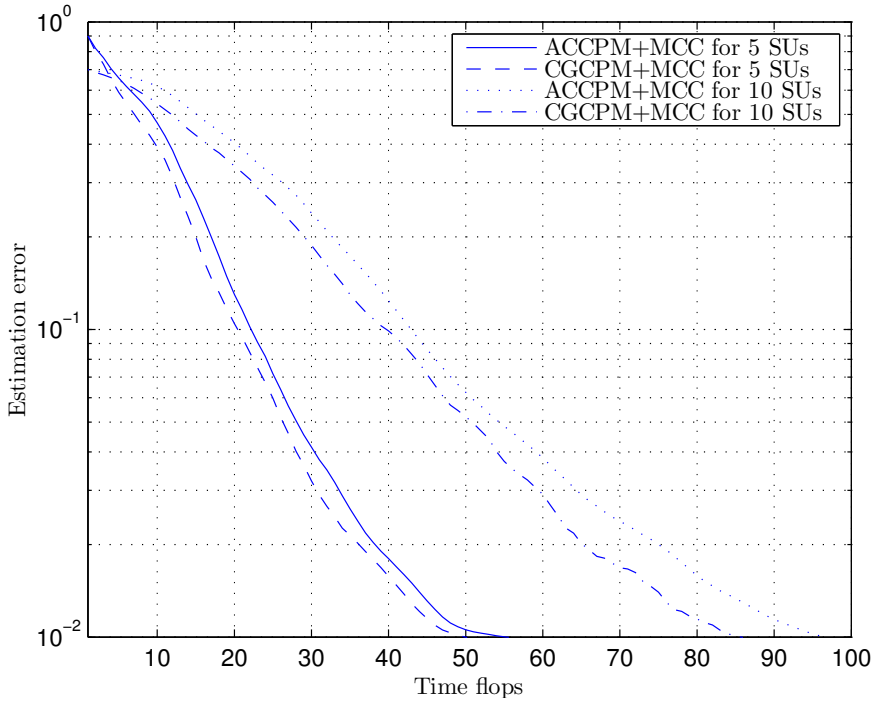

FIGURE 4.11: Interference channel gain vector estimation error progress vs time of CPM based methods and MCC feedback for 5 and 10 SUs

error at 85 time flops. This provides us a convergence gain of 10 time flops which is increased compared to the 5 SU case and of course greater protection to the PU receiver with the CGCPM based method. Nevertheless, this gain in learning speed comes with a penalty. As noted in earlier section, the *Hit and Run* calculation of the CGCPM requires the generation of many random samples within the polytope $\mathcal{P}_t$. The number of these samples grows exponentially with the number of problem dimensions. Hence, in order for the CBS, where the CG computation takes place, to perform this calculation an exponentially increasing computational burden is needed. This means that the larger the CRN a CBS must coordinate, the more computations the CBS needs to perform in order to achieve the fastest convergence possible.

Subsequently, the proposed algorithms are tested for slow fading interference channels where T_c is chosen to be equal to 250 probing and sensing periods, $T_p + T_s$. The corresponding time window based on the empirical rule of $\lfloor \frac{t_c}{N} \rfloor$ for $N = 5$ SUs is $t_w = 50$ inequality pairs and the rest of the algorithm settings remain the same with the fixed channel experiment case. In addition, 100 random SU topology scenarios are generated for a duration of 3 block periods which correspond to 750 probing and sensing periods and where 2 interference channel changes occur. In these experiments the simple AL benchmark method can be no longer used, since it can be only exploited for learning static interference channels, and the binary feedback is not taken into account as it was proven earlier that it is inferior to the multilevel MCC feedback. Consequently, in

this section we compare the performance of the CPM-based methods using the MCC feedback.

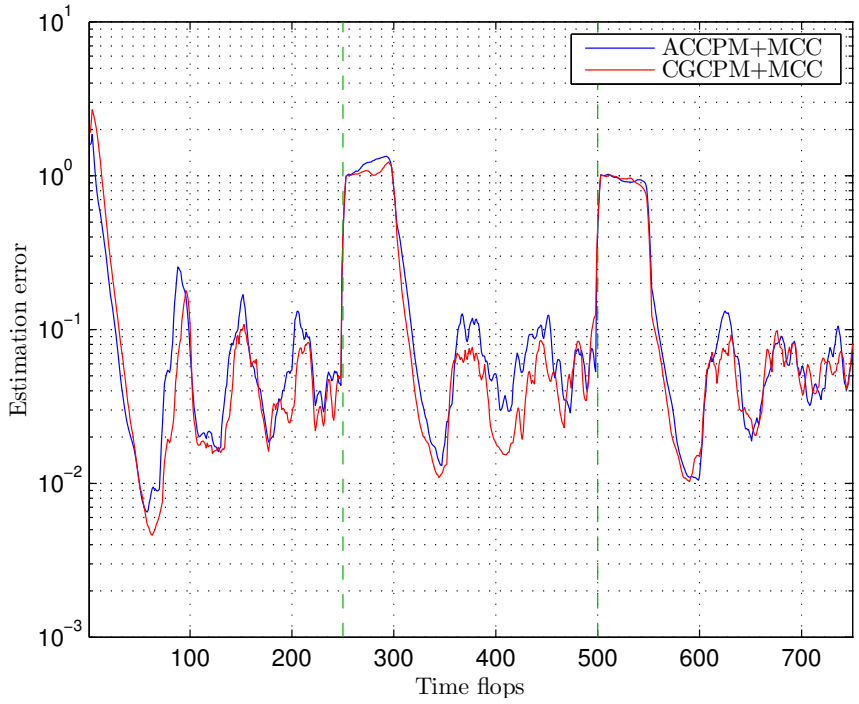

FIGURE 4.12: Interference channel gain vector estimation error progress vs time of CPM methods using MCC feedback for slow fading channels

Once more, the first diagrams concern the learning error of the methods which depict an average of all the random SU topology simulations. In Fig. 4.12, the learning error diagrams show variations, because the learning approach in the dynamic channel scenario is window based and not maximum likelihood based like in [75]. Thus, the results have peaks and valleys instead of being smooth. Nevertheless, the advantage of this approach is that the obsolescence and thus the credibility of each inequality is not dependent any more on the arbitrary probit model and on a forgetting factor whose value choice is impractical. Moreover, the length of the window can be easily distinguished in every channel change where there is a constant average error of almost 100% for 50 time flops. This is because for the learning algorithm to completely "forget" any inequality pair about the previous interference channel vector and proceed to the next one, a number of time flops equal to the observation window is necessary. It can also be observed that between the two CPMs the CGCPM delivers marginally less estimation error with only in one case surpassing the 10% error barrier.

Next, we provide the $I_{harm,av}$ and $U^{tot}_{SU,av}$ diagrams in Fig. 4.13 and Fig. 4.14 respectively. The main advantage observed in these diagrams of the CGCPM-based method over the ACCPM-based one is that despite the number of peaks and valleys which is roughly the same for both techniques, the CGCPM appears to have smaller variations

in both diagrams. This provides better protection to the PU as shown in Fig. 4.13, since it causes less interference to the PU, and closer pursue of the optimization objective, the CRN capacity maximization, as shown in Fig. 4.14. In order to evaluate better the results of the diagrams in Fig. 4.13 and Fig. 4.14, the average $I_{harm,av}$ over time, $\overline{I}_{harm,av}$, and the average CRN capacity over time, $\overline{U}^{tot}_{SU,av}$, are calculated for the 3 blocks and compared to derive further solid performance conclusions besides the convergence rate. For the ACCPM based method, these time average metrics are $\overline{I}_{harm,av} = -95.7$dBm and $\overline{U}^{tot}_{SU,av} = 8.24$Mbps, while for the CGCPM based method they are $\overline{I}_{harm,av} = -96.9$dBm and $\overline{U}^{tot}_{SU,av} = 8.45$Mbps. We notice that the CPM used in this section, the CGCPM, delivers on average 1.2dB less harmful PU interference and 2.5% more CRN capacity compared to the ACCPM used in [75]. Basically, our enhancement contributes to better adaptation and faster learning especially for large CRNs, closer pursue of the optimization objective and most importantly better protection of the PU.

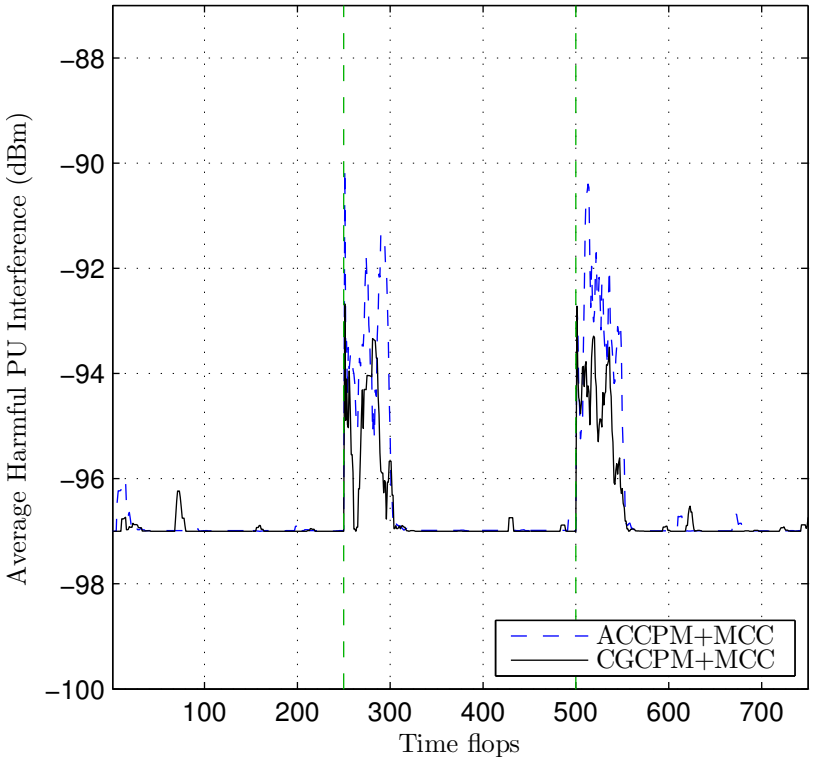

FIGURE 4.13: $I_{harm,av}$ progress vs time using MCC feedback for slow fading channels

4.7 Summary

In this chapter, we proposed a simultaneous PC and interference channel learning algorithm using the MCC feedback. This sensing output is more informative than the binary

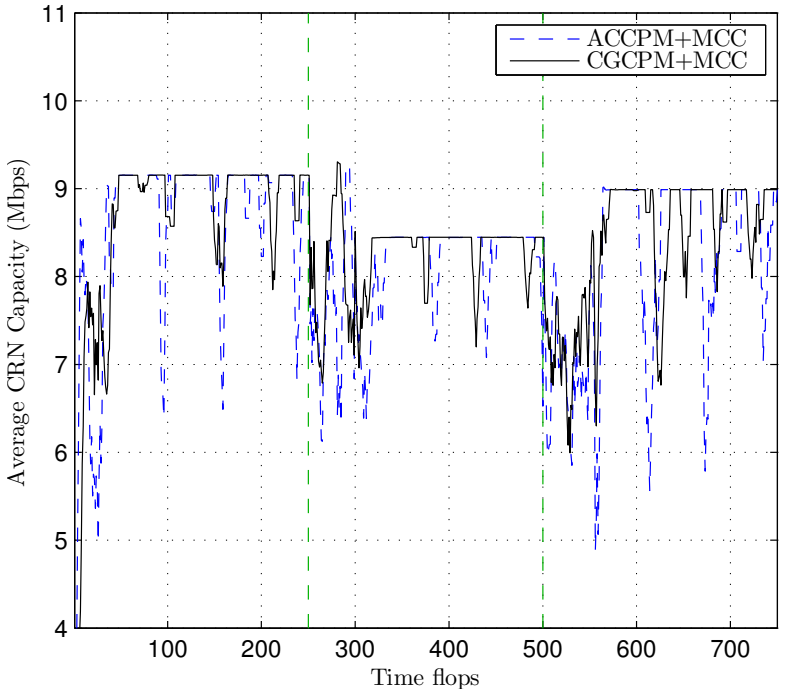

FIGURE 4.14: $U^{tot}_{SU,av}$ vs time using MCC feedback for slow fading channels

ACK/NACK feedback and easier to obtain, since it does not require the implementation of an actual PU decoder on the SU sensing module. The proposed technique was applied in a CR scenario where a CRN with centralized structure access the frequency band of a PU operating under an ACM protocol and learns the unknown interference channels while maximizing its total capacity. New methods from the AL research area, the CPMs, were utilized for the design of the algorithm and compared to a simple AL benchmark learning method we initially demonstrated in this chapter and published in [11]. The chosen CPMs were the ACCPM and the CGCPM inspired by the cognitive beamforming mechanism developed in [75]. Additionally, a window-based solution was introduced for the case of slow fading interference channels. Initially, the results prove the superiority of the MCC feedback whose use provides us an implicit CSI of the PU link more informative than the binary feedback and thus delivers faster convergence. Subsequently, a comparison of the methods was performed which points out the better learning rate of the CPMs to our benchmark method and the small but yet distinguishable, especially in large CRNs, difference between the CGCPM-based approach and the ACCPM-based one, our second benchmark technique. The CGCPM-based algorithm manages to be faster in static interference channel scenarios, more adaptive, more protective to the PU and with less variations in dynamic interference channel scenarios due to its more intelligent choice of probing power vectors.

An extension of this work will be the probabilistic version of the proposed algorithm which takes into account how accurate the output of the MCC process is by utilizing a

reliability factor for each feedback. This will be demonstrated in the next chapter and even though this issue was addressed using a maximum likelihood approach in [75], the proposed solution was not consistent in the AL framework, since no convergence proof was provided.

Chapter 5

Active Learning of the Interference Constraint with Uncertain ACK/NACK Feedback in Cognitive Radio Networks

In this chapter, intelligent probing methods for interference constraint learning are proposed to allow a centralized CRN access the frequency band of a PU in an underlay cognitive communication scenario. The main idea is that the CRN probes the PU and subsequently eavesdrops the reverse PU link to acquire the binary ACK/NACK packet. This feedback is implicit CSI of the PU link, indicating whether the probing-induced interference is harmful or not. The intelligence of this sequential probing process lies in the selection of the power levels of the SUs which aims to minimize the number of probing attempts, a clearly AL procedure, and expectantly the overall PU QoS degradation. The enhancement introduced herein is that we incorporate the probability of each feedback being correct into these intelligent probing mechanisms by using Bayesian AL methods. First, a simple Bayesian AL technique based on the Probabilistic Bisection Algorithm (PBA) is demonstrated for probing the PU system and learning the interference channel gains using the observed PU ACK/NACK feedback and its probability of being correctly decoded. Next, sophisticated multivariate Bayesian AL methods are presented which introduce the deterministic CPMs into this Bayesian framework. Among these, one method is proven optimal and its effectiveness is demonstrated through numerical simulations. The rest of these multivariate Bayesian AL solutions are suboptimal and relate to prior AL work from the literature and computationally cheap CPM adaptations

which still outperform existing AL methods from other authors and are ideal fo high dimensional scenarios. The content of this chapter is published in [12, 13].

5.1 Introduction

In this chapter, we focus on an underlay CR scenario, where SUs may transmit in a PU frequency band as long as the induced to the PU receiver interference is under a certain threshold. In general, the underlay approach is related to Power Control (PC) or Beamforming (BF) problems where the CR users must intelligently select their transmit power levels or beamforming vectors to optimize some operation metric and satisfy the PU interference constraint. In all these scenarios, an architecture suggestion for the deployed CRN could be the CR users to be coordinated by a Cognitive Base Station (CBS) using a dedicated control channel [6] which denotes a centralized structure and is more applicable than a decentralized CRN where CR users are partially independent and pass messages among each other.

An essential piece of information of these PC or BF problems regarding the PU interference constraint is accurate CSI of the interference channels. However, the legacy system (PU) was not originally designed to exchange any information, hence the two networks are not able to directly communicate. This indicates that no feedback about the induced to the PU interference can be transmitted to the CRN (SUs) in order to infer the interference channel gains. Since no cooperation between the PU and SU systems is expected, the CRN must somehow learn this interference CSI once it is deployed. In the CR context though, a common approach to overcome this issue is the CRs to use the PU reverse link feedback, check how this changes because of the CR operation and thus calculate the SU-to-PU channel gains in a sequential manner. This iterative procedure is clearly a probing scheme which combines carefully selecting the CR transmitting parameters and eavesdropping the PU reverse link feedback. Capturing and exploiting this feedback bridges the gap of PU and SU system segregation and enables learning in the CRN part. In previous work, this was obtained from the binary ACK/NACK packet of the reverse PU link [74, 75] for underlay PC or BF problems.

Another kind of feedback introduced in [9, 10] and presented in the Chapter 4 is the Modulation and Coding Classification (MCC) information. Assuming that the PU link operates under an Adaptive Coding and Modulation (ACM) protocol, whenever the PU link quality deteriorates due to CR induced interference, the PU changes its Modulation and Coding Scheme (MCS) to a more robust one. If the CR is equipped with an MCC module, detecting this MCS transition is feasible and in fact easier than decoding a PU ACK/NACK packet. The most crucial benefit of exploiting the MCC feedback instead of

the ACK/NACK one is that the MCC feedback in contrast with the binary ACK/NACK packet delivers more than 1-bit information and therefore speeds up the learning process as demonstrated in [10]. Nevertheless, in any learning process utilizing either MCC or ACK/NACK feedback there is a crucial issue which rises from classifying the PU signal to an MCS or decoding the PU message to obtain the ACK/NACK feedback. This problem is linked to the low SINR conditions of the sensed PU signal on the CRN side which may occur and make the classification or the decoding unreliable. Therefore, if a learning mechanism which exploits a sensed feedback is to be applied, it should also incorporate the uncertainty of the feedback resulting from realistic SINR conditions.

In this chapter, the MCC approach used in Chapter 4 is not adopted and the binary ACK/NACK packet is chosen as the learning facilitator. The main reason for doing so is that the advantages of the MCC feedback were exhibited in Chapter 4 and also it was clearly shown that any cognitive scenario considering a binary feedback of the PU link quality can perform even better with the multilevel MCC feedback. Therefore, in this chapter we utilize this rudimentary binary feedback, the ACK/NACK packet, in order to focus on developing more sophisticated learning mechanisms.

5.1.1 Contributions

Herein, AL probing methods suitable for centrally organized CRNs are demonstrated which rapidly estimate the interference channel gains from multiple SU transmitters to a PU receiver. The case study assumes that the PU link is operating under a communications protocol where the receiver sends an ACK/NACK packet to the transmitter to acknowledge positively or negatively the receipt of messages. A common practice in the CR regime which is adopted here as well is the CRN to capture this packet from the PU feedback link and exploit it to learn the SU-to-PU channel gains. In this scenario, obtaining this binary feedback takes place in the CBS using a sensing antenna and a PU feedback packet decoder. This piece of information is utilized to implement sequential probing techniques where the SUs constantly adjust their transmit power levels according to CBS directives and monitor whether the ACK/NACK packet changes state. These intelligent probing designs aim to minimize the number of probing attempts so that once the CRN is deployed in the PU system's environment, it may quickly learn the interference channels and then optimize its operation while satisfying the PU interference constraint which depends on the SU-to-PU channel gains.

Nevertheless, when utilizing the binary feedback a practical consideration must be taken into account. Due to low SINR conditions of the sensed PU signal by the CBS sensing antenna, the ACK/NACK packet decoding may be imperfect. Therefore, this feedback

uncertainty which is expressed quantitatively by the probability of correct decoding (P_{cd}) must also be included in an AL probing mechanism. In this chapter, a simple Bayesian AL mechanism based on the univariate PBA and multivariate Bayesian AL methods which combine the Bayesian rationale of the PBA and the properties of the deterministic CPMs are implemented in order to include P_{cd} within this AL framework.

In summary, this chapter delivers specifically the following major contributions:

- The design of a simple Bayesian AL scheme based on the PBA and which is a Bayesian extension of our AL solution demonstrated in Chapter 4.
- The novel construction of an optimal multivariate Bayesian AL method designed for probing the PU and learning fast interference channel gains.
- An optimality proof is provided for the proposed multivariate Bayesian AL method.
- A computationally cheap and analytical CPM adaptation is given as a Bayesian AL technique suitable for high dimensional problems.
- Simulations show convergence rates for our optimal multivariate Bayesian AL method and the cheap CPM adaptation faster than the ones of the simple Bayesian AL method developed first and published in [12] and the Probit Maximum Likelihood (ML) approach of [75] which is used as a benchmark technique.
- Results are given about the PU QoS degradation during all the examined AL methods in order to empirically prove that the faster an AL method is, the more protective it is to the PU link.

5.1.2 Structure

The remainder of this chapter is structured as follows: Section 2 reviews in detail prior work related to cognitive scenarios using an uncertain PU link feedback. Section 3 provides the system model and the problem formulation. Section 4 presents a simple Bayesian AL scheme we developed based on the PBA. Section 5 presents a set of multivariate Bayesian AL methods for interference channel gain learning and studies the optimality of the proposed technique. Section 6 elaborates on computational techniques necessary for the implementation of multivariate Bayesian AL. In Section 7, the simulation results which are obtained from the application of all the presented techniques are shown and compared with existing methods. Section 8 gives the concluding remarks.

5.2 Related work

Previous works in the field of cognitive underlay methods using rudimentary feedback have focused on PC and BF scenarios with different assumptions, protocols, system models and constraints. Assuming no cooperation between the CRN and the PU system, this feedback is acquired by most commonly eavesdropping the PU reverse link channel and decoding the PU ACK/NACK packet or by sensing the PU signal and applying MCC in order to track the PU MCS change [10]. The general form of these underlay CR scenarios is the optimization of an SU system metric, such as total CRN throughput, worst SU throughput or SU SINR, subject to QoS constraints for PUs, like SINR, data rate or outage probability [5] which the CRN needs to learn. Hence, these study cases involve both learning PU constraints and solving an optimization problem which may be tackled in a centralized manner by a central decision maker or in a decentralized way by each SU individually. Most of the learning techniques are based on a simple iterative scheme of probing the PU system and acquiring the feedback indicating how the PU operation is changed. An additional discrimination of these problems is based on the reliability of the feedback itself which in many cases is questionable and introduces uncertainty into the learning and optimization procedures.

First, we describe the group of these scenarios which incorporate perfect feedback knowledge. A stochastic approximation algorithm is exploited in [81] for distributed BF which exhibits slow convergence rate as it basically is a random exploration technique. In [74], the one-bit null space learning algorithm (OBNSLA) is developed, which essentially is a blind realization of the Cyclic Jacobi Technique, in order to learn the null space of the interference channel matrix in a MIMO underlay cognitive scenario. Finally, a sign algorithm is established by the authors of [80] for transmit BF using 1-bit feedback to coarsely update the antenna weights in an LMS-like manner. It is worth noting that in the latter work, the case of feedback error is discussed but not addressed extensively.

In the uncertain feedback problems studied by the research community, the notable work in [92] has considered the centralized weighted sum-rate maximization topic under average SU power and probabilistic PU interference constraints. In this study, the optimization objective is achieved only after the interference channel gain learning process is terminated, a very common tactic for handling the aforementioned learning and optimization general structure of these problems. In its learning part, the recursive Bayesian estimation is employed by using imperfect CSI feedback which may potentially be as elementary as the binary ACK/NACK packet. Furthermore, significant work has been conducted in this area by tackling the uncertain feedback within a Partially Observable Markov Decision Process framework [61, 93] where uncertainty is introduced with a

belief factor related to the reliability of the feedback information. In [61], a binary Spectrum Sensing feedback has been used to enable CRs apply a Reinforcement Learning procedure, the Q-Learning, to regulate the aggregated interference to the PU and in [93] a distributed channel admission solution is formulated based on Dynamic Programming, while in their previous work [94] a SU power control policy is also included using the same formulation but without elaborating on the belief factor enhancement. In [95], a methodical overview of all the Reinforcement Learning applications in CRNs based on the Markov Decision Process framework is provided. Additionally, the authors of [75] proposed a CPM based learning algorithm where probing the PU system targets to both learning interference channel matrices and maximizing the SNR at the SU receiver side in an underlay cognitive BF scenario. In this work, the feedback error follows a Gaussian Cumulative Distribution Function (CDF) model and a Maximum Likelihood AL approach is proposed, but without any theoretical convergence guarantees.

In this chapter, we rely on the AL rationale of [75] applied in the underlay PC problem by using the ACK/NACK feedback and we focus only on learning the unknown interference channel gains without optimizing any SU system metric. The investigated scenario considers a centralized CRN where SUs are coordinated by a CBS using a dedicated control channel which usually as a structure exhibits faster learning and adaptation rates than the decentralized approach. First, a simple Bayesian AL method is described based on the PBA and which is also used later on in this chapter as a benchmark method. Next, we develop various multivariate Bayesian AL schemes among which one is optimal in the AL sense and another is suboptimal but ideal for high dimensional problems. At the end, we provide results to compare all these methods with the benchmark techniques of [12, 75].

5.3 System Model and Problem Formulation

In this section, we describe the system model which resembles the one assumed in Chapter 4. Initially, let us consider a PU link and N SU links existing in the same frequency band as shown in Fig. 5.1. Furthermore, a Frequency Division Multiple Access (FDMA) method allows SU links to operate in separate sub-bands of the PU frequency band and without interfering with each other, but still aggregately causing interference to the PU system. The structure of the CRN is again a centralized one where the SUs are dictated their operational parameters and coordinated by the CBS using a dedicated control channel. The examined scenarios in this chapter are considering the PU, the sensing and the unknown interference channels to follow the quasi static block fading model which applies for telecommunication links such as the satellite or the

backhauling ones, but not for mobile ones where channels change rapidly. Also in this chapter, we focus on channel power gains g which we agreed on referring to as channel gains.

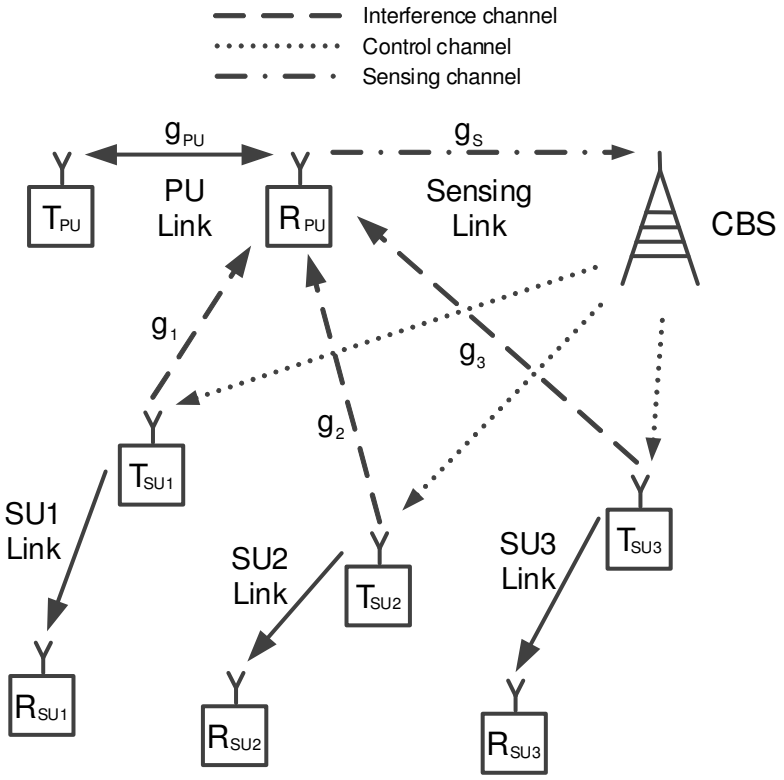

FIGURE 5.1: The PU system and the CRN

As far as the interference to the PU link is concerned, this is caused by the transmitter part of each SU link to the receiver of the PU link similarly to Chapter 4. Taking into account that the SU links transmit solely in the PU frequency band, the aggregated interference on the PU side is defined as:

$$I_{PU} = \mathbf{g}\ \mathbf{p}^{\intercal} \tag{5.1}$$

where $\mathbf{g}$ is the unknown interference channel gain vector $[g_1, ..., g_N]$ with g_i being the SU_i-to-PU interference channel gain and $\mathbf{p}$ is the SU power vector $[p_1, ..., p_N]$ with p_i being the SU_i transmit power. The SU power levels $[p_1, ..., p_N]$ are communicated from the CBS to the SUs through the CRN control channel and they define the messaging overhead of this network. In a practical setting, these transmit power commands from the CBS to the SUs demand $\lceil \log_2(N_{pl}) \rceil$ bits for each SU if we assume that the SU power range is discretized to N_{pl} power levels. However, in this chapter as well, SU power levels are considered to be continuous variables. Additionally, the SINR of the

PU is defined as:

$$SINR_{PU} = 10\log_{10}\left(\frac{g_{PU}p_{PU}}{I_{PU}+N_{PU}}\right)\text{dB} \tag{5.2}$$

where g_{PU} is the PU link channel gain, p_{PU} is the PU transmit power and N_{PU} is the PU receiver noise power.

In this study, we consider that the CBS is equipped with a secondary omnidirectional antenna only for sensing the signal of the PU reverse link and a module for decoding the binary ACK/NACK feedback. From this decoding process, the CRN is able to obtain a feedback observation, Z, and infer whether the induced interference to the PU, I_{PU}, is harmful or not for the PU data packet reception by the PU receiver. Assuming that N_{PU} and the received power remain the same at the PU receiver side, the minimum required $SINR_{PU}$, γ, corresponds to a particular unknown maximum allowed I_{PU} value, I_{th}, below which an ACK is sent and over which a NACK is transmitted to the PU transmitter. Subsequently, the observed feedback Z is defined as:

$$Z = \begin{cases} +1 & \text{if } \mathbf{g}\,\mathbf{p}^{\intercal} \leq I_{th} \\ \\ -1 & \text{if } \mathbf{g}\,\mathbf{p}^{\intercal} > I_{th} \end{cases}. \tag{5.3}$$

This piece of information will be exploited in the next section to learn the PU interference constraint determined as:

$$\mathbf{g}\,\mathbf{p}^{\intercal} = I_{th}. \tag{5.4}$$

A necessary simplification of the information gained by (5.3) is that the g_i gains normalized to I_{th} are adequate for defining the interference constraint (5.4) similarly to Chapter 4. Therefore, if $\mathbf{h} = \frac{\mathbf{g}}{I_{th}}$, the observed feedback can also be written as:

$$Z = \begin{cases} +1 & \text{if } \mathbf{h}\,\mathbf{p}^{\intercal} \leq 1 \\ \\ -1 & \text{if } \mathbf{h}\,\mathbf{p}^{\intercal} > 1 \end{cases} \tag{5.5}$$

while the normalized version of (5.4) is expressed as:

$$\mathbf{h}\,\mathbf{p}^{\intercal} = 1. \tag{5.6}$$

Due to the realistic limitation of low SNR sensing channel, the feedback packet has a probability of being correctly decoded, P_{cd}. In most Automatic Repeat reQuest (ARQ) and Hybrid ARQ mechanisms [96] which utilize identical mechanisms of request for retransmission, feedback packets are transmitted in blocks as bits which allows us to derive lower bounded analytical expressions or exact numerical values based on Bit

Error Rate (BER) curves for the P_{cd} of each feedback packet depending on the MCS of the ACK/NACK block and the SNR of the sensed PU signal. Thus the feedback observation, Z, has a probability of being correct and can be expressed as:

$$\Pr[Z \text{ being correct}] = P_{cd} \tag{5.7}$$

and a complementary probability indicating how likely Z is incorrect:

$$\Pr[Z \text{ being incorrect}] = 1 - P_{cd}. \tag{5.8}$$

The main structure of the system model and of the problem formulation have retained the form described in Chapter 4. The only differences are that the sensing SU link is between the CBS and the PU transmitter, the cooperative MCC process is now replaced by an ACK/NACK decoding process and instead of assuming perfect PU feedback, there is now a probability of this feedback being correct. Later on, we investigate AL methods which consider this uncertainty information to infer (5.6) with the least probing attempts possible.

5.4 A Simple Bayesian Active Learning Algorithm for Interference Channel Gain Learning

In this section, a probing method is described for estimating the interference hyperplane defined in (5.6), where the CRN designs a probing power vector, probes the PU and subsequently decodes the PU ACK/NACK packet from the PU reverse link to check whether the probing power vector caused harmful interference or not. The design of the SU power levels is achieved by a computationally cheap Bayesian AL method which takes into account the uncertainty defined in (5.7) and (5.11) in order to find the interference hyperplane with as less probing attempts as possible. To this direction, an adaptation of the PBA [97] for higher dimensions is implemented. We suggest performing PBA searches on N linear segments in order to find N intersection points with the interference hyperplane and hence the hyperplane itself similarly to our initially developed technique in Chapter 4 which is depicted in Fig. 4.2 for 2 dimensions. The only difference between the two schemes is that here we employ a Bayesian updating of the uncertainty region of each intersection point. This work has been published in [12].

In the beginning, we need to define the linear segments on which the consecutive 1-D PBAs are performed. Again, just like in Chapter 4, end points are sought in a random fashion and once these points, $\mathbf{p_i^{end}}$, $i = 1, ..., N$, are found the PBAs can be

performed to detect the intersection points, $\mathbf{p_i^{in}}$, $i = 1, \ldots, N$ and hence the interference hyperplane. When the intersection points of the line segments, $\mathbf{p_i^{in}}$, $i = 1, \ldots, N$, are estimated and the interference hyperplane is defined, the h_i gains can be found using (4.11) from Chapter 4.

We continue with a detailed description of the PBA application on each linear segment. As explained in Chapter 4 the parametric form of any point on a linear segment with arbitrary end points $\mathbf{Q_1}$ and $\mathbf{Q_2}$ is given by $\mathbf{Q}(\theta) = \theta\mathbf{Q_1} + (1-\theta)\mathbf{Q_2}$, where $\theta \in [0, 1]$. Hence, the PBA is basically performed within the θ region $[0, 1]$ to find a θ^*, which corresponds to the intersection point of this segment with the interference hyperplane. Now, let us elaborate more on the feedback likelihood originally described in (5.7) and (5.11). Assuming that a power vector $\mathbf{p}$ which corresponds to a specific $\theta \in [0, 1]$ is applied, then the following holds:

$$\Pr[Z|\theta^*, \theta] = \begin{cases} P_{cd} & \text{if } Z = +1 \text{ and } \theta \leq \theta^* \\ 1 - P_{cd} & \text{if } Z = +1 \text{ and } \theta^* \leq \theta \\ P_{cd} & \text{if } Z = -1 \text{ and } \theta^* \leq \theta \\ 1 - P_{cd} & \text{if } Z = -1 \text{ and } \theta \leq \theta^* \end{cases}. \tag{5.9}$$

This piece of information can be used to update in a recursive Bayesian way the probability density function (pdf) defined on the θ region $[0, 1]$ and which represents our knowledge of where θ^* truly lies. This means that at time step t of the PBA, the applied probing power vector $\mathbf{p}(t)$, or testing point, which corresponds to a $\theta(t)$ can be used to update this pdf, $f_t(\theta = \theta^*)$, and find its posterior, $f_{t+1}(\theta = \theta^*)$, as:

$$f_{t+1}(\theta) = \frac{\Pr[Z_t|\theta^*, \theta(t)]\ f_t(\theta)}{MaL}. \tag{5.10}$$

The denominator term, MaL, is called the marginal likelihood and it is actually a normalization constant which guarantees that the posterior pdf integrates to 1. Usually, it is computed as the integral of the numerator in (5.10) which here is an easy computational task, since it is a 1-D issue. The prior pdf, $f_0(\theta)$, over $[0, 1]$ is considered to be uniform. In order to probe as less times as possible and give an AL sense in this probing process, the PBA advocates that we should design $\mathbf{p}(t+1)$ so that $\theta(t+1)$ is the median of the posterior pdf $f_{t+1}(\theta)$ at each step. An overview of the entire process described in this section is presented in Algo. 3.

Algorithm 3 Interference hyperplane estimation algorithm

Search Ω^N for endpoints, $\mathbf{p_i^{end}}$, $i = 1, ..., N$, to define N linear segments
$t = 0$
for $i = 1, \ldots, N$ **do**
  $t_i = 0$
  Assume a uniform prior pdf $f_{t_i}(\theta)$ over the interval $[0, 1]$ for the i_{th} linear segment, choose its median, design $\mathbf{p}(t)$ as $\mathbf{p}(t) = 1/2\ \mathbf{p_i^{end}}$ and probe the PU system
  Decode ACK/NACK packet and obtain $Z(t)$
  Find $f_{t_i+1}(\theta)$ using the update equation (5.10)
  repeat
    $t = t + 1$ and $t_i = t_i + 1$
    Choose the median θ_m of $f_{t_i}(h)$, design $\mathbf{p}(t) = \theta_m\ \mathbf{p_i^{end}}$ and probe the PU
    Decode ACK/NACK packet and obtain $Z(t)$
    Find $f_{t_i+1}(\theta)$ using the update equation (5.10)
  until $\sigma_{f_{t_i+1}} \leq \epsilon$
  Choose the median θ_m of $f_{t_i+1}(h)$ and define $\mathbf{p_i^{in}} = \theta_m\ \mathbf{p_i^{end}}$
end for
Calculate $\mathbf{h}$ using (4.11)

5.5 Multivariate Bayesian Active Learning Methods

In this chapter, the goal is to design SU probing power vectors, $\mathbf{p}$, using uncertain observations of ACK/NACK feedback, Z, in order to learn as fast as possible the unknown normalized interference channel gain vector, $\mathbf{h}$. From here on the true value of the unknown normalized interference channel gain vector will be denoted as $\mathbf{h}^*$. These unknown parameters define the constraints (5.4) and (5.6) which constitute the PU interference constraint in underlay cognitive scenarios, also referred to as the interference hyperplane. This uncertainty based AL probing idea was first explored as a cognitive BF problem by the authors of [75] who managed to simultaneously learn the null space of an unknown interference channel matrix and maximize the SNR at the SU receiver side. The iterative nature of this proactive probing strategy which is also adopted in Chapter 4 can be employed in this scenario as well as illustrated in Fig. 5.2, where the CRN probes the PU and subsequently monitors the ACK/NACK feedback sent by the PU receiver in order to infer the interference hyperplane.

In general, sequential uncertain pieces of knowledge are incorporated using a Bayesian approach where recursive Bayesian estimation is the main knowledge extraction tool. To this direction, a multivariate Bayesian Learning method is implemented by using the uncertain observations and their corresponding SU probing power vectors. Now, let us assume that following t probing attempts, $\mathbf{p}_{0:(t-1)} = \{\mathbf{p}(0), .., \mathbf{p}(t-1)\}$, the CBS has observed t pieces of ACK/NACK feedback, $Z_{0:(t-1)} = \{Z_0, .., Z_{(t-1)}\}$. To describe in detail the recursive Bayesian updating, first we need to define the feedback conditional likelihood in this process using (5.7) and (5.8) as the probability of Z conditioned on

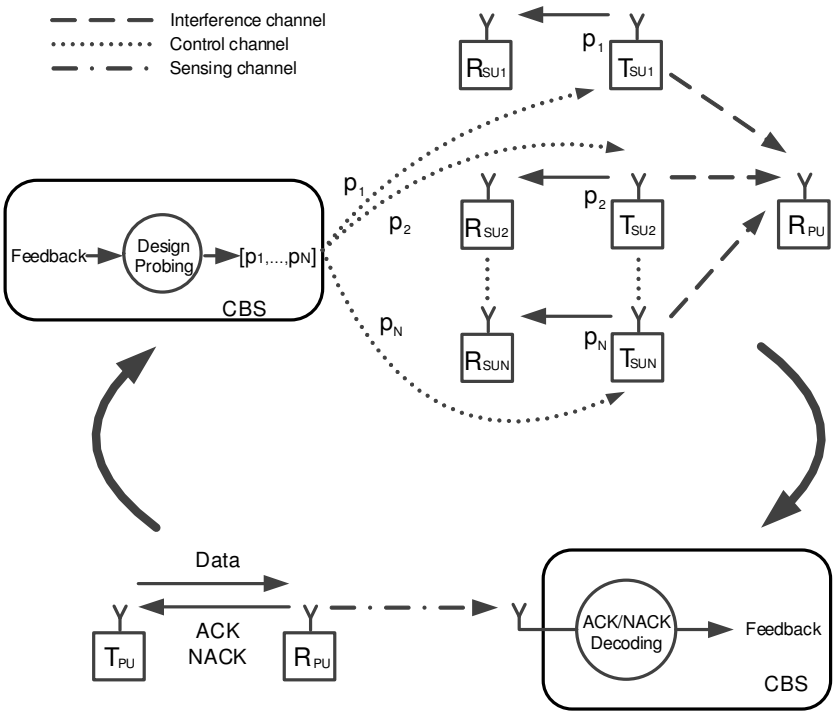

FIGURE 5.2: The Active Learning probing scheme

the unknown parameter $\mathbf{h}^*$ and the probing power vector:

$$\Pr[Z|\mathbf{h}=\mathbf{h}^*\ \mathbf{p}^\intercal] = \begin{cases} P_{cd} & \text{if } Z=+1 \text{ and } \mathbf{h}\ \mathbf{p}^\intercal \leq 1 \\ 1-P_{cd} & \text{if } Z=+1 \text{ and } \mathbf{h}\ \mathbf{p}^\intercal > 1 \\ P_{cd} & \text{if } Z=-1 \text{ and } \mathbf{h}\ \mathbf{p}^\intercal > 1 \\ 1-P_{cd} & \text{if } Z=-1 \text{ and } \mathbf{h}\ \mathbf{p}^\intercal \leq 1 \end{cases}. \tag{5.11}$$

Similarly to the previous section, this expression is actually a robust threshold likelihood metric determined by the uncertainty of the feedback observation, Z.

After a new probing power vector $\mathbf{p}(t)$ and a piece of feedback, Z_t, the $\mathbf{h}$ posterior probability density function (pdf) according to the Bayes rule is expressed as:

$$\begin{aligned} &f_{t+1}(\mathbf{h}) = \Pr[\mathbf{h}=\mathbf{h}^*|Z_{0:t},\mathbf{p}_{0:t}] = \\ &\frac{\Pr[Z_t|\mathbf{h}=\mathbf{h}^*,\mathbf{p}(t),Z_{0:(t-1)},\mathbf{p}_{0:(t-1)}]\ \Pr[\mathbf{h}=\mathbf{h}^*|\mathbf{p}(t),Z_{0:(t-1)},\mathbf{p}_{0:(t-1)}]}{\Pr[Z_t|\mathbf{p}(t),Z_{0:(t-1)},\mathbf{p}_{0:(t-1)}]} \end{aligned} \tag{5.12}$$

which indicates the probability of where $\mathbf{h}^*$ lies in the $\mathbf{h}$ space given $Z_{0:t}$ and $\mathbf{p}_{0:t}$. In (5.12), we also show the equivalence of the $f_{t+1}(\mathbf{h})$ pdf with the conditions $Z_{0:t}$ and $\mathbf{p}_{0:t} = \{\mathbf{p}(0),..,\mathbf{p}(t)\}$ which represent the knowledge gained until the t step. Here, a necessary remark about the first term of the numerator in (5.12) must be made which simplifies (5.12) and which will also help us later. The observation Z_t is conditionally

independent of the previous observations $Z_{0:(t-1)}$ and probing power vectors $\mathbf{p}_{0:(t-1)}$ given $\mathbf{h} = \mathbf{h}^*$ and $\mathbf{p}(t)$ and therefore $\Pr[Z_t|\mathbf{h} = \mathbf{h}^*, \mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]$ can be written as $\Pr[Z_t|\mathbf{h} = \mathbf{h}^*, \mathbf{p}(t)]$ which is basically the likelihood expression in (5.11). Moreover, the second term of the numerator, $\Pr[\mathbf{h} = \mathbf{h}^*|\mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]$, can be written as $\Pr[\mathbf{h} = \mathbf{h}^*|Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]$ which is basically the prior pdf, $f_t(\mathbf{h})$. This happens because our knowledge about $\mathbf{h}^*$ given $Z_{0:(t-1)}$ and $\mathbf{p}_{0:(t-1)}$ does not change by additionally knowing $\mathbf{p}(t)$. After these simplifications the following form of (5.12) is delivered:

$$f_{t+1}(\mathbf{h}) = \frac{\Pr[Z_t|\mathbf{h} = \mathbf{h}^*, \mathbf{p}(t)]\ f_t(\mathbf{h})}{\Pr[Z_t|\mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]}. \tag{5.13}$$

The denominator term is called the marginal likelihood and even though it is difficult to be calculated, it is actually a normalization constant which guarantees that the posterior pdf integrates to 1. Usually, it is computed as the integral of the numerator in (5.13) which in our case is an N dimensional integration over the $\mathbf{h}$ region and computationally intractable. In Section 6, we will explain why its computation is not necessary for the application of the considered AL algorithms. A general assumption when applying recursive Bayesian estimation and employed here as well is the prior pdf $f_0(\mathbf{h})$ to be a uniform non informative pdf [98], which is the maximum entropy pdf for random variables within a bounded domain and therefore guarantees that no specific value of $\mathbf{h}$ is favoured in the beginning of this recursive process.

As stated in the beginning of this section, the target of a sequential AL probing is to select SU power vectors which aim to learn $\mathbf{h}^*$ with the minimum number of probing attempts assuming the coarse likelihood function described in (5.11). The proposed method in this chapter is inspired by a univariate Bayesian AL algorithm, the PBA [97], and the deterministic multivariate AL techniques, the CPMs [85]. In the previous section and in Chapter 4, the effectiveness of both methods has separately been shown in CR learning scenarios with reliable and uncertain PU feedback, but by maintaining their basic form. Here, we suggest a combination in order to formulate a multivariate Bayesian AL technique. In brief, the PBA has proven that assuming a recursive Bayesian updating for estimating a 1-D parameter and an uncertain binary feedback which indicates whether the true value of the 1-D parameter lies right or left of a testing point, the fastest way to learn its value is to always test in the next step the median of the posterior derived from the sequential Bayesian updating. On the other hand, in the deterministic multivariate AL case, the CPMs, which are extensions of the bisection algorithm to higher dimensions, theoretically guarantee that converging strategies to locate a point within a uniform uncertainty region exist and they are implemented by making linear cuts which pass through specific points of the uncertainty region in each step. These points can be the Center of Gravity (CG), the Analytic Center (AC) or

the center of the Minimum Volume Ellipsoid (MVE) covering the region. The resulting CPMs are named respectively CGCPM, ACCPM and MVE-CPM.

5.5.1 The Optimal Cutting Plane in Bayesian Active Learning with Robust Threshold Likelihood functions

In this section, we investigate the optimal design strategy of a SU probing power vector, which represents a hyperplane in the $\mathbf{h}$ space, that should be chosen in each step of this recursive Bayesian estimation process in order to optimally reduce the expected posterior pdf entropy after N_T probing power vectors, or cutting hyperplanes, with their corresponding pieces of feedback, $Z_{0:(N_T-1)}$. To achieve this, we employ Theorem 5.5.1, which is a multivariate extension of the PBA [97], and provide its proof, which is also a multivariate adaptation of the PBA optimality proof, in Appendix A to improve the continuity of this manuscript. Additionally, in Section 6, a numerical approximation is provided for the optimal cutting plane.

Theorem 5.5.1. Given a limited number of N_T probing attempts, pieces of feedback, Z, with conditional likelihood as in (5.11) and a Bayesian updating rule for $f_{t+1}(\mathbf{h})$ as in (5.13), the probing power vector sequence, $\{\mathbf{p}(0), ..., \mathbf{p}(N_T-1)\}$, which achieves the minimum expected entropy of $f_{N_T}(\mathbf{h})$ corresponds to the median regressors of the $\{f_0(\mathbf{h}), ..., f_{N_T-1}(\mathbf{h})\}$ pdf's, $\{\mathbf{p}_{med}(0), ..., \mathbf{p}_{med}(N_T-1)\}$.

Proof. The proof of this theorem can be found in Appendix A. □

5.5.2 Suboptimal Cutting Planes in Bayesian Active Learning

In the previous subsection, Theorem 5.5.1 shows that an AL technique within the Bayesian Learning framework must choose training samples, represented as hyperplanes in the $\mathbf{h}$ space, which cut as evenly as possible the posterior pdf of each step. This reminds us of the CPMs where instead of designing exact cutting hyperplanes, we rely on the geometric properties of specific points of convex polyhedra for which every hyperplane passing from them cuts the polyhedron in two halfspaces whose volumes have a proven lower bound. As far as pdf's are concerned and not just convex bodies, a fundamental theoretical result for partitioning by hyperplanes in N dimensions was first given in [99], where it was proven that for any pdf, there exists at least one point for which every hyperplane passing from it divides the pdf in two parts whose probability masses have a proven lower bound of $1/(N+1)$. Nevertheless, this is merely an existence theoretical result which does not define explicitly this point in the support region of a pdf. An additional theorem proven in [100] states that specifically for a log-concave

multivariate pdf every hyperplane passing from its mean has the property of cutting the pdf in probability masses of at least $1/e$, which is the same theoretical bound for the CGCPM in convex polyhedra [99].

In this subsection, we also consider designing SU probing power vectors which pass through the mean and the mode of each step's posterior pdf, which are equivalents of the CG and the AC respectively. Specifically for the mean case, if $f_t(\mathbf{h})$ is the posterior pdf attained after the $(t-1)$ step and $\mathbf{h}_{mean}(t)$ is the $f_t(\mathbf{h})$ posterior pdf mean which is calculated as:

$$\mathbf{h}_{mean}(t) = \frac{\int \mathbf{h}\ f_t(\mathbf{h})\ dV_{\mathbf{h}}}{\int f_t(\mathbf{h})\ dV_{\mathbf{h}}} \tag{5.14}$$

then the probing power vector $\mathbf{p}(t)$ for which:

$$\mathbf{h}_{mean}(t)\ \mathbf{p}^{\intercal}(t) = 1 \tag{5.15}$$

represents a cutting hyperplane in the $\mathbf{h}$ space which passes through $\mathbf{h}_{mean}(t)$. Alternatively, we examine the performance of an AL method where if the $f_t(\mathbf{h})$ posterior pdf mode or Maximum A Posteriori (MAP) point, $\mathbf{h}_{MAP}(t)$, is defined as:

$$\mathbf{h}_{MAP}(t) = \arg\max_{\mathbf{h}} \left(f_t(\mathbf{h})\right) \tag{5.16}$$

then the $\mathbf{p}(t)$ for which:

$$\mathbf{h}_{MAP}(t)\ \mathbf{p}^{\intercal}(t) = 1 \tag{5.17}$$

describes a cutting hyperplane which passes through $\mathbf{h}_{MAP}(t)$.

Finally, we demonstrate the Shallow-cut deterministic MVE-CPM which is suitable for uncertain pieces of feedback [101]. The standard MVE-CPM is basically an ellipsoidal approximation of the uncertainty region of the true value of $\mathbf{h}$, $\mathbf{h}^*$. If $\mathbf{h}_{EC}(t)$ is the center of this ellipsoid, then a probing power vector $\mathbf{p}(t)$ for which:

$$\mathbf{h}_{EC}(t)\ \mathbf{p}^{\intercal}(t) = 1 \tag{5.18}$$

defines a cutting plane that passes through $\mathbf{h}_{EC}(t)$ in the $\mathbf{h}$ space. With each such cutting hyperplane and its corresponding feedback, the shape of this ellipsoid, represented by its matrix $\mathbf{P}$, is updated and specifically it shrinks and its center moves towards $\mathbf{h}^*$. The updating equations for $\mathbf{h}_{EC}(t+1)$ and the ellipsoid matrix $\mathbf{P}_{t+1}$ are respectively:

$$\mathbf{h}_{EC}(t+1) = \mathbf{h}_{EC}(t) - \frac{1+N\alpha}{N+1}\ \tilde{\mathbf{p}}(t)\ \mathbf{P}_t \tag{5.19}$$

and

$$\mathbf{P}_{t+1} = \frac{N^2(1-\alpha^2)}{N^2-1}\left(\mathbf{P}_t - \frac{2(1+N\alpha)}{(N+1)(1+\alpha)}\,\mathbf{P}_t\,\tilde{\mathbf{p}}^{\intercal}(t)\,\tilde{\mathbf{p}}(t)\,\mathbf{P}_t\right) \tag{5.20}$$

where $\tilde{\mathbf{p}}$ is a normalized subgradient term given by:

$$\tilde{\mathbf{p}}(t) = \frac{\mathbf{p}(t)}{\sqrt{\mathbf{p}(t)\,\mathbf{P}_t\,\mathbf{p}^{\intercal}(t)}} \tag{5.21}$$

and α is heuristically determined as:

$$\alpha = -2\,\frac{1-P_{cd}}{N}. \tag{5.22}$$

The design of α is basically a simple linear function of P_{cd} which guarantees that in the case of the most uncertain feedback, $P_{cd} = 0.5$, the ellipsoid updated based on (5.19) and (5.20) remains the same and that when $P_{cd} = 1$, (5.19) and (5.20) are identical to the updating equations of the neutral-cut MVE-CPM. Lastly, we must mention as practical considerations about the MVE-CPM that the computational complexity in each step of this method is $\mathcal{O}(N^2)$ and that usually the initial ellipsoid is chosen to represent a circular uncertainty region with an arbitrary center.

Here, we need to point out an important issue in AL which was emphasized in Chapter 4, the necessity of exploration. Reducing the uncertainty for $\mathbf{h}^*$ must be performed by approaching this exact value uniformly from all directions. This means that the training samples in an AL process, in this case the power probing vectors, must be diversified and this can be accomplished by choosing cutting planes of random direction uniformly. Therefore, we need first to define how to uniformly sample a random direction $\boldsymbol{\theta}$, where $\boldsymbol{\theta}$ is a unit vector. This problem is related to the uniform hypersphere point picking which has been tackled by generating N, the hypersphere dimensions, random values according to a 1-D Gaussian distribution with zero mean and arbitrary variance, σ^2, and set each one as variable values of a vector $\boldsymbol{\eta}$:

$$\eta_i \sim \mathcal{N}(0,\,\sigma^2),\ i = 1, \ldots, N \tag{5.23}$$

Then, $\boldsymbol{\theta}$ is produced by normalizing $\boldsymbol{\eta}$ to its magnitude, $\boldsymbol{\theta} = \frac{\boldsymbol{\eta}}{\|\boldsymbol{\eta}\|}$. The endpoint of the resulting unit vector $\boldsymbol{\theta}$ is uniformly distributed on the surface of the unit hypersphere.

Now, particularly for our study, in order to produce a power vector which represents a cutting hyperplane of random direction, $\mathbf{p}(t)$ must be parallel to a randomly generated $\boldsymbol{\theta}$, $\mathbf{p}(t) = \beta\boldsymbol{\theta}$ where $\beta \in \mathbb{R}$, and according to the CPM used, it must satisfy (5.15), (5.17)

or (5.18). After some processing, $\mathbf{p}(t)$ is expressed for the Mean CPM as:

$$\mathbf{p}(t) = \beta\boldsymbol{\theta} = \frac{\boldsymbol{\theta}}{\mathbf{h}_{mean}(t)\ \boldsymbol{\theta}^{\intercal}} \tag{5.24}$$

for the MAP CPM as:

$$\mathbf{p}(t) = \beta\boldsymbol{\theta} = \frac{\boldsymbol{\theta}}{\mathbf{h}_{MAP}(t)\ \boldsymbol{\theta}^{\intercal}} \tag{5.25}$$

and for the MVE-CPM as:

$$\mathbf{p}(t) = \beta\boldsymbol{\theta} = \frac{\boldsymbol{\theta}}{\mathbf{h}_{EC}(t)\ \boldsymbol{\theta}^{\intercal}}. \tag{5.26}$$

Moreover, all the coordinates of $\mathbf{p}(t)$, which represent power levels, must be non negative, otherwise a new $\boldsymbol{\theta}$ has to be generated until a valid power vector is produced. In the next section, besides presenting numerical approximations of $\mathbf{h}_{mean}(t)$ and $\mathbf{h}_{MAP}(t)$, an exploration strategy will be directly introduced in the numerical estimation of $\mathbf{p}_{med}(t)$.

5.6 Numerical Approximations for Cutting Plane Estimation

A common problem when dealing with analytically intractable multivariate pdf's, as in our case (5.13), is how to estimate the mean, the mode and the median hyperplane of these density functions which are key points of the investigated Bayesian AL methods as shown in the previous section. To tackle this issue, the research community has developed sophisticated sampling methods based on Markov chain random walks, the Markov Chain Monte Carlo (MCMC) techniques [102]. Learning about probability models by simulating them and generating random samples from them has proven to be more efficient than theoretical approximations of the desired distributions, but still more computationally expensive especially when the dimension number of the target multivariate pdf grows.

One of the most commonly used sampling algorithms is the *Hit and Run* algorithm which was first thoroughly elaborated in [103] and has also been applied in Chapter 4. The simplest form of this sampling mechanism is to start from a point $\mathbf{x}_0$ in the support region S of a pdf f, choose uniformly a random direction $\boldsymbol{\theta}_0$, find the linear segment within S which is defined by the line passing through $\mathbf{x}_0$ and having direction $\boldsymbol{\theta}_0$ and compute the conditional density function along this linear segment. Subsequently, perform a 1-D random sampling over the linear segment using the conditional density function in order to find the first point of the random walk, $\mathbf{x}_1$, and repeat this process with starting point $\mathbf{x}_1$ to generate the second one and so on. As far as the practical details of this sampling algorithm are concerned, first, we already defined in Section

5 how to uniformly sample a random direction $\boldsymbol{\theta}$, a problem which is related to the uniform hypersphere point picking. Additionally, the conditional density function along the linear segment of each random point generating step must be determined. Given a point $\mathbf{x}$ and a vector $\boldsymbol{\theta}$, the parametrized expression of a linear segment defined by them and within S is described as $\mathbf{x} + \lambda\boldsymbol{\theta}$, where $\lambda \in [\lambda_l, \lambda_u]$. The conditional density function $\pi(\lambda)$ based on which the 1-D random sampling is performed can be written as:

$$\pi(\lambda) = \frac{f(\mathbf{x} + \lambda\boldsymbol{\theta})}{\int_{\lambda_l}^{\lambda_u} f(\mathbf{x} + \nu\boldsymbol{\theta})\, d\nu}. \tag{5.27}$$

Once an adequate number of random points is produced by the aforementioned random walk process, all the required characteristics of the f pdf can be extracted so that the exhibited cutting planes in Section 5 can be determined. At first, let us examine how the median hyperplane of a pdf can be estimated which according to our previous analysis is optimal in Bayesian AL with robust threshold likelihood functions. Given a set of points $\{\mathbf{x}_1, ..., \mathbf{x}_{N_r}\}$ with real valued weights $\{f(\mathbf{x}_1), ..., f(\mathbf{x}_{N_r})\}$ like in our case due to the MCMC technique, a hyperplane expressed as in (A.1) and which satisfies the condition $C(\mathbf{w}) = 1/2$, where $C(.)$ is defined in Appendix A, is actually called in robust statistics literature a halving hyperplane. The thin difference between a halving and a median hyperplane is that the latter is also a halving hyperplane [104] and moreover it minimizes the sum of the weighted distances of the points from it:

$$\mathbf{w}_{med} = \arg\min_{\mathbf{w}} \left(\sum_{i=1}^{N_r} f(\mathbf{x}_i) \frac{\mid \mathbf{x}_i\, \mathbf{w}^\intercal - 1 \mid}{\|\mathbf{w}\|} \right) \tag{5.28}$$

where N_r is the number of the random points. In our work, we focus on the solution of (5.28) and not just halving hyperplanes, basically because it delivers a unique hyperplane towards a specific direction and not a set of hyperplanes from which we need to select somehow one candidate.

As far as the suboptimal cutting planes are concerned, we are interested in regressors passing from the mean and the mode of a pdf. The mean of f can be computed as:

$$\mathbf{x}_{mean} = \frac{\sum_{i=1}^{N_r} \mathbf{x}_i f(\mathbf{x}_i)}{\sum_{i=1}^{N_r} f(\mathbf{x}_i)} \tag{5.29}$$

while the mode or MAP point can be calculated as:

$$\mathbf{x}_{MAP} = \frac{\sum\limits_{i:f(\mathbf{x}_i)=f_{max}} \mathbf{x}_i f(\mathbf{x}_i)}{\sum\limits_{i:f(\mathbf{x}_i)=f_{max}} f(\mathbf{x}_i)}. \tag{5.30}$$

It is worth noting that the suboptimal MVE-CPM presented previously does not need such numerical techniques to work. The reason is that all its parameters can be analytically calculated without using the computationally expensive MCMC's.

In the linear piecewise optimization problem of (5.28), the median cutting plane solution is unique for hyperplanes of a specific direction. Nevertheless, this does not mean that it is unique for all directions, a fact which can be taken advantage of to introduce exploration in the final solution. An adaptation which can be made is to uniformly sample a random unit vector, $\mathbf{w}_{rand}$, and solve (5.28) for $\mathbf{w} = \kappa \mathbf{w}_{rand}$, where $\kappa \in \mathbb{R}$. Consequently, (5.28) becomes:

$$\kappa_{med} = \arg\min_{\kappa} \left(\sum_{i=1}^{N_r} f(\mathbf{x}_i) \frac{\mid \kappa \ \mathbf{x}_i \ \mathbf{w}_{rand}^{\mathsf{T}} - 1 \mid}{\| \kappa \ \mathbf{w}_{rand} \|} \right) \tag{5.31}$$

and

$$\mathbf{w}_{med} = \kappa_{med} \ \mathbf{w}_{rand}. \tag{5.32}$$

All the aforementioned numerical approximations of the mean, $\mathbf{x}_{mean}$, the mode, $\mathbf{x}_{MAP}$, and the median hyperplane, $\mathbf{w}_{med}$, concern an arbitrary pdf $f(\mathbf{x})$. Respectively, for our Bayesian AL techniques, these approximations will be used in each time step t of the AL procedures in order to estimate $\mathbf{h}_{mean}(t)$, $\mathbf{h}_{MAP}(t)$ and $\mathbf{p}_{med}(t)$ of $f_t(\mathbf{h})$. One last detail about the $\mathbf{p}_{med}(t)$ estimation is that it must have non negative elements, since they denote power levels. Therefore, if (5.31) and (5.32) do not produce a $\mathbf{w}_{med}$ for which $\mathbf{w}_{med} \geq \mathbf{0}$ holds, then a new $\mathbf{w}_{rand}$ must be generated until a valid power vector is delivered. For all these numerical approximations, a critical remark which must be made is that the denominator term or normalization factor of (5.13) can be omitted either because of fraction reduction in (5.27), (5.29), (5.30) or due to redundancy in (5.31). Therefore, we can use the unnormalized version of (5.13) which basically is the product of the collected likelihood functions and the uniform prior pdf instead of the actual f values.

5.7 Results

In this section, we provide simulation results to compare the performance of the optimal and suboptimal Bayesian AL methods presented in Section 5, the method we previously developed in Section 4 and the ML based mechanism of [75], which is used as a benchmark technique. The mean and mode crossing and median AL methods are based on MCMC sampling in the $\mathbf{h}$ space and they are numerical, accurate, but also require a great amount of samples. The benchmark technique can also be considered as a Probit MAP scheme for estimating the MAP point assuming again that the prior pdf is the uniform one. Specifically, the only difference in the Probit MAP technique is that the likelihood function, instead of the (5.11) form, is expressed as:

$$\Pr[Z|\mathbf{h}] = \begin{cases} \Phi\left(\frac{1-\mathbf{h}\,\mathbf{p}^{\intercal}}{\sigma_m}\right) & \text{if } Z=+1 \\ \Phi\left(\frac{\mathbf{h}\,\mathbf{p}^{\intercal}-1}{\sigma_m}\right) & \text{if } Z=-1 \end{cases} \tag{5.33}$$

where $\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{z^2}{2}}dz$ is the standard Gaussian cdf and σ_m is the standard deviation of the likelihood Probit model. Using this model, the MAP estimation becomes a fast and easily solvable convex optimization problem using numerical algorithms, but with less accuracy than the MCMC based MAP calculation.

In subsections 7.2 and 7.3, the figures show the channel estimation error depending on the number of time flops where each time flop is the time period necessary for the CBS to decode the ACK/NACK packet and design the SU probing power vector. The interference channel gain vector estimation error metric at each time flop is defined as the normalized root-square error $\frac{\|\mathbf{h}(t)-\mathbf{h}^*\|}{\|\mathbf{h}^*\|}$ and basically demonstrates the learning efficiency of each method. The error figure results are obtained as the average of the error metric defined earlier over 100 SU random topologies, which deliver 100 random draws of interference channel gain vectors $\mathbf{h}^*$. Subsequently, the figures of subsection 7.4 present the aggregated interference caused to the PU during each AL process demonstrated in this chapter. As all these AL methods progress in time, it is important to examine the degradation of the PU link quality which can be measured as the induced harmful interference. To this direction, we designed a metric which measures the PU interference exceeding I_{th} and is defined as $I_{harm}(t) = H\left(I_{PU}(t) - I_{th}\right) * I_{PU}(t)$, where H is the Heaviside step function. This parameter of harmful interference is also averaged over the 100 SU random topologies to deliver the corresponding average metric $I_{harm,av}(t)$. This metric is a clear indicator of PU protection during the probing process of all the AL schemes. Even though limiting the induced PU interference is not taken into account in any of the presented probing design techniques, useful conclusions can be extracted for PU protection.

In Table 5.1, all the AL methods described in this chapter are collected with the corresponding abbreviations for identifying them in the performance figures and with the equations or references required for their implementation or understanding. Later on, the aforementioned techniques are compared and the resulting diagrams are divided in two categories, the MCMC based and the computationally cheap ones. The computationally cheap techniques are the MVE-CPM based AL method, the one proposed in Section 4 and the benchmark procedure [75]. This second category is expected to have worse learning performance than the first one, but its AL candidates are recommended for learning interference hyperplanes of high dimensions, where MCMC sampling fails.

TABLE 5.1: An overview of the examined AL methods

Method	**Figure Abbreviation**	**Related Equations or References**
MCMC based Median CPM	MCMC Median	(5.31) and (5.32)
MCMC based Mean CPM	MCMC Mean	(5.29) and (5.24)
MCMC based MAP CPM	MCMC MAP	(5.30) and (5.25)
MVE CPM	MVE CPM	(5.19), (5.20), (5.21), (5.22) and (5.26)
Probit based MAP CPM	Probit MAP	[75]
1-D PBA based AL method	1-D PBA	[12]

An algorithmic description of all the Bayesian AL methods provided in this section is presented in Algo. 4 where a limited "budget" of N_T probing attempts is considered.

Algorithm 4 Bayesian Active Learning for interference hyperplane estimation

Assume a uniform $f_0(\mathbf{h})$ or an initial ellipsoid $\{\mathbf{P}_0, \mathbf{h}_{EC}(0)\}$
for $t = 0 : (N_T - 1)$ **do**
1: Design $\mathbf{p}(t)$ using the related equations from Table 5.1 and probe the PU
2: Acquire uncertain observation Z_t
3: Incorporate Z_t to define $f_{t+1}(\mathbf{h}|Z_{0:t})$ or update ellipsoid to $\{\mathbf{P}_{t+1}, \mathbf{h}_{EC}(t+1)\}$
end for
Choose $\mathbf{h}^*$ as $\mathbf{h}_{mean}(N_T)$ or $\mathbf{h}_{EC}(N_T)$

5.7.1 Simulation Parameters

As far as the technical parameters of the simulations are concerned, the PU receiver is chosen to normally operate and acknowledge with ACK packets when interference is below $I_{th} = -97$dBm, a limit unknown to the CRN. The examined scenarios consider $N = 5$ and $N = 10$ SUs which are dispersed uniformly within a 3km range around the PU receiver. The interference channel gains that are unknown to the CRN are assumed

to follow an exponential path loss model $g_i = \frac{1}{d_i^4}$, where d_i is the distance of the SU_i from the PU receiver in metres. The remaining scenario parameters are the maximum transmit power, p_{max_i}, which is set to 23dBm for all SUs, and the probability of the CBS correctly decoding the ACK/NACK packets, P_{cd}, which is selected to take three values $\{0.7,\ 0.8,\ 0.9\}$.

Additionally, a practical consideration which must be taken into account is the necessary number of samples for the MCMC based AL methods to be accurate, which in the MCMC literature is not well defined. Using the MCMC convergence diagnostics method of [105], we have concluded that $N_r = 20000$ is acceptable for median estimation error of 1% in $N = 5$ dimensions. For the mean and mode estimations, the error is around 0.1% for the same N_r. In the case of $N = 10$, the corresponding errors are 1.6% for the median regressor and 0.1% for the mean and the mode with $N_r = 150000$ samples, which are similar to the previous ones. In [105], the estimation of a statistical feature of a pdf using MCMCs is monitored as the sampling proceeds. Particularly, after each sample, a series of cumulative sums of residuals concerning the feature of interest is constructed. Subsequently, its smoothness is analysed in order to decide if the MCMC estimation of this feature has converged and thus presume that it is safe to stop sampling. A comparative review of the most popular MCMC convergence diagnostics tools can be found in [106] where the advantages and disadvantages of each method are explained.

5.7.2 Estimation Performance of MCMC Based AL Methods

Initially, let us see in Fig. 5.3, 5.4 and 5.5 the performance of the first category AL methods for $N = 5$ SUs. Here, it can be clearly seen that as P_{cd} is reduced, the three AL techniques require more time flops, meaning probing attempts, to correctly estimate $\mathbf{h}^*$. Furthermore, the median based AL method outperforms in speed both the mean and MAP based probing schemes. More specifically, in the case of $P_{cd} = 0.9$, Fig. 5.3, for an estimation error 1% the median method achieves convergence in 105 time flops, whereas the corresponding numbers of time flops for the mean and MAP based techniques are 116 and 158 respectively. This convergence gain is also observed for $P_{cd} = 0.8$ in Fig. 5.4 where in 200 time flops the median, mean and MAP based methods have corresponding estimation errors 2.1%, 5.5% and 27% and for $P_{cd} = 0.7$ in Fig. 5.5 where in 500 time flops the respective estimation errors are 1.8%, 2.9% and 23%.

Moreover, to clearly show that the median based method is faster than the mean and the MAP based ones, we need to increase the problem dimensions, the number of the SUs. The observed differences among the necessary probing attempts of the MCMC based methods in order to reach $\mathbf{h}^*$ within some certain error should be increased as the CRN

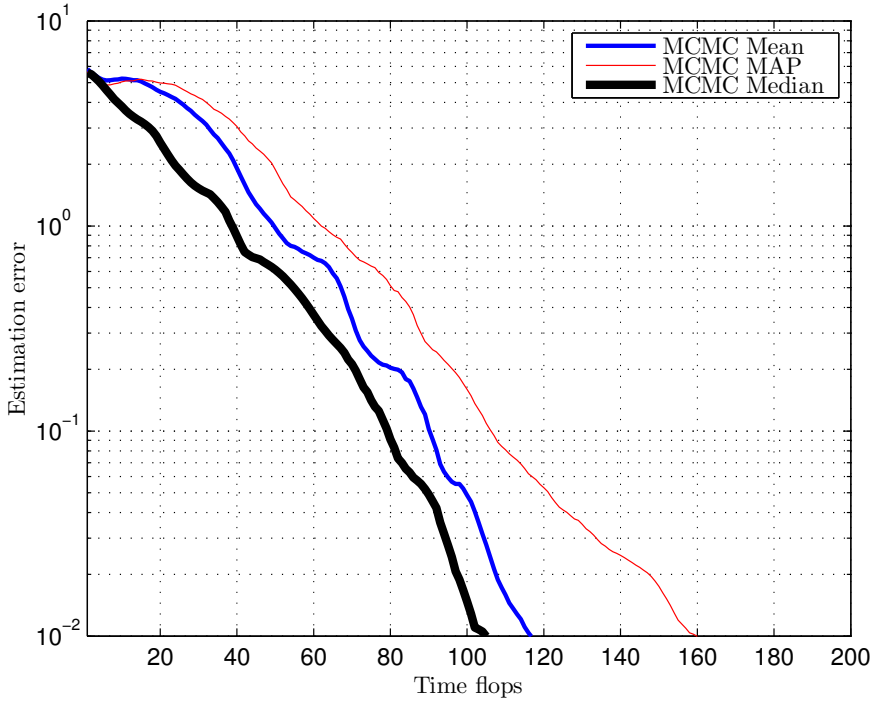

FIGURE 5.3: Interference channel gain vector estimation error progress vs time of the MCMC based AL methods for $P_{cd} = 0.9$ and $N = 5$ SUs

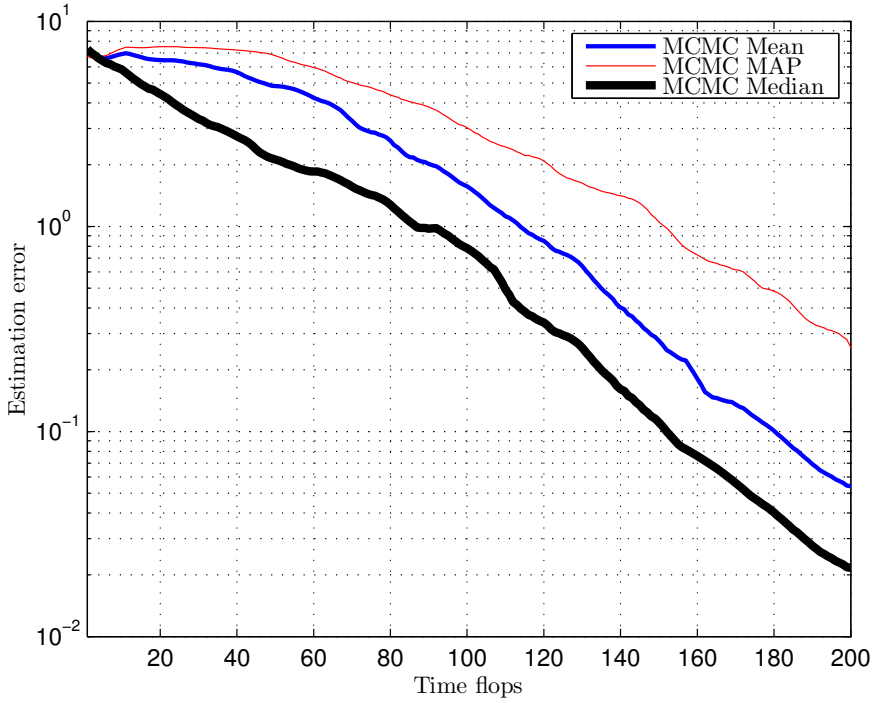

FIGURE 5.4: Interference channel gain vector estimation error progress vs time of the MCMC based AL methods for $P_{cd} = 0.8$ and $N = 5$ SUs

grows. Otherwise, one could argue that there is a N limit beyond which the median based technique is not optimal. The next diagram in Fig. 5.6 shows the estimation

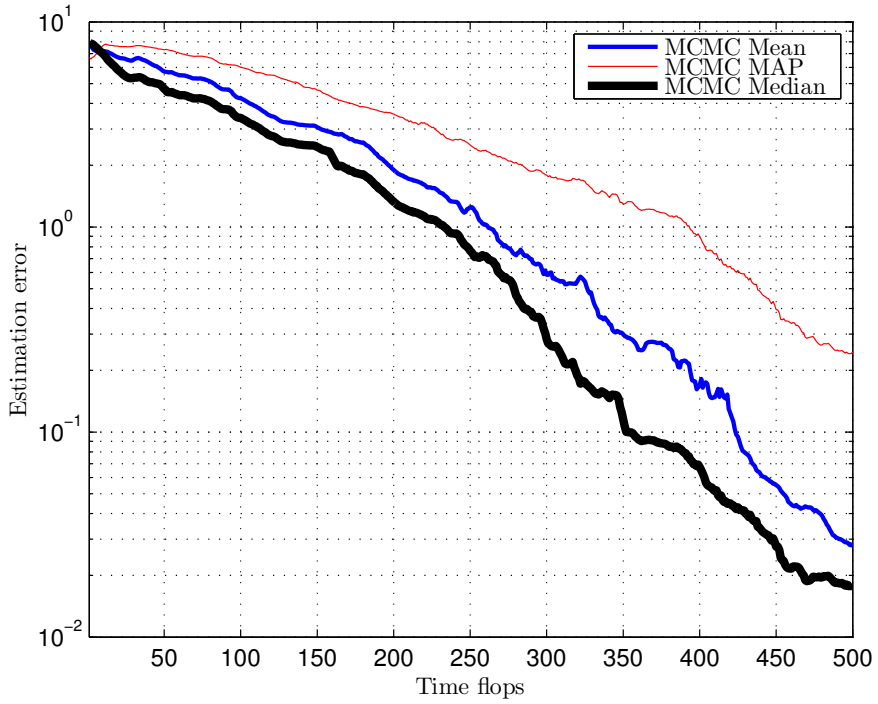

FIGURE 5.5: Interference channel gain vector estimation error progress vs time of the MCMC based AL methods for $P_{cd} = 0.7$ and $N = 5$ SUs

performance of the first group AL methods for $N = 10$ SUs and $P_{cd} = 0.9$.

Compared to the diagram in Fig. 5.3, it is clearly shown that first of all the probing attempts to achieve estimation error of 1% have increased for all methods which is expected to happen, since in the $N = 10$ case $\mathbf{h}^*$ has higher dimensions and therefore more coefficients. Second, the convergence gains among the MCMC based methods have also increased which experimentally validates that the higher the problem dimensions, the larger the performance differences among these methods. Specifically, as seen in Fig. 5.6, the median and mean based AL schemes achieve an estimation error of 1% at 208 and 252 time flops respectively, whereas the MAP based can hardly compete them. This delivers us a convergence gain of 44 time flops between the median and the mean based method for $N = 10$ SUs, while the convergence gain between the same schemes for $N = 10$ SUs is 9 time flops as observed in Fig. 5.3.

However, as mentioned earlier, the high learning speed of the MCMC based probing schemes comes with a penalty. The *Hit and Run* calculation of the median regressors and the mean and MAP points requires the generation of many random samples in the $\mathbf{h}$ space according to the pdf of each step. The number of these samples grows exponentially with the problem dimensions and can be specified by the aforementioned convergence diagnostics tool [105]. Particularly in our scenarios, as we emphasized in subsection 7.1, for the $N = 5$ and $N = 10$ SU cases we used respectively $N_r = 20000$ and

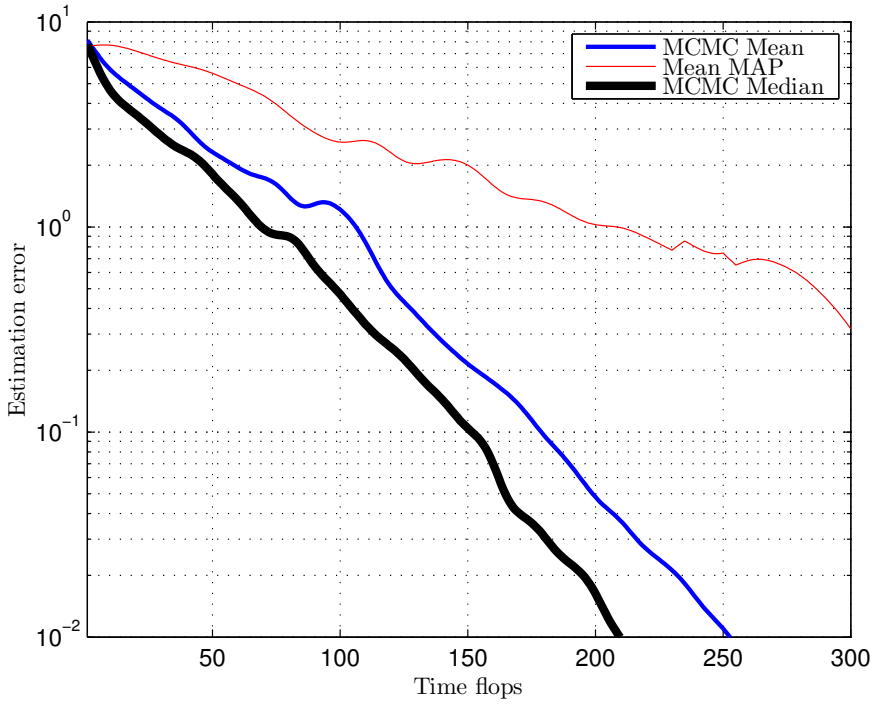

FIGURE 5.6: Interference channel gain vector estimation error progress vs time of the MCMC based AL methods for $P_{cd} = 0.9$ and $N = 10$ SUs

$N_r = 150000$ samples. Therefore, in order for the CBS, where all these computations take place, to design the SU probing power vectors an exponentially increasing to N computational burden is demanded. This means that the larger the CRN a CBS must coordinate, the more computations the CBS needs to perform in order to achieve fast convergence performances.

5.7.3 Estimation Performance of Computationally Cheap AL Methods

To tackle the computational issue due to MCMC usage, we have also tested the performance of the analytical MVE-CPM based AL technique we developed in this chapter, of the Probit MAP scheme in [75] and of the Bayesian AL scheme from Section 4 which is essentially based on Bayesian 1-D grid estimators and thus it is also computationally effective.

At first, as seen in Fig. 5.7 and 5.8 for $P_{cd} = 0.9$ and $P_{cd} = 0.8$ respectively, the AL mechanisms of this second category have worse convergence rate than the previous ones, which was expected. Furthermore, when comparing the last three techniques, it can easily be observed that the suboptimal MVE-CPM based AL method developed in Section 5 converges faster than the other two ones. Particularly, for $P_{cd} = 0.9$ an estimation error of 1% is achieved by the MVE-CPM in 398 time flops, whereas by the

scheme of Section 4 in 452. Similarly, for $P_{cd} = 0.8$ the MVE-CPM learns $\mathbf{h}^*$ in 708 time flops, while the scheme of Section 4 in 765 time flops. The benchmark method in both cases does not perform well and exhibits slow learning rate, basically because it uses an additive Gaussian model for the feedback error resulting in approximating the likelihood functions of (5.11) as Gaussian cdf's, which is a very rough approximation. Here, we must also mention that a reason for testing the MCMC based MAP method was to prove that choosing the MAP point, or at least an almost exact estimation of it through MCMC sampling, for AL purposes is not optimal. Furthermore, below the barrier of $P_{cd} = 0.8$, these techniques do not manage to converge, hence results for $P_{cd} = 0.7$ have not been included.

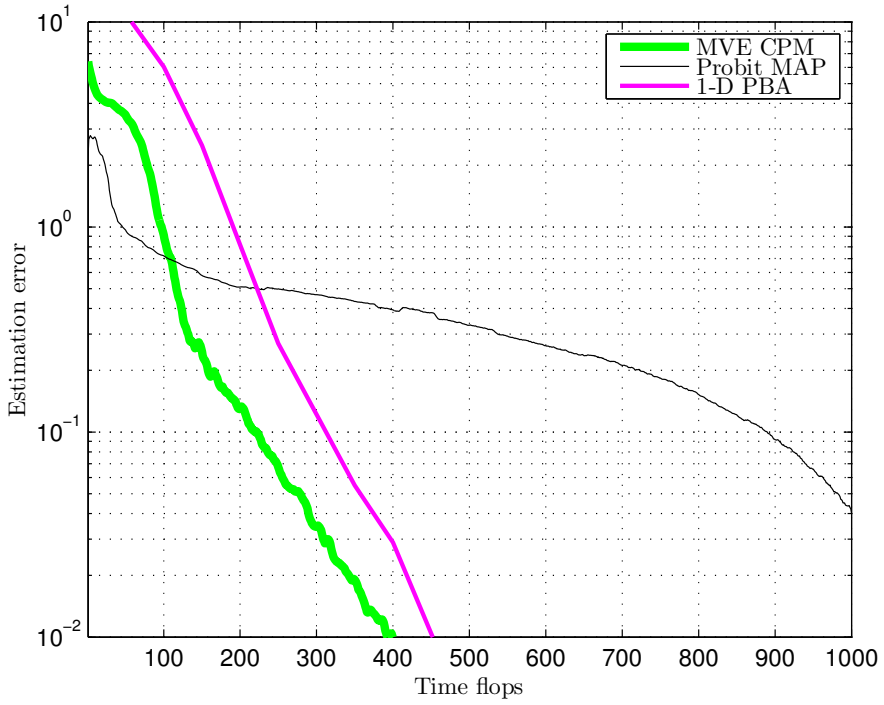

FIGURE 5.7: Interference channel gain vector estimation error progress vs time of the computationally cheap AL methods for $P_{cd} = 0.9$ and $N = 5$ SUs

Finally, in order to confirm that the performance ranking of the computationally cheap AL methods does not change as the problem dimensions grow, the estimation error diagram for $N = 10$ SUs and $P_{cd} = 0.9$ is given in Fig. 5.9. The Probit MAP scheme appears to converge slowly, while the MVE-CPM based AL method and the scheme of Section 4 achieve an estimation error of 2.3% in 1342 and 2000 time flops respectively. Compared to the convergence gain between the two last techniques for $N = 5$ SUs as shown in Fig. 5.7, in this case a greater gain is delivered in favour of the MVE-CPM based AL method. Hence, it can be safely concluded that for this group, the MVE-CPM based scheme is optimal.

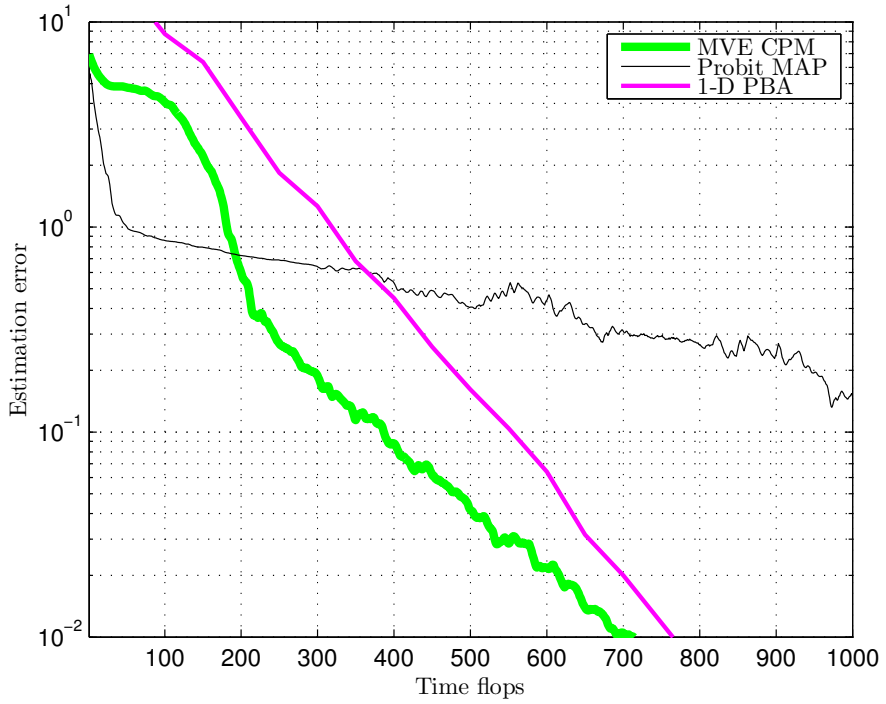

FIGURE 5.8: Interference channel gain vector estimation error progress vs time of the computationally cheap AL methods for $P_{cd} = 0.8$ and $N = 5$ SUs

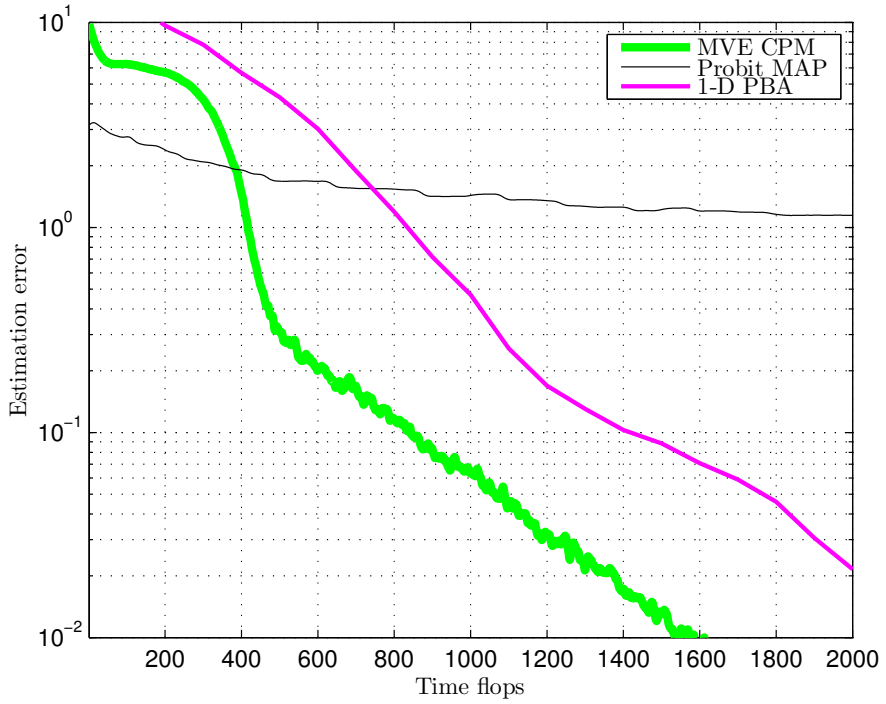

FIGURE 5.9: Interference channel gain vector estimation error progress vs time of the computationally cheap AL methods for $P_{cd} = 0.9$ and $N = 10$ SUs

5.7.4 Interference induced by MCMC Based and Computationally Cheap AL Methods

In this subsection, we provide the $I_{harm,av}$ diagrams for MCMC based and computationally cheap AL methods in Fig. 5.10 and Fig. 5.11 respectively. These diagrams enable us to distinguish which AL methods are optimal in terms of protecting the PU, since an important aspect of all the aforementioned probing schemes is the harmful PU interference caused by each AL process. The results demonstrated here correspond to $P_{cd} = 0.8$, firstly because the $I_{harm,av}$ curves become more distinguishable as the convergence time increases and secondly for the reason that we wish to compare the two method groups, MCMC based and computationally cheap, but below the barrier of $P_{cd} = 0.8$, the computationally cheap techniques do not manage to converge. Additionally, the number of SUs is chosen to be $N = 5$ instead of $N = 10$, because specifically the results in Fig. 5.11 are easier to be discriminated. Hence, choosing $P_{cd} = 0.8$ and $N = 5$ SUs is suitable for acquiring readable results and drawing solid conclusions.

Initially, in Fig. 5.10, we observe that after 200 time flops the mean and median AL schemes induce interference close to the $I_{th} = -97$dBm, but still the median based MCMC AL method provides better protection to the PU, since it causes less interference to the PU through time. As far as the computationally cheap AL methods are concerned, in Fig. 5.11 we first notice that the Probit MAP AL scheme, which is used as a benchmark method, approaches the $I_{th} = -97$dBm slower than the MVE-CPM or the 1-D PBA based AL method and also induces more harmful interference to the PU overall. Moreover, the MVE-CPM compared to the 1-D PBA based method of Section 4 generates less harmful PU interference, without any high interference spikes and with smoother convergence to the $I_{th} = -97$dBm. Conclusively, it can be derived from Fig. 5.10 and Fig. 5.11 that the MVE-CPM is worse than all MCMC based AL schemes, but still comparable in terms of induced harmful interference, and that in general the faster the estimation performance of an AL is, the less interference it causes to the PU. Additionally, the more informative an AL process is, the closer to the PU interference threshold it probes. This indicates that in an AL setting, even though the only probing design metric is the information gain, a probing power vector which delivers more information is also more cautious towards the PU link operation.

5.8 Summary

In this chapter, we proposed probing methods which can be used by a centralized CRN for PU interference constraint fast learning using uncertain ACK/NACK PU feedback.

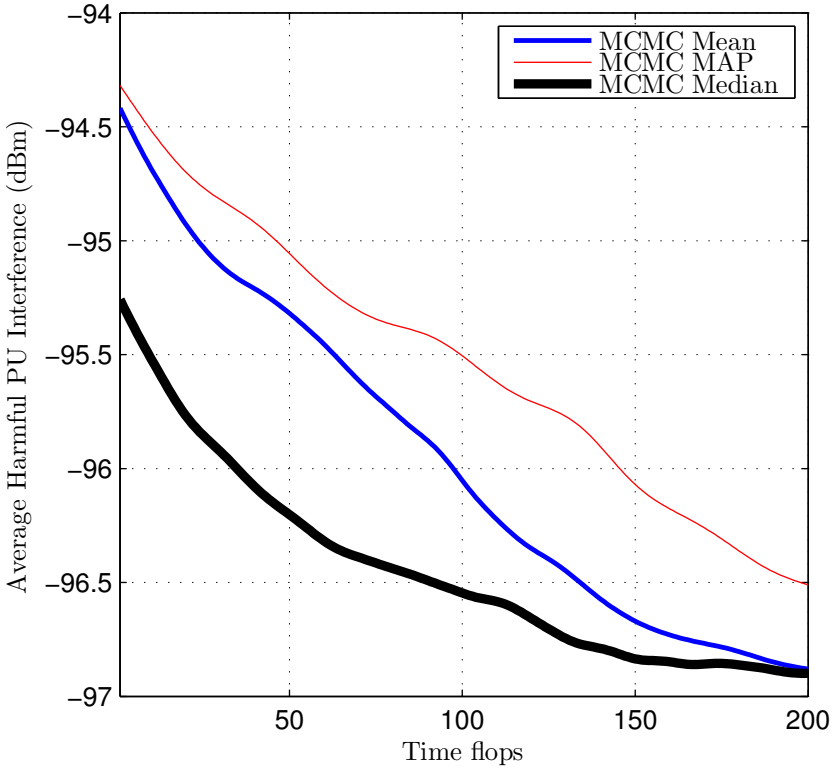

FIGURE 5.10: $I_{harm,av}$ progress vs time of the MCMC based AL methods for $P_{cd} = 0.8$ and $N = 5$ SUs

The proposed techniques were inspired by the deterministic multivariate CPMs and the univariate PBA. The first method we suggest, whose optimality is also proven, is a median based Bayesian AL design of the SU probing power vectors using MCMC sampling and the second one is an MVE-CPM adaptation that is less accurate, but computationally affordable and suitable for large CRN's. The superior performance of these methods compared to existing ones in the AL field [12, 75] was demonstrated through numerical simulations in static channel scenarios for interference channel gain learning. Additionally, results were given for the induced PU interference, which prove that the median based Bayesian AL method and the MVE-CPM adaptation are more protective to the PU among the MCMC based and the computationally cheap AL techniques respectively.

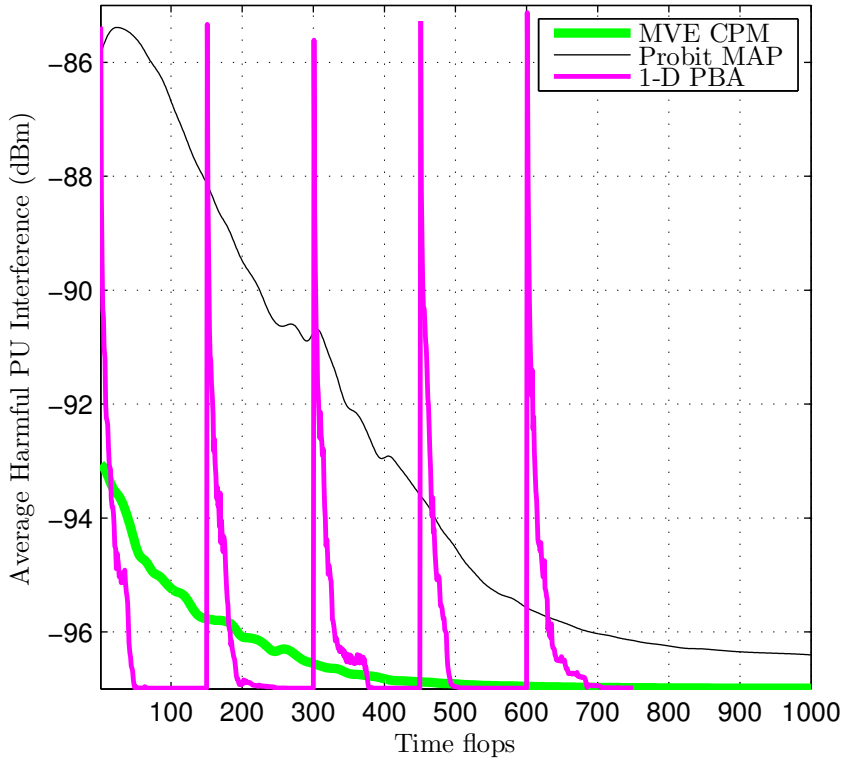

FIGURE 5.11: $I_{harm,av}$ progress vs time of the computationally cheap AL methods for $P_{cd} = 0.8$ and $N = 5$ SUs

Chapter 6

Constrained Bayesian Active Learning of the Interference Constraint in Cognitive Radio Networks

In this chapter, the target and the scenario setting of Chapter 5 is retained but with two important differences. First, the acquired binary ACK/NACK packets are no longer considered uncertain and second the AL process takes into consideration not only gaining as more information as possible over time about the interference channel gains, but also limiting the harmful probing-induced interference events. A provenly optimal solution for this constrained AL problem is obtained and implemented with a sophisticated, accurate and fast Bayesian Learning method, the EP. The effectiveness of this solution is demonstrated through numerical simulations.

6.1 Introduction

In the CR literature, the binary ACK/NACK packet of the reverse PU link has extensively been used as a piece of feedback for estimating PU receiver maps [107], approximating the Lagrange multiplier of the interference constraint in decentralized PC schemes [73] and maximizing or minimizing the power delivered respectively to the SU or PU receiver by adapting the transmit antenna weights in BF scenarios [80]. In the next section, we thoroughly describe other works where a purely learning objective is adopted using this rudimentary piece of feedback. Another kind of feedback introduced

in [9, 10] is the MCC information. Assuming that the PU link operates under an ACM protocol, whenever the PU link quality deteriorates due to CR induced interference, the PU changes its MCS to a more robust one. This PU reaction knowledge can also be taken advantage of for interference channel gain learning purposes. In this chapter though, the binary ACK/NACK feedback is taken into account to facilitate learning on the CRN side.

Furthermore, a practical and convenient architecture design for the CRNs in most of these cognitive scenarios is the CR users to be coordinated by a CBS using a dedicated control channel [6]. This structure is also chosen here and denotes a centralized network setting which is more applicable than a decentralized CRN where CR users are partially independent and pass messages among each other.

6.1.1 Contributions

Herein, a Constrained Bayesian AL probing method suitable for centrally organized CRNs is demonstrated which rapidly estimates the interference channel gains from multiple SU transmitters to a PU receiver while limiting under a threshold the number of harmful probing power vectors over a certain time window. This case study assumes that the PU link is operating under a communications protocol where the receiver sends an ACK/NACK packet to the transmitter to acknowledge positively or negatively the receipt of messages. A common practice in the CR regime which was adopted in Chapter 5 and here as well is the CRN to capture this packet from the PU feedback link and exploit it to learn the SU-to-PU channel gains. In this scenario, obtaining this binary feedback takes place in the CBS using a sensing antenna and a PU feedback packet decoder. This piece of information is utilized to implement a sequential probing technique where the SUs constantly adjust their transmit power levels according to CBS directives and monitor whether the ACK/NACK packet changes state.

This intelligent probing design aims to minimize the number of probing attempts which are needed for learning the SU-to-PU channel gains over a time window subject to maintaining the ratio of the harmful probing attempts under a limit. Hence, once the CRN is deployed in the PU system's environment, it may quickly learn the interference channels without severely degrading the PU communication system and then optimize its operation while satisfying the PU interference constraint which depends on the SU-to-PU channel gains. The introduced constraint in this AL process is of practical significance, because it represents the time ratio during which the PU system cannot efficiently operate which is basically an average over time outage probability constraint, a well defined

design parameter in practical systems. This problem setting is tackled using the Constrained DP framework. Additionally, exactly because this probing process is sequential, the probing vector design must be implemented fast and accurately at each time step. To this direction, an advanced Bayesian Learning, the Expectation Propagation (EP) [108], is implemented analytically for the first time to the authors' knowledge to facilitate the AL goal.

In summary, this chapter delivers specifically the following major contributions:

- The novel construction of a provenly optimal Constrained Bayesian AL method designed for probing the PU and learning fast interference channel gains while maintaining the ratio of harmful probing attempts under a limit.
- A computationally cheap, fast and analytical implementation of a sophisticated and accurate Bayesian Learning technique, the Expectation Propagation, suitable for the sequential probing design nature of our problem.
- Simulations show fast learning convergence rates for our Constrained Bayesian AL method and most importantly adequate satisfaction of the harmful interference constraint.

6.1.2 Structure

The remainder of this chapter is structured as follows: Section II reviews in detail prior work related to cognitive learning scenarios using the ACK/NACK feedback of the PU reverse link. Section III provides the system model and the problem formulation. Section IV presents a fast and accurate Bayesian Learning method, the Expectation Propagation, for interference channel gain learning. Section V elaborates on the optimal Constrained Bayesian AL probing technique for interference channel gain learning. In Section VI, the simulation results obtained from the application of the proposed technique are shown. Section VII gives the concluding remarks and future work in this topic.

6.2 Related work

In the field of cognitive underlay methods, rudimentary PU feedback has been used for learning purposes in PC and BF scenarios with different assumptions, protocols, system models and constraints. Most commonly, this is acquired by eavesdropping the PU reverse link channel and decoding the PU ACK/NACK packet. The general form of these underlay CR scenarios is the optimization of an SU system metric, such as total

CRN throughput, worst SU throughput or SU SINR, subject to QoS constraints for PUs, e.g. SINR, data rate or outage probability [5] whose parameters the CRN needs to learn. Hence, these study cases involve learning PU constraints which may be tackled in a centralized manner by a central decision maker or in a decentralized way by each SU individually. Most of the learning techniques are based on a simple iterative scheme of probing the PU system and acquiring the feedback indicating how the PU operation is affected.

In this group of CR learning works, learning the null space of the interference channel matrix in a MIMO underlay cognitive scenario has been tackled by the one-bit null space learning algorithm [74], which essentially is a blind realization of the Cyclic Jacobi Technique. Furthermore, in [61], a binary Spectrum Sensing feedback has been used to enable CRs apply a Reinforcement Learning procedure, the Q-Learning, to regulate the aggregated interference to the PU. Additionally, in [92], the centralized weighted sum-rate maximization under average SU power and probabilistic PU interference constraints has been considered. In this study, the optimization objective is achieved only after the interference channel gain learning process is terminated, a very common tactic for handling the aforementioned learning and optimization general structure of these problems. In its learning part, the recursive Bayesian estimation is employed by using imperfect CSI feedback which may potentially be as elementary as the binary ACK/NACK packet.

Next, we describe CR learning problems using binary PU feedback which aim at intelligently designing the SU probing attempt in order to learn as fast as possible the unknown constraints of the CR operation, an AL design rationale. Initially, the authors of [75] proposed a Cutting Plane Method based learning algorithm where probing the PU system targets to both learning interference channel matrices and maximizing the SNR at the SU receiver side in an underlay cognitive BF scenario. In the previous chapter, whose content was published in [13], we focused only on learning the unknown interference channel gains without optimizing any SU system metric. We proposed an optimal multivariate Bayesian AL method for intelligent probing which incorporates the probability of each feedback being correct and a suboptimal AL method ideal for CRNs with many SUs.

At this point, we need to specify the broader connections of the AL problem setting which led us to the methodology used in this work. AL is tightly connected to a statistical framework called Bayesian Experimental Design [109] which in its turn is closely related to the theory of optimal Decision Making (DM). Therefore, researchers from the DM field have exploited a DP approach to sequentially design experiments [110]. In [111], the problem of state tracking with active observation control is also tackled in a similar fashion where a Kalman-Like state estimator is developed. Next, AL problems with

constraints were developed by the research community which exploited Constrained DP [48, 112] to actively classify human body states with biometric device sensing costs [113] and to operate a sensor network with communication costs [114]. In this chapter, we combine this Constrained DP framework with a sophisticated Bayesian Learning tool, the EP. Moreover, we enhance the accuracy and the speed of the EP by utilizing recent advances in Statistics from the Econometrics research community [115].

6.3 System Model and Problem Formulation

The system model of this chapter is identical to the one of Chapter 5 with the only difference that acquiring the ACK/NACK packet from the PU reverse link is now accurate. More specifically, a PU link and N SU links exist in the same frequency band as shown in Fig. 6.1. Furthermore, a FDMA method allows SU links to operate in separate sub-bands of the PU frequency band and without interfering with each other, but still aggregately causing interference to the PU system. The structure of the CRN is again a centralized one where the SUs are dictated their operational parameters and coordinated by the CBS using a dedicated control channel. The examined scenarios in this study are considering the PU, the sensing and the unknown interference channels to follow the quasi static block fading model and similar to Chapter 5 we focus on channel power gains g.

As far as the interference to the PU link is concerned, this is caused by the transmitter part of each SU link to the receiver of the PU link. Taking into account that the SU links transmit solely in the PU frequency band, the aggregated interference on the PU side is defined as:

$$I_{PU} = \mathbf{g}\,\mathbf{p}^{\intercal} \tag{6.1}$$

where $\mathbf{g}$ is the unknown interference channel gain vector $[g_1, ..., g_N]$ with g_i being the $\mathrm{SU_i}$-to-PU interference channel gain and $\mathbf{p}$ is the SU power vector $[p_1, ..., p_N]$ with p_i being the $\mathrm{SU_i}$ transmit power. The SU power levels $[p_1, ..., p_N]$ are communicated from the CBS to the SUs through the CRN control channel and they define the messaging overhead of this network. Additionally, the SINR of the PU is defined as:

$$SINR_{PU} = 10\log_{10}\left(\frac{g_{PU}p_{PU}}{I_{PU} + N_{PU}}\right)\mathrm{dB} \tag{6.2}$$

where g_{PU} is the PU link channel gain, p_{PU} is the PU transmit power and N_{PU} is the PU receiver noise power.

In this chapter as well, we consider that the CBS is equipped with a secondary omni-directional antenna only for sensing the signal of the PU reverse link and a module for

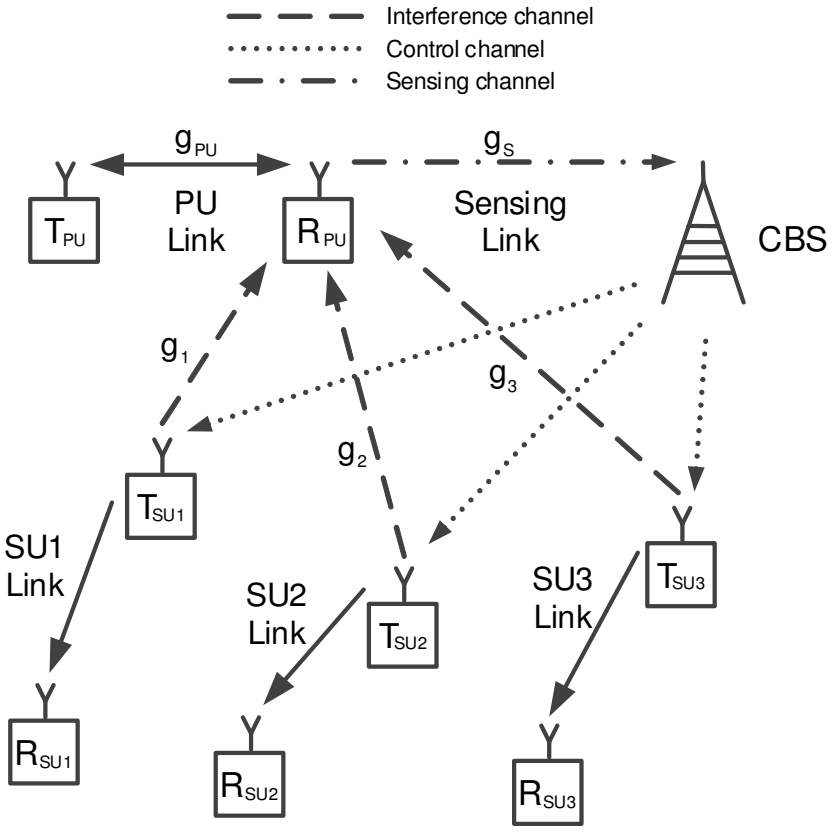

FIGURE 6.1: The PU system and the CRN

decoding the binary ACK/NACK feedback. From this decoding process, the CRN is able to obtain a feedback observation, Z, and infer whether the induced interference to the PU, I_{PU}, is harmful or not for the PU data packet reception by the PU receiver. Assuming that N_{PU} and the received power remain the same at the PU receiver side, the minimum required $SINR_{PU}$, γ, corresponds to a particular unknown maximum allowed I_{PU} value, I_{th}, below which an ACK is sent and over which an NACK is transmitted to the PU transmitter. Subsequently, the observed feedback Z is defined as:

$$Z = \begin{cases} +1 & \text{if } \mathbf{g}\,\mathbf{p}^{\intercal} \leq I_{th} \\ \\ -1 & \text{if } \mathbf{g}\,\mathbf{p}^{\intercal} > I_{th} \end{cases}. \tag{6.3}$$

This piece of information will be exploited in the next sections to learn the PU interference constraint determined as:

$$\mathbf{g}\,\mathbf{p}^{\intercal} \leq I_{th}. \tag{6.4}$$

A necessary simplification of the information gained by (6.3) is that the g_i gains normalized to I_{th} are adequate for defining the interference constraint (6.4). Therefore, if

$\mathbf{h} = \frac{\mathbf{g}}{I_{th}}$, the observed feedback can also be written as:

$$Z = \begin{cases} +1 & \text{if } \mathbf{h}\,\mathbf{p}^{\mathsf{T}} \leq 1 \\ \\ -1 & \text{if } \mathbf{h}\,\mathbf{p}^{\mathsf{T}} > 1 \end{cases} \tag{6.5}$$

while the normalized version of (6.4) is expressed as:

$$\mathbf{h}\,\mathbf{p}^{\mathsf{T}} \leq 1. \tag{6.6}$$

In the next section, we elaborate on a sophisticated and computationally fast Bayesian ML method which exploits the observed feedback of (6.5) to infer (6.6). Later, in section 6.5, we propose a Constrained Bayesian AL method which achieves learning (6.6) using the technique described in section 6.4. The particularity of this Constrained Bayesian AL method is that it designs sequentially the SU probing power vectors in order to learn the PU interference constraint with the least probing attempts possible while maintaining a limited number of probing attempts which cause harmful interference.

6.4 Bayesian Learning using Expectation Propagation

In this section, we present a probabilistic way to learn the unknown normalized interference channel gain vector, $\mathbf{h}$, given a set of SU probing power vectors and the corresponding ACK/NACK pieces of feedback. The true value of the unknown normalized interference channel gain vector will be denoted as $\mathbf{h}^*$ from here on. These unknown parameters define the constraints (6.4) and (6.6) which constitute the PU interference constraint. The data sets of the SU probing power vectors and the ACK/NACK pieces of feedback basically represent the feature vector set and the label set respectively in the ML sense and we demonstrate how to learn the linear classifier, or else the interference hyperplane, denoted by (6.4) and (6.6) in the Bayesian way. The reason for following this Bayesian direction will be clearly revealed in the next section, but let us just state here that deriving a pdf for $\mathbf{h}^*$ will be proven useful for the AL setting of this chapter.

In general, Bayesian ML uses the Bayes rule as the main knowledge extraction tool. To describe in detail the Bayes rule application, first we need to define the feedback, or label, conditional likelihood in this process as the probability of Z conditioned on the

unknown parameter $\mathbf{h}^*$:

$$\Pr[Z|\mathbf{h}=\mathbf{h}^*,\mathbf{p}^\intercal]=\begin{cases}1 & \text{if } Z=+1 \text{ and } \mathbf{h}\,\mathbf{p}^\intercal\leq 1\\ 0 & \text{if } Z=+1 \text{ and } \mathbf{h}\,\mathbf{p}^\intercal>1\\ 1 & \text{if } Z=-1 \text{ and } \mathbf{h}\,\mathbf{p}^\intercal>1\\ 0 & \text{if } Z=-1 \text{ and } \mathbf{h}\,\mathbf{p}^\intercal\leq 1\end{cases}. \tag{6.7}$$

This expression is actually a threshold likelihood metric determined by the feedback observation, Z, and the power vector $\mathbf{p}$. A similar expression was given in Chapter 5 which also included the probability of correctly decoding the binary feedback. We may also describe the likelihood function form based on the "version space duality" introduced by Vapnik [40]. According to this, when we deal with learning linear classifiers, feature vectors are hyperplanes in the parameter or version space and vice versa. Hence, when a learning procedure tries to estimate the parameters of a hyperplane, the version, it actually tries to localize a point in the parameter or version space. In our problem, the feature space corresponds to the power vector space and the version space to the $\mathbf{h}$ space. In addition, by combining a power vector, or feature vector, and its respective piece of ACK/NACK feedback, or label, an inequality is obtained which in the $\mathbf{h}$ space, or version space, represents a linear inequality. Therefore, the likelihood function may also be thought of as a halfspace defined by $\mathbf{p}$ and Z or alternatively as a multivariate form of the Heaviside step function in the version space.

Now, let us assume that following t probing attempts, $\mathbf{p}_{0:(t-1)}=\{\mathbf{p}(0),..,\mathbf{p}(t-1)\}$, the CBS has observed t pieces of ACK/NACK feedback, $Z_{0:(t-1)}=\{Z_0,..,Z_{(t-1)}\}$, which all together constitute the data known until the $(t-1)$ power vector and ACK/NACK feedback pair, D_{t-1}. After a new probing power vector $\mathbf{p}(t)$ and a piece of feedback, Z_t, the $\mathbf{h}$ posterior pdf according to the recursive form of the Bayes rule is expressed as:

$$f_{t+1}(\mathbf{h})=\Pr[\mathbf{h}=\mathbf{h}^*|Z_{0:t},\mathbf{p}_{0:t}]=\Pr[\mathbf{h}=\mathbf{h}^*|D_t]=\frac{\Pr[Z_t|\mathbf{h}=\mathbf{h}^*,\mathbf{p}(t),D_{t-1}]\ \Pr[\mathbf{h}=\mathbf{h}^*|\mathbf{p}(t),D_{t-1}]}{\Pr[Z_t|\mathbf{p}(t),D_{t-1}]} \tag{6.8}$$

which indicates the probability of where $\mathbf{h}^*$ lies in the $\mathbf{h}$ space given D_t. In (6.8), we also show the equivalence of the $f_{t+1}(\mathbf{h})$ pdf with the condition D_t which represents the knowledge gained until the t step. Here, a necessary remark about the first term of the numerator in (6.8) must be made which simplifies (6.8) and which will also help us later. The observation Z_t is conditionally independent of the previous observations $Z_{0:(t-1)}$ and probing power vectors $\mathbf{p}_{0:(t-1)}$ given $\mathbf{h}=\mathbf{h}^*$ and $\mathbf{p}(t)$ and therefore $\Pr[Z_t|\mathbf{h}=\mathbf{h}^*,\mathbf{p}(t),Z_{0:(t-1)},\mathbf{p}_{0:(t-1)}]$ can be written as $\Pr[Z_t|\mathbf{h}=\mathbf{h}^*,\mathbf{p}(t)]$ which is basically the likelihood expression in (6.7). Moreover, the second term of the numerator, $\Pr[\mathbf{h}=\mathbf{h}^*|\mathbf{p}(t),Z_{0:(t-1)},\mathbf{p}_{0:(t-1)}]$, can be written as $\Pr[\mathbf{h}=\mathbf{h}^*|Z_{0:(t-1)},\mathbf{p}_{0:(t-1)}]$ which is basically

the prior pdf, $f_t(\mathbf{h})$. This happens because our knowledge about $\mathbf{h}^*$ given $Z_{0:(t-1)}$ and $\mathbf{p}_{0:(t-1)}$ does not change by additionally knowing $\mathbf{p}(t)$. After these simplifications the following form of (6.8) is delivered:

$$f_{t+1}(\mathbf{h}) = \frac{\Pr[Z_t|\mathbf{h}=\mathbf{h}^*, \mathbf{p}(t)]\ f_t(\mathbf{h})}{\Pr[Z_t|\mathbf{p}(t), D_{t-1}]}. \tag{6.9}$$

The denominator term is called the marginal likelihood and even though it is difficult to be calculated, it is actually a normalization constant which guarantees that the posterior pdf integrates to 1. Usually, it is computed as the integral of the numerator in (6.9) which in our case is an N dimensional integration over the $\mathbf{h}$ region and computationally intractable. A general assumption in Bayesian ML is the prior pdf $f_0(\mathbf{h})$ to be a uniform non informative pdf [98], which is the maximum entropy pdf for random variables within a bounded domain and therefore guarantees that no specific value of $\mathbf{h}$ is favored in the beginning of this learning process. The derived relation (6.9) is identical to the recursive form of the Bayes rule in Chapter 5, but with a different form of likelihood functions.

Alternatively, the posterior pdf expressed in (6.9) can be written in a non-recursive form as:

$$f_{t+1}(\mathbf{h}) = \frac{\prod_{i=0}^{t} \Pr[Z_i|\mathbf{h}=\mathbf{h}^*, \mathbf{p}(i)]}{\prod_{i=0}^{t} \Pr[Z_i|\mathbf{p}(i), D_{i-1}]}\ f_0(\mathbf{h}) \tag{6.10}$$

where again the denominator term is a normalization factor whose computation will be shown unnecessary. The reason we first expressed the posterior pdf in a recursive form is that it will be proven useful in the next section due to the sequential nature of the AL process. Moreover, in Bayesian ML, we should not always take for granted that the posterior pdf is proportional to the *likelihood function product times the prior pdf* which indeed holds for conditionally independent samples. This is the reason why we should always start from decomposing probabilistically our data set in the Bayes rule expression and first derive its recursive form. More importantly, it is necessary for our AL setting, which relates to Bayesian Experimental Design, to show in detail the conditional independences occurring even when training samples, here our power probing vectors, are judiciously designed based on previous training samples and their labels.

Now, let us rewrite (6.10) in a more compact way in order to focus solely on the likelihood function product and thus approximate it using EP [108]. Each likelihood function can be expressed as $l_i(\mathbf{h}) = \Pr[Z_i|\mathbf{h}=\mathbf{h}^*, \mathbf{p}(i)]$ and hence the likelihood function product of (6.10) is now $\prod_{i=0}^{t} l_i(\mathbf{h})$. This product is basically a product of halfspace indicator functions and it defines along with $f_0(\mathbf{h})$ and the denominator term of (6.10), the marginal

likelihood, a uniform pdf with a polyhedral support region. This pdf is not easy to be handled and its statistical properties, like its mean or covariance, are not easily computed. In the previous chapter, this was tackled by using MCMC sampling methods, which are accurate but computationally expensive as the dimensions of the version space increase.

In this section, we show how to approximate $\prod_{i=0}^{t} l_i(\mathbf{h})$ and thus the deriving posterior pdf using EP. The rationale of the EP is to approximate this product by finding an approximation $\tilde{l}_i(\mathbf{h})$ for each $l_i(\mathbf{h})$. This is done by initializing arbitrarily the likelihood function approximations and iteratively filtering each one of them considering the rest approximations stable. This filtration process is based on minimizing the KL divergence of $l_j(\mathbf{h}) \prod_{i=0, i\neq j}^{t} l_i(\mathbf{h})$ and $\tilde{l}_j(\mathbf{h}) \prod_{i=0, i\neq j}^{t} l_i(\mathbf{h})$ and it is performed enough times to ensure that all $\tilde{l}_i(\mathbf{h})$ have been corrected sufficiently so that $\prod_{i=0}^{t} \tilde{l}_i(\mathbf{h})$ approximates $\prod_{i=0}^{t} l_i(\mathbf{h})$ well enough. A detailed algorithmic description of EP is presented in Algo. 5.

Algorithm 5 The Expectation Propagation algorithm

Initialize arbitrarily $\{\tilde{l}_0(\mathbf{h}), \tilde{l}_1(\mathbf{h}), ..., \tilde{l}_t(\mathbf{h})\}$
for $k = 1 : N_{EP}$ **do**
  for $j = 0 : t$ **do**
    $\tilde{l}_j(\mathbf{h}) :=$
    $$\arg\min_{\tilde{l}_j(\mathbf{h})} KL\left(l_j(\mathbf{h}) \prod_{i=0, i\neq j}^{t} \tilde{l}_i(\mathbf{h}) \parallel \tilde{l}_j(\mathbf{h}) \prod_{i=0, i\neq j}^{t} \tilde{l}_i(\mathbf{h}) \right)$$
  end for
end for

Usually, the outer loop iterations of EP, N_{EP}, are chosen to be maximum 5, which is also used in this work. Nevertheless, a more elaborate stopping criterion could be used such as a limit on the KL divergence between the resulting product $\prod_{i=0}^{t} \tilde{l}_i(\mathbf{h})$ of one step of the outer loop and the previous one. In Bayesian ML, this sophisticated iterative filtration for likelihood function approximations has proven to be a very accurate method for approximate inference. However, to the authors' knowledge, all the existing EP approaches rely on numerical quadratures or independence assumptions between the latent variables to facilitate the computations. Next, we describe in more detail the EP implementation and we show how to tackle *analytically* the KL divergence minimization, the critical step of the EP algorithm, without independence assumptions between the latent variables. This will lead to greater accuracy and faster implementation of this sophisticated tool.

So far, an abstract description of the EP algorithm has been given and its basic principles have been explained. In general, each approximation in the EP algorithm is considered to have the form of a multivariate normal pdf, a strategy which is also followed here.

Consequently, the product of multivariate normal pdf's, which appears in the KL divergence minimization step, based on Gaussian identities is also a multivariate normal pdf. More specifically, if $\tilde{l}_i(\mathbf{h}) = \mathcal{N}(\mathbf{h}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ for $i = 0, ..., t$, then their product is an un-normalized multivariate normal pdf proportional to a multivariate normal pdf, $\mathcal{N}(\mathbf{h}; \boldsymbol{\mu}_{tot}, \boldsymbol{\Sigma}_{tot})$, where assuming vectors are row vectors:

$$\boldsymbol{\Sigma}_{tot}^{-1} = \sum_{i=0}^{t} \boldsymbol{\Sigma}_i^{-1} \tag{6.11}$$

and

$$\boldsymbol{\mu}_{tot} = \left(\sum_{i=0}^{t} \boldsymbol{\mu}_i \boldsymbol{\Sigma}_i^{-1} \right) \boldsymbol{\Sigma}_{tot}. \tag{6.12}$$

Hence, the second part of the KL divergence in the core stage of the EP method, $\tilde{l}_j(\mathbf{h}) \prod_{i=0, i\neq j}^{t} \tilde{l}_i(\mathbf{h})$, and the approximation product in the first part, $\prod_{i=0, i\neq j}^{t} \tilde{l}_i(\mathbf{h})$, are basically un-normalized multivariate normal pdf's. For notation simplification, $\prod_{i=0, i\neq j}^{t} \tilde{l}_i(\mathbf{h})$, which is called the *cavity function*, will be symbolized from now on as $\tilde{l}_{-j}(\mathbf{h})$. Now, as far as the KL divergence minimization is concerned, when Gaussian approximations are used, then this is achieved by *moment matching*. A similar theoretical result is also true for all approximations in the exponential family. *Moment matching* means that the two functions whose KL divergence needs to be minimized must have the same moments and since the second function is an un-normalized multivariate normal one, this results to matching the 0_{th}, 1_{st} and 2_{nd} moments of the two parts. This basically indicates that the function to be refined in each EP step, $\tilde{l}_j(\mathbf{h})$, must be adjusted so that the moments of $\tilde{l}_j(\mathbf{h})\ \tilde{l}_{-j}(\mathbf{h})$ are equal to the ones of $l_j(\mathbf{h})\ \tilde{l}_{-j}(\mathbf{h})$.

This is the breaking point of the EP algorithm. Calculating the moments of the true likelihood function and the cavity function product could not be implemented so far analytically or in a computationally cheap way. Researchers have tried numerical integration or independence assumptions to simplify the results, but no exact and analytical solution has ever been delivered for basic likelihood function forms. Now, let us examine the function $l_j(\mathbf{h})\ \tilde{l}_{-j}(\mathbf{h})$. First, we have already shown that $\tilde{l}_{-j}(\mathbf{h})$ is an un-normalized multivariate normal function and we have described $l_j(\mathbf{h})$ as a halfspace indicator function. Thus, $l_j(\mathbf{h})\ \tilde{l}_{-j}(\mathbf{h})$ is actually a one-side truncated multivariate Gaussian and what we need is to calculate its 0_{th}, 1_{st} and 2_{nd} moments, q, $\mathbf{q}$ and $\mathbf{Q}$. To improve the continuity of this manuscript, the analytical moment calculation of a one-side truncated multivariate Gaussian can be found in Appendix B.

Once these moments are computed, $\tilde{l}_j(\mathbf{h})$ is defined using (6.11) and (6.12) as a multivariate normal pdf with covariance matrix:

$$\boldsymbol{\Sigma}_j^{-1} = \mathbf{Q}^{-1} - \boldsymbol{\Sigma}_{-j}^{-1} \tag{6.13}$$

and mean:

$$\boldsymbol{\mu}_j = \left(\mathbf{q}\,\mathbf{Q}^{-1} - \boldsymbol{\mu}_{-j}\,\boldsymbol{\Sigma}_{-j}^{-1}\right)\boldsymbol{\Sigma}_j. \tag{6.14}$$

We also need to highlight that matching the 0_{th} moments does not offer essentially better approximations, because multiplying $\tilde{l}_j(\mathbf{h})$ with a constant may lead to unwanted results in this iterative filtration process. Still, we mentioned this earlier for the sake of completeness and explained it by showing that the product of multivariate normal pdf's is un-normalized.

In the end of this section, we elaborate on the prior pdf, $f_0(\mathbf{h})$. Most commonly, the prior pdf is chosen to represent a prior belief about $\mathbf{h}^*$. Moreover, it should also facilitate us computationally in order to have a well defined posterior pdf. If the likelihood functions are approximated with Gaussian ones, then a reasonable choice for $f_0(\mathbf{h})$ is also to be Gaussian. Here though, we use another function to show exactly the potential of EP. A closer to reality choice for $f_0(\mathbf{h})$ is to define it as a uniform pdf over some bounding box in the $\mathbf{h}$ space. This could represent for example minimum and maximum possible values for $\mathbf{h}^*$. Thus, the prior could be described as a hyper-rectangle which can also be written as the product of 2^N halfspace indicator functions and therefore participate in the EP process.

6.5 Constrained Bayesian Active Learning of Interference Channel Gains

The goal of this chapter is to design SU probing power vectors, $\mathbf{p}$, using observations of ACK/NACK feedback, Z, in order to learn as fast as possible the unknown normalized interference channel gain vector, $\mathbf{h}^*$, while ensuring that the number of probing power vectors causing harmless interference over a time horizon is always above a certain limit. This means that assuming a limited number of N_T probing attempts, $\{\mathbf{p}(0), ..., \mathbf{p}(N_T - 1)\}$ and their corresponding pieces of feedback, $\{Z_0, ..., Z_{N_T-1}\}$, we wish to minimize the uncertainty of our knowledge about $\mathbf{h}^*$, formally represented by the entropy of $f_{N_T}(\mathbf{h})$, subject to maintaining the sum of $Z_t = -1$, where $t = 0, ..., N_T - 1$, below a threshold and which is equivalent to controlling the sum of $Z_t = +1$, where $t = 0, ..., N_T - 1$, above a corresponding limit. This practical constraint is essential for the PU system operation, since the actual deterioration of its link does not depend on the total or average amount

of interference over time caused by the CRN, but on the time ratio during which harmful interference occurs because of SU probing attempts.

In the previous section, we showed the recursive Bayesian update (6.9) which modifies our knowledge about $\mathbf{h}^*$ step by step. This will be our main tool for handling the iterative nature of this proactive probing strategy. Next, we investigate the optimal design policy of a SU probing power vector, which represents a hyperplane in the $\mathbf{h}$ space, that should be chosen in each step of this recursive Bayesian estimation process in order to optimally reduce the posterior pdf entropy after N_T probing power vectors, $\{\mathbf{p}(0), ..., \mathbf{p}(N_T-1)\}$, with their corresponding pieces of feedback, $Z_{0:(N_T-1)}$, subject to $\sum_{t=0}^{N_T-1} \mathbf{1}_{\{Z_t=+1\}} \geq \alpha$, where $\mathbf{1}_{\{..\}}$ is the indicator function and α is the protection time ratio during which the PU link operation must remain undisrupted. The constraint can also be written as $\sum_{t=0}^{N_T-1} Z_t \geq (2\alpha-1)N_T$. This multistage constrained optimization problem can be expressed in the spirit of DP [116] as finding the optimal probing rule that maps $\{f_0, .., f_{N_T-1}\}$ to $\{\mathbf{p}(0), .., \mathbf{p}(N_T-1)\}$ in order to achieve the maximum average entropy reduction from the $f_0(\mathbf{h})$ to the $f_{N_T}(\mathbf{h})$ pdf subject to the aforementioned constraint. In a formal manner, we seek the optimal probing design policy $\pi^*_{0:(N_T-1)} = \{\mathbf{p}(0) = \mu^*(f_0), .., \mathbf{p}(N_T-1) = \mu^*(f_{N_T-1})\}$ which solves the following constrained optimization problem over all possible feedback sequences derived by this policy:

$$\max_{\pi} \quad E^{\pi}[\mathcal{H}(f_0) - \mathcal{H}(f_{N_T})|\mathbf{p}(N_T-1), D_{N_T-2}] \tag{6.15a}$$

$$\text{s.t.} \quad E^{\pi}\left[\sum_{t=0}^{N_T-1} Z_t|\mathbf{p}(N_T-1), D_{N_T-2}\right] \geq (2\alpha-1)N_T \tag{6.15b}$$

where $\mathcal{H}$ is the entropy operator of a pdf. This approach is a constrained equivalent of the proof of Theorem 5.5.1 given in Appendix A for Chapter 5. The objective function of (6.15) which is the conditional expectation of the information gain of an arbitrary policy π can also be expressed in an additive form:

$$\begin{aligned} &E^{\pi}[\mathcal{H}(f_0) - \mathcal{H}(f_{N_T})|\mathbf{p}(N_T-1), D_{N_T-2}] = \\ &E^{\pi}[\mathcal{H}(f_0) - \mathcal{H}(f_1)|\mathbf{p}(0) + ... \\ &+ E^{\pi}[\mathcal{H}(f_{k-1}) - \mathcal{H}(f_k)|\mathbf{p}(k-1), D_{k-2} + ... \\ &+ E^{\pi}[\mathcal{H}(f_{N_T-1}) - \mathcal{H}(f_{N_T})|\mathbf{p}(N_T-1), D_{N_T-2}]...] \end{aligned} \tag{6.16}$$

where we added and subtracted all the entropy terms of the intermediate pdf's to form an additive gain over time and similarly the left part of the constraint of (6.15) can be

written as:

$$E^{\pi}\left[\sum_{t=0}^{N_T-1} Z_t|\mathbf{p}(N_T-1), D_{N_T-2}\right] =$$
$$E^{\pi}[Z_0|\mathbf{p}(0) + ... + E^{\pi}[Z_{k-1}|\mathbf{p}(k-1), D_{k-2} + ...$$
$$+ E^{\pi}[Z_{N_T-1}|\mathbf{p}(N_T-1), D_{N_T-2}]...]. \tag{6.17}$$

After we invert the entropy subtractions, in order to reform the optimization problem into a minimization one, and move the left part of (6.15b) on the right side, we create the Lagrangian of this multistage problem as:

$$J^{\lambda}_{0:(N_T-1)} = E^{\pi}[\mathcal{H}(f_1) - \mathcal{H}(f_0) - \lambda Z_0|\mathbf{p}(0) + ...$$
$$+ E^{\pi}[\mathcal{H}(f_k) - \mathcal{H}(f_{k-1}) - \lambda Z_{k-1}|\mathbf{p}(k-1), D_{k-2} + ...$$
$$+ E^{\pi}[\mathcal{H}(f_{N_T}) - \mathcal{H}(f_{N_T-1}) - \lambda Z_{N_T-1}|\mathbf{p}(N_T-1), D_{N_T-2}]...]+$$
$$+ \lambda(2\alpha-1)N_T \tag{6.18}$$

where λ is the KKT multiplier related to (6.15b) and which has to be non-negative. Now, we need to minimize $J^{\lambda}_{0:(N_T-1)}$ for an abstract λ and we can do so without including the last term $\lambda(2\alpha-1)N_T$, since it is independent of the policy π. The new form of the Lagrangian will thus be $\Lambda^{\lambda}_{0:(N_T-1)} = J^{\lambda}_{0:(N_T-1)} - \lambda(2\alpha-1)N_T$. Additionally, to bring our problem closer to the DP formulation, we define the subtail problem Lagrangian or Lagrangian-to-go, $\Lambda^{\lambda}_{k:(N_T-1)}$, as:

$$\Lambda^{\lambda}_{k:(N_T-1)} =$$
$$E^{\pi}[\mathcal{H}(f_{k+1}) - \mathcal{H}(f_k) - \lambda Z_k|\mathbf{p}(k), D_{k-1} + ...$$
$$+ E^{\pi}[\mathcal{H}(f_{N_T}) - \mathcal{H}(f_{N_T-1}) - \lambda Z_{N_T-1}|\mathbf{p}(N_T-1), D_{N_T-2}]...]. \tag{6.19}$$

and we denote its minimum value as $\Lambda^{*\lambda}_{k:(N_T-1)}$. By employing the principle of optimality, we have:

$$\Lambda^{*\lambda}_{k:(N_T-1)} =$$
$$\min_{\pi} E^{\pi}\left[\mathcal{H}(f_{k+1}) - \mathcal{H}(f_k) - \lambda Z_k|\mathbf{p}(k), D_{k-1} + \Lambda^{*\lambda}_{(k+1):(N_T-1)}\right] \tag{6.20}$$

and based on this, we may proceed with the backward induction logic of DP.

Before we continue though with the DP solution of our constrained multistage problem, let us first redefine the multivariate cumulative distribution function (cdf) in a more "natural" than the usual way. Assuming a multivariate pdf f in $S \subseteq \mathbb{R}^N$ and a vector $\mathbf{x} = [x_1, ..., x_N]$, usually its cdf F is defined as $F(\mathbf{x}) = \Pr[X_1 \leq x_1, ..., X_N \leq x_N]$ which

is the joint probability of its components $X_1, ..., X_N$, that are scalar valued random variables, being less or equal than the values $x_1, ..., x_N$ respectively. Nevertheless, this definition is not geometrically smooth and commonly used just because it is easy to be computed in case of independent $\mathbf{x}$ components. Here, we describe it more strictly and not just by using a "box limit"-like definition. Assuming a hyperplane in $\mathbb{R}^n$:

$$\mathbf{x}\ \mathbf{w}^\intercal = 1 \tag{6.21}$$

we alternatively determine the cdf C of a multivariate pdf f as:

$$C(\mathbf{w}) = \Pr[\mathbf{x}\ \mathbf{w}^\intercal \leq 1] = \int\limits_{\mathbf{x}\ \mathbf{w}^\intercal \leq 1} f(\mathbf{x})\ dV_{\mathbf{x}}. \tag{6.22}$$

For our case study, this means that the posterior cdf after the $(t-1)$ step, $C_t(\mathbf{p})$, is expressed as:

$$C_t(\mathbf{p}) = \Pr[\mathbf{h}\ \mathbf{p}^\intercal \leq 1 | \mathbf{h} = \mathbf{h}^*, D_{t-1}] = \int\limits_{\mathbf{h}\ \mathbf{p}^\intercal \leq 1} f_t(\mathbf{h})\ dV_{\mathbf{h}} \tag{6.23}$$

and the support region of $f_t(\mathbf{h})$ is limited to the positive orthant of the $\mathbf{h}$ space, $\mathbb{R}_+^N$, because the interference channel gains can only have non negative values.

Further on, we elaborate on the marginal likelihood of (6.9). In the event of $Z_t = +1$, the conditional probability $\Pr[Z_t|\mathbf{p}(t), D_{t-1}]$ can also be written according to the Bayes sum rule, the product rule and the conditional independences from Section IV as in

(6.24).

$$\Pr[Z_t=+1|\mathbf{p}(t),D_{t-1}]=\int\limits_{\mathbb{R}_+^N}\Pr[Z_t=+1,\mathbf{h}=\mathbf{h}^*|\mathbf{p}(t),D_{t-1}]\;dV_{\mathbf{h}}=$$

$$\int\limits_{\mathbb{R}_+^N}\frac{\Pr[Z_t=+1|\mathbf{h}=\mathbf{h}^*,\mathbf{p}(t),D_{t-1}]\;\Pr[\mathbf{h}=\mathbf{h}^*,\mathbf{p}(t),D_{t-1}]}{\Pr[\mathbf{p}(t),D_{t-1}]}dV_{\mathbf{h}}=$$

$$\int\limits_{\mathbb{R}_+^N}\Pr[Z_t=+1|\mathbf{h}=\mathbf{h}^*,\mathbf{p}(t),D_{t-1}]\Pr[\mathbf{h}=\mathbf{h}^*|\mathbf{p}(t),D_{t-1}]dV_{\mathbf{h}}=$$

$$\int\limits_{\mathbf{h}\;\mathbf{p}^\intercal\leq 1}\Pr[Z_t=+1|\mathbf{h}=\mathbf{h}^*,\mathbf{p}(t),D_{t-1}]\Pr[\mathbf{h}=\mathbf{h}^*|\mathbf{p}(t),D_{t-1}]dV_{\mathbf{h}}+$$

$$\int\limits_{\mathbf{h}\;\mathbf{p}^\intercal>1}\Pr[Z_t=+1|\mathbf{h}=\mathbf{h}^*,\mathbf{p}(t),D_{t-1}]\Pr[\mathbf{h}=\mathbf{h}^*|\mathbf{p}(t),D_{t-1}]dV_{\mathbf{h}}=$$

$$\int\limits_{\mathbf{h}\;\mathbf{p}^\intercal\leq 1}\Pr[Z_t=+1|\mathbf{h}=\mathbf{h}^*,\mathbf{p}(t),D_{t-1}]\Pr[\mathbf{h}=\mathbf{h}^*|D_{t-1}]dV_{\mathbf{h}}+$$

$$\int\limits_{\mathbf{h}\;\mathbf{p}^\intercal>1}\Pr[Z_t=+1|\mathbf{h}=\mathbf{h}^*,\mathbf{p}(t),D_{t-1}]\Pr[\mathbf{h}=\mathbf{h}^*|D_{t-1}]dV_{\mathbf{h}}=$$

$$\int\limits_{\mathbf{h}\;\mathbf{p}^\intercal\leq 1}\Pr[Z_t=+1|\mathbf{h}=\mathbf{h}^*,\mathbf{p}(t),D_{t-1}]\;f_t(\mathbf{h})dV_{\mathbf{h}}+$$

$$\int\limits_{\mathbf{h}\;\mathbf{p}^\intercal>1}\Pr[Z_t=+1|\mathbf{h}=\mathbf{h}^*,\mathbf{p}(t),D_{t-1}]\;f_t(\mathbf{h})dV_{\mathbf{h}}=$$

$$\int\limits_{\mathbf{h}\;\mathbf{p}^\intercal\leq 1}\Pr[Z_t=+1|\mathbf{h}=\mathbf{h}^*,\mathbf{p}(t)]\;f_t(\mathbf{h})dV_{\mathbf{h}}+\int\limits_{\mathbf{h}\;\mathbf{p}^\intercal>1}\Pr[Z_t=+1|\mathbf{h}=\mathbf{h}^*,\mathbf{p}(t)]\;f_t(\mathbf{h})dV_{\mathbf{h}}=$$

$$\int\limits_{\mathbf{h}\;\mathbf{p}^\intercal\leq 1}f_t(\mathbf{h})dV_{\mathbf{h}}=C_t(\mathbf{p}(t)) \tag{6.24}$$

A similar expression can also be derived for the $Z_t=-1$ event:

$$\Pr[Z_t=-1|\mathbf{p}(t),D_{t-1}]=1-C_t(\mathbf{p}(t)). \tag{6.25}$$

Moving on with our DP solution, we apply the backward induction logic of DP and first solve $\min\limits_{\pi} E^{\pi}\left[\Lambda^{\lambda}_{(N_T-1):(N_T-1)}\right]$ which is equivalent to:

$$\min_{\mathbf{p}(N_T-1)} E^{\pi}[\mathcal{H}(f_{N_T})-\mathcal{H}(f_{N_T-1})-\lambda Z_{N_T-1}|\mathbf{p}(N_T-1),D_{N_T-2}]. \tag{6.26}$$

Now, let us first evaluate the term $E^{\pi}[\mathcal{H}(f_{N_T})-\mathcal{H}(f_{N_T-1})-\lambda Z_{N_T-1}|\mathbf{p}(N_T-1),D_{N_T-2}]$, where $E^{\pi}[.]$ is basically the expectation over the two possible observations $Z_{N_T-1}=+1$

and $Z_{N_T-1} = -1$, by using (6.9) and the equivalence of the conditions D_{N_T-2} and f_{N_T-1}:

$$\begin{aligned}
&E^{\pi}[\mathcal{H}(f_{N_T}) - \mathcal{H}(f_{N_T-1}) - \lambda Z_{N_T-1}|\mathbf{p}(N_T-1), f_{N_T-1}] = \\
&E^{\pi}\left[E_{\mathbf{h}}\left[-\log(f_{N_T-1})\right]\right] - E^{\pi}\left[E_{\mathbf{h}}\left[-\log(f_{N_T-1})\right]\right] + \\
&+ E^{\pi}[E_{\mathbf{h}}\left[-\log(\Pr[Z_{N_T-1}|\mathbf{h}=\mathbf{h}^*, \mathbf{p}(N_T-1)])\right]|\mathbf{p}(N_T-1), f_{N_T-1}] - \\
&- E^{\pi}[E_{\mathbf{h}}\left[-\log(\Pr[Z_{N_T-1}|\mathbf{p}(N_T-1), f_{N_T-1}])\right]|\mathbf{p}(N_T-1), f_{N_T-1}] - \\
&- \lambda E^{\pi}[Z_{N_T-1}|\mathbf{p}(N_T-1), f_{N_T-1}].
\end{aligned} \tag{6.27}$$

The last three remaining terms can be further processed. With the help of (6.7) for $\Pr[Z_{N_T-1}|\mathbf{h}=\mathbf{h}^*, \mathbf{p}(N_T-1)]$, the third term can be analyzed as:

$$\begin{aligned}
&E^{\pi}[E_{\mathbf{h}}\left[-\log(\Pr[Z_{N_T-1}|\mathbf{h}=\mathbf{h}^*, \mathbf{p}(N_T-1)])\right]|\mathbf{p}(N_T-1), f_{N_T-1}] = \\
&E^{\pi}\left[E_{\mathbf{h}}\left[-\log(\Pr[Z_{N_T-1}|\mathbf{h}=\mathbf{h}^*, \mathbf{p}(N_T-1)])\right]|f_{N_T-1}\right] = \\
&E^{\pi}\left[-\log(\Pr[Z_{N_T-1}|\mathbf{h}=\mathbf{h}^*, \mathbf{p}(N_T-1)])\right] = 0
\end{aligned} \tag{6.28}$$

where we exploited the fact that Z_{N_T-1} does not depend on f_{N_T-1} given $\mathbf{h} = \mathbf{h}^*$ and $\mathbf{p}(N_T-1)$. Additionally, by using (6.24) and (6.25) which again lead us to omit $E_{\mathbf{h}}$, since $\Pr[Z_{N_T-1}|\mathbf{p}(N_T-1), f_{N_T-1}]$ is stable over the $\mathbf{h}$ domain, the fourth term becomes:

$$\begin{aligned}
&E^{\pi}[E_{\mathbf{h}}\left[-\log(\Pr[Z_{N_T-1}|\mathbf{p}(N_T-1), f_{N_T-1}])\right]|\mathbf{p}(N_T-1), f_{N_T-1}] = \\
&E^{\pi}\left[-\log(\Pr[Z_{N_T-1}|\mathbf{p}(N_T-1), f_{N_T-1}])\right] = \\
&- C_{N_T-1}(\mathbf{p}(N_T-1))\log(C_{N_T-1}(\mathbf{p}(N_T-1))) - \\
&- (1 - C_{N_T-1}(\mathbf{p}(N_T-1)))\log((1 - C_{N_T-1}(\mathbf{p}(N_T-1)))).
\end{aligned} \tag{6.29}$$

Finally, we elaborate on the fifth term:

$$\begin{aligned}
&\lambda E^{\pi}[Z_{N_T-1}|\mathbf{p}(N_T-1), f_{N_T-1}] = \\
&\lambda[(+1)\Pr[Z_{N_T-1} = +1|\mathbf{p}(N_T-1), f_{N_T-1}] + (-1)\Pr[Z_{N_T-1} = -1|\mathbf{p}(N_T-1), f_{N_T-1}]] = \\
&\lambda[C_{N_T-1}(\mathbf{p}(N_T-1)) - (1 - C_{N_T-1}(\mathbf{p}(N_T-1)))].
\end{aligned} \tag{6.30}$$

We observe that minimizing (6.27) using (6.28), (6.29) and (6.30) over $\mathbf{p}(N_T-1)$ is equivalent to minimizing (6.27) over C_{N_T-1}, since the term $\mathbf{p}(N_T-1)$ appears only inside $C_{N_T-1}(.)$. Consequently, this results to the following problem where we include (6.28), (6.29) and (6.30) in (6.27) and simplify the notation for the sake of space with the help of $C = C_{N_T-1}(\mathbf{p}(N_T-1))$:

$$\Lambda^{\lambda}_{(N_T-1):(N_T-1)} = C\log(C) + (1-C)\log(1-C) - \lambda(2C-1) \tag{6.31}$$

and thus (6.26) becomes:

$$\min_{C} \left[C\log(C) + (1-C)\log(1-C) - \lambda(2C-1)\right]. \tag{6.32}$$

Solving (6.32) by imposing $\frac{\partial \Lambda^{\lambda}_{(N_T-1):(N_T-1)}}{\partial C} = 0$ results to the value of $C = \frac{e^{2\lambda}}{1+e^{2\lambda}}$ which delivers $\Lambda^{*\lambda}_{(N_T-1):(N_T-1)} = \lambda - \log(1+e^{2\lambda})$. We notice that this minimum value of the Lagrangian-to-go $\Lambda^{\lambda}_{(N_T-1):(N_T-1)}$ is a constant value and independent of the time step. This allows us to state that by moving backwards in time at the $(k+1)$ time step, the accumulated constant values of the of the Lagrangian's-to-go yield the following:

$$\Lambda^{*\lambda}_{(k+1):(N_T-1)} = ((N_T-1)-(k+1)+1)\left(\lambda - \log(1+e^{2\lambda})\right). \tag{6.33}$$

Proceeding with our DP solution, we now solve (6.20) using the same procedure as before and we obtain that:

$$\Lambda^{*\lambda}_{k:(N_T-1)} = (N_T-k)\left(\lambda - \log(1+e^{2\lambda})\right) \tag{6.34}$$

which for $k=0$ gives $\Lambda^{*\lambda}_{0:(N_T-1)} = N_T\left(\lambda - \log(1+e^{2\lambda})\right)$. Consequently, the dual function $q(\lambda)$ of (6.15), which is always concave, is defined as:

$$\begin{aligned} q(\lambda) = J^{*\lambda}_{0:(N_T-1)} = \Lambda^{*\lambda}_{0:(N_T-1)} + \lambda(2\alpha-1)N_T = \\ = N_T\left(\lambda - \log(1+e^{2\lambda})\right) + \lambda(2\alpha-1)N_T \end{aligned} \tag{6.35}$$

which enables us to rewrite (6.15) as:

$$\max_{\lambda} \; q(\lambda) \tag{6.36a}$$

$$\text{s.t.} \;\; \lambda \geq 0 \tag{6.36b}$$

and solve this by imposing $\frac{\partial q(\lambda)}{\partial \lambda} = 0$ which delivers $\lambda^* = 0.5\log(\frac{\alpha}{1-\alpha})$. For $\alpha \geq 0.5$, which is the lower reasonable limit of the time ratio during which the CRN probes protectively to the PU system, we always have $\lambda^* > 0$ and therefore the constraint (6.15b) is active because of the complementary slackness condition. Finally, we conclude by using λ^* that the optimal probing design policy must satisfy $C_t(\mathbf{p}(t)) = \alpha$ or equivalently $\mathbf{p}(t) = \mu^*(f_t(\mathbf{h})) = C_t^{-1}(\alpha)$ and for this reason $\pi^*_{0:(N_T-1)} = \{\mathbf{p}(0) = C_0^{-1}(\alpha), .., \mathbf{p}(N_T-1) = C^{-1}_{N_T-1}(\alpha)\}$.

At this point, we must emphasize some aspects of the optimal policy. This constrained DP problem must somehow take into account the actual obtained pieces of feedback and not just the expected ones derived from the probabilistic formulation of our problem. This is similar to inventory control problems with stock constraints where we may know

probabilistically the product demands over time, but we also need to include into the inventory control the actual demands already arrived before each time step. This means that the protection time ratio α should be adapted to the past feedback observations.

Now, let us take a closer look to the optimal policy at some arbitrary time step k. The multistage optimization problem in time step k has a form similar to the one of (6.15), only that this time we are interested in maximizing the information gain in the remaining steps and still maintaining the overall violation constraint:

$$\max_{\pi} \quad E^{\pi}[\mathcal{H}(f_k) - \mathcal{H}(f_{N_T})|\mathbf{p}(N_T-1), D_{N_T-2}] \tag{6.37a}$$

$$\text{s.t.} \quad E^{\pi}\left[\sum_{t=0}^{N_T-1} Z_t|\mathbf{p}(N_T-1), D_{N_T-2}\right] \geq (2\alpha-1)N_T \tag{6.37b}$$

We observe that the constraint (6.37b) can also be written as:

$$E^{\pi}\left[\sum_{t=k}^{N_T-1} Z_t|\mathbf{p}(N_T-1), D_{N_T-2}\right] \geq (2\alpha-1)N_T - \sum_{t=0}^{k-1} Z_t \tag{6.38}$$

since the pieces of feedback $\{Z_0, ..., Z_{k-1}\}$ already happened. If we manage to reformulate the left hand side of (6.38) in the fashion of (6.15b), then the problem defined by (6.37a) and (6.38) is solved with the same optimal policy derived for (6.15), but with a different α. Specifically, we wish the left hand side of (6.38) to have the form $(2\alpha_k - 1)(N_T - k)$ which by equating the two expressions generates the following α_k value:

$$\alpha_k = \frac{2\alpha N_T - k - \sum_{t=0}^{k-1} Z_t}{2(N_T-k)} \tag{6.39}$$

Therefore, the overall optimal adaptive policy can now be expressed as $\pi^*_{0:(N_T-1)} = \{\mathbf{p}(0) = C_0^{-1}(\alpha_0), .., \mathbf{p}(N_T-1) = C_{N_T-1}^{-1}(\alpha_{N_T-1})\}$ where $\alpha_0 = \alpha$.

Additionally, we need to point out an important issue in AL which was emphasized in our previous work [10, 13], the necessity of exploration. Reducing the uncertainty of our knowledge about $\mathbf{h}^*$ must be performed by approaching this exact value uniformly from all directions. This means that the training samples in an AL process, in this case the power probing vectors, must be diversified and this can be accomplished by choosing hyperplanes in the version space of random direction uniformly. Therefore, we need first to define how to uniformly sample a random direction $\boldsymbol{\theta}$, where $\boldsymbol{\theta}$ is a unit vector. This problem is related to the uniform unit hypersphere point picking which has been thoroughly described in [10, 13]. Hence, in order to produce a power vector which represents a hyperplane of random direction, $\mathbf{p}(t)$ must be parallel to a randomly

generated $\boldsymbol{\theta}$, $\mathbf{p}(t) = \beta\boldsymbol{\theta}$ where $\beta \in \mathbb{R}$, and it must also satisfy $C_t(\mathbf{p}(t)) = \alpha$ according to our previous analysis. In a formal manner, this is expressed using (6.22) as:

$$\int\limits_{\mathbf{h}\ \beta\boldsymbol{\theta}^{\intercal}\leq 1} f_t(\mathbf{h})\ dV_{\mathbf{h}} = C_t(\beta\boldsymbol{\theta}) = \alpha. \tag{6.40}$$

At this point, we make use of the Gaussian approximation of each step's posterior pdf which we developed in Section IV with the help of EP. In accordance with that result, $f_t(\mathbf{h})$ can be approximated by the normalized version of $\prod_{i=0}^{t-1} \tilde{l}_i(\mathbf{h})$ which we denote as $\tilde{f}_t(\mathbf{h})$. So, (6.40) now becomes:

$$\int\limits_{\mathbf{h}\ \beta\boldsymbol{\theta}^{\intercal}\leq 1} \tilde{f}_t(\mathbf{h})\ dV_{\mathbf{h}} = \alpha \tag{6.41}$$

With the help of the transformation scheme described in Appendix B and after some processing, we obtain that:

$$\mathbf{p}(t) = \beta\boldsymbol{\theta} \tag{6.42}$$

where $\beta = \frac{1}{F^{-1}(\alpha;c_1,c_2)}$ and $F^{-1}(.)$ is the inverse cdf of the univariate normal pdf with mean c_1 and variance c_2. Furthermore, $c_1 = \boldsymbol{\theta}\ \tilde{\boldsymbol{\mu}}^{\intercal}(t)$, where $\tilde{\boldsymbol{\mu}}(t)$ is the mean row vector of $\tilde{f}_t(\mathbf{h})$, and $c_2 = \sum_{i=1}^{N} \theta_i \boldsymbol{\theta}(\tilde{\boldsymbol{\Sigma}}_{:,i}(t))^{\intercal}$, where $\tilde{\boldsymbol{\Sigma}}_{:,i}(t)$ is the i_{th} column of the covariance matrix of $\tilde{f}_t(\mathbf{h})$. Moreover, all the coordinates of $\mathbf{p}(t)$, which represent power levels, must be non negative, otherwise a new $\boldsymbol{\theta}$ has to be generated until a valid power vector is produced.

6.6 Results

In this section, we provide simulation results to demonstrate the performance of the Constrained Bayesian AL method presented in this chapter. The figures of this section show the channel estimation error depending on the number of time flops where each time flop is the time period necessary for the CBS to decode the ACK/NACK packet, design the SU probing power vector and probe the PU system. The interference channel gain vector estimation error metric at each time flop is defined as the normalized root-square error $\frac{\|\tilde{\boldsymbol{\mu}}(t)-\mathbf{h}^*\|}{\|\mathbf{h}^*\|}$ similarly to the previous chapters. The error figure results are obtained as the average of the error metric defined earlier over 100 SU random topologies, which deliver 100 random draws of interference channel gain vectors $\mathbf{h}^*$. Moreover, each figure of subsection 6.6.2 is followed by a metric which examines the protection of the PU link quality as the proposed method progresses in time. This can be measured by the time ratio during which the induced interference caused to the PU system is

harmless. This is actually the time ratio during which pieces of feedback $Z_t = +1$ occur, $\frac{\sum_{t=0}^{N_T-1} \mathbf{1}_{\{Z_t=+1\}}}{N_T}$. This parameter of harmless interference is also averaged over the 100 SU random topologies to deliver the corresponding average protection metric α_{sim}.

6.6.1 Simulation Parameters

As far as the technical parameters of the simulations are concerned, the PU receiver is chosen to normally operate and acknowledge with ACK packets when interference is below $I_{th} = -97$dBm, a limit unknown to the CRN. The examined scenarios consider $N = 5$ and $N = 10$ SUs which are dispersed uniformly within a 3km range around the PU receiver. The interference channel gains that are unknown to the CRN are assumed to follow an exponential path loss model $g_i = \frac{1}{d_i^4}$, where d_i is the distance of the SU_i from the PU receiver in meters. Additionally, the protection time ratio α takes the following values $\{0.5,\ 0.7,\ 0.9\}$ where $\alpha = 0.5$ basically means that protecting the PU is not considered at all. The remaining scenario parameter is the "budget" of N_T probing attempts which can also be considered as the pilot time window and it is assumed to be $N_T = 100$ for the $N = 5$ SU case and $N_T = 200$ for the $N = 10$ SU case.

6.6.2 Estimation Performance of the Constrained Bayesian AL Method

Initially, let us see in Fig. 6.2, 6.3 and 6.4 the performance of the proposed Constrained Bayesian AL technique for $N = 5$ SUs. Here, it can be clearly seen that as α is increased, more probing attempts are required to correctly estimate $\mathbf{h}^*$. More specifically, in the case of $\alpha = 0.5$, Fig. 6.2, for an estimation error 1%, convergence is achieved in 72 time flops, whereas the corresponding number of time flops for $\alpha = 0.7$ is 89 as it can be seen in Fig. 6.3. Furthermore, the convergence of our method becomes worse for $\alpha = 0.9$ as shown in Fig. 6.4, where after 100 probing attempts, the estimation error is 6%. These results prove that as the design parameter of PU protection α increases, the CBS designs less harmful for the PU system probing power vectors, but also less informative about $\mathbf{h}^*$.

As far as the α_{sim} metric for these three cases is concerned, for $\alpha = 0.5$, $\alpha = 0.7$ and $\alpha = 0.9$, the resulting α_{sim} values are $\alpha_{sim} = 0.49$, $\alpha_{sim} = 0.68$ and $\alpha_{sim} = 0.87$ respectively. The small differences between the target values of the protection time ratio, α, and the simulated ones, α_{sim}, appear because of the inaccurate estimation of the each step posterior pdf using the EP. Even though EP is a very accurate, sophisticated and fast method for density estimation, the approximated posterior pdf's still have some

deviation from the real ones. This results in computing power vectors which satisfy (6.41) but not its exact version, (6.40).

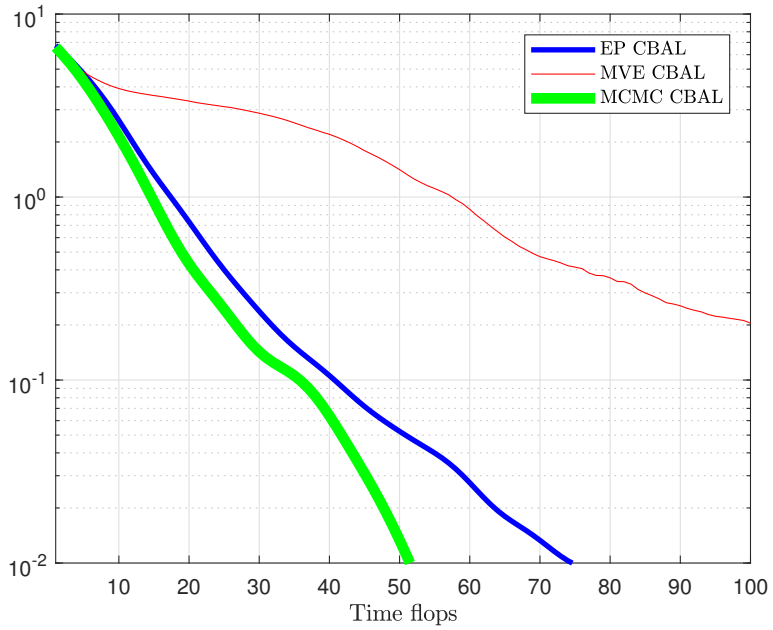

FIGURE 6.2: Interference channel gain vector estimation error progress vs time of the Constrained Bayesian AL method for $\alpha = 0.5$ and $N = 5$ SUs

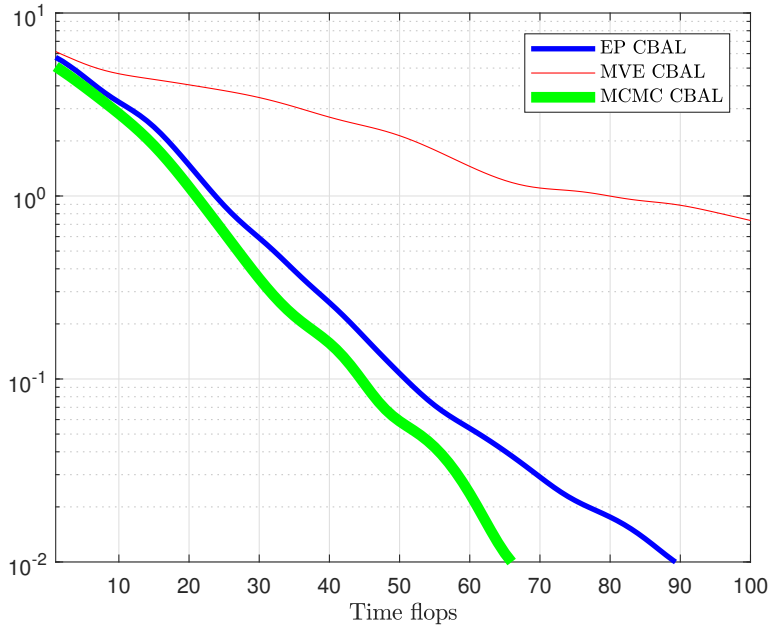

FIGURE 6.3: Interference channel gain vector estimation error progress vs time of the Constrained Bayesian AL method for $\alpha = 0.7$ and $N = 5$ SUs

Next, we examine for $N = 10$ SUs and designed protection time ratio $\alpha = 0.7$ the performance of our technique which is illustrated in Fig. 6.5. After $N_T = 200$ time flops, or probing attempts, the $\mathbf{h}^*$ estimation error is 2.5% and the simulated protection time

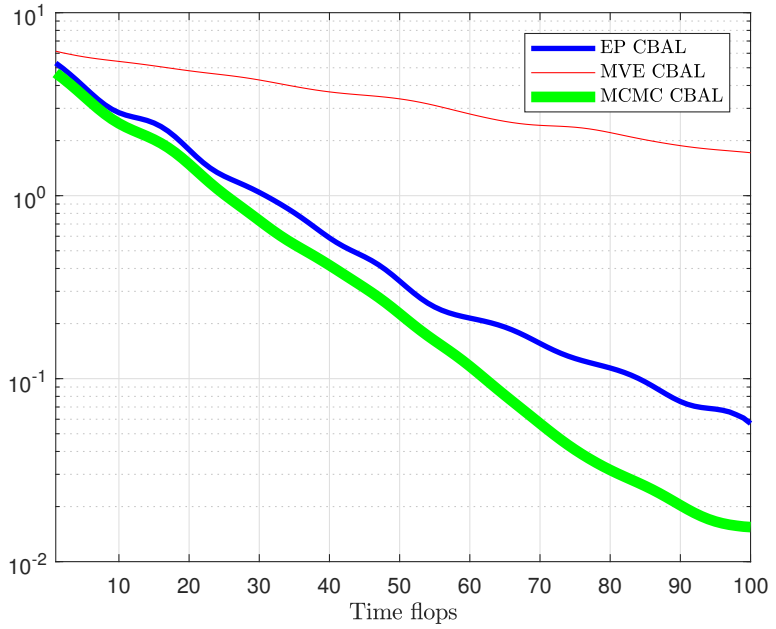

FIGURE 6.4: Interference channel gain vector estimation error progress vs time of the Constrained Bayesian AL method for $\alpha = 0.9$ and $N = 5$ SUs

ratio is $\alpha_{sim} = 0.67$. The reason for checking the learning efficiency of the Constrained Bayesian AL method for $N = 10$ SUs is to observe its behavior when the learning problem dimensions grow. We observe by comparing the results of Fig. 6.3 and 6.5 that the convergence time for an estimation error of 2.5% increases from 66 time flops in Fig. 6.3 to 200 in Fig. 6.5. Hence, we could empirically claim that the convergence rate of our method depending on the number of SUs, N, is approximately of order $\mathcal{O}(N \log_2 N)$.

6.7 Summary

In this chapter, we proposed a sequential probing method in order for a centralized CRN to learn fast the PU interference constraint using the ACK/NACK PU feedback while limiting the number of PU outage events under a certain threshold. This problem was formulated within the Constrained DP framework and its optimal solution policy was implemented with the help of an advanced, fast and accurate Bayesian Learning technique, the EP, which was for the first time developed analytically without independence assumptions about the latent variables. The performance of this method was demonstrated through numerical simulations in static channel scenarios for interference channel gain learning. Additionally, results were given for the induced PU outage time ratio to confirm that the simulated PU protection metric α_{sim} is satisfactorily close to the target, or design, PU protection time ratio α.

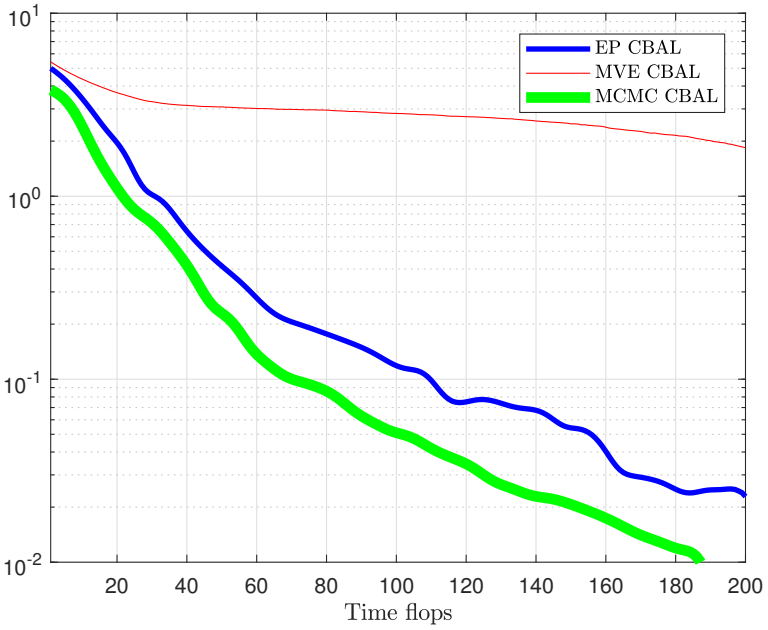

FIGURE 6.5: Interference channel gain vector estimation error progress vs time of the Constrained Bayesian AL method for $\alpha = 0.7$ and $N = 10$ SUs

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis, we have demonstrated various Machine Learning applications in underlay Cognitive Radio scenarios. These ML ideas were developed to enhance the Spectrum Sensing and Decision Making capabilities of a CRN. First, in Chapter 3, a Modulation and Coding Classification process was constructed aiming at enabling the CR to identify the PU Modulation and Coding Scheme. This MCC process considered Higher Order Statistical features of the sensed PU signal and an efficient ML classifier, the Support Vector Machine, to identify the PU modulation scheme and the log-likelihood ratios of the PU signal code syndromes to find the PU encoder. Combining in a serial way these identification procedures enable a CR to find the PU MCS even in low sensing SNR conditions. Moreover, a simple collaborative MCC scheme was proposed to allow a Cognitive Base Station fuse multiple MCS estimates from a group of CRs. This SS ability played an important role in bridging the communication gap between the CRN and the PU system.

In Chapter 4, assuming a PU system operating based on an Adaptive Coding and Modulation protocol, we exploited this PU reaction, the PU MCS adjustment, both for optimizing the CRN performance and learning how to mitigate the induced to the PU interference. The multilevel MCC sensing *feedback* as implicit CSI of the PU link facilitates the CRN to constantly observe the impact of the aggregated interference it causes. In the examined scenario, the CRN intelligently adjusts the SU transmit power levels in order to maximize the CRN throughput, a Power Control (PC) optimization objective, and ensure learning the unknown PU interference constraint of the aforementioned optimization problem. For this scenario setting, the Cutting Plane Methods were used and provided the perfect framework for this joint optimization and learning problem

with high convergence learning rates which are guaranteed theoretically. Specifically, the proposed scheme, an adaptation of the Center of Gravity CPM, managed to deliver better learning results compared to the existing methods in the literature and furthermore better protection to the PU system during this combined optimization and learning process. The work of this chapter established the *Active Learning* spirit of this thesis on which we further elaborated in the next chapter by considering uncertain PU pieces of feedback.

In Chapter 5, we aimed exclusively at learning the interference PU constraint and not at optimizing the CRN performance and adopted the same sequential AL probing rationale as in Chapter 4. Here, we considered the ACK/NACK binary packet instead of the MCC feedback and acquired it by decoding the reverse PU link messages. This rudimentary piece of feedback which is a binary indicator of harmful or harmless induced interference is chosen in order to solely focus on developing sophisticated AL techniques for learning the PU interference constraint. The enhancement introduced in this chapter is that the probability of each binary feedback being correct is taken into account in this intelligent probing mechanism by constructing univariate and multivariate Bayesian AL methods inspired respectively by the Probabilistic Bisection Algorithm (PBA) and the CPMs. Specifically, a median based optimal Bayesian AL method and an MVE-CPM based AL scheme, which is less accurate but computationally affordable, were implemented for designing the SU probing power vectors. Simulations were performed and the learning superiority of both techniques was shown compared to existing ones in the literature and to our first Bayesian AL attempt presented in Section 4 of Chapter 5. Finally, results were provided proving that both of these schemes are also more protective to the PU among the MCMC based and the computationally cheap AL techniques respectively.

In Chapter 6, we presented a constrained version of the AL scenario in Chapter 5. Again, a Bayesian AL perspective is adopted but with a practical for the PU system constraint, limiting under a threshold the number of harmful probing-induced interference events or equivalently of NACK packet observations over a certain time window. In order to design such a probing policy, the problem was studied within the framework of Constrained DP and a provenly optimal solution for this Constrained AL problem was obtained. Furthermore, we utilized a sophisticated, accurate and fast Bayesian Learning method, the EP, instead of the slow MCMC methods of Chapter 5 for the implementation of this Constrained Bayesian AL case study. The effectiveness of this solution is demonstrated through numerical simulations.

7.2 Future Work

As part of our future work, we intend to elaborate more on the AL principle, find different application areas of it and exploiting Bayesian knowledge for Decision Making and Optimization purposes. Particularly, the possible improvements and extensions are:

1. Developing Bayesian AL techniques for fading interference channel gain learning based on the forgetting factor approach of [75].
2. Implementing decentralized AL schemes with a message passing mechanism suitable for distributed CRNs and cooperative Cognitive Radar scenarios.
3. Focusing on more practical and commercial scenarios where AL principles can be used such as targeted advertising and active caching.
4. Handling the acquired probabilistic knowledge of interference channel gains, or any other unkown parameters, in Stochastic Programming or Robust Optimization problems for defining the optimal average or worst case CRN operation respectively.
5. Deriving probing designs which not only actively learn and maximize information gain of unknown parameters, but also incorporate other metrics such as CRN throughput, CRN energy consumption and PU induced interference. These metrics could be incorporated into an optimization problem, but just one of them as an objective, while the rest as constraints, so that we could still guarantee some lower or upper bounds for the latter.
6. Designing a Multi-objective optimization problem to study the trade-off's in satisfying the aforementioned different objectives.
7. Embarking on purely Decision Making tools to explore possible mechanisms which can be applied in AL scenarios under different assumptions and system models.

Appendix A

Proof of Theorem 5.5.1

First of all, let us define the multivariate cumulative distribution function (cdf) in a more "natural" than the usual way. Assuming a multivariate pdf f in $S \subseteq \mathbb{R}^N$ and a vector $\mathbf{x} = [x_1, ..., x_N]$, usually its cdf F is defined as $F(\mathbf{x}) = \Pr[X_1 \leq x_1, ..., X_N \leq x_N]$ which is the joint probability of its components $X_1, ..., X_N$, that are scalar valued random variables, being less or equal than the values $x_1, ..., x_N$ respectively. Nevertheless, this definition is not geometrically smooth and commonly used just because it is easy to be computed in case of independent $\mathbf{x}$ components. Here, we describe it more strictly and not just by using a "box limit"-like definition. Assuming a hyperplane in $\mathbb{R}^n$:

$$\mathbf{x}\ \mathbf{w}^\intercal = 1 \tag{A.1}$$

we alternatively determine the cdf C of a multivariate pdf f as:

$$C(\mathbf{w}) = \Pr[\mathbf{x}\ \mathbf{w}^\intercal \leq 1] = \int\limits_{\mathbf{x}\ \mathbf{w}^\intercal \leq 1} f(\mathbf{x})\ dV_{\mathbf{x}}. \tag{A.2}$$

Specifically for our case study, the posterior cdf after the $(t-1)$ step, $C_t(\mathbf{p})$, is expressed as:

$$C_t(\mathbf{p}) = \Pr[\mathbf{h}\ \mathbf{p}^\intercal \leq 1 | \mathbf{h} = \mathbf{h}^*, Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}] = \int\limits_{\mathbf{h}\ \mathbf{p}^\intercal \leq 1} f_t(\mathbf{h})\ dV_{\mathbf{h}} \tag{A.3}$$

and the support region of $f_t(\mathbf{h})$ is limited to the positive orthant of the $\mathbf{h}$ space, $\mathbb{R}^N_+$, because the interference channel gains can only have non negative values.

Further on, we elaborate on the marginal likelihood of (5.13). In the event of $Z_t = +1$, the conditional probability $\Pr[Z_t | \mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]$ can also be written according to the Bayes sum rule, the product rule and the conditional independences from Chapter

5 as:

$$
\begin{aligned}
&\Pr[Z_t = +1|\mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}] = \\
&\int_{\mathbb{R}_+^N} \Pr[Z_t = +1, \mathbf{h} = \mathbf{h}^*|\mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]\, dV_{\mathbf{h}} = \\
&\int_{\mathbb{R}_+^N} \frac{\Pr[Z_t = +1|\mathbf{h} = \mathbf{h}^*, \mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]\ \Pr[\mathbf{h} = \mathbf{h}^*, \mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]}{\Pr[\mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]}\, dV_{\mathbf{h}} = \\
&\int_{\mathbb{R}_+^N} \Pr[Z_t = +1|\mathbf{h} = \mathbf{h}^*, \mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]\ \Pr[\mathbf{h} = \mathbf{h}^*|\mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]\, dV_{\mathbf{h}} = \\
&\int_{\mathbf{h}\ \mathbf{p}^\intercal \leq 1} \Pr[Z_t = +1|\mathbf{h} = \mathbf{h}^*, \mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]\ \Pr[\mathbf{h} = \mathbf{h}^*|\mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]\, dV_{\mathbf{h}} + \\
&\int_{\mathbf{h}\ \mathbf{p}^\intercal > 1} \Pr[Z_t = +1|\mathbf{h} = \mathbf{h}^*, \mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]\ \Pr[\mathbf{h} = \mathbf{h}^*|\mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]\, dV_{\mathbf{h}} = \\
&\int_{\mathbf{h}\ \mathbf{p}^\intercal \leq 1} \Pr[Z_t = +1|\mathbf{h} = \mathbf{h}^*, \mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]\ \Pr[\mathbf{h} = \mathbf{h}^*|Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]\, dV_{\mathbf{h}} + \\
&\int_{\mathbf{h}\ \mathbf{p}^\intercal > 1} \Pr[Z_t = +1|\mathbf{h} = \mathbf{h}^*, \mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]\ \Pr[\mathbf{h} = \mathbf{h}^*|Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]\, dV_{\mathbf{h}} = \\
&\int_{\mathbf{h}\ \mathbf{p}^\intercal \leq 1} \Pr[Z_t = +1|\mathbf{h} = \mathbf{h}^*, \mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]\ f_t(\mathbf{h})\, dV_{\mathbf{h}} + \\
&\int_{\mathbf{h}\ \mathbf{p}^\intercal > 1} \Pr[Z_t = +1|\mathbf{h} = \mathbf{h}^*, \mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}]\ f_t(\mathbf{h})\, dV_{\mathbf{h}} = \\
&\int_{\mathbf{h}\ \mathbf{p}^\intercal \leq 1} \Pr[Z_t = +1|\mathbf{h} = \mathbf{h}^*, \mathbf{p}(t)]\ f_t(\mathbf{h})\, dV_{\mathbf{h}} + \\
&\int_{\mathbf{h}\ \mathbf{p}^\intercal > 1} \Pr[Z_t = +1|\mathbf{h} = \mathbf{h}^*, \mathbf{p}(t)]\ f_t(\mathbf{h})\, dV_{\mathbf{h}} = \\
&\int_{\mathbf{h}\ \mathbf{p}^\intercal \leq 1} P_{cd}\ f_t(\mathbf{h})\, dV_{\mathbf{h}} + \int_{\mathbf{h}\ \mathbf{p}^\intercal > 1} (1 - P_{cd})\ f_t(\mathbf{h})\, dV_{\mathbf{h}} = \\
&C_t(\mathbf{p}(t))\ P_{cd} + (1 - C_t(\mathbf{p}(t)))\ (1 - P_{cd})
\end{aligned}
\tag{A.4}
$$

which in the spirit of [97] is a function of $\mathbf{p}(t)$, $\gamma_t(\mathbf{p}(t))$, and can be written as:

$$
\gamma_t(\mathbf{p}(t)) = C_t(\mathbf{p}(t))\ P_{cd} + (1 - C_t(\mathbf{p}(t)))\ (1 - P_{cd}). \tag{A.5}
$$

A similar expression can also be derived for the $Z_t = -1$ event:

$$
\Pr[Z_t = -1|\mathbf{p}(t), Z_{0:(t-1)}, \mathbf{p}_{0:(t-1)}] = 1 - \gamma_t(\mathbf{p}(t)). \tag{A.6}
$$

After all this preliminary work, we continue with our original goal of this proof, which is to find the optimal probing rule that maps $\{f_0, .., f_{N_T-1}\}$ to $\{\mathbf{p}(0), .., \mathbf{p}(N_T-1)\}$ in order to achieve the maximum average entropy reduction from the $f_0(\mathbf{h})$ to the $f_{N_T}(\mathbf{h})$ pdf. Based on the Dynamic Programming (DP) proof of the univariate case [97], our goal expressed in a formal manner is to seek the optimal probing design policy $\pi^*_{0:(N_T-1)} = \{\mathbf{p}(0) = \mu^*(f_0), .., \mathbf{p}(N_T-1) = \mu^*(f_{N_T-1})\}$ which maximizes the expected information gain over all possible feedback sequences derived by this policy π^*:

$$\pi^*_{0:(N_T-1)} = \arg\max J_{\pi_{0:(N_T-1)}}. \tag{A.7}$$

The expected information gain of an arbitrary policy π, $J_{\pi_{0:(N_T-1)}}$, is expressed as the conditional expectation:

$$J_{\pi_{0:(N_T-1)}} = E^{\pi}[\mathcal{H}(f_0) - \mathcal{H}(f_{N_T})|\mathbf{p}(N_T-1), f_{N_T-1}] \tag{A.8}$$

where $\mathcal{H}$ is the entropy operator of a pdf. To create a multistage version of our objective in the DP spirit [116], we rewrite (A.8) as:

$$\begin{aligned} &J_{\pi_{0:(N_T-1)}} = E^{\pi}[\mathcal{H}(f_0) - \mathcal{H}(f_{N_T})|\mathbf{p}(N_T-1), f_{N_T-1}] = \\ &E^{\pi}[\mathcal{H}(f_0) - \mathcal{H}(f_1) + ... \\ &+ \mathcal{H}(f_{N_T-1}) - \mathcal{H}(f_{N_T})|\mathbf{p}(N_T-1), f_{N_T-1}] = \\ &E^{\pi}\left[\sum_{k=0}^{N_T-1}(\mathcal{H}(f_k) - \mathcal{H}(f_{k+1})|\mathbf{p}(k), f_k)\right] \end{aligned} \tag{A.9}$$

where we added and subtracted all the entropy terms of the intermediate pdf's to form an additive gain over time.

To solve (A.7), DP is using a backward induction logic, where if we define $J_{\pi_{k:(N_T-1)}}$ as:

$$\begin{aligned} &J_{\pi_{k:(N_T-1)}} = E^{\pi}\left[\sum_{k}^{N_T-1}(\mathcal{H}(f_k) - \mathcal{H}(f_{k+1})|\mathbf{p}(k), f_k)\right] = \\ &E^{\pi}\left[\mathcal{H}(f_k) - \mathcal{H}(f_{k+1})|\mathbf{p}(k), f_k\right] + J_{\pi_{(k+1):(N_T-1)}} \end{aligned} \tag{A.10}$$

then we first need to solve:

$$\begin{aligned} &\max_{\mu} J_{\pi_{N_T-1}} = \\ &\max_{\mu} E^{\pi}\left[\mathcal{H}(f_{N_T-1}) - \mathcal{H}(f_{N_T})|\mathbf{p}(N_T-1), f_{N_T-1}\right] \end{aligned} \tag{A.11}$$

and secondly we must solve:

$$\begin{aligned}&\max_{\mu} J_{\pi_{k:(N_T-1)}} = \\ &\max_{\mu} E^{\pi}\left[\mathcal{H}(f_k) - \mathcal{H}(f_{k+1})|\mathbf{p}(k), f_k\right] + J_{\pi^*_{(k+1):(N_T-1)}}\end{aligned} \tag{A.12}$$

for an arbitrary $k \in \{0, .., (N_T - 2)\}$. If the two resulting rules, meaning the functions which assign a probing power vector, our decision, to a posterior pdf, our state, are identical, then by induction we may say that this is the optimal design policy μ^* and that it satisfies (A.7).

Now, let us start from solving (A.11). Using the updating equation (5.13), we evaluate $E^{\pi}\left[\mathcal{H}(f_{N_T-1}) - \mathcal{H}(f_{N_T})|\mathbf{p}(N_T-1), f_{N_T-1}\right]$ over the two possible events $Z_{N_T-1} = +1$ and $Z_{N_T-1} = -1$:

$$\begin{aligned}&E^{\pi}\left[\mathcal{H}(f_{N_T-1}) - \mathcal{H}(f_{N_T})|\mathbf{p}(N_T-1), f_{N_T-1}\right] = \\ &E^{\pi}\left[E_{\mathbf{h}}\left[-\log(f_{N_T-1})\right]\right] - E^{\pi}\left[E_{\mathbf{h}}\left[-\log(f_{N_T-1})\right]\right] - \\ &E^{\pi}\left[E_{\mathbf{h}}\left[-\log(\Pr[Z_{N_T-1}|\mathbf{h}=\mathbf{h}^*, \mathbf{p}(N_T-1)])\right]|\mathbf{p}(N_T-1), f_{N_T-1}\right] + \\ &E^{\pi}\left[E_{\mathbf{h}}\left[-\log(\Pr[Z_{N_T-1}|\mathbf{p}(N_T-1), f_{N_T-1}])\right]|\mathbf{p}(N_T-1), f_{N_T-1}\right].\end{aligned} \tag{A.13}$$

The last two remaining terms can be further processed. With the help of (5.11) for $\Pr[Z_{N_T-1}|\mathbf{h}=\mathbf{h}^*, \mathbf{p}(N_T-1)]$, the third term can be analyzed as:

$$\begin{aligned}&E^{\pi}\left[E_{\mathbf{h}}\left[-\log(\Pr[Z_{N_T-1}|\mathbf{h}=\mathbf{h}^*, \mathbf{p}(N_T-1)])\right]|\mathbf{p}(N_T-1), f_{N_T-1}\right] = \\ &E^{\pi}\left[E_{\mathbf{h}}\left[-\log(\Pr[Z_{N_T-1}|\mathbf{h}=\mathbf{h}^*, \mathbf{p}(N_T-1)])\right]|f_{N_T-1}\right] = \\ &E^{\pi}\left[-\log(\Pr[Z_{N_T-1}|\mathbf{h}=\mathbf{h}^*, \mathbf{p}(N_T-1)])\right] = \\ &-P_{cd}\log(P_{cd}) - (1-P_{cd})\log(1-P_{cd})\end{aligned} \tag{A.14}$$

since Z_{N_T-1} does not depend on f_{N_T-1} given $\mathbf{h} = \mathbf{h}^*$ and $\mathbf{p}(N_T-1)$. The operator $E^{\pi}[.]$ is basically the expectation over the two possible observations $Z_{N_T-1} = +1$ and $Z_{N_T-1} = -1$. This result tells us that this term does not depend on the design of $\mathbf{p}(N_T-1)$ and therefore it does not participate in the maximization of $J_{\pi_{(N_T-1)}}$. Additionally, by using (A.5) and (A.6) which again lead us to omit $E_{\mathbf{h}}$, since $\Pr[Z_{N_T-1}|\mathbf{p}(N_T-1), f_{N_T-1}]$ is

stable over the $\mathbf{h}$ domain, the fourth term becomes:

$$
\begin{aligned}
&E^{\pi}\left[E_{\mathbf{h}}\left[-\log(\Pr[Z_{N_T-1}|\mathbf{p}(N_T-1), f_{N_T-1}])\right]|\mathbf{p}(N_T-1), f_{N_T-1}\right] = \\
&E^{\pi}\left[-\log(\Pr[Z_{N_T-1}|\mathbf{p}(N_T-1), f_{N_T-1}])|\mathbf{p}(N_T-1), f_{N_T-1}\right] = \\
&E^{\pi}\left[-\log(\Pr[Z_{N_T-1}|\mathbf{p}(N_T-1), f_{N_T-1}])\right] = \\
&-\gamma_{N_T-1}(\mathbf{p}(N_T-1))\log(\gamma_{N_T-1}(\mathbf{p}(N_T-1)))- \\
&(1-\gamma_{N_T-1}(\mathbf{p}(N_T-1)))\log((1-\gamma_{N_T-1}(\mathbf{p}(N_T-1)))).
\end{aligned}
\tag{A.15}
$$

After elaborating on $J_{\pi_{(N_T-1)}}$, we reached the conclusion that the probing rule that maximizes $J_{\pi_{(N_T-1)}}$ is achieved by maximizing the quantity derived from (A.15), which occurs for $\gamma_{N_T-1}(\mathbf{p}(N_T-1)) = 1/2$ and consequently for $C_{N_T-1}(\mathbf{p}(N_T-1)) = 1/2$. The same result is derived from solving (A.12) with a similar analysis which delivers that the optimal $\mathbf{p}(k)$ design rule for maximizing $J_{\pi_{k:(N_T-1)}}$ is $C_k(\mathbf{p}(k)) = 1/2$. Thus, the overall design policy that solves (A.7) is $\mathbf{p}(k) = \mu^*(f_k) = C_k^{-1}(1/2) = \mathbf{p}_{med}(k)$. This result indicates that in order to reach as fast as possible our learning solution, the probing power vectors should always be chosen as median regressors (bisectors) of the current posterior pdf estimate, $\{\mathbf{p}(0), .., \mathbf{p}(N_T-1)\} = \{\mathbf{p}_{med}(0), .., \mathbf{p}_{med}(N_T-1)\}$.

Appendix B

Moments of a one side truncated multivariate normal pdf

Assuming a multivariate normal pdf $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\mathbf{x}, \boldsymbol{\Sigma}_\mathbf{x})$ of N dimensions and a halfspace indicator function:

$$g(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{a}\,\mathbf{x}^\intercal \leq b \\ 0 & \text{if } \mathbf{a}\,\mathbf{x}^\intercal > b \end{cases} \tag{B.1}$$

where $\mathbf{a}\,\mathbf{x}^\intercal = b$ is the hyperplane limit of this halfspace and $\mathbf{a}$ and $\mathbf{x}$ are row vectors, then $h(\mathbf{x}) = g(\mathbf{x})\ \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\mathbf{x}, \boldsymbol{\Sigma}_\mathbf{x})$ is an un-normalized one side truncated multivariate normal pdf. Next, we determine the 0_{th}, 1_{st} and 2_{nd} moments of $h(\mathbf{x})$, q, $\mathbf{q}$ and $\mathbf{Q}$, based on the moment related integrals, c, $\mathbf{c}$ and $\mathbf{C}$:

$$c = \int_{\mathbb{R}^N} h(\mathbf{x})\ dV_\mathbf{x} = \int_{\mathbf{a}\,\mathbf{x}^\intercal \leq b} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\mathbf{x}, \boldsymbol{\Sigma}_\mathbf{x})\ dV_\mathbf{x} \tag{B.2}$$

$$\mathbf{c} = \int_{\mathbb{R}^N} \mathbf{x}\ h(\mathbf{x})\ dV_\mathbf{x} = \int_{\mathbf{a}\,\mathbf{x}^\intercal \leq b} \mathbf{x}\ \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\mathbf{x}, \boldsymbol{\Sigma}_\mathbf{x})\ dV_\mathbf{x} \tag{B.3}$$

$$\mathbf{C} = \int_{\mathbb{R}^N} \mathbf{x}^\intercal\mathbf{x}\ h(\mathbf{x})\ dV_\mathbf{x} = \int_{\mathbf{a}\,\mathbf{x}^\intercal \leq b} \mathbf{x}^\intercal\mathbf{x}\ \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\mathbf{x}, \boldsymbol{\Sigma}_\mathbf{x})\ dV_\mathbf{x}. \tag{B.4}$$

Note that c is a constant which represents the mass or the normalization factor of $h(\mathbf{x})$, $\mathbf{c}$ is a vector of integrals and $\mathbf{C}$ is a matrix of integrals. The moments can be written as:

$$q = c \tag{B.5}$$

$$\mathbf{q} = \frac{\mathbf{c}}{c} \tag{B.6}$$

$$\mathbf{Q} = \frac{\mathbf{C}}{c} - \mathbf{q}^{\intercal}\mathbf{q}. \tag{B.7}$$

The problem of computing these moments lies on the computation of the integrals in (B.2), (B.3) and (B.4).

Now, if we define an $N \times N$ transformation matrix T such as:

$$T = \begin{bmatrix} a_1 & 0 & 0 & \dots & 0 & 0 \\ a_2 & 1 & 0 & \dots & 0 & 0 \\ a_3 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{N-1} & 0 & 0 & \dots & 1 & 0 \\ a_N & 0 & 0 & \dots & 0 & 1 \end{bmatrix} \tag{B.8}$$

and determine a new random variable $\mathbf{y} = \mathbf{x}\,T$, then $\mathbf{y}$ will also be normally distributed, $\mathbf{y} \sim \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\mathbf{y}}, \boldsymbol{\Sigma}_{\mathbf{y}})$, where:

$$\boldsymbol{\mu}_{\mathbf{y}} = \boldsymbol{\mu}_{\mathbf{x}}\,T \tag{B.9}$$

and

$$\boldsymbol{\Sigma}_{\mathbf{y}} = T^{\intercal}\boldsymbol{\Sigma}_{\mathbf{x}}T. \tag{B.10}$$

This helps us transform the integrals in (B.2), (B.3) and (B.4) by using the change-of-variables technique. The Jacobian matrix $J_{\mathbf{x}\to\mathbf{y}}$ is actually equal to $T^{\intercal}$, hence the infinitesimal volume $dV_{\mathbf{x}}$ can be rewritten as $\frac{dV_{\mathbf{y}}}{|det(T^{\intercal})|}$ or $\frac{dV_{\mathbf{y}}}{|det(T)|}$. Using this and changing the integral limits delivers the following for (B.2):

$$\begin{aligned} c = & \int\limits_{\mathbf{a}\,\mathbf{x}^{\intercal}\leq b} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}})\, dV_{\mathbf{x}} = \\ & \int\limits_{y_1\leq b} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}})\, \frac{dV_{\mathbf{y}}}{|det(T)|} = \\ & \int_{-\infty}^{y_1=b} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \frac{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}})}{|det(T)|}\, dV_{\mathbf{y}} = \\ & \int_{-\infty}^{y_1=b} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\mathbf{y}}, \boldsymbol{\Sigma}_{\mathbf{y}})\, dV_{\mathbf{y}} \end{aligned} \tag{B.11}$$

where in the last line we used the relation of the two pdf's of the random variables $\mathbf{x}$ and $\mathbf{y}$. Similarly, for $\mathbf{c}$ and $\mathbf{C}$, we have:

$$\mathbf{c} = \left(\int_{-\infty}^{y_1=b} \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} \mathbf{y}\, \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_\mathbf{y}, \boldsymbol{\Sigma}_\mathbf{y})\, dV_\mathbf{y} \right) T^{-1} \tag{B.12}$$

and

$$\mathbf{C} = (T^{-1})^\intercal \left(\int_{-\infty}^{y_1=b} \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} \mathbf{y}^\intercal \mathbf{y}\, \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_\mathbf{y}, \boldsymbol{\Sigma}_\mathbf{y})\, dV_\mathbf{y} \right) T^{-1}. \tag{B.13}$$

Consequently, the problem of calculating the moments of a one side truncated multivariate Gaussian pdf has been transformed into calculating the moments of another one side truncated multivariate Gaussian pdf where the truncation occurs vertically to the axis $y_1 y_1'$. This is the study object of Appendix C.

Appendix C

Moments of a one vertical side truncated multivariate normal pdf

In this section, we elaborate on the moments of one vertical side truncated multivariate normal pdf's. In the statistics literature, the truncation subject has been extensively investigated using many kinds of truncations, such as box-like and elliptical ones. Here, we present a simplified case of calculating the moments of a doubly truncated multivariate normal pdf recently studied in [115] and which actually concerns a hyper-rectangle truncation. The simplification introduced here will lead us to computing the moments of the one vertical side truncated multivariate normal pdf. Assuming a multivariate normal pdf $\mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$ in N dimensions and a hyper-rectangle defined by the inequalities $a_i \leq x_i \leq b_i$ for $i = 1,..,N$, the authors of [115] managed to find simple recursive relations for the moment related integrals and therefore allow the fast computation of doubly truncated multivariate normal pdf's moments.

More specifically, if $\mathbf{a} = [a_1, ..., a_N]$ and $\mathbf{b} = [b_1, ..., b_N]$, then $L_{\mathbf{k}}(\mathbf{a},\mathbf{b};\boldsymbol{\mu},\boldsymbol{\Sigma})$ is the integral defined as:

$$L_{\mathbf{k}}(\mathbf{a},\mathbf{b};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \int_{a_1}^{b_1} ... \int_{a_N}^{b_N} \mathbf{x}^{\mathbf{k}}\, \mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma})\, dV_{\mathbf{x}} \tag{C.1}$$

where $\mathbf{x}^{\mathbf{k}}$ stands for $x_1^{k_1} \cdot ... \cdot x_N^{k_N}$. For example, if we wish to compute the integral $\int_{a_1}^{b_1} ... \int_{a_4}^{b_4} x_1 x_3\, \mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma})\, dV_{\mathbf{x}}$ for $N = 4$, then $\mathbf{k} = [1,0,1,0]$. Additionally, we denote by $\mathbf{r}_{(i)}$ a row vector $\mathbf{r}$ with its i_{th} element removed, by $\mathbf{R}_{i,(j)}$ the i_{th} row of a matrix $\mathbf{R}$ with its j_{th} element removed, by $\mathbf{R}_{(i),j}$ the j_{th} column of a matrix $\mathbf{R}$ with its i_{th} element removed and by $\mathbf{R}_{(i),(j)}$ a matrix $\mathbf{R}$ with its i_{th} row and j_{th} column removed. In [115], it is shown that if we let $\mathbf{e}_i$ denote an N-dimensional row vector with its i_{th} element equal to one and zeros otherwise, then:

$$L_{\mathbf{k}+\mathbf{e}_i}(\mathbf{a},\mathbf{b};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mu_i L_{\mathbf{k}}(\mathbf{a},\mathbf{b};\boldsymbol{\mu},\boldsymbol{\Sigma}) + \mathbf{e}_i \boldsymbol{\Sigma} \mathbf{c}_{\mathbf{k}}^{\intercal} \tag{C.2}$$

where $\mathbf{c_k}$ is an N-dimensional row vector with its j_{th} element equal to:

$$\begin{aligned}\mathbf{c_{k},}_j &= k_j \; L_{\mathbf{k}-\mathbf{e}_j}(\mathbf{a},\mathbf{b};\boldsymbol{\mu},\boldsymbol{\Sigma}) + \\ &+ a_j^{k_j} \; \mathcal{N}(a_j;\mu_j,\boldsymbol{\Sigma}_{j,j}) \; L_{\mathbf{k}_{(j)}}(\mathbf{a}_{(j)},\mathbf{b}_{(j)};\tilde{\boldsymbol{\mu}}_j^{\mathbf{a}},\tilde{\boldsymbol{\Sigma}}_j) + \\ &+ b_j^{k_j} \; \mathcal{N}(b_j;\mu_j,\boldsymbol{\Sigma}_{j,j}) \; L_{\mathbf{k}_{(j)}}(\mathbf{a}_{(j)},\mathbf{b}_{(j)};\tilde{\boldsymbol{\mu}}_j^{\mathbf{b}},\tilde{\boldsymbol{\Sigma}}_j)\end{aligned} \tag{C.3}$$

and

$$\tilde{\boldsymbol{\mu}}_j^{\mathbf{a}} = \boldsymbol{\mu}_{(j)} + \boldsymbol{\Sigma}_{j,(j)} \frac{a_j - \mu_j}{\boldsymbol{\Sigma}_{j,j}} \tag{C.4}$$

$$\tilde{\boldsymbol{\mu}}_j^{\mathbf{b}} = \boldsymbol{\mu}_{(j)} + \boldsymbol{\Sigma}_{j,(j)} \frac{b_j - \mu_j}{\boldsymbol{\Sigma}_{j,j}} \tag{C.5}$$

$$\tilde{\boldsymbol{\Sigma}}_j = \boldsymbol{\Sigma}_{(j),(j)} - \frac{\boldsymbol{\Sigma}_{(j),j} \; \boldsymbol{\Sigma}_{j,(j)}}{\boldsymbol{\Sigma}_{j,j}}. \tag{C.6}$$

Hence, if we intend to obtain the integrals $\int_{a_1}^{b_1} ... \int_{a_N}^{b_N} x_m \, \mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) \, dV_{\mathbf{x}}$ for $m = 1, ..., N$ and calculate the mean of a doubly truncated multivariate normal pdf, then we should set $\mathbf{k} = \mathbf{0}$ and $\mathbf{e}_i = \mathbf{e}_m$ in (C.2). Next, we should divide the results with the normalization constant of the truncated Gaussian $\int_{a_1}^{b_1} ... \int_{a_N}^{b_N} \mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) \, dV_{\mathbf{x}}$, which in [115] is calculated using the inclusion-exclusion principle, a combinatorics technique. Similarly, for the 2_{nd} order moment, we are interested in computing integrals of the form $\int_{a_1}^{b_1} ... \int_{a_N}^{b_N} x_m x_n \, \mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) \, dV_{\mathbf{x}}$ for $m = 1, ..., N$ and $n = 1, ..., N$ which can be acquired by setting $\mathbf{k} = \mathbf{e}_m$ and $\mathbf{e}_i = \mathbf{e}_n$ in (C.2).

Now, if we let $\mathbf{a}$ and $\mathbf{b}$, which define the box-like truncation, be respectively $[-\infty, ..., -\infty]$ and $[b_1, \infty, ..., \infty]$, then the aforementioned recursive relations concern the moments of a one vertical side truncated multivariate normal pdf, where the cutting hyperplane is $x_1 = b_1$ and the hyper-rectangle is now the halfspace $x_1 \leq b_1$. The relations (C.2), (C.3), (C.4), (C.5) and (C.6) are simplified and moreover we have the benefit of not using the inclusion-exclusion principle, which for large N can be computationally demanding, for the calculation of the mass of the truncated $\mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$. This happens because $\int_{-\infty}^{b_1} \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} \mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) \, dV_{\mathbf{x}}$ is actually equal to $\int_{-\infty}^{b_1} \mathcal{N}(x_1;\mu_1,\boldsymbol{\Sigma}_{1,1}) \, dx_1$.

Bibliography

[1] FCC, "Spectrum policy task force report," *ET Docket 02-155*, 2002.

[2] Q. Zhao and B. Sadler, "A Survey of Dynamic Spectrum Access," *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 79–89, May 2007.

[3] J. Mitola, "Cognitive radio an integrated agent architecture for software defined radio," Ph.D. dissertation, KTH Royal Institute of Technology Stockholm, Stockholm, Sweden, 2000.

[4] S. Haykin, "Cognitive Radio: Brain-Empowered Wireless Communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

[5] A. Goldsmith, S. Jafar, I. Maric, and S. Srinivasa, "Breaking Spectrum Gridlock With Cognitive Radios: An Information Theoretic Perspective," *Proc. IEEE*, vol. 97, no. 5, pp. 894–914, May 2009.

[6] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks Journal (Elsevier)*, vol. 50, pp. 2127–2159, Sept. 2006.

[7] G. Zhao, G. Y. Li, and C. Yang, "Proactive Detection of Spectrum Opportunities in Primary Systems with Power Control," *IEEE Trans. Wireless Commun.*, vol. 8, no. 9, pp. 4815–4823, Sept. 2009.

[8] A. Tsakmalis, S. Chatzinotas, and B. Ottersten, "Automatic Modulation Classification for Adaptive Power Control in Cognitive Satellite Communications," in *Proc. 7th ASMS Conf. and 13th SPSC Workshop*, Sept. 2014, pp. 234–240.

[9] ——, "Modulation and Coding Classification for Adaptive Power Control in 5G Cognitive Communications," in *Proc. IEEE 14th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun. 2014, pp. 234–238.

[10] ——, "Centralized Power Control in Cognitive Radio Networks Using Modulation and Coding Classification Feedback," *IEEE Trans. Cognitive Commun. and Networking*, vol. 2, no. 3, pp. 223–237, Sept. 2016.

[11] ——, "Power Control in Cognitive Radio Networks Using Cooperative Modulation and Coding Classification," in *Proc. 10th Int. Conf. on Cognitive Radio Oriented Wireless Netw. (CROWNCOM)*, Apr. 2015.

[12] ——, "Active Interference Constraint Learning with Uncertain Feedback for Cognitive Radio Networks," in *Proc. IEEE International Conf. on Commun. (ICC)*, May 2016.

[13] ——, "Interference Constraint Active Learning with Uncertain Feedback for Cognitive Radio Networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4654–4668, Jul. 2017.

[14] M. Bkassiny, S. K. Jayaweera, Y. Li, and K. A. Avery, "Wideband Spectrum Sensing and Non-Parametric Signal Classification for Autonomous Self-Learning Cognitive Radios," *IEEE Trans. Wireless Commun.*, pp. 2596–2605, 2012.

[15] M. Bkassiny, S. K. Jayaweera, and Y. Li, "Multidimensional Dirichlet Process-Based Non-Parametric Signal Classification for Autonomous Self-Learning Cognitive Radios," *IEEE Trans. Wireless Commun.*, vol. 12, no. 11, pp. 5413–5423, Nov. 2013.

[16] H. Yang, X. Xie, and R. Wang, "SOM-GA-SVM Detection Based Spectrum Sensing in Cognitive Radio," *8th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, 2012.

[17] K. M. Thilina, K. W. Choi, N. Saquib, and E. Hossain, "Pattern classification techniques for cooperative spectrum sensing in cognitive radio networks: SVM and W-KNN approaches," *IEEE Global Communications Conference (GLOBECOM)*, pp. 1260–1265, 2012.

[18] J. J. Popoola and R. van Olst, "A Novel Modulation-Sensing Method," *IEEE Veh. Technol. Magazine*, vol. 6, no. 3, pp. 60–69, Aug. 2011.

[19] M. Petrova, P. Mahonen, and A. Osuna, "Multi-class classification of analog and digital signals in cognitive radios using Support Vector Machines," *7th International Symposium on Wireless Communication Systems (ISWCS)*, pp. 986–990, 2010.

[20] H.-C. Wu, M. Saquib, and Z. Yun, "Novel Automatic Modulation Classification Using Cumulant Features for Communications via Multipath Channels," *IEEE Trans. Wireless Commun.*, vol. 7, no. 8, pp. 1536–1276, Aug. 2008.

[21] A. F. Cattoni, M. Ottonello, M. Raffetto, and C. S. Regazzoni, "Neural Networks Mode Classification based on Frequency Distribution Features," *2nd International*

Conference on Cognitive Radio Oriented Wireless Networks and Communications, pp. 251–257, 2007.

[22] J. Lunden and V. Koivunen, "Automatic Radar Waveform Recognition," *IEEE J. Sel. Topics Signal Process.*, p. 124–136, 2007.

[23] N. Shetty, S. Pollin, and P. Paweczak, "Identifying Spectrum Usage by Unknown Systems using Experiments in Machine Learning," *IEEE Wireless Communications and Networking Conference (WCNC)*, 2009.

[24] W. A. Gardner and C. M. Spooner, "The Cumulant Theory of Cyclostationary Time-Series, Part I: Foundation," *IEEE Trans. Signal Process.*, vol. 42, no. 12, pp. 3387–3408, Dec. 1994.

[25] ——, "The Cumulant Theory of Cyclostationary Time-Series, Part II: Development and Applications," *IEEE Trans. Signal Process.*, vol. 42, no. 12, pp. 3409–3429, Dec. 1994.

[26] B. Ramkumar, "Automatic Modulation Classification for Cognitive Radios Using Cyclic Feature Detection," *IEEE Circuits and Syst. Mag.*, vol. 9, no. 2, pp. 27–45, 2nd Quarter 2009.

[27] K. Kim, I. A. Akbar, K. K. Bae, J. Um, C. M. Spooner, and J. H. Reed, "Cyclostationary Approaches to Signal Detection and Classification in Cognitive Radio," *2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 212–215, 2007.

[28] A. Fehske, J. Gaeddert, and J. H. Reed, "A new approach to signal classification using spectral correlation and neural networks," *1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 144–150, 2005.

[29] H. Hu, J. Song, and Y. Wang, "Signal Classification based on Spectral Correlation Analysis and SVM in Cognitive Radio," *22nd International Conference on Advanced Information Networking and Applications (AINA)*, pp. 883–887, 2008.

[30] Y. Tang, Q. Zhang, and W. Lin, "Artificial Neural Network Based Spectrum Sensing Method for Cognitive Radio," *6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, 2010.

[31] A. Al-Habashna, O. A. Dobre, R. Venkatesan, and D. C. Popescu, "Second-Order Cyclostationarity of Mobile WiMAX and LTE OFDM Signals and Application to Spectrum Awareness in Cognitive Radio Systems," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 1, pp. 26–42, Feb. 2012.

[32] M. Marey, O. Dobre, and R. Inkol, "Classification of Space-Time Block Codes Based on Second-Order Cyclostationarity with Transmission Impairments," *IEEE Trans. Wireless Commun.*, pp. 2574–2574, 2012.

[33] A. Punchihewa, Q. Zhang, O. A. Dobre, C. M. Spooner, S. Rajan, and R. Inkol, "On the Cyclostationarity of OFDM and Single Carrier Linearly Digitally Modulated Signals in Time Dispersive Channels: Theoretical Developments and Application," *IEEE Trans. Wireless Commun.*, vol. 9, no. 8, pp. 2588–2599, Aug. 2010.

[34] O. A. Dobre, M. Oner, S. Rajan, and R. Inkol, "Cyclostationarity-Based Robust Algorithms forQAM Signal Identification," *IEEE Commun. Lett.*, vol. 16, no. 1, pp. 12–15, Jan. 2012.

[35] O. Dobre, Y. Bar-Ness, and W. Su, "Higher-order cyclic cumulants for high order modulation classification," *Proc. IEEE MILCOM*, p. 112–117, 2003.

[36] C. M. Spooner, "Classification of Co-Channel Communication Signals Using Cyclic Cumulants," in *Proc. 29th Asilomar Conf. on Signals, Syst. and Computers*, Oct. 1995, pp. 531–536.

[37] H.-C. Wu, M. Saquib, and Z. Yun, "Novel Automatic Modulation Classification Using Cumulant Features for Communications via Multipath Channels," *IEEE Trans. Wireless Commun.*, pp. 3098–3105, 2008.

[38] V. Choquese, M. Marazin, L. Collin, K. Yao, and G. Burel, "Blind Recognition of Linear Space-Time Block Codes: A Likelihood-Based Approach," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1290–1299, Mar. 2010.

[39] T. Xia and H. Wu, "Novel Blind Identification of LDPC Codes Using Average LLR of Syndrome a Posteriori Probability," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 632–640, Feb. 2014.

[40] V. N. Vapnik, *The Nature of Statistical Learning Theory.* New York: Springer-Verlag, 2000.

[41] Y. A. Eldemerdash, M. Marey, O. A. Dobre, G. K. Karagiannidis, and R. Inkol, "Fourth-Order Statistics for Blind Classification of Spatial Multiplexing and Alamouti Space-Time Block Code Signals," *IEEE Trans. Commun.*, vol. 61, no. 6, pp. 2420–2431, Jun. 2013.

[42] F. Hameed, O. A. Dobre, and D. C. Popescu, "On the Likelihood-Based Approach to Modulation Classification," *IEEE Trans. Wireless Commun.*, vol. 8, no. 12, pp. 1536–1276, Dec. 2009.

[43] O. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET Communications*, vol. 1, no. 2, p. 137–156, Apr. 2007.

[44] R. Moosavi and E. Larsson, "A Fast Scheme for Blind Identification of Channel Codes," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2011, pp. 1–5.

[45] F. Wang, J. Jianwei Huang, and Y. Zhao, "Delay Sensitive Communications over Cognitive Radio Networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 4, pp. 1402–1411, Apr. 2012.

[46] Q. Zhao, S. Geirhofer, L. Tong, and B. M. Sadler, "Opportunistic Spectrum Access via Periodic Channel Sensing," *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 785–796, Feb. 2008.

[47] Y. Li, S. Jayaweera, M. Bkassiny, and K. Avery, "Optimal Myopic Sensing and Dynamic Spectrum Access in Cognitive Radio Networks with Low-Complexity Implementations," *IEEE Trans. Wireless Commun.*, vol. 11, no. 7, pp. 2412–2423, Jul. 2012.

[48] L. Lopez-Ramos, A. Marques, and J. Ramos, "Jointly Optimal Sensing and Resource Allocation for Multiuser Interweave Cognitive Radios," *IEEE Trans. Wireless Commun.*, vol. 13, no. 11, pp. 5954–5967, Nov. 2014.

[49] Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized Cognitive MAC for Opportunistic Spectrum Access in Ad Hoc Networks: A POMDP Framework," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 3, pp. 589–600, Apr. 2007.

[50] K. W. Choi and E. Hossain, "Opportunistic Access to Spectrum Holes Between Packet Bursts: A Learning-Based Approach," *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2497–2509, Aug. 2011.

[51] J. Nie and S. Haykin, "A Q-Learning-Based Dynamic Channel Assignment Technique for Mobile Communication Systems," *IEEE Trans. Veh. Technol.*, vol. 48, no. 5, pp. 1676–1687, Sep. 1999.

[52] ——, "A Dynamic Channel Assignment Policy Through Q-Learning," *IEEE Trans. Neural Netw.*, vol. 10, no. 6, pp. 1443–1455, Nov. 1999.

[53] D. V. Djonin and V. Krishnamurthy, "Q-Learning Algorithms for Constrained Markov Decision Processes With Randomized Monotone Policies: Application to MIMO Transmission Control," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 2170–2181, May 2007.

[54] M. Levorato, S. Firouzabadi, and A. Goldsmith, "A Learning Framework for Cognitive Interference Networks with Partial and Noisy Observations," *IEEE Trans. Wireless Commun.*, vol. 11, no. 9, pp. 3101–3111, Sep. 2012.

[55] Y. Zhang, F. Fu, , and M. van der Schaar, "On-Line Learning and Optimization for Wireless Video Transmission," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3108–3124, Jun. 2010.

[56] K. T. Phan, T. Le-Ngoc, M. van der Schaar, and F. Fu, "Optimal Scheduling over Time-Varying Channels with Traffic Admission Control: Structural Results and Online Learning Algorithms," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4434–4444, Sep. 2013.

[57] W. Zhong, G. Chen, S. Jin, and K. K. Wong, "Relay Selection and Discrete Power Control for Cognitive Relay Networks via Potential Game," *IEEE Trans. Signal Process.*, vol. 62, no. 20, pp. 5411–5424, Oct. 2014.

[58] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, , and Y. D. Yao, "Opportunistic Spectrum Access in Unknown Dynamic Environment: A Game-Theoretic Stochastic Learning Solution," *IEEE Trans. Wireless Commun.*, vol. 11, no. 4, pp. 1380–1391, Apr. 2012.

[59] P. Zhou, Y. Chang, and J. Copeland, "Reinforcement Learning for Repeated Power Control Game in Cognitive Radio Networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 1, pp. 54–69, Jan. 2012.

[60] D. Niyato and E. Hossain, "Dynamics of Network Selection in Heterogeneous Wireless Networks: An Evolutionary Game Approach," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 2008–2017, May 2009.

[61] A. Galindo-Serrano and L. Giupponi, "Distributed Q-Learning for Aggregated Interference Control in Cognitive Radio Networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 4, pp. 1823–1834, May 2010.

[62] F. Fu and M. van der Schaar, "Learning to Compete for Resources in Wireless Stochastic Games," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 1904–1919, May 2009.

[63] H. P. Shiang and M. van der Schaar, "Distributed Resource Management in Multihop Cognitive Radio Networks for Delay-Sensitive Transmission," *IEEE Trans. Veh. Technol.*, vol. 58, no. 2, pp. 941 – 953, Feb. 2009.

[64] M. A. Khan, H. Tembine, and A. V. Vasilakos, "Game Dynamics and Cost of Learning in Heterogeneous 4G Networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 1, pp. 198–213, Jan. 2012.

[65] C. Long, Q. Zhang, B. Li, H. Yang, and X. Guan, "Non-Cooperative Power Control for Wireless Ad Hoc Networks with Repeated Games," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 6, pp. 1101–1112, Aug. 2007.

[66] M. Maskery, V. Krishnamurthy, and Q. Zhao, "Decentralized Dynamic Spectrum Access for Cognitive Radios: Cooperative Design of a Non-Cooperative Game," *IEEE Trans. Commun.*, vol. 57, no. 2, pp. 459–469, Feb. 2009.

[67] Q. Wu, Y. Xu, J. Wang, L. Shen, J. Zheng, and A. Anpalagan, "Distributed Channel Selection in Time-Varying Radio Environment: Interference Mitigation Game With Uncoupled Stochastic Learning," *IEEE Trans. Veh. Technol.*, vol. 62, no. 9, pp. 4524–4538, Nov. 2013.

[68] Z. Han, R. Zheng, and H. V. Poor, "Repeated Auctions with Bayesian Nonparametric Learning for Spectrum Access in Cognitive Radio Networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 3, pp. 890–900, Mar. 2011.

[69] M. Bennis, S. M. Perlaza, P. Blasco, Z. Han, and H. V. Poor, "Self-Organization in Small Cell Networks: A Reinforcement Learning Approach," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3202–3212, Jul. 2013.

[70] S. K. Jayaweera, M. Bkassiny, and K. A. Avery, "Asymmetric Cooperative Communications Based Spectrum Leasing via Auctions in Cognitive Radio Networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2716–2724, Aug. 2011.

[71] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 429–445, Apr. 1996.

[72] B. Parhami, "Voting Algorithms," *IEEE Transactions on Reliability*, pp. 617–629, 1994.

[73] S. Huang, X. Liu, and Z. Ding, "Decentralized Cognitive Radio Control Based on Inference from Primary Link Control Information," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 2, pp. 394–406, Feb. 2011.

[74] Y. Noam and A. J. Goldsmith, "The One-Bit Null Space Learning Algorithm and Its Convergence," *IEEE Trans. Signal Process.*, vol. 61, no. 24, pp. 6135–6149, Dec. 2013.

[75] B. Gopalakrishnan and N. D. Sidiropoulos, "Cognitive Transmit Beamforming from Binary CSIT," *IEEE Trans. Wireless Commun.*, vol. 14, no. 2, pp. 895–906, Feb. 2014.

[76] L. Zhang, Y. C. Liang, and Y. Xin, "Joint Beamforming and Power Allocation for Multiple Access Channels in Cognitive Radio Networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 617–629, Jan. 2008.

[77] S. Parsaeefard and A. Sharafat, "Robust Distributed Power Control in Cognitive Radio Networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 4, pp. 609–620, Apr. 2013.

[78] E. Dall'Anese, S. Kim, G. Giannakis, and S. Pupolin, "Power Control for Cognitive Radio Networks Under Channel Uncertainty," *IEEE Trans. Wireless Commun.*, vol. 10, no. 10, pp. 3541–3551, Oct. 2011.

[79] Y. Yang and S. Aissa, "Increased Spectrum Access Opportunities for Secondary Users Exploiting Adaptive Modulation in Primary Links," *IEEE Trans. Veh. Technol.*, vol. 64, no. 7, pp. 2911–2924, July 2015.

[80] B. C. Banister and J. R. Zeidler, "A Simple Gradient Sign Algorithm for Transmit Antenna Weight Adaptation With Feedback," *IEEE Trans. Signal Process.*, vol. 51, no. 5, pp. 1156–1171, May 2003.

[81] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac, "Distributed Transmit Beamforming Using Feedback Control," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 411–426, Jan. 2010.

[82] J. Xu and R. Zhang, "Energy Beamforming With One-Bit Feedback," *IEEE Trans. Signal Process.*, vol. 62, no. 20, pp. 5370–5381, Oct. 2014.

[83] S. Maleki, S. Chatzinotas, B. Evans, K. Liolis, J. Grotz, A. Vanelli-Coralli, and N. Chuberre, "Cognitive Spectrum Utilization in Ka Band Multibeam Satellite Communications," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 24–29, Mar. 2015.

[84] E. Lagunas, S. K. Sharma, S. Maleki, S. Chatzinotas, and B. Ottersten, "Resource Allocation for Cognitive Satellite Communications with Incumbent Terrestrial Networks," *IEEE Cognitive Commun. Netw.*, Nov. 2015.

[85] S. Boyd and L. Vandenberghe, "Localization and Cutting-Plane Methods," EE364b Lecture notes, Stanford University, Apr. 2008.

[86] L. Lovász, "Hit-and-run mixes fast," *Mathematical Programming*, vol. 86, no. 3, pp. 443–461, 1999.

[87] D. Bertsimas and S. Vempala, "Solving Convex Programs by Random Walks," *Journal of the ACM*, vol. 51, pp. 540–556, 2004.

[88] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[89] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[90] "IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and physical layer (PHY) specifications: High Speed Physical Layer in the 5 GHz band," *IEEE Std 802.11a-1999*, pp. 1–102, 1999.

[91] "IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and physical layer (PHY) specifications: Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz band," *IEEE Std. 802.11g-2003*, pp. 1–67, 2003.

[92] A. G. Marques, L. M. Lopez-Ramos, G. B. Giannakis, and J. Ramos, "Resource Allocation for Interweave and Underlay CRs Under Probability-of-Interference Constraints," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 10, pp. 1922–1933, Nov. 2012.

[93] F. E. Lapiccirella, Z. Ding, and X. Liu, "Improved spectrum access control of cognitive radios based on primary ARQ signals," *IET Commun.*, vol. 6, no. 8, pp. 900–908, May 2012.

[94] F. E. Lapiccirella, S. Huang, X. Liu, and Z. Ding, "Feedback-based access and power control for distributed multiuser cognitive networks," *Information Theory and Application (ITA) Workshop*, pp. 85–89, Feb. 2009.

[95] W. Wang, A. Kwasinski, D. Niyato, and Z. Han, "A Survey on Applications of Model-Free Strategy Learning in Cognitive Wireless Networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1717–1757, 3rd Quart. 2016.

[96] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Prentice-Hall, 1995.

[97] R. Waeber, P. I. Frazier, and S. G. Henderson, "Bisection Search with Noisy Responses," *SIAM J. Control Optim.*, vol. 51, no. 3, pp. 2261–2279, May 2013.

[98] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[99] B. Grunbaum, "Partitions of mass-distributions and of convex bodies by hyperplanes," *Pacific Journal of Mathematics*, vol. 10, no. 4, pp. 1256–1261, Dec. 1960.

[100] A. Caplin and B. Nalebuff, "Aggregation and Social Choice: A Mean Voter Theorem," *Econometrica*, vol. 59, no. 1, pp. 1–23, Jan. 1991.

[101] S. Boyd and C. Barratt, *Linear Controller Design: Limits of Performance*. Prentice-Hall, 1991.

[102] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*. Boca Raton, FL: Chapman & Hall/CRC, 1996.

[103] C. J. P. Bélisle, H. E. Romeijn, and R. L. Smith, "Hit-and-run algorithms for generating multivariate distributions," *Math. Oper. Res.*, vol. 18, no. 2, pp. 255–266, May 1993.

[104] H. Martini and A. Schöbel, "Median hyperplanes in normed spaces - a survey," *Discrete Applied Mathematics (Elsevier)*, vol. 89, pp. 181–195, Dec. 1998.

[105] B. Yu and P. Mykland, "Looking at Markov Samplers through Cusum Path Plots: a single diagnostic idea," *Statist. Comput.*, vol. 8, no. 3, pp. 275–286, Aug. 1998.

[106] M. Cowles and B. Carlin, "Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review," *Journal of the American Statistical Association*, vol. 91, no. 434, pp. 883–904, Jun. 1996.

[107] A. Marques, E. Dall'Anese, and G. Giannakis, "Cross-Layer Optimization and Receiver Localization for Cognitive Networks Using Interference Tweets," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 3, pp. 641–653, Mar. 2014.

[108] T. Minka, "A family of algorithms for approximate bayesian inference," Ph.D. dissertation, MIT, Cambridge, MA, 2001.

[109] K. Chaloner and I. Verdinelli, "Bayesian Experimental Design: A Review," *Stat. Sci.*, vol. 10, no. 3, pp. 273–304, Aug. 1995.

[110] I. Ben-Gal and M. Caramanis, "Sequential DOE via dynamic programming," *IIE Trans.*, vol. 34, no. 12, pp. 1087–1100, May 2002.

[111] D. Zois, M. Levorato, and U. Mitra, "Active Classification for POMDPs: A Kalman-Like State Estimator," *IEEE Trans. Signal Process.*, vol. 62, no. 23, pp. 6209–6214, Dec. 2014.

[112] M. Ono, M. Pavone, Y. Kuwata, and J. Balaram, "Chance-Constrained Dynamic Programming with Application to Risk-Aware Robotic Space Exploration," *Springer US Autonomous Robots*, vol. 39, no. 4, pp. 555–571, Dec. 2015.

[113] D. Zois and U. Mitra, "Active State Tracking With Sensing Costs: Analysis of Two-States and Methods for n-States," *IEEE Trans. Signal Process.*, vol. 65, no. 11, pp. 2828–2843, Jun. 2017.

[114] J. Williams, J. Fisher, and A. Willsky, "Approximate Dynamic Programming for Communication-Constrained Sensor Network Management," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4300–4311, Aug. 2007.

[115] R. Kan and C. Robotti, "On Moments of Folded and Truncated Multivariate Normal Distributions," *Journal of Computational and Graphical Statistics*, Apr. 2017.

[116] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 1995.