NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

PERFORMANCE ANALYSIS OF M-QAM WITH VITERBI SOFT-DECISION DECODING

by

Rogerio Correa Manso

March 2003

Thesis Advisor:	Tri T. Ha
Second Reader:	Jan E. Tighe

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 2003	Master's Thesis

4. TITLE AND SUBTITLE: Performance Analysis of M-QAM With Viterbi Soft-Decision Decoding	5. FUNDING NUMBERS
6. AUTHOR(S) Rogerio Correa Manso	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release; distribution is unlimited.	

13. ABSTRACT *(maximum 200 words)*

This thesis derives design tools for determining and improving performances of communication links that use M-QAM coherent demodulators associated with Viterbi soft-decision decoding (SDD) in Additive White Gaussian Noise (AWGN) and Nakagami-m channels. Performance analyses for 16-QAM, 64-QAM, QPSK and BPSK associated with up to three convolutional codes, including the one used by the IEEE 802.11a standard and the dual-k code, are presented as practical applications. The main tools relate to the analytical derivation of upper bounds of the probability of bit error (P_b) for any M-ary coherent demodulator followed by SDD, a methodology for improving an upper bound of P_b tightening it to realistic data, and the obtaining of the specific $b(d)$ spectrum for any convolutional code intended to operate with a certain M-symbol modulation. All derivations involve statistical considerations over the AWGN and Nakagami-m channels, as well as in-depth analyses of modulator constellations. The tools and models developed can provide great optimization to bandwidth-limited system designs that require high data rates, especially the wireless ones. Consequently, they have great application to many fields of digital communications, such as cellular telephony, wireless networking, satellite links, ship-to-shore and ship-to-ship communications.

14. SUBJECT TERMS	15. NUMBER OF PAGES
M-QAM, Convolutional Code, Viterbi Soft-Decision Decoding, Dual-k, IEEE 802.11a, AWGN, Nakagami.	127
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

PERFORMANCE ANALYSIS OF M-QAM WITH VITERBI SOFT-DECISION DECODING

Rogerio C. Manso
Lieutenant Commander, Brazilian Navy
B.S., Federal University of Rio de Janeiro, 1988

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2003

Author: Rogerio C. Manso

Approved by: Tri T. Ha
Thesis Advisor

Jan E. Tighe
Second Reader

John Powers
Chairman, Electrical and Computer Engineering Department

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis derives design tools for determining and improving performances of communication links that use M-QAM coherent demodulators associated with Viterbi soft-decision decoding (SDD) in Additive White Gaussian Noise (AWGN) and Nakagami-m channels. Performance analyses for 16-QAM, 64-QAM, QPSK and BPSK associated with up to three convolutional codes, including the one used by the IEEE 802.11a standard and the dual-k code, are presented as practical applications. The main tools relate to the analytical derivation of upper bounds of the probability of bit error (P_b) for any M-ary coherent demodulator followed by SDD, a methodology for improving an upper bound of P_b tightening it to realistic data, and the obtaining of the specific $b(d)$ spectrum for any convolutional code intended to operate with a certain M-symbol modulation. All derivations involve statistical considerations over the AWGN and Nakagami-m channels, as well as in-depth analyses of modulator constellations. The tools and models developed can provide great optimization to bandwidth-limited system designs that require high data rates, especially the wireless ones. Consequently, they have great application to many fields of digital communications, such as cellular telephony, wireless networking, satellite links, ship-to-shore and ship-to-ship communications.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGEMENT

A child without adequate support will never become a great adult. Two fundamental ingredients are necessary for any enterprise success and out of control of those who are directly endeavoring: support and confidence from third parties. I am extremely grateful to those who supported and trusted me during the accomplishment of this thesis and would like to acknowledge them here.

To my beloved wife, Teresa Cristina, for always believing me, for her kind support and for overcoming my absences during the hard times of research.

To my little daughter, Taina, for the daily inspiration of her smile and for respecting my study moments.

To my parents, Dr. Gilberto Manso and Dr. Elisete Manso, for the life example, childhood and youth education they provided me, fundamental elements for any success that I can reach in my lifetime.

To my advisor, Professor Tri Ha, for his knowledge, straightforward guidance, incentive, confidence and distinction that he treated me during the accomplishment of this thesis. Dr. Ha proposed me gradual and coherent challenges as well as shared valuable information that culminated with the final body of this work.

Finally, I express my respect and acknowledgement to the Brazilian Navy for honoring me with the rare opportunity and the sponsorship of these two years of studies at the Naval Postgraduate School.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The increasing demand for high data rates in communication systems is forcing manufacturers and engineers to design links that will work close to the limit of the channel capacity. In other words, links that should operate at the highest data rate possible within the expected level of reliability. Within the military context, one possible application for these links would be the usage of the high-frequency (HF) spectrum to obtain over horizon communications that could reach more than 10 Mbps without satellite intermediation. Ships and troops could keep an intranet link operating in HF. On the other hand, commercial applications like cellular telephony and wireless networking have enormous interest in broadening their data rates consuming the minimum bandwidth possible.

In this scenario, the *M*-quadrature amplitude modulation (M-QAM), a non-binary technique, performs a great role. It demonstrates high efficiency in bandwidth-limited systems, but demands high power in order to obtain the required reliability. Due to this later characteristic, M-QAM should be always associated with an efficient forward-error correcting (FEC) code to operate in an acceptable signal-to-noise ratio (SNR). At this point, one class of FEC that demonstrates excellent performance is the convolutional codes, especially when the receiver employs the most efficient decoding algorithm designed specifically for them, the Viterbi soft-decision decoding (SDD). Certainly, the association of M-QAM with SDD will be very important to meet the highest possible data rate requirements.

The presence of M-QAM is more noticeable in wire communications than in wireless due to the non-existence of the propagation obstacles, and the possibility of placing regenerators in the midway. Those propagation obstacles, such as multipath, shadowing and Doppler effect, end up producing fading in the receiver process, which decays considerably the SNR for wireless applications. However, the “big boom” of the cellular telephony and wireless networking are changing this scenario by forcing high data rates and, consequently, pulling M-QAM into the wireless world. As an example,

most standards for the third-generation cellular system (3G) are being experimentally deployed based on M-QAM nowadays.

Probably, due to being relative new into the wireless environment, the literature does not provide analytical models to determine performances of M-QAM with SDD up to this point. All analytical analyses involving SDD are done for binary modulation techniques, such as binary phase shift keying (BPSK).

This thesis provides a deeper study in the major concepts involving M-QAM with SDD and derives tools for determining and improving performances of such systems. It considers the classical process of upper-bounding the probability of bit error for BPSK with SDD systems, and extends it, by means of a vector approach, to a generic model of non-binary coherent demodulators followed by SDD. Consequently, the model for M-QAM becomes a trivial special case of the previous generic model. Regarding the channel corruption, the models are obtained for the additive white Gaussian noise (AWGN) scenario in a first consideration. As a second phase, in order to account for fading, the generic and M-QAM models are updated to consider the effects of frequency-nonselective small fading Nakagami-m channels.

The models' derivation resides on considering the worst case sequence of symbols being transmitted. However, the probability that all wrong decoded symbols in a sequence form a worst case is very small for large M's. The thesis takes advantage of this fact and develops a technique to realistically improve performances for large M's. As a result of such improvements, many upper bounds can be tightened to realistic views of the coding potential.

Another interesting aspect arises when trying to utilize a convolutional code that is considered good for BPSK, into a M-QAM channel. Does it obtain good performance as well? Can a code perform well for every M-QAM, regardless of M? In order to answer those questions, a more intense attention is devoted to the $b(d)$ spectrum and the code free distance, which are the most important convolutional code parameters responsible for determining the code performance. As a result, a numerical method for adapting $b(d)$ spectrum to be used in a certain M-QAM channel is developed. Actually,

the method generates different $b(d)$ spectra depending on M and based on the code's transfer function matrix.

As a practical application for all tools and models previously discussed, the thesis analyzes and plots the performances of four modulations: 16-QAM, 64-QAM, BPSK and QPSK operating in both AWGN and Nakagami fading channels. Each of the modulations is associated with the (2, 1) convolutional code used by IEEE 802.11a standard and the $(2k, k)$ dual-k code. For the 16-QAM case, an extra code is also used due to demonstrating excellent performance with relative low data rate loss.

Finally, the thesis summarizes some interesting results such as the range of good performances for the IEEE 802.11a code as well as its different $b(d)$ spectra, validity of the improvement technique, formulas for upper bounds of M-QAM with SDD operating in either AWGN or Nakagami fading channels, and relative differences in performances between fading and non-fading environments.

The tools here developed can provide great optimization on designs of any M-QAM link and increase the efficiency of bandwidth-limited systems. Consequently, they could be considered extremely worthy for obtaining high data rates in many fields of digital communications, especially the wireless ones, such as cellular telephony, wireless networking, satellite links, ship-to-shore and ship-to-ship communications, etc.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

As the internet era imposes itself as being an absolute necessity in the modern world, it increases the demand on high data rate exchange for all existent digital communication systems in order to allow all colors, pictures, audio and video information to be transmitted.

In the wireless world, many systems and standards, originally designed to carry just voice channels, which demand a meager 3.3 kHz of bandwidth, are being modified to meet the high data rate requirements. New standards are just being created and experimentally deployed, such as the third-generation cellular system (3G). This trend also reaches the military environment, which seeks the many options to provide high data links to mobile units such as ships, aircrafts and troops. Intranet services could be provided to an operating unit in a long distance communication by exploiting traditional channels formerly used to transmit just voice, such the high-frequency (HF) links.

In the world of wire communication, the trend for higher data rates is much easier to resolve due to the non-existence of fading, i.e., propagation obstacles that tremendously affect the wireless performance, such as multipath, shadowing and Doppler effect. As an example, a simple wire telephone line can be interfaced by a very affordable dial-up V.90 modem and be able to carry around 50 kbps with no changes to the carrier company systems [1]. If more bandwidth is needed, a very-high-digital subscriber line (VDSL) can provide a pair of wires with up to 51 Mbps downstream. Offices around the world are creating a shared information work environment by simply interconnecting their computers with local area network (LAN) cables and obtaining 100 Mbps of data rate.

In this meantime, wireless communications was greatly expanded during the last decade due to great improvements in quality and affordability of the second-generation cellular systems (2G). The code division multiple-access (CDMA) technique was developed for 2G and is now regulated by the IS-95 standard, opening a new paradigm on

sharing the electromagnetic medium among various users. Nothing would be more natural than to expect wireless to provide most of the useful services yet available by wire. For that purpose, the communication engineer community is devoting considerable efforts to overcome the propagation matters, especially within urban areas. The establishment of the IEEE 802.11a standard is a solid result of those efforts. It regulates wireless networking and provides data rates up to 54 Mbps when employing 64-QAM modulation and the orthogonal frequency-division multiplexing (OFDM) technique [2].

The new wireless standards previously cited are based on certain principles for maximizing the data rate in restricted bandwidth. One of them concerns the importance of utilizing a non-binary modulation technique, called M-QAM, to increase the channel capacity. Another concept highlights employing a forward error correction (FEC) coding scheme to compensate for the increase of the probability of bit error due to channel fading and M-QAM power inefficiency.

The use of convolutional codes as FEC is known to provide excellent results. Additionally, their optimizations occur in the presence of Viterbi soft-decision decoding (SDD) [3]. However, the literature does not provide analytical models to determine performances of non-binary M-QAM modulations associated with SDD up to this point. All analytical analyses involving SDD are done for binary modulation techniques, such as binary phase shift keying (BPSK), and at most extended the results to quadrature phase shift keying (QPSK), considered as two orthogonal BPSK modulations occurring at the same time. Models to analyze performances of M-QAM with SDD would be very useful in the process of providing high data rates for wireless communications, especially for systems subjected to severe fading conditions.

B. OBJECTIVE

The objective of this thesis is to provide design tools and analytical models to analyze performances of M-QAM modulations using SDD either in the absence or presence of fading conditions.

The main developed tools relate to:

- the analytical derivation of upper bounds for the probability of bit error (Pb) when modeling any M-ary coherent demodulator followed by SDD in pure AWGN and in Nakagami fading channels,
- the application of the previous models to obtain formulas for the special case of M-QAM modulations,
- the obtainment of the $b(d)$ spectrum for any convolutional code,
- the necessary adaptation on the $b(d)$ spectrum of a certain convolutional code originally designed for binary channels, to be used in non-binary channels,
- a technique that improves the performance analyses of M-QAM channels by realistically tightening the upper bounds of Pb,
- the application of the previous tools for determining performance analyses of QPSK, 16-QAM and 64-QAM, as well as BPSK, when using dual-k and IEEE 802.11a convolutional codes in AWGN and Nakagami fading channels, and
- an interesting option of convolutional code to associate with 16-QAM.

C. THESIS OUTLINE

Chapter II can be divided into two parts depending on the levels of information provided. The first part, compounded by Sections A, B and C, reviews the fundamentals on M-QAM modulation and convolutional codes. Most of the symbology utilized throughout the thesis is established there. The general formula for determining the probability of bit error associated with the use of FEC is explained together with the splitting of this process in two phases. The first looks for the $b(d)$ spectrum that is a characteristic of the convolutional code, while the second searches for an upper bound for the probability of error in the SDD algorithm, $P_2(d)$, applied to a certain modulation constellation. The second part of Chapter II includes Section D and is devoted to explaining methods for determining the $b(d)$ spectrum of any convolutional code. Accordingly, an analytical and a numerical methodology are shown together with a procedure for adapting $b(d)$ spectra of convolutional codes originally designed to operate in binary channels, to be used in non-binary channels. At the end of the chapter

the $b(d)$ spectra of the code employed by the IEEE 802.11a standard are obtained for BPSK, QPSK, 16-QAM and 64-QAM as an exemplification of the exposed numerical methodology. The correspondent spectra are summarized in a table for quick access.

In Chapter III, the upper bound of $P_2(d)$ for any M-ary coherent demodulator followed by a SDD is analytically derived for a pure AWGN scenario. This upper bound is further associated with the $b(d)$ spectra previously obtained in Chapter II and applied in the special case of M-QAM rectangular constellations. The final result is a model for determining probabilities of bit error for any M-QAM modulation with SDD subjected only to AWGN. Furthermore, a technique that improves the upper bound of $P_2(d)$ trying to tighten it to realistic data is developed. As a practical application, the performance analyses of 16-QAM, 64-QAM, BPSK and QPSK, associated with up to three different convolutional codes are shown, in which the dual-k and the IEEE 802.11a code are included. The results of an excellent convolutional code to be used only with 16-QAM comes out of these analyses, which also take into account the technique of upper bound improvements discussed previously.

Chapter IV discusses the basic concepts of Nakagami fading models and derives formulas for uncoded M-QAM in this particular environment. Following the same principles applied to the uncoded channels, the model for determining probabilities of bit error of non-binary modulations with SDD in AWGN obtained in Chapter III is expanded for Nakagami fading channels. The performance analyses of the four modulations in Chapter III are now repeated for the new model utilizing four fading figures m: 3, 2, 1 and 0.5. The codes are the same as before and the technique of upper bound improvements is also applied.

In Chapter V, a summary of the obtained design tools and models is presented together with the main results from their application to the specific cases of 16-QAM, 64-QAM, BPSK and QPSK operating with dual-k and the IEEE 802.11a convolutional codes. Also, it suggests the application of the results of this thesis in future work.

The appendix finally provides an instance of how to apply the analytical method to derive a general formula for the $b(d)$ spectra of dual-k convolutional codes.

II. M-QAM WITH CONVOLUTIONAL CODES

Convolutional codes are not easy to study unless all the basic concepts involving them, as well as the systems that surround them, are very well understood. This chapter recalls the fundamentals on M-QAM modulation, convolutional codes, and probability of bit errors. All these concepts will be extensively used throughout the thesis. The chapter also discusses in greater detail an important parameter of convolutional codes for determining performances: the $b(d)$ spectrum. It demonstrates how to overcome some practical problems involving $b(d)$ spectra, such as how to determine and adapt them to different modulations.

A. M-QAM

M-ary Quadrature Amplitude Modulation (M-QAM) is a non-binary memoryless modulation technique in which one of M different symbols is transmitted per time using two orthogonal carriers (in quadrature). Each symbol is composed of q bits. The following formulas apply:

$$s(t) = A_1 \Psi_1(t) + A_2 \Psi_2(t), \tag{2.1}$$

$$q = \log_2 M, \tag{2.2}$$

where

$$\Psi_1(t) = \begin{cases} \sqrt{2/T_S}\cos(2\pi f_c t), & \text{for } 0 \le t \le T_S \\ 0, & \text{otherwise} \end{cases}$$

$$\Psi_2(t) = \begin{cases} -\sqrt{2/T_S}\sin(2\pi f_c t), & \text{for } 0 \le t \le T_S \\ 0, & \text{otherwise,} \end{cases}$$

in which T_s represents one symbol's transmission time interval, f_c is the carrier frequency, and A_1, as well as A_2, are the orthogonal carrier coefficients.

The main advantage to using M-QAM is the gain in bit rate it provides. Notice that in the interval T_s, q bits are transmitted simultaneously and are embedded in only one

analog carrier. The final effect is an increasing of q times in the bit rate for M-QAM when compared with a binary modulation technique such as binary phase shift keying (BPSK). In other words, it increases the digital bit rate that can be transmitted within a limited bandwidth channel. One popular application of M-QAM is computer modems used for dial-up connections when the user wants to maximize the communication bit rate over a restricted 3.3 kHz voice channel.

This advantage does not come without a price, which is the increasing of the probability of bit error (P_b) at a certain signal-to-noise-ratio (SNR) in the channel when using M-QAM instead of a binary modulation. The larger the M, the worse the P_b will be for the same SNR. In the end, the simplest way to keep P_b at an adequate level required by the communication link is to add more power to the transmitted signal. The final result will be the enhancement of the SNR. Another factor to consider is the relative complexity of the transceivers that also grow proportionally with M.

The M-QAM is completely defined by its constellation, which graphically represents the M possible symbols as dots on a Cartesian plot. The coordinates (x, y) of each constellation dot are the coefficients A_1 and A_2 of the correspondent symbol and are addressed by Equation (2.1). A popular constellation is one that is rectangular in shape, shown in Figure 2.1 for M equal to 16. This type is used by 2,400-bits/s V.22 bis computer modems [1].

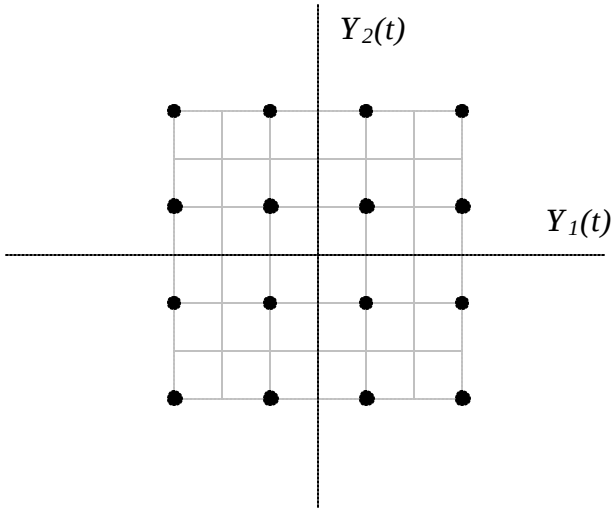


Figure 2.1. The 16-QAM Rectangular Constellation [After Ref. 1]

Even though all the concepts developed in this thesis are generic and could be applied to any M-QAM, a numerical focus will be given to four particular *M*'s: 2, 4, 16 and 64. These values have vast applications in numerous communication systems and define the BPSK, QPSK, 16-QAM and 64-QAM modulations, respectively.

B. CONVOLUTIONAL CODES

Another way to overcome the increasing of P_b for a certain SNR when using M-QAM is to codify the information bits to be transmitted (channel coding) in such a way that only a small set of all possible received bit sequences are considered valid. The receipt of one invalid sequence indicates that an error has occurred in the channel. The decoder will then determine the closest valid sequence matching the received sequence and declare this valid sequence as the original sent instead of the one received. This process of encoding and decoding the information bits is called *forward error correction* (FEC) technique [1].

A special type of FEC commonly associated with M-QAM is one that combines previous input bits, stored in a memory, with the actual bits at the encoder input to generate the coded output bit sequence. Since new input bits constantly replace previously stored bits in a linear shift-register manner, the final result is a convolution operation between the input bit stream and a certain built-in function. This type of FEC is known as *convolutional code*.

A convolutional code can be precisely described by some parameters. The main parameters are briefly depicted below but only in order to review basic concepts on the subject to be used throughout this thesis. Extensive detailed explanations are found in the reference literature.

1. Code Rate (R_c) and Constraint Length (K)

A (n, k) convolutional code causes the input bits to the encoder to be shifted into and along the shift register k bits at a time. The number of output bits for each k-bit input sequence is n bits [3]. Consequently, the code rate is defined by the formula

$$R_C = k / n . \tag{2.3}$$

Notice that n is always greater than k causing R_c to be less than 1. Typical values for R_c are between 1/4 and 7/8. Since R_c is less than 1, it can be seen that the use of convolutional code reduces the final maximum bit rate capable of being transmitted in a bandpass communication channel. Also, notice that a small value for the code rate R_c indicates a high degree of redundancy, which should provide more effective error control at the expense of increasing the bandwidth of the encoded signal [1].

The convolutional encoder memory consists of a shift register with K (k-bit) stages and n linear algebraic function generators (modulo-2 adders). The parameter K is called the *constraint length* of the convolution code [3]. Therefore, each one of the n output bits is affected by K sets of k input bits.

2. Generator Polynomials and Transfer-Function Matrix

A convenient way to represent the k-bit inputs and n-bit outputs along time is through polynomials in D, where the indeterminate D is interpreted as a delay operator. Each of the n modulo-2 adders is connected to the various shift register stages and is responsible for the generation of one of the n output bits. Consequently, it is possible to imagine a certain set of *generator polynomials* $G_i^{(j)}(D)$, organized in a matrix template, when multiplied by a polynomial representing the k-bit input, *X(D)*, will lead to another polynomial representing the n-bit output, *Y(D)*. The mathematical representation follows.

$$Y(D) = X(D)G(D) , \tag{2.4}$$

where

$$Y(D) = \left[Y^{(0)}(D) \quad Y^{(1)}(D) \quad \ldots \quad Y^{(n-1)}(D) \right],$$

$$X(D) = \left[X^{(0)}(D) \quad X^{(1)}(D) \quad \ldots \quad X^{(k-1)}(D) \right],$$

and

$$G(D) = \begin{bmatrix} G_0^{(0)}(D) & G_0^{(1)}(D) & \ldots & G_0^{(n-1)}(D) \\ G_1^{(0)}(D) & G_1^{(1)}(D) & \ldots & G_1^{(n-1)}(D) \\ \ldots & \ldots & \ldots & \ldots \\ G_{k-1}^{(0)}(D) & G_{k-1}^{(1)}(D) & \ldots & G_{k-1}^{(n-1)}(D) \end{bmatrix}.$$

The matrix *G(D)* is called the *transfer-function matrix* for the encoder [4]. The various generator polynomials $G_i^{(j)}(D)$ that compound *G(D)* are usually represented by their coefficients expressed in octal base. For example, $(14)_8$ means $(001\ 100)_2$ and would represent $G_i^{(j)}(D) = D^3 + D^2$.

3. Trellis Diagram, Coded Symbols and Hamming Distance

A very compact and convenient way to represent graphically all possible convolutional code outputs as a function of their inputs is by means of a *trellis diagram*. Each branch in the trellis expresses not only the *n*-bit output associated with the *k*-bit input, but also graphically indicates the correspondent encoder's internal state change. The set of successive branches defines a sequence of outputs also known as a path of code words.

Figure 2.2 shows the *n*-bit output of the convolutional encoder feeding an M-QAM modulator that accepts *q* bits per time. Considering $n > q$, those *n*-bits in *N* sets of *q*-bits can be grouped, where *q* is given by Equation (2.2) and *N* is essentially an integer number. On the other hand, if $n \leq q$, *N* sets of *n* bits will be necessary to make a group of *q* bits. In both cases, the *q* bits are the ones that travel simultaneously through the channel

in the form of a symbol, which is called a *coded symbol*. Therefore, each branch in the trellis can either represent N coded symbols ($n > q$) or only 1 coded symbol ($n \leq q$), depending on the situation. Furthermore, any path in the trellis, since it is composed of various branches, embraces many coded symbols.

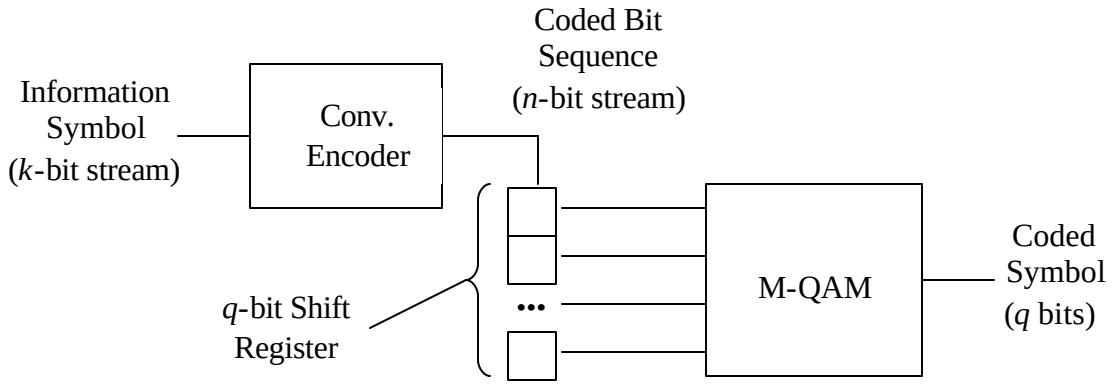


Figure 2.2. Block Diagram of a M-QAM Modulator Using a Convolutional Encoder

The *Hamming distance* or simply *distance* between two paths y' and y'' in the trellis, represented by $d(y',y'')$, is the number of coded symbols in which the two paths differ [4].

4. All-Zero Path, Minimum Free Distance (d_{free}) and Transfer Function

A convolutional encoder usually starts operating with its internal shift register completely cleared, i.e., in the all-zero state. In this state, its response to an all-zero input will always be an all-zero output. In other words, the encoder will remain in the all-zero state forever until a non-zero input occurs. The *all-zero path* is the horizontal line connecting all zero-state-dots (S_0) in the trellis diagram. It expresses the correct decision that a decoder should make when the transmission consists of a zero-information sequence. This is very useful in determining convolutional code performances. The method consists of constantly sending a zero-information sequence. In case a channel

error occurs, the probabilities and possibilities on the behavior of the decoder algorithm can be estimated by studying the paths that could leave and merge at the all-zero path.

The *minimum free distance*, denoted d_{free}, is the minimum Hamming distance between all pairs of paths that start and stop at S_0 [4]. The larger d_{free}, the greater the possibility of correcting eventual channel errors. Usually the increase of d_{free} is accompanied by an increase in the constraint length for the same code rate, which will result in enhancing the encoder complexity. One of the goals of achieving a good convolutional code is to obtain the largest d_{free} for a same constraint length and same code rate.

Another useful resource for studying convolutional codes that will be referenced later in this thesis is the *transfer function*. Basically, it shows a mathematical expression that reports to the contribution of *all paths that necessarily leave and arrive at the state* S_0. The expression below is an example [3] of a transfer function for a convolutional code with $R_C = 1/3$ and $K = 3$,

$$T(D, N) = \frac{N D^6}{1 - 2N D^2}.$$

The meaning of the factors D and N can be better understood by expanding $T(D,N)$ in an infinite series as follows:

$$T(D, N) = \frac{N D^6}{1 - 2N D^2} = ND^6 + 2N^2 D^8 + 4N^3 D^{10} + \ldots = \sum_{d=6}^{\infty} a_d N^{(d-4)/2} D^d, \quad (2.5)$$

where

$$a_d = \begin{cases} 2^{(d-6)/2} & (\text{even } d) \\ 0 & (\text{odd } d). \end{cases}$$

Looking at the series in Equation (2.5), the first term ND^6 informs that there is only one possible decoding path presenting one (exponent of N) information symbol different than the zero-information-symbol that corresponds to six (exponent of D) coded symbols different than the zero-coded-symbol. The second term in Equation (2.5), $2N^2 D^8$, means

that there are two possible decoding paths presenting two (exponent of N) information symbols different than the zero-information-symbol that correspond to eight (exponent of D) coded symbols different than the zero-coded-symbol. The interpretation of the other terms follows this same rule.

The transfer function is obtained by building and solving a system of state equations based on the state diagram of the convolutional code circuit. Each possible state transition corresponds to one branch. For this reason, the definitions of D and N are related to branches.

C. PROBABILITY OF BIT ERROR

The most important figure of merit in a communication link is the *probability of bit error for the information bits* (P_b) as a function of the SNR. Every digital communication link has to comply with a certain maximum P_b associated with a realistic SNR that could be present in the channel. The level of this threshold depends on how tolerant the receiving digital system is to errors .

The formulas for determining P_b versus SNR vary accordingly to the convolutional code, interference noise, demodulator and decoder algorithm employed. For convolutional codes, the optimum decoding involves a search through the trellis for the most probable sequence, what is achieved by means of the Viterbi algorithm [3].

Notice in Figure 2.2 that even though the convolutional encoder generates n bits at the output, the considered coded symbols are q bits long. Those q bits are those that really travel through the channel and are subjected to errors.

A general formula for determining an upper bound on P_b for M-QAM with a convolutional code can be derived as follows. Consider first:

$$P_b = \frac{1}{k} P_M , \tag{2.6}$$

where P_M is the average probability of an information symbol (k-bit long), which feeds the convolutional encoder during the transmission, is retrieved in error at the end of the

reception process, and k is the number of information bits that compound an information symbol, i.e., the number of bits shifted into the encoder per time.

The application of the concepts of union bound, first-event error and transfer function make it possible to study error probabilities for the Viterbi decoding of a convolutional code [3, 4]. One of the well known results of such studies is the general formula below that depicts an upper bound for P_M.

$$P_M \leq \sum_{d=d_{free}}^{\infty} b(d) P_2(d), \tag{2.7}$$

where $b(d)$ is the total number of information symbols (k-bit long) different from the zero-symbol associated with all paths with distance d from the all-zero path, and $P_2(d)$ is the probability of choosing a wrong path in the trellis in a pairwise comparison of the all-zero path (correct path) with another path else (wrong path) with distance d from the all-zero path.

Substituting Inequality (2.7) in Equation (2.6), the general formula for an upper bound of P_b is obtained,

$$P_b \leq \frac{1}{k} \sum_{d=d_{free}}^{\infty} b(d) P_2(d). \tag{2.8}$$

Notice by the definitions of Equation (2.7) that $b(d)$ depends only on the convolutional code employed. It relates the number of *information symbols* in error with the number of *coded symbols* detected in error. Since $b(d)$ is created by a convolution operation, its expression is mathematically feasible either by employing an analytical method or a numerical method as will be seen later in this chapter.

On the other hand, $P_2(d)$ is completely independent on the convolutional code, being a result solely of the type of noise existent in the channel, the demodulation and the decoder algorithm. Its calculation is complex and usually done by upper bounding considerations over a pattern of an all-zero sequence transmission as explained earlier.

The independence between $b(d)$ and $P_2(d)$ makes it possible to split the problem of determining P_b in two completely isolated phases that will be put together by Equation (2.8). The first phase consists of the analysis of the convoulutional code in order to determine $b(d)$, whereas the second issue concerns the channel model as well as the receiving process in order to estimate a reliable upper bound for $P_2(d)$.

D. DETERMINING THE PARAMETER $b(d)$

There are two efficient methods for determining $b(d)$. The first is analytically from the convolutional code transfer function, and the second is numerically from the generator polynomials. Both can become difficult for some codes, especially those that are good, requiring some simplification arguments or some knowledge on where to stop.

1. Obtaining $b(d)$ from the Transfer Function

This is an analytical process involving two steps [3]. The first is to expand the transfer function $T(D,N)$ of the convolutional code into an infinite series in powers of D and N. The second step consists of taking the partial derivative of this series with respect to N, and making N equal to 1 afterwards in order to obtain another series in powers of D only. The expression for $b(d)$ becomes the equation that defines the coefficients of this last series as a function of d.

The Appendix shows an application of this method for obtaining a useful formula for $b(d)$ of the dual-k convolutional code.

2. Obtaining $b(d)$ from the Generator Polynomials

The process of obtaining the transfer function for a convolutional code is not trivial. For large constraint lengths, the complexity of the process most of the times avoid reaching a final formula. Hopefully, by observing the abrupt tapering form of $P_2(d)$ as d increases, and that $b(d)$ will be employed in Equation (2.8), it can usually be seen that only the first five terms of $b(d)$ are significant for determining P_b. Under these circumstances, a numerical method can be applied to the convolutional code core in order to precisely compute just the first few $b(d)$'s.

The method uses the code generator polynomials for forcefully generating a large number of paths in the trellis beginning and ending at the zero state. These paths contain a relatively large number of branches, usually more than 3 times the constraint length, to cover a sufficient range of d's. The process of path generating must follow a trend in an ascending order of d. The computation of the number of coded symbols different than the zero-symbol contained in each of those paths gives d. The number of information symbols different than the zero-information-symbol embedded in the information sequence related to the considered path provides the contribution for $b(d)$ made by this specific path. If all those contributions for $b(d)$ are continuously accumulated (summed) in a multi-dimensional vector, where each coordinate represents one d, after computing a large number of paths, the vector coordinates will express the $b(d)$ spectrum. Since the path generation overall behavior follows an ascending order, it can be easily concluded that the accuracy of the first terms of $b(d)$ increases as the number of generated paths becomes larger. Furthermore, those terms will freeze as soon as all paths that could contribute to their correspondent $b(d)$'s have been generated. The process should stop when all terms of interest, usually the first three or five, are obtained.

The flowchart in Figure 2.3 shows the basic algorithm for computing the $b(d)$ as described above. It can be seen that the process first generates the information-bit sequence at block 2 and later the coded-bit sequence at block 5. Considering that the information sequence generation obeys an ascending order of numbers of information symbols in error, the correspondent coded sequences will present an overall behavior of ascending order of *d*. The *overall* word stands since among some successive paths, one path could have *d* less than its previous path due to the characteristics of convolutional codes.

Notice the presence of a filter in block 4 in Figure 2.3 to avoid the codification of invalid information sequences. Recall that the Viterbi decoding process applied to an all-zero coded sequence is always trying to compare the metric of paths that could leave and merge with the all-zero path against the metric of the all-zero path itself. Thus, a path with a large number of branches that leaves and bounces off the all-zero path instead of merging with it should be rejected. In fact, this path, at the moment it bounces off the all-zero path, can be split into two distinct paths with smaller *d*, which was probably already computed earlier in the process.

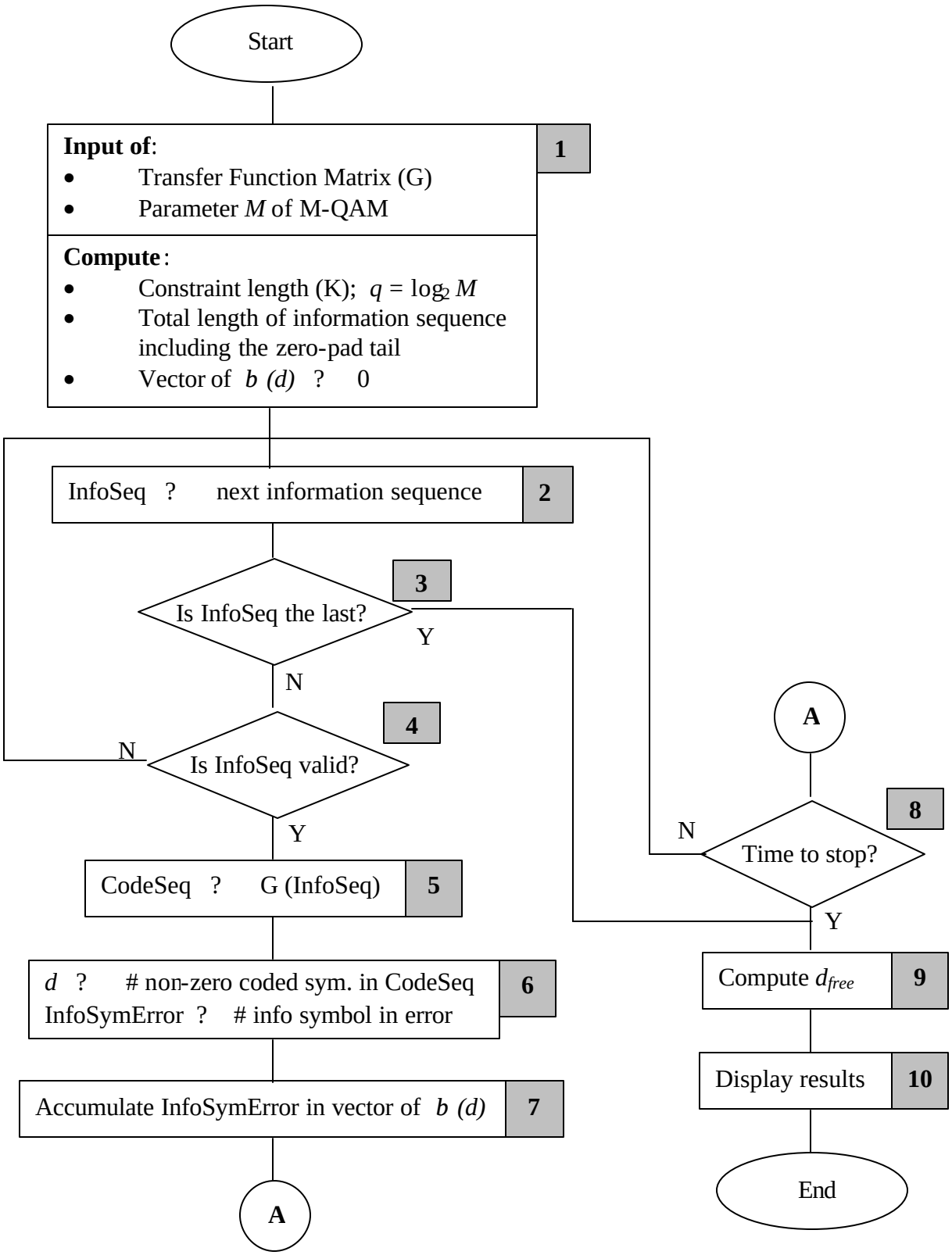


Figure 2.3. Algorithm for Obtaining $b(d)$ from the Generator Polynomials of a Convolutional Code

Figure 2.4 illustrates an example of an allowed path with six branches. Notice that when it merges with the all-zero path; it remains there to the end.

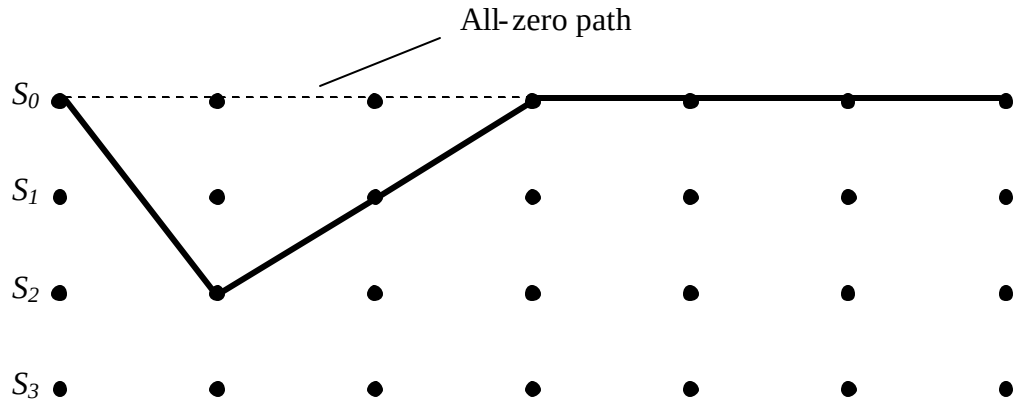


Figure 2.4. Example of a Valid Path in the Trellis That Must Be Considered When Computing $b(d)$

Conversely, Figure 2.5 shows a path that bounces off of the all-zero path and must be rejected.

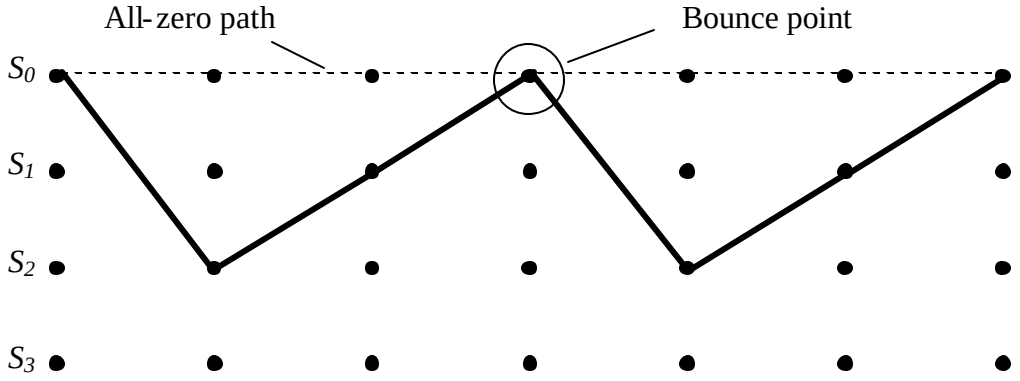


Figure 2.5. Example of an Invalid Path in the Trellis That Must Be Rejected When Computing $b(d)$

Of course, the same computation principle leading to $b(d)$ could be used to achieve other useful parameters such as $a(d)$, the total number of paths (code words) with distance d from the all-zero path. For this purpose, some extra computations should be added to block 7 in Figure 2.3.

As can be seen, this forceful process of determining $b(d)$ is based on a relatively large number of loops, dependent on the total length established for the information sequence. For instance, if someone works with a code that has $k = 1$ (binary input) and constraint length $K = 7$, considering that the number of constraint lengths for determining information sequence length is fixed in 3, the algorithm would have to generate and check $2^{3*7} = 2^{21} = 2{,}097{,}152$ information sequences. It is easy to realize that this number grows exponentially with K and can lead to an expensive computation cost. Therefore, a large effort should be allocated to stop the process at block 8 in Figure 2.3.

Some optimizations could be inserted in the process to avoid the generation of information sequences that a priori is known to yield to an invalid path. For example, if there were 12 branches instead of six in Figure 2.5, all combinations of paths beginning with their first six branches following the pattern indicated in the figure would be considered invalid. These combinations can be called a "family" and since the first member of this family, the path indicated in Figure 2.5, is considered discarded, the entire family can be a priori also discarded, and a considerable amount of computation resources can be saved.

As a final observation, the minimum free distance (d_{free}) of the code is also achievable at the end of the process. The $b(d)$ spectrum (computed vector) will present zeros as the first terms, indicating that there are no paths with $d = 1, 2, 3, \ldots, (d_{free} - 1)$ that could leave and merge with the all-zero path. The first non-zero term in the $b(d)$ spectrum represents d_{free} and can be easily obtained by searching in the final vector coordinates. This computation is performed in block 9 in Figure 2.3.

3. Adapting $b(d)$ for Various Coded Symbol Lengths

Sometimes the definition of $b(d)$ is a little confusing and some details pass unnoticed. For example, as discussed previously, $b(d)$ is a function of the convolutional code employed. However, the same convolutional code can generate different $b(d)$ when attached to different M-QAM modulators. Notice that the distance *d* is related to the number of coded symbols different than the zero-symbol. As seen in Section II.B.4, the length of the coded symbol varies with *M* according to Equation (2.2).

Consider a certain system of a convolutional encoder generating one fixed coded bit sequence as a response to a fixed information sequence at its input. If the option of connecting this system to one of the two M-QAM modulators presenting different *M*'s is available, the same coded bit sequence would be clustered in a different amount of *q* bits in order to generate the coded symbols for each modulator considered. Consequently, different *d*'s would result, which would impact $b(d)$ directly, even though the information bit sequence and the convolutional encoder remains the same for the two systems.

Very often a communication engineer knows that certain convolutional code works very well for a certain M-QAM channel and wants to improve the bandwidth utilization by employing a modulator with higher *M* and would like to estimate the performance impact that would result if the same convolutional code is used. However, the $b(d)$ spectrum tabled for that code is specific for the original M-QAM channel. The question is how can a new $b(d)$ spectrum be achieved for the new M-QAM considered?

Looking at Figure 2.3, block 6 is responsible for adapting information and coded sequences into information and coded symbols. It is this part of the algorithm that really takes into account the *q* bits of the M-QAM channel to reflect on the $b(d)$ estimation. Thus, the answer for the previously stated question is that if the generator polynomials

for the considered code is known, the engineer can simply use the algorithm depicted in Figure 2.3 provided with the generator polynomials coefficients and the new desired M.

The flexibility on counting information and coded symbols for the same original sequences makes this process a very useful tool to map different $b(d)$ spectra for the same convolutional code. Notice, however, that a convolutional code that works very well for a certain M-QAM channel may not perform when working with a different *M*.

4. The $b(d)$ Spectra for IEEE 802.11a Convolutional Code When Applied to BPSK, QPSK, 16-QAM and 64-QAM Modulators

A program in Matlab was developed for this thesis based on the algorithm presented in Figure 2.3. One of the various codes checked by this program is the industry-standard convolutional code used by the IEEE 802.11a widely applied to wireless networks. The code has the following characteristics [5]: transfer function matrix $G = [133\ 171]_8$, $R_C = 1/2$, and $K = 7$. Table 2.1 shows the $b(d)$'s spectra obtained for this code when used with four different M-QAM modulators. The parameter *i* denotes the increment that should be added on d_{free} to obtain *d*, i.e., $d = d_{free} + i$.

Modulation	d_{free}	ß(d)				
		i = 0	**i = 1**	**i = 2**	**i = 3**	**i = 4**
BPSK	10	36	0	211	0	1404
QPSK	6	1	10	38	92	346
16-QAM	4	8	44	323	2033	11575
64-QAM	3	12	140	1784	19873	207985

Table 2.1. The $b(d)$'s Spectra for the Industry-Standard Convolutional Code ($R_C = 1/2$) Used by the IEEE 802.11a When Applied to Different M-QAM Modulators.

Notice in this case that the code rate is 1/2. Thus, the convolutional encoder can only be applied to M-QAM modulators presenting q equal to 1 or q as a multiple of 2, i.e., {$q = 1$ (BPSK or 2-QAM), 2 (QPSK or 4-QAM), 4 (16-QAM), 6 (64-QAM), 8 (256-QAM), ... }, through the arguments presented in Section II.B.4. Analogously, if the code rate were 1/3, allowed q should be {$q = 1$ (BPSK or 2-QAM), 3 (8-QAM), 6 (64-QAM), 9 (512-QAM), ... }.

E. SUMMARY

Chapter II reviewed some fundamentals on M-QAM modulation, convolutional codes, and probability of bit errors. The general formula for determining the probability of bit error associated with the use of FEC was highlighted, and special attention was given to issues related to the obtainment of $b(d)$ spectrum. The most important part of Chapter II presented a numerical method for determining the $b(d)$ spectrum of any convolutional code. As an application, the $b(d)$ spectra of the IEEE 802.11a convolutional code were obtained for BPSK, QPSK, 16-QAM and 64-QAM. Additionally, a formula for determining $b(d)$ spectra of dual-k codes was analytically derived. In the next chapter, all the concepts and results obtained thus far will be utilized

to obtain models, formulas and performance analyses of M-QAM with Viterbi soft-decision decoding in the AWGN scenario.

THIS PAGE INTENTIONALLY LEFT BLANK

III. PERFORMANCE ANALYSIS IN AN AWGN CHANNEL

As explained in Section II.C, the most important figure of merit in a communication link is the *probability of bit error for the information bits* (P_b) as a function of the SNR. Therefore, the performance of every digital link can be estimated by plotting its P_b against the SNR. This chapter derives a model for determining an upper bound of P_b for any M-QAM modulation with Viterbi soft-decision decoding subjected only to AWGN. Furthermore, a technique that improves this upper bound will also be developed and applied to performance analyses involving 16-QAM, 64-QAM, BPSK and QPSK.

The computation of P_b is performed on the receiver part of the link, and its circuits are in an opposite order when compared with the transmitter as shown in Figure 3.1.

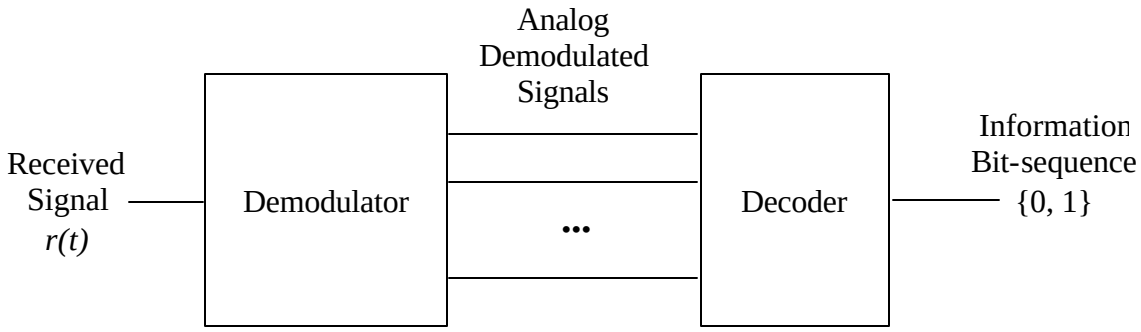


Figure 3.1. Block Diagram of the Main Receiver Circuits.

The parameter P_b will depend on both the demodulator and the decoder. The dependence on the demodulator is related with how sensitive this circuit is to the channel noise, i.e., how the noise associated with the signal at its input affects the demodulator outputs. On the other hand, the dependence on the decoder rests on the type of processing performed on its analog input (demodulator output) in order to decide which information bit sequence has been transmitted.

This thesis will concentrate on all the performance analyses considering the demodulation performed by a *M-QAM coherent demodulator* and the decoding done by a *Viterbi soft-decision decoder* (SDD). Notwithstanding this configuration requires more complexity from circuits and algorithms than when using a non-coherent demodulator and a hard-decision decoder, the results obtained are much better. The goal is to concentrate on the best performances since today's technology provides relatively cheap resources to implement complex circuits on a large scale.

A. VITERBI SOFT-DECISION DECODING

The *Viterbi soft-decision decoding* (SDD) is a process of determining the information bit- sequence related to one or more sets of demodulated coded symbol outputs by using a *maximum-likelihood sequence estimator* (MLSE). The principle is to exploit the interdependence (memory) that exists among successive coded symbols, established by the convolutional encoder, in order to choose the most probable coded symbol that was actually transmitted given the reception of a certain signal corrupted by noise.

Recalling that the trellis characterizes the memory in the transmitted signal, the MLSE algorithm searches among all possible paths through the trellis for a certain path that maximizes the probability that the received signals represent the coded symbols expressed by this path. For this purpose, the decoder associates a metric, called *correlation metric*, to each path. This metric is mathematically described for a certain path i, compounded by B branches, according to Ref. [3] as

$$CM^{(i)} = \sum_{j=1}^{B} m_j^{(i)} = \sum_{j=1}^{B} \log f_R\left(\mathbf{R}_j \mid \mathbf{C}_j^{(i)}\right), \tag{3.1}$$

where $f_R\left(\mathbf{R}_j \mid \mathbf{C}_j^{(i)}\right)$ represents the conditional probability density function (pdf) of the vector that expresses the received signals of the j-th branch ($\mathbf{R}_j$) given that the coded symbols of the j-th branch of the i-th path ($\mathbf{C}_j^{(i)}$) are considered to have been transmitted.

Notice by this definition that $m_j^{(i)}$ can be considered a branch metric causing the correlation metric to be the summation of all branch metrics in the path.

After computing the correlation metric for all possible paths reaching a considered stage in the trellis (B branches ahead of some starting point), the MLSE simply chooses the path with the highest metric and discards all others. Considering that the number of paths to compute the metrics grows exponentially with the number of branches, some optimizations are done in order to save computation resources. For example, taking advantage of the fact that the correlation metric is cumulative, the algorithm needs only to keep track of a fixed number, $2^{k(K-1)}$, of the highest metrics and their corresponding paths at a certain stage in order to compute the next stage metric. Those tracked paths are called *survivors* [3].

B. ADDITIVE WHITE GAUSSIAN NOISE CHANNEL

The *additive white Gaussian noise (AWGN)* is a term that refers to the characteristics of the *thermal noise*, which is present in all communication systems and is the predominant noise source for many of them. It corrupts the signal in an *additive* fashion causing:

$$r(t) = s(t) + n(t), \tag{3.2}$$

where *r(t)* is the total received signal, *s(t)* is the component due to the transmitted signal, and *n(t)* is the noise component.

Thermal noise is considered *white* because its two-sided power spectral density $G_n(f) = N_0/2$ is *flat for all frequencies of interest* [6]. It also presents a zero-mean *Gaussian* pdf depicted by

$$f_N(n) = \frac{1}{\sqrt{2\pi s^2}} e^{\left[\frac{-n^2}{2s^2}\right]}. \tag{3.3}$$

From this point forward the notation $N(\bar{X}, s_X^2)$ will be used to denote a normal or Gaussian random variable (r.v.) X with mean $\bar{X}$ and variance s_X^2.

The autocorrelation function of the AWGN is equal to $s^2 d(t)$. Also, $G_n(f) = N_0/2$ is the Fourier transform of this autocorrelation, which causes:

$$s^2 = \frac{N_0}{2}. \tag{3.4}$$

A channel is designated as an AWGN channel when its impairments are limited to the degradation caused by the thermal noise. Particularly, the AWGN present in a signal is split within a coherent M-QAM demodulator into its two *orthogonal* channels. Such noise can be represented by a bidimensional vector of orthogonal AWGN components expressed by

$$N_x = [n_{x1} \;\; n_{x2}], \tag{3.5}$$

where n_{x1} and n_{x2} are independent-identical-distributed (iid) Gaussian random variables with pdf's given by Equation (3.3).

C. DETERMINING THE PARAMETER $P_2(d)$ FOR M-QAM

Recalling Section II.C, the type of noise in a channel affects only $P_2(d)$. In this Section, a general formula for determining the upper bound of $P_2(d)$ for every M-ary modulation, whether orthogonal or non-orthogonal, will be analytically derived for an AWGN channel and customized for the special case of M-QAM modulators that use a rectangular constellation. Furthermore, a way to improve this upper bound, i.e., make it tighter, will be shown in an attempt to match the realistic communication characteristics for a particular link. The overall result will reflect directly on the predictions of the channel performance.

1. Derivation of an Upper Bound of $P_2(d)$ for any M-ary Coherent Demodulator Followed by a Soft-Decision Viterbi Decoder

In this Section the M-ary bidimensional case will be considered in a first analysis and then the results will be extended to a multi-dimensional modulation scheme.

Assume the block diagram in Figure 3.2.

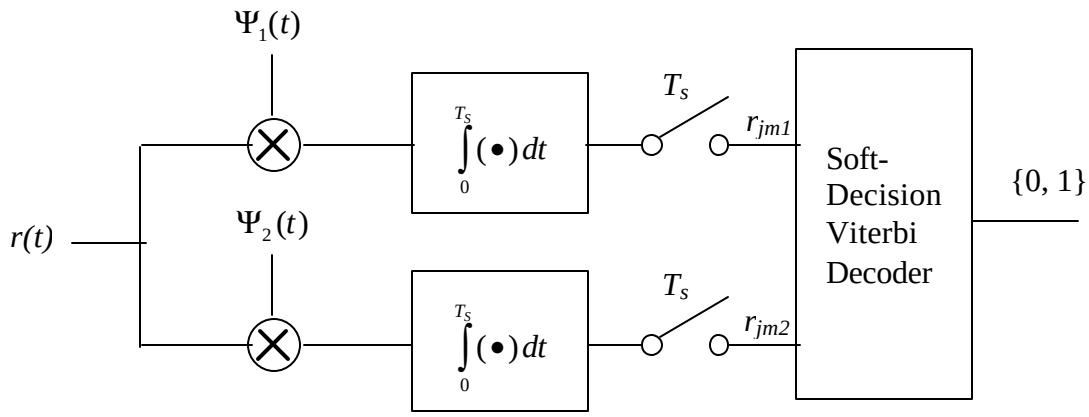


Figure 3.2. Block Diagram of an M-QAM Coherent Demodulator Followed by a Soft-Decision Viterbi Decoder.

The goal is to determine an upper bound for $P_2(d)$, which is the probability of choosing a wrong path in the trellis in a pairwise comparison of the all-zero path with another path that differs in d symbols from the all-zero path. For this purpose, the symbology that follows will be established.

The vector C_{jm} is the coded symbol related to the m^{th} symbol of the j^{th} branch in the trellis. Recall from Section II.2.4 that the n-bit output of the convolutional code can be split in L sets of q-bit to adjust itself to the channel symbol length. This vector is depicted by its two components as $C_{jm} = \left[c_{jm1} \quad c_{jm2} \right]$, where c_{jm1} and c_{jm2} can assume any discrete value mapped in the modulation constellation.

The vector R_{jm} represents the two outputs from the demodulator, i.e., the Viterbi decoder inputs, corresponding to the m^{th} symbol of the j^{th} branch in the trellis. It is expressed as $R_{jm} = \left[r_{jm1} \quad r_{jm2} \right]$, where r_{jm1} and r_{jm2} are shown in Figure 3.2.

The vector N_{jm} is the AWGN vector representation like the one in Equation (3.5) that affects the reception of the m^{th} symbol of the j^{th} branch in the trellis. Equation (3.2) can be written in a vector form as

$$R_{jm} = C_{jm} + N_{jm}. \tag{3.6}$$

Consider the following metric definition for the j^{th} branch of the i^{th} path in the trellis:

$$\begin{aligned} m_j^{(i)} &= \log f_R(\{R_{j1}\ R_{j2}\ R_{j3}\ \ldots\ R_{jN}\ \} \mid \{C_{j1}^{(i)}\ \ C_{j2}^{(i)}\ \ C_{j3}^{(i)}\ \ \ldots\ \ C_{jN}^{(i)}\}) \\ &= \log f_R(\{R_{jm},\ m = 1, 2, \ldots N\} \mid \{C_{jm}^{(i)},\ m = 1, 2, \ldots N\}) \\ &= \log f_R(\{R_{jm}\} \mid \{C_{jm}^{(i)}\}) \end{aligned}, \tag{3.7}$$

where N is the number of coded symbols transmitted per information symbol.

Since R_{j1} is independent of R_{jm} for any m different than 1, it follows

$$\log f_R(\{R_{jm}\} \mid \{C_{jm}^{(i)}\}) = \prod_{m=1}^{N} f_R(R_{jm} \mid \{\ C_{jm}^{(i)}\ \}). \tag{3.8}$$

Considering that the reception of R_{j1} is only being affected by C_{j1}, the following relationship stands

$$f_R(R_{jm} \mid \{\ C_{jm}^{(i)}\ \}) = f_R(\ R_{jm} \mid C_{jm}^{(i)}\ ). \tag{3.9}$$

Substituting Equation (3.9) in (3.8), and Equation (3.8) in (3.7) yields

$$m_j^{(i)} = \log \prod_{m=1}^{N} f_R(R_{jm} \mid C_{jm}^{(i)}) = \sum_{m=1}^{N} \log f_R(R_{jm} \mid C_{jm}^{(i)}). \tag{3.10}$$

However, due to the orthogonality of components 1 and 2 as well as the characteristics of the coherent detection employed, the conditional probability of Equation (3.10) can be derived as

$$\begin{aligned} f_R(R_{jm} \mid C_{jm}^{(i)}) &= f_R(r_{jm1} \mid c_{jm1}^{(i)})\ f_R(r_{jm2} \mid c_{jm2}^{(i)}) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}}\ e^{\left[\frac{-(r_{jm1} - c_{jm1}^{(i)})^2}{2\sigma^2}\right]} . \frac{1}{\sqrt{2\pi\sigma^2}}\ e^{\left[\frac{-(r_{jm2} - c_{jm2}^{(i)})^2}{2\sigma^2}\right]}, \end{aligned}$$

which implies

$$f_R(R_{jm} \mid C_{jm}^{(i)}) = \frac{1}{2\pi\sigma^2}\ e^{\left[\frac{-1}{2\sigma^2} \cdot \| R_{jm} - C_{jm}^{(i)} \|^2\right]}. \tag{3.11}$$

Substituting Equation (3.11) in (3.10) results in

$$m_j^{(i)} = \sum_{m=1}^{N} \log \left[\frac{1}{2ps^2} \; e^{\left[\frac{-1}{2s^2} \cdot \| R_{jm} - C_{jm}^{(i)} \|^2 \right]} \right]$$
$$= N \log \frac{1}{2ps^2} + \log(e) . \sum_{m=1}^{N} \left[-\frac{1}{2s^2} \| R_{jm} - C_{jm}^{(i)} \|^2 \right] .$$

Since N is a constant for the considered convolutional code, the above equation can be rewritten as follows

$$m_j^{(i)} = K_1 + K_2 \sum_{m=1}^{N} \| R_{jm} - C_{jm}^{(i)} \|^2 , \tag{3.12}$$

where

$$K_1 = N \log\left(\frac{1}{2ps^2} \right) , \text{ and } K_2 = -\frac{1}{2s^2} \log e .$$

As seen in Section III.A, the Soft-Decision Viterbi decoder computes an entire path metric for all paths that reach an intended stage in order to discard the lower metrics and keep the highest one. The correlation metric for this decoder is given by Equation (3.1) and rewritten below:

$$CM^{(i)} = \sum_{j=1}^{B} m_j^{(i)} . \tag{3.13}$$

Substituting Equation (3.12) in (3.13) yields

$$CM^{(i)} = \sum_{j=1}^{B} \left(K_1 + K_2 \sum_{m=1}^{N} \| R_{jm} - C_{jm}^{(i)} \|^2 \right)$$
$$= BK_1 + K_2 \sum_{j=1}^{B} \sum_{m=1}^{N} \| R_{jm} - C_{jm}^{(i)} \|^2 . \tag{3.14}$$

Considering the linear property of the convolutional codes, a sequence of all-zero symbols (C_0) can be assumed to be transmitted without any loss of generality. Following this transmission pattern, the Viterbi decoder should choose the all-zero symbol path, called path 0, as the correct path to be processed.

Suppose that, due to channel errors, the Viterbi decoder chooses an incorrect path 1 instead of the correct path 0. Also assume that this wrong path 1 contains *d* erroneous symbols, i.e., *d* symbols different than C_0. Therefore, an initial formula for $P_2(d)$ can be written as

$$P_2(d) = \Pr\left[CM^{(1)} \geq CM^{(0)}\right] = \Pr\left[CM^{(1)} - CM^{(0)} \geq 0\right]. \tag{3.15}$$

Substituting Equation (3.14) in (3.15) results in

$$\begin{aligned} P_2(d) &= \Pr\left[BK_1 + K_2 \sum_{j=1}^{B}\sum_{m=1}^{N} \| R_{jm} - C_{jm}^{(1)} \|^2 - BK_1 - K_2 \sum_{j=1}^{B}\sum_{m=1}^{N} \| R_{jm} - C_{jm}^{(0)} \|^2 \geq 0\right] \\ &= \Pr\left[K_2 \sum_{j=1}^{B}\sum_{m=1}^{N} \left(\| R_{jm} - C_{jm}^{(1)} \|^2 - \| R_{jm} - C_{jm}^{(0)} \|^2\right) \geq 0\right] \end{aligned}.$$

In Equation (3.12) recall that K_2 is a negative constant, which leads to

$$P_2(d) = \Pr\left[\sum_{j=1}^{B}\sum_{m=1}^{N}\left(\| R_{jm} - C_{jm}^{(1)} \|^2 - \| R_{jm} - C_{jm}^{(0)} \|^2\right) \leq 0\right]. \tag{3.16}$$

Considering that a sequence of all-zero symbols is the one being transmitted, the following stands

$$\begin{aligned} C_{jm}^{(0)} &= C_0, \quad \forall\, j \text{ and } m \\ R_{jm} &= C_0 + N_{jm}. \end{aligned} \tag{3.17}$$

Replacing Equations (3.17) in (3.16) yields

$$P_2(d) = \Pr\left[\sum_{j=1}^{B}\sum_{m=1}^{N}\left(\| C_0 + N_{jm} - C_{jm}^{(1)} \|^2 - \| C_0 + N_{jm} - C_0 \|^2\right) \leq 0\right]$$

$$= \Pr\left[\sum_{j=1}^{B}\sum_{m=1}^{N}\left(\| \left(C_{jm}^{(1)} - C_0\right) - N_{jm} \|^2 - \| N_{jm} \|^2\right) \leq 0\right]$$

$$= \Pr\left[\sum_{j=1}^{B}\sum_{m=1}^{N}\left(\| C_{jm}^{(1)} - C_0 \|^2 - 2\left(C_{jm}^{(1)} - C_0\right).N_{jm}\right) \leq 0\right]. \tag{3.18}$$

Recalling that $C_{jm}^{(1)} = C_{jm}^{(0)} = C_0$, for all combinations of j and m except for d symbols, Equation (3.18) can be rewritten as follows

$$P_2(d) = \Pr\left[\sum_{l=1}^{d} \left(\| C_l^{(1)} - C_0 \|^2 - 2 (C_l^{(1)} - C_0) N_l \right) \le 0 \right]. \tag{3.19}$$

Expanding the noise vector in its orthogonal components in Equation (3.19) results in

$$P_2(d) = \Pr\left[\sum_{l=1}^{d} \left(\| C_l^{(1)} - C_0 \|^2 - 2 (c_{l1}^{(1)} - c_{01}) n_{l1} - 2 (c_{l2}^{(1)} - c_{02}) n_{l2} \right) \le 0 \right]. \tag{3.20}$$

At this point, notice that $P_2(d)$ will be very dependent on the wrong symbol sequence $C_l^{(1)}$, for l = 1, 2, ... d. However, for a given sequence of $C_l^{(1)}$, i.e., path 1, it can be seen that the left side of the inequality in Equation (3.20) is a Gaussian random variable due to n_{l1} and n_{l2} being zero-mean Gaussian r.v.'s and the other terms being deterministic values. This random variable will be called X, which leads to

$$X = N\left(\bar{X}, \; s_X^2 \right) = N\left(\sum_{l=1}^{d} \bar{X}_l, \; \sum_{l=1}^{d} s_{Xl}^2 \right), \tag{3.21}$$

where

$$\bar{X}_l = \| C_l^{(1)} - C_0 \|^2, \tag{3.22}$$

and

$$s_{Xl}^2 = 4\left((c_{l1}^{(1)} - c_{01})^2 + (c_{l2}^{(1)} - c_{02})^2 \right) s^2 = 4 \| C_l^{(1)} - C_0 \|^2 s^2$$

$$= 4 \bar{X}_l s^2. \tag{3.23}$$

Here an upper bound can be established by assuming that all $C_l^{(1)}$'s, for every l, are equal to a worst-case symbol, C_W, in such a way that $P_2(d)$ becomes the highest possible.

If C_W is a constant vector for every l, then Equation (3.21) can be rewritten as

$$X = N(\bar{X}, \; s_X^2) = N(d\bar{X}_W, \; d s_{XW}^2). \tag{3.24}$$

The problem now is to find a symbol C_W different than C_0 among all possibilities within the modulation constellation that could be the worst-case symbol. For this purpose, a useful theorem should be derived involving normal distributions.

Theorem: consider two Gaussian r.v.'s X and Y

$$N\left(\overline{X}, s_X^2\right) \text{ and } N\left(\overline{Y}, s_Y^2\right),$$

then

$$\Pr[X \le 0] \ge \Pr[Y \le 0] \Leftrightarrow \overline{X}\, s_Y \le \overline{Y}\, s_X .$$

Proof:

$$\Pr[X \le 0] \ge \Pr[Y \le 0]$$

$$1 - Q\left(\frac{0-\overline{X}}{s_X}\right) \ge 1 - Q\left(\frac{0-\overline{Y}}{s_Y}\right)$$

$$1 - \left(1 - Q\left(\frac{\overline{X}}{s_X}\right)\right) \ge 1 - \left(1 - Q\left(\frac{\overline{Y}}{s_Y}\right)\right)$$

$$Q\left(\frac{\overline{X}}{s_X}\right) \ge Q\left(\frac{\overline{Y}}{s_Y}\right).$$

Since the Q-function is a descending monotonic function, in order for the above inequality to be true, the following stands

$$\frac{\overline{X}}{s_X} \le \frac{\overline{Y}}{s_Y}$$

$$\overline{X} s_Y \le \overline{Y} s_X .$$

Notice that the reverse also works, which concludes the theorem's proof.

By the above theorem, it can be stated that, within a set of Gaussian r.v.'s with different means $\{X_i, i = 1, 2, \ldots, N\}$, the one that has the lowest mean will comply with the following property

$$\Pr\left[X_k \le 0\right] \ge \Pr\left[X_{i\neq k} \le 0\right], \text{ where } \overline{X}_k \le \overline{X}_{i\neq k}, \forall\, i\,,$$

except if one of the other r.v.'s with a higher mean presents such a large standard deviation that makes

$$\overline{X}_k s_{i\neq k} > \overline{X}_{i\neq k} s_k\,, \tag{3.25}$$

which would hurt the theorem.

Therefore, C_W is thus chosen to make the mean in Equation (3.24) the lowest possible and then checked if there is any other remaining choice that would make the Inequality (3.25) true.

Looking at Equation (3.22), it can be seen that the minimum mean in Equation (3.24) is obtained by choosing a C_W that makes

$$||C_W - C_0||^2 = D_{0,\min}^2\,, \tag{3.26}$$

where $D_{0,\min}$ is the *Euclidean distance* between C_0 and its closest neighbors in the constellation. Notice that there is at least one C_W for every constellation, but in some cases, there are more than one. However, for those cases, any closest neighbor chosen to be C_W will yield the same statistical results (mean and variance) because of Equations (3.22) and (3.23).

Substituting Equation (3.26) in Equations (3.22) and (3.23) yields

$$\overline{X}_W = D_{0,\min}^2\,, \tag{3.27}$$

and

$$s_{XW}^2 = 4 D_{0,\min}^2 s^2\,. \tag{3.28}$$

The next step is to determine if there is another possible choice of C_W, called C_{WW}, which would comply with the following relationships

$$\overline{X}_{WW} > \overline{X}_W \quad \text{and} \quad \overline{X}_W s_{X_{WW}} > \overline{X}_{WW} s_{X_W}\,, \tag{3.29}$$

and could make the exception pointed by inequality (3.25) true.

With Equations (3.22) and (3.23), the second inequality at (3.29) becomes

$$\bar{X}_W \sqrt{4\,\bar{X}_{WW}\,s^2} > \bar{X}_{WW} \sqrt{4\,\bar{X}_W\,s^2}\,.$$

Since both $\bar{X}_W$ and $\bar{X}_{WW}$ are positive values due to Equation (3.22), it is possible to square both sides of the above inequality and obtain

$$4s^2 \bar{X}_W^2 \bar{X}_{WW} > 4s^2 \bar{X}_{WW}^2 \bar{X}_W$$

$$\bar{X}_W > \bar{X}_{WW}\,.$$

This latter result hurts the hypothesis formulated by the first inequality in Equation (3.29). It means that C_{WW} does not really exist and the choice of C_W as the closest symbol to C_0 can be definitely considered the one that will conduct to an upper bound for $P_2(d)$ in Equation (3.20).

Finally, it should be recalled that the left side of the inequality in Equation (3.20) represents a Gaussian random variable defined by Equation (3.24). Thus, an upper bound for $P_2(d)$ can be set as

$$\begin{aligned} P_2(d) &\le \Pr[X \le 0] \\ P_2(d) &\le 1 - Q\left(\frac{0 - d\bar{X}_W}{\sqrt{d s_{XW}^2}}\right) = Q\left(\frac{d\bar{X}_W}{\sqrt{d s_{XW}^2}}\right). \end{aligned} \tag{3.30}$$

Substituting Equation (3.27) and (3.28) in (3.30) yields

$$P_2(d) \le Q\left(\frac{d\,D_{0,\min}^2}{\sqrt{4d\,D_{0,\min}^2\,s^2}}\right) = Q\left(\sqrt{\frac{d^2\,D_{0,\min}^4}{4d\,D_{0,\min}^2\,s^2}}\right) = Q\left(\sqrt{\frac{d\,D_{0,\min}^2}{4\,s^2}}\right),$$

which implies

$$P_2(d) \le Q\left(\sqrt{\frac{d\,D_{0\min}^2}{2N_0}}\right), \tag{3.31}$$

where the last step stands due to $s^2 = N_0/2$, expressed in Equation (3.4).

Notice that $D_{0,\min}$ can vary depending on the C_0 chosen to be the zero-symbol in the constellation. Therefore, in order to guarantee that Inequality (3.31) will provide the

real upper bound, C_0 should be chosen to be the one that yields $D_{0,\min} = D_{min}$, where D_{min} is the *minimum Euclidean distance* in the constellation. Thus, the final formula to determine $P_2(d)$ becomes

$$P_2(d) \leq Q\left(\sqrt{\frac{dD_{\min}^2}{2N_0}}\right). \tag{3.32}$$

Due to the orthogonality achieved by coherent demodulators, all the assumptions thus far are also valid for multi-dimensional M-ary modulations such as M-FSK, as well as unidimensional ones such as BPSK. Consequently, $P_2(d)$ found in Inequality (3.32) can be considered a general upper bound to be applied to every *M-ary coherent demodulator* followed by a *soft-decision Viterbi decoder.*

2. Upper Bound of $P_2(d)$ for M-QAM

In this Section, Equation (3.32) will be customized for the cases of BPSK and QPSK, special cases of M-QAM in which $M = 2$ and $M = 4$, respectively, as well as two other M-QAM coherent demodulators, 16-QAM and 64-QAM. All of them will be considered to be using rectangular constellations. Notice that Equation (3.32) can be applied to every type of constellation. The particularization for the rectangular one results since this type is the most frequently used in practice [3].

For rectangular constellations, regardless of M, D_{min} is always equal to a constant $2A$, where A is one unit of the grid represented in Figure 2.1. Hence, it stands

$$D_{\min}^2 = 4A^2. \tag{3.33}$$

Applying Equation (3.33) in (3.32) leads to

$$P_2(d) \leq Q\left(\sqrt{\frac{d\left(4A^2\right)}{2N_0}}\right) = Q\left(\sqrt{\frac{2dA^2}{N_0}}\right). \tag{3.34}$$

Equation (3.34) will be used to derive the well-known formula for a BPSK demodulator followed by a soft-decision Viterbi decoder [3]:

$$P_2(d)_{BPSK} \le Q\left(\sqrt{2\gamma_b R_C d}\right).$$

Assume the following symbology:

- γ_{cs} is the average energy of one coded symbol E_{CS} / N_0 (average SNR for one coded symbol),
- γ_{cb} is the average energy of one coded bit / N_0 (average SNR for one coded bit), and
- γ_b is the average energy of one information bit / N_0 (average SNR for one information bit).

As regards these parameters, it can be stated that:

$$\gamma_{cs} = \frac{E_{cs}}{N_0}, \tag{3.35}$$

$$\gamma_{cs} = q\gamma_{cb}, \tag{3.36}$$

and

$$\gamma_{cb} = R_C \gamma_b. \tag{3.37}$$

Substituting Equations (3.37) in (3.36) and the result into Equation (3.35) yields

$$E_{cs} = q R_C \gamma_b N_0. \tag{3.38}$$

For BPSK, considering each coded symbol to be equiprobable to occur in the channel, the energy per coded symbol can be expressed by:

$$E_{cs} = E\left[E_{cs_i}\right] = \sum_{i=1}^{2} \Pr[\text{symbol } C_i] E_{cs_i} = \frac{1}{2}\left(2A^2\right) = A^2. \tag{3.39}$$

Replacing the results of Equations (3.38) and (3.39) in (3.34) leads to

$$P_2(d) \le Q\left(\sqrt{\frac{2dE_{cs}}{N_0}}\right) = Q\left(\sqrt{\frac{2dqR_C\gamma_b N_0}{N_0}}\right) = Q\left(\sqrt{2dqR_C\gamma_b}\right). \tag{3.40}$$

Recalling that q = 1 bit for BPSK, Equation (3.40) becomes

$$P_2(d)_{BPSK} \le Q\left(\sqrt{2dR_C\gamma_b}\right). \tag{3.41}$$

This latter result is exactly the one expected to be found.

For the QPSK, the energy per coded symbol can be written as:

$$e_{cs} = E\left[e_{cs_i}\right] = \sum_{i=1}^{4} \Pr\left[\text{symbol } C_i\right] e_{cs_i} = 4\frac{1}{4}\left(2A^2\right) = 2A^2 . \tag{3.42}$$

Replacing the results of Equations (3.38) and (3.42) in (3.34) leads to

$$P_2(d) \le Q\left(\sqrt{\frac{2d\dfrac{e_{cs}}{2}}{N_0}}\right) = Q\left(\sqrt{\frac{dqR_C g_b N_0}{N_0}}\right) = Q\left(\sqrt{dqR_C g_b}\right). \tag{3.43}$$

Recalling that q = 2 bits for QPSK, Inequality (3.43) assumes its final format as

$$P_2(d)_{QPSK} \le Q\left(\sqrt{2dR_C g_b}\right). \tag{3.44}$$

Notice that this result is identical to the one obtained for BPSK.

Now is the time to see how the upper bound of $P_2(d)$ for the 16-QAM behaves. The constellation is shown in Figure 2.1. Notice that Equations (3.34) and (3.6) are the same, even though q changes to $q = \log_2 16 = 4$ bits.

Using the same arguments as for BPSK, the following computation stands

$$e_{cs} = E\left[e_{cs_i}\right] = \sum_{i=1}^{16} \Pr\left[C_i\right] e_{cs_i} = \frac{1}{16} 4(2A^2 + 10A^2 + 10A^2 + 18A^2) = 10A^2 . \tag{3.45}$$

Replacing Equation (3.45) in (3.34) causes

$$P_2(d) \le Q\left(\sqrt{\frac{2d\dfrac{e_{cs}}{10}}{N_0}}\right) = Q\left(\sqrt{\frac{de_{cs}}{5N_0}}\right). \tag{3.46}$$

Finally, substituting Equation (3.38) in (3.46) yields

$$P_2(d)_{16-QAM} \le Q\left(\sqrt{\frac{dqR_C g_b N_0}{5N_0}}\right),$$

which results in

$$P_2(d)_{16-QAM} \le Q\left(\sqrt{\frac{4}{5} dR_C g_b}\right). \tag{3.47}$$

For the 64-QAM, the development is the same but $q = \log_2 64 = 6$ bits, leading to

$$e_{cs} = E\left[e_{cs_i}\right] = \sum_{i=1}^{16} \Pr[C_i] e_{cs_i} = 42A^2. \tag{3.48}$$

Replacing Equation (3.48) in (3.34) results in

$$P_2(d) \le Q\left(\sqrt{\frac{2d\frac{e_{cs}}{42}}{N_0}}\right) = Q\left(\sqrt{\frac{de_{cs}}{21N_0}}\right). \tag{3.49}$$

Therefore, by substituting Equation (3.38) in (3.46), the final formula becomes

$$P_2(d)_{64-QAM} \le Q\left(\sqrt{\frac{dqR_C g_b N_0}{21N_0}}\right) = Q\left(\sqrt{\frac{6dR_C g_b}{21}}\right),$$

which yields

$$P_2(d)_{64-QAM} \le Q\left(\sqrt{\frac{2}{7} dR_C g_b}\right). \tag{3.50}$$

3. Improvements on the Upper Bound of $P_2(d)$

The core of the analytical development of the latter upper bound for $P_2(d)$ resides in Equation (3.21). The worst-case scenario for the result of this equation was assumed. However, some questions can be posed. Is the worst-case assumption too conservative for a certain application in terms of leading to a loose upper bound? Can different hypotheses other than the worst case be assumed and still have a reliable but tighter upper bound?

How far can this process go? The answers to these questions are the subject of this Section.

Rewrite Equation (3.21) in a different fashion by replacing Equations (3.22) and (3.23) in it.

$$f_X(x) \;=\; N\left(\sum_{l=1}^{d} \| C_l^{(1)} - C_0 \|^2,\; 4s^2 \sum_{l=1}^{d} \| C_l^{(1)} - C_0 \|^2\right). \tag{3.51}$$

The worst case was shown to happen when all d symbols are the ones closest to $\boldsymbol{C_0}$ in the constellation. In this case $\| C_l^{(1)} - C_0 \|^2 = D_{\min}^2$, which led the summation in Equation (3.51) to be the smallest possible, $d\, D_{\min}^2$, and the final result to be Equation (3.32).

Notice, however, that there are many symbol combinations for filling all d slots when $d = 3$ and $M = 16$. The probability of all symbols being the closest ones to $\boldsymbol{C_0}$ tends to be very small for those cases. Even though small these special symbol combinations must be considered for the worst-case scenario, in which 100% of all possible combinations result in $\sum_{l=1}^{d} \| C_l^{(1)} - C_0 \|^2 \geq d D_{\min}^2$. Consider all combinations that lead to the second smallest possible value for the latter summation instead of the smallest one (worst case). The percentage of the total d-symbol combinations this second smallest value will cover can be expressed as

$$\begin{aligned} P_{\text{cov}} &= 1 - \Pr\left[\text{all } d \text{ symbols have distance } D_{\min} \text{ to } C_0\right] \\ &= 1 - \Pr\left[\| C_l^{(1)} - C_0 \|^2 = D_{\min}^2, \text{for } l = 1, 2, \ldots, d\right] = 1 - P_{wc}. \end{aligned} \tag{3.52}$$

If the worst-case probability P_{wc} for a certain link is very small, the second smallest possible value for the summation could be assumed as the worst case without significant losses in generality.

Consider the definition of an average distance, D_{AV}, dependent on the specific combination of all d symbols chosen to be the wrong path, given by:

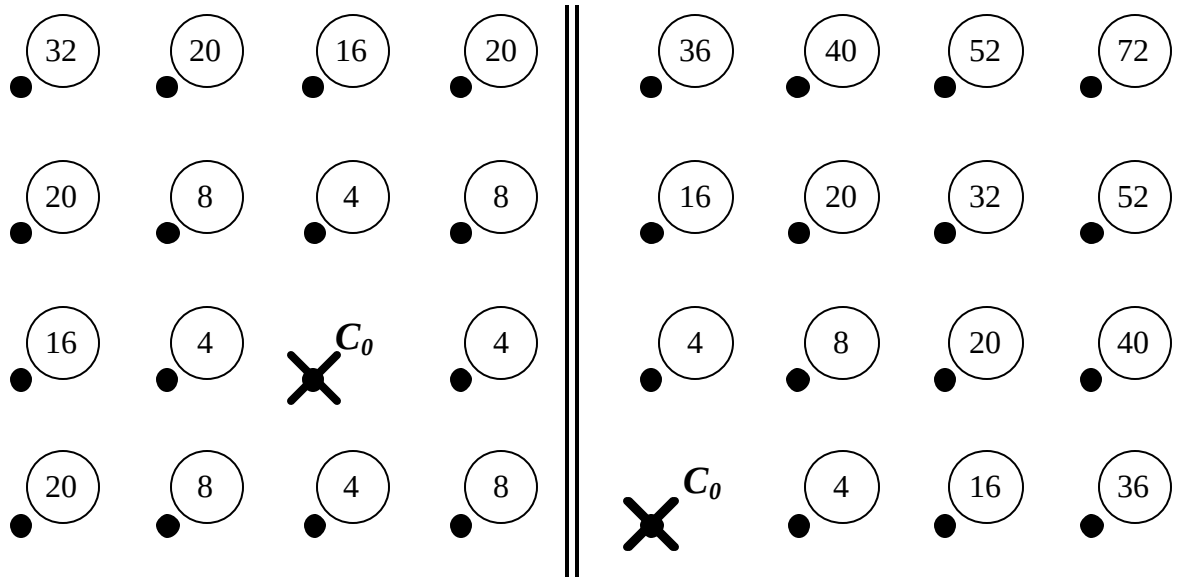
$$\sum_{l=1}^{d} \| C_l^{(1)} - C_0 \|^2 = dD_{AV}^2$$

$$D_{AV}^2 = \frac{1}{d} \sum_{l=1}^{d} \| C_l^{(1)} - C_0 \|^2. \tag{3.53}$$

Notice, in the worst case that $D_{AV}^2 = D_{\min}^2$, and for all other cases, $D_{AV}^2 \geq D_{\min}^2$. The idea of making a different choice for the worst-case sequence impacts directly in Equation (3.32) in which $D_{\min}^2$ will be replaced by D_{AV}^2. Even if D_{AV}^2 is just a little bit greater than $D_{\min}^2$, due to the Q-function, the parameter $P_2(d)$ will be considerably reduced.

The bigger the parameters *d* and *M*, the smaller the worst-case probability P_{wc} in Equation (3.52) will be, and consequently, the closer to 100% the P_{cov} will become, and the closer to $D_{\min}^2$ the parameter D_{AV}^2 will reach. Recall that *d* is at least equal to d_{free}, which usually is greater than 3 for good convolutional codes. Summarizing the issue, the main goal to improve $P_2(d)$ would be to determine an acceptable D_{AV}^2 for $d = d_{free}$ that can replace $D_{\min}^2$ in Equation (3.32). Even though the effort begins by considering $d = d_{free}$, as will be seen later, the same D_{AV}^2 found for d_{free} works rather well for the other terms of $P_2(d)$ with *d* greater than d_{free}.

Applying Equation (3.53) for all possible *d*-symbol combinations, it can be seen that D_{AV}^2 lies on a finite and discrete domain of values. Therefore, the best way to study the possible choices of D_{AV}^2 is to plot its *discrete probability density function* (dpdf) for a certain d_{free}. Notice, however, that the dpdf will be very dependent on the symbol in the constellation chosen to be $\boldsymbol{C_0}$. Figure 3.3 shows a 16-QAM constellation with two different possible choices of $\boldsymbol{C_0}$. The circled numbers indicate the square of the Euclidean distance, D^2, normalized by the grid unit A^2 for every symbol different than the considered $\boldsymbol{C_0}$. For every *l* in Equation (3.53), each correspondent term $\| C_l^{(1)} - C_0 \|^2$ can assume any value of D^2 represented in Figure 3.3. Notoriously, the numbers for configuration (b) are much greater than (a).



(a) - C_0 is one of the most internal symbols

(b) - C_0 is one of the corner symbols

Figure 3.3. Square of the Normalized Euclidean Distance (D^2/A^2) Related to the 16-QAM Rectangular Constellation Symbols for 2 Different Choices of C_0.

The choices of C_0 that provide larger values for $||C_l^{(1)} - C_0||^2$ will lead to greater D_{AV}^2 and, consequently, will achieve better results for $P_2(d)$, i.e., a smaller $P_2(d)$. In order to be conservative, this study will focus on rectangular constellations choosing C_0 to be the one resulting in the smallest D_{AV}^2 and the largest $P_2(d)$, which is the worst choice available in terms of $P_2(d)$. For any rectangular constellation, it is easy to see that the conservative choice of C_0 falls on any one of the most internal symbols because they present the largest number of neighbors with Euclidean distance equal to D_{min}. The option (a) in Figure 3.3 represents this choice.

Consider a practical example. Suppose the analysis of D_{AV}^2 for a 16-QAM rectangular constellation is associated with a dual-4 convolutional code, i.e., dual-k in which $k = 4$ to match the 4-bit coded symbols of the 16-QAM. In the Appendix it is

shown that d_{free} = 4 for dual-k convolutional codes. Thus, by computing the dpdf of D_{AV}^2 for all possible sequences of 4 symbols in which each symbol can be any of the 15 symbols different than $\boldsymbol{C_0}$, the plot shown in Figure 3.4 can be obtained. The set of different values for D_{AV}^2 , i.e., the horizontal axis, are again normalized by the factor A^2.

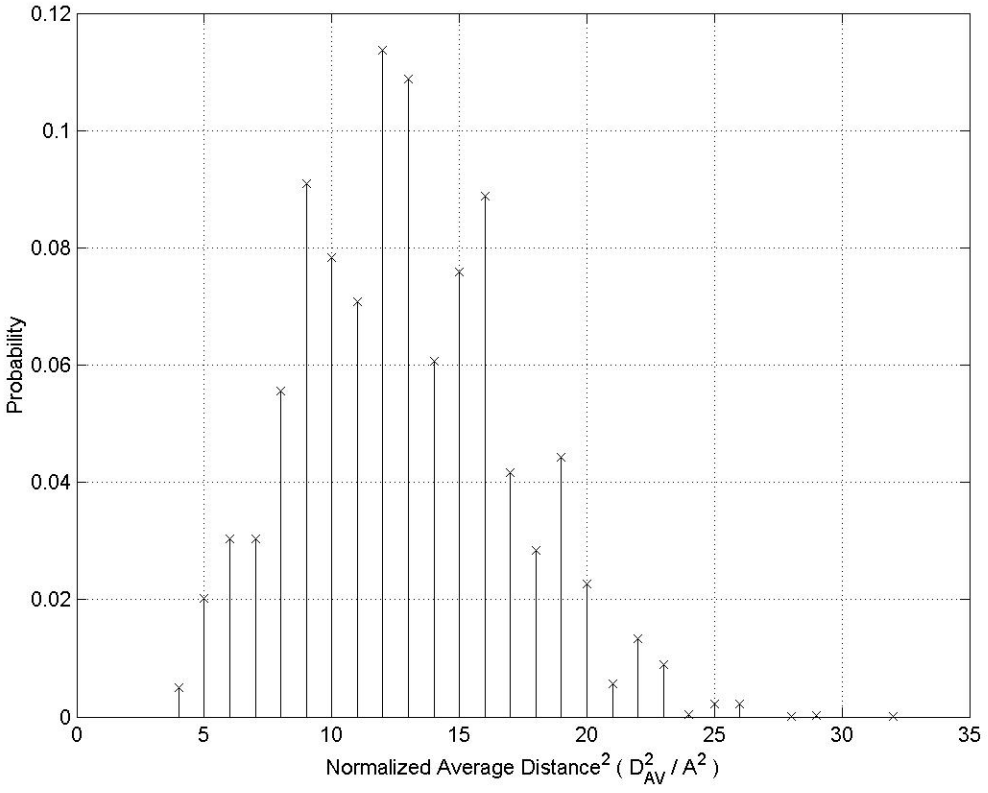


Figure 3.4. Dpdf of D_{AV}^2 for a 16-QAM Rectangular Constellation, d = 4, and Conservative Choice of $\boldsymbol{C_0}$.

Even though the number of possible combinations for a sequence of 4 symbols is equal to $15^4 = 50625$, due to Equation (3.53), those combinations result in only 26 different values for D_{AV}^2 . The point where $D_{AV}^2 / A^2 = 4$ represents the probability of one worst-case sequence ($D_{AV}^2 = D_{\min}^2$) occurring. This value can also be determined analytically by considering the existence of 4 different closest symbols to $\boldsymbol{C_0}$ in the

constellation. Thus, there are 4^4 possible sequences that will result from $D_{AV}^2 = D_{\min}^2$. The probability that one of these sequences occurs is given by

$$P_{wc} = \frac{4^4}{15^4} = 0.00506 = 0.5\%.$$

Substituting this result in Equation (3.52), it can be stated that the choice of $D_{AV}^2 = 5A^2$ instead of $D_{\min}^2 = 4A^2$ would be valid for 99.5% of all possible transmissions in the channel and could lead to $d = d_{free} = 4$. What about the other values of $P_2(d)$ used for the calculation of P_b in which $d > d_{free}$? Since their d is different from d_{free}, should a specific D_{AV}^2 be determined for each d or can the the value obtained for d_{free} be applied to all d's?

Notice that when d increases, the domain of the possible D_{AV}^2 also increases and all probabilities of the previous dpdf will be diluted along this new longer domain. Consequently, by using a certain threshold determined for d_{free} in a path presenting $d > d_{free}$, the probability of obtaining D_{AV}^2 greater than this threshold is greater than this equivalent probability when using the same threshold for a path in which $d = d_{free}$. In other words, a threshold that is good for d_{free} will be even better for $d > d_{free}$. To summarize, the analytical conclusion of the 16-QAM example can be restated as: "... the choice of $D_{AV}^2 = 5A^2$ instead of $D_{\min}^2 = 4A^2$ would be valid for *at least* 99.5% of *all transmissions in the channel*".

Figure 3.5 shows the difference in performances for the discussed example. The curve for $P_{cov} = 99.5\%$ was determined following the equation

$$P_b = \frac{1}{k} \sum_{d=d_{free}}^{\infty} b(d)\, P_2(d)\Big|_{D_{\min}^2 = D_{AV}^2 = 5A^2}. \qquad (3.54)$$

However,

$$P_2(d)\Big|_{D_{\min}^2 = D_{AV}^2 = 5A^2} = Q\left(\sqrt{\frac{dD_{AV}^2}{2N_0}}\right) = Q\left(\sqrt{\frac{d\left(5A^2\right)}{2N_0}}\right) = Q\left(\sqrt{\frac{d\left(5A^2\right)e_{cs}}{2e_{cs}N_0}}\right)$$

$$P_2(d)\Big|_{D_{\min}^2 = D_{AV}^2 = 5A^2} = Q\left(\sqrt{\frac{d\left(5A^2\right)}{2e_{cs}} g_{cs}}\right) = Q\left(\sqrt{\frac{d\left(5A^2\right)}{2e_{cs}} qR_C g_b}\right) \tag{3.55}$$

Substituting Equation (3.45) in (3.55) yields

$$P_2(d)\Big|_{D_{\min}^2 = D_{AV}^2 = 5A^2} = Q\left(\sqrt{\frac{d\left(5A^2\right)}{2\left(10A^2\right)} q R_C g_b}\right) = Q\left(\sqrt{\frac{d}{2(2)} 4 R_C g_b}\right) = Q\left(\sqrt{d R_C g_b}\right). \tag{3.56}$$

Compare Equation (3.56) with (3.47) and notice that the argument of the Q-function is slightly smaller now, which results in a tighter upper bound. Substituting Equation (3.56) in (3.54) and recalling that $R_c = 1/2$, $d_{free} = 4$ and $k = 4$ for the dual-4 code, obtains

$$P_b = \frac{1}{4} \sum_{d=4}^{\infty} b(d) Q\left(\sqrt{\frac{1}{2} d g_b}\right). \tag{3.57}$$

Using the formula in the Appendix for $b(d)$ of dual-4 code leads to $b(d)$ = 15, 60, 570, 2820, and 18165, for d = 4, 5, 6, 7 and 8, respectively. Approximating Equation (3.57) with its five first terms, the final formula for P_b with P_{cov} = 95.5% in Figure 3.5 is obtained as

$$\begin{aligned} P_b = \frac{1}{4}\Big[& 15 Q\left(\sqrt{2 g_b}\right) + 60 Q\left(\sqrt{2.5 g_b}\right) + 570 Q\left(\sqrt{3 g_b}\right) \\ & + 2820 Q\left(\sqrt{3.5 g_b}\right) + 18165 Q\left(\sqrt{4 g_b}\right)\Big]. \end{aligned} \tag{3.58}$$

Notice that the simple assumption of $D_{AV}^2 = 5A^2$ made the upper bound for P_b tighter in 1 dB, which can make a difference when designing a communication link. As stated previously, this assumption is valid for 99.5% of the transmissions. However, the remaining 0.5% will still follow the original upper bound, which could be considered bad if 100% of the data were submitted to it, but considering that only 0.5% of the data will ever truly be submitted, this original upper bound can become very acceptable.

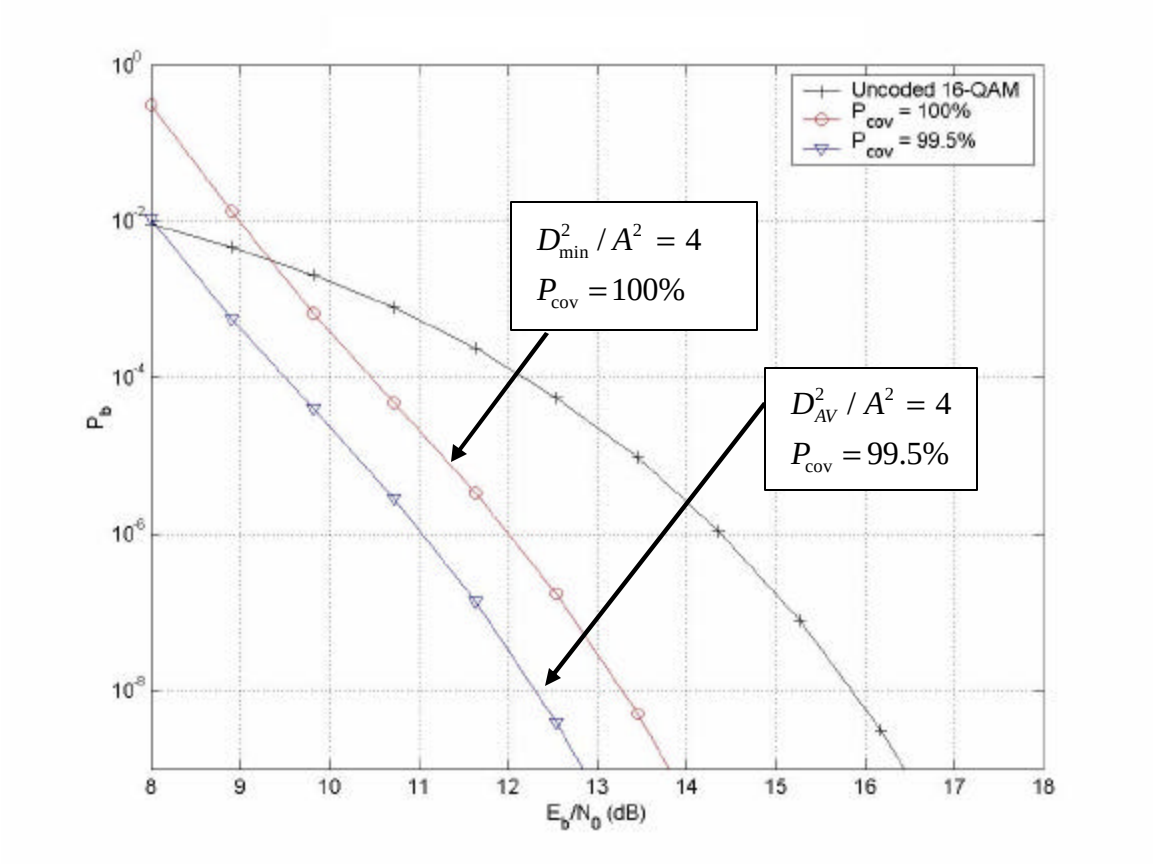


Figure 3.5. Difference on Performances when Changing $D^2_{\min}$ on the Calculation of the Upper Bound of $P_2(d)$ from $D^2_{\min} = 4A^2$ to $D^2_{AV} = 5A^2$ for a 16-QAM Channel with Dual-4 Convolutional Code and Soft-Decision Viterbi Decoding.

How far can the process of increasing $D^2_{\min}$ go? Can it be increased to a point where 95% of the channel data will follow the new correspondent upper bound? Probably yes, but the precise answer to these questions will only depend on the application intended for the communication link. Some practical experiments should be done in order to define the acceptance threshold. Anyway, this approach provides a tremendous enhancement to the upper bound, allowing a more realistic view of the behavior of some good, simple and cheap configurations that probably would be discarded at first glance if considering just the worst-case upper bound.

Recall that Equation (3.52) was created assuming that the second smallest possible value for D_{AV}^2 was chosen. However, for general choices that could be greater than the second smallest, this formula will change to

$$P_{\text{cov}}(D_{AV,n}^2) = 1 - \sum_{i=1}^{n-1} \Pr\left[\frac{1}{d}\sum_{l=1}^{d} || C_l^{(1)} - C_0 ||^2 = D_{AV,i}^2\right], \tag{3.59}$$

where $P_{\text{cov}}(D_{AV,n}^2)$ represents the percentage of the total d-symbol combinations that the n-th smallest possible value of D_{AV}^2 will cover. Note that $D_{AV,1}^2 = D_{\min}^2$ by the previous symbology.

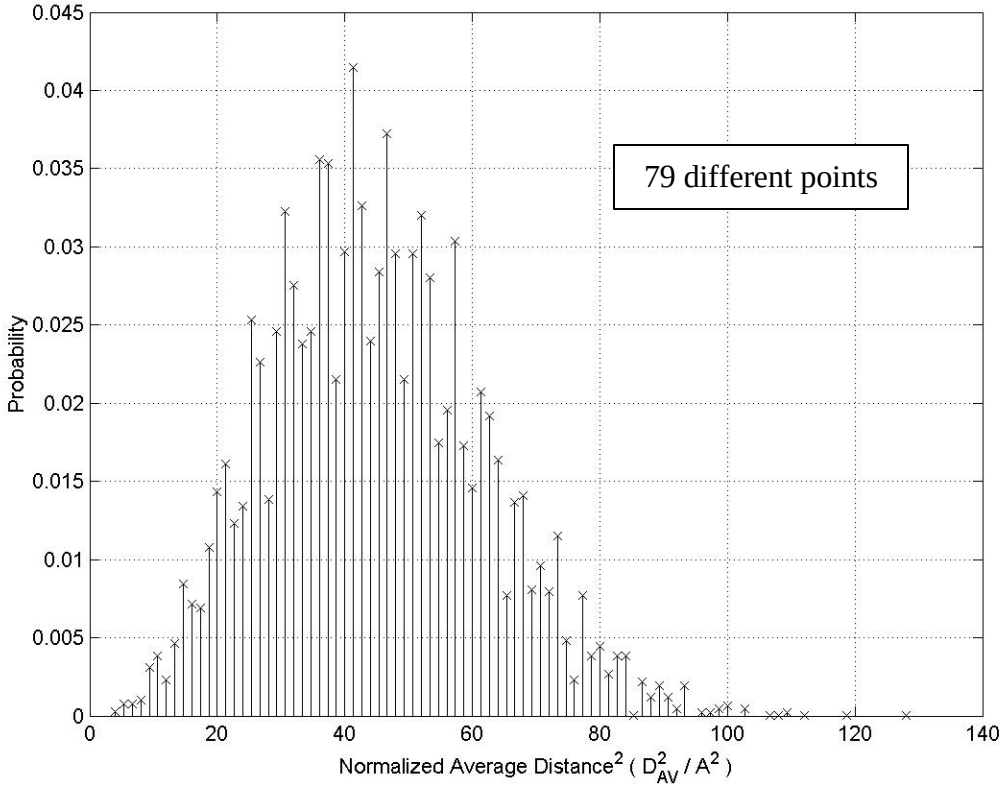


Figure 3.6. Dpdf of D_{AV}^2 for a 64-QAM Rectangular Constellation, d = 3, and Conservative Choice of $\boldsymbol{C_0}$.

Figures 3.6 and 3.7 describe two dpdf's for a same 64-QAM constellation in which the first plot corresponds to a path presenting d = 3, while, in the latter, d = 4. The

previously mentioned effect of diluting the dpdf of D_{AV}^2 / A^2 as d increases is easily identifiable in these figures. The number of possible values for D_{AV}^2 found in both cases were 79 and 110, respectively. Notice that in Figure 3.7, since the D_{AV}^2 domain is greater, the overall height of the dpdf is smaller than that in Figure 3.6. The probability of finding one value for D_{AV}^2 that is smaller than a certain fixed threshold reduces when the dpdf shortens.

As previously exposed, since the D_{AV}^2 determined for a path with d = d_{free} will be even better for another path with $d > d_{free}$, Figures 3.6 and 3.7 can be used to study the association of a 64-QAM modulator with different convolutional codes presenting $d_{free} = 3$ (Figure 3.6) and $d_{free} = 4$ (Figure 3.7). For instance, if $D_{AV}^2 / A^2 = 20$ is chosen for improving $P_2(d)$, by entering the data of the dpdf's in Equation (3.59), the probabilities that the improved upper bound covers all data transmission would be 95.0% for $d_{free} = 3$, and 97.2% for $d_{free} = 4$. Notice that for the first case, in which $d_{free} = 3$, there are 12 values in the horizontal axis of the dpdf in Figure 3.6 that precedes 20, whereas for $d_{free} = 4$, this number increases to 16.

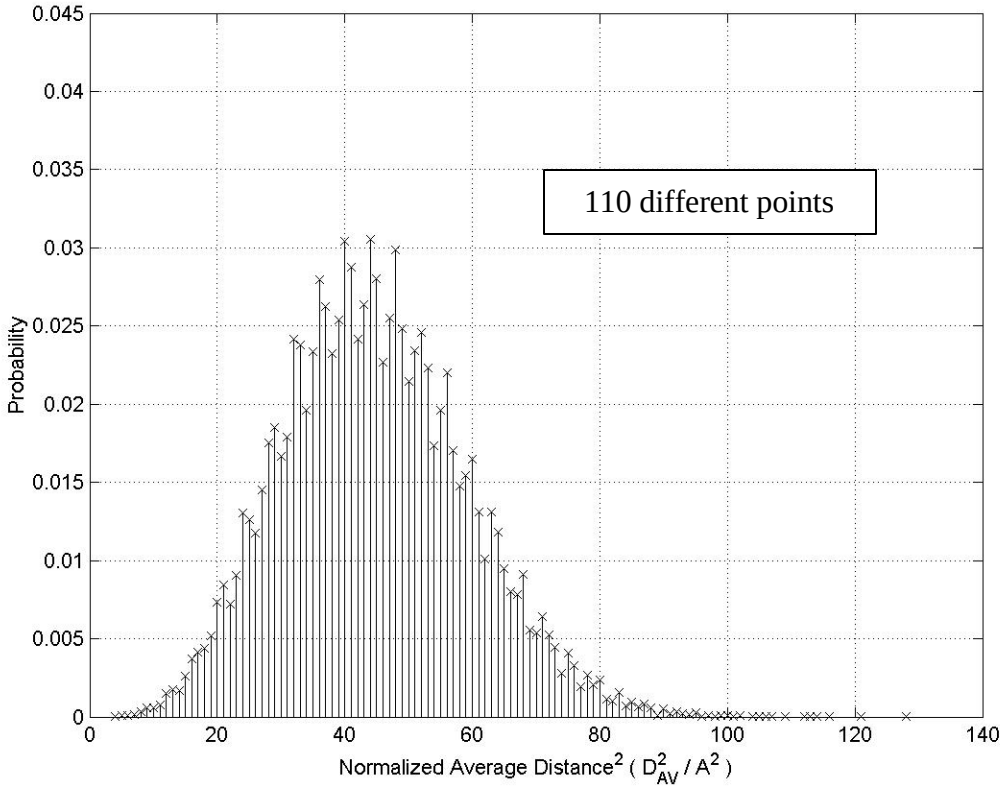


Figure 3.7. Dpdf of D_{AV}^2 for a 64-QAM Rectangular Constellation, d = 4, and Conservative Choice of $\boldsymbol{C_0}$.

Although there are 16 values for D_{AV}^2 / A^2 smaller than 20 to be considered in Figure 3.7, these values are very small due to the large possibilities of combining 63 symbols in 4 slots ($63^4 = 15{,}752{,}961$). Figure 3.8 shows a zoom-in on the first values of the dpdf in Figure 3.7. Notice how small the 16 values are. The sum of all of them is 0.028, and when applied to Equation (3.59), can result in 97.2%. If, instead of 97.2%, the specification for the upper bound was 99.5% of the data, a value of $D_{AV}^2 = 13A^2$ would be appropriate when looking up the dpdf in Figure 3.8.

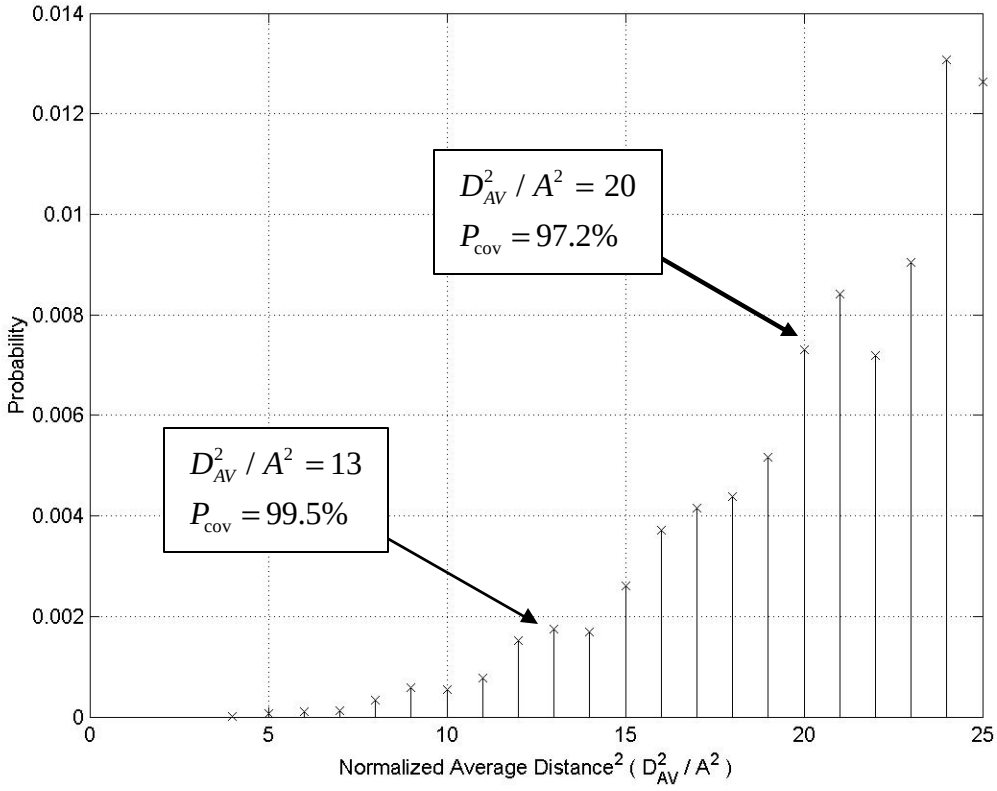


Figure 3.8. Initial Part of the dpdf of D_{AV}^2 for a 64-QAM Rectangular Constellation, d = 4, and Conservative Choice of $\boldsymbol{C_0}$.

As a final comment on this topic, this method of improving the upper bound for $P_2(d)$ results in breaking the complete independence between $b(d)$ and $P_2(d)$. Although most of the independence remains, when considerations are made about d_{free} applied to $P_2(d)$, they create an implicit link between the convolutional code and the demodulation scheme. In conclusion, the methodology herein exposed is directly dependent on the specific communication system being considered and not solely on an isolated $P_2(d)$.

D. PERFORMANCE ANALYSIS

Finding good convolutional codes is not easy. They are normally found by computational researches on a try-and-error basis. The goal of this Section is not to find the best convolutional code that can be applied to a certain M-QAM modulator, but to demonstrate how the concepts developed thus far in this thesis can be applied to measure realistic performances in communication systems when combining a certain M-QAM modulator with a certain convolutional code. It is much more a question of showing analysis tools and overall behaviors than obtaining the best results. A straightforward way to do that is to run performance analyses over a limited number of interesting target systems.

The focus will be concentrated on three convolutional codes and four different M-QAM modulators that will be presented in the following order for didactic purposes: 16-QAM, 64-QAM, BPSK, and QPSK. The codes were chosen due to either being well known or by having demonstrated outstanding performance during computational research. The plots of P_b versus E_b/N_0 were obtained from simulations taking into account the methods for determining $b(d)$ and the formulas for finding the upper bound of $P_2(d)$ and P_b presented thus far in this chapter. The improvements in the upper bound of $P_2(d)$, the subject of the latter Section, were also part of the results. Although the BPSK channel is not a non-binary channel, it has been included here for the purpose of comparison since its characteristics are usually well known to the reader and very well cited in the literature.

Sometimes the uncoded 8-QAM performance will be necessary for comparison between the options of sending coded or uncoded symbols through the channel. The 8-QAM modulator will then be assumed to have the constellation in Figure 3.9, which leads to the following expression for its uncoded performance:

$$P_b = \frac{2}{3} Q\left(\sqrt{\frac{D_{\min}^2}{2N_0}}\right) = \frac{2}{3} Q\left(\sqrt{g_b}\right). \tag{3.60}$$

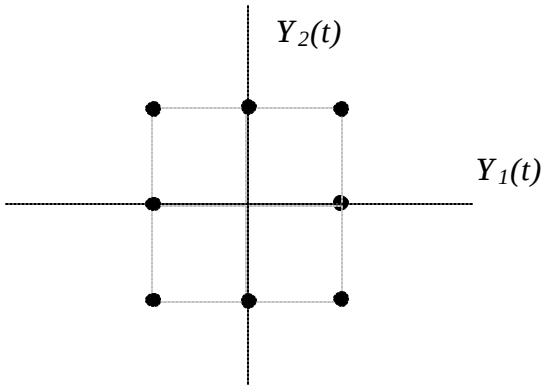


Figure 3.9. The 8-QAM Rectangular Constellation

The three convolutional codes are the following:

- *(2k, k) dual-k* that is appropriate to work with BPSK, QPSK, 16-QAM and 64-QAM, or in general, with all 2^{2k} - QAM, and has its $b(d)$ determined by the formula developed in the Appendix
- *(2, 1) IEEE 802.11a code* that can be applied to BPSK, QPSK, 16-QAM and 64-QAM, widely employed in wireless networks. The correspondent $b(d)$'s for each modulation scheme was determined by the numerical method discussed in Section II.D.2, and shown in Table 2.1
- *(4, 3) CHL code* from Ref. [7]. This code will be referenced from this point forward as *Chang-Hwang-Lin*, or CHL for short, in honor of the three discoverers. It is appropriate for 16-QAM only and counts on having, as an advantage, a relatively high code rate compared with the other two (2, 1) codes. By this latter characteristic, CHL does not compromise the maximum bit rate capacity available in the channel very much. Table 3.1 depicts the main characteristics of this code, where *G* is the octal format representation of the transfer-function matrix *G(D)* as defined in Chapter II.

K	d_{free}	$b(d)$				
		i = 0	i = 1	i = 2	i = 3	i = 4
4	4	91	3371	64050	1243923	22648788
$G = \begin{pmatrix} 07 & 10 & 11 & 14 \\ 15 & 04 & 06 & 10 \\ 00 & 11 & 02 & 16 \end{pmatrix}_8$						

Table 3.1. Main Characteristics of the (4, 3) Chang-Hwang-Lin Code Employed in this Thesis.

1. 16-QAM with Dual-4, IEEE 802.11a and CHL

Figure 3.10 shows the various performances for a 16-QAM with a dual-4 encoder. The performance of an uncoded QPSK channel was also plotted because it represents an alternative to sending the same 2 information bits contained in the 4-bit 16-QAM coded symbol during one transmission. Recall that $R_c = 1/2$ for dual-k codes. Keeping this alternative in mind, the first goal of a good convolutional code with $R_c = 1/2$ applied to a 16-QAM channel is to exceed the performance of the uncoded QPSK.

Looking at Figure 3.10, the dual-4 is able to exceed the uncoded QPSK during 94.4% to 97.5% of all transmissions for values of $?_b$ greater than 8 dB. Notice that the consideration of only the first upper bound corresponding to $P_{cov} = 100\%$ would at once discard the dual-4 code.

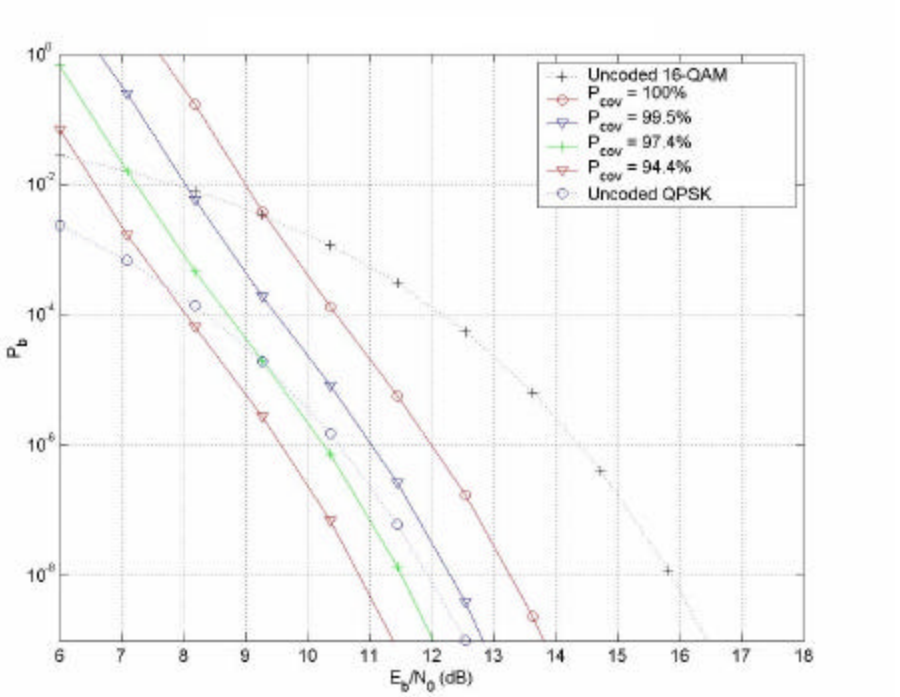


Figure 3.10. 16-QAM with (8, 4) Dual-4 and Viterbi Soft-Decision Decoding and Improvements in the Upper Bound.

Another interesting behavior to note is how fast the upper bound tightens itself at the first improvement steps. Although the amount of reduction on P_{cov} increases during the last improvements, the corresponding variation in the upper bound becomes smaller. This behavior can be explained by looking at the dpdf of the 16-QAM constellation for $d_{free} = 4$ in Figure 3.4. The first point corresponds to a small probability that makes P_{cov} vary just a few small steps. On the other hand, the subsequent improvements reflect at least in 3% of the changes in P_{cov}. The analysis of the dpdf determines what can be expected from the improvements.

Table 3.2 shows the correspondence between the values of $D^2_{\min}$ and P_{cov} considered for each improvement in P_b. These values can be achieved using the dpdf in Figure 3.4 and the method described in Section III.C.3. The first four were used in the plot in Figure 3.10.

	P_{cov} (%)				
	100	**99.5**	**97.4**	**94.4**	**91.4**
D_{AV}^2 / A^2	4	5	6	7	8

Table 3.2. Values of D_{AV}^2 / A^2 Correspondent to Different Probabilities of Coverage (P_{cov}) for a 16-QAM Rectangular Constellation Using Convolutional Code with d_{free} = 4.

Finally, notice that due to improvements, it is possible to tighten the original upper bound up to 2.5 dB and still have a reliable curve that covers 94.4 % of all data in the channel.

Figure 3.11 illustrates the behavior of the (2, 1) IEEE 802.11a code when applied to a 16-QAM channel. In general, the behavior is very close to the dual-4, being slightly worse.

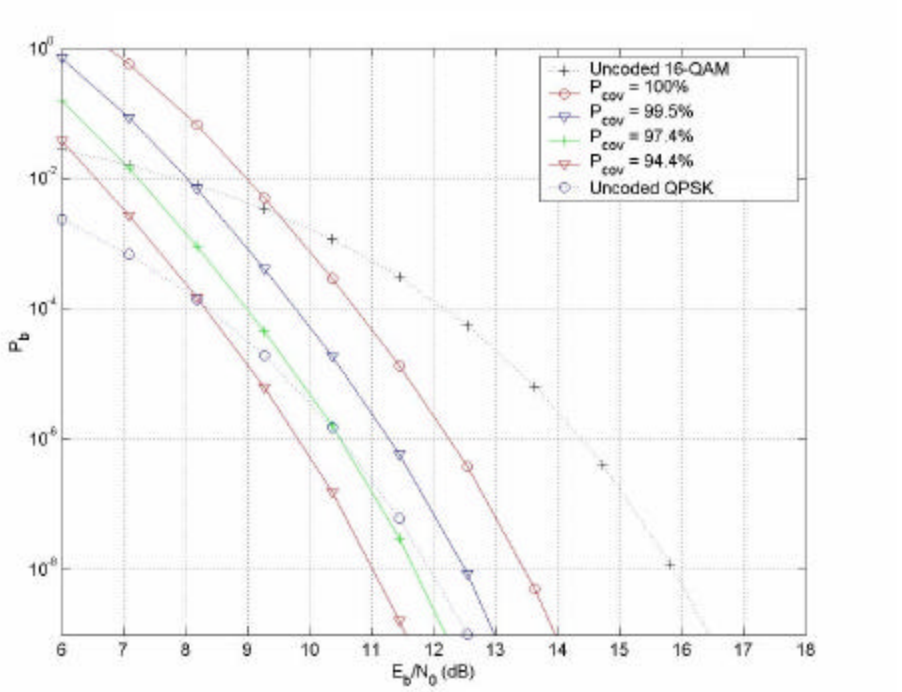


Figure 3.11. 16-QAM with (2, 1) IEEE 802.11a Convolutional Code, Viterbi Soft-Decision Decoding, and Improvements in the Upper Bound.

All the considerations made for the dual-4 are still valid except that only one bit is shifted per time into the (2, 1) IEEE 802.11a encoder contrasting with the four bits of the dual-4.

Figure 3.12 depicts the behavior of the (4, 3) CHL code working with the same 16-QAM channel. The performance now is almost 1.5 dB better than the other two cases. Considering that this code consumes only 25% of the channel bandwidth in contrast to the 50% achieved by the other two, the obtained result becomes much better. In fact, notice that the basis for the performance comparison in Figure 3.12 is the uncoded 8-QAM curve instead of the uncoded 4-QAM used on both (8, 4) dual-4 and (2, 1) IEEE 802.11a. The difference occurs because now there are three information bits per transmitted coded symbol instead of two, which could be done through an uncoded 8-QAM channel.

The relative step improvements in the upper bound when applying the (4, 3) CHL can be seen to be the same in Figure 3.12 as those obtained in Figures 3.10 and 3.11. As was demonstrated previously, the effect of the convolutional code on those improvements depends only on its d_{free}. Since, coincidently, all three codes exposed here present $d_{free} = 4$ when working with the 16-QAM channel, the improvement steps will be exactly the same for all of them.

Clearly, the best choice among all three codes to associate it with a 16-QAM modulator falls to the (4, 3) CHL. When trying to express this performance in one single metric, note in Figure 3.12 that only 8 dB are required for the SNR to achieve $P_b = 10^{-5}$ for 94.4% of the data. It represents a gain of 4.5 dB when compared with the uncoded 8-QAM at the expense of only 25% of bandwidth loss.

Another remarkable fact about (4, 3) CHL code is its $K = 4$, considered a relative small constraint length for good codes. Just for the sake of comparison, IEEE 802.11a code presents $K = 7$, although its shift registers size is equal to 1, which is smaller than 3 used by the (4, 3) CHL.

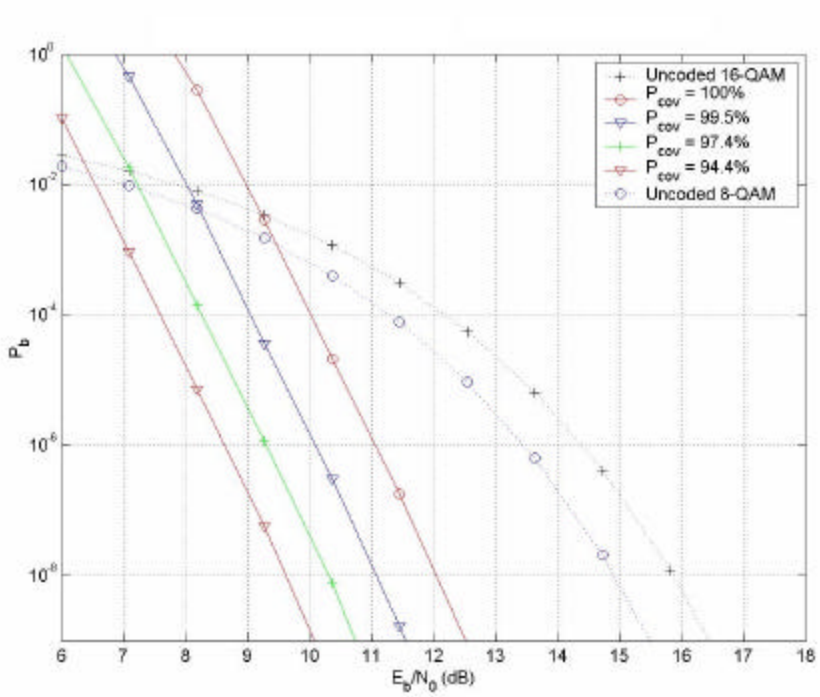


Figure 3.12. 16-QAM with (4, 3) Chang-Hwang-Lin Convolutional Code, Viterbi Soft-Decision Decoding, and Improvements in the Upper Bound.

2. 64-QAM with Dual-6 and IEEE 802.11a

Figure 3.13 illustrates some performances of a 64-QAM channel associated with a dual-6 encoder. In this case, since $R_c = 1/2$ and $q = 6$ bits, the alternative uncoded channel used for comparison is the 8-QAM.

Just looking at the performance of the upper bound that covers 100% of the data in the channel clearly indicates that the dual-6 code does not lead to significant enhancements over the uncoded 64-QAM channel. On the contrary, the small gain it provides is much less than the choice of an uncoded 8-QAM. However, considering the improved upper bounds, this apparently weak dual-6 code becomes a very interesting option. Notice especially the first improved upper bound that covers 99.6% of the data. It represents about 5 dB in the coding gain analysis along a range that practically goes through all possible sequences in the channel. In fact, considering only the curve that $P_{cov} = 99.6\%$, it overcomes most of the gain obtained if choosing the uncoded 8-QAM.

The characteristic of the large shift in the upper bound for 64-QAM when compared with 16-QAM is very comprehensible when analyzing the dpdf of D_{AV}^2 for both constellations shown in Figures 3.4 and 3.7, respectively. As explained in Section III.C.3, when commenting about Figure 3.7, the greater granularity presented by the dpdf of 64-QAM points out to a D_{AV}^2 for covering 99.6 % of all possible sequences much farther from $4A^2$ (D_{AV}^2 for 100%) than this same parameter for 16-QAM. In other words, the D_{AV}^2 that makes P_{cov} = 99.6% is much greater for the 64-QAM than for the 16-QAM constellation, even though both present $4A^2$ as their minimum possible value for D_{AV}^2. The farther the considered D_{AV}^2 is from the minimum, the larger the first improved upper bound will shift from the original one.

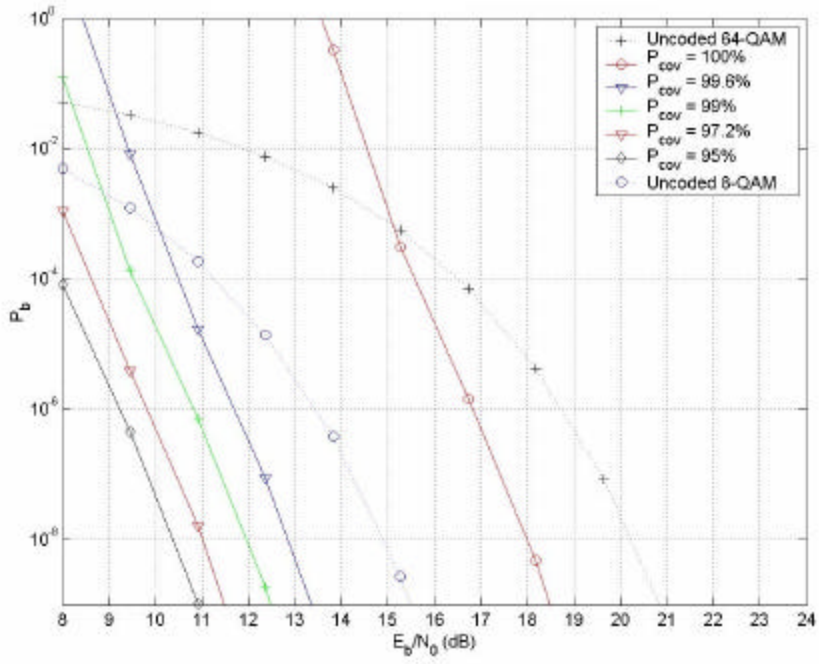


Figure 3.13. 64-QAM with (12, 6) Dual-6 Convolutional Code, Viterbi Soft-Decision Decoding, and Improvements in the Upper Bound.

Table 3.3 shows the correspondence between values of D_{AV}^2 chosen to be $D_{\min}^2$, and P_{cov} for each improvement on P_b.

	P_{cov} (%)				
	100	**99.6**	**99.0**	**97.2**	**95.0**
D_{AV}^2 / A^2	4	13	16	20	23

Table 3.3. Values of D_{AV}^2 / A^2 Correspondent to Different Probabilities of Coverage (P_{cov}) for a 64-QAM Rectangular Constellation Using Convolutional Code with d_{free} = 4.

Finally, it is remarkable how the dual-6 code demonstrates an excellent performance with 64-QAM after all improvements. Notice on the curve for P_{cov} = 95% that it requires only 8.5 dB to provide a bit error rate of 10^{-5}. This performance is comparable with that shown for 16-QAM with (4, 3) CHL code in the latter Section. Furthermore, this result is enhanced by the fact that the dual-6 could be categorized as a poor code to work with 64-QAM when analyzing only the most conservative upper bound, which would lead to a complete distortion of a realistic behavior.

Figure 3.14 shows the behavior of the (2, 1) IEEE 802.11a code when applied to a 64-QAM channel. The overall performance of this code can be seen to be approximately 2 dB worse than the dual-6 when considering the improved upper bounds.

The same comments related to the shifts on the improved upper bounds made for dual-6 is also applicable for the (2, 1) IEEE 802.11a. However, notice that d_{free} now is 3 instead of 4. Consequently, the dpdf for this case is the one shown in Figure 3.6, which presents less granularity than that in Figure 3.7 for d_{free} = 4. Less granularity means smaller distances from $4A^2$ for the D_{AV}^2 related with the first improved upper bound, which leads to a smaller shift. In fact, notice that the first shift in the upper bound for the (2, 1) IEEE 802.11a is 4 dB and reaches P_{cov} = 99.4%, while approximately this same parameter for the dual-6 code reaches 5 dB.

The small d_{free} presented by the (2, 1) IEEE 802.11a when associated with 64-QAM can be pointed out to be the cause of this code being outperformed by the dual-6. However, notice that for P_{cov} = 95% the (2, 1) IEEE 802.11a code does not disappoint,

achieving 8 dB of coding gain from the uncoded 16-QAM curve at $P_b = 10^{-5}$, and 4.5 dB when compared with the uncoded 8-QAM.

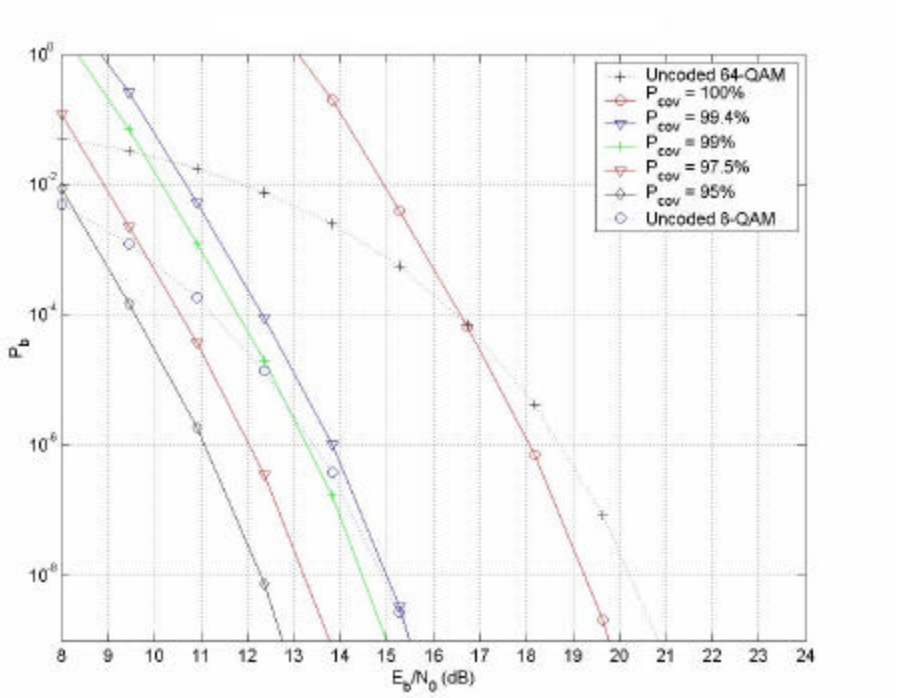


Figure 3.14. 64-QAM with (2, 1) IEEE 802.11a Convolutional Code, Viterbi Soft-Decision Decoding, and Improvements in the Upper Bound.

Figure 3.15 again demonstrates the performance of the two previously discussed codes but are now plotted together and only for the most conservative upper bound. Clearly, it can be seen that the dual-6 is better than the IEEE 802.11a. Moreover, a performance analysis based only on these plots would lead to the discarding of both codes in favor of the uncoded 8-QAM option, which was seen to be a wrong conclusion. This last fact highlights the great usefulness of the upper bound in improving methodology for large constellations. The larger the constellation and the d_{free}, the better will be the improved upper bounds.

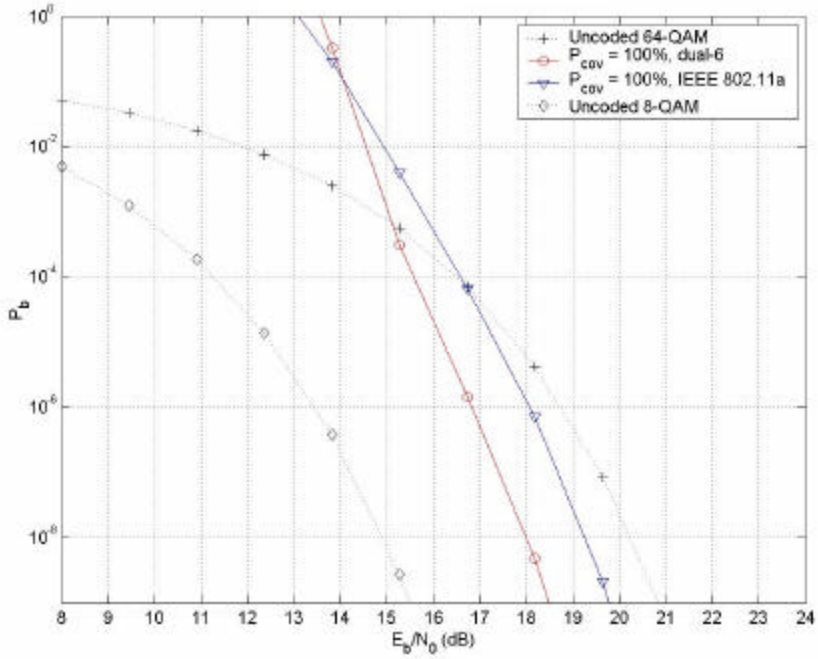


Figure 3.15. 64-QAM with (12, 6) Dual-6 and (2, 1) IEEE 802.11a Convolutional Codes for P_{cov} = 100% Using Viterbi Soft-Decision Decoding.

3. BPSK with Dual-1 and IEEE 802.11a

Figure 3.16 reports the performance of the (2, 1) dual-1 and (2, 1) IEEE 802.11a codes in a BPSK channel. In this case, improvements in the upper bound are no longer possible due to the nonexistence of a possible dpdf of D_{AV}^2 for BPSK. Recall that BPSK presents only two symbols and, regardless of the symbol chosen to be C_0, there is only one option left for a possible erroneous coded symbol, which leads to $D_{AV}^2 = D_{\min}^2 = 4A^2$, for every d_{free}.

Clearly, it can be seen that the IEEE 802.11a is much better than the dual-6 in providing almost twice the coding gain in dB. Notice that the relative performance here is the opposite of the 16-QAM and 64-QAM cases for the same analysis. Recall from Table 2.1 that d_{free} for the IEEE 802.11a decreases as M increases, obtaining the maximum value of 10 exactly for BPSK. Whereas, for the dual-k code, since the number of memory

elements in the shift-registers accompany the increasing of M, its d_{free} remains constant at 4. When d_{free} is higher for the IEEE 802.11a code than for the dual-k, the first code outperforms the second.

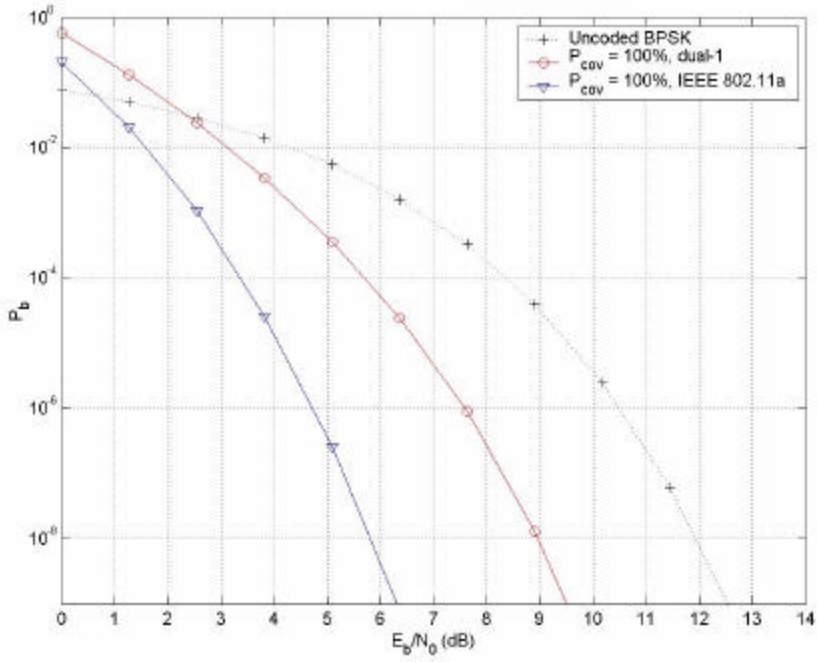


Figure 3.16. BPSK with (2, 1) Dual-1 and (2, 1) IEEE 802.11a Convolutional Codes Using Viterbi Soft-Decision Decoding.

4. QPSK with Dual-2 and IEEE 802.11a

The QPSK modulation is midway between BPSK and 16-QAM in terms of the possibility of improvements. Contrary to BPSK, it can present a dpdf of D_{AV}^2 but is not as prolific as a 16 or 64-QAM. The QPSK constellation is very symmetrical, which allows any choice of the C_0 symbol to have the same result. In other words, there is no conservative choice for C_0 in the QPSK rectangular constellation. Also, once C_0 is chosen, among the other three remaining symbols, only one of them presents an Euclidean distance different than the minimum, which allows the existence of a timid dpdf that can only obtain higher granularities if d becomes too large. Figure 3.17 depicts

the dpdf's for the values of d_{free} equals to 4 and 6, corresponding to the dual-2 and IEEE 802.11a codes, respectively.

The dpdf for d = 4 can be seen to have very small granularity presenting only five points. Therefore, the minimum value for D_{AV}^2 , $4A^2$, can be present in 20% of the possible sequences in the channel. This means that the minimum improvement achievable for the upper bound will embrace only 80% of the possible transmission, which can considerably damage theoretical performance forecasts. On the other hand, when d increases just a little bit and becomes six, the dpdf granularity increases to seven points and the minimum improvement can now cover 91.2% of the data. Notice in Figure 3.17 that there is 8.8% of probability of happening the minimum average distance for d = 6. Although, P_{cov} = 91.2% is much less than the minimum of 95% considered before for 16-QAM and 64-QAM, it is possible to imagine a design considering this value if it yields considerable improvement in the upper bound. The restrictions discussed in this paragraph clearly demonstrate how the efficiency of the upper bound improvement directly depends on the magnitude of M.

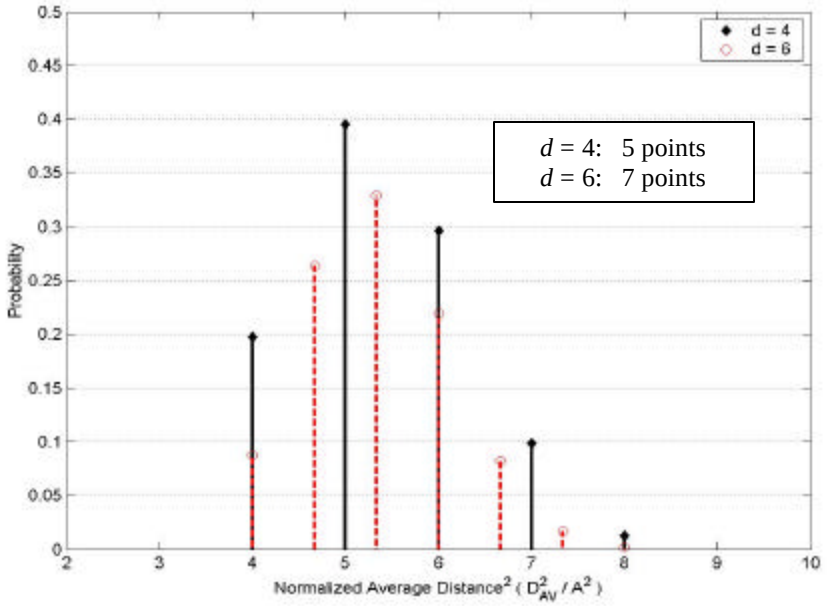


Figure 3.17. Dpdf of D_{AV}^2 for a QPSK Rectangular Constellation Considering Two Values of d, 4 and 6.

As a result of the previous issues about QPSK dpdf's, the performance analysis does not consider improvements for the upper bound of the dual-2 code because it presents d_{free} = 4. The only improvement to consider is for the IEEE 802.11a code, which shows d_{free} = 6 when working with QPSK, as can be seen in Table 2.1. Actually, this procedure is limited to a first step in which $D_{AV}^2 = 4.6667\,A^2$ that corresponds to P_{cov} = 91.2%. Figure 3.18 illustrates the QPSK performance analysis for the two considered codes. Since R_c = 1/2 for both codes, the alternative uncoded channel that QPSK can compare with is the BPSK, which can be seen to perform the same as the uncoded QPSK. In fact, QPSK works like BPSK operating in two completely independent orthogonal channels, which explains the identity of these metrics. This means, in practice, that there is no alternative uncoded channel to use instead of coded QPSK. Consequently, any possible coding gain will be very welcome. The reader can see how to this point the QPSK case differs from the 16-QAM and 64-QAM, presenting at the same time, characteristics of a binary and a non-binary modulation technique.

Looking at Figure 3.18 for P_{cov} = 100%, the IEEE 802.11a code can be seen to provide more coding gain than the dual-2 code, even though the difference between these gains is now smaller than for the BPSK case. Notice that the d_{free} for the IEEE 802.11a code drops from ten to six when the modulation changes from BPSK to QPSK. However, six is still greater than four, d_{free} for the dual-2 code, pushing IEEE 802.11a to overcome the performance of dual-2 for QPSK.

Concerning the improvement in the upper bound for the IEEE 802.11a code in Figure 3.18, the resultant shift can be seen to be very small, almost 0.7 dB. This is justified because D_{AV}^2 (= $4.6667A^2$, for this case) is too close to the minimum distance $4A^2$. The consequent loss in P_{cov} does not compensate for this small improvement and the best decision should be to disregard this curve. On the other hand, for the performance of IEEE 802.11a with P_{cov} = 100%, the coding gain at $P_b = 10^{-5}$ can be seen to be 4 dB, which is less than the 5.5 dB obtained for BPSK, but still a good coding gain. Notice that the dual-2 code provides only 2.5 dB for this gain.

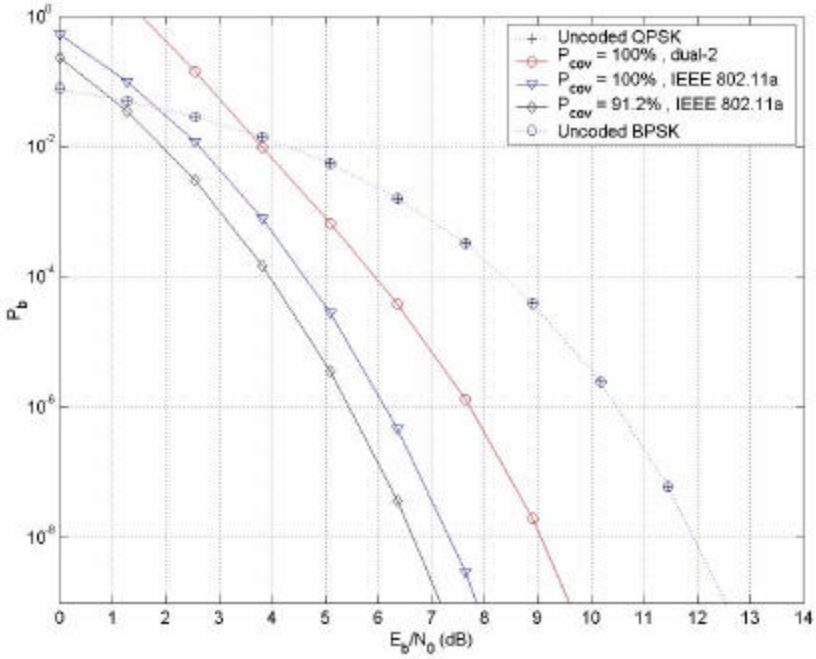


Figure 3.18. QPSK with (4, 2) Dual-2 and (2, 1) IEEE 802.11a Convolutional Codes Using Viterbi Soft-Decision Decoding.

E. SUMMARY

Chapter III dealt only with the pure AWGN scenario, deriving the upper bound of $P_2(d)$ for any M-ary coherent demodulator followed by a SDD. This upper bound was customized for M-QAM and generated a model for determining P_b for any M-QAM modulation with SDD. The chapter also demonstrated how to take advantage of the large number of symbols existing in M-QAM and improve P_b in an attempt to tighten it to more realistic data in the channel. As a practical application, the performance analyses of 16-QAM, 64-QAM, BPSK and QPSK, associated with up to three different convolutional codes were shown. Chapter IV will illustrate how to adapt all results obtained in this chapter to derive P_b for Nakagami fading channels.

IV. PERFORMANCE ANALYSIS IN A NAKAGAMI-M CHANNEL

All issues discussed in Chapter III only account for one type of random variable in the receiving process, which is related to the AWGN. Although this approach is suitable for many types of communication links, such as free space and wired communications, unfortunately it does not completely satisfy the requirements of some of the most common applications. Whenever obstacles are present between the transmitter and the receiver that can create scattering and multipath, the communication engineer must consider the effect of more random variables in the channel model. This is the case with wireless networks and wireless telephony, in which walls, buildings, cars, trees and all sort of obstacles can be between the two communicating points. Other examples are the radio ionospheric communications in the HF band and ionospheric forward scatter in VHF. In these cases, the ionosphere is responsible for reflecting and scattering the signal, and allowing it to reach the receiver when there is no line-of-sight (LOS). This chapter makes a parallel distinction between the pure AWGN scenario studied in Chapter III and the scenario in which all the effects of midway elements can be modeled by a Nakagami-m fading channel.

A. NAKAGAMI-M CHANNEL

Many elements can affect signal propagation, such as multipath, Doppler effects, shadowing by surrounding objects and changes in environmental characteristics. The final result is a channel that has a randomly time variant impulse response. In other words, if the same impulse is transmitted successive times, the correspondent received signals will be different each time. The differences can be in the number, in the relative delays and in the shape of the received signals related to each transmitted impulse. Such channels are named *time-variant channels* and can be categorized in terms of the frequency selectivity and fading characteristics they introduce in the signal.

The best model for a certain communication link depends on its bandwidth and the environment in which the communication end-points will operate. A large number of channels present the characteristic of non-distorting signal harmonics, i.e., the frequency

response of the channel is *flat* for the entire bandwidth. Also, the fading conditions in those channels vary so slowly during a symbol interval (T_S) that the resultant attenuation can be considered *constant* along each T_S. Even being constant during a certain T_S, notice that the fading conditions can vary from symbol to symbol. Those channels form a category named *frequency-nonselective slowly fading channels* (or *flat-slow-fading channels*) and result in a multiplicative distortion of the transmitted signal by a random variable attenuation factor a as in Ref. [3].

The Nakagami-m channel to be considered in this thesis is a frequency non-selective slowly fading channel in which a is characterized statistically by a Nakagami-m distribution and has $\mathrm{E}\left[a^2\right] = 1$. It is the best statistical model for this category of channels and is applicable in many communication systems that work under fading effects, such as wireless networks and wireless telephony. A Nakagami-m channel is the best fit for data signals received in urban radio multipath channels [3], but it also embraces other models in special cases such as Rayleigh, utilized when there is no LOS, and Ricean distributed, appropriate in the presence of LOS. Notice that the channel noise remains the same AWGN as previously seen. Therefore, in the subsequent analyses, two types of random variables will be considered, a, that follows the Nakagami-m distribution, and n_{xi}, for $i = 1$ and 2, that follows the Gaussian distribution and represents the two components of the AWGN that are present in the two orthogonal channels of the coherent M-QAM demodulator.

Applying the Nakagami-m distribution [3] to a leads to

$$f_a(a) = \frac{2}{\Gamma(m)}\left(\frac{m}{\mathrm{E}\left[a^2\right]}\right)^m a^{(2m-1)} e^{\frac{-ma^2}{\mathrm{E}\left[a^2\right]}}, \tag{4.1}$$

where $\Gamma(m)$ is the Gamma function, which is defined as

$$\Gamma(m) = \int_0^\infty t^{(m-1)} e^{-t}\, dt, \qquad m > 0.$$

The parameter *m* is called *fading figure* and can be seen as the ratio

$$m = \frac{\mathrm{E}\left[a^2\right]^2}{\mathrm{E}\left[\left(a^2 - \mathrm{E}\left[a^2\right]\right)\right]}, \qquad m \geq \frac{1}{2}. \tag{4.2}$$

Notice that the denominator of *m* is the variance of the second moment of a.

Recall that

$$\mathrm{E}\left[a^2\right] = 1. \tag{4.3}$$

Substituting Equation (4.3) in Equations (4.1) and (4.2) yields

$$f_a(a) = \frac{2}{\Gamma(m)}(m)^m a^{(2m-1)} e^{-ma^2}, \tag{4.4}$$

and

$$m = \frac{1}{\mathrm{E}\left[\left(a^2 - 1\right)\right]}, \qquad m \geq \frac{1}{2}. \tag{4.5}$$

Notice in Equation (4.4) that when $m = 1$, $f_a(a)$ reduces to $2a\,e^{-a^2}$, which is a *Rayleigh* distribution. Also, when a^2 remains constant and equal to 1 for every received symbol, no variance exists in the attenuation factor, which causes Equation (4.5) to obtain $m = \infty$. This is what characterizes a *no fading* scenario in which only AWGN affects the receiving process. All the plots obtained in Chapter III are included in this case.

As stated previously, the attenuation factor a imposes a multiplicative distortion in the transmitted signal in the cases of frequency-nonselective slowly fading channels. Thus, the new received signal can be represented as

$$r(t) = a\,s(t) + n(t), \tag{4.6}$$

where *s(t)* is the transmitted signal component, and *n(t)* characterizes the AWGN.

By Equation (4.6), the average signal energy contained in one received symbol in the presence of attenuation factor a can be expressed as

$$e_{Sa} = a^2 e_S, \tag{4.7}$$

where e_S is the average signal energy contained in one received symbol if no attenuation exists in the channel.

Since the noise remains the same, Equation (4.7) leads to

$$\gamma_{Sa} = a^2 \gamma_S , \tag{4.8}$$

where γ_{Sa} is the average SNR in a symbol for a given a and γ_S is the average SNR in a symbol when there is no attenuation.

Also, the overall average SNR in a symbol can be expressed as

$$\bar{\gamma}_{Sa} = \mathrm{E}\left[a^2\right]\gamma_S = \gamma_S . \tag{4.9}$$

Substituting Equation (4.9) in (4.8), results in

$$\gamma_{Sa} = a^2 \bar{\gamma}_{Sa} . \tag{4.10}$$

Since $a \geq 0$, Equation (4.10) leads to a one-to-one mapping between a and γ_{Sa}. Hence, the pdf of γ_{Sa} can be straightforwardly obtained by

$$f_{\gamma_{Sa}}(\gamma_{Sa}) = \frac{1}{\left|d\gamma_{Sa}/da\right|} f_a(a)\Big|_{a=f^{-1}(\gamma_{Sa})} . \tag{4.11}$$

Working with Equation (4.10), the following relationships stand

$$a = \sqrt{\frac{\gamma_{Sa}}{\bar{\gamma}_{Sa}}} , \tag{4.12}$$

and

$$\left|\frac{d\gamma_{Sa}}{da}\right| = 2a\bar{\gamma}_{Sa} . \tag{4.13}$$

Applying Equations (4.4), (4.12) and (4.13) into (4.11) leads to

$$f_{\gamma_{Sa}}(\gamma_{Sa}) = \frac{m^m}{\Gamma(m)\left(\bar{\gamma}_{Sa}\right)^m} \gamma_{Sa}^{(m-1)} e^{\frac{-m\gamma_{Sa}}{\bar{\gamma}_{Sa}}} . \tag{4.14}$$

The interpretation of the latter equation can be as follows. If a sequence of a same symbol is sent in a Nakagami-m channel with fading figure *m*, the set of all SNR of each of the equivalent received symbols, γ_{Sa}, would constitute a random variable that varies accordingly to the pdf provided by Equation (4.14). Notice that the reason in the previous description for a same symbol being sent is only to simplify the understanding. Recall that in M-QAM modulation, different symbols can have different amplitudes leading to different SNR. This explains the application of the term "average" for the definition of γ_{Sa} in Equation (4.8). Also note that even though γ_{Sa} is a random variable, $\bar{\gamma}_{Sa}$ in Equation (4.14) is a deterministic value equivalent to the SNR that those symbols would obtain in the absence of attenuation in the channel.

B. PROBABILITY OF BIT ERROR FOR UNCODED M-QAM IN NAKAGAMI-M CHANNEL

The performance analyses presented in this thesis always compare the probability of bit error P_b of a M-QAM modulation that makes use of a certain convolutional FEC versus the option of sending uncoded symbols through the same channel. Thus, in order to study the M-QAM performances in a Nakagami-m channel, it is necessary to obtain the corresponding expressions relating P_b with the average SNR per information bit, called $\bar{\gamma}_b$. Recall that in the case of Nakagami-m channels, since each symbol can be received with different attenuations, the SNR per information bit, γ_b, is a random variable that can be expressed as a function of γ_{Sa}. Therefore, in order to compare performances in Nakagami-m channels with those previously obtained, the average $\bar{\gamma}_b$ must be used since it is a deterministic value and represents the same γ_b employed for the pure AWGN case. The difference in the notations will be kept to emphasize the averaging reference.

In a pure AWGN scenario, the following expressions stand for determining the upper bound of P_b of an optimum receiver for a M-QAM uncoded signal [3]:

$$P_b = \frac{1}{q} P_M \,, \tag{4.15}$$

and

$$P_M = N_M \, Q\left(\sqrt{\frac{D_{\min}^2}{2N_0}} \right), \tag{4.16}$$

where q is the number of bits per symbol, P_M is the probability of symbol error, N_M is the largest number of neighboring points that are at distance $D_{\min}^2$ from any constellation points, and $D_{\min}^2$ is the minimum Euclidean distance between signal points in the constellation.

Equation (4.16) can be rewritten as

$$P_M = N_M \, Q\left(\sqrt{\frac{D_{\min}^2}{2e_S} \frac{e_S}{N_0}} \right) = N_M \, Q\left(\sqrt{\frac{D_{\min}^2}{2e_S} \gamma_S} \right), \tag{4.17}$$

where e_S is the average energy of one received coded symbol.

Consider a Nakagami-m channel that presented a certain attenuation factor α applied to a received symbol and correctly compensated for by the detector, i.e., a specific given α. Due to the energy issues discussed in Equation (4.7), P_M can now be computed for this specific symbol as a function of α by means of

$$P_M(\alpha) = N_M \, Q\left(\sqrt{\frac{D_{\min}^2}{2e_S} \alpha^2 \gamma_S} \right). \tag{4.18}$$

By substituting Equation (4.8) in (4.18), P_M can be expressed as a function of $\gamma_{S\alpha}$ as

$$P_M(\gamma_{S\alpha}) = N_M \, Q\left(\sqrt{\frac{D_{\min}^2}{2e_S} \gamma_{S\alpha}} \right). \tag{4.19}$$

Notice that Equation (4.19) is valid only if g_{Sa} is known, i.e., this is a conditional P_M. In order to determine the average upper bound for P_M, an integration with the pdf of the random variable g_{Sa} is necessary and is given as

$$P_M = \int_{-\infty}^{+\infty} P_M(g_{Sa})\, f_{g_{Sa}}(g_{Sa})\, dg_{Sa} \,. \tag{4.20}$$

Substituting Equations (4.19) and (4.14) in (4.20) leads to

$$P_M = N_M \int_0^{+\infty} Q\left(\sqrt{\frac{D_{\min}^2}{2e_S} g_{Sa}}\right) \frac{m^m}{\Gamma(m)\left(\bar{g}_{Sa}\right)^m} g_{Sa}^{(m-1)}\, e^{\frac{-m g_{Sa}}{\bar{g}_{Sa}}}\, dg_{Sa} \,. \tag{4.21}$$

The analytical solution for this integration is not easily obtained. However, following the solution given in [8] for a similar integral when analyzing the special case of BPSK, after a development involving Gauss' hypergeometric function and Pochhammer's Symbol, the following result can be achieved,

$$P_M = N_M \sqrt{\frac{1}{1+\left(\frac{D_{\min}^2 \bar{g}_{Sa}}{4e_S m}\right)^{-1}}} \frac{\Gamma\left(m+\frac{1}{2}\right)}{2\sqrt{p}\,\Gamma(m+1)\left(1+\frac{D_{\min}^2 \bar{g}_{Sa}}{4e_S m}\right)^m} \cdot \left[1+\sum_{k=1}^{\infty} \frac{\prod_{n=0}^{k-1}\left(m+\frac{1}{2}+n\right)\left(\frac{1}{1+\frac{D_{\min}^2 \bar{g}_{Sa}}{4e_S m}}\right)^k}{\prod_{n=0}^{k-1}(m+1+n)}\right]. \tag{4.22}$$

From the previous discussion about $\bar{g}_b$ at the beginning of this Section, notice that

$$\bar{g}_{Sa} = q\bar{g}_b \tag{4.23}$$

Substituting Equations (4.22) and (4.23) in (4.15), the final equation for P_b is obtained as

$$P_b = \frac{N_M}{q}\sqrt{\frac{1}{1+\left(\frac{D_{\min}^2\, q\bar{g}_b}{4e_S m}\right)^{-1}}}\;\frac{\Gamma\left(m+\frac{1}{2}\right)}{2\sqrt{p}\;\Gamma(m+1)\left(1+\frac{D_{\min}^2\, q\bar{g}_b}{4e_S m}\right)^{m}} \cdot \left[1+\sum_{k=1}^{\infty}\frac{\prod_{n=0}^{k-1}\left(m+\frac{1}{2}+n\right)\left(\frac{1}{1+\frac{D_{\min}^2\, q\bar{g}_b}{4e_S m}}\right)^{k}}{\prod_{n=0}^{k-1}(m+1+n)}\right]. \tag{4.24}$$

Notice that Equation (4.24) is generic and can be applied to every M-QAM constellation.

In Section III.C.2 it was demonstrated that each M-QAM constellation has a different ratio $D_{\min}^2 / e_S$ that depends on *M*. Also, for rectangular constellations in which $M = 2^q$ and *q* is even, there is a special method in [3] for precisely determining N_M that leads to

$$N_M = \frac{4\left(\sqrt{M}-1\right)}{\sqrt{M}}. \tag{4.25}$$

For odd *q*, N_M approaches the average number of neighboring points that are at a distance $D_{\min}^2$. Table 4.1 summarizes the ratios $D_{\min}^2 / e_S$ as well as the corresponding N_M's for each M-QAM rectangular constellation considered in this thesis. The only modulator that presents odd *q* in Table 4.1 is the 8-QAM, which its constellation is depicted in Figure 3.9.

Modulation	M	q	$D_{\min}^2 / e_S$	N_M
BPSK	2	1	4	1
QPSK	4	2	2	2
8-QAM	8	3	2/3	2
16-QAM	16	4	2/5	3
64-QAM	64	6	2/21	3.5

Table 4.1. Useful Coefficients for Rectangular M-QAM Constellations.

For the special cases of rectangular constellations, the values in Table 4.1 should be used into Equation (4.24) to obtain the various performances. For BPSK and QPSK, since $N_M / q = D_{\min}^2 \, q / (4e_S) = 1$ for both, Equation (4.24) provides exactly the same expressions for these two modulations, which matches the equivalent behavior in the AWGN only scenario.

Figures 4.1, 4.2 and 4.3 show the P_b for the uncoded BPSK/QPSK, 16-QAM and 64-QAM, respectively. When $m = 1$, as discussed previously, the Rayleigh distribution for the attenuation is observed. Recall that small values of *m* means high variances of a^2, which signifies more severe conditions of fading. Note the presence of a tremendous drop in the performance for small *m*'s (0.5, 1 and 1.5) in all three plots. Moreover, in this high fading situation, there is almost no difference in performances among BPSK/QPSK, 16-QAM and 64-QAM. They become more evident for larger *m*'s.

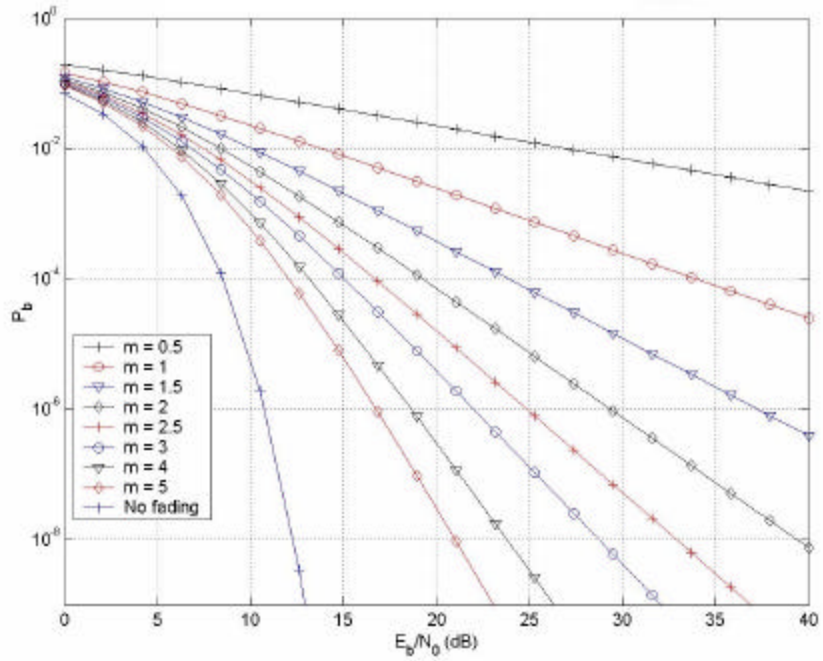


Figure 4.1. Uncoded BPSK/QPSK in Nakagami-m Channel.

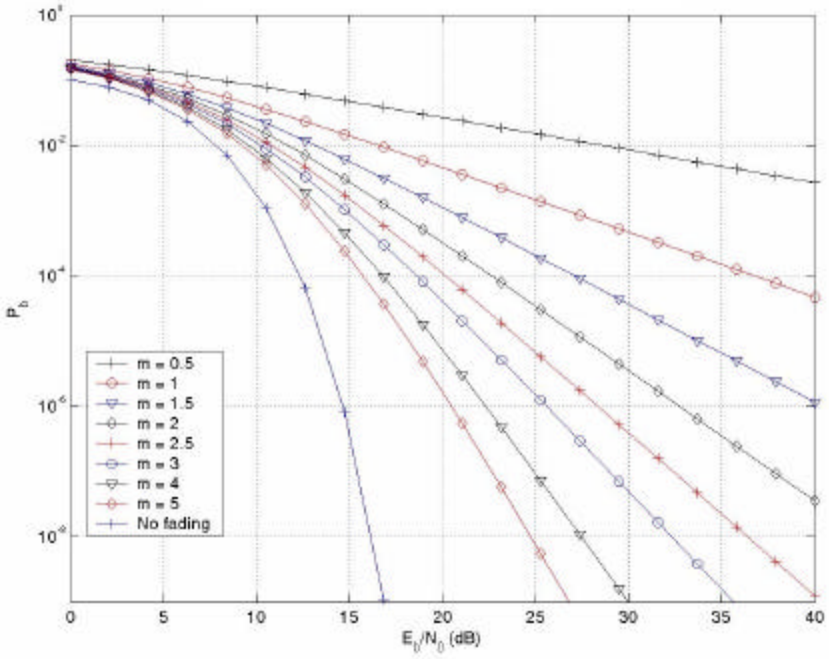


Figure 4.2. Uncoded 16-QAM in Nakagami-m Channel.

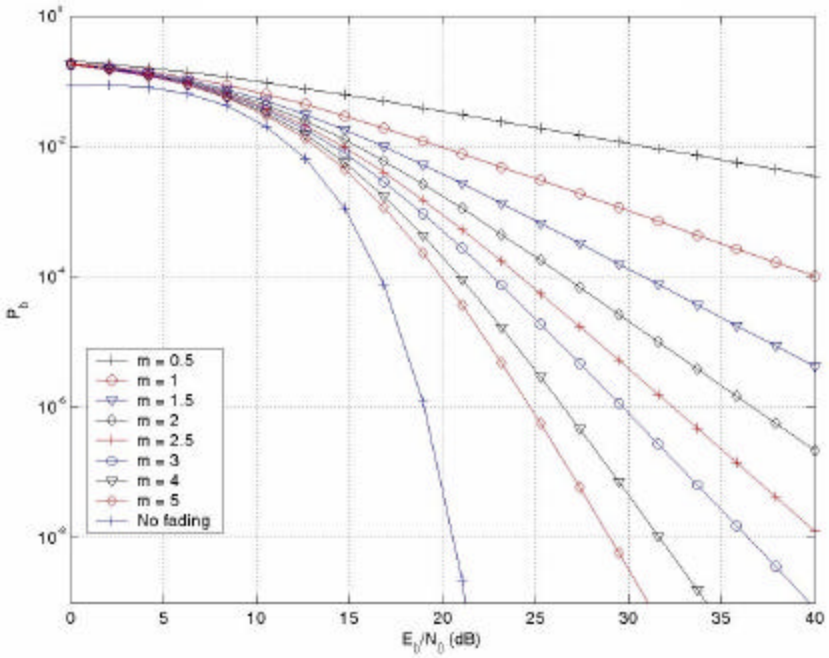


Figure 4.3. Uncoded 64-QAM in Nakagami-m Channel.

C. PROBABILITY OF BIT ERROR FOR M-QAM WITH SDD IN NAKAGAMI-M CHANNEL

The principles of determining the upper bound of the probability of bit error P_b for M-QAM when coding is being used in a Nakagami-m channel are exactly the same as previously seen for AWGN, indicating that the general equation shown below remains valid,

$$P_b \le \frac{1}{k} \sum_{d=d_{free}}^{\infty} b(d) P_2(d). \tag{4.26}$$

The parameter $b(d)$, which corresponds to an exclusive characteristic of the code being utilized, is the same seen in Chapter II since the code itself did not change. The only parameter that varies is $P_2(d)$. As seen in the latter Section, P_b of an uncoded symbol increases tremendously for severe fading conditions ($m < 1.5$). Therefore, the occurrence of a larger number of wrong paths should be naturally expected through the decoding process.

In this development, the same thread of thoughts followed in Section III.C.1 will be adapted to the Nakagami conditions in order to obtain the expressions for $P_2(d)$. The final P_b will be obtained by substituting this new $P_2(d)$ and $b(d)$, already computed in Chapter II, in Equation (4.26).

1. Upper Bound of $P_2(d)$ for any M-ary Coherent Demodulator and SDD in Nakagami-m Channel

Although the channel is now a frequency nonselective and slowly fading Nakagami channel, the receiver is still the same and is shown in Figure 3.2. Consider the reception of a sequence of coded symbols that presented various attenuation factors a's correctly compensated for by the detector, i.e., consider a given sequence of a's. Each a is related to only one received symbol in the sequence and is considered constant during the entire symbol period T_S. Therefore, the notation a_{jm} will refer to a particular a that affected the reception of the m^{th} symbol of the j^{th} branch in the trellis.

The expression for the received signal $r(t)$ changes to the one expressed in Equation (4.6), which can be written in a vector form as

$$R_{jm} = a_{jm} C_{jm} + N_{jm} . \tag{4.27}$$

The branch metric $m_j^{(i)}$, associated with the j^{th} branch of the i^{th} path in the trellis, also changes slightly. Since the decoder should compensate for the various a_{jm} in the signal during the decoding process, this new metric becomes

$$m_j^{(i)} = \log f_R(\{R_{j1}\ R_{j2}\ R_{j3} \ldots R_{jN}\} | \{a_{j1}C_{j1}^{(i)}\ \ a_{j2}C_{j2}^{(i)}\ \ a_{j3}C_{j3}^{(i)}\ \ \ldots\ \ a_{jN}C_{jN}^{(i)}\})$$

$$= \log f_R(\{R_{jm},\ \ m=1, 2, \ldots N\} | \{a_{jm}C_{jm}^{(i)},\ \ m=1, 2, \ldots N\})$$

$$m_j^{(i)} = \log f_R(\{R_{jm}\} | \{a_{jm}C_{jm}^{(i)}\}) . \tag{4.28}$$

Following the same development as in Section III.C.1, Equation (4.28) becomes

$$m_j^{(i)} = K_1 + K_2 \sum_{m=1}^{N} \| R_{jm} - a_{jm} C_{jm}^{(i)} \|^2, \tag{4.29}$$

where

$$K_1 = N \log\left(\frac{1}{2p s^2} \right), \quad \text{and} \qquad K_2 = -\frac{1}{2 s^2} \log e \ .$$

Considering the correlation metric adopted by the SDD, the probability of choosing a wrong path in the trellis in a pairwise comparison of the all-zero path (correct path) with a wrong path 1, which presents distance d from the all-zero path, given a certain set of a_{jm}, is represented by $P_2(d)$ and can now be written as

$$P_2\left(d, \{a_{jm}\}\right) = \Pr\left[CM^{(1)} \geq CM^{(0)} \right]$$

$$P_2\left(d, \{a_{jm}\}\right) = \Pr\left[\sum_{j=1}^{B} \sum_{m=1}^{N} \left(\| R_{jm} - a_{jm} C_{jm}^{(1)} \|^2 - \| R_{jm} - a_{jm} C_{jm}^{(0)} \|^2 \right) \leq 0 \right]. \tag{4.30}$$

Considering again that a sequence of all-zero symbols is the one being transmitted, hence

$$C_{jm}^{(0)} = C_0, \ \ \forall j \ \text{and} \ m, \tag{4.31}$$

and Equation (4.27) becomes

$$R_{jm} = a_{jm} C_0 + N_{jm} \, . \tag{4.32}$$

Substituting Equations (4.31) and (4.32) in (4.30) yields

$$P_2\left(d, \{a_{jm}\}\right) = \Pr\left[\sum_{j=1}^{B} \sum_{m=1}^{N} \left(\| a_{jm} C_0 + N_{jm} - a_{jm} C_{jm}^{(1)} \|^2 - \| a_{jm} C_0 + N_{jm} - a_{jm} C_0 \|^2 \right) \leq 0 \right]$$

$$P_2\left(d, \{a_{jm}\}\right) = \Pr\left[\sum_{j=1}^{B} \sum_{m=1}^{N} \left(\| a_{jm} \left(C_{jm}^{(1)} - C_0 \right) - N_{jm} \|^2 - \| N_{jm} \|^2 \right) \leq 0 \right]$$

$$P_2\left(d, \{a_{jm}\}\right) = \Pr\left[\sum_{j=1}^{B} \sum_{m=1}^{N} \left(a_{jm}^2 \| C_{jm}^{(1)} - C_0 \|^2 - 2 a_{jm} (C_{jm}^{(1)} - C_0) . N_{jm} \right) \leq 0 \right]. \tag{4.33}$$

Recalling that $C_{jm}^{(1)} = C_{jm}^{(0)} = C_0$, for all combinations of j and m except for d symbols, Equation (4.33) can be rewritten as follows

$$P_2\left(d,\{a_l\}\right) = \Pr\left[\sum_{l=1}^{d}\left(a_l^2 \,\|\, C_l^{(1)} - C_0 \,\|^2 - 2a_l(C_l^{(1)} - C_0)N_l\right) \le 0\right]. \tag{4.34}$$

Expanding the noise vector in its orthogonal components in Equation (4.34) results in

$$P_2\left(d,\{a_l\}\right) = \Pr\left[\sum_{l=1}^{d}\left(a_l^2 \,\|\, C_l^{(1)} - C_0 \,\|^2 - 2a_l(c_{l1}^{(1)} - c_{01})n_{l1} - 2a_l(c_{l2}^{(1)} - c_{02})n_{l2}\right) \le 0\right]. \tag{4.35}$$

Notice that the set of all a_l, for $l = 1$ to d, is considered "given" by the first assumption of this development. Furthermore, the sequence of $C_l^{(1)}$, for $l = 1$ to d, that compounds path 1, is also known by the decoder. Therefore, due to n_{l1} and n_{l2} being independent-identical-distributed zero-mean Gaussian r.v.'s and the other terms being deterministic values, the left side of the inequality in Equation (4.35) can be seen to be a Gaussian random variable X expressed as

$$X \;=\; N\left(\sum_{l=1}^{d}\overline{X}_l,\; \sum_{l=1}^{d}s_{Xl}^{\,2}\right), \tag{4.36}$$

where

$$\overline{X}_l = a_l^2 \,\|\, C_l^{(1)} - C_0 \,\|^2, \tag{4.37}$$

and

$$s_{Xl}^{\,2} = 4a_l^{\,2}\left((c_{l1}^{(1)} - c_{01})^2 + (c_{l2}^{(1)} - c_{02})^2\right)s^2 = 4a_l^{\,2}\,\|\, C_l^{(1)} - C_0 \,\|^2\, s^2$$

$$= 4\overline{X}_l s^2. \tag{4.38}$$

Here an upper bound can be established by assuming that all $C_l^{(1)}$'s, for every l, are equal to a worst case symbol, C_W, in such a way that $P_2\left(d\right)$ becomes the highest possible.

If C_W is a constant vector for every l, then Equation (4.36) becomes

$$X = N\left(\| C_W - C_0 \|^2 \sum_{l=1}^{d} a_l^2, \quad 4\| C_W - C_0 \|^2 \, s^2 \sum_{l=1}^{d} a_l^2 \right). \tag{4.39}$$

In order to maximize the probability in Equation (4.35) and obtain an upper bound for $P_2(d)$, C_W must be the closest symbol to C_0. The same theorem and arguments from Section III.C.1 can be used to demonstrate this fact. This situation leads to

$$\| C_W - C_0 \|^2 = D_{0,\min}^2 . \tag{4.40}$$

Substituting Equation (3.26) in (3.24) and the result in Equation (4.35) yields

$$P_2\left(d, \{a_l\}\right) = \Pr[X \le 0] = \Pr\left[N\left(D_{0,\min}^2 \sum_{l=1}^{d} a_l^2, \quad 4D_{0,\min}^2 s^2 \sum_{l=1}^{d} a_l^2 \right) \le 0 \right]$$

$$= \quad 1 - Q\left(\frac{0 - D_{0,\min}^2 \sum_{l=1}^{d} a_l^2}{\sqrt{4D_{0,\min}^2 s^2 \sum_{l=1}^{d} a_l^2}} \right)$$

$$= Q\left(\sqrt{\frac{D_{0,\min}^4 \left(\sum_{l=1}^{d} a_l^2 \right)^2}{4D_{0,\min}^2 s^2 \sum_{l=1}^{d} a_l^2}} \right) = Q\left(\sqrt{\frac{D_{0,\min}^2 \sum_{l=1}^{d} a_l^2}{4s^2}} \right),$$

which implies

$$P_2\left(d, \{a_l\}\right) = Q\left(\sqrt{\frac{D_{0,\min}^2 \sum_{l=1}^{d} a_l^2}{2N_0}} \right), \tag{4.41}$$

where the last step stands due to $s^2 = N_0 / 2$.

Notice that $D_{0,min}$ can vary, depending on the C_0 chosen to be the zero-symbol in the constellation. Therefore, in order to guarantee that Equation (3.31) will be the real upper bound for a given set of a_l, C_0 should be chosen to be the one that yields

$D_{0,\min} = D_{\min}$, where D_{min} is the *minimum Euclidean distance* in the constellation. Equation (3.31) can be rewritten as

$$P_2\left(d,\{a_l\}\right) = Q\left(\sqrt{\frac{D_{\min}^2 \sum_{l=1}^{d} a_l^2}{2N_0}}\right). \tag{4.42}$$

All the definitions made in this chapter related to symbol energy are still valid for coded symbol energy. However, the letter "C" will be added in the notations to emphasize the coded symbol scenario. The following expressions stands

$$e_{CSa_l} = a_l^2 e_{CS}, \tag{4.43}$$

$$g_{CSa_l} = a_l^2 g_{CS}, \tag{4.44}$$

$$\overline{g}_{CSa_l} = \mathrm{E}\left[a_l^2\right] g_{CS} = g_{CS}, \tag{4.45}$$

and

$$\overline{g}_{CSa_l} = qR_C\overline{g}_b \tag{4.46}$$

Equation (3.32) can be rearranged as

$$P_2\left(d,\{a_l\}\right) = Q\left(\sqrt{\frac{D_{\min}^2}{2e_{CS}}\frac{e_{CS}}{N_0}\sum_{l=1}^{d} a_l^2}\right) = Q\left(\sqrt{\frac{D_{\min}^2}{2e_{CS}} g_{CS}\sum_{l=1}^{d} a_l^2}\right)$$

$$P_2\left(d,\{a_l\}\right) = Q\left(\sqrt{\frac{D_{\min}^2}{2e_{CS}}\sum_{l=1}^{d} a_l^2 g_{CS}}\right). \tag{4.47}$$

Replacing Equation (4.44) in (4.47) yields

$$P_2\left(d,\left\{g_{CSa_l}\right\}\right) = Q\left(\sqrt{\frac{D_{\min}^2}{2e_{CS}}\sum_{l=1}^{d} g_{CSa_l}}\right). \tag{4.48}$$

Assume a random variable *z* defined as

$$z = \sum_{l=1}^{d} g_{CSa_l}. \tag{4.49}$$

Substituting Equation (4.49) in (4.48) leads to

$$P_2(d,z) = Q\left(\sqrt{\frac{D_{\min}^2}{2e_{CS}}z}\right). \tag{4.50}$$

In order to determine the average upper bound for $P_2(d)$, an integration with the pdf of the random variable z is necessary and is given as

$$P_2(d) = \int_{-\infty}^{+\infty} P_2(d,\, z)\, f_Z(z)\, dz\,. \tag{4.51}$$

Notice that *z* is the sum of *d* independent-identical-distributed random variables g_{CSa_l}, which have pdf already derived in Section IV.A and equal to

$$f_{g_{CSa_l}}\left(g_{CSa_l}\right) = \frac{m^m}{\Gamma(m)\left(\overline{g}_{CSa_l}\right)^m}\, g_{CSa_l}{}^{(m-1)}\, e^{\frac{-m g_{CSa_l}}{\overline{g}_{CSa_l}}}\,. \tag{4.52}$$

Following the solution in Ref. [8] for a similar development involving BPSK and SDD, by working with Laplace transforms, it can be demonstrated that the pdf of *Z* will be

$$f_Z(z) = \frac{m^{md}}{\Gamma(md)\left(\overline{g}_{CSa_l}\right)^{md}}\, z^{(md-1)}\, e^{\frac{-mz}{\overline{g}_{CSa_l}}}\,. \tag{4.53}$$

Applying Equation (4.23) in (4.53) yields

$$f_Z(z) = \frac{m^{md}}{\Gamma(md)\left(q\,R_C\,\overline{g}_b\right)^{md}}\, z^{(md-1)}\, e^{\frac{-mz}{q\,R_C\,\overline{g}_b}}\,. \tag{4.54}$$

Substituting Equations (4.50) and (4.54) in (4.20) leads to

$$P_2(d) = \int_{-\infty}^{+\infty} Q\left(\sqrt{\frac{D_{\min}^2}{2e_{CS}}z}\right)\frac{m^{md}}{\Gamma(md)\left(q\,R_C\,\overline{g}_b\right)^{md}}\, z^{(md-1)}\, e^{\frac{-mz}{q\,R_C\,\overline{g}_b}}\, dz\,. \tag{4.55}$$

Note how Equation (4.55) resembles Equation (4.21). There is no difference with respect to the integration variable. Consequently, the analytical solution can be obtained by following the steps in [8], as for Equation (4.21). The final result can be expressed as

$$P_2(d) = \sqrt{\frac{1}{1+\left(\frac{D_{\min}^2\, q\, R_C\, \bar{g}_b}{4 e_{CS}\, m}\right)^{-1}}}\ \frac{\Gamma\left(md+\frac{1}{2}\right)}{2\sqrt{p}\,\Gamma(md+1)\left(1+\frac{D_{\min}^2\, q\, R_C\, \bar{g}_b}{4 e_{CS}\, m}\right)^{md}}$$

$$\cdot\left[1+\sum_{k=1}^{\infty}\frac{\prod_{n=0}^{k-1}\left(md+\frac{1}{2}+n\right)\left(\frac{1}{1+\frac{D_{\min}^2\, q\, R_C\, \bar{g}_b}{4 e_{CS}\, m}}\right)^{k}}{\prod_{n=0}^{k-1}(md+1+n)}\right]. \qquad (4.56)$$

Due to the orthogonalities existent in the demodulating process, all the assumptions thus far in this Section for a bidimensional coherent demodulator are also valid for multi-dimensional M-ary modulations such as M-FSK. The unidimensional types, such as BPSK, can be viewed as special cases that are also embraced by the previous development. Consequently, Equation (4.56) can be considered a general formula for the upper bound of $P_2(d)$ that can be applied to every *M-ary coherent demodulator* followed by a *soft-decision Viterbi decoder* in a *Nakagami-m channel*.

2. Upper Bound of $P_2(d)$ for M-QAM and SDD in Nakagami-m Channel

For the special cases of M-QAM rectangular constellations, $P_2(d)$ can be obtained by simply substituting the correspondent values of Table 4.1 in Equation (4.56). Notice that "$D_{\min}^2 / e_S$" in Table 4.1 can be used for $D_{\min}^2 / e_{CS}$ in Equation (4.56) since the coded symbol is the one that really modulates the carrier, performing the role of one regular symbol in uncoded systems.

After consulting Table 4.1, the expression $D_{\min}^2 q/(4e_{CS})$ leads to the following values to be directly applied into Equation (4.56) for BPSK, QPSK, 8-QAM, 16-QAM, and 64-QAM, respectively: 1, 1, 1/2, 2/5 and 1/7. Notice that BPSK and QPSK provide the same result for $P_2(d)$ again.

3. Improvements on the Upper Bound of $P_2(d)$ in Nakagami-m Channel

All considerations regarding improvements on the upper bound of $P_2(d)$ made in Section III.C.3 are still valid for a Nakagami-m channel. The previous study was based on the nature of the constellation, the probability of transmitting some specific code sequences, and d_{free}. There is no relation with the type of channel being used. Therefore, instead of using $D_{\min}^2/e_{CS}$ in Equation (4.56), D_{AV}^2/e_{CS} should be employed. The parameter D_{AV}^2 represents the square of the new average distance to be used as $D_{\min}^2$.

A practical procedure for applying a certain improvement represented by D_{AV}^2/A^2 can be derived as follows:

$$\frac{D_{AV}^2}{e_{CS}} = \frac{D_{AV}^2 D_{\min}^2}{e_{CS} D_{\min}^2} = \frac{D_{\min}^2 D_{AV}^2}{e_{CS} 4A^2} = \frac{D_{\min}^2}{e_{CS}} \frac{D_{AV}^2/A^2}{4}. \qquad (4.57)$$

Equation (4.57) shows that multiplying the parameter $D_{\min}^2/e_{CS}$ from Table 4.1 by the desired D_{AV}^2/A^2 and dividing the result by 4, the parameter D_{AV}^2/e_{CS} for substituting $D_{\min}^2/e_{CS}$ in Equation (4.56) is obtained.

The same values of D_{AV}^2/A^2 correspond to the same and already derived values of P_{cov}.

D. PERFORMANCE ANALYSIS

This Section revisits all performance analyses made in Chapter III by replacing the former pure AWGN scenario by a Nakagami fading channel. For this purpose, it

utilizes all results obtained in the two latter Sections concerning the estimation of P_b for uncoded and coded M-QAM modulations as a function of the average signal-to-noise ratio per bit, $\bar{g}_b$. The same three convolutional codes and four M-QAM modulators are presented. Each plot of P_b versus $\bar{g}_b$ is associated with a certain configuration and is shown for four values of m: 0.5, 1, 2, and 3. Recall that m = 0.5 represents the minimum value that m can assume and indicates a very severe fading condition, while m = 1 means a Rayleigh channel with no LOS, and m = 2 resembles a Ricean channel in which a LOS exists. Finally, m = 3 demonstrates a more favorable fading condition and can be considered a relative high value of m.

1. 16-QAM with Dual-4, IEEE 802.11a and CHL

Figures 4.4, 4.5, 4.6 and 4.7 show the performance of the 16-QAM in Nakagami fading channels for m = 3, 2, 1 and 0.5, respectively. In these figures, it is remarkable how large the coding gain becomes as the fading conditions worsen, reaching a value of more than 20 dB for m = 1 at $P_b = 10^{-5}$. It is interesting to notice how close to each other the uncoded curves become as m decreases. When m = 0.5, the uncoded 16-QAM, 8-QAM and QPSK curves unite themselves into a single curve. These three uncoded curves provide a means for comparing the performances of sending coded versus uncoded data through a channel that is operating at the bandwidth limit. Recall that the uncoded curve must be the QPSK when comparing the use of IEEE 802.11a and dual-4 codes due to R_c = 1/2 in both, while it changes to uncoded 8-QAM when considering the use of (4, 3) CHL. Although attractive in the pure AWGN scenario, the option of a smaller uncoded modulation becomes worthless in the Nakagami fading channel. Even considering a large m environment, as shown in Figure 4.4, notice that the uncoded QPSK curve does not interlace the coded curves as it did in the AWGN.

Notice that the legends in Figures 4.4, 4.5, 4.6 and 4.7 display a percentage at the end of the CHL labels. This number indicates that an improvement in the upper bound was used for the correspondent curve. It represents the percentage of the total possible

symbol combinations that are covered by this improved upper bound, P_{cov}. Note, however, that if this percentage equals 100%, no improvements were applied. In order to avoid overloading the graphs with many curves that would lead to the same conclusions, the improvement technique was applied only for the best of the three codes, CHL, and limited to two steps.

An interesting point to highlight is the maintenance of a constant step-gain for the improvements as the fading condition changes. As can be seen in Figures 4.4, 4.5, 4.6 and 4.7, the average gains were about 1 dB and 2.5 dB for the two improvements and they remained constant in all four fading scenarios.

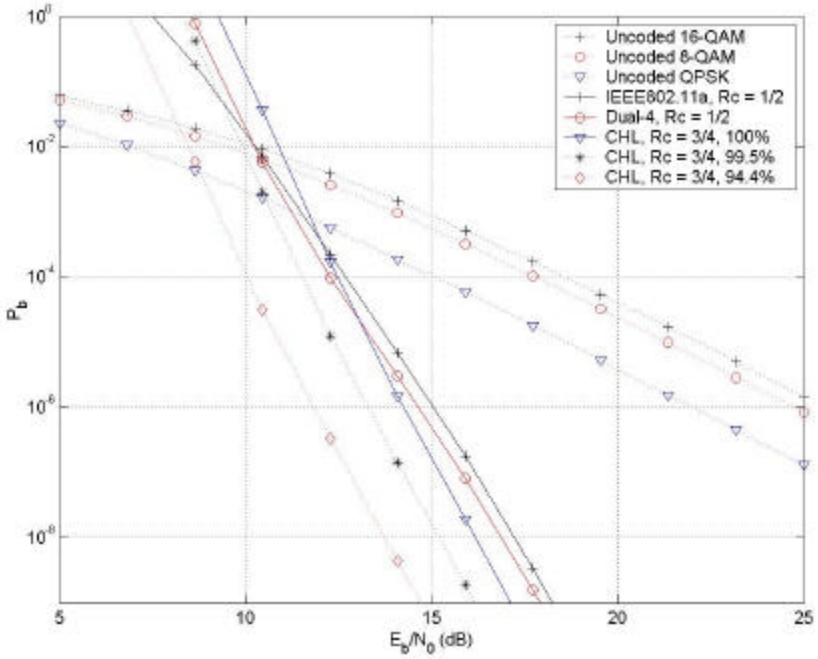


Figure 4.4. Probability of Bit Error for 16-QAM with SDD in Nakagami Fading (m = 3) Channel Using IEEE 802.11a, Dual-4 and CHL Convolutional Codes.

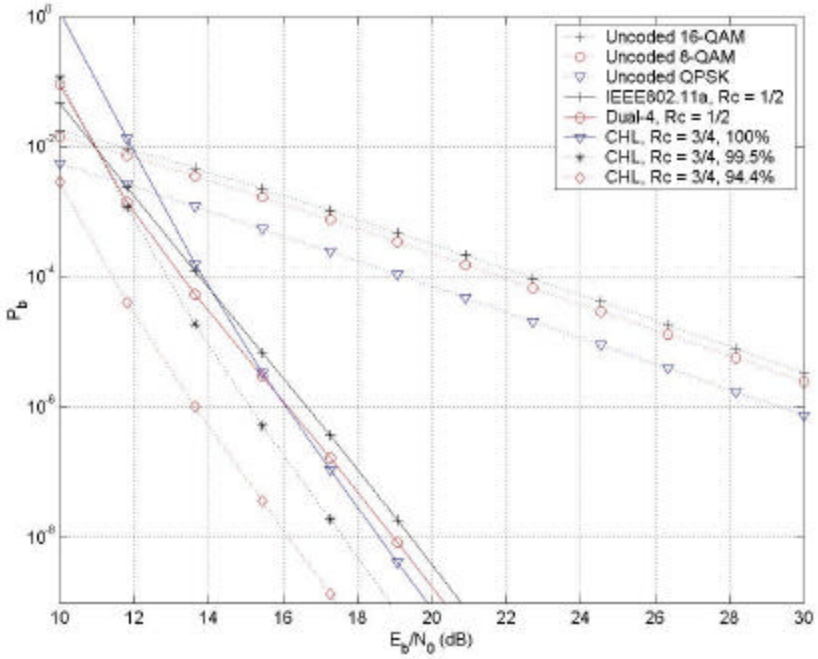


Figure 4.5. Probability of Bit Error for 16-QAM with SDD in Nakagami Fading (*m* = 2) Channel Using IEEE 802.11a, Dual-4 and CHL Convolutional Codes.

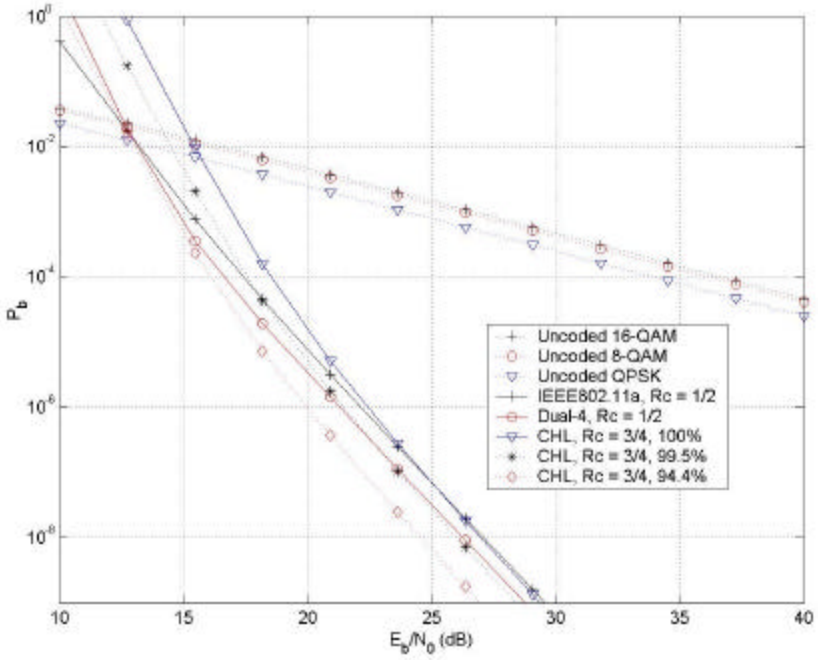


Figure 4.6. Probability of Bit Error for 16-QAM with SDD in Nakagami Fading (*m* = 1) Channel Using IEEE 802.11a, Dual-4 and CHL Convolutional Codes.

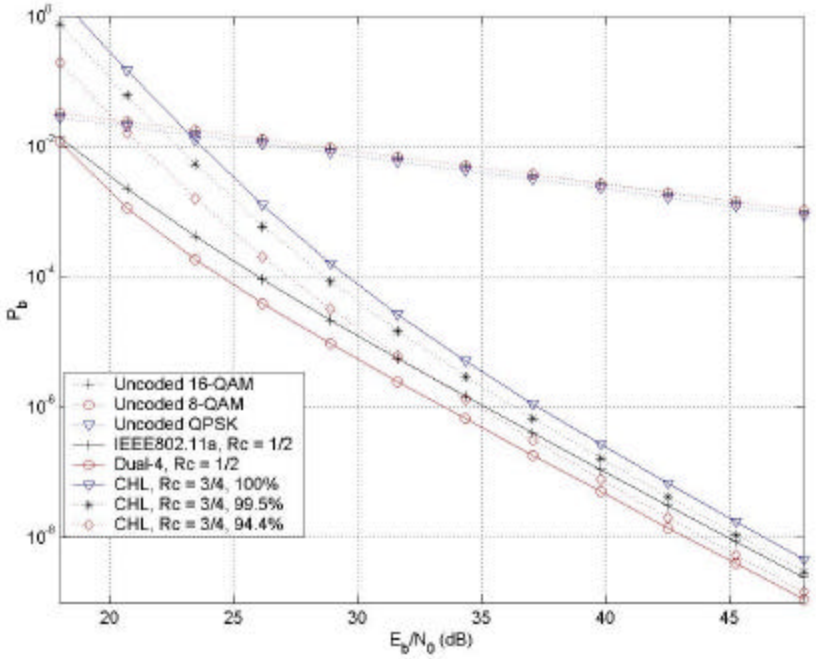


Figure 4.7. Probability of Bit Error for 16-QAM with SDD in Nakagami Fading (m = 0.5) Channel Using IEEE 802.11a, Dual-4 and CHL Convolutional Codes.

In terms of relative performance among codes, the dual-4 always slightly exceeds the IEEE 802.11a for all m's. On the other hand, when comparing CHL with dual-4 and IEEE 802.11a, it demonstrates that it is only better for m = 3. It is practically equal for m = 2 and loses approximately 2 dB when m is less or equal to 1. Therefore, the CHL seems to be more sensitive to fading than the other two. The reader should recall, however, that the CHL code presents the great advantage of compromising only 25% of the bit rate, whereas the use of either IEEE 802.11a or dual-k imposes 50% to this loss. Considering the pros and cons, the CHL code could also be considered the best choice among all three for 16-QAM in a Nakagami fading channel.

2. 64-QAM with Dual-6 and IEEE 802.11a

Figures 4.8, 4.9, 4.10 and 4.11 illustrate the performance of the 64-QAM associated with both dual-6 and IEEE 802.11a codes operating in four Nakagami fading

channels with m = 3, 2, 1 and 0.5, respectively. Notice again how large the coding gain becomes as the fading conditions worsen even using modulations with great *M*. Figures 4.10 and 4.11 show coding gains greater than 20 dB at $P_b = 10^{-5}$ for both codes considered.

The dual-6 code also proved to be better than IEEE 802.11a in Nakagami channels, ensuring its superiority already presented in the pure AWGN scenario. For this reason, the dual-6 code was plotted together with two improvements in the upper bound, which provided a 5 and 7.5 dB of gain for the channel analysis. Once more, these gains kept constant for the different *m*'s, as can be seen in each one of the corresponding figures. Notice also that the step gains obtained for Nakagami channels due to improvements are the same achieved for AWGN, which can confirm the efficiency of this technique even for severe fading environments operating with large *M* modulation schemes.

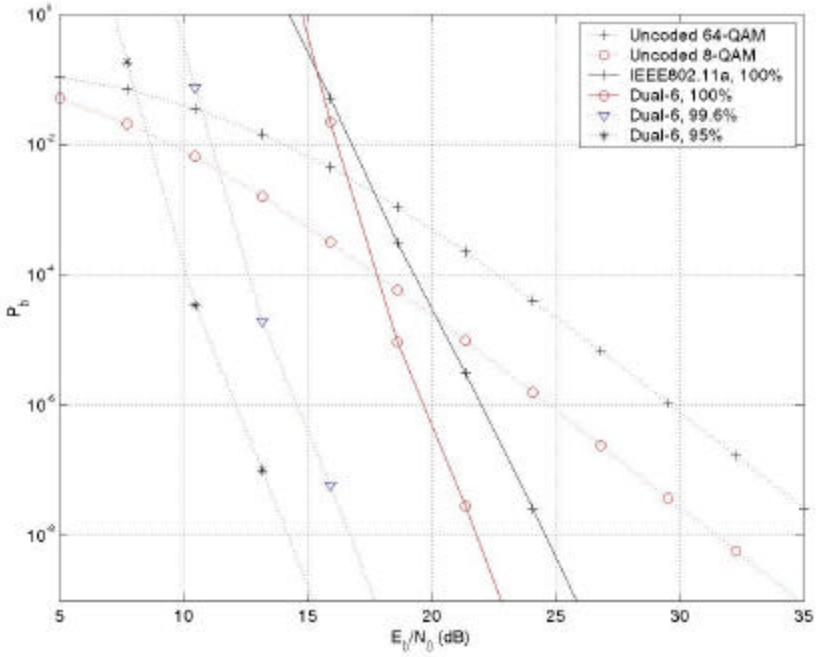


Figure 4.8. Probability of Bit Error for 64-QAM with SDD in Nakagami Fading (*m* = 3) Channel Using IEEE 802.11a and Dual-6 Convolutional Codes.

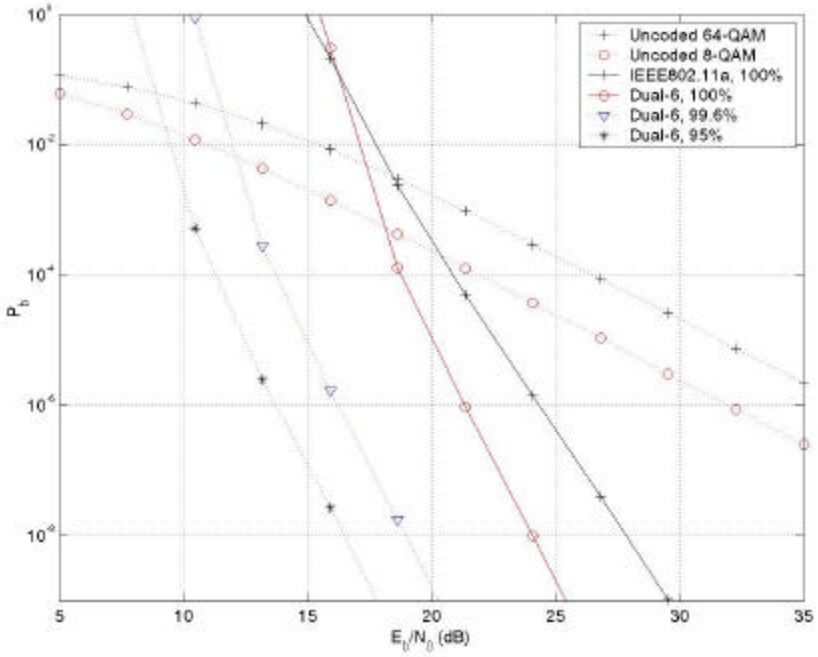


Figure 4.9. Probability of Bit Error for 64-QAM with SDD in Nakagami Fading (m = 2) Channel Using IEEE 802.11a and Dual-6 Convolutional Codes.

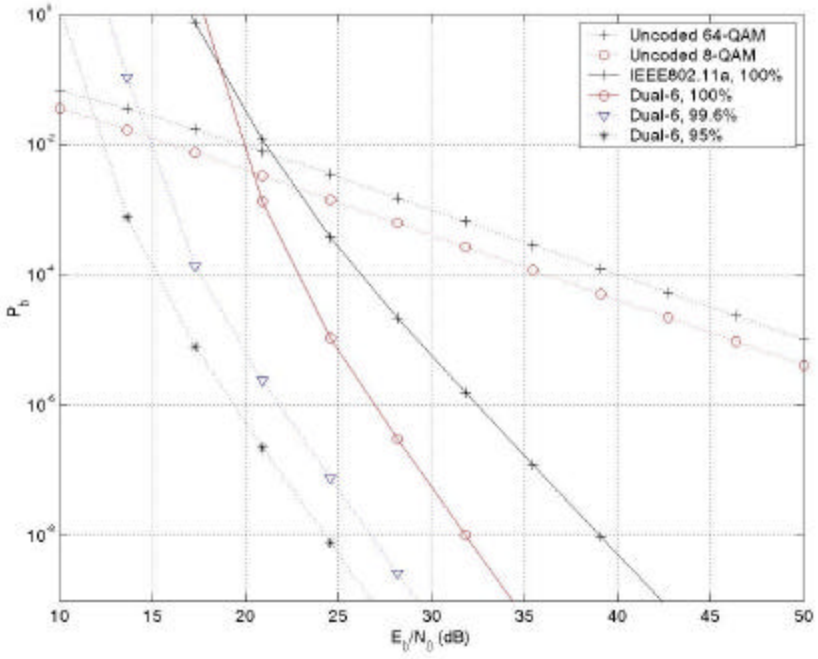


Figure 4.10. Probability of Bit Error for 64-QAM with SDD in Nakagami Fading (m = 1) Channel Using IEEE 802.11a and Dual-6 Convolutional Codes.

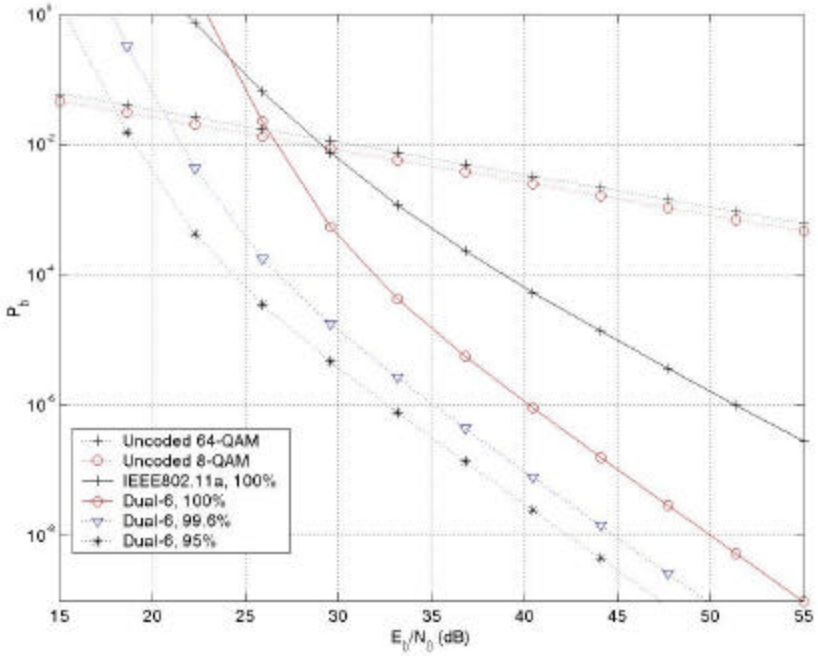


Figure 4.11. Probability of Bit Error for 64-QAM with SDD in Nakagami Fading (m = 0.5) Channel Using IEEE 802.11a and Dual-6 Convolutional Codes.

3. BPSK and QPSK with Dual-*k* and IEEE 802.11a

Figures 4.12, 4.13, 4.14 and 4.15 display the performance of the BPSK and QPSK when associated with both dual-*k* and IEEE 802.11a codes operating in four Nakagami fading channels with m = 3, 2, 1 and 0.5 respectively. Recall that the uncoded BPSK and QPSK, as demonstrated in Section IV.B, present the same performance. The differences in P_b come from the coding process that will provide different $b(d)$'s for each of these modulations.

The IEEE 802.11a code proved to be better than dual-*k* for both BPSK and QPSK also in Nakagami channels, ensuring its superiority already established in the pure AWGN scenario. No improvements are possible to BPSK, as already seen in Chapter III, and one improvement is shown in all four figures for QPSK operating with IEEE 802.11a code. The same small improvement gain, approximately 0.7 dB, achieved in pure AWGN was again confirmed in the Nakagami channel actuating with QPSK.

Notice how great the IEEE 802.11a code performs with BPSK under severe fading conditions. In Figure 4.14, for $m = 1$, BPSK performance is even better than the one obtained for uncoded BPSK in pure AWGN. Also in Figure 4.14, IEEE 802.11a with QPSK and no improvement achieves almost the same performance as in pure AWGN for $P_b = 10^{-5}$. In other words, the IEEE 802.11a code is capable of eliminating a severe fading effect from the channel in both BPSK and QPSK at the expense of losing half bandwidth.

Another interesting behavior to highlight is the close performance maintained between BPSK and QPSK through all different fading conditions when operating with dual-k code. As the reader recalls, the dual-k code keeps $d_{free} = 4$ for every M, which greatly influences the close behavior previously cited. On the other hand, the IEEE 802.11a code varies its d_{free} according to Table 2.1. Since its d_{free} becomes smaller as M increases, the performance also decreases.

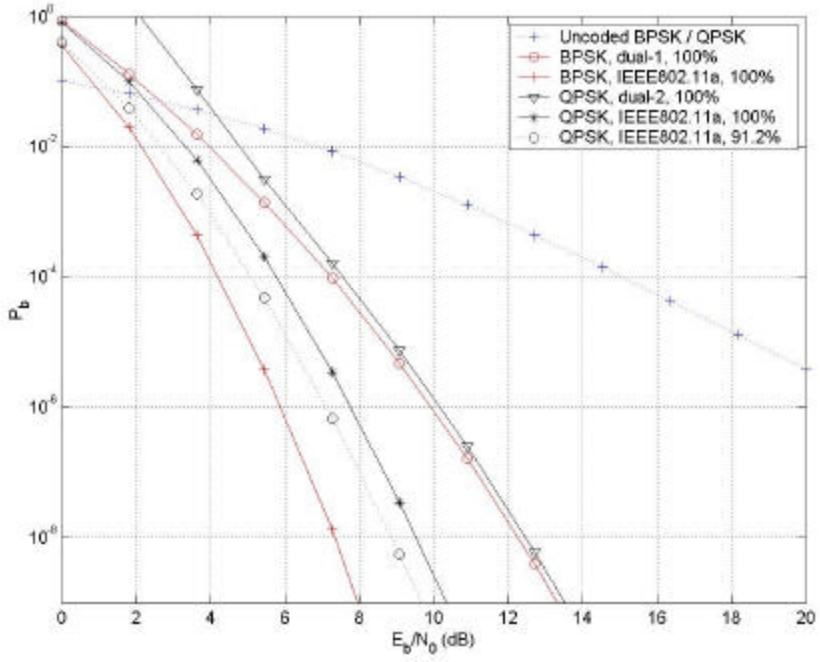


Figure 4.12. Probability of Bit Error for BPSK and QPSK with SDD in Nakagami Fading ($m = 3$) Channel Using IEEE 802.11a and Dual-k Convolutional Codes.

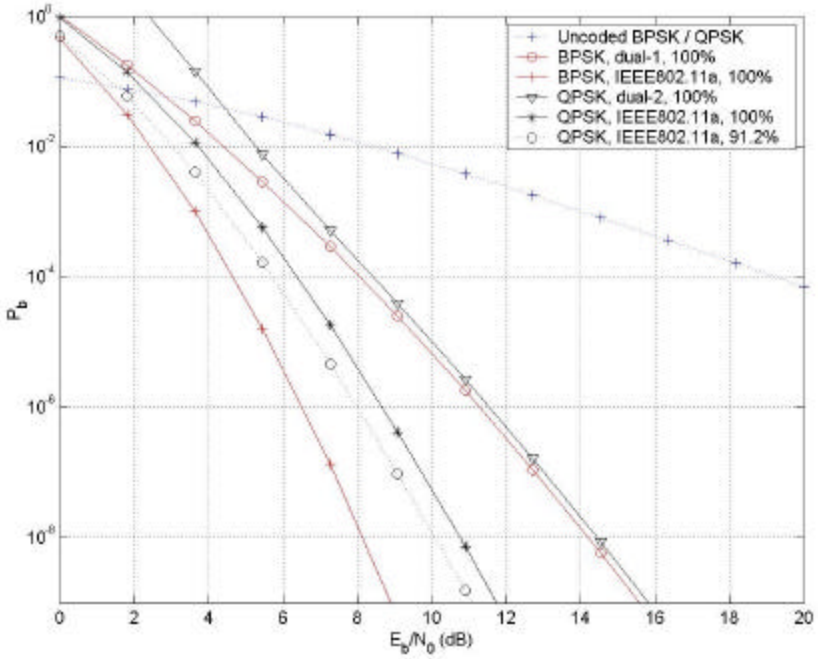


Figure 4.13. Probability of Bit Error for BPSK and QPSK with SDD in Nakagami Fading (m = 2) Channel Using IEEE 802.11a and Dual-k Convolutional Codes.

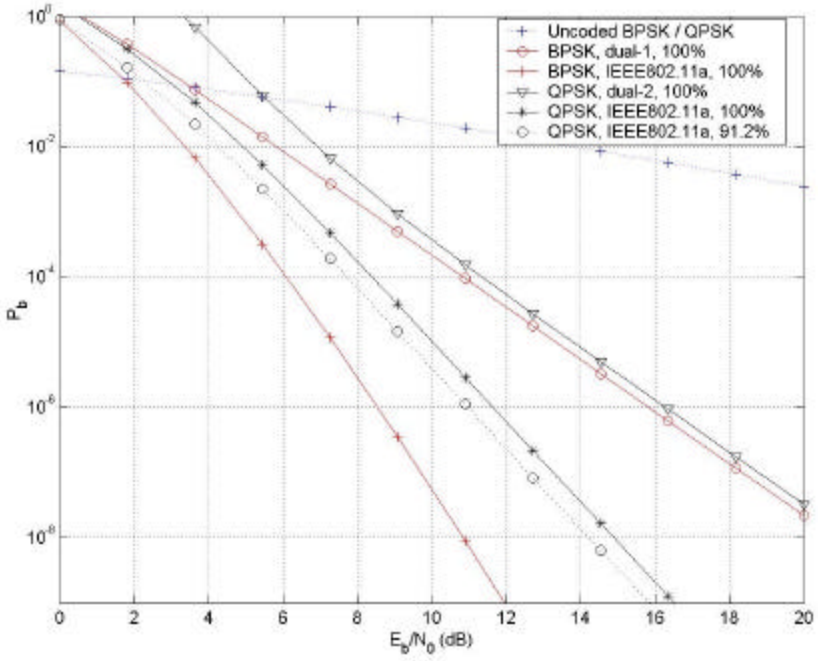


Figure 4.14. Probability of Bit Error for BPSK and QPSK with SDD in Nakagami Fading (m = 1) Channel Using IEEE 802.11a and Dual-k Convolutional Codes.

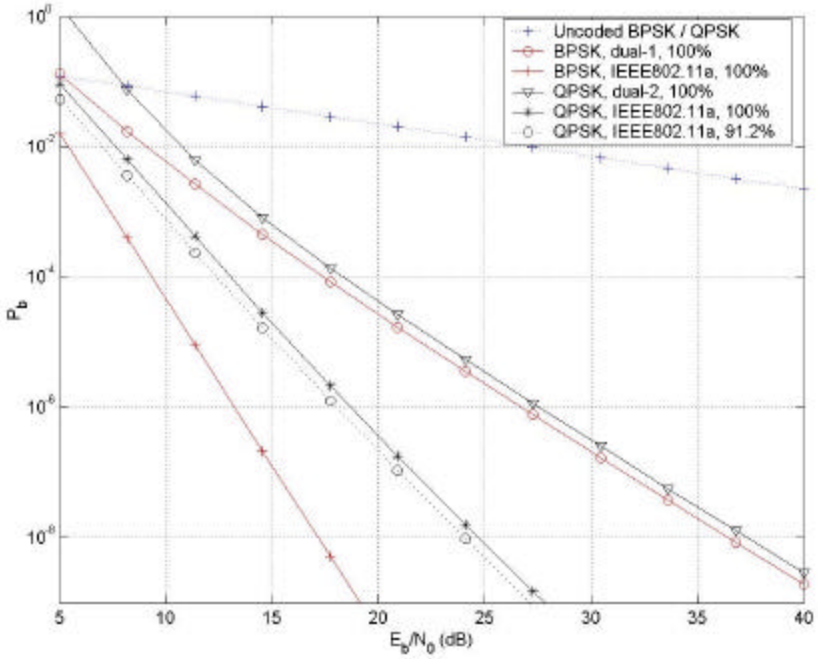


Figure 4.15. Probability of Bit Error for BPSK and QPSK with SDD in Nakagami Fading (m = 0.5) Channel Using IEEE 802.11a and Dual-k Convolutional Codes.

E. SUMMARY

Chapter IV expanded the results originally obtained from the AWGN scenario to Nakagami fading channels. A second random variable expressing the fading effects now modeled a new upper bound of $P_2(d)$ for any M-ary coherent demodulator followed by SDD. The use of this upper bound in the special case of M-QAM modulations led to various performance analyses involving 16-QAM, 64-QAM, BPSK and QPSK under four fading figures m: 3, 2, 1 and 0.5. The reader had the opportunity to notice how the upper bound improvement technique is still valid and how convolutional codes can largely enhance performances under fading conditions. With Chapter IV, the last goals initially proposed for this thesis were achieved, leaving to the next chapter the task of summarizing all results.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND FUTURE WORK

Many design tools and two analytical models were developed in this thesis. They have a general scope of applicability much larger than the special case of communication systems that make use of M-QAM modulations with Viterbi Soft-Decision decoding. The main conclusions of each of those tools and models will now be summarized per chapter together with a suggestion for future work at the end.

A. CONCLUSIONS

In Chapter II, a numerical method for determining the $b(d)$ spectrum of any convolutional code was provided. This method turns into a general tool that can be applied to every convolutional code. As a specific application, it was shown how to adapt $b(d)$ spectra obtained for binary convolutional codes to be used in many M-QAM modulation. The special case of the IEEE 802.11a code was analyzed and a table with its various $b(d)$ spectra was presented. It is interesting to notice how its d_{free} associated with each M-QAM modulation decays as M increases. For instance, the use of IEEE 802.11a convolutional code with BPSK leads to a $d_{free} = 10$, whereas this value drops to 3 when the same code operates with 64-QAM.

In Chapter III, a model for studying just the case AWGN channels was created. A simple formula for the upper bound of $P_2(d)$ for any M-ary coherent demodulator followed by a SDD was analytically derived. This formula is valid for M-QAM and also for the cases of orthogonal modulations such as M-FSK. A technique that improves the upper bound of $P_2(d)$ trying to tighten it to realistic data was shown. The possibilities of application and the efficiency of this technique were demonstrated depending on how large M is. As an example of extreme cases, no improvements are admitted to BPSK, while a realistic gain of up to 7.5 dB was achieved for 64-QAM. The performance analyses of 16-QAM, 64-QAM, BPSK and QPSK, associated with up to three different convolutional codes were shown. Those analyses demonstrated that the results for BPSK differ from QPSK when considering the different $b(d)$ spectra they present for the same

code. The IEEE 802.11a code demonstrates itself better than dual-k code only when associated with BPSK and QPSK. When using 16-QAM and 64-QAM, the dual-k code shows more efficiency. The CHL code demonstrates excellent performance with 16-QAM especially considering that it consumes only half the bandwidth consumed by dual-k and IEEE 802.11a codes. Its coding gain reached 5.5 dB at $P_b = 10^{-5}$ when considering the use of improvements on the upper bound.

In Chapter IV, the model for AWGN channels from Chapter III was expanded for the Nakagami fading channels, originating the second model of this thesis. A formula for the upper bound of $P_2(d)$ for any M-ary coherent demodulator followed by a SDD was analytically derived for Nakagami fading channels, showing itself valid for both M-QAM and for the cases of orthogonal modulations such as M-FSK. The performance analyses of 16-QAM, 64-QAM, BPSK and QPSK were repeated for the new model utilizing four fading figures m: 3, 2, 1 and 0.5. Considering all different configurations of modulations and codes, the same *relative* behavior among performances for the AWGN scenario was confirmed in Nakagami fading channels. As a general characteristic, convolutional codes associated to M-QAM with SDD demonstrated large coding gains, greater than 20 dB at $P_b = 10^{-5}$, under severe fading conditions. Also, this gain increases as the fading becomes worse. Concerning the upper bound improvements, they generate relative gains that maintain themselves constant as the fading figure of the channel varies. Additionally, as could be expected, those gains are equal to those provided in the pure AWGN scenario. The uncoded channels of the four studied modulations were also depicted and clearly demonstrated a trend for the same performances, no matter which M-QAM is being used, when operating in severe fading conditions ($m \leq 1.5$).

In the Appendix, a general formula to determine $b(d)$ spectra of dual-k codes is analytically derived.

B. FUTURE WORK

Since the topics in this thesis have general application, many previous studies can now be revisited using the developed models in order to extend their results for the cases of M-QAM with SDD such as the performance analysis of OFDM in Reference [8].

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX. DERIVATION OF $b(d)$ FOR DUAL-*K* CONVOLUTIONAL CODES

The dual-*k* codes compound a class of nonbinary convolutional codes that are easily decoded by means of the Viterbi algorithm using either soft-decision or hard-decision decoding. It consists of two ($K = 2$) *k*-bit shift-register stages and $n = 2k$ function generators. Its output is two *k*-bit symbols for an input of one *k*-bit symbols ($R_C = 1/2$). The general form for the transfer function of a rate 1/2 dual-*k* code is expressed as

$$T(D, N, J) = \frac{(2^k - 1)D^4 J^2 N}{1 - NJ\left[2D + \left(2^k - 3\right)D^2\right]},$$

where *D* and *N* are defined in Section II.B.4 and the exponent on *J* is equal to the number of branches in a given path [3].

The expression for $T(D,N)$ can be obtained by making $J = 1$ in the formula for $T(D,N,J)$,

$$T(D,N) = \frac{(2^k - 1)D^4 N}{1 - N\left[2D + \left(2^k - 3\right)D^2\right]}. \tag{A.1}$$

For the first step, $T(D,N)$ will be expanded into an infinite series in powers of D and *N*. Assume

$$A = 2^k - 1 \;\Rightarrow\; (A-2) = 2^k - 3, \tag{A.2}$$

and rewrite Equation (A.1) as

$$T(D,N) = \frac{AD^4 N}{1 - N\left[2D + (A-2)D^2\right]}. \tag{A.3}$$

Let

$$B = A - 2, \tag{A.4}$$

so, Equation (A.3) can be developed as

$$T(D,N) = \frac{AD^4N}{1-N\left[2D+BD^2\right]} = AD^4N\sum_{j=0}^{\infty}\left(N\left[2D+BD^2\right]\right)^j$$

$$= AD^4N\sum_{j=0}^{\infty}\left(2+BD\right)^j D^jN^j = A\sum_{j=0}^{\infty}\left(2+BD\right)^j D^{j+4}N^{j+1}$$

$$= A\sum_{j=0}^{\infty}\left(\sum_{i=0}^{j}\binom{j}{i}\left(BD\right)^i 2^{(j-i)}\right)D^{j+4}N^{j+1}$$

$$T(D,N) = A\sum_{j=0}^{\infty}\left(\sum_{i=0}^{j}\binom{j}{i}B^i 2^{(j-i)}D^{j+4+i}\right)N^{j+1}. \qquad \text{(A.5)}$$

Assuming that

$$x = j+4+i \quad \Rightarrow \quad i = x-(j+4), \qquad \text{(A.6)}$$

Equation (A.5) becomes

$$T(D,N) = A\sum_{j=0}^{\infty}\left(\sum_{x=j+4}^{2j+4}\binom{j}{x-(j+4)}B^{(x-(j+4))}2^{(2j-x+4)}D^x\right)N^{j+1}. \qquad \text{(A.7)}$$

Considering

$$c(j,x) = \binom{j}{x-(j+4)}B^{(x-(j+4))}2^{(2j-x+4)} \qquad \text{(A.8)}$$

applied to Equation (A.7), the following can be obtained,

$$T(D,N) = A\sum_{j=0}^{\infty}\sum_{x=j+4}^{2j+4}c(j,x)D^xN^{(j+1)}$$

$$T(D,N) = A\left[N\sum_{x=4}^{4}c(0,x)D^x + N^2\sum_{x=5}^{6}c(1,x)D^x + N^3\sum_{x=6}^{8}c(2,x)D^x + \ldots\right]$$

$$T(D,N) = A\left[Nc(0,4)D^4 + N^2c(1,5)D^5 + N^2c(1,6)D^6 + N^3c(2,6)D^6 + N^3c(2,7)D^7 + N^3c(2,8)D^8 + \ldots\right]$$

$$
\begin{aligned}
T(D,N) = A\Big[& N\ c(0,4)\,D^4 \\
& + N^2 c(1,5)\,D^5 \\
& + \left(N^2 c(1,6) + N^3 c(2,6)\right)D^6 \\
& + \left(N^3 c(2,7) + N^4 c(3,7)\right)D^7 \\
& + \left(N^3 c(2,8) + N^4 c(3,8) + N^5 c(4,8)\right)D^8 \\
& + \ \ldots\Big]
\end{aligned}
,
$$

which yields

$$
T(D,N) = A\sum_{d=4}^{\infty}\left[\sum_{n=f(d)}^{d-3} c(n-1,\,d)\,N^n\right]D^d\,, \tag{A.9}
$$

where $f(d) = \begin{cases} \dfrac{d-2}{2} & \text{(even } d\text{)} \\ \dfrac{d-1}{2} & \text{(odd } d\text{).} \end{cases}$

The second step is to find the partial derivative of Equation (A.9) with respect to *N*, and make *N* equal to 1 afterwards in order to obtain another series in powers of D only.

$$
\frac{\partial T(D,N)}{\partial N} = A\sum_{d=4}^{\infty}\left[\sum_{n=f(d)}^{d-3} n\ c(n-1,\,d)\,N^{(n-1)}\right]D^d\,,
$$

hence

$$
\left.\frac{\partial T(D,N)}{\partial N}\right|_{N=1} = \sum_{d=4}^{\infty}\left[A\sum_{n=f(d)}^{d-3} n\ c(n-1,\,d)\right]D^d = \sum_{d=4}^{\infty} b(d)\,D^d\,. \tag{A.10}
$$

As can be seen, the expression for $b(d)$ turns out to be the equation that defines the coefficients of this last series as a function of d. Hence, applying Equations (A.2), (A.4)and (A.8) to Equation (A.10) the following is obtained:

$$
b(d) = (2^k - 1)\sum_{n=f(d)}^{d-3} n\binom{n-1}{d-(n-1)-4}(2^k-3)^{(d-(n-1)-4)}\,2^{(2(n-1)-d+4)}\,.
$$

Finally, it can be concluded:

$$b(d) = (2^k - 1) \sum_{n=f(d)}^{d-3} \binom{n-1}{d-n-3} (2^k - 3)^{(d-n-3)} \, 2^{(2n-d+2)} \, n \, ,$$

where $f(d) = \begin{cases} \dfrac{d-2}{2} & \text{(even } d \text{ and } d \geq 4\text{)} \\ \dfrac{d-1}{2} & \text{(odd } d \text{ and } d \geq 5\text{).} \end{cases}$

Notice that the minimum free distance d_{free} = 4 symbols (4k bits).

LIST OF REFERENCES

[1] L. W. Couch II, *Digital and Analog Communication Systems*, Upper Saddle River, NJ, Prentice-Hall, 2001.

[2] Institute of Electrical and Electronics Engineers, 802.11a, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer Extension in the 5 GHz Band*, 16 September 1999.

[3] J. G. Proakis, *Digital Communication*s, New York, NY, McGraw-Hill, 2001.

[4] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*, Upper Saddle River, NJ, Prentice-Hall, 1995.

[5] IEEE Standard 802.11a, *Supplement to IEEE Standard for Information Technology*, p. 16, 1999.

[6] B. Sklar, *Digital Communications - Fundamentals and Applications*, Prentice Hall, New Jersey, 2001.

[7] J. Chang, D. Hwang and M. Lin, "Some Extended Results on the Search for Good Convolutional Codes," *IEEE Transactions on Information Theory*, Vol. 43, No. 5, p. 1695, 1997.

[8] P. Count, "Performance Analysis of OFDM in Frequency Selective, Slowly Fading Nakagami Channels," Master's Thesis, Naval Postgraduate School, Monterey, California, December 2001.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Chairman, Code EC
   Department of Electrical of Computer Engineering
   Naval Postgraduate School
   Monterey, California

4. Professor Tri T. Ha, Code EC/Ha
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

5. CDR Jan E. Tighe
   Naval Information Warfare Activity
   Ft. Meade, Maryland

6. CDR Marcelo J. A. Motta
   Directorate of Telecommunications of the Navy
   Brazilian Navy
   Rio de Janeiro, Brazil

7. LCDR Rogerio C. Manso
   Rio de Janeiro, Brazil