National Library of Canada
Acquisitions and Bibliographic Services Branch
395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada
Direction des acquisitions et des services bibliographiques
395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

Study of
Trellis Coded Modulation
and
Error Control Coding

by

Marc Trichard

A thesis submitted to
the School of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree

Master of Applied Science

Ottawa-Carleton Institute for Electrical Engineering
Faculty of Engineering
Department of Electrical Engineering
University of Ottawa
Ottawa, Ontario, Canada, K1N 6N5

November 3, 1995

ISBN 0-612-07865-5

Canada

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

I hereby declare that I am the sole author of this thesis.

I authorize the University of Ottawa to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Marc Trichard

I further authorize the University of Ottawa to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Marc Trichard

Abstract

In this thesis, we examine some general theoretical features of Trellis Coded Modulation. The codes in use are the recursive Ungerboeck convolutional codes and they are combined with Quadrature Amplitude Modulation over an Additive White Gaussian Noise channel.

The search of optimum mappings is first treated and the question of having some smaller punctured constellations from a larger one is considered. Performance evaluation of TCM codes is provided according to geometric conditions that the mappings must stand in order to reduce the complexity of the calculation.

Computation of transfer functions is provided and this leads to a theoretical analysis for probability expressions of event, symbol and bit errors. A second order term expression is also elaborated which takes into consideration both error events on the trellis and errors on parallel transitions.

Computer simulations have been run along with High Density Television transmission parameters. The results are presented and discussed. The use of error control coding techniques makes a great improvement in performance. For this purpose, Reed-Solomon codes have been concatenated with TCM codes, and gone through interleaving to ensure a higher error recovery efficiency.

Acknowledgments

I would like to thank my supervisor, Dr. Jean-Yves Chouinard, for his constant help and strong support on this work. During the past two years, he managed to direct my studying and searching. Above all, he conveyed my attraction for communications and research.

I am extremely thankful to Dr. Abbas Yongaçoğlu for his great availability and consistent advices whenever I begged them. His suggestions were staid, clear and largely beneficial.

An important part of this thesis has been promoted by CRC, thanks to Mr. Bernard Caron who trustfully permitted to subsidize it. His confidence and patience were a lot appreciated. Many comments and discussions provided by Dr. Yiyan Wu improved the implementation of the simulations and managed to advance them continuously.

I wish to thank the large efforts and help of Dr. Joseph Boutros who always answered demanding questions in a very detailed and complete manner.

Contents

List of Figures

List of Tables

List of Acronyms

ATV	Advanced television
AWGN	Additive white Gaussian noise
BER	Bit error rate
CROSS	Non-rectangular quadrature amplitude shift keying
EER	Event error rate
HDTV	High definition television
PAM	Pulse amplitude modulation
QAM	Quadrature amplitude modulation
QASK	Quadrature amplitude shift keying
RS	Reed-Solomon (code)
SER	Symbol error rate
SNI	Signal-to-interference-noise ratio
SNR	Signal-to-noise ratio
TCM	Trellis coded modulation
dB	Decibel

List of Symbols

m	number of input bits to the TCM encoder
k	number of input bits to the convolutional encoder of the TCM code
n	number of output bits from the convolutional encoder of the TCM code
ν	number of registers of the TCM encoder
$N_\nu = 2^\nu$	number of states of the TCM encoder
r_i	i-th register of TCM encoder state
h^j	parity-check coefficients of the TCM encoder
(n, k, ν)	parameters of a TCM code (For Ungerboeck codes, $n - k = 1$.)
$X_t = \left(x_t^{(m)}, x_t^{(m-1)}, \ldots, x_t^{(1)}\right)$	input symbol to TCM encoder at time t
$Y_t = \left(y_t^{(m)}, y_t^{(m-1)}, \ldots, y_t^{(1)}, y_t^{(0)}\right)$	output symbol of TCM encoder at time t
$S_t = \left(s_t^{(m)}, s_t^{(m-1)}, \ldots, s_t^{(1)}, s_t^{(0)}\right)$	corresponding signal mapped at time t
$E_t = \left(e_t^{(m)}, e_t^{(m-1)}, \ldots, e_t^{(1)}, e_t^{(0)}\right)$	symbol error which occurred at time t
$\mathbf{S}_t^L = [S_t, S_{t+1}, \ldots, S_{t+L-1}]$	sequence of L signals from index t to $t + L - 1$
$\mathbf{S}^L = [S_1, S_2, \ldots, S_L]$	sequence of L signals with no time dependence
$\mathbf{E}^L$	sequence of L error symbols
$\mathbf{Y}^L$	sequence of L output symbols
$\mathbf{X}^L$	sequence of L input symbols
$\mathbf{W}(\mathbf{E}^L)$	*event error weight function* for the event error $\mathbf{E}^L$
$t_{s_i \to s_j}$	transition from state s_i to state s_j
$Y_{t_{s_i \to s_j}}$	output symbol from transition $t_{s_i \to s_j}$
$\mathbf{G}(E)$	*error weight matrix* for a symbol error E
$\mathbf{G}_{ij}$	transition matrix from state i to j

$\mathbf{G}$	matrix of error event distribution
$\mathcal{I}$	vector of transitions from the zero-state to all others
$\mathcal{I}d_x$	identity matrix of $x \times x$ elements
$\mathcal{O}$	vector of transitions from any non zero-state to zero state
$\mathcal{G}$	matrix of transitions within the non zero-states
$\mathcal{Y}_0$	subset of signals used for *uniformity* of TCM scheme
$\tilde{Y}$	characteristic output symbol of *uniformity* property
$T(D)$	transfer function of a TCM code
$T(D, I)$	extended transfer function for bit error evaluation
$T(D, Y)$	extended transfer function for symbol error evaluation
$Z = \exp\left(-\frac{1}{4N_0}\right)$	Bhattacharyya bound
$\mathcal{P}_e$	probability of event error
$\mathcal{P}_S$	probability of symbol error
$\mathcal{P}_b$	probability of bit error
Δ_0	minimum distance between points of a constellation
Δ_k	minimum distance between points of a subset of order k
E_S	average energy per symbol over the constellation
N_0	half side spectrum of AWGN distribution
d_{free}	minimum distance between two paths after an event error occurred on one of them, called *free* distance of the TCM code
d_{next}	second next minimum distance between two paths
$N_e(d_{free})$	average number, over the trellis states, of d_{free}-distant paths
$N_S(d_{free})$	average number, over the states of the trellis, of incorrect output symbols on a d_{free}-distant path from the correct one
$N_b(d_{free})$	average number, over the states of the trellis, of incorrect output bits on a d_{next}-distant path from the correct one
$N_e(d_{next})$	average number, over the trellis states, of d_{next}-distant paths
$N_S(d_{next})$	average number, over the states of the trellis, of incorrect output symbols on a d_{next}-distant path from the correct one

$N_b(d_{next})$	average number, over the states of the trellis, of incorrect output bits on a d_{next}-distant path from the correct one
α_{cod}	half ratio of the square of the free distance of the code over the average energy per symbol ($\alpha_{cod} = \frac{d^2_{free}}{2E_S}$)
α_{unc}	half ratio of the square of the minimum distance Δ_0 between points of the constellation over the average energy per symbol ($\alpha_{unc} = \frac{\Delta_0^2}{2E_S}$)
β_{cod}	half ratio of the difference between the square of the free and next distances of the code over the average energy per symbol ($\beta_{cod} = \frac{d^2_{next} - d^2_{free}}{2E_S}$)
ρ	signal to noise ratio E_S/N_0
ρ_i	signal to interference noise ratio E_S/E_{S_i}
erfc()	complementary error function
$Q()$	Complementary distribution function of a unit variance zero mean Gaussian random variable $\left(Q(x) = \frac{1}{2}\text{erfc}\left(\frac{x}{\sqrt{2}}\right)\right)$

Chapter 1

Introduction

1.1 Background

Communications have always been the way different cultures have met each other. They are the main source of knowledge successive generations have grown with and transmitted to their offspring. They are both causes and consequences of endless human advance in discovery of the universe, of life, of infinity. They manage human consciousness, rules and law.

Men have run after them and for them: forty two kilometers separated Athens from Marathon, site of a victory the news of which was carried by a long-distance runner! Eighty days was the incredible bet of the famous though fictitious Phileas Fog to make the complete turn around the earth. At that time, the existence of train and the use of steamboats helped a lot... Communications went through the human progress, and the writings flew in the air, over oceans, inside historical planes piloted by Mermoz, Saint-Exupery and all others.

All this was converging towards an exploding demand for data transmission and other communication. Telegraphs first used the hertzian channel and Marconi established the first radio link between Europe and America in 1901. These two examples consisted of text and voice transmission: still image processing and transmission were missing. Development of new techniques diversified the extent of the means of

communications. Today, satellites play a major role in information transmission, as available bandwidth expands.

Presently, High Definition Television (HDTV) is about to succeed the older television broadcasting standards through a fully digital implementation. Severe constraints such as a high bit rate, same bandwidth allocation to ensure compatibility as prior transmission standards keep the main difficulties. Powerful modulation is to be selected, which requires efficient error control coding.

This thesis provides a study of Trellis Coded Modulation (TCM): optimum mappings have been searched, theoretical performances evaluated, and a transmission model designed over an Additive White Gaussian Noise (AWGN) channel. Simulation and theoretical results are compared. Reed-Solomon codes are then concatenated with TCM. A final improvement is provided by interleaving, which confirms the high performances that were expected and are required for HDTV broadcasting.

1.2 Motivation

As we were investigating some practical details and simulating the performance, we observed a large variation between theoretical and simulation results for TCM. The discrepancies have been explored and it came out that a wrong, or at least an extremely poor mapping had been used. This was the first step of the search. It led to a systematic construction of optimum mappings, and we studied the problem of extending a mapping to a higher level mapping.

This considerably reduced the gap between the theoretical and experimental results, but the discrepancy was still significant. It was therefore desirable to measure the exact theoretical probabilities of errors, since up to now, the only references dealt with event errors rather than symbol errors, and parameters were known asymptotically. Furthermore, we focused on the bit error rate evaluation. The computations required important justifications, for the algorithm in use to evaluate theoretical performances of TCM was very powerful, though restricted to some strict condition, the

uniformity condition [1], that had also to be observed and verified.

The program that has been used in computations could give more than first order parameters for the probability expressions. Thus, second order terms have been studied and a general expression with second order terms has been derived.

Having resolved the issues related to TCM, the transmission model could be improved by employing concatenation. For this purpose, RS codes have been computed and interleaving has been used to cope with burst errors.

1.3 Contributions

This thesis attempts to add to the existing literature about Trellis Coded Modulation some theoretical evaluations. These are:

1. construction of good mappings adapted to TCM
   - definition and derivation of optimum mappings at up to the third order
   - mappings satisfying the *uniformity condition* to provide tractable evaluation
2. exact theoretical expressions for the probabilities of error of TCM transmission model with a Quadrature Amplitude Modulation (QAM) over an AWGN channel rather than asymptotic parameters (N_{free}, ...);
3. second order evaluation for probabilities of error in order to provide more realistic theoretical expressions which are relevant for medium signal-to-noise-ratio (SNR). This would particularly concern:
   - bit error rate so as to permit a valid comparison between models with different channel code rates or different spectral efficiency;
   - symbol error rate rather than event error rate for more reliable theoretical expressions.

Simulations have been run and the performance of concatenated TCM codes with Reed-Solomon (RS) codes has been explored for HDTV transmission parameters.

1.4 Outline of the thesis

Chapter 2 of this thesis provides a study of a systematic search of optimum mappings for any level of TCM QAM. The question of the possible extension of an optimum mapping to a larger one is particularly focused on. An optimization inside each sub-constellation is also suggested.

Chapter 3 provides a review and an complete justification of the theoretical performance evaluation of TCM codes. Strict conditions, such as the *uniformity condition* are explored and explained by examples and extended to the particular case of Ungerboeck recursive codes. The optimum mappings derived in chapter 2 are then shown to satisfy the *uniformity condition*. Observation of its effect over the same example is reported and this clarifies its behavior for different types of error.

Theoretical expressions have been computed and reported in chapter 4. This chapter also contains an analysis of the computed transfer functions of the codes, based on an inspection of the trellis. Second order terms and larger parameters have been carefully given, whenever they could be determined.

Chapter 5 studies a different transmission model adapted for even spectral efficiency models. It deals separately with the in-phase and quadrature components of the mapped signals. Through verification of the *uniformity condition*, theoretical expressions for the transfer functions have also been derived.

Chapter 6 is a report of the first and second order parameters for the two different transmission models and provides event, symbol and bit error rate curves for different approximations and codes.

In chapter 7, an investigation of HDTV broadcasting is reported. The transmission parameters are given and a simulation model is described. Simulations results are presented, discussed and compared with theoretical expressions for probability of

error, for both models.

Conclusions and suggestions for further research are provided in chapter 8.

In the appendices, various details related to the calculations are presented.

Chapter 2

Design of optimum mappings

2.1 Introduction

In data transmission schemes, both detection and correction of errors can be achieved through the addition of redundant bits: the TCM schemes used in this thesis provide redundancy using a convolutional encoder. In this thesis, all the codes are recursive and taken from Ungerboeck [12], because they are famous and easy to deal with. Following the way this modulation scheme was originally designed [11], only some of the input bits (in-bits) are used by the convolutional encoder. These will be referred to as the uncoded in-bits while the bits used to compute the redundant bit will be called coded in-bits. Under the same notation, uncoded in-bits and output bits (out-bits) are identical, whereas the coded out-bits consist of the coded in-bits and the redundant bit. Thus, the uncoded out-bits form subsets of possible input symbols inside which the coded out-bits of the encoder are the same. The coded out-bits are the most significant ones and may easily be recovered using the trellis structure, because of the memory of the code. Meanwhile, the uncoded out-bits are chosen to be the least significant bits. Figure 2.1 illustrates this TCM coding scheme. If t denotes time and m the number of bits per input symbol, the output symbol is defined as:

$$Y_t = \left(y_t^{(m)}, y_t^{(m-1)}, \ldots, y_t^{(1)}, y_t^{(0)}\right)$$

and the input symbol in terms of the in-bits is:

$$X_t = \left(x_t^{(m)}, x_t^{(m-1)}, \ldots, x_t^{(1)}\right)$$

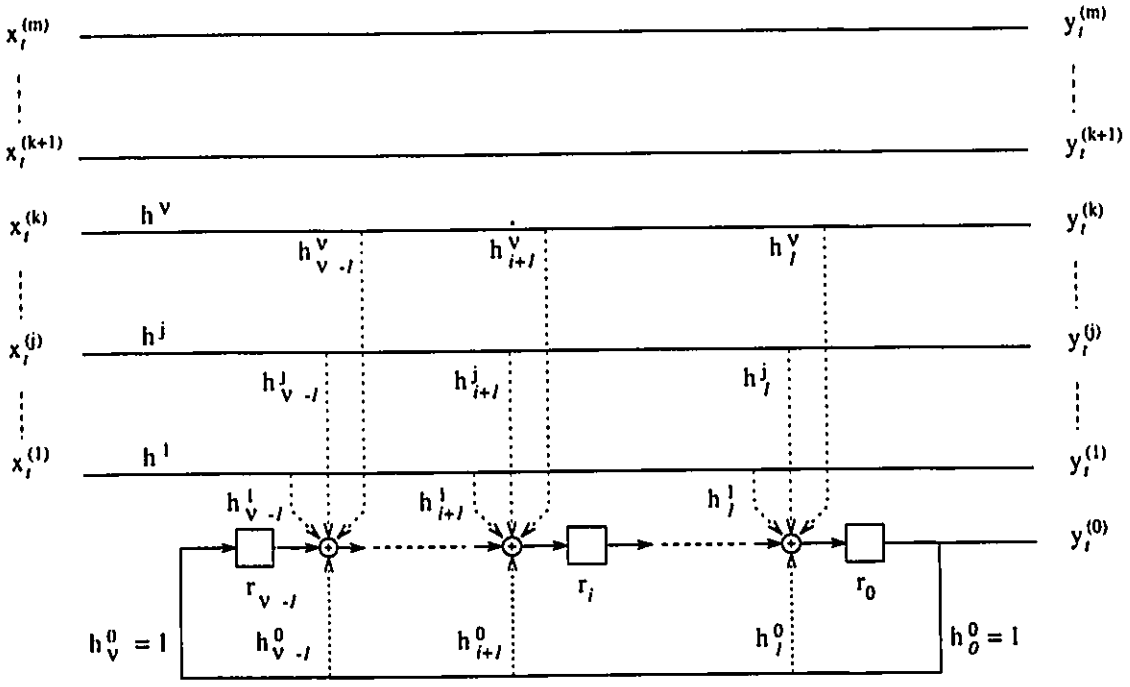

Figure 2.1: Block diagram of a general recursive convolutional encoder used in a TCM.

The TCM principle consists of assigning, for a given signal constellation, signals to output symbols of the TCM encoder. This association between outputs of the code and points of a constellation is termed signal mapping. The function which associates a signal on the constellation to an output symbol is called $f()$, as depicted in Figure 2.2.

It is necessary to minimize the probability of error for the uncoded bits in order to ensure good TCM encoder performance. Each possible coded output of the convolutional code corresponds to a subset of input symbols. These subsets are symbols subsets. The mapping is constructed in such a way that the minimum distance between the points of the corresponding constellation subset is maximized. Since the metric of the Viterbi decoder consists of the Euclidean distance between constellation

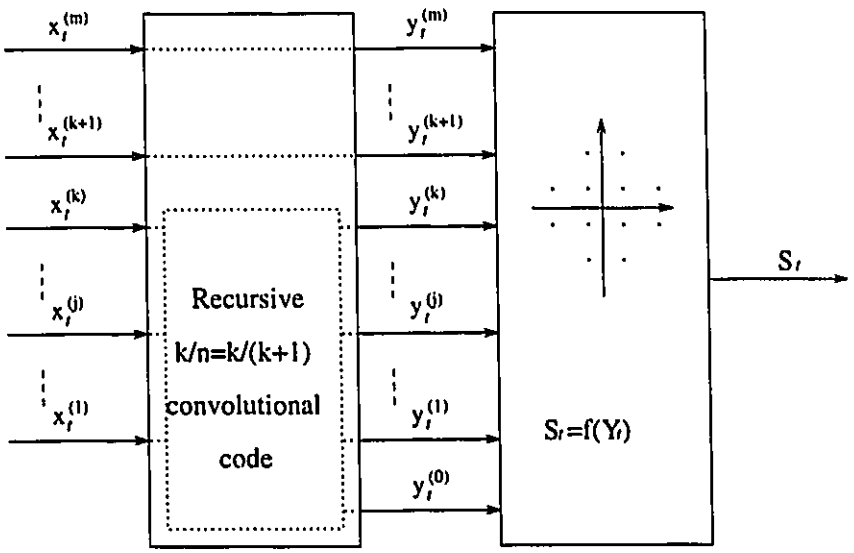

Figure 2.2: Model of TCM coding scheme.

points, this construction model takes the problem of uncoded out-bits into account and controls the possible errors which may occur on them.

As far as QAM is concerned, the focus will be mainly on the 16-QASK, the so-called 32-CROSS and the 64-QASK modulations. In this thesis, no higher than a 64 point constellation has been used because simulations have been run for digital HDTV transmission models, for which 64-QASK is sufficiently bandwidth efficient.

In this chapter, we define a class of optimum mapping based on the consideration of the minimum distances between points inside subsets. This chapter is a study of the derivation of these optimum mappings and determines whether an optimum mapping can be an extension of a smaller optimum one. One may ask: "Does a 32-CROSS optimum mapping contain all 16-QASK optimum mapping points, with '0' as the least significant bit?" or: "Do some points from the 16-QASK mapping need to be assigned to exterior ones on the 32-CROSS in order to make the latter optimum?"

To answer these questions, in section 2.3, the mappings of simple 4-QASK, 8-CROSS and 16-QASK modulations have been considered. Then, section 2.4 considers the 32-CROSS mapping, since, as it will be seen later, it reveals an incompatibility be-

tween the two constraints of optimization and the extension of the previous 16-QASK mapping. 64-QASK will be dealt with in section 2.5: again, the same incompatibility occurs.

2.2 Definition of optimum mappings

The goal of this section is to deal with the discrepancy of error on parallel transitions during the trellis analysis in the demodulator. We do not want to consider whether the mapping of the coded out-bits may improve the performance of TCM code or not by permuting some subsets. On the opposite, our only consideration is made on the structure of the subsets themselves. Therefore, the derivation of mappings that deal with the problem of uncoded out-bits is closely related to the distances between the symbols of a subset.

A mapping is called optimum at the first order, if the minimum distance between the symbols of the subsets is maximized. It is called optimum at the second order, if it is optimum at the first order and if the second minimum distance between the symbols of the subsets is maximized. It is called optimum at the n^{th} order, if it is optimum at the n-1^{th} order, and if the n^{th} minimum distance between the symbols of the subsets is maximized.

As an example, consider a code structure with no parallel transition, or no uncoded in-bits. Then, the constellation is made of subsets of one single point. There cannot be any error on parallel transitions and therefore, the distances between symbols inside a subset do not exist. On the other hand, we may still consider these distances as zero. The maximum value taken by the minimum distance is also zero. Thus, any mapping has its minimum distance inside subsets being maximum. And finally, any mapping is optimum! This particular case is not very relevant because as far as TCM is concerned, good codes have been discovered [12] where the number of coded in-bits is one or two, but hardly three. And any transmission model for which the number of bits per symbol has to be 4 or more would use a TCM code with parallel transitions.

Consider a code structure with a pair of parallel transitions: each subset has two points. If all these subsets have equal distance between their two symbols, once they are optimum at the first order, they also are at any order.

However, if the size of the subsets is large, say more than 8, there might be a maximum number of different distances of $\binom{8}{2}$ and a minimum number of different distances of 8, reached when all points are aligned. In this case, dealing with an optimality of order more than 3 is complex indeed. In this thesis, optimum mappings at an order up to 3 are considered, but only when the subsets have 4 points, which is a case where it is easy to deal with the second and third minimum distances.

The study of optimum mappings at the first order may easily be done whatever the size of the subsets is. Whenever there is no ambiguity, the term optimum mapping will be used instead of optimum mapping at the first order. Then, if optimality at a higher order is considered, we will use the appropriate term, say optimum mapping at the order n.

2.3 4-QASK, 8-CROSS and 16-QASK optimum mappings

In this chapter, the graphical symbols used in the figures of mappings are (almost) arbitrarily chosen: they were only chosen in order to easily distinguish them. The reason is that the number of uncoded in-bits is unknown: if there are no uncoded in-bits, or no parallel transition, the constellation has to be shared into subsets of one single point. If there is one uncoded in-bit, the constellation has to be shared into subsets of two points. If there are $m-k$ uncoded in-bits, the constellation has to be shared into subsets of 2^{m-k} points. Thus, the number of subsets and consequently the number of graphical symbols to represent them may differ as long as the code structure does. The problem is kept general and the corresponding subsets could be the ones obtained after either the first, the second or more construction steps.

2.3.1 4-QASK optimum mapping

For the particular case of a 4-QASK mapping, there cannot be any parallel transition in the trellis code, nor may the constellation be shared in two subsets of two points, but only may be considered as 4 subsets of one single point because there only is one (coded) in-bit and two (coded) out-bits. Thus, as mentioned earlier, any 4-QASK mapping is optimum at any order. However, a typical construction was used to derive the following 4-QASK (optimum) mapping has been derived. This is because we expect larger constellations to match this one and this construction method leads to optimum mappings at the first order for larger constellations.

The construction of an optimum mapping for 4-QASK is described in Figure 2.3. It consists of only one step where the initial set of four 4-QASK signals is divided into two subsets, chosen such that the minimum distance between the points of these subsets is maximum. Again, it should be noted that this minimum distance is not a distance between symbols inside a given subset.

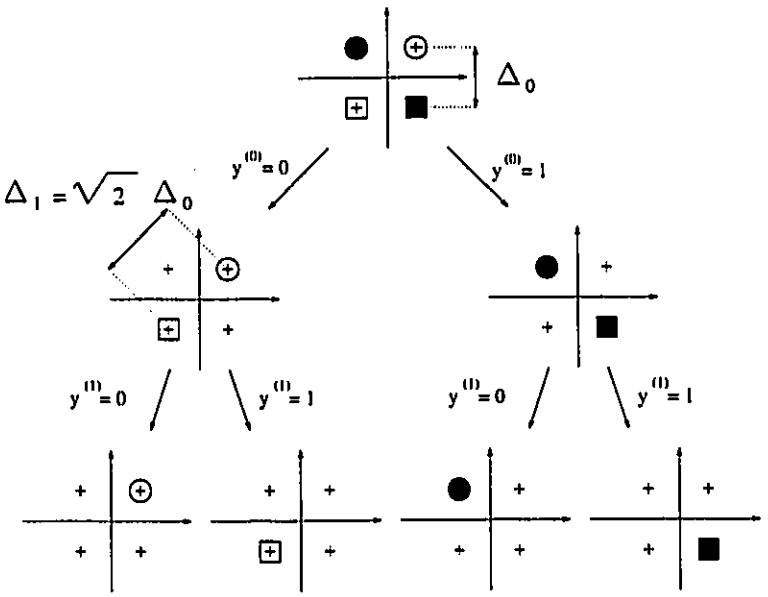

Figure 2.3: Construction of an optimum mapping for 4-QASK.

It is easy to derive the minimum distance Δ_1 between the elements of the two

subsets in term of the minimum distance Δ_0 between elements of the initial set:

$$\Delta_1 = \sqrt{2}\Delta_0$$

An optimum mapping for 4-QASK is then represented in Figure 2.4.

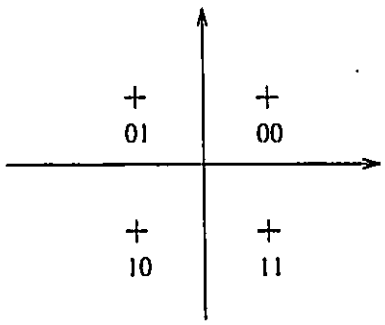

Figure 2.4: Optimum mapping of bits $y^{(1)}y^{(0)}$ to constellation points for 4-QASK.

2.3.2 8-CROSS optimum mapping

Here, all possible steps are considered in constructing an optimum mapping for any situation. In this case, there may be one uncoded bit or none. Therefore, the constellation may be shared into 4 subsets of 2 points or 8 subsets of one single point. Each of these two situations may lead to different optimum mappings. Since the latter is always optimum, its corresponding optimum mapping is chosen to be the one of the former.

In order to try to extend the previous 4-QASK mapping, at each step, the points that are inside the 4-QASK mapping shape must match their first bits $y^{(0)}$ and $y^{(1)}$ with the 4-QASK mapping and their third bit $y^{(2)}$ has to be 0, whereas the outside points must have their third bit $y^{(2)}$ equal to 1. Thus an extension of the 4-QASK mapping to a larger 8-CROSS one can be constructed. Figure 2.5 depicts the construction of an 8-CROSS mapping, extending the previous 4-QASK optimum one.

This construction makes the minimum distance between points of any of the two

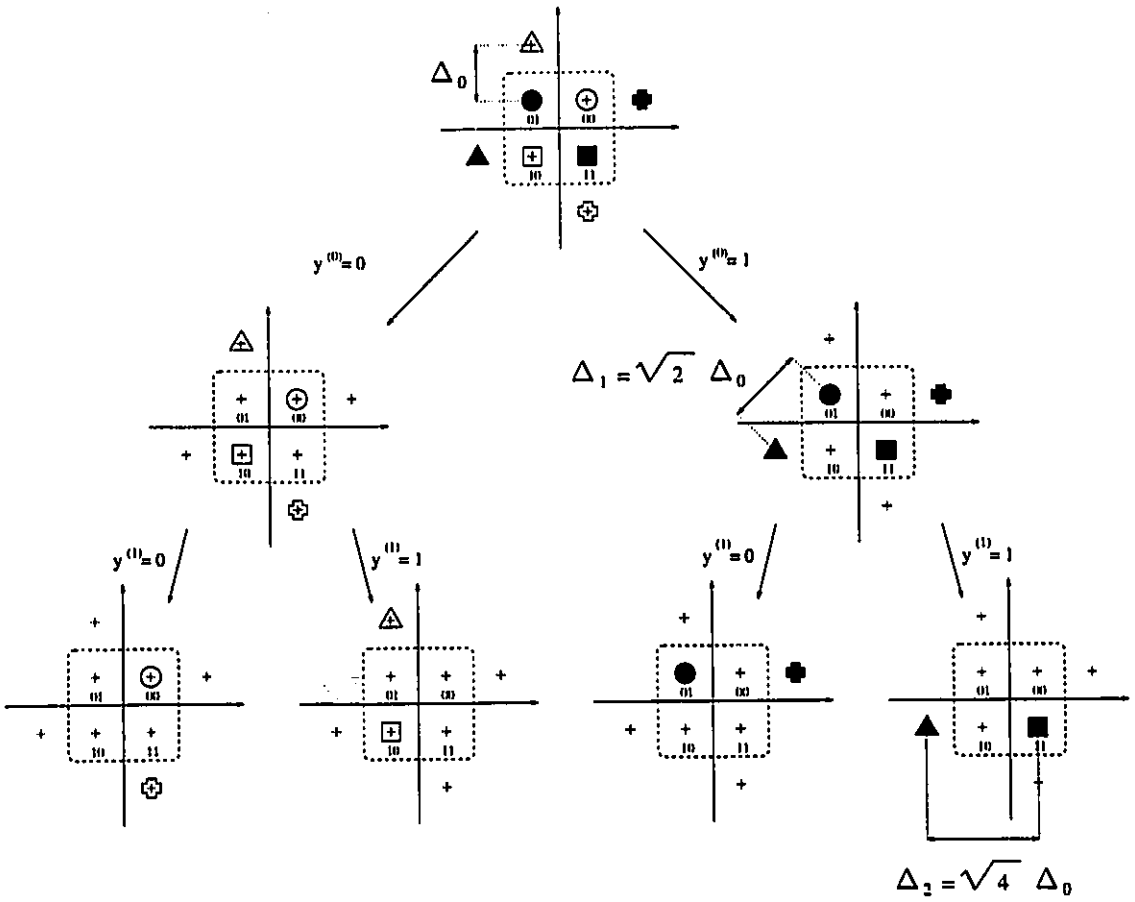

Figure 2.5: Construction of an 8-CROSS mapping by extending the previous 4-QASK optimum one.

first subsets:

$$\Delta_1 = \sqrt{2}\Delta_0$$

and the minimum distance between any points in the next four subsets is:

$$\Delta_2 = \sqrt{4}\Delta_0$$

The construction maximizes the minimum distance Δ_2, and the resulting mapping is optimum at the first order. Again, for this case where there is only one uncoded bit, once the mapping is optimum at the first order, it is optimum at any order.

The last step is not represented in Figure 2.5. The rule is: the third bit $y^{(2)}$ of all points is assigned to a "0" or a "1" depending on whether they are inside the 4-QASK shape or not. This ensures that the 8-CROSS mapping matches the 4-QASK mapping.

The optimum mapping for 8-CROSS is given in Figure 2.6.

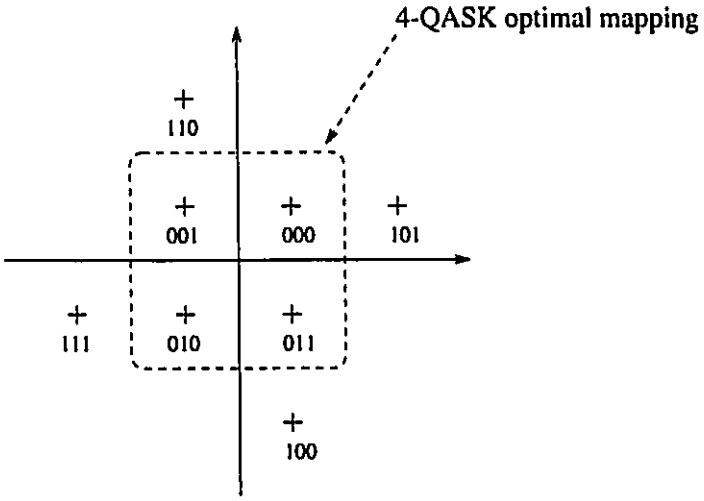

Figure 2.6: Optimum mapping for an 8-CROSS constellation.

2.3.3 16-QASK optimum mapping

The number of uncoded in-bits is still an unknown variable: the corresponding subsets could be the ones obtained after either the first, the second or the third construction

step, depending on whether the number of uncoded bits is 1 (2 subsets of 8 points), 2 (4 subsets of 4 points) or 3 (8 subsets of 2 points). Again, the same extension rule is applied: whenever a point is inside the 8-CROSS mapping shape, it must match its three first bits $y^{(0)}$, $y^{(1)}$ and $y^{(2)}$ with the 8-CROSS mapping and its fourth bit $y^{(3)} = 0$, whereas the outside points must have its fourth bit as $y^{(2)} = 1$.

Figure 2.7 illustrates the construction of a 16-QASK mapping, extending the previous 8-CROSS optimum one.

For this construction, the minimum distance between the points in the two first subsets is:

$$\Delta_1 = \sqrt{2}\Delta_0$$

the minimum distance between the points in the next four subsets is:

$$\Delta_2 = \sqrt{4}\Delta_0$$

and the minimum distance between the points in the last eight subsets is:

$$\Delta_3 = \sqrt{8}\Delta_0$$

Thus, the minimum distances Δ_2 or Δ_3 are maximized, and the resulting mapping is optimum at the first order for each type of partition.

The last step is not shown: using the previous rule, the fourth bit $y^{(3)}$ of a point is assigned to 0 (or 1) according to whether it is inside the 8-CROSS shape (or outside), in order to match this 16-QASK mapping to the former 8-CROSS mapping.

Figure 2.8 shows the final optimum mapping for a 16-QASK.

2.3.4 Conclusion

It has been shown that it is possible to construct an optimum mapping for TCM using 16-QASK in such a way that by puncturing it by one bit an optimum 8-CROSS constellation results. Puncturing again yields the optimum 4-QASK constellation. This is a useful result: whatever the number of bits that each symbol carries, the mapped signal is obtained using only one table. It is now worth determining whether or not this can be done for the 32-CROSS and 64-QASK constellations.

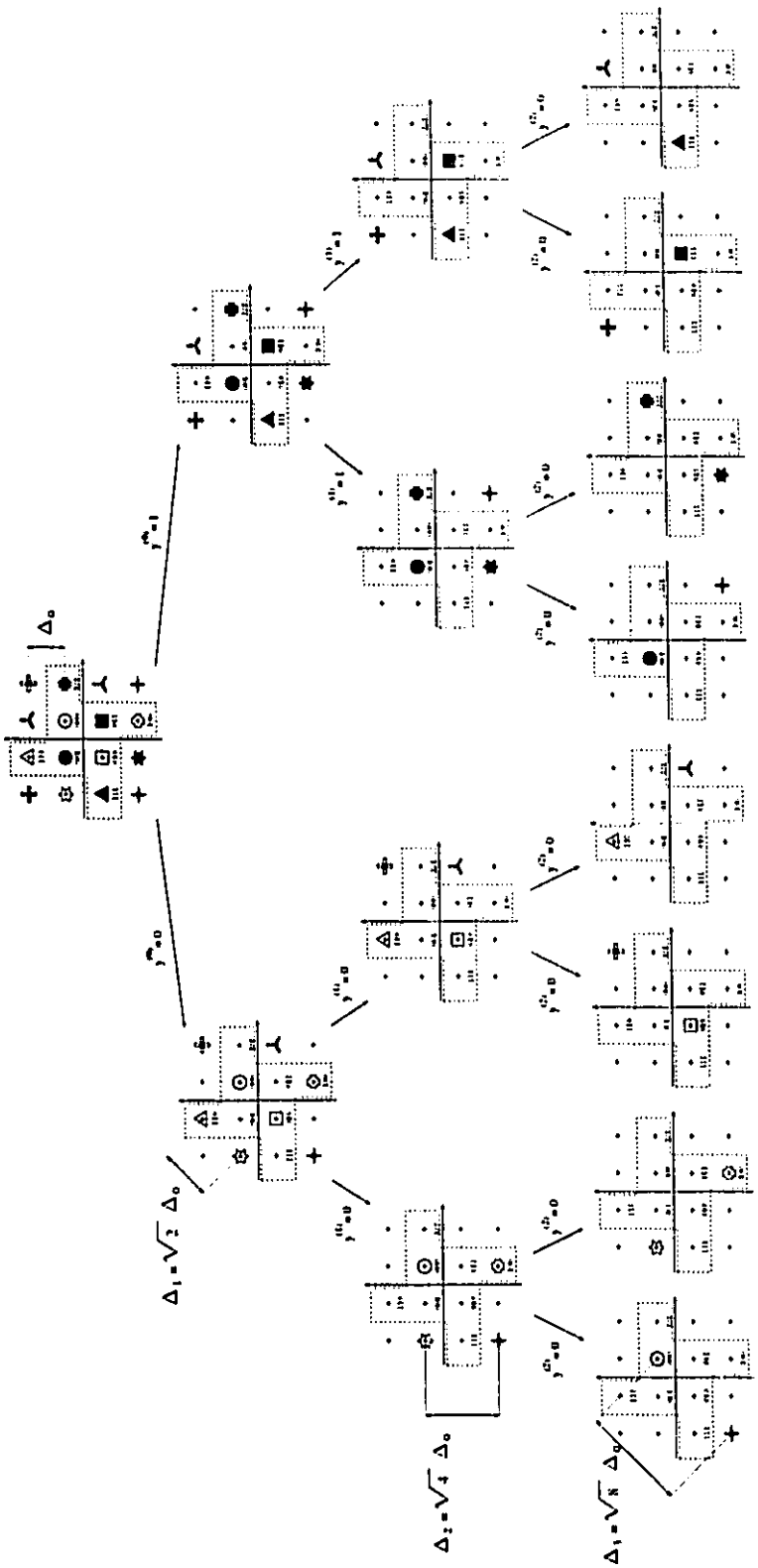

Figure 2.7: Construction of a 16-QASK mapping by extending the 8-CROSS optimum mapping.

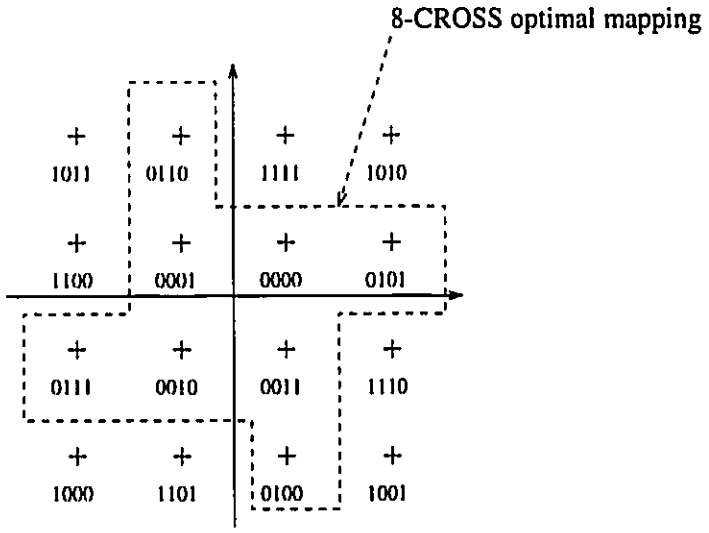

Figure 2.8: Optimum mapping for 16-QASK.

2.4 32-CROSS mapping

Consider now the problem of extending the 16-QASK optimum mapping to a 32-CROSS one.

2.4.1 32-CROSS mapping as an extension of the optimum 16-QASK

The first two steps in the construction of a 32-CROSS mapping are straightforward. Figure 2.9 represents this construction. It results in four subsets with identical structures. As the complete representation of such a construction requires too much space, and since these four subsets lead to similar group structures, only the first subset, corresponding to $y^{(1)}y^{(0)} = 00$, will be dealt with here.

The other cases can be treated in order to get the final extended mapping given in Figure 2.11. They are not reported here.

It appears in Figure 2.10 that there is no improvement in the performance of such a mapping used with 1 (3 uncoded in-bits) rather than 2 (2 uncoded in-bits) coded in-bits, since the minimum distance from the third step to the last one remains

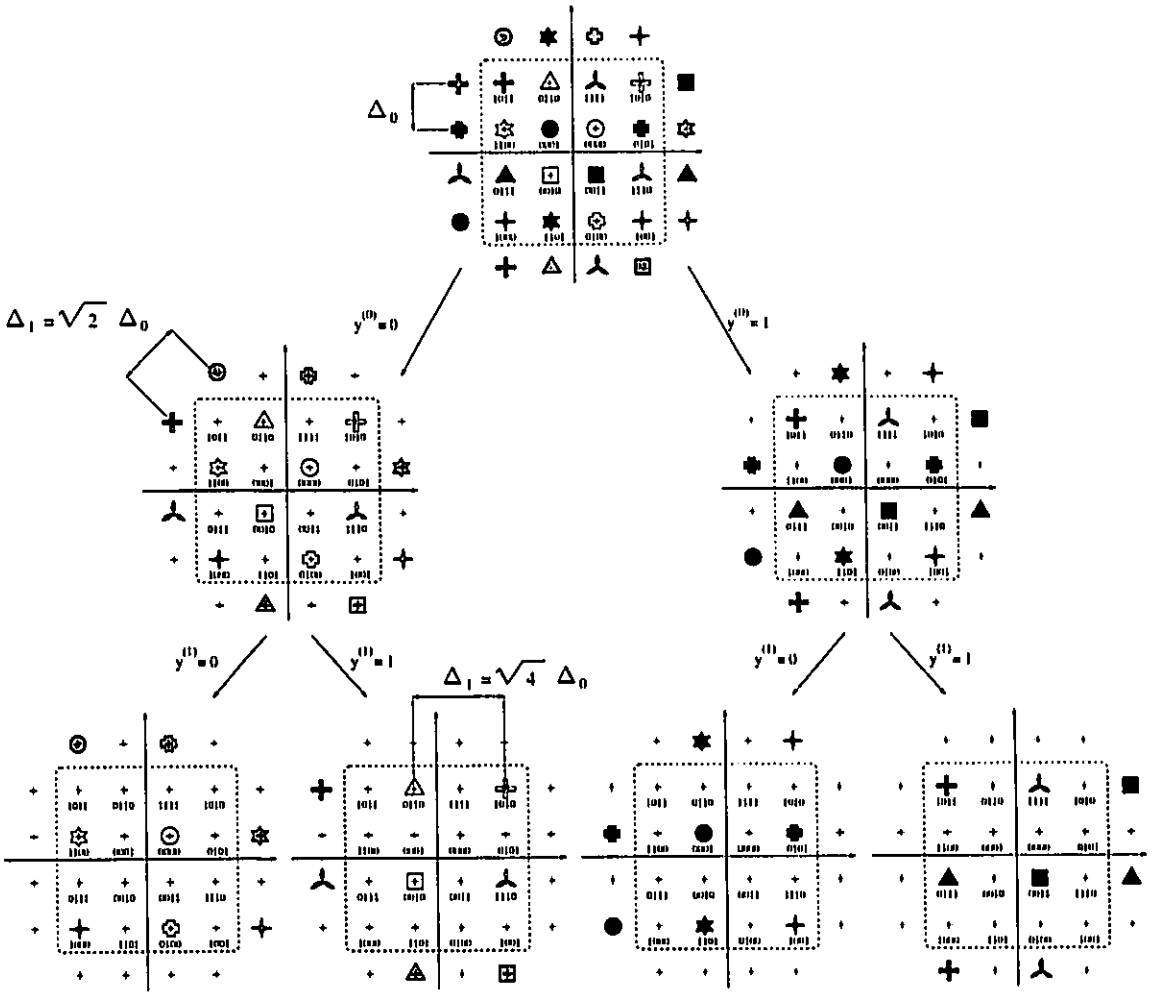

Figure 2.9: Construction of a 32-CROSS mapping by extending the previous 16-QASK optimum one: only the first two construction steps are shown.

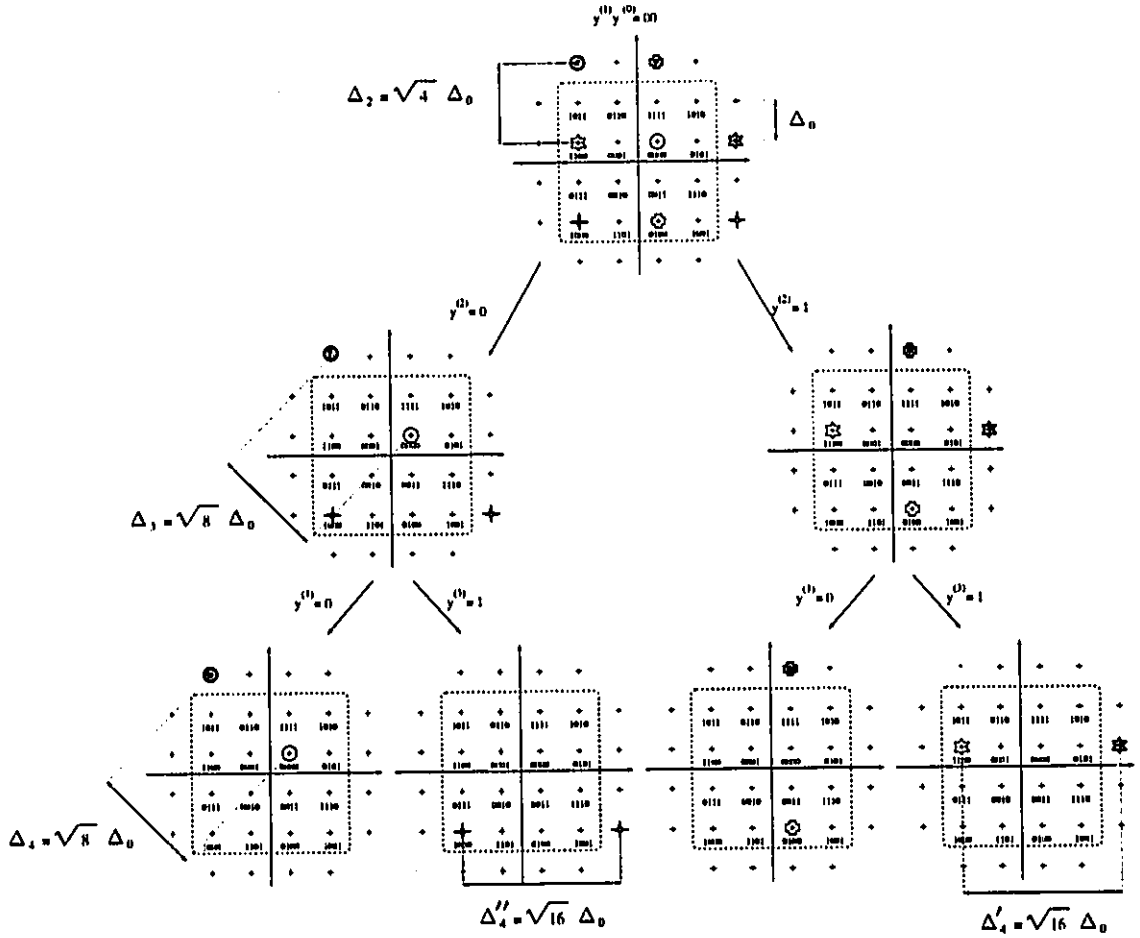

Figure 2.10: Construction of a 32-CROSS mapping by extending the previous 16-QASK optimum one: the next two steps for the first sub-constellation are given.

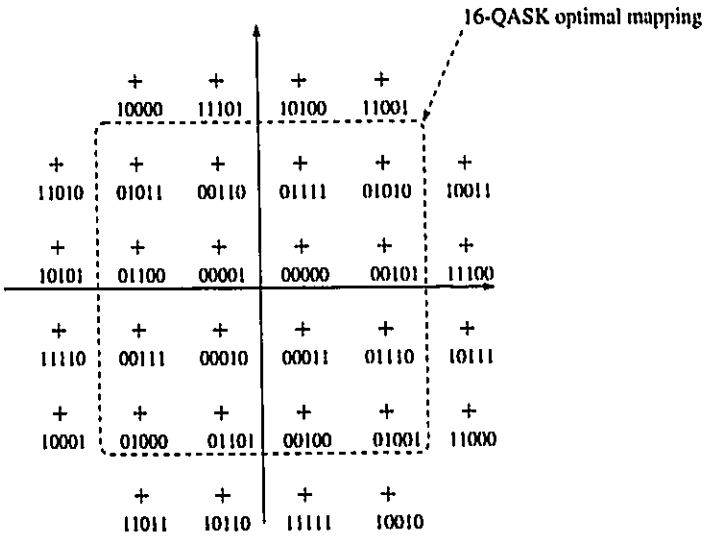

Figure 2.11: Extended 32-CROSS mapping from the 16-QASK optimum mapping.

constant, i.e.:

$$\Delta_4 = \Delta_3 = \sqrt{8}\Delta_0$$

This occurs because of a different structure for subsets $y^{(2)}y^{(1)}y^{(0)}$: when $y^{(2)}$ is 1, the four points form a square. The next step would separate the opposite corners and increase the minimum distance Δ_3 to:

$$\Delta_4' = \sqrt{16}\Delta_0$$

which is the second minimum distance at this level of construction of subsets. This second minimum distance cannot be increased and the extended 32-CROSS mapping is then optimum at the second order.

On the other hand, when $y^{(2)}$ is 0, the distance from the most central point to the three others is Δ_3, and there is no way Δ_4 can be increased. However, the extension rule makes the second derived distance from the $y^{(2)}y^{(1)}y^{(0)} = 000$ subset

$$\Delta_4'' = \sqrt{16}\Delta_0$$

rather than $\sqrt{32}\Delta_0$, as it would have been the case if the new vector $y^{(3)}y^{(2)}y^{(1)}y^{(0)} = 1000$ had consisted of the two points which are outside the 16-QASK shape on the

$y^{(2)}y^{(1)}y^{(0)} = 000$ subset. Therefore, the extension rule for this 32-CROSS mapping is not optimum at the third order.

The same conclusion is reached for the 3 other subsets corresponding to $y^{(1)}y^{(0)} =$ 01, 10, 11 since each of them are obtained from $y^{(1)}y^{(0)} = 00$ through a rotation by π, $\frac{\pi}{2}$ and $-\frac{\pi}{2}$.

2.4.2 32-CROSS optimum mapping at the third order

In this section, we consider the last point and design the signal mapping in such a way as to get a third minimum distance (Figure 2.12):

$$\Delta_4'' = \sqrt{32}\Delta_0$$

Otherwise, Figure 2.12 is quite similar to Figure 2.10 except that it does not contain any 16-QASK bits-to-symbols corresponding to an extension constraint.

In the final optimum 32-CROSS mapping represented in Figure 2.13, the four pairs of points that have been inverted from the optimum mapping at the second order only to the optimum mapping at the third order order are pointed out. This second 32-CROSS mapping improves the previous one at the second order, when the 32 points constellation is divided into 16 sub-constellations of 2 points each. This would occur if within the 4 in-bits, 3 were used by the convolutional encoder to compute the redundant coded out-bit, and the last one would then cause the existence of two parallel transitions on the trellis code. However, the Δ_4 distance would at the first order remains equal to $\sqrt{8}\Delta_0$.

Figure 2.13 represents the 32-CROSS optimum mapping. The four arrows indicate the rotation corresponding to the four subsets identified by $y^{(1)}y^{(0)}$.

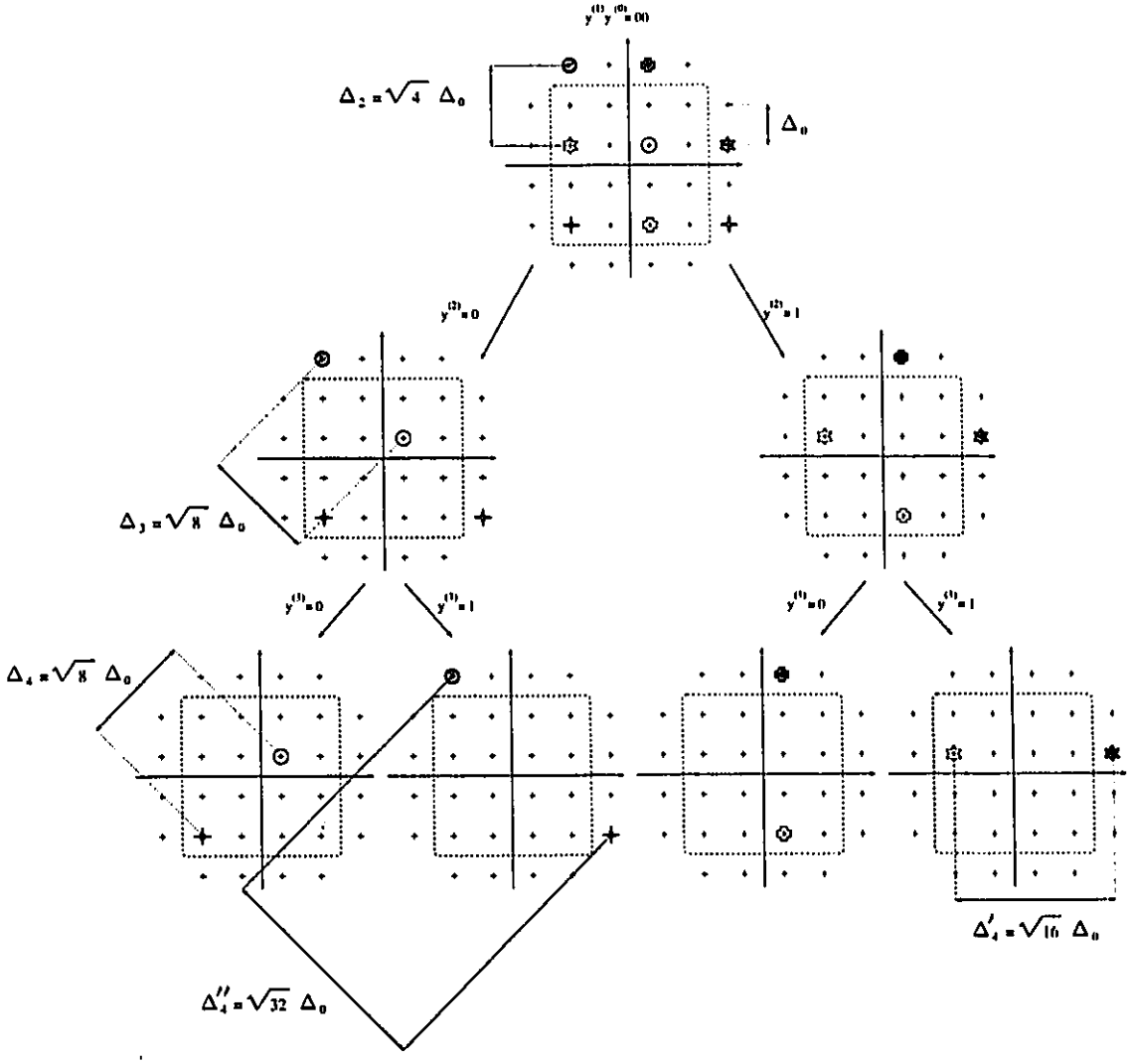

Figure 2.12: Construction of the 32-CROSS optimum mapping at the third order.

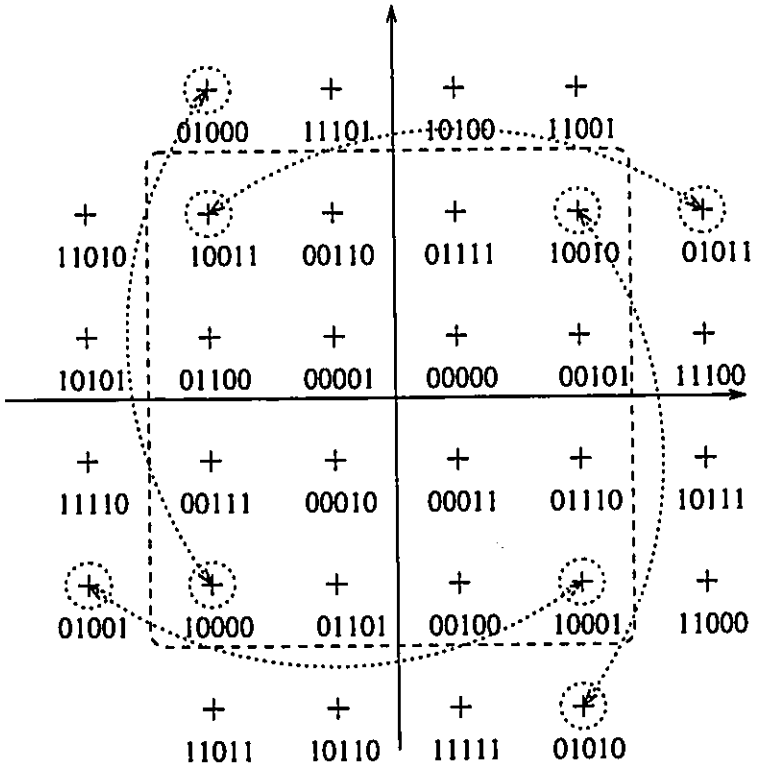

Figure 2.13: 32-CROSS optimum mapping.

2.5 64-QASK mapping

2.5.1 Extension of the optimum 32-CROSS mapping at the first order

The first step of this new construction is provided in Figure 2.14. In this Figure, the first subset $y^{(0)} = 0$ is treated up to the third step. The second subset $y^{(0)} = 1$ is presented on Figure 2.15. Observation of both Figure 2.14 and Figure 2.15 provides the values of the minimum distances:

$$\Delta_1 = \sqrt{2}\Delta_0$$

$$\Delta_2 = \sqrt{4}\Delta_0$$

$$\Delta_3 = \sqrt{8}\Delta_0$$

These first three steps end up with 8 subsets of 8 points. They are of two different types: on all the ones where $y^{(2)}$ is equal to 1, the four points inside the 32-CROSS shape form a square. On all zero $y^{(2)}$ subsets, the four inside points do not form a square, and will certainly be a source of failure in the optimization of the construction of a 64-QASK mapping under the extension rule constraint: they are positioned the same way as when constructing 32-CROSS mapping for which the extension rule failed in the optimization attempt. Figure 2.16 shows only the two subsets $y^{(2)}y^{(1)}y^{(0)} = 000$ and $y^{(2)}y^{(1)}y^{(0)} = 100$, which are different types.

Figure 2.16 shows the values of the minimum distances for subsets having 4 and 2 points respectively:

$$\Delta_4 = \Delta_3 = \sqrt{8}\Delta_0$$

$$\Delta_5 = \sqrt{32}\Delta_0$$

This illustrates the way limitations in the performance of a QASK mapping are transmitted to a larger one through extension matching: from the included 32-CROSS mapping, $\Delta_4 = \Delta_3$ cannot be avoided.

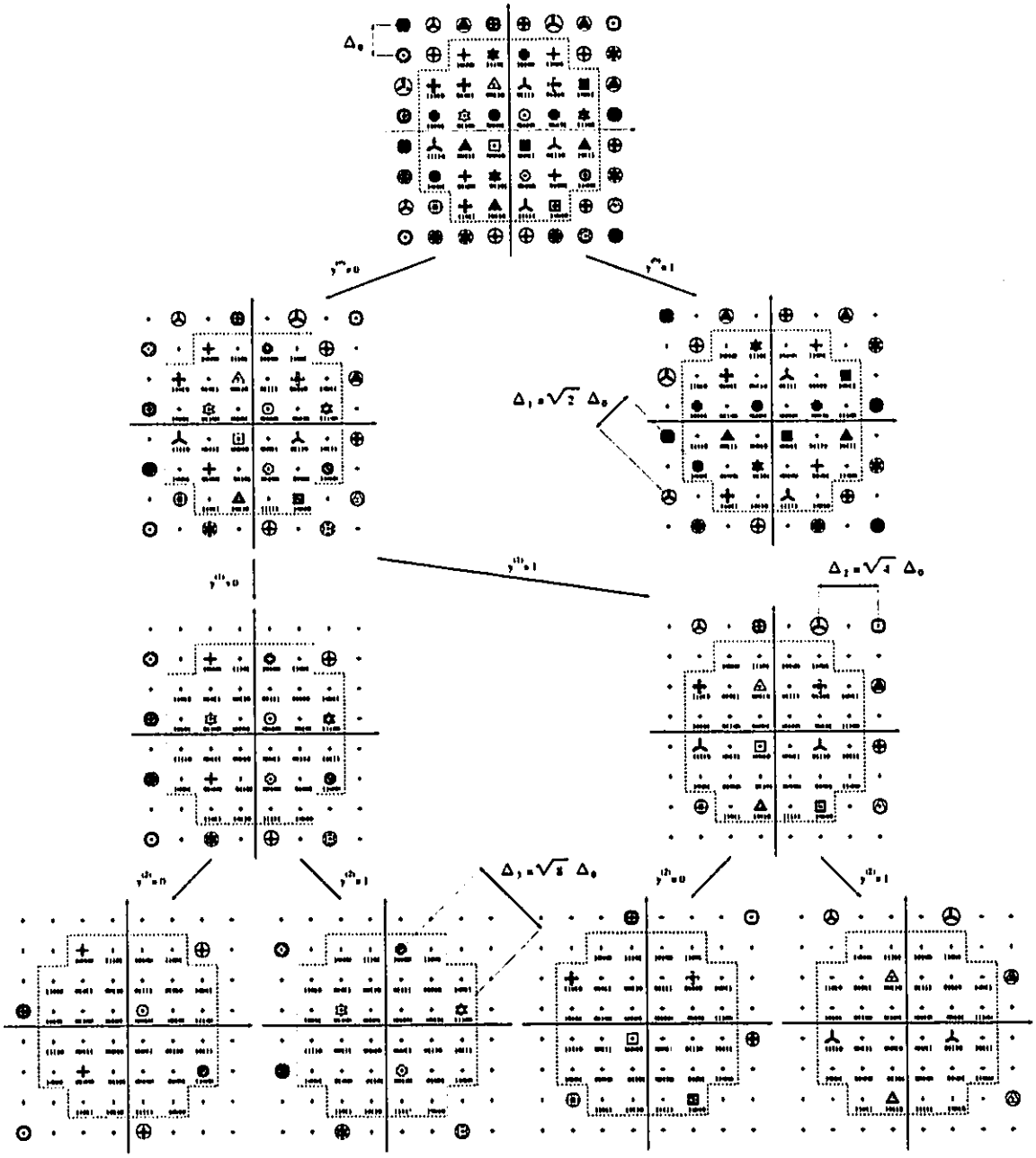

Figure 2.14: Construction of a 64-QASK mapping by extending the optimum 32-CROSS mapping at the first order: first step and next two steps for the first sub-constellation.

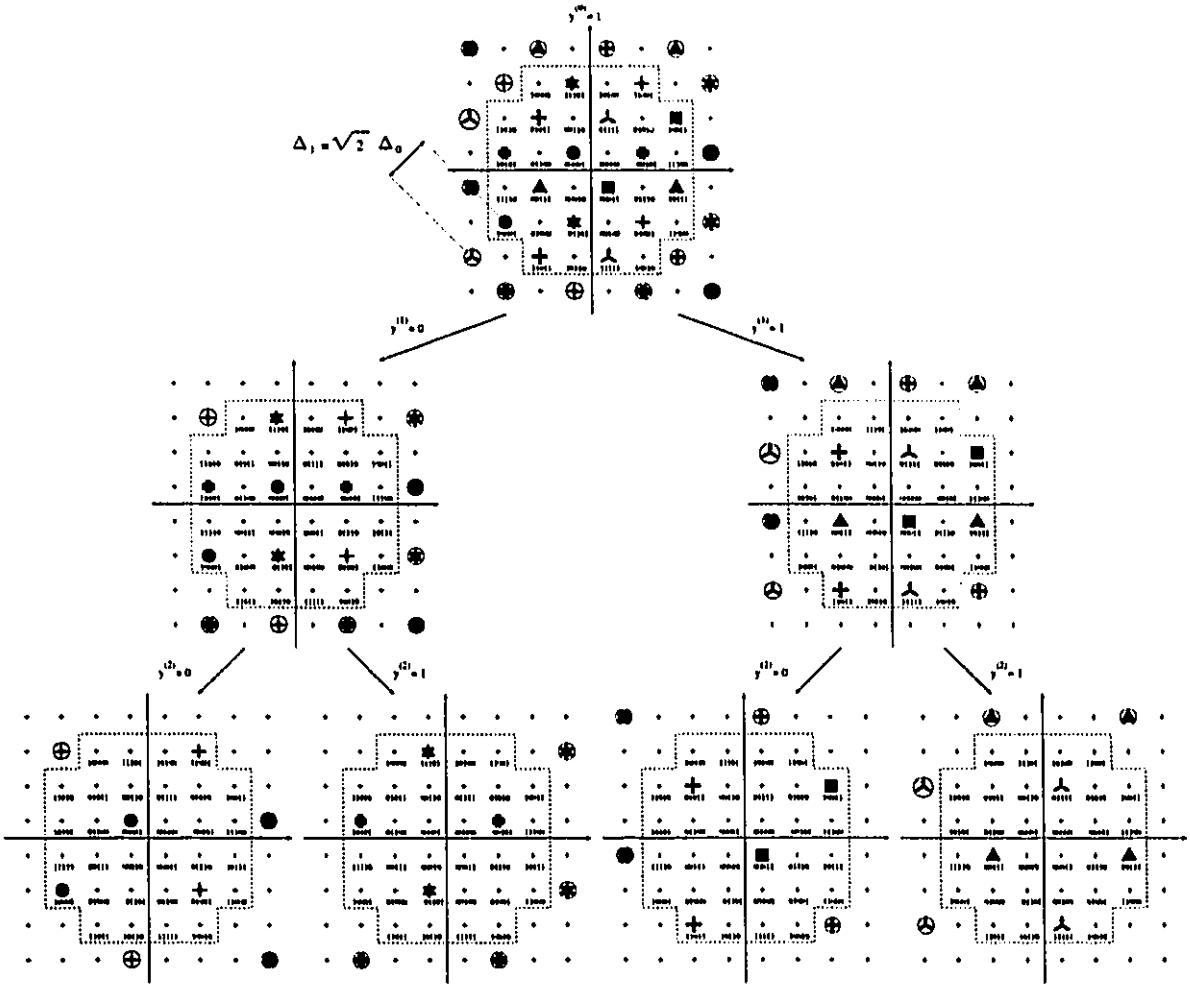

Figure 2.15: Construction of a 64-QASK mapping by extending the optimum 32-CROSS mapping at the first order: next two steps for the second sub-constellation.

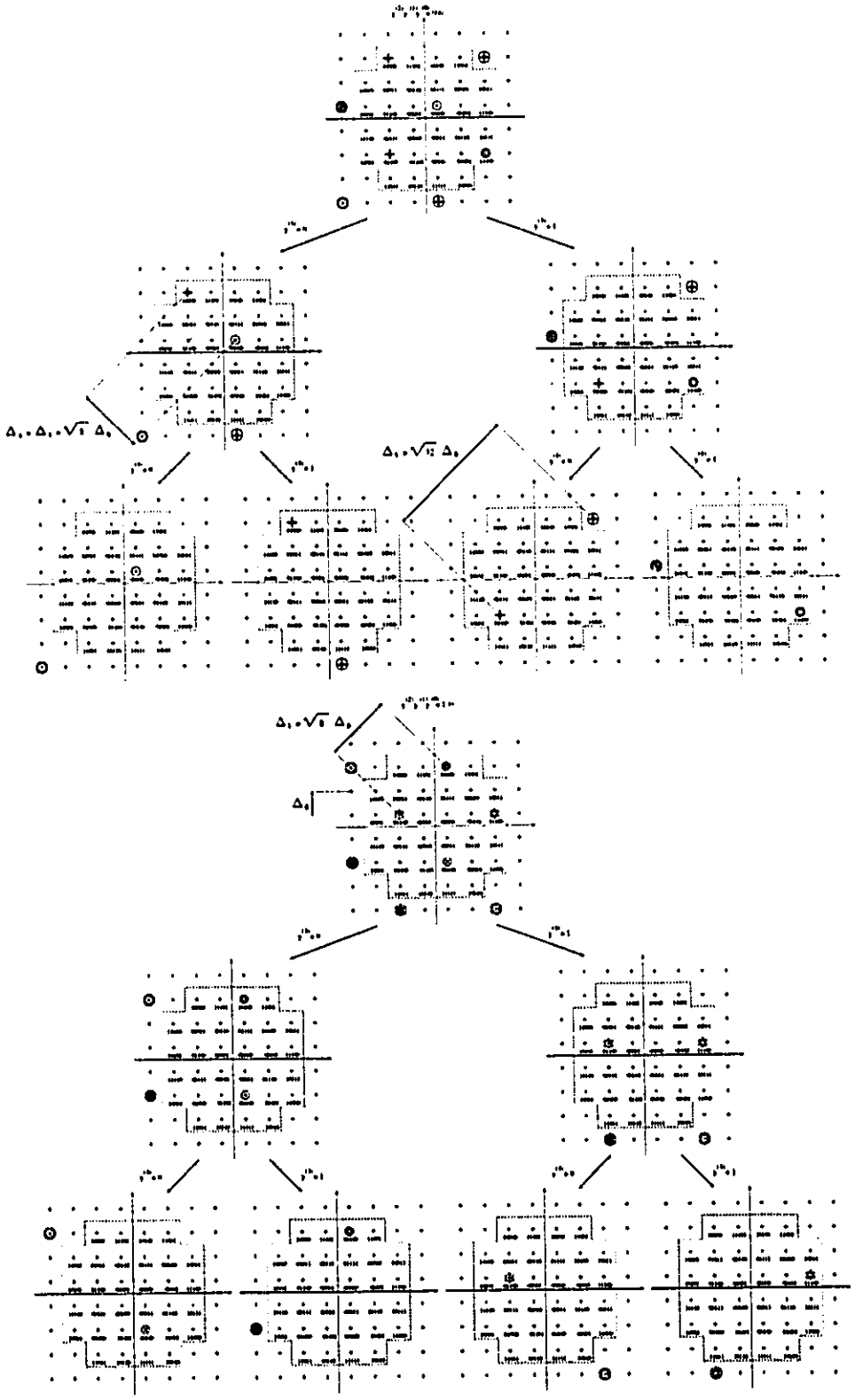

Figure 2.16: Construction of a 64-QASK mapping by extending the optimum 32-CROSS mapping at the first order: next three steps for the two first sub-constellations obtained in first part.

Again, it is observed that there is no great improvement using 4 coded in-bits (2 uncoded in-bits and 16 subsets of 4 points) rather than 3 coded in-bits (3 uncoded in-bits and 8 subsets of 8 points) since $\Delta_4 = \Delta_3$.

The resulting mapping is optimum at the first order only. It is depicted in Figure 2.17.

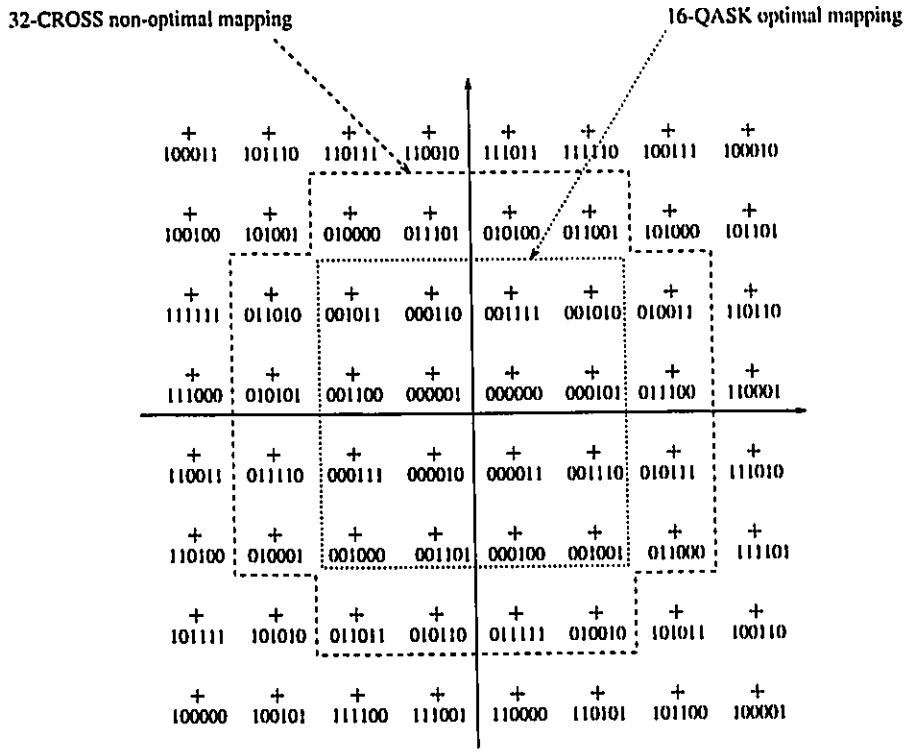

Figure 2.17: Extended 64-QASK mapping from the 32-CROSS optimum mapping at the first order.

2.5.2 Extension of the optimum 32-CROSS mapping at the third order

This new mapping has the exact same construction until the 8 first subsets of 8 points are obtained. That is, we can derive the structure subsets of 4 and 2 points from any two sub-constellations of type $y^{(2)} = 0$ and $y^{(2)} = 1$. For instance, on subset $y^{(2)}y^{(1)}y^{(0)} = 000$ derived in Figure 2.14, the old 32-CROSS optimum mapping (at the

first order) was simply replaced with the 32-CROSS optimum mapping at the third order, from which we obtain the expected subsets, as depicted in Figure 2.18.

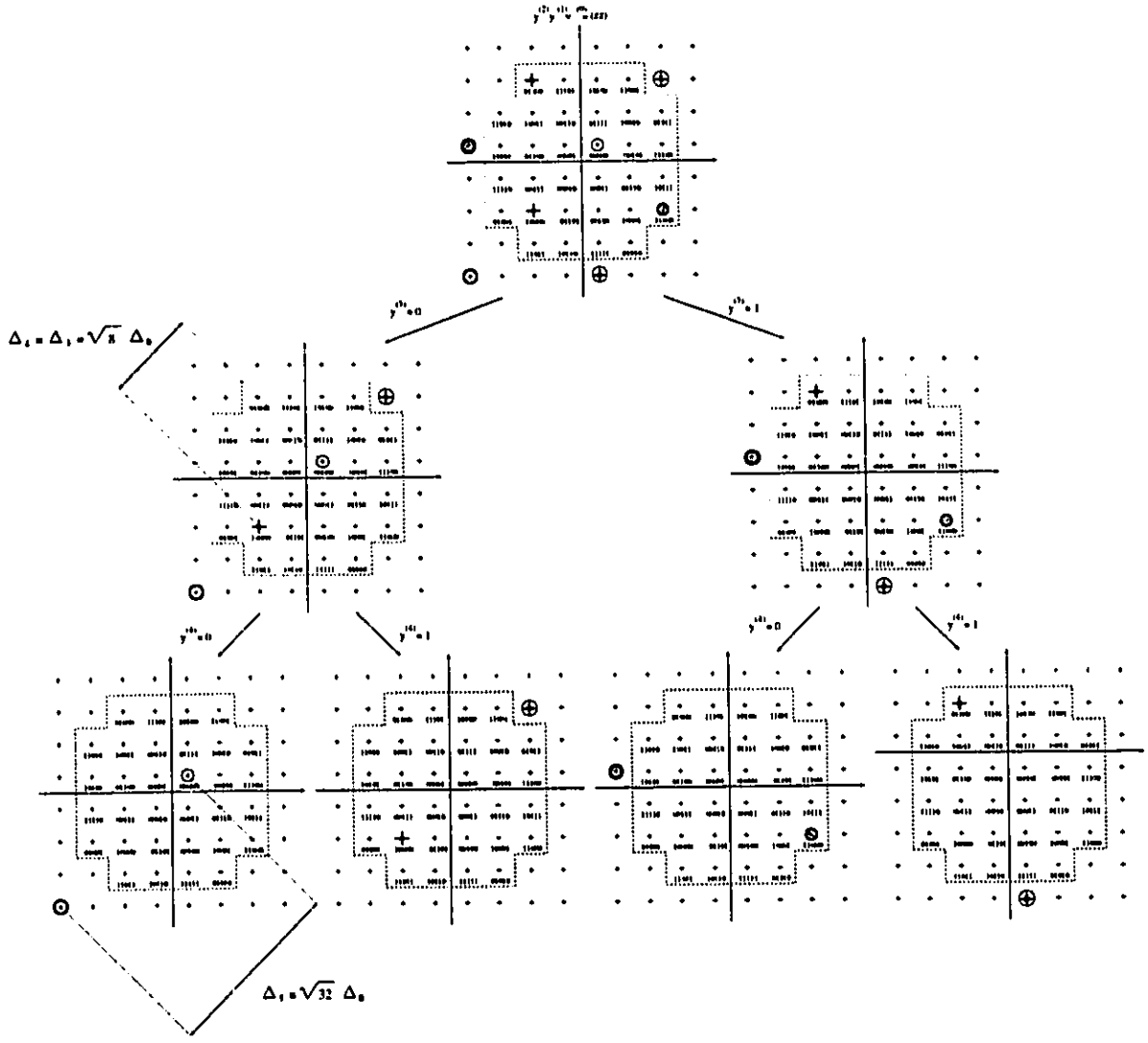

Figure 2.18: Construction of a 64-QASK mapping by extending the optimum 32-CROSS mapping at the third order: next three steps for the first sub-constellation.

Subset $y^{(2)}y^{(1)}y^{(0)} = 100$ gives the same grouping as previously obtained. The new mapping is given on Figure 2.19.

Within the 8 pairs of points which have been inverted from the previous mapping and are pointed out in the last figure, 4 are contained in the 32-CROSS shape and have already been brought out, whereas the others stay outside and reflect the consequent

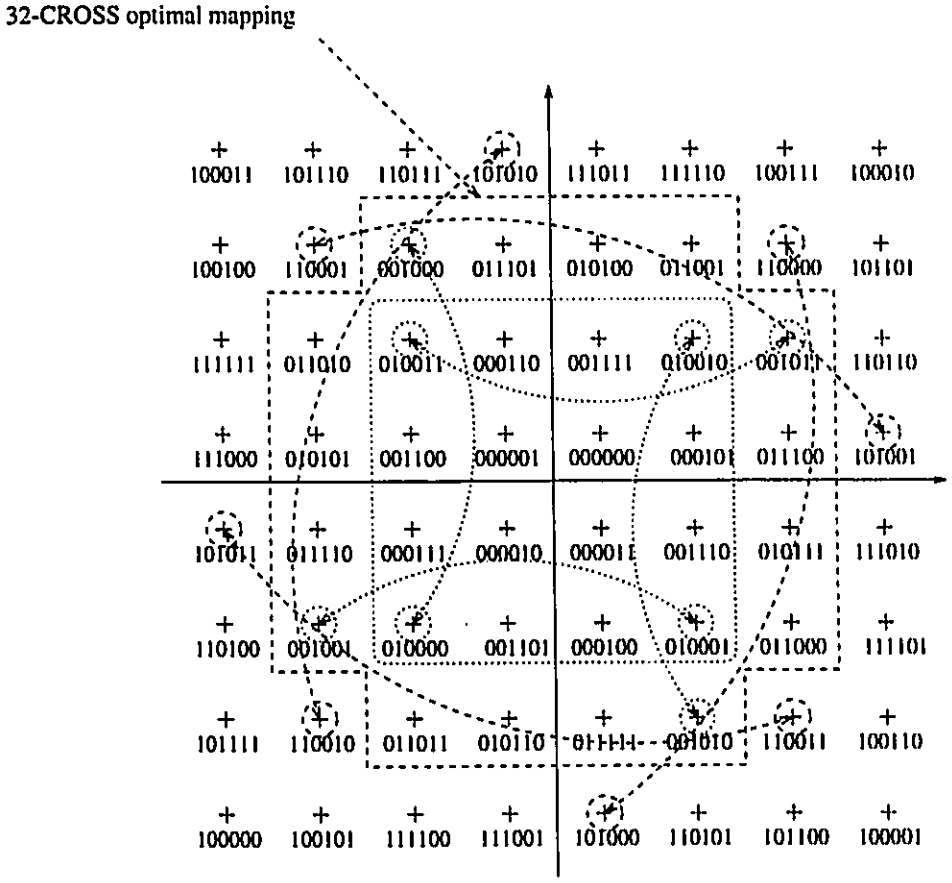

Figure 2.19: Extended 64-QASK mapping from the 32-CROSS optimum one.

changes in this new mapping.

Recalling the way the optimum 32-CROSS mapping at the third order was derived after the optimum 32-CROSS at the first order only, a careful analysis of the former subset $y^{(2)}y^{(1)}y^{(0)} = 000$ and indicates that even if the optimum 32-CROSS is taken to construct a 64-QASK extended mapping, no improvement could be achieved. This is because the central point in subset $y^{(2)}y^{(1)}y^{(0)} = 000$ is distant from the three others which are contained inside the 32-CROSS shape at a distance Δ_3. Hence, the only way to improve the 64-QASK mapping requires the grouping of this central point with the three ones located on the contour line of the 64-QASK constellation. This makes the task of deriving an optimum 64-QASK at the first order mapping from any smaller optimum 32-CROSS mapping impossible. It proves that the mapping of Figure 2.19 is not optimum at the first order when the 64 point constellation is partitioned into 16 subsets of 4 points, that is if among the 5 in-bits, 2 are uncoded and 3 are convolutionally encoded.

2.5.3 64-QASK optimum mapping

Figure 2.20 shows the construction of a 64-QASK optimum mapping. It only deals with the first three steps whereas Figure 2.21 completes the study and only deals with the first two subsets $y^{(2)}y^{(1)}y^{(0)} = 000$ and $y^{(2)}y^{(1)}y^{(0)} = 100$. It helps to show that this construction could arrange a nicer grouping for subset $y^{(2)}y^{(1)}y^{(0)} = 000$, closer to the one that has always been previously developed for subset $y^{(2)}y^{(1)}y^{(0)} = 100$:

$$\Delta_3 = \sqrt{8}\Delta_0$$

$$\Delta_4 = \sqrt{16}\Delta_0$$

$$\Delta_5 = \sqrt{32}\Delta_0$$

This can only be accomplished by eliminating the extension rule constraint.

The optimum 64-QASK mapping is drawn in Figure 2.22. It is important to point out that the last 64-QASK optimum mapping could match the previous 16-QASK

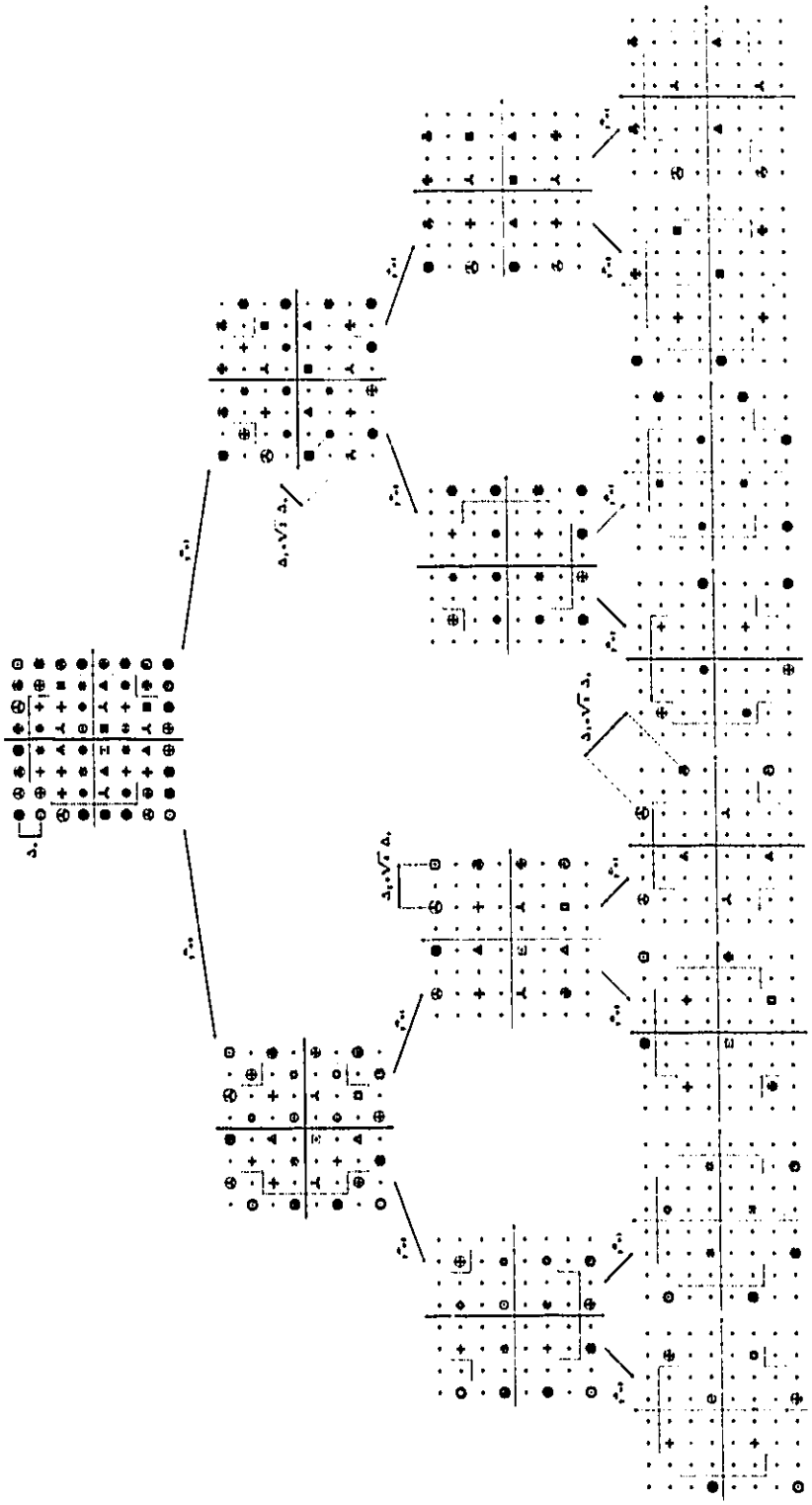

Figure 2.20: Construction of a 64-QASK optimum mapping: first three steps.

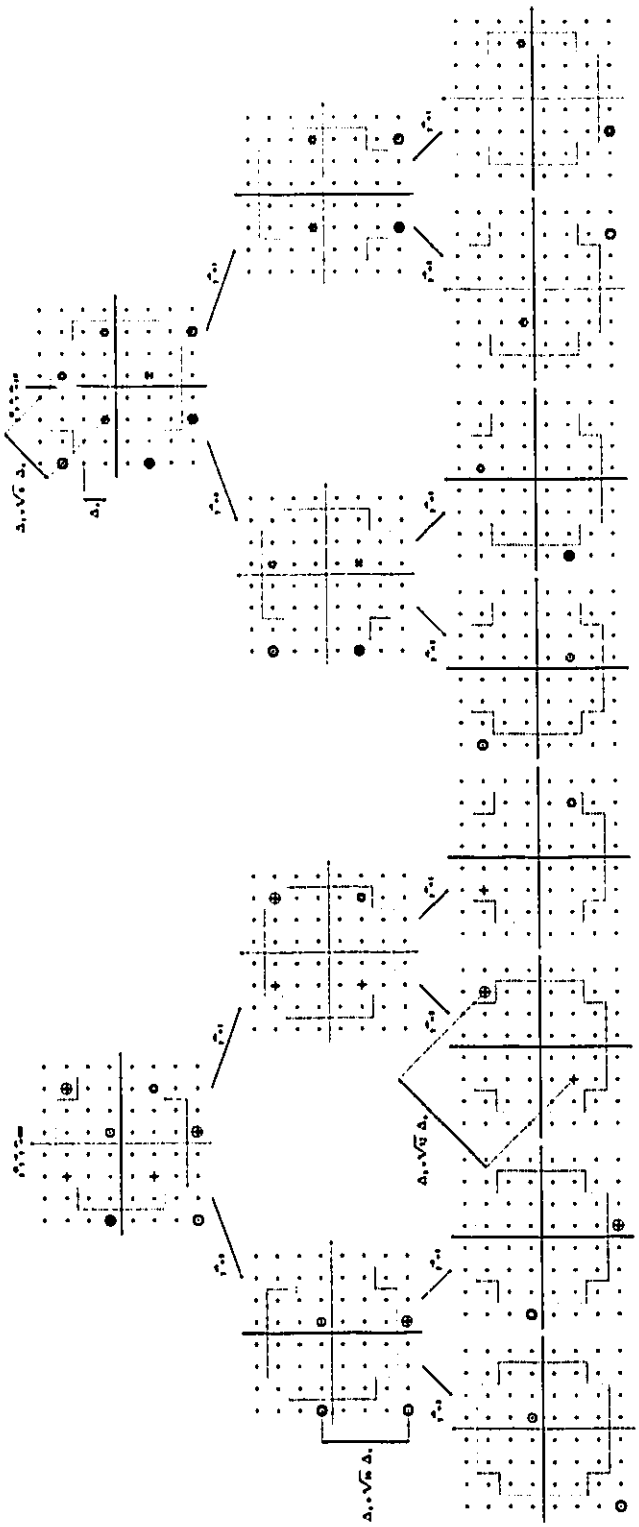

Figure 2.21: Construction of a 64-QASK optimum mapping: next three steps for the two first sub-constellations obtained in first part.

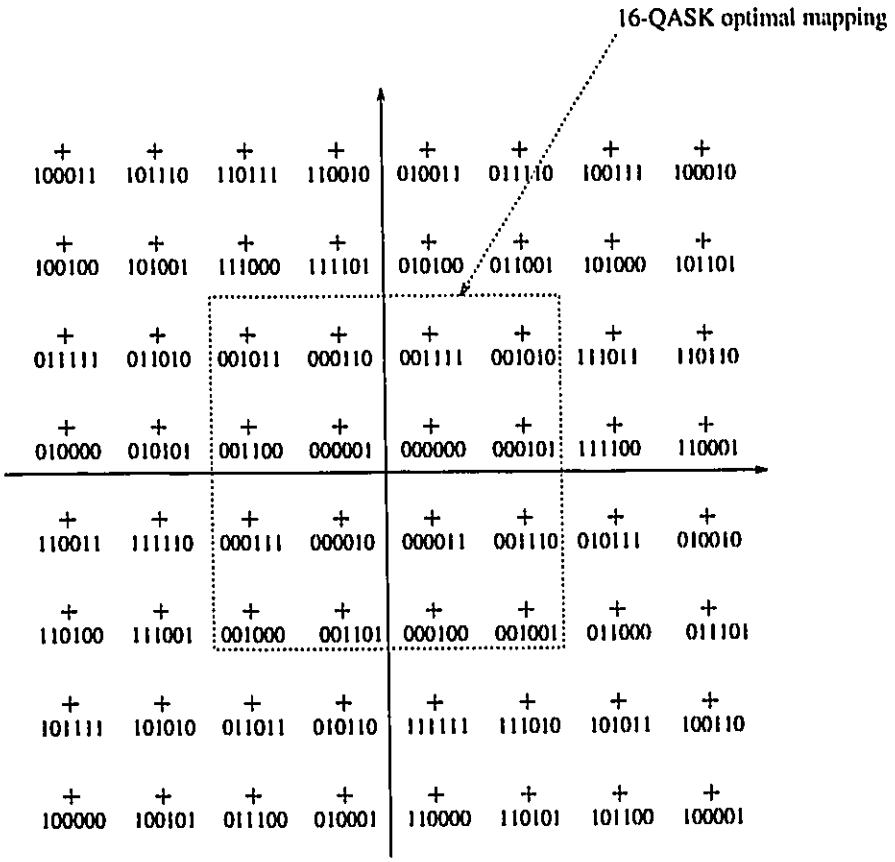

Figure 2.22: 64-QASK optimum mapping.

optimum mapping: points contained inside the 16-QASK shape have their two last bits $y^{(5)}$ and $y^{(6)}$ equal to 0.

2.6 Optimization inside a sub-constellation

This systematic construction method stands until the complete constellation has been divided into the appropriate number of subsets. Recall that when m denotes the number of in-bits and k denotes the number of coded in-bits (bits entering the convolutional encoder). Then, in a 2^{m+1} points constellation, there are 2^{k+1} subsets of 2^{m-k} points. Inside a particular subset, there are $m-k$ uncoded in-bits: there is no constraint on them, since they induce no effect on the occurrence of an event error. If the demodulator erroneously decides that a received symbol is from the same subset as the transmitted one, this error is transparent to the trellis decoding, say with the Viterbi decoder. This corresponds to an error in the parallel transitions. Thus, inside a subset, the performance is the same as with uncoded QAM. Hence, to ensure an optimum distribution of symbols on the points of such a subset, other techniques are needed. For instance, we may refer to the Gray code and the Hamming distance. However, this would only improve the probability of bit error and would simply reduce it by a constant factor.

As an example, consider the 32-QAM constellation with 2 parallel transitions on the trellis. The 8 subsets of 4 points are of two types. Some form a square, while others do not. Within all the possible sub-mappings, under the symmetry properties of the two kinds of subsets, we select an optimum one such that the lowest corresponding binary symbols remain unchanged from the previous complete optimum mapping. That is, if any change is needed inside a sub-constellation to improve the mapping, we first try to keep $y^{(4)}y^{(3)} = 00$ the same, then $y^{(4)}y^{(3)} = 10$, etc. Figure 2.23 represents the subsets $y^{(2)}y^{(1)}y^{(0)} = 100$ where, under each point, on the left, the mapping induced by the complete previous optimization for the two last bits $y^{(4)}y^{(3)}$ may be seen, while, on the right, a better one is given, that we could call the Gray

code mapping.

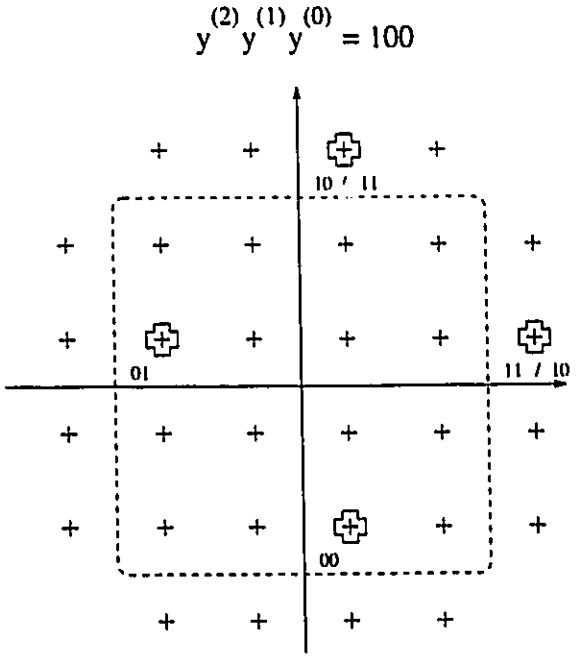

Figure 2.23: Optimizing the mapping inside the 4 points subset $y^{(2)}y^{(1)}y^{(0)} = 100$ of a 32-QAM mapping.

Figure 2.24 corresponds to the subset $y^{(2)}y^{(1)}y^{(0)} = 000$. The 6 possible distances are:

$$\sqrt{8}\Delta_0, \sqrt{8}\Delta_0, \sqrt{8}\Delta_0, \sqrt{16}\Delta_0, \sqrt{16}\Delta_0, \sqrt{32}\Delta_0$$

Two of them will separate two points with associated symbols of Hamming distance 2, the four other two points with symbols of Hamming distance 1. On Figure 2.24, the left symbols are the ones induced by the complete previous optimum mapping; the right one is the other possibility. For each of them, there are exactly two pairs of symbols with Hamming distance 1 and one pair with Hamming distance 2 at Euclidean distance $\sqrt{8}\Delta_0$; at least one pair of symbols with Hamming distance 1 at Euclidean distance $\sqrt{16}\Delta_0$. The difference concerns the Euclidean distances of the two other kinds of pairs of symbols: they have a Hamming distance of 1 and 2, and a Euclidean distance of exclusively either $\sqrt{16}\Delta_0$ or $\sqrt{32}\Delta_0$. The optimized one has the pair of symbols of Euclidean distance $\sqrt{16}\Delta_0$ with minimum Hamming distance 1. It

turns out to be the right one. This means that the complete previous optimization could also be improved for non rectangular sub-constellations.

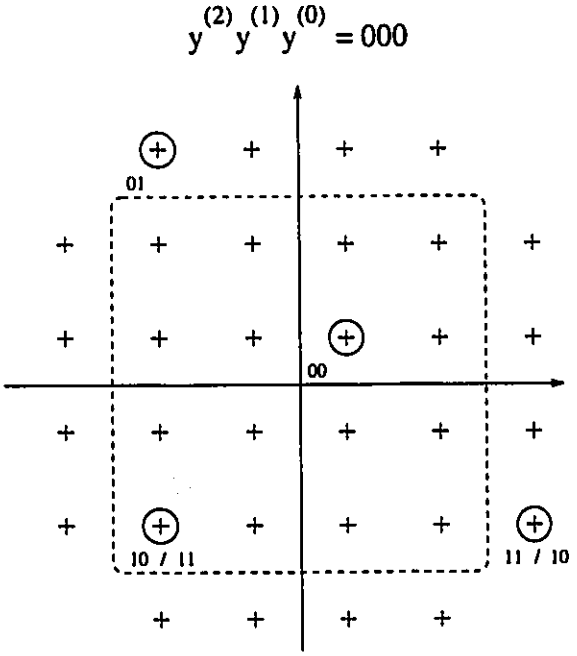

Figure 2.24: Optimizing the mapping inside the 4 points subset $y^{(2)}y^{(1)}y^{(0)} = 000$ of a 32-QAM mapping.

This particular selection of subsets was rather simple to deal with. For subsets of 8 points or more, this method may be more complex and long, though systematic. We decided not to investigate it any further. But this stresses the possibility to improve any mapping individually, for all possible sizes of sub-constellations. As a result, from now on, we realize that we do not use any real optimum mappings if we use the ones derived before when the subsets have 4 or more points. Later on, their theoretical performances will be evaluated and they will be used in computer simulations.

2.7 Conclusion

This investigation results in the difficulty to meet the two constraints of extension of a smaller mapping and the maximization of the minimum distances between the points

of the derived subsets. It appears that this difficulty is caused by the so called CROSS constellations, that is, with points which are not in a perfect square. It is likely that the 8-CROSS and 16-QASK optimized mappings matching is a coincidence. It is very probable that the 128-CROSS mapping would cause the same kind of incompatibility. Otherwise, any square constellation could match a smaller one. As a consequence, whenever TCM codes are to be used, the first condition is the selection of an optimum mapping, which may not be a simple extension of a smaller one.

Chapter 3

Theoretical performance evaluation for TCM codes

In this chapter, we first review the basic concepts of performance evaluation of trellis codes for TCM. The material in sections 3.1 and 3.2 are partly developed in [1]. In section 3.1, a classical expression for the probability of event error is derived. The description given in reference [1] did not appear to the author fully clear and detailed. Appendix A clarifies some points and justifies the "incomplete" steps. Also, Example 4.1 from [1, page 103] seems to be inappropriate: it consists of a 2 states encoder for which any pair of states has a transition, and only a single transition. In this thesis, we provide an example in section 3.1.2 which deals with an encoder with parallel transitions and pair of states with no transition between each other.

In section 3.2, we present the so-called *uniformity condition*, as developed in [1], which allows reduction for the complexity of the theoretical expression for the error probability for TCM. From observations of Example 3.1.2, the principle of this condition is explored and the proof is extended to the recursive Ungerboeck codes. Meanwhile, even if the *uniformity condition* (4.19) from [1, page 107] is sufficient, the proof is incomplete, and this defect can be observed through the same Example 3.1.2. Appendix B presents a proof of the *uniformity condition* in a very general case. In section 3.3 we verify the validity of this condition on the mappings derived

in chapter 2.

Sections 3.4 and 3.5 extend the previous theoretical probability of event error to bit and symbol errors, whereas section 3.6 provides improved error expressions based on tighter upperbounds. The development of section 3.5 has been found nowhere and thus no reference are cited in it.

Finally, section 3.7 analyzes the theoretical performances of TCM codes and section 3.8 explores several different limitations in order to derive a synthetic expression. A new point is the second order specification for expression of probability of errors.

3.1 First investigation

3.1.1 Probability of event error and transfer function in terms of the *error weight matrices* on the trellis of TCM code

For any convolutional encoder structure, let X_t, Y_t and S_t be the input symbol to the encoder, the output coded symbol of the encoder, and the corresponding signal at time t respectively. Denote by $\mathbf{S}_t^L = [S_t, S_{t+1}, \ldots, S_{t+L-1}]$ a sequence of L signals from time t to $t+L-1$, and $\tilde{\mathbf{S}}_t^L = \left[\tilde{S}_t, \tilde{S}_{t+1}, \ldots, \tilde{S}_{t+L-1}\right]$ another sequence such that:

$$\begin{cases} S_{t-1} = \tilde{S}_{t-1} \\ S_\tau \neq \tilde{S}_\tau \qquad \text{for } \tau = t, t+1, \ldots, t+L-1 \\ S_{t+L} = \tilde{S}_{t+L} \end{cases}$$

The one-to-one correspondence between an output symbol Y_t and its associated signal S_t implies that the two correspondent signal sequences $\mathbf{Y}_t^L$ and $\tilde{\mathbf{Y}}_t^L$ are the trellis outputs for of two sequences of transitions, that is, two paths, which started diverging at time t and merged later at time $t+L$. Meanwhile, an *event error* $\mathbf{E}_t^L \neq \mathbf{0}$ occurred.

The demodulator produces such an *error event* $\mathbf{E}_t^L$ when it chooses, instead of the transmitted sequence $\mathbf{S}_t^L$, a different one $\tilde{\mathbf{S}}_t^L \neq \mathbf{S}_t^L$. Under the assumption of a

sufficiently long transmission, we can disregard [1, page 100] the time dependence of this error probability. The probability for this time-independent error to occur is:

$$\mathcal{P}_e\{\mathbf{S}^L\} = \sum_{\tilde{\mathbf{S}}^L \neq \mathbf{S}^L} \mathcal{P}\{\tilde{\mathbf{S}}^L | \mathbf{S}^L\}$$

Since an event error may start from different states, or equivalently, from different received signals, we refer to the union bound to express the probability of event error as:

$$\begin{aligned} \mathcal{P}_e &= \sum_L \sum_{\mathbf{S}^L} \mathcal{P}\{\mathbf{S}^L\} \mathcal{P}_e\{\mathbf{S}^L\} \qquad (3.1) \\ \mathcal{P}_e &= \sum_L \sum_{\mathbf{S}^L} \sum_{\tilde{\mathbf{S}}^L \neq \mathbf{S}^L} \mathcal{P}\{\mathbf{S}^L\} \mathcal{P}\{\tilde{\mathbf{S}}^L | \mathbf{S}^L\} \\ &= \sum_L \left(\sum_{\substack{\mathbf{S}^L \neq \tilde{\mathbf{S}}^L \\ \mathbf{S}^L}} \mathcal{P}\{\mathbf{S}^L\} \mathcal{P}\{\mathbf{S}^L \rightarrow \tilde{\mathbf{S}}^L\} \right) \end{aligned}$$

where $\{\mathbf{S}^L \rightarrow \tilde{\mathbf{S}}^L\}$ represents the event of receiving $\tilde{\mathbf{S}}^L$ when $\mathbf{S}^L$ was actually transmitted.

Because of the one-to-one correspondence between output symbols and signals, any factor $\mathcal{P}\{\mathbf{S}^L\}$ may be replaced by $\mathcal{P}\{\mathbf{Y}^L\}$. Then, the Bhattacharyya bound is used to introduce pairwise probabilities of error, $\mathcal{P}\{\mathbf{Y}^L \rightarrow \tilde{\mathbf{Y}}^L\}$ which represent the integral of the noise over half of the two-dimensional space containing the decision region of $\tilde{\mathbf{Y}}^L$ instead of over its decision region itself:

$$\mathcal{P}_e \leq \sum_L \left(\sum_{\substack{\mathbf{Y}^L \neq \tilde{\mathbf{Y}}^L \\ \mathbf{Y}^L}} \mathcal{P}\{\mathbf{Y}^L\} \mathcal{P}\{\mathbf{Y}^L \rightarrow \tilde{\mathbf{Y}}^L\} \right) \qquad (3.2)$$

If $\oplus$ denotes the bit-by-bit modulo 2 addition, an index change $\mathbf{E}^L = \tilde{\mathbf{Y}}^L \oplus \mathbf{Y}^L$ in the last summation produces a new expression:

$$\mathcal{P}_e \leq \sum_L \sum_{\mathbf{E}^L \neq \mathbf{0}} \left(\sum_{\mathbf{Y}^L} \mathcal{P}\{\mathbf{Y}^L\} \mathcal{P}\{\mathbf{Y}^L \rightarrow \mathbf{Y}^L \oplus \mathbf{E}^L\} \right)$$

and under the long enough transmission assumption, the source symbols are independent and have equal probabilities, so that $\mathcal{P}\{Y_l\} = \frac{1}{N_\nu 2^m}$, since from any state,

they are 2^m equiprobable transitions and N_ν states. Meanwhile, the last factor may be rewritten with the use of the Bhattacharyya bound [1]:

$$\begin{aligned}
\mathcal{P}\{\mathbf{Y}^L \to \mathbf{Y}^L \oplus \mathbf{E}^L\} &= \mathcal{P}\{\mathbf{S}^L \to \tilde{\mathbf{S}}^L\} \\
&= \frac{1}{2}\operatorname{erfc}\left(\frac{\left\|\mathbf{S}^L - \tilde{\mathbf{S}}^L\right\|}{2\sqrt{N_0}}\right) \\
&\leq \frac{1}{2}\exp\left(-\frac{\left\|\mathbf{S}^L - \tilde{\mathbf{S}}^L\right\|^2}{4N_0}\right) \\
&= \frac{1}{2}\exp\left(-\frac{\sum_{l=1}^{L}\left\|S_l - \tilde{S}_l\right\|^2}{4N_0}\right) \\
&= \frac{1}{2}\prod_{l=1}^{L}\exp\left(-\frac{\left\|S_l - \tilde{S}_l\right\|^2}{4N_0}\right)
\end{aligned} \tag{3.3}$$

Therefore, introducing $Z = \exp\left(-\frac{1}{4N_0}\right)$:

$$\begin{aligned}
\sum_{\mathbf{Y}^L}\mathcal{P}\{\mathbf{Y}^L\}\mathcal{P}\{\mathbf{Y}^L \to \mathbf{Y}^L \oplus \mathbf{E}^L\} &\leq \frac{1}{2}\sum_{\mathbf{Y}^L}\mathcal{P}\{\mathbf{Y}^L\}\left(\prod_{l=1}^{L} Z^{\|f(Y_l)-f(Y_l\oplus E_l)\|^2}\right) \\
&\leq \frac{1}{2}\mathbf{W}(\mathbf{E}^L)
\end{aligned} \tag{3.4}$$

where $\mathbf{W}(\mathbf{E}^L)$ is a weight function defined by:

$$\mathbf{W}(\mathbf{E}^L) = \sum_{\mathbf{Y}^L}\mathcal{P}\{\mathbf{Y}^L\}\prod_{l=1}^{L}\left(Z^{\|f(Y_l)-f(Y_l\oplus E_l)\|^2}\right) \tag{3.5}$$

Equation (3.2) finally becomes:

$$\mathcal{P}_e \leq \frac{1}{2}\sum_{L}\sum_{\mathbf{E}^L\neq\mathbf{0}}\mathbf{W}(\mathbf{E}^L) \tag{3.6}$$

It is proved in Appendix A, that if $N_\nu = 2^\nu$ denotes the number of states of the code, $\mathbf{W}(\mathbf{E}^L)$ is equal to:

$$\mathbf{W}(\mathbf{E}^L) = \frac{1}{N_\nu}{}^{\mathrm{t}}\mathbf{1}\left(\prod_{l=1}^{L}\mathbf{G}(E_l)\right)\mathbf{1}\Big|_{D=Z} \tag{3.7}$$

where $\mathbf{1}$ denotes a vector of N_ν elements equal to 1, ${}^{\mathrm{t}}\mathbf{1}$ its transpose and the matrices

$\mathbf{G}(E_l)$, called *error weight matrices*, are functions of the variable D defined by:

$$[\mathbf{G}(E_l)]_{p,q} = \begin{cases} 0 & \text{if no transition from state } p \text{ to state } q \\ \frac{1}{2^m} \sum_{t_{p-q}} D^{\|J(Y_{t_{p-q}}) - J(Y_{t_{p-q}} \oplus E_l)\|^2} & \text{otherwise} \end{cases} \tag{3.8}$$

Then the probability of event error is such that:

$$\mathcal{P}_e \leq \frac{1}{2} \sum_{L=0}^{\infty} \sum_{E^L \neq 0} \frac{1}{N_\nu} {}^{\mathrm{t}}\mathbf{1} \left(\prod_{l=1}^{L} \mathbf{G}(E_L) \right) \mathbf{1} \Big|_{D=Z} = \frac{1}{2} \frac{1}{N_\nu} {}^{\mathrm{t}}\mathbf{1}\mathbf{G}\mathbf{1} \Big|_{D=Z} \tag{3.9}$$

where

$$\mathbf{G} = \sum_{L=0}^{\infty} \sum_{E^L \neq 0} \prod_{l=1}^{L} \mathbf{G}(E_l) \tag{3.10}$$

Since $\mathbf{G}(E_l)$ is a function of D, $\left(\frac{1}{N_\nu} {}^{\mathrm{t}}\mathbf{1}\mathbf{G}\mathbf{1}\right)$ is also a function of D and is called the *transfer function* of the code and denoted by $T(D)$.

$$T(D) = \frac{1}{N_\nu} {}^{\mathrm{t}}\mathbf{1}\mathbf{G}\mathbf{1} \tag{3.11}$$

An element (p, q) of the matrix $\mathbf{G}$ provides an upperbound on the probability of an error event that starts at state p and ends at state q, and takes in consideration error events of all length L. The sum of its elements over row p gives an upper bound on the probability of error event that starts at state p, whereas the sum of its elements over a column provides an upper bound for an error event which ends at state q.

An upper bound for the probability of error is then provided by:

$$\mathcal{P}_e \leq \frac{1}{2} T(D) \Big|_{D=Z=e^{-\frac{1}{4N_0}}} \tag{3.12}$$

In the Equality of (3.5), the dependency between the binary error vectors Y_l does not appear. Furthermore, because of the linearity of the code, the error event sequence $\mathbf{E}^L$, as a difference, or a sum, between two sequences of output symbols is also a sequence of outputs. Thus, it is possible to represent them on a state diagram related to the trellis code. The branches of such an error diagram are then matrices, and in [1] it is called the *error state diagram*, with sums of *error weight matrices* $\mathbf{G}(E)$ as labels for its branches, by opposition to the *code state diagram* with its *scalar weights.*

On the *error state diagram*, we denote by $\mathbf{G}_{ij}$ the label of the transition from state i to state j. This matrix label is the *error weight matrix* corresponding to the symbol error $E_{ij} = Y_{t_{i-j}}$: $\mathbf{G}(E_{ij}) = \mathbf{G}(Y_{t_{i-j}})$ where $Y_{t_{i-j}}$ represents the output symbol of the code through one transition $i \rightarrow j$ of the trellis code:

$$\mathbf{G}_{ij} = \sum_{t_{i-j}} \mathbf{G}(Y_{t_{i-j}}) \tag{3.13}$$

where $\mathbf{G}(Y_{t_{i-j}})$ is defined by Equation (3.8).

3.1.2 Example and other formulation

As an example of the use of the *error weight matrices*, consider the trellis of the simple Ungerboeck code ($n = 2, k = 1, \nu = 2$), shown in Figure 3.1, with $m = 2$ in-bits.

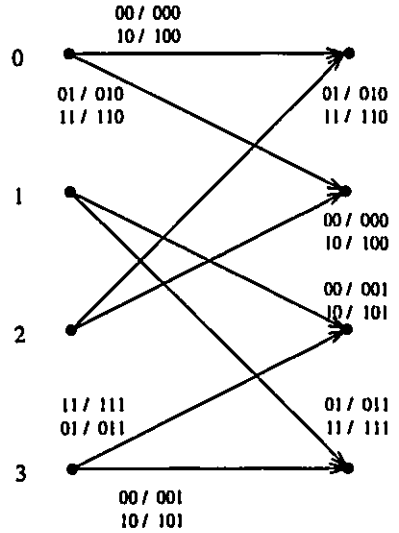

Figure 3.1: Trellis of TCM code $(2, 1, 2)$ with $m = 2$.

The corresponding *error state diagram*, represented in Figure 3.2, may be separated into two distinct ones, as shown by Figures 3.3 and 3.4.

Figure 3.3 represents the matrix of transition from state 1 to state 2 for any length

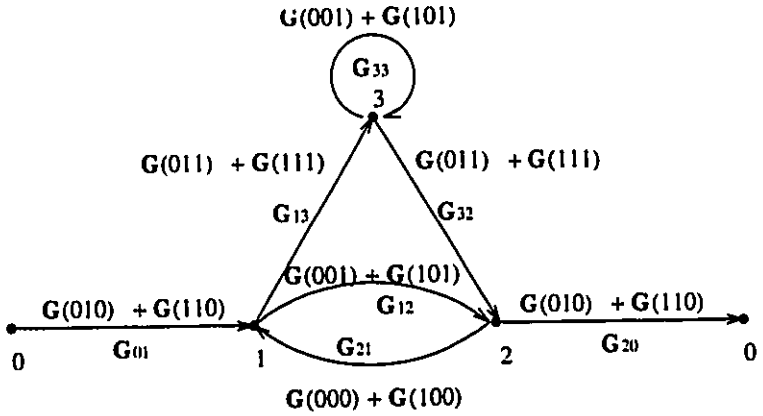

Figure 3.2: *Error state diagram* of TCM code $(2, 1, 2)$ with $m = 2$.

L. Let $\tilde{\mathbf{G}}_{12}$ be this matrix:

$$\begin{aligned}\tilde{\mathbf{G}}_{12} &= \mathbf{G}_{12} + \mathbf{G}_{13}\mathbf{G}_{32} \\ &\quad + \mathbf{G}_{13}(\mathbf{G}_{33})\mathbf{G}_{32} \\ &\quad + \mathbf{G}_{13}(\mathbf{G}_{33})^2\mathbf{G}_{32} \\ &\quad + \mathbf{G}_{13}(\mathbf{G}_{33})^3\mathbf{G}_{32} \\ &\quad + \cdots \\ &= \mathbf{G}_{12} + \mathbf{G}_{13}(\mathbf{I}_4 - \mathbf{G}_{33})^{-1}\mathbf{G}_{32}\end{aligned} \tag{3.14}$$

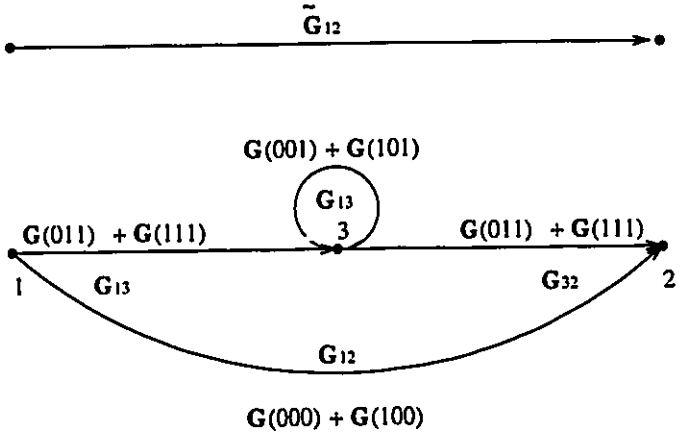

Figure 3.3: Extraction of matrix $\tilde{\mathbf{G}}_{12}$.

and similarly, Figure 3.4 provides:

$$\begin{aligned} \mathbf{G} &= \mathbf{G}_{01}\tilde{\mathbf{G}}_{12}\mathbf{G}_{20} \\ &+ \mathbf{G}_{01}\tilde{\mathbf{G}}_{12}\left(\mathbf{G}_{21}\tilde{\mathbf{G}}_{12}\right)\mathbf{G}_{20} \\ &+ \mathbf{G}_{01}\tilde{\mathbf{G}}_{12}\left(\mathbf{G}_{21}\tilde{\mathbf{G}}_{12}\right)^{2}\mathbf{G}_{20} \\ &+ \mathbf{G}_{01}\tilde{\mathbf{G}}_{12}\left(\mathbf{G}_{21}\tilde{\mathbf{G}}_{12}\right)^{3}\mathbf{G}_{20} \\ &+ \cdots \\ &= \mathbf{G}_{01}\tilde{\mathbf{G}}_{12}\left(\mathbf{I}_4 - \mathbf{G}_{21}\tilde{\mathbf{G}}_{12}\right)^{-1}\mathbf{G}_{20} \end{aligned} \tag{3.15}$$

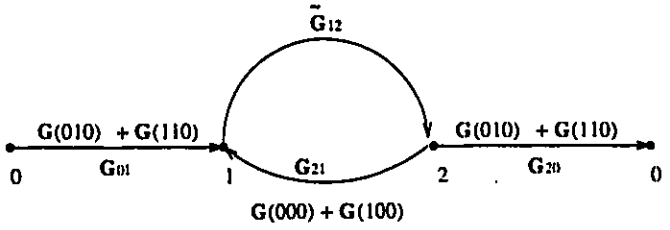

Figure 3.4: *Error state diagram* in terms of $\tilde{\mathbf{G}}_{12}$.

The two last Equations, (3.14) and (3.15), have finally consisted of a matrix inversion for which the elements are themselves matrices, precisely *error weight matrices*!

Let $\mathcal{G}$ be this super-matrix, and let $\mathcal{I}$ and $\mathcal{O}$ be the initial and final super-vectors:

$$\begin{aligned} \mathcal{I} &= \begin{bmatrix} \mathbf{G}_{01} \\ \mathbf{G}_{02} \\ \mathbf{G}_{03} \end{bmatrix} \\ \mathcal{O} &= \begin{bmatrix} \mathbf{G}_{10} \\ \mathbf{G}_{20} \\ \mathbf{G}_{30} \end{bmatrix} \\ \mathcal{G} &= \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} & \mathbf{G}_{13} \\ \mathbf{G}_{21} & \mathbf{G}_{22} & \mathbf{G}_{23} \\ \mathbf{G}_{31} & \mathbf{G}_{32} & \mathbf{G}_{33} \end{bmatrix} \end{aligned} \tag{3.16}$$

We may only represent them on the *error state diagram* of the code, and we suggest another representation of it, where all the non-zero states have also been doubled, as in Figure 3.5:

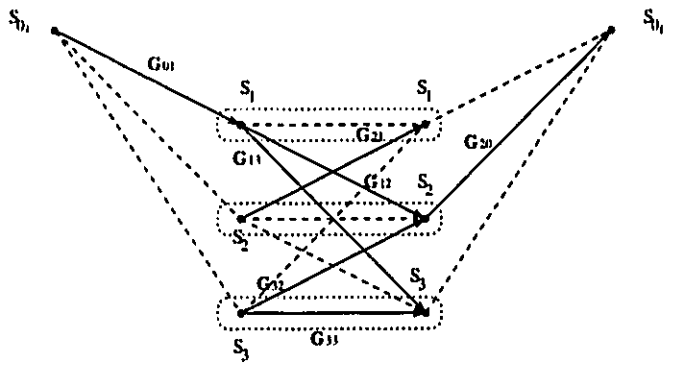

Figure 3.5: Extended representation of the *error state diagram.*

We then have, using the symbolic variables S_{0i} (initial state S_0), S_1, S_2, S_3, S_{0f} (final state S_0)for each state:

$$\begin{cases} S_1 = \mathbf{G}_{11}S_1 + \mathbf{G}_{21}S_2 + \mathbf{G}_{31}S_3 + \mathbf{G}_{01}S_{0i} \\ S_2 = \mathbf{G}_{12}S_1 + \mathbf{G}_{22}S_2 + \mathbf{G}_{32}S_3 + \mathbf{G}_{02}S_{0i} \\ S_3 = \mathbf{G}_{13}S_1 + \mathbf{G}_{23}S_2 + \mathbf{G}_{33}S_3 + \mathbf{G}_{03}S_{0i} \\ S_{0f} = \mathbf{G}_{10}S_1 + \mathbf{G}_{20}S_2 + \mathbf{G}_{30}S_3 \end{cases} \tag{3.17}$$

Let

$$\mathcal{S} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix}$$

so that Equation (3.17) may be rewritten in terms of the variables defined in Equation (3.16):

$$\begin{aligned} \mathcal{S} &= {}^{t}\mathcal{G}\mathcal{S} + S_{0i}\mathcal{I} \\ S_{0f} &= {}^{t}\mathcal{O}\mathcal{S} \\ \mathbf{G} = \frac{S_{0f}}{S_{0i}} &= {}^{t}\mathcal{O}\left(\mathcal{I}d_{(N_\nu-1)} - {}^{t}\mathcal{G}\right)^{-1}\mathcal{I} \end{aligned} \tag{3.18}$$

where $\mathcal{I}d_{(N_\nu-1)}$ denotes en identity matrix of dimension $(N_\nu - 1) \times (N_\nu - 1)$.

Then, the transfer function is provided according to Equation (3.11) and the upper bound for the probability of symbol error is given by Equation (3.12).

To refer to our example, we have:

$$\mathcal{G} = \begin{bmatrix} 0 & G_{12} & G_{13} \\ G_{21} & 0 & 0 \\ 0 & G_{32} & G_{33} \end{bmatrix}$$

and all the G_{ij} matrices are the sum of two different *error weight matrices* as seen in Figure 3.2.

Recall that in Equation (3.18), $\mathcal{G}$ is a $(N_\nu - 1) \times (N_\nu - 1)$ matrix, elements of which are sums of the *error weight* $(N_\nu \times N_\nu)$ *matrices*. It is not reasonable to consider the transfer function evaluation of TCM codes any longer unless there exists a way to perform their computation with a lower and acceptable level of complexity. Zehavi and Wolf [14] pointed out two conditions to reduce this complexity. The first one defines a class of codes: linear codes followed by a non-linear mapping. The TCM codes are of this type. The second one is a relation which expresses conservation of distances over two sets of symbols at the output of the mapping. [1] defined a rule to satisfy the second condition, the *uniformity condition*, based on geometric properties of the mappings.

3.2 The *uniformity condition*

This part is differently developed in [1]. It was worth applying the method to the coder from example 3.1.2, and closely relate it to the theory developed in section 3.1. As a result, it appeared that the existence of an *isometry* on the mapping was sufficient only for some kind of errors and was not necessary otherwise. This section therefore restates the *uniformity condition*, and derives it from the observation of example 3.1.2. Then, it expresses it and proves it only for the Ungerboeck codes, the only TCM codes that have been used in this thesis. Appendix B deals with the general case.

This rule is based on the concept of *row-uniformity* and *column-uniformity*, developed in [1]. Here, we only deal with row-uniformity: row properties of a matrix and the column properties of its transpose are strictly identical. Since $T(D)$ is a scalar,

$T(D) = {}^{t}T(D)$ and Equation (3.11) may also be rewritten:

$$\begin{aligned} T(D) &= \frac{1}{N_\nu} {}^{t}\mathbf{1}\mathbf{G}\mathbf{1} \\ &= \frac{1}{N_\nu} {}^{t}\left({}^{t}\mathbf{1}\mathbf{G}\mathbf{1}\right) \\ &= \frac{1}{N_\nu} {}^{t}\mathbf{1}\,{}^{t}\mathbf{G}\mathbf{1} \end{aligned}$$

Thus, $T(D)$ is expressed similarly with either $\mathbf{G}$ or ${}^{t}\mathbf{G}$ and any consideration on the rows of $\mathbf{G}$ applies also to the rows of ${}^{t}\mathbf{G}$, that is, on the columns of $\mathbf{G}$. Hence, the problem reduces to the simple consideration of row properties.

$\mathbf{G}$, as defined in Equation (3.18), is an infinite sum of products of some $\mathbf{G}_{ij}$ matrices and $T(D)$ is given by $T(D) = \frac{1}{N_\nu} {}^{t}\mathbf{1}\mathbf{G}\mathbf{1}$. Since $\mathbf{G}_{ij}$ is a sum of *error weight matrices*, we may carefully look at a product of the form:

$$\mathbf{G}(E)\mathbf{1}$$

If $\mathbf{1}$ is an eigenvector of all the $\mathbf{G}(E)$ matrices, then it is also an eigenvector of a sum of some of them:

$$\left(\mathbf{G}(E_i) + \mathbf{G}(E_j)\right)\mathbf{1} = (\alpha_i + \alpha_j)\,\mathbf{1}$$

and of any product of some of their sums:

$$\mathbf{G}_1\mathbf{G}_2\mathbf{1} = \mathbf{G}_1\alpha_2\mathbf{1} = \alpha_2\mathbf{G}_1\mathbf{1} = \alpha_2\alpha_1\mathbf{1}$$

But, this condition is verified if and only if the sum of the elements over a row of a transition matrix $\mathbf{G}(E)$ do not depend on this row:

$$\begin{bmatrix} * & * & * & * \\ * & * & * & * \\ o & o & o & o \\ \bullet & \bullet & \bullet & \bullet \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \sum * \\ \sum * \\ \sum o \\ \sum \bullet \end{bmatrix} = \alpha \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\Leftrightarrow \sum * = \sum * = \sum o = \sum \bullet = \alpha$$

According to [1], this property is named *uniformity* and when the matrix $\mathbf{G}$ of a code verifies it, the TCM scheme is called *uniform*.

In our last example, because of the trellis structure of the code, all *error weight matrices* $\mathbf{G}(E)$ are of the form:

$$\mathbf{G}(E) =$$

$$\frac{1}{4}\begin{bmatrix} D\|f(000)-f(e^{(2)}e^{(1)}e^{(0)})\|^2 & D\|f(010)-f(e^{(2)}\overline{e^{(1)}}e^{(0)})\|^2 & 0 & 0 \\ 0 & 0 & D\|f(001)-f(e^{(2)}e^{(1)}\overline{e^{(0)}})\|^2 & D\|f(011)-f(e^{(2)}\overline{e^{(1)}e^{(0)}})\|^2 \\ D\|f(010)-f(e^{(2)}\overline{e^{(1)}}e^{(0)})\|^2 & D\|f(000)-f(e^{(2)}e^{(1)}e^{(0)})\|^2 & 0 & 0 \\ 0 & 0 & D\|f(011)-f(e^{(2)}\overline{e^{(1)}e^{(0)}})\|^2 & D\|f(001)-f(e^{(2)}e^{(1)}\overline{e^{(0)}})\|^2 \end{bmatrix}$$

$$+$$

$$\frac{1}{4}\begin{bmatrix} D\|f(100)-f(\overline{e^{(2)}}e^{(1)}e^{(0)})\|^2 & D\|f(110)-f(\overline{e^{(2)}e^{(1)}}e^{(0)})\|^2 & 0 & 0 \\ 0 & 0 & D\|f(101)-f(\overline{e^{(2)}}e^{(1)}\overline{e^{(0)}})\|^2 & D\|f(111)-f(\overline{e^{(2)}e^{(1)}e^{(0)}})\|^2 \\ D\|f(110)-f(\overline{e^{(2)}e^{(1)}}e^{(0)})\|^2 & D\|f(100)-f(\overline{e^{(2)}}e^{(1)}e^{(0)})\|^2 & 0 & 0 \\ 0 & 0 & D\|f(111)-f(\overline{e^{(2)}e^{(1)}e^{(0)}})\|^2 & D\|f(101)-f(\overline{e^{(2)}}e^{(1)}\overline{e^{(0)}})\|^2 \end{bmatrix}$$

We observe that $\mathbf{1}$ is an eigenvector of any transition matrix if and only if, for any error $E = e^{(2)}e^{(1)}e^{(0)}$:

$$\begin{aligned} D\|f(011)-f(e^{(2)}\overline{e^{(1)}e^{(0)}})\|^2 &+ D\|f(111)-f(\overline{e^{(2)}e^{(1)}e^{(0)}})\|^2 + \\ &\quad D\|f(001)-f(e^{(2)}e^{(1)}\overline{e^{(0)}})\|^2 + D\|f(101)-f(\overline{e^{(2)}}e^{(1)}\overline{e^{(0)}})\|^2 \\ = D\|f(010)-f(e^{(2)}\overline{e^{(1)}}e^{(0)})\|^2 &+ D\|f(110)-f(\overline{e^{(2)}e^{(1)}}e^{(0)})\|^2 + \\ &\quad D\|f(000)-f(e^{(2)}e^{(1)}e^{(0)})\|^2 + D\|f(100)-f(\overline{e^{(2)}}e^{(1)}e^{(0)})\|^2 \end{aligned} \tag{3.19}$$

In fact, inside the elements of $\mathbf{G}(E)$, states 0 and 2 are associated with the subset

$$\mathcal{Y}_0 = \{000, 010, 100, 110\}$$

while states 1 and 3 are associated with

$$\overline{\mathcal{Y}_0} = \mathcal{Y}_0 \oplus \tilde{Y} = \{001, 011, 101, 111\}$$

Consider the TCM encoder scheme drawn in Figure 2.1. Since $y^{(0)} = r_0$, the 2^m vectors $Y = \left(y^{(m)}, y^{(m-1)}, \ldots, y^{(1)}, 0\right)$ correspond to all even states, while all the 2^m other vectors $Y = \left(y^{(m)}, y^{(m-1)}, \ldots, y^{(1)}, 1\right)$ correspond to all odd states. Hence,

$$\tilde{Y} = 1$$

In order to express this point more generally, a definition of $\mathcal{Y}_0$ is given in Appendix B. This Appendix is based on [14] which gave a definition of $\mathcal{Y}_0$ based on a property of linear codes, closely related to the all zero-state. Also, there exists a theory on conversion from feedback form convolutional codes to feedforward form ones. Algorithms are developed and provided by Porath in [8].

We here only apply the concept of the *uniformity condition* on some particular type of TCM codes, the Ungerboeck recursive ones, depicted in Figure 2.1, and point out the practical meaning of it on this example. Using this $\mathcal{Y}_0$ subset, Equation (3.19) becomes:

$$\sum_{Y \in \mathcal{Y}_0} D^{\|f(Y)-f(Y\oplus E)\|^2} = \sum_{Y \in \overline{\mathcal{Y}_0}} D^{\|f(Y)-f(Y\oplus E)\|^2} = \sum_{Y \in \mathcal{Y}_0} D^{\|f(Y\oplus \hat{Y})-f(Y\oplus \hat{Y}\oplus E)\|^2} \qquad (3.20)$$

Let us focus on Equation (3.20). Depending on the symbol error E, or more precisely, whether E is even or odd, symbol $Y \oplus E$ may be either in $\mathcal{Y}_0$ or in its complementary set $\overline{\mathcal{Y}_0}$. Consider a fixed error E. Lemma 2 from [14] states the existence of two cases, as far as E is concerned. For our particular codes, these cases turn out to be:

first case: E is odd.

$$\forall Y \in \mathcal{Y}_0 \ , \ Y \oplus E \in \overline{\mathcal{Y}_0}$$

Let Y_0 be a particular symbol in $\mathcal{Y}_0$. Then, $Y_0 \oplus E \in \overline{\mathcal{Y}_0}$. Thus, there exists a unique one $Y_1 \in \mathcal{Y}_0$ such that: $\overline{Y_1} = Y_1 \oplus \hat{Y} \in \overline{\mathcal{Y}_0}$. This symbol Y_1 verifies:

$$\overline{Y_1} = Y_1 \oplus \hat{Y} = Y_0 \oplus E$$

or equivalently:

$$Y_1 \oplus \hat{Y} \oplus E = Y_0$$

Hence, from the left hand side of Equation (3.20), the term corresponding to Y_0 is:

$$D^{\|f(Y_0)-f(Y_0\oplus E)\|^2}$$

and the right hand side one corresponding to Y_1 is

$$\begin{aligned} D^{\|f(Y_1\oplus\tilde{Y})-f(Y_1\oplus\tilde{Y}\oplus E)\|^2} &= D^{\|f(Y_1\oplus\tilde{Y})-f(Y_0)\|^2} \\ &= D^{\|f(Y_0\oplus E)-f(Y_0)\|^2} \end{aligned}$$

Finally, we obtain:

$$\forall Y_0 \in \mathcal{Y}_0 \ , \ \exists! \overline{Y_1} \in \overline{\mathcal{Y}_0} : \ D^{\|f(Y_0\oplus E)-f(Y_0)\|^2} = D^{\|f(\overline{Y_1}\oplus E)-f(\overline{Y_1})\|^2} \qquad (3.21)$$

which proves that (3.20) is term by term verified. This one-to-one correspondence comes from the fact that:

$$\{f(Y), Y \in \mathcal{Y}_0\} = \left\{f(Y \oplus \tilde{Y} \oplus E), \in \mathcal{Y}_0\right\}$$

Note that the distances $\|f(Y_0) - f(Y_0 \oplus E)\|$ and $\|f(Y_0 \oplus \tilde{Y}) - f(Y_0 \oplus \tilde{Y} \oplus E)\|$ are not individually preserved, but globally remain constant by pairs, with $\|f(Y_1) - f(Y_1 \oplus E)\|$ and $\|f(Y_1 \oplus \tilde{Y}) - f(Y_1 \oplus \tilde{Y} \oplus E)\|$.

second case: E is even.

$$\forall Y \in \mathcal{Y}_0 \ , \ Y \oplus E \in \mathcal{Y}_0$$

Here, the pairs of points from left hand side of Equation (3.20) are all in $\mathcal{Y}_0$ and their corresponding points from the right hand side are in $\overline{\mathcal{Y}_0}$. Now, we may think about an isometry which would transform the subset $\mathcal{Y}_0$ into $\overline{\mathcal{Y}_0}$:

$$Y \in \mathcal{Y}_0 : f(Y) \longmapsto f(Y \oplus \tilde{Y})$$

The distances $\|f(Y) - f(Y \oplus E)\|^2$ and $\|f(Y \oplus \tilde{Y}) - f(Y \oplus \tilde{Y} \oplus E)\|^2$ are then preserved: Equation (3.20) is still verified and the code is *uniform*.

Thus, instead of keeping matrices on the *error state diagram*, we may compute the transfer function with simple scalars.

Before the implementation of these transfer functions, we must verify, for all the mappings described in chapter 2, the validity of the *uniformity condition* of these codes. This is provided in section 3.3.

3.3 *Uniformity condition* on our mappings

It is now necessary to verify that the mappings developed in chapter 2 obey the *uniformity condition*. This verification is provided here for the optimum 4-QAM and 8-CROSS mappings. For the higher level QAM constellations, it has also been done, but is not reported here because it was as simple to verify it as for 4-QAM and 8-CROSS. Based on the original problem and the first derived mathematical expressions for probability of error, the case of example 3.1.2 is particularly focused on and is treated back.

Observe Figure 3.6. There exists an isometry which transforms the subset $\mathcal{Y}_0$ into $\overline{\mathcal{Y}_0}$: a reflection along the y-axis.

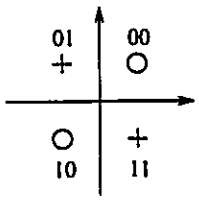

Figure 3.6: Subsets $\mathcal{Y}_0$ and $\overline{\mathcal{Y}_0}$ for a TCM 4-QAM.

Observe now the 8-CROSS case, on Figure 3.7. There still exists an isometry that maps the elements of $\mathcal{Y}_0$ into the ones of $\overline{\mathcal{Y}_0}$: a rotation by $-\frac{\pi}{2}$. Since the 8-QAM mapping is an extension of the 4-QAM one, we may think there is a contradiction between these two isometries. In fact, for the very simple 4-QAM mapping, they have the same effect on $\mathcal{Y}_0$.

To illustrate the behavior of our *uniformity condition*, we will now have a careful look at this 8-CROSS mapping. Since, for all the recursive convolutional codes we use, the so-called symbol $\tilde{Y} = 1$, we may first observe that whenever the error E is even, all Y and $Y \oplus E$ are contained in $\mathcal{Y}_0$. On the other hand, if E is odd, the distances of each side of the distances conservation equation are distances between two points, one taken from $\mathcal{Y}_0$ and the other from $\overline{\mathcal{Y}_0}$.

Figure 3.8 shows the way pairs of signal points are transformed through the isometry for an error symbol $E = 010$ on the left, and $E = 101$ on the right. On the

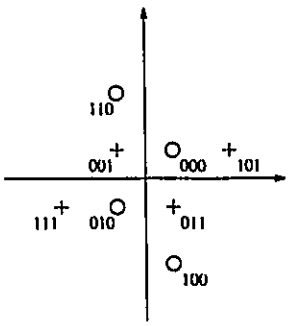

Figure 3.7: Subsets $\mathcal{Y}_0$ and $\overline{\mathcal{Y}_0}$ for a TCM 8-QAM.

left, the distances follow the full effect of the rotation induced by the addition of E. On the right, a double arrow illustrates the way any couple of points is mapped to another one: their distances are not preserved individually when each symbol Y is replaced by $Y + \tilde{Y}$, but they are transformed from one to each other which makes the sum of the total distances preserved.

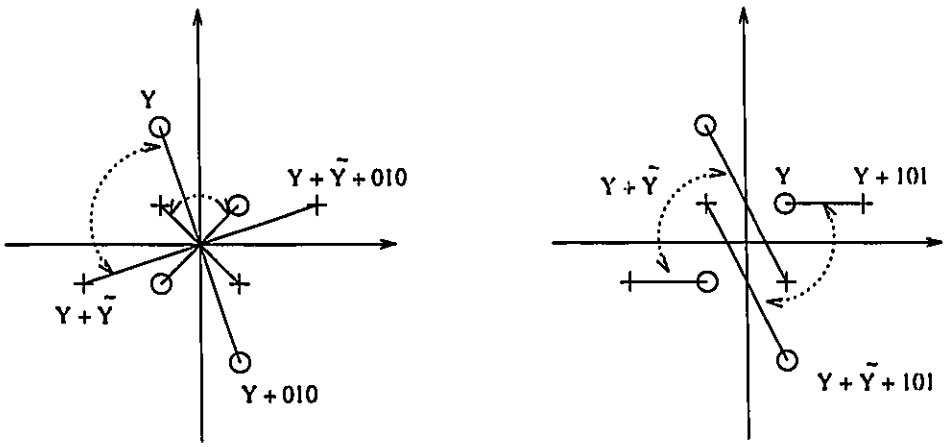

Figure 3.8: Behavior of the distances between points of $\mathcal{Y}_0$ with symbol errors $E = 010$ and $E = 101$.

The same technique can be applied to all the other mappings to show that all the optimum mappings satisfy the *uniformity condition* through rotation by $-\frac{\pi}{2}$ that maps $\mathcal{Y}_0$ to $\overline{\mathcal{Y}_0}$. Especially, optimum mappings at the first order only and at the third order satisfy have this property. They will be used in transfer function implementation to provide comparisons of performances for our TCM codes. Once we are assured of

this *uniform property* of the code, we will compute all labels $\mathbf{G}_{ij}$ as scalars. From now on, the *error weight matrices* $\mathbf{G}(Y_{t_{i\_j}})$ used to express the transition matrices in Equation (3.13) are replaced by simple scalar weight functions, $\mathbf{W}(E_{t_{i\_j}})$:

$$\mathbf{G}_{ij} = \sum_{t_{i\_j}} \mathbf{W}(E_{t_{i\_j}}) \tag{3.22}$$

$$\mathbf{W}(E) = \frac{1}{2^m} \sum_{Y \in \mathcal{Y}_0} D^{\|J(Y)-J(Y \oplus E)\|^2} \tag{3.23}$$

If the mappings had been optimized according to section 2.6, the isometry would perhaps no longer be the same. However, if a theoretical evaluation of trellis code was to be derived according to Equations (3.18), (3.22) and (3.23), there must first exist an isometry satisfying the *uniformity condition* to make it valid.

3.4 Bit error probability evaluation on the trellis of a TCM code

We recall here the possibility to extend the computation of the transfer function to a two-variables dependency D and I. On each transition t_{i_j}, instead of $\mathbf{W}(E_{ij})$, we may label a function $\mathbf{W}(E_{ij})I^{\epsilon_{ij}}$ where ϵ_{ij} is the number of incorrect input bits, those equal to 1 on the *error state diagram*. The derivative of the transfer function with respect to I, when I is set to 1 is then equal to m times the upper probability of a bit error to occur since m is the total number of bits at the input of the TCM code. This is the method used to compute the theoretical bit error probability. When the *uniformity condition* is satisfied, $\mathbf{W}(E)$ may be expressed in a scalar form and the theoretical bit error probability is obtained with:

$$\mathbf{G}_{ij} = \sum_{t_{i\_j}} \mathbf{W}(E_{t_{i\_j}}) I^{\epsilon_{ij}} \tag{3.24}$$

$$\mathbf{W}(E) = \frac{1}{2^m} \sum_{Y \in \mathcal{Y}_0} D^{\|J(Y)-J(Y \oplus E)\|^2} \tag{3.25}$$

Once again, under the *uniformity condition*, Equation (3.11) is simplified, since in Equations (3.10) or (3.18), $\mathcal{G}$, $\mathcal{I}$ and $\mathcal{O}$ are matrices, the elements of which are no

longer matrices, but the $\mathbf{G}_{ij}$ scalars:

$$\mathbf{G} = {}^{t}\mathcal{O}\left(\mathcal{I}d_{(N_\nu-1)} - {}^{t}\mathcal{G}\right)^{-1}\mathcal{I} \tag{3.26}$$

Equation (3.26) was the one used for computations. Indeed, $\mathbf{G}$ is now a 1×1 matrix and from Equation (3.11), $T(D)$, which is the sum of its elements, is simply expressed by: $T(D) = G$. An upper bound on the probability of bit error is given by Equation (3.27):

$$\mathcal{P}_b \leq \frac{1}{2m}\left.\frac{\partial T(D,I)}{\partial I}\right|_{D=e^{-\frac{1}{4N_0}},I=1} \tag{3.27}$$

3.5 Symbol error probability evaluation on the trellis code

In Equation (3.1), when an event error occurs, i.e. $S^L \mapsto \check{S}^L$, the L corresponding symbols of the sequence Y^L are wrong. Thus, as with Equation (3.1), the symbol error probability $\mathcal{P}_S$ may be expressed according to:

$$\begin{aligned}\mathcal{P}_S &\leq \sum_L \sum_{\mathbf{S}^L} L\,\mathcal{P}\{\mathbf{S}^L\}\mathcal{P}_e\{S^L\} \\ &= \sum_L \sum_{\mathbf{S}^L} \sum_{\mathbf{S}^L \neq \check{\mathbf{S}}^L} L\,\mathcal{P}\{\mathbf{S}^L\}\mathcal{P}\{\check{\mathbf{S}}^L|\mathbf{S}^L\} \\ &= \sum_L \sum_{\mathbf{Y}^L \neq \mathbf{0}} L\,\mathbf{W}(\mathbf{E}^L)\end{aligned} \tag{3.28}$$

where $\mathbf{W}(\mathbf{E}^L)$ is defined in Equation (3.5).

Once more, we may extend the definition of the transfer function. On any transition $t_{i \longrightarrow j}$, we may multiply the *error weight* $\mathbf{W}(E_{ij})$ by a factor (Y^1) where the exponent of Y, say 1 here, represents the number of transitions through which the one-length path $(i \longmapsto j)$ has passed. The final expression of the transfer function $T(D,Y)$ is a sum of the form $\sum_i c_i D^{d_i} Y^{n_i}$ where n_i is the number of transitions through which the c_i existing paths with *error weight* D^{d_i} have passed. Thereafter,

the derivative of $T(D,Y)$ with regards to Y becomes, when Y is set to 1: $\sum_i n_i \left(c_i D^{d_i}\right)$ and the transition *error weights* $\mathbf{G}_{ij}$ of the *error state diagram* could be defined as:

$$\mathbf{G}_{ij} = \sum_{t_{i\rightarrow j}} \mathbf{W}(E_{t_{i\rightarrow j}})Y \tag{3.29}$$

If the *uniformity condition* is satisfied, the transition *error weights* are scalars, which makes the computation rather easy to perform. An upper bound for the symbol error rate is then equal to:

$$\mathcal{P}_S \leq \frac{1}{2}\frac{\partial T(D,Y)}{\partial Y}\bigg|_{D=e^{-\frac{1}{4N_0}},Y=1} \tag{3.30}$$

The complete transfer function $T(D,I,Y)$ could be computed with

$$\mathbf{G}_{ij} = \sum_{t_{i\rightarrow j}} \mathbf{W}(E_{t_{i\rightarrow j}})I^{c_{ij}}Y$$

using Equations (3.26) and (3.11) to derive:

$$\begin{aligned}
\mathcal{P}_e &\leq \frac{1}{2}T(D,I,Y)\Big|_{D=e^{-\frac{1}{4N_0}},I=1,Y=1} \\
\mathcal{P}_S &\leq \frac{1}{2}\frac{\partial T(D,I,Y)}{\partial Y}\bigg|_{D=e^{-\frac{1}{4N_0}},I=1,Y=1} \\
\mathcal{P}_b &\leq \frac{1}{m}\frac{1}{2}\frac{\partial T(D,I,Y)}{\partial I}\bigg|_{D=e^{-\frac{1}{4N_0}},I=1,Y=1}
\end{aligned}$$

However, a simpler practical solution has been chosen for the algorithm. To do so, a general transfer function $T(D,X)$ has been considered. We could then select either $X = I$ and use the Hamming weight of the output of each transition as its exponent, or $X = Y$ and leave 1 as its exponent for each transition. The latter is used to derive the theoretical expression for the probability of symbol errors while the former for the bit error probability.

3.6 Improved error probabilities

In Equation (3.5), the *Bhattacharyya* bound was used:

$$\mathcal{P}\{\mathbf{S}^L \rightarrow \tilde{\mathbf{S}}^L\} = \frac{1}{2}\operatorname{erfc}\left(\frac{\|\mathbf{S}^L - \tilde{\mathbf{S}}^L\|}{2\sqrt{N_0}}\right) \leq \frac{1}{2}\exp\left(-\frac{\|\mathbf{S}^L - \tilde{\mathbf{S}}^L\|^2}{4N_0}\right) \tag{3.31}$$

using the inequality (3.32):

$$\operatorname{erfc}\left(\sqrt{x}\right) \leq \exp\left(-x\right), \; x \geq 0 \tag{3.32}$$

with

$$x = \frac{\|\mathbf{S}^L - \tilde{\mathbf{S}}^L\|^2}{4N_0}$$

Since

$$\|\mathbf{S}^L - \tilde{\mathbf{S}}^L\| \geq d_{free} \tag{3.33}$$

we could use a tighter inequality as given in (3.34):

$$\operatorname{erfc}\left(\sqrt{x+y}\right) \leq \operatorname{erfc}\left(\sqrt{x}\right) \exp\left(-y\right), \; x \geq 0, \; y \geq 0 \tag{3.34}$$

with

$$\begin{aligned} x &= \frac{d_{free}^2}{4N_0} \\ y &= \frac{\|\mathbf{S}^L - \tilde{\mathbf{S}}^L\|^2 - d_{free}^2}{4N_0} \end{aligned}$$

$$\begin{aligned} \mathcal{P}\{\mathbf{S}^L \rightarrow \tilde{\mathbf{S}}^L\} &\leq \frac{1}{2}\operatorname{erfc}\left(\frac{d_{free}}{2\sqrt{N_0}}\right) \exp\left(\frac{d_{free}^2}{4N_0}\right) \exp\left(-\frac{\|\mathbf{S}^L - \tilde{\mathbf{S}}^L\|^2}{4N_0}\right) \\ &\leq \frac{1}{2}\operatorname{erfc}\left(\frac{d_{free}}{2\sqrt{N_0}}\right) \exp\left(\frac{d_{free}^2}{4N_0}\right) Z^{\|\mathbf{S}^L - \tilde{\mathbf{S}}^L\|^2} \end{aligned} \tag{3.35}$$

where $Z = \exp\left(-\frac{1}{4N_0}\right)$.

This leads to new upper bounds for event, symbol and bit error probabilities in the form of:

$$\mathcal{P}_e \leq \frac{1}{2}\operatorname{erfc}\left(\frac{d_{free}}{2\sqrt{N_0}}\right) \exp\left(\frac{d_{free}^2}{4N_0}\right) T(D)\Big|_{D=e^{-\frac{1}{4N_0}}} \tag{3.36}$$

$$\mathcal{P}_S \leq \frac{1}{2}\operatorname{erfc}\left(\frac{d_{free}}{2\sqrt{N_0}}\right) \exp\left(\frac{d_{free}^2}{4N_0}\right) \left.\frac{\partial T(D,Y)}{\partial Y}\right|_{D=e^{-\frac{1}{4N_0}}, Y=1} \tag{3.37}$$

$$\mathcal{P}_b \leq \frac{1}{2m}\operatorname{erfc}\left(\frac{d_{free}}{2\sqrt{N_0}}\right) \exp\left(\frac{d_{free}^2}{4N_0}\right) \left.\frac{\partial T(D,I)}{\partial I}\right|_{D=e^{-\frac{1}{4N_0}}, I=1} \tag{3.38}$$

3.7 Analysis of error probability upperbounds for TCM codes

Under the following notation for the series expansions of $T(D)$, $\left.\frac{\partial T(D,Y)}{\partial Y}\right|_{Y=1}$ and $\frac{1}{m}\left.\frac{\partial T(D,I)}{\partial I}\right|_{I=1}$:

$$T(D) = N_e(d_{free})D^{d^2_{free}} + N_e(d_{next})D^{d^2_{next}} + \cdots \quad (3.39)$$

$$\left.\frac{\partial T(D,Y)}{\partial Y}\right|_{Y=1} = N_S(d_{free})D^{d^2_{free}} + N_S(d_{next})D^{d^2_{next}} + \cdots \quad (3.40)$$

$$\frac{1}{m}\left.\frac{\partial T(D,I)}{\partial I}\right|_{I=1} = N_b(d_{free})D^{d^2_{free}} + N_b(d_{next})D^{d^2_{next}} + \cdots \quad (3.41)$$

for high signal-to-noise ratios, Equations (3.12), (3.30), and (3.27) give:

$$\mathcal{P}_e \simeq N_e(d_{free})\exp\left(-\frac{d^2_{free}}{4N_0}\right)\left[1+\frac{N_e(d_{next})}{N_e(d_{free})}\exp\left(-\frac{d^2_{next}-d^2_{free}}{4N_0}\right)\right] \quad (3.42)$$

$$\mathcal{P}_S \simeq N_S(d_{free})\exp\left(-\frac{d^2_{free}}{4N_0}\right)\left[1+\frac{N_S(d_{next})}{N_S(d_{free})}\exp\left(-\frac{d^2_{next}-d^2_{free}}{4N_0}\right)\right] \quad (3.43)$$

$$\mathcal{P}_b \simeq N_b(d_{free})\exp\left(-\frac{d^2_{free}}{4N_0}\right)\left[1+\frac{N_b(d_{next})}{N_b(d_{free})}\exp\left(-\frac{d^2_{next}-d^2_{free}}{4N_0}\right)\right] \quad (3.44)$$

whereas the improved upper bounds given by Equations (3.36), (3.37), and (3.38) provide:

$$\mathcal{P}_e \simeq \frac{1}{2}N_e(d_{free})\,\mathrm{erfc}\left(\frac{d_{free}}{2\sqrt{N_0}}\right)\left[1+\frac{N_e(d_{next})}{N_e(d_{free})}\exp\left(-\frac{d^2_{next}-d^2_{free}}{4N_0}\right)\right] \quad (3.45)$$

$$\mathcal{P}_S \simeq \frac{1}{2}N_S(d_{free})\,\mathrm{erfc}\left(\frac{d_{free}}{2\sqrt{N_0}}\right)\left[1+\frac{N_S(d_{next})}{N_S(d_{free})}\exp\left(-\frac{d^2_{next}-d^2_{free}}{4N_0}\right)\right] \quad (3.46)$$

$$\mathcal{P}_b \simeq \frac{1}{2}N_b(d_{free})\,\mathrm{erfc}\left(\frac{d_{free}}{2\sqrt{N_0}}\right)\left[1+\frac{N_b(d_{next})}{N_b(d_{free})}\exp\left(-\frac{d^2_{next}-d^2_{free}}{4N_0}\right)\right] \quad (3.47)$$

It is clear that the approximations given by Equations (3.12), (3.30), and (3.27) are rather loose. From now on, only approximations (3.36), (3.37), and (3.38) will be used.

It should be noted that up to this point, we have only dealt with the performances of the TCM transmission scheme with consideration only for the trellis decoder. However, each symbol carries $m-k$ uncoded bits. Therefore, if an error occurs at the

output of the demodulator, which only affects these bits, it is not taken into account by the performance evaluation analysis made above. On the other hand, it does not propagate along the sequence of transmitted symbols. Here lies the importance of the maximization of the minimum distance between points of a sub-constellation, as explained in chapter 2. It also appears as a possible defect of the ability of the demodulator to limit errors depending on the order of Δ_{k+1} and d_{free}. If Δ_{k+1} is smaller than d_{free}, for high SNR, at first order, it is very unlikely that an error event will occur. In this case, there are almost always only single symbol errors. Then, the performance of TCM is the same type as uncoded QAM on each different sub-constellation. At first order, the event and symbol error rates are the same:

$$\begin{aligned} \mathcal{P}_e &= \mathcal{P}_S \\ &= \frac{1}{2} N_S(\Delta_{k+1}) \operatorname{erfc}\left(\frac{\Delta_{k+1}}{2\sqrt{N_0}}\right) \end{aligned} \tag{3.48}$$

where $N_S(\Delta_{k+1})$ denotes the average number, over the points of the constellation, of neighbors distant from this point by Δ_{k+1} on the sub-constellation of this point. When Δ_{k+1} is larger than d_{free}, at first order and for high SNR, d_{free} limits the performance of TCM. It is very unlikely that an error occurs on parallel transitions.

Up to now, we implicitly considered only this case. To derive more general probability of error expressions, we have to recall that d_{free}, such as we just used before, was ν-dependent, since ν denotes the memory length of the coder. From now on, we refer to it as $d_{free}(\nu)$. The effective free distance of TCM is therefore defined by:

$$d_{free} = \begin{cases} \min\{d_{free}(\nu), \Delta_{k+1}\} & \text{if } k < m \\ d_{free}(\nu) & \text{if } k = m \end{cases} \tag{3.49}$$

This distinction is extremely important in considering the TCM (2,1,2) code with a 4-QAM mapping. It is reported later that $d_{free}(\nu = 2)$ is equal to:

$$d_{free}(\nu = 2) = \sqrt{5}\Delta_0$$

while Δ_{k+1} is smaller:

$$\Delta_{k+1} = \Delta_2 = \sqrt{4}\Delta_0$$

Thus, when $m = k = 1$, the 4 point constellation is divided into 4 symbolic sub-constellations of one single point. The trellis has no parallel transitions and the limitation of performance is determined by $d_{free}(\nu = 2) = \sqrt{5}\Delta_0$. The absence of parallel transitions may be expressed by:

$$\text{if } m = k,\ \Delta_{k+1} = \infty$$

Otherwise, the constellation is decomposed into 4 sub-constellations of 2 or more points. In this case, there exist some parallel transitions on the trellis and the minimum distance between the points of a sub-constellation is indeed $\Delta_2 = \sqrt{4}\Delta_0$.

Now, consider Equation (3.49) with $m > k$. Following the discussion of [5], for a given number k of coded in-bits, an increase on the memory length ν of the code causes an increase on the free distance $d_{free}(\nu)$ of the code. Meanwhile, the minimum distance Δ_{k+1} reached on two parallel transitions remains constant. Thus, increasing the memory length of the encoder complicates decoding without improving the performance.

3.8 General expression in terms of SNR

We will discuss here the last equations, and consider only the first term in order to compare them with the uncoded QAM equations.

Let E_S be the average energy per symbol and let α_{cod} be defined by:

$$\alpha_{cod} = \frac{\left(\frac{d_{free}}{\Delta_0}\right)^2}{2\frac{E_S}{\Delta_0^2}} = \frac{d_{free}^2}{2E_S} \tag{3.50}$$

If ρ denotes the signal-to-noise ratio $\frac{E_S}{N_0}$, the above approximations become:

$$\mathcal{P}_e \simeq \frac{1}{2}N_e(d_{free})\ \text{erfc}\left(\sqrt{\frac{\alpha_{cod}\rho}{2}}\right) = N_e(d_{free})\ \text{Q}\left(\sqrt{\alpha_{cod}\rho}\right) \tag{3.51}$$

$$\mathcal{P}_S \simeq \frac{1}{2}N_S(d_{free})\ \text{erfc}\left(\sqrt{\frac{\alpha_{cod}\rho}{2}}\right) = N_S(d_{free})\ \text{Q}\left(\sqrt{\alpha_{cod}\rho}\right) \tag{3.52}$$

$$\mathcal{P}_b \simeq \frac{1}{2}N_b(d_{free})\ \text{erfc}\left(\sqrt{\frac{\alpha_{cod}\rho}{2}}\right) = N_b(d_{free})\ \text{Q}\left(\sqrt{\alpha_{cod}\rho}\right) \tag{3.53}$$

where $Q(x)$ is the Gaussian integral function:

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

Since only the factor $N(d_{free})$ differs in both the bit, symbol and error event probabilities, we will not write the index e, s nor b on the following equations for any of the variables $\mathcal{P}$ and $N(d_{free})$. Instead, we will use the index cod and unc to refer respectively to (coded) TCM and uncoded QAM transmission schemes.

Thus, the error probability of TCM scheme is given by:

$$\mathcal{P}_{cod} \simeq N_{cod}\, Q\left(\sqrt{\alpha_{cod}\rho_{cod}}\right) \tag{3.54}$$

where

$$\rho_{cod} = \frac{E_{S\,cod}}{N_0} \tag{3.55}$$

N_{cod} is the average number of points on the constellation, which are output from a state of the encoder for which there exist two paths distant from d_{free} on the trellis code. This result is to be compared to the error probability of uncoded QAM scheme:

$$\mathcal{P}_{unc} \simeq N_{unc}\, Q\left(\sqrt{\alpha_{unc}\rho_{unc}}\right) \tag{3.56}$$

where N_{unc} is the average number of neighboring points on the constellation which are distant from Δ_0, the minimum distance between two points of the constellation. This corresponds to a single state trellis with parallel transitions. The number of parallel transitions is equal to the number of possible output symbols, which is itself equal to the number of possible input symbols since there is no redundancy. Among the set of distinct pairs of parallel transitions from a state (the single one) to a following one (the same single one), there exists a subset of some of them for which the points of the constellation associated with the output symbols (according to a certain mapping table in use for such a transmission scheme), are distant from the minimal distance Δ_0: this minimal distance Δ_0 would be the free distance of such a code. The average number of pairs of parallel transitions distant from Δ_0 is equal

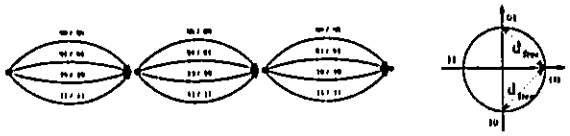

Figure 3.9: Trellis code of uncoded QAM with $m = 2$ in-bits.

to the number of neighboring points on the constellation, denoted by N_{unc}. This is depicted in Figure 3.9.

Finally, α_{unc} (resp. ρ_{unc}) is similarly expressed to α_{cod} (resp. ρ_{cod}) by substituting d_{free} (resp. E_{cod}) by $d_{min} = \Delta_0$ (resp. E_{unc}), and $N_{cod} = N_{free}$ by $N_{unc} = N_{neighbor}$:

$$\begin{aligned} \alpha_{unc} &= \frac{\left(\frac{d_{min}}{\Delta_0}\right)^2}{2\frac{E_{Sunc}}{\Delta_0^2}} \\ \rho_{unc} &= \frac{E_{Sunc}}{N_0} \end{aligned} \tag{3.57}$$

This indicates that we may consider the uncoded QAM transmission scheme as a TCM QAM one, with a rate $R_c = 1$ and zero memory length code! This can help determine the improvement of TCM over just QAM, and explains why Ungerboeck [10] defined the asymptotic coding gain as:

$$\gamma_{dB} = 10\log_{10}\left(\frac{\left(\frac{d_{free}}{\Delta_0}\right)^2}{\frac{E_{Scod}}{E_{Sunc}}}\right) \tag{3.58}$$

or, with the notations of (3.50) and (3.57):

$$\gamma_{dB} = 10\log_{10}\left(\frac{\alpha_{cod}}{\alpha_{unc}}\right) \tag{3.59}$$

With the Ungerboeck codes, the number k of coded in-bits for two-dimensional TCM is either 1 (for TCM $(2,1,2)$ code) or 2 (for the TCM $(3,2,\nu)$ codes). Thus, we are particularly interested in the values of Δ_2 and Δ_3 for all the mappings derived in chapter 2. Tables 3.1 and 3.2 contain these values, and make a difference whether m is equal to k or not.

Table 3.1: Values of Δ_2 for TCM code $(2,1,2)$.

m	Δ_2
$m{=}k$	∞
$m{>}k$	$\sqrt{4}\Delta_0$

Table 3.2: Values of Δ_3 for any $(3,2,\nu)$ TCM code.

m	Δ_3
$m{=}k$	∞
$m{>}k$	$\sqrt{8}\Delta_0$

3.8.1 Two term expressions in terms of SNR

In this section, we no longer refer to the previous variable α with an index $_{cod}$ but $_{free}$ or $_{\Delta_{k+1}}$ whether d_{free} or $d_{\Delta_{k+1}}$ is the minimum one. Along with these two indexes, we also use $_{next}$ for variables n and β which specify the probability of error at second order. Hence, we face the three following cases.

Two term approximation when d_{free} and d_{next} are dominant

To end this section, consider TCM where:

$$d_{free} = d_{free}(\nu) < d_{next} < \Delta_{k+1}$$

In this case, up to the second order term, the limiting factor is d_{next}, but not Δ_{k+1}. That is, the second most probable types of errors is an event error of distance d_{next} but not an error on parallel transitions. Equation (3.54) up to the second order term becomes:

$$\mathcal{P}_{cod} \simeq N_{cod}(d_{free})\, \mathrm{Q}\left(\sqrt{\alpha_{free}\rho_{cod}}\right)\left[1 + n_{cod}(d_{next})\exp\left(-\beta_{next}\rho_{cod}\right)\right] \tag{3.60}$$

where

$$\alpha_{free} = \frac{d_{free}^2}{2E_S} \tag{3.61}$$

$$n_{cod}(d_{next}) = \frac{N_{cod}(d_{next})}{N_{cod}(d_{free})} \tag{3.62}$$

$$\beta_{next} = \frac{d_{next}^2 - d_{free}^2}{4E_{Scod}} \tag{3.63}$$

$$\tag{3.64}$$

This will probably be useful to compare the theoretical and experimental results for intermediate SNR. Of course, for high SNR, only the first term may be necessary.

Two term approximation when d_{free} and Δ_{k+1} are dominant

In the case where

$$d_{free} = d_{free}(\nu) < \Delta_{k+1} < d_{next}$$

the second most probable kind of errors are on parallel transitions. Therefore,

$$\mathcal{P}_{cod} \simeq N_{cod}(d_{free})\, Q\left(\sqrt{\alpha_{free}\rho_{cod}}\right) + N_{cod}(\Delta_{k+1})\, Q\left(\sqrt{\delta_{\Delta_{k+1}}\rho_{cod}}\right) \tag{3.65}$$

where

$$\delta_{\Delta_{k+1}} = \frac{\Delta_{k+1}^2}{2E_{Scod}} \tag{3.66}$$

Using the bound in (3.34), this may look like Equation (3.60):

$$\mathcal{P}_{cod} \simeq N_{cod}(d_{free})\, Q\left(\sqrt{\alpha_{free}\rho_{cod}}\right)\left[1 + n_{cod}(d_{\Delta_{k+1}}) \exp\left(-\beta_{\Delta_{k+1}}\rho_{cod}\right)\right] \tag{3.67}$$

where

$$n_{cod}(d_{\Delta_{k+1}}) = \frac{N_{cod}(\Delta_{k+1})}{N_{cod}(d_{free})} \tag{3.68}$$

$$\beta_{\Delta_{k+1}} = \frac{\delta_{\Delta_{k+1}} - \alpha_{cod}}{2} = \frac{\Delta_{k+1}^2 - d_{free}^2}{4E_{Scod}} \tag{3.69}$$

Two term approximation when Δ_{k+1} and d_{free} are dominant

If

$$d_{free} = \Delta_{k+1} < d_{free}(\nu) < d_{next}$$

the two most probable kind of errors are on parallel transitions, with an event error distance d_{free}. Therefore,

$$\mathcal{P}_{cod} \simeq N_{cod}(\Delta_{k+1})\, Q\left(\sqrt{\delta_{\Delta_{k+1}}\rho_{cod}}\right) + N_{cod}(d_{free})\, Q\left(\sqrt{\alpha_{cod}\rho_{cod}}\right) \tag{3.70}$$

Again, combined with inequality (3.34) gives:

$$\mathcal{P}_{cod} \simeq N_{cod}(\Delta_{k+1})\, Q\left(\sqrt{\delta_{\Delta_{k+1}}\rho_{cod}}\right)\left[1 + n_{cod}(d_{free}) \exp\left(-\beta_{free}\rho_{cod}\right)\right] \tag{3.71}$$

where

$$n_{cod}(d_{free}) = \frac{N_{cod}(d_{free})}{N_{cod}(\Delta_{k+1})} \tag{3.72}$$

$$\beta_{free} = \frac{\alpha_{cod} - \delta_{\Delta_{k+1}}}{2} = \frac{d_{free}^2 - \Delta_{k+1}^2}{4E_{S_{cod}}} \tag{3.73}$$

Other cases

Recall that Δ_{k+1} is the minimum distance between points of a sub-constellation. Let Δ'_{k+1} be the second next minimum distance. The last situation would be the following one:

$$\Delta_{k+1} < \Delta'_{k+1} < d_{free}$$

However, with the Ungerboeck codes with a memory length ν less than 9 (which makes a 512 states encoder!), it is never the case. Therefore, it is not necessary to proceed on this any further.

Above, we only considered the cases with strict inequalities. The only possible equality would have been:

$$\begin{aligned} \Delta_{k+1} &= d_{free} \\ \text{or}\quad \Delta_{k+1} &= d_{next} \end{aligned}$$

In such a case, the coefficient of the first order term should be the sum of the corresponding coefficients N_{cod}. We write no formal expression for it but if this later case occurs, it will be correctly dealt with. The worst case could have been:

$$\begin{cases} d_{free} = \Delta_{k+1} \\ d_{next} = \Delta'_{k+1} \end{cases}$$

However, as shown later, $\frac{\Delta_{k+1}^{\prime 2}}{\Delta_{k+1}^2}$ is always equal to 2 or 4, while the ratio $\frac{d_{next}^2}{d_{free}^2}$ is never more than $\frac{4}{3}$.

3.8.2 Synthetic expression

Let d_{free} and d_{scnd} be the two minimum distances within $\{d_{free}(\nu), d_{next}, \Delta_{k+1}\}$. The previous expressions reduce to the general one:

$$\mathcal{P}_{cod} \simeq N_{cod}(d_{free})\, Q\left(\sqrt{\alpha_{d_{free}}\rho_{cod}}\right)\left[1 + n_{cod}(d_{scnd})\exp\left(-\beta_{d_{scnd}}\rho_{cod}\right)\right] \tag{3.74}$$

where

$$\begin{aligned} \alpha_{d_{free}} &= \frac{d_{free}^2}{2E_{Scod}} \\ n_{cod}(d_{scnd}) &= \frac{N_{cod}(d_{scnd})}{N_{cod}(d_{free})} \\ \beta_{d_{scnd}} &= \frac{d_{scnd}^2 - d_{free}^2}{4E_{Scod}} \end{aligned} \tag{3.75}$$

3.8.3 Conclusion

It has been shown in this chapter that the performance of TCM scheme can be expressed theoretically. These theoretical expressions are very complex, so they are probably intractable. However, the problem becomes more tractable if the mappings of the TCM codes obey the *uniformity condition.*

Moreover, TCM error probabilities for event, symbol and bit errors were presented, and given at second order. The limitations are ruled by two critical distances that are the minimum distance of the trellis code and the minimum distance between points inside the sub-constellations of the constellation.

Chapter 4

Transfer functions for TCM codes

In chapter 3, theoretical expressions for event, symbol and bit error for TCM codes were developed. The *uniformity condition* was satisfied on the mappings which allows for a reduction of the complexity of the calculation, and an implementation was then easier to process. The programs require a large amount of computer memory and could not deal with more than sixteen states. It would be desirable to develop a new algorithm in order to proceed further and determine all second order term parameters. This has not been done in this thesis.

4.1 Four state encoder $(n = 2, k = 1, \nu = 2)$ with no parallel transitions

We refer to the example developed in section 3.1.2 but with $m = 1$. The trellis of this code is shown in Figure 4.1 and the optimum mapping constructed in Figure 2.4, with 4 subsets of 1 point each. Note that the minimum distance between two points of the constellation is $\Delta_0 = 2$.

The simplicity of this example was very useful to verify our computations of the transfer functions. Meanwhile, we faced the problem of a non-reduced fraction found by the program. Therefore, the computation of a series expansion was necessary. This was only done for $T(D, X)|_{X=1}$ and $\frac{\partial T(D,X)}{\partial X}\Big|_{X=1}$, with X being either I or Y,

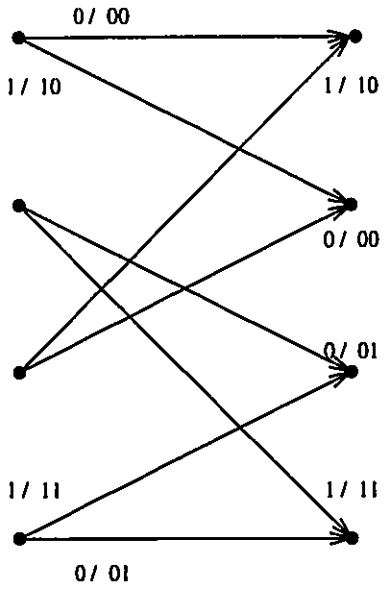

Figure 4.1: Trellis of the TCM $(2,1,2)$ code with $m = 1$.

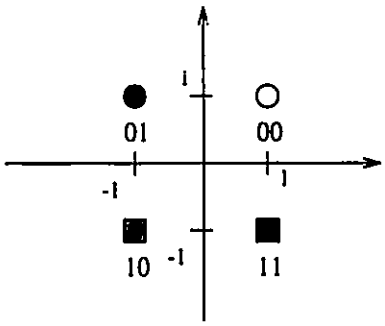

Figure 4.2: 4-QASK mapping used for the TCM $(2,1,2)$ code with $m = 1$.

depending if bit errors or symbol errors were considered.

The computer program gave the following results:

$$\begin{aligned}
\mathbf{W}(00) &= 1.0D^0 \\
\mathbf{W}(01) &= 1.0D^4 \\
\mathbf{W}(10) &= 1.0D^8 \\
\mathbf{W}(11) &= 1.0D^4
\end{aligned}$$

With $X = I$:

$$\mathcal{I} = \begin{bmatrix} \mathbf{G}_{01} \\ \mathbf{G}_{02} \\ \mathbf{G}_{03} \end{bmatrix} = \begin{bmatrix} \{1.0D^8\}I^1 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathcal{O} = \begin{bmatrix} \mathbf{G}_{10} \\ \mathbf{G}_{20} \\ \mathbf{G}_{30} \end{bmatrix} = \begin{bmatrix} 0 \\ \{1.0D^8\}I^1 \\ 0 \end{bmatrix}$$

$$\mathcal{G} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} & \mathbf{G}_{13} \\ \mathbf{G}_{21} & \mathbf{G}_{22} & \mathbf{G}_{23} \\ \mathbf{G}_{31} & \mathbf{G}_{32} & \mathbf{G}_{33} \end{bmatrix} = \begin{bmatrix} 0 & \{1.0D^4\}I^0 & \{1.0D^4\}I^1 \\ \{1.0D^0\}I^0 & 0 & 0 \\ 0 & \{1.0D^4\}I^1 & \{1.0D^4\}I^0 \end{bmatrix}$$

After the matrix inversion and multiplication with the two vectors $\mathcal{I}$ and $\mathcal{O}$, we obtained:

$$\begin{aligned}
T(D,I) &= \frac{\{1.0D^{20}-2.0D^{24}+1.0D^{28}\}I^2+\{1.0D^{24}-1.0D^{28}\}I^4}{\{1.0D^0-3.0D^4+3.0D^8-1.0D^{12}\}I^0+\{-1.0D^8+1.0D^{12}\}I^2} \\
T(D) &= T(D,I)|_{I=1} \\
&= \frac{1.0D^{20}-1.0D^{24}}{1.0D^0-3.0D^4+2.0D^8} \\
&= 1.0D^{20}\{1.0D^0+2.0D^4+4.0D^8+8.0D^{12}+16.0D^{16}+32.0D^{20}+\cdots\} \\
\frac{1}{m}\frac{\partial T(D,I)}{\partial I}\Big|_{I=1} &= \frac{2.0D^{20}-6.0D^{24}+4.0D^{28}+2.0D^{32}-2.0D^{36}}{1.0D^0-6.0D^4+13.0D^8-12.0D^{12}+4.0D^{16}} \\
&= 2.0D^{20}\{1.0D^0+3.0D^4+7.0D^8+16.0D^{12}+36.0D^{16}+80.0D^{20}+\cdots\}
\end{aligned}$$

The following results were computed manually with both direct matrix inversion technique and combination of Equations (3.15) and (3.14):

$$T(D) = \frac{\mathbf{G}_{01}\left[\mathbf{G}_{12} + \frac{\mathbf{G}_{13}}{1-\mathbf{G}_{33}}\mathbf{G}_{32}\right]\mathbf{G}_{20}}{1 - \mathbf{G}_{21}\left[\mathbf{G}_{12} + \frac{\mathbf{G}_{13}}{1-\mathbf{G}_{33}}\mathbf{G}_{32}\right]} \tag{4.1}$$

They are simplified versions of the latter:

$$\begin{aligned} T(D,I) &= D^{20}I^2\tfrac{1-D^4+D^4I^2}{(1-D^4)^2-D^8I^2} \\ T(D) &= D^{20}\tfrac{1}{1-2D^4} \\ \tfrac{1}{m}\tfrac{\partial T(D,I)}{\partial I}\Big|_{I=1} &= 2D^{20}\tfrac{1-D^4-D^8}{1-4D^4+4D^8} \end{aligned}$$

These reliable results ensured the proper implementation of the computations and we wanted to proceed the calculations further. With $X = Y$:

$$\begin{aligned} T(D,Y) &= \tfrac{\{1.0D^{20}\}Y^3+\{-1.0D^{24}\}Y^5}{\{1.0D^0\}Y^0+\{-1.0D^4\}Y^1+\{-2.0D^4\}Y^2+\{1.0D^8\}Y^3+\{1.0D^8\}Y^4} \\ T(D,Y)|_{Y=1} &= \tfrac{1.0D^{20}-1.0D^{24}}{1.0D^0-3.0D^4+2.0D^8} \\ &= T(D,I)|_{I=1} \\ &= T(D) \\ \tfrac{\partial T(D,Y)}{\partial Y}\Big|_{Y=1} &= \tfrac{3.0D^{20}-9.0D^{24}+9.0D^{28}-3.0D^{32}}{1.0D^0-6.0D^4+13.0D^8-12.0D^{12}+4.0D^{16}} \\ &= 3.0D^{20}\{1.0D^0 + 3.0D^4 + 8.0D^8 + 20.0D^{12} + 48.0D^{16} + 112.0D^{20} + \cdots\} \end{aligned}$$

Since Δ_0 was equal to 2 in the program, we derived the values:

$$\begin{aligned} d_{free}(\nu) &= \sqrt{5}\Delta_0 \\ d_{next} &= \sqrt{6}\Delta_0 \end{aligned}$$

It was easy to check in the trellis that there is one error event of distance $d_{free}(\nu) = \sqrt{5}\Delta_0$, along with 3 erroneous symbols, and 2 event errors of distance $d_{next} = \sqrt{6}\Delta_0$. One of them is 4 symbols long, the other 5, and the number of symbols in error through a d_{next}-long event error is therefore less than 9, as given in $\frac{\partial T(D,Y)}{\partial Y}\Big|_{Y=1}$. This is depicted in Figure 4.3.

Also note that an event error of minimum distance $d_{free}(\nu)$ carries an average of 2 in-bits in error. This can also be read from the trellis. Moreover, the transfer function is able to tell us that the error events of Euclidean distance d_{next} have a total of 6 bits in error. Thus, with no parallel transitions, we have a good two term approximation for any probability of error of TCM $(2,1,2)$ code.

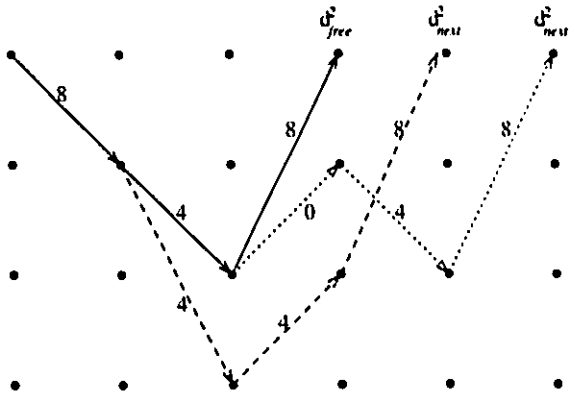

Figure 4.3: Event errors of distances d_{free} and d_{next} on a 4-QASK mapping used for TCM code $(2,1,2)$ with $m = 1$.

4.2 Four state encoder $(n = 2, k = 1, \nu = 2)$ with two or more parallel transitions

With some parallel transitions on the trellis, the limitations of the TCM code are now affected by $\Delta_2 = \sqrt{4}\Delta_0$ which is less than the free distance of the convolutional encoder. The free distance of the TCM code is therefore equal to Δ_2. As an example, the computations of the transfer functions of the convolutional encoder when $m = 2$ has been done with the mapping of Figure 4.4.

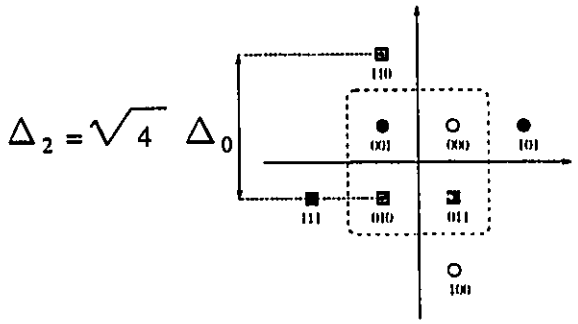

Figure 4.4: 8-CROSS mapping used for TCM code $(2,1,2)$ with $m = 2$.

$$\mathbf{W}(000) \;=\; 1.0D^0$$

$$
\begin{aligned}
\mathbf{W}(001) &= 0.5D^1 + 0.5D^{20} \\
\mathbf{W}(010) &= 0.5D^8 + 0.5D^{10} \\
\mathbf{W}(011) &= 0.5D^4 + 0.5D^{20} \\
\mathbf{W}(100) &= 1.0D^{16} \\
\mathbf{W}(101) &= 0.5D^4 + 0.5D^{20} \\
\mathbf{W}(110) &= 1.0D^8 \\
\mathbf{W}(111) &= 0.5D^4 + 0.5D^{20}
\end{aligned}
$$

$$
\mathcal{I} = \begin{bmatrix} \{0.5D^8 + 0.5D^{10}\}I^1 + \{1.0D^8\}I^2 \\ 0 \\ 0 \end{bmatrix}
$$

$$
\mathcal{O} = \begin{bmatrix} 0 \\ \{0.5D^8 + 0.5D^{10}\}I^1 + \{1.0D^8\}I^2 \\ 0 \end{bmatrix}
$$

With $X = I$:

$$
\mathcal{G} = \begin{bmatrix} 0 & \alpha & \alpha I \\ \beta & 0 & 0 \\ 0 & \alpha I & \alpha \end{bmatrix} \tag{4.2}
$$

with

$$
\begin{cases} \alpha &= \{0.5D^4 + 0.5D^{20}\}I^0 + \{0.5D^4 + 0.5D^{20}\}I^1 \\ \beta &= \{1.0D^0\}I^0 + \{1.0D^{16}\}I^1 \end{cases}
$$

$$
\begin{aligned}
T(D,I)|_{I=1} &= 2.25D^{20}\{1.0D^0 + 2.0D^4 + 4.0D^8 + 8.0D^{12} + 17.0D^{16} + \cdots\} \\
\tfrac{1}{m}\left.\tfrac{\partial T(D,I)}{\partial I}\right|_{I=1} &= 4.3125D^{20}\{1.0D^0 + 2.783D^4 + 6.609D^8 + 15.304D^{12} + 35.783D^{16} + \cdots\}
\end{aligned}
$$

With $X = Y$:

$$
\begin{aligned}
T(D,Y)|_{Y=1} &= 2.25D^{20}\{1.0D^0 + 2.0D^4 + 4.0D^8 + 8.0D^{12} + 17.0D^{16} + \cdots\} \\
\left.\tfrac{\partial T(D,Y)}{\partial Y}\right|_{Y=1} &= 6.75D^{20}\{1.0D^0 + 3.0D^4 + 8.0D^8 + 20.0D^{12} + 49.0D^{16} + \cdots\}
\end{aligned}
$$

This means that the mean number of error events of minimum distance $d_{f;\,\infty}(\nu) = \sqrt{5}\Delta_0$ is 2.25 and when such an error event occurs, it creates 3 symbols in error

which makes an average number of possible symbols in error of 6.75. However, for high SNR, an error event has Euclidean distance at least $\sqrt{5}\Delta_0$ and is a lot less likely to occur than an error on parallel transitions, with distance $\sqrt{4}\Delta_0$: the calculation of the transfer functions is no longer relevant at the first order.

However, second order expressions are developed in this thesis, and transfer functions were also processed for eight and sixteen parallel transitions to derive second order terms: the first term of these transfer functions is the second in the final probability of error expression. More terms than necessary are given to observe the evolution of the transfer functions when the number of parallel transitions increases.

4.2.1 Four parallel transitions

The optimum mapping of Figure 4.5 was derived from Figure 2.8. It contains 4

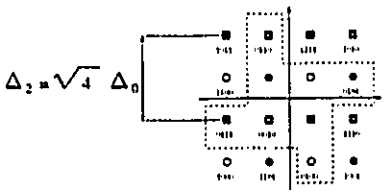

Figure 4.5: 16-QASK mapping used for the TCM $(2, 1, 2)$ code with $m = 3$.

subsets of 4 points.

With $X = I$, the program provided:

$$\begin{aligned}
\mathbf{W}(0000) &= 1.0D^0 \\
\mathbf{W}(0001) &= \mathbf{W}(0011) = \mathbf{W}(1001) = \mathbf{W}(1011) = 0.250D^4 + 0.5D^{20} + 0.250D^{36} \\
\mathbf{W}(0010) &= 0.250D^8 + 0.5D^{40} + 0.250D^{72} \\
\mathbf{W}(0100) &= \mathbf{W}(1100) = 1.0D^{16} \\
\mathbf{W}(0101) &= \mathbf{W}(0111) = \mathbf{W}(1101) = \mathbf{W}(1111) = 0.5D^4 + 0.250D^{20} + 0.250D^{52} \\
\mathbf{W}(0110) &= \mathbf{W}(1110) = 0.5D^8 + 0.5D^{40} \\
\mathbf{W}(1000) &= 1.0D^{32} \\
\mathbf{W}(1010) &= 1.0D^8
\end{aligned}$$

Using Equation (4.2), and the notation:

$$\mathcal{I} = \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix}$$

$$\mathcal{O} = \begin{bmatrix} 0 \\ a \\ 0 \end{bmatrix}$$

we obtained:

$$\begin{cases} a = \{0.250D^8 + 0.5D^{40} + 0.250D^{72}\}I^1 \\ \quad +\{1.5D^8 + 0.5D^{40}\}I^2 \\ \quad +\{0.5D^8 + 0.5D^{40}\}I^3 \\ \alpha = \{0.250D^4 + 0.5D^{20} + 0.250D^{36}\}I^0 \\ \quad +\{0.750D^4 + 0.750D^{20} + 0.250D^{36} + 0.250D^{52}\}I^1 \\ \quad +\{0.5D^4 + 0.250D^{20} + 0.250D^{52}\}I^2 \\ \beta = \{1.0D^0\}I^0 \\ \quad +\{1.0D^{16} + 1.0D^{32}\}I^1 + \{1.0D^{16}\}I^2 \end{cases}$$

$$\begin{aligned} T(D) &= T(D,I)|_{I=1} \\ &= 7.594D^{20}\{1.0D^0 + 3.0D^4 + 9.0D^8 + 27.0D^{12} + 82.0D^{16} + \cdots\} \\ \frac{1}{m}\frac{\partial T(D,I)}{\partial I}\Big|_{I=1} &= 13.641D^{20}\{1.0D^0 + 4.206D^4 + 15.402D^8 + 54.557D^{12} + 189.661D^{16} + \cdots\} \end{aligned}$$

With $X = Y$, we finally obtain:

$$\begin{aligned} T(D,Y)|_{Y=1} &= 7.594D^{20}\{1.0D^0 + 3.0D^4 + 9.0D^8 + 27.0D^{12} + 82.0D^{16} + \cdots\} \\ \frac{\partial T(D,Y)}{\partial Y}\Big|_{Y=1} &= 22.781D^{20}\{1.0D^0 + 4.5D^4 + 18.0D^8 + 67.5D^{12} + 244.0D^{16} + \cdots\} \end{aligned}$$

4.2.2 Eight parallel transitions

Optimum mapping at the third order

The optimum mapping of Figure 4.6 was derived from Figure 2.13. It contains 4

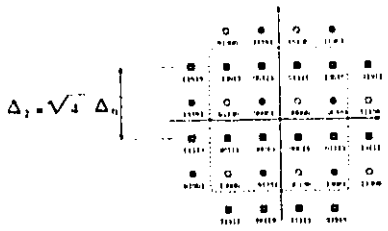

Figure 4.6: 32-CROSS optimum mapping used for the TCM $(2, 1, 2)$ code with $m = 4$.

subsets of 8 points.

With $X = I$, some of the elements of the weight function matrices $\mathbf{W}(E^L)$ are:

$$
\begin{aligned}
\mathbf{W}(00000) &= 1.0D^0 \\
\mathbf{W}(00001) &= \mathbf{W}(00011) = 0.125D^4 + 0.250D^{20} + 0.125D^{36} + 0.250D^{52} + 0.250D^{68} \\
\mathbf{W}(00010) &= 0.125D^8 + 0.250D^{40} + 0.125D^{72} + 0.250D^{104} + 0.250D^{136} \\
\mathbf{W}(00100) &= 0.750D^{16} + 0.250D^{80} \\
\mathbf{W}(11100) &= 1.0D^{16}
\end{aligned}
$$

$$
\left\{
\begin{aligned}
a &= \{0.125D^8 + 0.250D^{40} + 0.125D^{72} + 0.250D^{104} + 0.250D^{136}\}I^1 \\
&\quad +\{1.750D^8 + 0.5D^{40} + 0.5D^{72} + 0.250D^{104}\}I^2 \\
&\quad +\{1.0D^8 + 2.0D^{40}\}I^3 \\
&\quad +\{0.5D^{40} + 0.250D^{72} + 0.250D^{104}\}I^4 \\
\alpha &= \{0.125D^4 + 0.250D^{20} + 0.125D^{36} + 0.250D^{52} + 0.250D^{68}\}I^0 \\
&\quad +\{0.5D^4 + 1.125D^{20} + 0.250D^{36} + 0.375D^{52} + 0.250D^{68} + 0.5D^{100}\}I^1 \\
&\quad +\{0.875D^4 + 0.5D^{20} + 0.375D^{36} + 0.5D^{52} + 0.375D^{68} + 0.125D^{100} + 0.250D^{116}\}I^2 \\
&\quad +\{0.125D^4 + 0.375D^{20} + 0.125D^{36} + 0.125D^{52} + 0.125D^{68} + 0.125D^{100}\}I^3 \\
\beta &= \{1.0D^0\}I^0 \\
&\quad +\{0.750D^{16} + 1.0D^{32} + 0.750D^{64} + 0.250D^{80} + 0.250D^{128}\}I^1 \\
&\quad +\{0.750D^{16} + 0.750D^{32} + 0.250D^{64} + 1.250D^{80}\}I^2 \\
&\quad +\{1.0D^{16}\}I^3
\end{aligned}
\right.
$$

$$
\begin{aligned}
T(D) &= T(D,I)|_{I=1} \\
&= 13.432D^{20}\{1.0D^0 + 3.250D^4 + 10.562D^8 + 34.328D^{12} + 112.951D^{16} + \cdots\} \\
\frac{1}{m}\frac{\partial T(D,I)}{\partial I}\Big|_{I=1} &= 20.9D^{20}\{1.0D^0 + 4.616D^4 + 18.591D^8 + 72.087D^{12} + 273.549D^{16} + \cdots\}
\end{aligned}
$$

With $X = Y$,

$$
\begin{aligned}
T(D,Y)|_{Y=1} &= 13.432D^{20}\{1.0D^0 + 3.250D^4 + 10.562D^8 + 34.328D^{12} + 112.951D^{16} + \cdots\} \\
\frac{\partial T(D,Y)}{\partial Y}\Big|_{Y=1} &= 40.295D^{20}\{1.0D^0 + 4.875D^4 + 21.125D^8 + 85.820D^{12} + 336.085D^{16} + \cdots\}
\end{aligned}
$$

Optimum mapping at the first order only

The optimum mapping at the first order only was derived from Figure 2.11, and contains 2 subsets of 16 points.

The calculation provided the same results. This was expected: the 32-CROSS mappings differed from their Δ_4'', when the sub-constellations corresponding to parallel transitions are composed of two elements, that is, if among the 4 in-bits entering the TCM encoder, 3 had been used to process the coded out-bit $y^{(0)}$. In this case, the last one, unused by the convolutional code, would create two parallel transitions, and the constellation would consist of 16 subsets of 2 points. This would only be possible if we used a $(n = 4, k = 3, \nu)$ TCM encoder.

4.2.3 Sixteen parallel transitions

For the same reason, all 64-QASK mappings gave the same results here. The optimum mapping of Figure 4.7 was derived from Figure 2.22. It contains 2 subsets of 32 points.

$$
\begin{aligned}
T(D) &= T(D,I)|_{I=1} \\
&= 16.413D^{20}\{1.0D^0 + 3.5D^4 + 12.250D^8 + 42.875D^{12} + 151.562D^{16} + \cdots\} \\
\frac{1}{m}\frac{\partial T(D,I)}{\partial I}\Big|_{I=1} &= 23.782D^{20}\{1.0D^0 + 4.742D^4 + 20.1D^8 + 82.610D^{12} + 333.642D^{16} + \cdots\}
\end{aligned}
$$

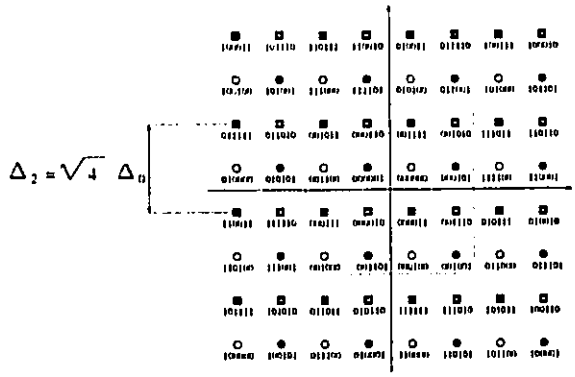

Figure 4.7: 64-QASK optimum mapping used for the TCM $(2, 1, 2)$ code with $m = 5$.

With $X = Y$:

$$T(D,Y)|_{Y=1} = 16.413D^{20}\{1.0D^0 + 3.5D^4 + 12.250D^8 + 42.875D^{12} + 151.562D^{16} + \cdots\}$$
$$\frac{\partial T(D,Y)}{\partial Y}\Big|_{Y=1} = 49.239D^{20}\{1.0D^0 + 5.250D^4 + 24.5D^8 + 107.187D^{12} + 451.689D^{16} + \cdots\}$$

4.3 Eight state encoder $(n = 3, k = 2, \nu = 3)$

The sizes of the previous results are huge and series expansions only allow to report them here. The program requires a lot of memory. More precisely, the inversion of a 7×7 matrix, of which the elements are polynomials of the two variables I and D, required such a long time and so much memory, that only a truncated version of the program was run. To do so, the polynomials of the form: $t(D)X^x$ were truncated with regards to the exponent of D in the polynomial $t(D)$, for all the elements of vectors $\mathcal{I}$ and $\mathcal{O}$, and matrix $\mathcal{G}$. With no parallel transition, the complete expression was calculated. With two or more transitions, the calculation has been truncated.

4.3.1 No parallel transitions

We used the 8-CROSS mapping described in Figure 4.8 which contains 8 subsets of 1 point.

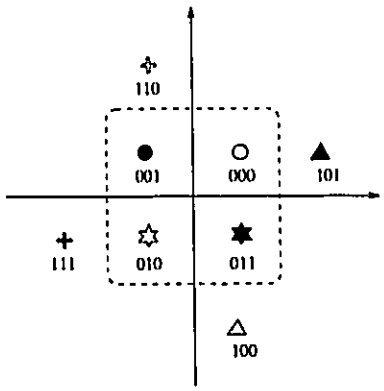

Figure 4.8: 8-CROSS optimum mapping used for the TCM $(3,2,3)$ code with $m = 2$.

$$
\begin{aligned}
T(D) &= 2.25D^{20}\{1.0D^0 + 2.0D^4 + 4.0D^8 + 8.0D^{12} + 17.0D^{16} + \cdots\} \\
\frac{1}{m}\left.\frac{\partial T(D,I)}{\partial I}\right|_{I=1} &= 5.4375D^{20}\{1.0D^0 + 2.207D^4 + 5.241D^8 + 12.138D^{12} + 28.598D^{16} + \cdots\} \\
\left.\frac{\partial T(D,Y)}{\partial Y}\right|_{Y=1} &= 6.75D^{20}\{1.0D^0 + 3.0D^4 + 8.0D^8 + 20.0D^{12} + 49.0D^{16.0} + \cdots\}
\end{aligned}
$$

4.3.2 Two parallel transitions

We used the 16-QASK mapping described in Figure 4.9. It contains 8 subsets of 2 points.

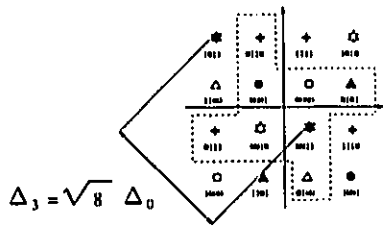

Figure 4.9: 16-QASK optimum mapping used for the TCM $(3,2,3)$ code with $m = 3$.

$$
\begin{aligned}
T(D) &= 7.594D^{20}\{1.0D^0 + 3.0D^4 + 9.0D^8 + 27.0D^{12.0} + \cdots\} \\
\frac{1}{m}\left.\frac{\partial T(D,I)}{\partial I}\right|_{I=1} &= 16.172D^{20}\{1.0D^0 + 3.548D^4 + 12.991D^8 + 46.016D^{12.0} + \cdots\} \\
\left.\frac{\partial T(D,Y)}{\partial Y}\right|_{Y=1} &= 22.781D^{20}\{1.0D^0 + 4.5D^4 + 18.0D^8 + 67.501D^{12} + \cdots\}
\end{aligned}
$$

4.3.3 Four parallel transitions

Optimum mapping at the third order

We used the 32-CROSS mapping described in Figure 4.10. It contains 8 subsets of 4 points.

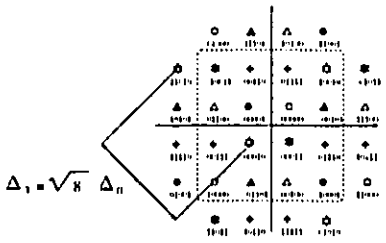

Figure 4.10: 32-CROSS optimum mapping used for TCM code $(3, 2, 3)$ with $m = 4$.

$$
\begin{aligned}
T(D) &= 13.432D^{20}\{1.0D^0 + 3.25D^4 + 10.562D^8 + 34.329D^{12.0} + \cdots\} \\
\tfrac{1}{m}\left.\tfrac{\partial T(D,I)}{\partial I}\right|_{I=1} &= 24.258D^{20}\{1.0D^0 + 3.977D^4 + 16.017D^8 + 62.109D^{12.0} + \cdots\} \\
\left.\tfrac{\partial T(D,Y)}{\partial Y}\right|_{Y=1} &= 40.295D^{20}\{1.0D^0 + 4.875D^4 + 21.125D^8 + 85.823D^{12} + \cdots\}
\end{aligned}
$$

Optimum mapping at the first order only

Each sub-constellation is a subset of 8 points. The minimum distance between its points is $\Delta_1 = \sqrt{2}\Delta_0$ and the only improvement we made on the 32-QAM mapping was effective on Δ_4'', that is for subsets of 2 points. This explains why we here ended up with the same results. In fact, with 2 in-bits used to process the redundant coded out-bit, all the mappings remain optimum.

4.3.4 Eight parallel transitions

Optimum mapping at the third order

We used the 64-CROSS mapping described in Figure 4.11. It contains 4 subsets of 16 points.

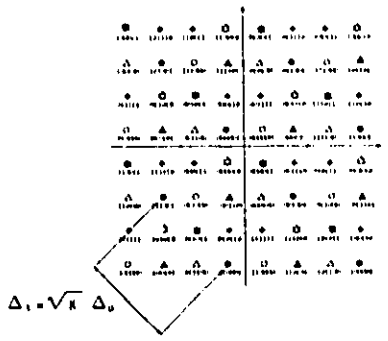

Figure 4.11: 64-QASK optimum mapping used for the TCM $(3,2,3)$ code with $m = 5$.

$$
\begin{aligned}
T(D) &= 16.413D^{20}\{1.0D^0 + 3.5D^4 + 12.25D^8 + 42.875D^{12.0} + \cdots\} \\
\frac{1}{m}\frac{\partial T(D,I)}{\partial I}\Big|_{I=1} &= 27.065D^{20}\{1.0D^0 + 4.167D^4 + 17.662D^8 + 72.589D^{12.0} + \cdots\} \\
\frac{\partial T(D,Y)}{\partial Y}\Big|_{Y=1} &= 49.239D^{20}\{1.0D^0 + 5.25D^4 + 24.5D^8 + 107.186D^{12} + \cdots\}
\end{aligned}
$$

Optimum mappings at the first order only

The improvement from optimum mappings at the second order only to the optimum mappings at the third order for a 64-QAM were made on Δ_4, which implies that it would have been effective for a mapping composed of 16 subsets of 4 points. Again, to verify the improvement, the TCM scheme requires more than 2 bits to be used for the computation of the redundant out-bit. The computed transfer function is kept the same here.

4.4 Sixteen state encoder $(n = 3, k = 2, \nu = 4)$

4.4.1 No parallel transitions

We used the 8-CROSS mapping described in Figure 4.8

$$
\begin{aligned}
T(D) &= 2.25D^{24}\{1.0D^0 + 4.0D^8 + \cdots\} \\
\frac{1}{m}\frac{\partial T(D,I)}{\partial I}\Big|_{I=1} &= 6.0D^{24}\{1.0D^0 + 5.5D^8 + \cdots\} \\
\frac{\partial T(D,Y)}{\partial Y}\Big|_{Y=1} &= 11.25D^{24}\{1.0D^0 + 6.2D^8 + \cdots\}
\end{aligned}
$$

4.4.2 Two parallel transitions

We used the 16-QASK mapping described in Figure 4.9

$$\begin{aligned} T(D) &= 11.391 D^{24} \{1.0 D^0 + 9.0 D^8 + \cdots\} \\ \frac{1}{m} \frac{\partial T(D,I)}{\partial I}\Big|_{I=1} &= 28.689 D^{24} \{1.0 D^0 + 12.97 D^8 + \cdots\} \\ \frac{\partial T(D,Y)}{\partial Y}\Big|_{Y=1} &= 56.953 D^{24} \{1.0 D^0 + 13.950 D^8 + \cdots\} \end{aligned}$$

4.4.3 Four parallel transitions

We used the 32-CROSS mapping described in Figure 4.10

$$\begin{aligned} T(D) &= 21.826 D^{24} \{1.0 D^0 + 10.563 D^8 + \cdots\} \\ \frac{1}{m} \frac{\partial T(D,I)}{\partial I}\Big|_{I=1} &= 48.234 D^{24} \{1.0 D^0 + \cdots\} \\ \frac{\partial T(D,Y)}{\partial Y}\Big|_{Y=1} &= 109.132 D^{24} \{1.0 D^0 + 16.372 D^8 + \cdots\} \end{aligned}$$

4.4.4 Eight parallel transitions

We used the 64-QASK mapping described in Figure 4.11 but the other optimum mappings at the first order only led to the same results.

$$\begin{aligned} T(D) &= 28.723 D^{24} \{1.0 D^0 + 12.25 D^8 + \cdots\} \\ \frac{1}{m} \frac{\partial T(D,I)}{\partial I}\Big|_{I=1} &= 56.391 D^{24} \{1.0 D^0 + \cdots\} \\ \frac{\partial T(D,Y)}{\partial Y}\Big|_{Y=1} &= 143.615 D^{24} \{1.0 D^0 + 18.988 D^8 + \cdots\} \end{aligned}$$

Because of the large amount of memory space which was required, the program could not derive any further results. Otherwise, this method provides the exact values of $N_c(dfree)$, $N_S(dfree)$ and $N_b(d_{free})$, and the second term of the transfer functions of all the Ungerboeck codes. In conclusion, Tables 6.3, 6.5 and 6.7 give the results of our theoretical investigation and some corresponding theoretical curves of probability are given.

4.5 Conclusion

In chapter 4, the transfer functions of some trellis code have been computed. To allow for the use of the *uniformity condition*, it was necessary to point out the existence of the conditional isometry in the constellations. The simplified algorithm for their calculation was then applied. They provide coefficients for the first two terms of their series expansion which will be useful to derive second order expression for event, symbol and bit probability of error.

Because of the large amount of memory space that the programs required, trellis codes with memory length only up to 8 have been dealt with. If needed, for higher than 8 state encoder, the parameters given by Ungerboeck in [12] will be taken later. It should be noted that they give only the first term and the average number of closest neighbors N_{free} is only known when the number of in-bits m tends towards infinity.

Chapter 5

64-QAM Trellis Coded Modulation Scheme

5.1 Introduction

For transmission schemes which require an even numbered spectral efficiency, for instance $4b/s/Hz$, we may think about different ways to use TCM codes. The information bits may be separated and coded such as to form independent sequences of in-phase and quadrature symbols. The two orthogonal mappings would be unidimensional and superimposed in order to form a perfect square for QAM constellation mappings. At the output of the channel, two Viterbi decoders in parallel would be used for decoding in-phase and quadrature signal components. For an input symbol composed of $2m$ bits, the output symbol from the TCM encoder would contain $2m+2$-bits. The redundancy of such a scheme is higher than for the previous 2^{2m+1} one, but it requires more transmit energy for a given bit error rate. Figure 5.1 illustrates how two unidimensional 8-ASK codes can be combined in quadrature to provide a 64-QAM TCM code.

This requires the design of optimum mappings, as studied in section 5.2. Ungerboeck provided an exhaustive list of TCM codes for different kind of modulations. For instance, he gave different codes for PAM, with $k = 1$. We decided to use the same

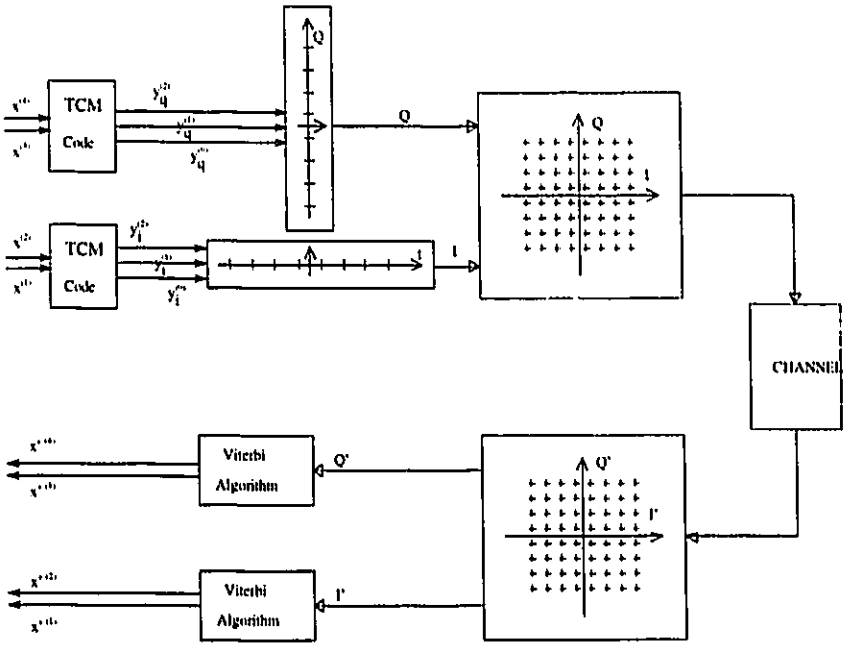

Figure 5.1: Example of two TCM 8-ASK codes combined in quadrature to form a 64-QAM TCM transmission scheme.

TCM codes as for the two-dimensional mapping. It is likely that their performances are less than the Ungerboeck ones for PAM. But this approach allows for the comparison of transmissions systems. The only point is that we need to evaluate theoretical performances of the TCM QAM codes applied on ASK mappings, for PAM. This is provided in section 5.4.

5.2 Design of optimum mappings

Figure 5.2 shows the construction of a possible optimum mapping used for a unidimensional 8-ASK. We may already note that whether the memory length of the number of input bits used by the TCM encoder to process the redundant coded out-bit, the sub-constellations have respectively two signal points ($y^{(1)}$ and $y^{(0)}$ determine them) or a single one (the three bits $y^{(2)}y^{(1)}y^{(0)}$ form them).

The question of extending optimum mappings is not discussed here. We have constructed an optimum mapping for 8-ASK and so as to include an optimum 4-ASK

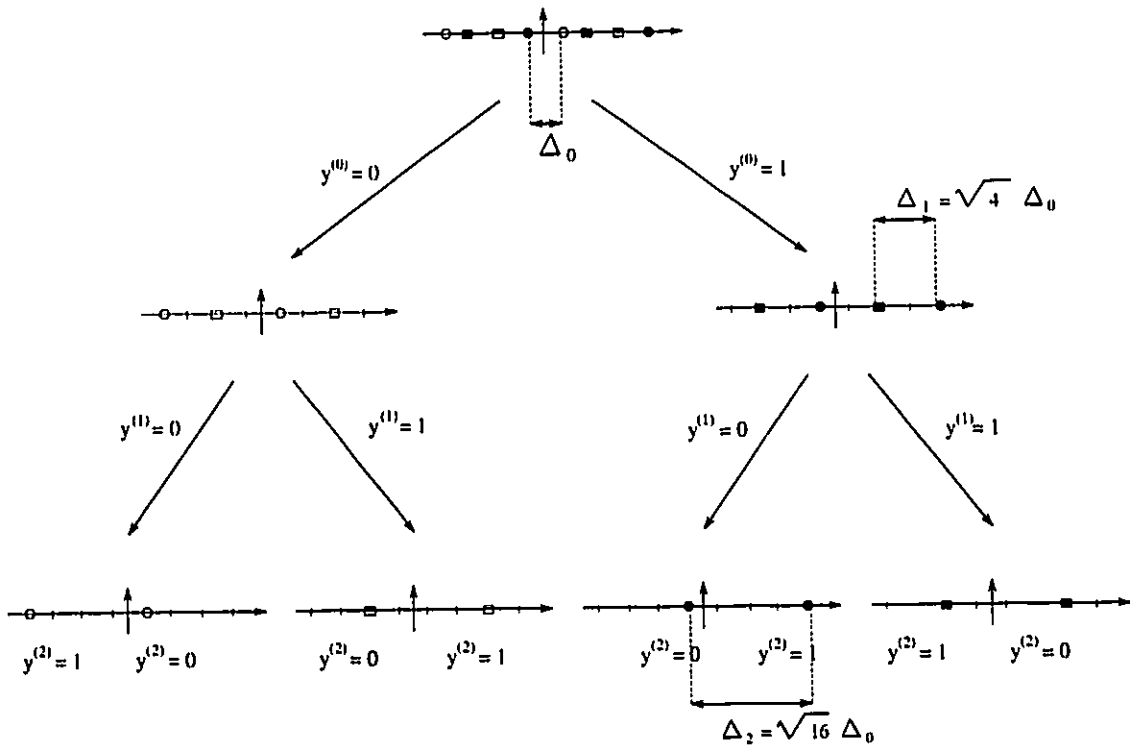

Figure 5.2: Construction of optimum mapping for unidimensional 8-ASK.

mapping. We chose not to go further since a 16-ASK unidimensional mapping would result in a global 256-QAM which does not meet the parameters of the case study on which all the simulations have been run, i.e. HDTV, as reported in chapter 7. In chapters 2 and 4, we only dealt with up to a 64-QASK mapping, which corresponds to a two 8-ASK unidimensional mapping in quadrature.

5.2.1 One input bit TCM code

The TCM code considered here is a $(n = 2, k = 1, \nu = 2)$ code. The final optimum mapping consists of 4 subsets of 1 point. It is represented in Figure 5.3.

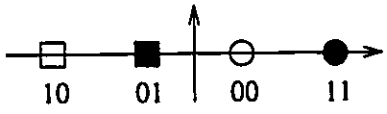

Figure 5.3: Final optimum mapping for unidimensional 4-ASK and $m = 1$.

5.2.2 Two input bit TCM code

If the TCM code is a $(n = 2, k = 1, \nu = 2)$ code, then the final mapping consists of 2 subsets of 4 points, for the trellis is made of two parallel transitions from one state to any other. This is represented in Figure 5.4.

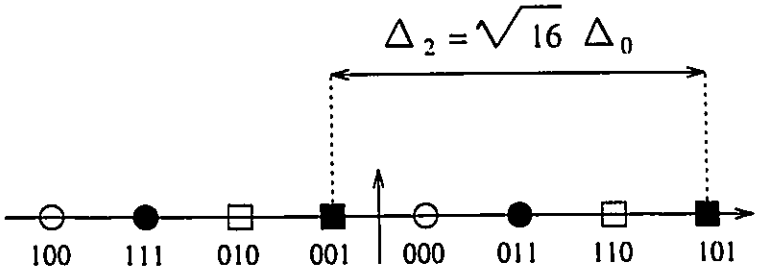

Figure 5.4: Final optimum mapping for unidimensional 8-ASK and $m = 2$, and $k = 1$.

Note that the previous 4-ASK mapping from Figure 5.3 consists of the 4 central points of the 8-ASK mapping from Figure 5.4: the latter is the extension of the former. We also read on Figure 5.4 the value of Δ_2, the minimum distance between parallel transitions:

$$\Delta_2 = 4\Delta_0 = \sqrt{16}\Delta_0$$

This particularly high value surely prevents parallel transitions from reducing performances of the TCM (2,1,2) code with an 8-ASK over the improvement that the trellis performs.

If we use any other TCM code, of the form $(n = 3, k = 2, \nu)$, the trellis code has no parallel transition and the corresponding mapping is represented in Figure 5.5. There are 8 subsets of 1 point.

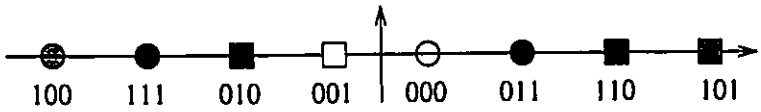

Figure 5.5: Final optimum mapping for unidimensional 8-ASK and $m = 2$ $(k = 2)$.

Note that extending the 8-ASK signal constellation would give $\Delta_3 = \sqrt{64}\Delta_0$, which definitely cannot limit any $(3, 2, \nu)$ TCM code performance on a 16-ASK or

more.

5.3 Theoretical performance evaluation for the new TCM codes

Let $\mathcal{P}'_e$ and $\mathcal{P}'_b$ be the probabilities of event error and bit error respectively.

Since the decoding scheme is the same for the in-phase and quadrature symbols, the bit error probability $\mathcal{P}'_b$ for our transmission with a two level TCM code is equal to the bit error probability $\mathcal{P}_b$ of a single TCM code with unidimensional mapping.

$$\mathcal{P}'_b = \mathcal{P}_b \tag{5.1}$$

As far as an error event is concerned, we may consider the reconstructed complete sequence of input symbols and decoded output symbols. An error event occurs for such a sequence if either an error event occurs on the in-phase component of the symbols, or on the quadrature component of the symbols. Because the error events on both components may occur simultaneously, the probability of an event error for the entire sequence of symbols is bounded according to Equation (5.2):

$$\begin{aligned} \mathcal{P}'_e &\leq \mathcal{P}_e(S_I) + \mathcal{P}_e(S_Q) \\ &\leq 2\mathcal{P}_e \end{aligned} \tag{5.2}$$

Similarly:

$$\begin{aligned} \mathcal{P}'_S &\leq \mathcal{P}_S(S_I) + \mathcal{P}_S(S_Q) \\ &\leq 2\mathcal{P}_S \end{aligned} \tag{5.3}$$

If we expect to compare this model with the 32-QAM mapping, we have to normalize the average signal-to-noise ratio ρ to $E_S^{(64-QASK)}$ instead of $E_S^{(8-ASK)}$: Equation (5.4) has to be rewritten into Equation (5.5).

$$\begin{aligned} \mathcal{P}_{cod} &= N_{cod}\, \mathrm{Q}\left(\sqrt{\alpha_{cod}\rho}\right) \\ \alpha_{cod} &= \frac{d^2_{free}}{2E_S^{(8-ASK)}} \\ \rho &= \frac{E_S^{(8-ASK)}}{N_0} \end{aligned} \tag{5.4}$$

$$
\begin{aligned}
\mathcal{P}^{(2)} &= \mathcal{P}^{(64-QASK_{2/3-2/3})} \\
&= cN^{(8-ASK)}\,\mathrm{Q}\left(\frac{d_{free}^{(ASK)}}{\sqrt{2N_0}}\right) \\
&= cN^{(8-ASK)}\,\mathrm{Q}\left(\frac{d_{free}^{(ASK)}}{\sqrt{2E_S^{(64-QASK)}}}\sqrt{\rho}\right) \\
&= cN^{(8-ASK)}\,\mathrm{Q}\left(\sqrt{\alpha_{cod}^{(64-QASK)}\rho}\right)
\end{aligned}
\tag{5.5}
$$

where ρ is now:

$$
\rho = \frac{E_S^{(64-QASK)}}{N_0} \tag{5.6}
$$

$$
\alpha_{cod}^{(64-QASK)} = \frac{d_{free}^{(ASK)^2}}{2E_S^{(64-QASK)}} \tag{5.7}
$$

and the coefficient c is 1 if the probability of bit error is considered or 2 if the probabilities of event or symbol error are.

Thus, once we have computed the transfer function for the unidimensional ASK TCM code, we may obtain an upper bound for both error event and bit error probabilities. However, to do so, the mappings must satisfy the *uniformity condition*.

Consider the three last Figures 5.3, 5.4 and 5.5. Recall that $\mathcal{Y}_0$ is the subset of all even symbols. It appears that its complement $\overline{\mathcal{Y}_0} = \mathcal{Y}_0 \oplus \tilde{Y}$ is its own transform through the y-axis. Therefore, the *uniformity condition* is satisfied and we may compute the transfer function as before.

5.4 Transfer function for the new TCM codes

5.4.1 Transfer function for the one input bit TCM code

The only TCM code we may use here is the TCM $(n = 2, k = 1, \nu = 2)$ code.

$$
\begin{aligned}
\mathbf{W}(00) &= 1.0D^0 \\
\mathbf{W}(01) &= 0.5D^4 + 0.5D^{36} \\
\mathbf{W}(10) &= 1.0D^{16} \\
\mathbf{W}(11) &= 1.0D^4
\end{aligned}
$$

$$
\begin{aligned}
T(D) &= 0.5D^{36}\{1.0D^0 + 2.5D^4 + 3.25D^8 + 5.125D^{12} + 7.562D^{16} + \cdots\} \\
\frac{1}{m}\frac{\partial T(D,I)}{\partial I}\Big|_{I=1} &= 1.0D^{36}\{1.0D^0 + 4.5D^4 + 6.25D^8 + 12.125D^{12} + 20.063D^{16} + \cdots\} \\
\frac{\partial T(D,Y)}{\partial Y}\Big|_{Y=1} &= 1.5D^{36}\{1.0D^0 + 3.5D^4 + 6.25D^8 + 12.375D^{12} + 22.063D^{16} + \cdots\}
\end{aligned}
$$

5.4.2 Transfer function for the two input bits TCM code

Four state encoder $(n = 2, k = 1, \nu = 2)$

$$
\begin{aligned}
\mathbf{W}(000) &= 1.0D^0 \\
\mathbf{W}(001) &= 0.25D^4 + 0.25D^{36} + 0.25D^{100} + 0.25D^{196} \\
\mathbf{W}(010) &= 0.5D^{16} + 0.5D^{144} \\
\mathbf{W}(011) &= 1.0D^4 \\
\mathbf{W}(100) &= 1.0D^{64} \\
\mathbf{W}(101) &= 0.5D^4 + 0.5D^{36} \\
\mathbf{W}(110) &= 1.0D^{16} \\
\mathbf{W}(111) &= 0.5D^{36} + 0.5D^{100}
\end{aligned}
$$

$$
\begin{aligned}
T(D) &= 1.688D^{36}\{1.0D^0 + 2.083D^4 + 3.562D^8 + 6.255D^{12} + \cdots\} \\
\frac{1}{m}\frac{\partial T(D,I)}{\partial I}\Big|_{I=1} &= 3.375D^{36}\{1.0D^0 + 2.653D^4 + 5.25D^8 + 10.577D^{12} + \cdots\} \\
\frac{\partial T(D,Y)}{\partial Y}\Big|_{Y=1} &= 5.062D^{36}\{1.0D^0 + 3.028D^4 + 6.979D^8 + 15.377D^{12} + \cdots\}
\end{aligned}
$$

Eight state encoder $(n = 3, k = 2, \nu = 3)$

$$
\begin{aligned}
T(D) &= 2.25D^{36}\{1.0D^0 + 1.562D^4 + 2.688D^8 + 4.691D^{12} + \cdots\} \\
\frac{1}{m}\frac{\partial T(D,I)}{\partial I}\Big|_{I=1} &= 4.875D^{36}\{1.0D^0 + 1.837D^4 + 3.668D^8 + 7.324D^{12} + \cdots\} \\
\frac{\partial T(D,Y)}{\partial Y}\Big|_{Y=1} &= 6.750D^{36}\{1.0D^0 + 2.417D^4 + 5.521D^8 + 11.988D^{12} + \cdots\}
\end{aligned}
$$

Sixteen state encoder $(n = 3, k = 2, \nu = 4)$

$$
\begin{aligned}
T(D) &= 1.688D^{40}\{1.0D^0 + 3.0D^8 + 9.574D^{16} + \cdots\} \\
\frac{1}{m}\frac{\partial T(D,I)}{\partial I}\Big|_{I=1} &= 4.219D^{40}\{1.0D^0 + 4.0D^8 + 16.231D^{16} + \cdots\} \\
\frac{\partial T(D,Y)}{\partial Y}\Big|_{Y=1} &= 8.438D^{40}\{1.0D^0 + 4.650D^{8.0} + \cdots\}
\end{aligned}
$$

Chapter 6

Tables of main parameters for the performance of TCM codes

We first recall that if Δ_0 denotes the minimum distance between two constellation points and d_0 half of this distance, then the values of the average energy per symbol E_S are given in Table 6.1. The calculation is provided in Appendix C.

Table 6.1: Computation of E_S for different modulation schemes.

	$\frac{E_S}{d_0^2} = 4\frac{E_S}{\Delta_0^2}$	$\frac{E_S}{\Delta_0^2}$
2^{2m}-QASK	$\frac{2}{3}\left[2^{2m}-1\right]$	$\frac{2^{2m}-1}{6}$
8-CROSS	$\frac{11}{2}$	$\frac{11}{8}$
2^{2m+1}-CROSS $(m>2)$	$\frac{2}{3}\left[\frac{31}{32}\times 2^{2m+1}-1\right]$	$\frac{\frac{31}{32}\times 2^{2m+1}-1}{6}$
2^m-ASK	$\frac{2^{2m}-1}{3}$	$\frac{2^{2m}-1}{12}$

As a comment, note that the average energy per symbol in a QAM mapping increases by a factor close to 2 on a QAM mapping, and 4 on a PAM mapping. Furthermore, the different kind of expression of E_S between a QASK constellation and a CROSS constellation leads to a slight deviation for the asymptotic gain whether we compare the improvement of a TCM CROSS QAM over an uncoded QASK one or to the one of a TCM QASK QAM over an uncoded CROSS one. This implies that the use of some CROSS mapping for uncoded QAM which may not be that obvious.

We may then derive the following Tables for TCM codes $(n = 2, k = 1, \nu = 2)$, $(n = 3, k = 2, \nu = 3)$ and $(n = 3, k = 2, \nu = 4)$. In these Tables, α_{cod} denotes $\alpha_{d_{free}}$ as given in Equations (3.75). Similarly, we use β_{cod} as $\beta_{d_{scnd}}$ and N_e, N_b, N_S, n_e, n_b and n_S without the index cod nor the variable (d_{free}) or (d_{scnd}). We simply give their appropriate values without specifying them.

6.1 Four state encoder $(n = 2, k = 1, \nu = 2)$

The calculations are different, depending on whether the trellis contains parallel transitions or not.

Table 6.2: d_{free} of TCM code $(n = 2, k = 1, \nu = 2)$.

	$(m, k = 1)$	$\left(\frac{d_{free}}{\Delta_0}\right)^2$	α_{cod}	α_{unc}	γ_{dB}
4-QASK	$m = k$	5	5	2	$10\log_{10}\left(\frac{5}{2}\right) = 3.98dB$
8-CROSS	$m = 2 > k$	4	$\frac{64}{11}$	4	$10\log_{10}\left(\frac{16}{11}\right) = 1.63dB$
2^{2m}-QASK	$m \geq 2 > k$	4	$\frac{12}{2^{2m}-1}$	$\frac{3}{\frac{31}{64}\times 2^{2m}-1}$	$10\log_{10}\left(\frac{31\times 2^{2m}-64}{16[2^{2m}-1]}\right)$
2^{2m+1}-CROSS	$m \geq 2 > k$	4	$\frac{12}{\frac{31}{16}\times 2^{2m}-1}$	$\frac{3}{2^{2m}-1}$	$10\log_{10}\left(\frac{64[2^{2m}-1]}{31\times 2^{2m}-16}\right)$
2^{2m}-QASK	$m \to \infty$	4			$10\log_{10}\left(\frac{31}{16}\right) = 2.87dB$
2^{2m+1}-CROSS	$m \to \infty$	4			$10\log_{10}\left(\frac{64}{31}\right) = 3.15dB$
4-ASK	$m = k$	9	$\frac{18}{5}$	2	$10\log_{10}\left(\frac{9}{5}\right) = 2.55dB$
2^m-ASK	$m > k$	9	$\frac{54}{2^{2m}-1}$	$\frac{24}{2^{2m}-4}$	$10\log_{10}\left(\frac{9[2^{2m}-1]}{4[2^{2m}-4]}\right)$
2^m-ASK	$m \to \infty$	9			$10\log_{10}\left(\frac{9}{4}\right) = 3.52dB$

Table 6.3 does not provide N_b nor n_b for two-dimensional mapping except for 32-QAM. It would be easy to do it with the help of the corresponding mappings. However, as explained in section 2.6, an optimization inside each constellation should first be constructed, which is not done in this thesis. However, an expression for the bit error probability will later be needed for 32-QAM. Hence, although the optimization inside each of the 4 sub-constellations of 8 points is not done, N_b has been counted with the final 32-QAM mapping of Figure 2.13. But N_b and n_b will be provided in

Table 6.3: Parameters for performances of code $(n = 2, k = 1, \nu = 2)$.

$d_{free} = d_{free}(\nu)$		N_e	N_b	N_S	$\left(\frac{d_{next}}{\Delta_0}\right)^2$	β_{cod}	n_e	n_b	n_S
4-QASK	$m = 1 = k$	1	2	3	6	$\frac{1}{2}$	2	3	3
$d_{free} = \Delta_2$		N_e		N_S	$\left(\frac{d_{next}}{\Delta_0}\right)^2$	β_{cod}	n_e		n_S
8-CROSS	$m = 2 > k$	1		1	5	$\frac{2}{11}$	2.25		6.75
16-QASK	$m = 3 > k$	2		2	5	$\frac{1}{10}$	3.38		11.40
32-CROSS	$m = 4 > k$	$\frac{5}{2}$	$\frac{21}{20}$	$\frac{5}{2}$	5	$\frac{1}{20}$	5.37		16.12
64-QASK	$m = 5 > k$	3		3	5	$\frac{1}{40}$	5.47		16.41
QASK/CROSS	$m \to \infty$	4		4	5				
4-ASK	$m = k$	0.5	1	1.5	10	$\frac{1}{5}$	2.5	4.5	3.5
8-ASK	$m = 3 > k$	1.69	4.22	8.44	10	$\frac{1}{21}$	3	4	4.65

the case where both $d_{free}(\nu)$ and d_{next} are less than the minimum distance Δ_{k+1} on parallel transitions.

6.2 Eight state encoder $(n = 3, k = 2, \nu = 3)$

Since the convolutional encoder processes two bits at a time, as far as QASK and CROSS mappings are concerned, the distance between parallel transitions, if any, is equal to:

$$\Delta_{k+1} = \Delta_3 = \sqrt{8}\Delta_0$$

Hence, for the TCM codes from which we derived information on probabilities of error in the previous chapters, that is TCM (3,2,3) and (3,2,4) codes, the free distance is never reached on parallel transitions.

For the 8-ASK mapping, there are no parallel transitions and the free distance of the code is also the free distance of the trellis.

As stated earlier, in order to use the TCM PAM scheme according to the model described in chapter 5 and to compare it with ordinary TCM QAM from chapter 4, unidimensional constellations no larger than 8-ASK are needed. Tables 6.5 and 6.7

Table 6.4: d_{free} of TCM code $(n = 3, k = 2, \nu = 3)$.

	$\left(\frac{d_{free}}{\Delta_0}\right)^2$	α_{cod}	α_{unc}	γ_{dB}
8-CROSS	5	$\frac{20}{11}$	4	$10\log_{10}\left(\frac{20}{11}\right) = 2.60dB$
2^{2m}-QASK	5	$\frac{15}{2^{2m}-1}$	$\frac{3}{\frac{31}{64}\times 2^{2m}-1}$	$10\log_{10}\left(\frac{5\left[31\times 2^{2m}-64\right]}{64\left[2^{2m}-1\right]}\right)$
2^{2m+1}-CROSS	5	$\frac{15}{\frac{31}{16}\times 2^{2m}-1}$	$\frac{3}{2^{2m}-1}$	$10\log_{10}\left(\frac{5\times 16\left[2^{2m}-1\right]}{31\times 2^{2m}-16}\right)$
$2^{2\infty}$-QASK	5			$10\log_{10}\left(\frac{155}{64}\right) = 3.84dB$
$2^{2\infty+1}$-CROSS	5			$10\log_{10}\left(\frac{80}{31}\right) = 4.11dB$
2^m-ASK	9	$\frac{54}{2^{2m}-1}$	$\frac{24}{2^{2m}-4}$	$10\log_{10}\left(\frac{9\left[2^{2m}-1\right]}{4\left[2^{2m}-4\right]}\right)$
2^∞-ASK	9			$10\log_{10}\left(\frac{9}{4}\right) = 3.52dB$

provide the codes for various quadrature constellations parameters.

Table 6.5: Parameters for performances of code $(n = 3, k = 2, \nu = 3)$.

$d_{free} = d_{free}(\nu)$	N_e	N_b	N_S	$\left(\frac{d_{second}}{\Delta_0}\right)^2$	β_{cod}	n_e	n_b	n_S
8-CROSS	2.25	5.44	6.75	6	$\frac{2}{11}$	2	2.21	3
16-QASK	7.59	16.17	22.78	6	$\frac{1}{10}$	3	3.55	4.5
32-CROSS	13.43	24.26	40.30	6	$\frac{1}{20}$	3.25	3.98	4.88
64-QASK	16.41	27.07	49.24	6	$\frac{1}{40}$	3.5	4.17	5.25
QASK/CROSS $m \to \infty$	16			6				
8-ASK	2.25	4.88	6.75	10	$\frac{1}{21}$	1.56	1.83	2.42

6.3 Sixteen state encoder $(n = 3, k = 2, \nu = 4)$

6.4 Performance evaluation for these TCM codes

With the Tables provided in sections 6.1, 6.2, and 6.3, some theoretical performances of TCM codes may be computed and plotted. These tables are as complete and accurate as possible and as a result, the theoretical and simulation probability of

Table 6.6: d_{free} of TCM code $(n = 3, k = 2, \nu = 4)$.

	$\left(\frac{d_{free}}{\Delta_0}\right)^2$	α_{cod}	α_{unc}	γ_{dB}
8-CROSS	6	$\frac{24}{11}$	4	$10\log_{10}\left(\frac{20}{11}\right) = 2.60dB$
2^{2m}-QASK	6	$\frac{18}{2^{2m}-1}$	$\frac{3}{\frac{31}{64}\times 2^{2m}-1}$	$10\log_{10}\left(\frac{3\left[31\times 2^{2m}-64\right]}{32\left[2^{2m}-1\right]}\right)$
2^{2m+1}-CROSS	6	$\frac{18}{\frac{31}{16}\times 2^{2m}-1}$	$\frac{3}{2^{2m}-1}$	$10\log_{10}\left(\frac{6\times 16\left[2^{2m}-1\right]}{31\times 2^{2m}-16}\right)$
$2^{2\infty}$-QASK	6			$10\log_{10}\left(\frac{93}{32}\right) = 4.63dB$
$2^{2\infty+1}$-CROSS	6			$10\log_{10}\left(\frac{96}{31}\right) = 4.91dB$
2^m-ASK	10	$\frac{60}{2^{2m}-1}$	$\frac{24}{2^{2m}-4}$	$10\log_{10}\left(\frac{5\left[2^{2m}-1\right]}{2\left[2^{2m}-4\right]}\right)$
2^∞-ASK	10			$10\log_{10}\left(\frac{5}{2}\right) = 3.98dB$

Table 6.7: Parameters for performances of code $(n = 3, k = 2, \nu = 4)$.

$d_{free} = d_{free}(\nu)$	N_e	N_b	N_S	$\left(\frac{d_{second}}{\Delta_0}\right)^2$	β_{cod}	n_e	n_b	n_S
8-CROSS	2.25	6	11.25	8	$\frac{4}{11}$	4	5.5	6.2
16-QASK	11.39	28.69	56.95	8	$\frac{6}{5}$	9	12.97	13.95
32-CROSS	21.83	48.23	109.13	8	$\frac{3}{5}$	10.56	-	16.37
64-QASK	28.72	56.39	143.61	8	$\frac{2}{7}$	12.25	-	18.99
QASK/CROSS $m \to \infty$	56			8				
8-ASK	1.69	4.22	8.44	12	$\frac{2}{21}$	3	4	4.65

error curves are very close to each other. For instance, they contain the first two terms which are significant in the range of medium signal to noise ratios. They also provide exact factors for determining upperbound expressions depending on the order of the QAM. However, they are composed of parameters which have been calculated according to some derivations based on particular mathematical upperbounds (see Equations (3.32) and (3.34)). Thus, we will first describe the behavior of different types of upperbounds that have been used. To do so, we selected the TCM (2,1,2) code for a 32-QAM signal constellation.

Figure 6.1 represents the symbol error probability upperbounds for the first term using Equation (3.32) (first order "loose" approximation) and Equation (3.34) (first order "tight" approximation). With the first and second term derived from Equation (3.34), two curves are plotted: one using the exact second term (second order tight approximation) and the other using once more Equation (3.34) for the second term, (second order loose approximation), as done in section 3.8.1, to derive a synthetic expression for any combination of $d_{free}(\nu)$, d_{next} and Δ_{k+1}. Figure 6.1 also represents the symbol error probability of uncoded 16-QAM and the asymptotic TCM 32-QAM when the number of parallel transitions infinitely increases. For these two last expressions, the average energy per symbol has been normalized to the 32-QAM one so that comparisons between them are meaningful.

It first appears, when the only first term is considered, that there is a rather large deviation between the two curves of Figure 6.1. For high SNR, the tighter one is certainly more reliable. If the second term is considered, there is a significant difference, especially for medium and low SNR. The second order upperbounds are probably closer to experimental results and are really similar to first order theoretical curve only if the probability of symbol error is less than 10^{-8} or $\frac{E_S}{N_0}$ is higher than 19 dB. Such low error probabilities are difficult to obtain experimentally to demonstrate that the results would perfectly match the theory. As a consequence, from now on, only second order approximations will be used, unless the second order parameters are still unknown, as it is the case for the TCM QAM (3,2,ν) codes, when ν is

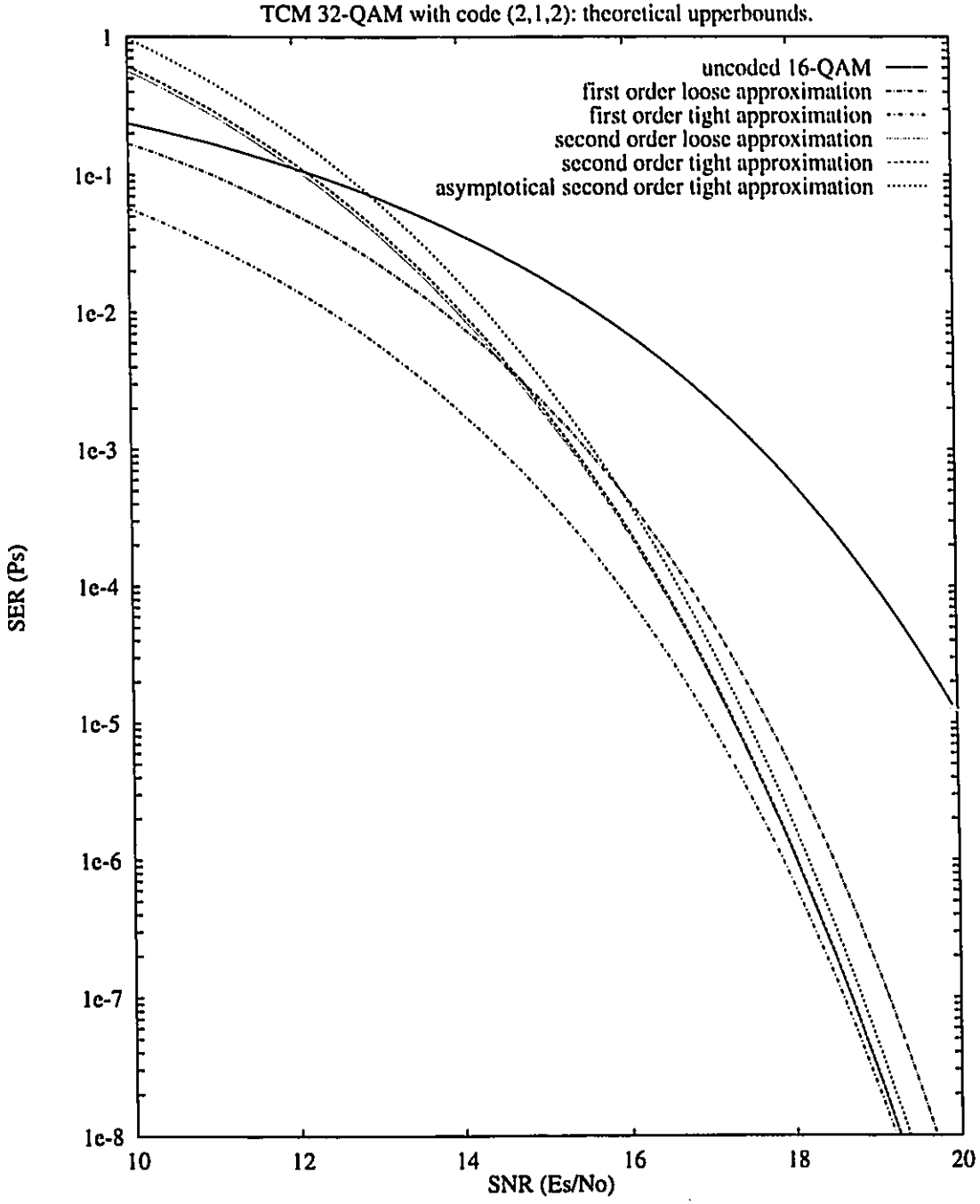

Figure 6.1: Different expressions for upperbounds of symbol error probability for a TCM 32-QAM with (2,1,2) code.

larger than 5. A second point is about the difference between the two second order expressions. They are very close, even for low SNR. This makes the synthetic second order expression given in Equation (3.74) relevant and reliable. Note that the small though non zero asymptotic relative gain between asymptotic symbol error rate, and the greater than $3dB$ asymptotic gain that may be read from the same figure. To read the asymptotic value of this gain, say $3.3dB$, a larger scale is needed, which strengthens the difficulty to make simulations match this gain. Also, the gain for 32-QAM is higher than the asymptotic one, when the number of parallel transitions tends to infinity.

We now select the TCM (3,2,3) code, for a 32-QAM constellation. Figure 6.2 presents curves for the probability for event, symbol, and bit error. TCM (2,1,2) code have no longer been chosen because in the 32-QAM case, the dominant term corresponds to error on parallel transitions. We now focus on relative expressions of bit, symbol and event errors rate on a trellis. The TCM (3,2,3) code is therefore useful, since the minimum distance on parallel transitions is greater than the second minimum distance on this trellis.

This is particularly useful to avoid any confusion between symbol error and event error. Recall that they are equal only if the minimum distance occurs on parallel transitions. It is not the case here: the ratio between symbol error rate and event error rate is 3 because on the trellis, an event error of distance d_{free} involves 3 symbols in error. As far as bit errors are concerned, there is no simple relationship between the bit error rate and the symbol error rate. However, tables from sections 6.1, 6.2, and 6.3 provide a regular relative ratio between them. For instance, for 32-QAM, the theoretical probability of bit error is equal to half the symbol error rate, for any code. This ratio seems to be constant with the considered TCM code, and this varies only with the order of the QAM signal constellation. This is extremely important, for even if experimental results do not perfectly match the theoretical ones, at least we expect their relative ratios to match the theoretical ones.

Then, we may observe the improvement of TCM when the complexity of the en-

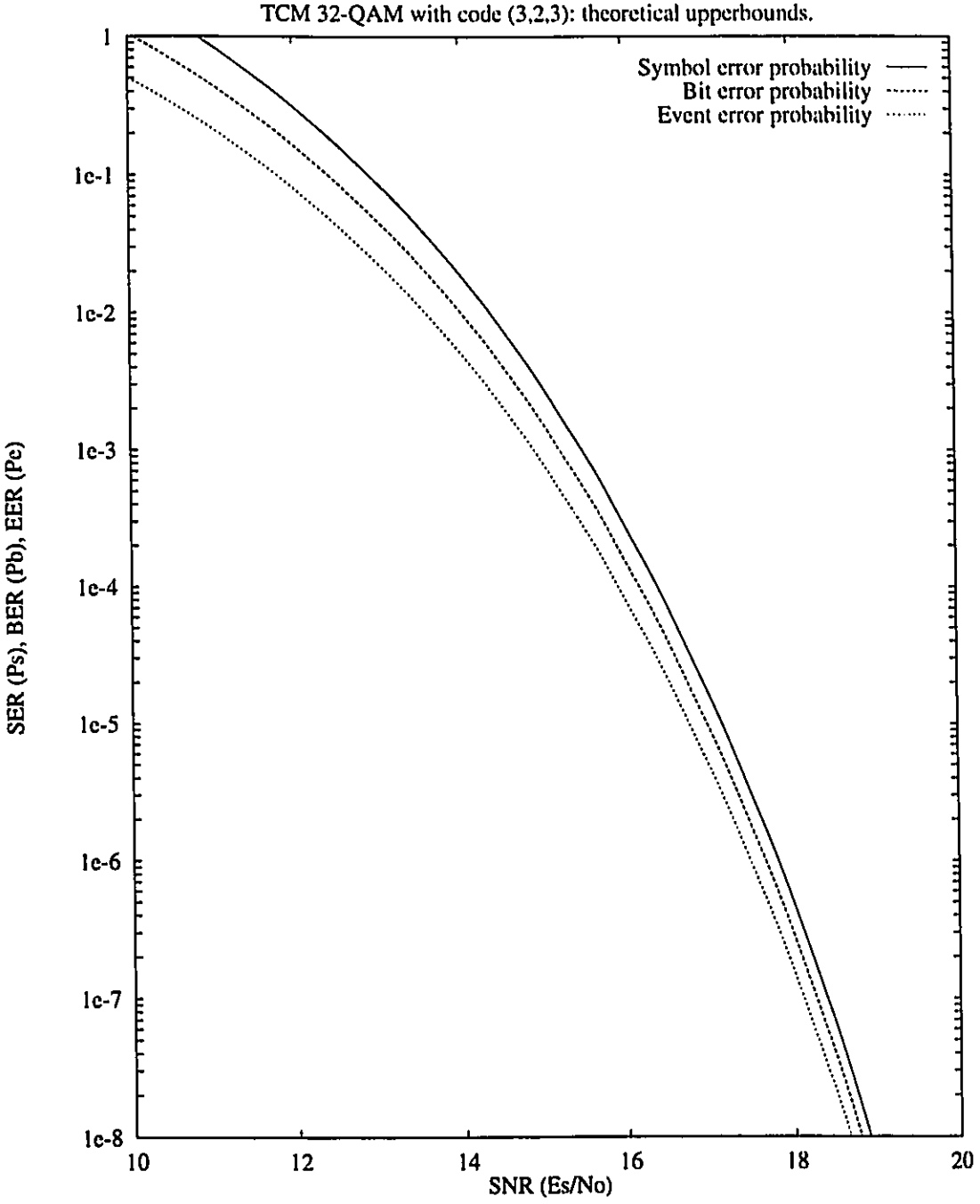

Figure 6.2: Event, symbol and bit error rates for the TCM (3,2,3) code on a 32-QAM signal constellation.

coder increases, that is, when the number of states of the encoder increases. Tables 6.5 and 6.7 provide the needed parameters for TCM (2,1,2), (3,2,3) and (3,2,4) codes. The corresponding curves of symbol error rate are plotted on Figure 6.3.

Here again, the expected asymptotic gain is only reached for high SNR. A series of curves for the probability of bit errors would result in a similar figure, except that the second order factor is still unknown.

To conclude this section, we expect to compare 32-QAM and 64-QAM 2/3-2/3 TCM schemes performances. To do so, we plotted on two different figures the probability of bit error and symbol error for TCM (3,2,3) code. Figure 6.4 presents the symbol error probability, whereas Figure 6.5 shows the bit rate.

The relative asymptotic variation between them is:

$$\begin{aligned} 10\log_{10}\left(\frac{\alpha^{(32-QAM)}}{\alpha^{(8-PAM)}}\right) &= 10\log_{10}\left(\frac{\frac{d^{(QAM)^2}_{free}}{d^{(PAM)^2}_{free}}}{\frac{E_S^{(32-CROSS)}}{E_S^{(64-QASK)}}}\right) \\ &= 10\log_{10}\left(\frac{7}{6}\right) \\ &= 0.65dB \end{aligned}$$

To express the probability of bit error in terms of the average energy per bit, we must refer to the relationship:

$$< E_b >= \frac{< E_S >}{number\ of\ information\ bits} \tag{6.1}$$

that is, here, in decibels:

$$< E_b >=< E_S > -6.02dB$$

since each symbol carries 4 information bits.

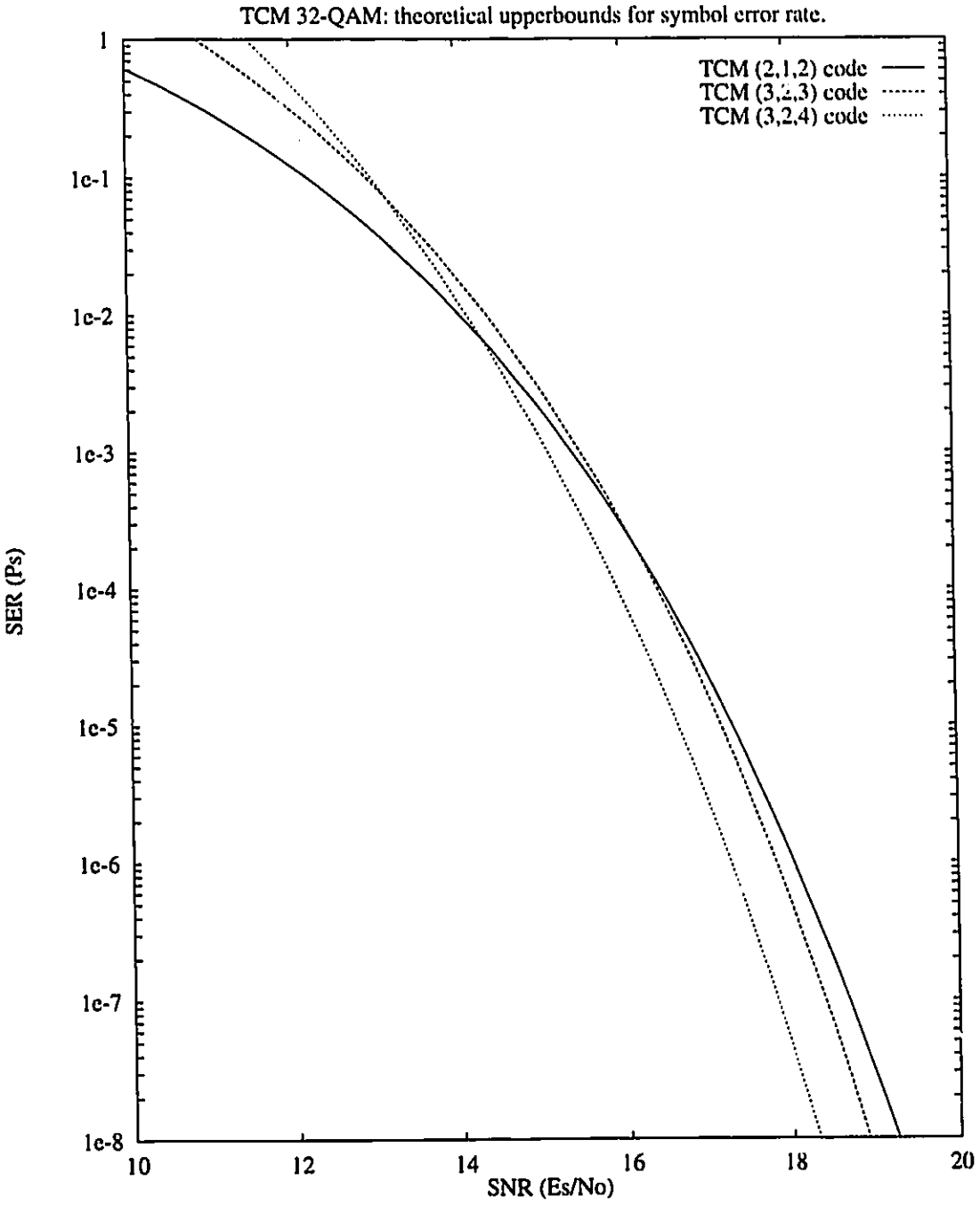

Figure 6.3: Symbol error rate for different TCM codes on a 32-QAM constellation.

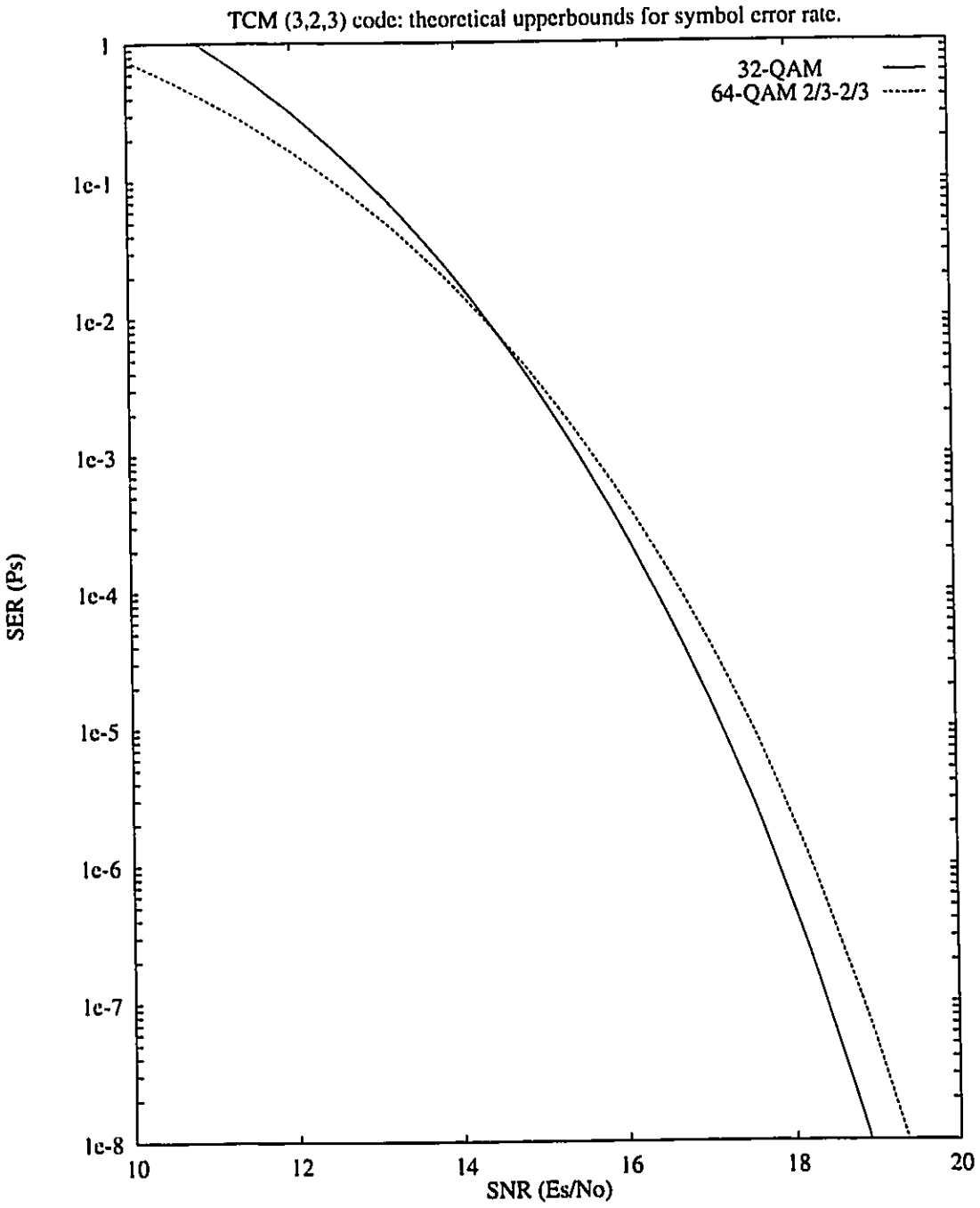

Figure 6.4: Symbol error rate of TCM (3,2,3) code for the 32-QAM and 64-QAM 2/3-2/3 TCM schemes.

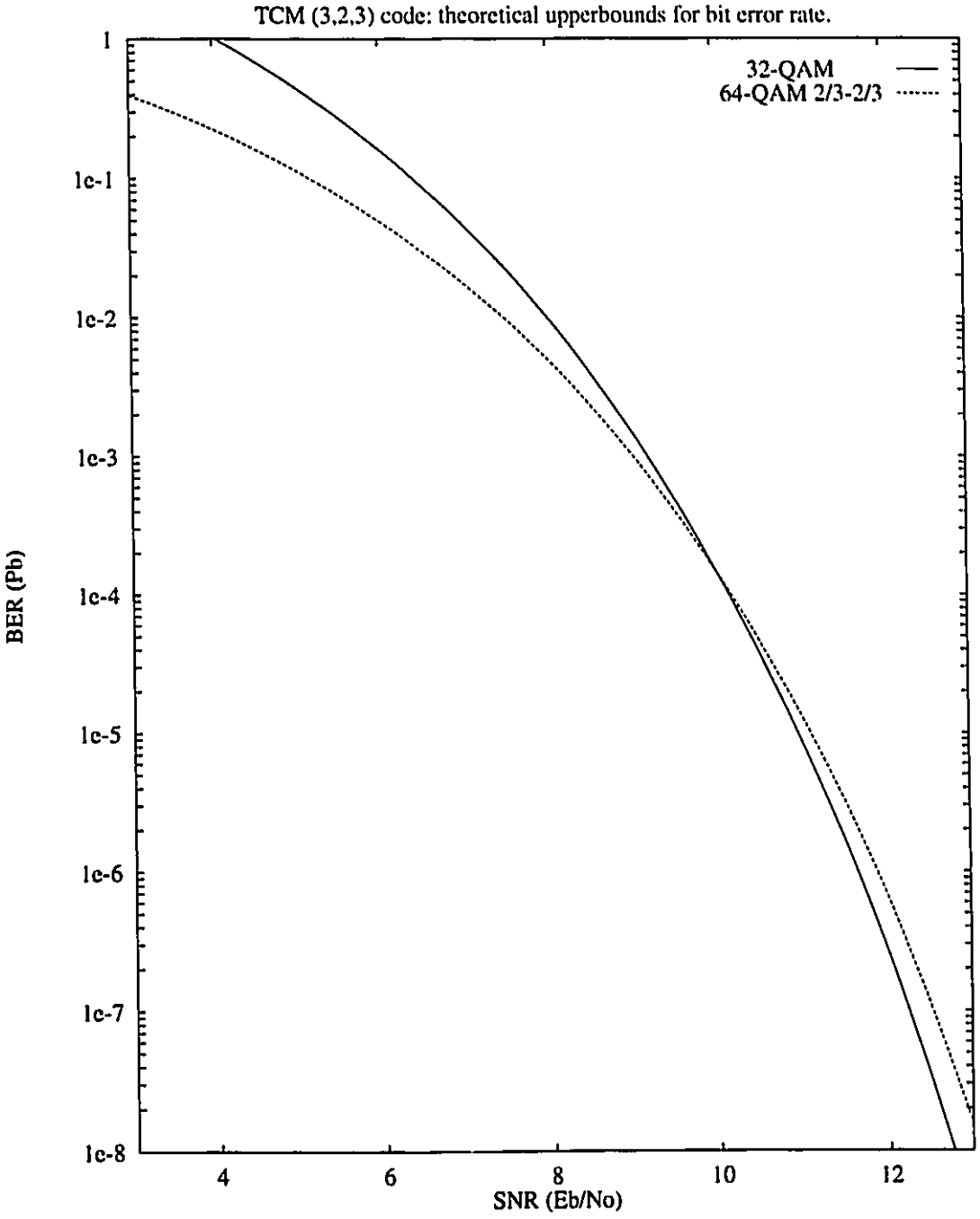

Figure 6.5: Bit error rate of TCM (3,2,3) code for the 32-QAM and 64-QAM 2/3-2/3 TCM schemes.

Chapter 7

Case study: simulation of HDTV transmission using TCM coding

7.1 Introduction

Advanced television research is about to result into a fully digital implementation. In North America, the "Grand Alliance" (composed of institutions, companies and research centers) is studying different proposed systems. The criteria of high-advanced definition quality requires a raw bit rate in the order of 1 Gb/s. These data bits are to be compressed with the use of a video source encoder. The present technology reduces the bit rate to about 15-20 Mb/s, by exploiting both spatial and temporal redundancy of the HDTV source signal, which is sufficient to provide satisfactory ATV quality service. However, this still demands very efficient error control for data transmission, because source compression techniques result in transmission errors which propagate over several data frames, in spite of error recovery techniques implemented through the video source encoder and decoder. Finally, the total data rate, including video, audio and auxiliary data channel, is around 18-20 Mb/s [13], and must fit, in North America, into a 6 MHz NTSC channel bandwidth for terrestrial broadcasting (practically 5 to 5.5 MHz, because the remaining bandwidth is used to adjust channel roll-off to prevent adjacent channel interference). The spectrum efficiency of such a

transmission system must be about 4 b/s/Hz which may be achieved by multilevel modulation techniques such as uncoded 16-QAM (assuming minimum bandwidth, or practically a zero rolloff transmission). If one transmission error per minute is tolerated [5], then a bit error rate in the order of 10^{-9} has to be achieved. With no consideration of the eventual use of error correcting codes, such a low BER, for an uncoded 16-QAM, means a very high ATV transmission power and makes it impossible the co-existence of ATV with the existing NTSC analog television services. As a conclusion, a very powerful channel coding must be implemented. To do so, the use of concatenated coding schemes is strongly needed.

The transmission model we now focus on will consist of an inner TCM code combined with an outer RS code. The simulation is run in the equivalent baseband channel model. The implementation is designed on a channel with AWGN and the possible addition of some statistically identical interference signals to measure the disturbances caused by co-channel data transmission. Note that since the simulations are performed in baseband, the co-channel interference signals are not filtered but kept as ideal QAM signals. They should not be mixed with adjacent interference signals since these wouldn't be QAM signals any more at the output of a frequency selective filter. They remain relevant if we consider a cell-partitioned space where two adjacent areas may emit different signals on the same channel frequency.

For a 4 b/s/Hz spectral efficiency, the TCM encoder maps each symbol of $m = 4$ bits into a $2^{m+1} = 32$ points constellation. The coding gain for a two-dimensional TCM code over a Gaussian channel is around 3 dB for a BER of 10^{-5}. The (N, K, t) RS code should allow us to achieve a 10^{-4} BER. RS codes are very convenient for the occurrence of burst errors at the output of the Viterbi decoder, which exploits the soft decision nature of the received signal from the noisy channel. However, the bandwidth constraint limits the number of additional symbols to less than 10% redundancy. A more precise description of RS codes is provided in section 7.3 . The additional use of interleaving techniques is expected to improve the performance of this transmission scheme, as described in section 7.4 by redistributing the errors more uniformly. But

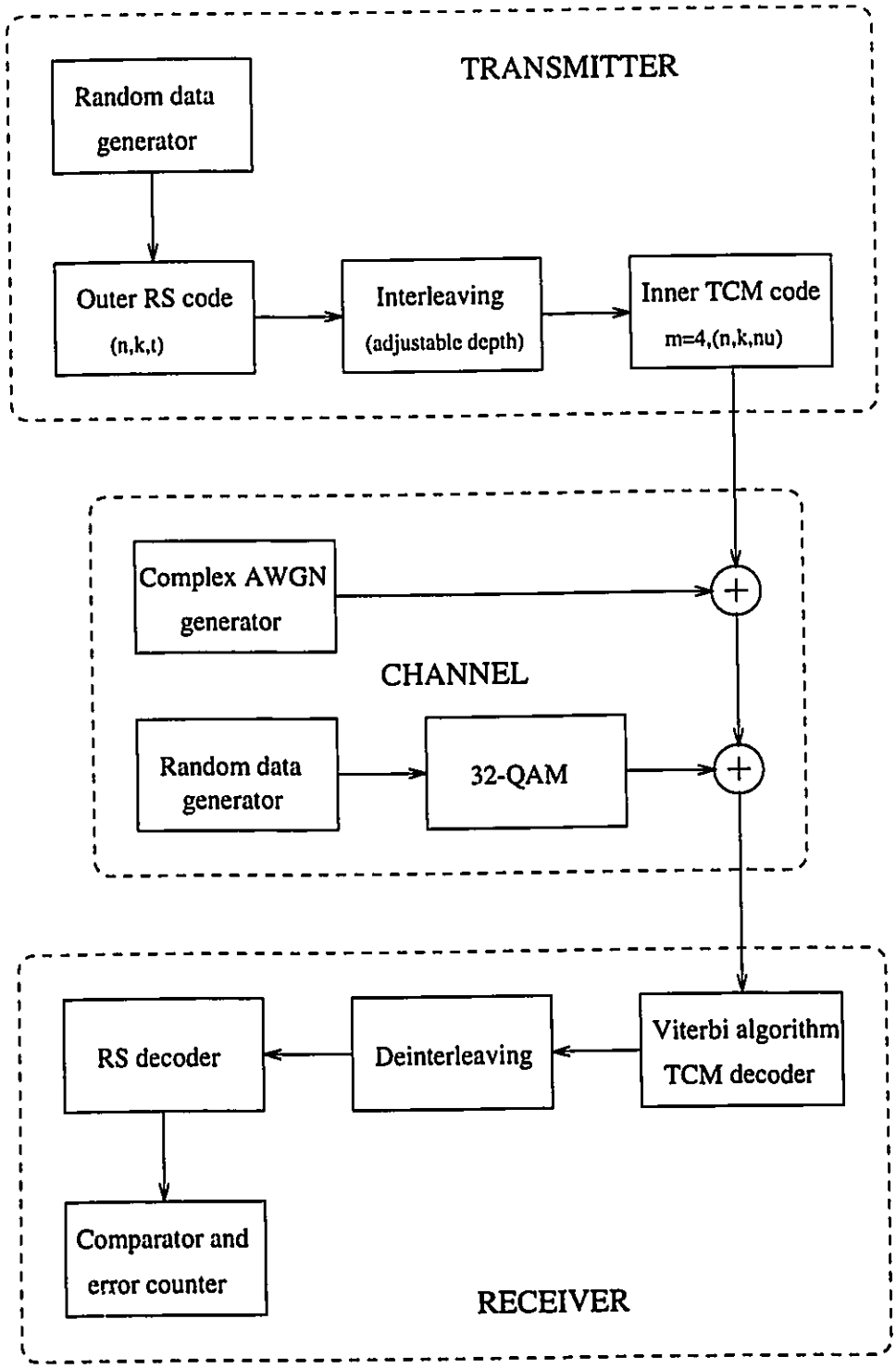

Figure 7.1: Block diagram of the equivalent baseband transmission model.

first of all, section 7.2 will provide a full description of what has been computed, as well as the results of the implementations for the TCM 32-QAM and two-level the TCM 64-QAM schemes.

7.2 Implementation of TCM codes

7.2.1 Selection of particular TCM codes

The simulation program was designed in such a way that any of the Ungerboeck codes could be chosen. However, because the simulations required extensive computing time to be run, we decided to select only two representative codes, i.e. the TCM $(3,2,3)$ and $(3,2,6)$ codes. Table 7.1 gives the main parameters of these two codes.

Table 7.1: Parameters for performances of the TCM (3,2,3) and (3,2,6) codes.

TCM code	$\frac{d^2_{free}}{\Delta_0^2}$	α_{cod}	N_e	N_b	N_S	$\frac{d^2_{next}}{\Delta_0^2}$	β_{cod}	n_e	n_b	n_S
(3,2,3) m=4	5	$\frac{1}{2}$	13.43	24.26	40.30	6	$\frac{1}{20}$	3.25	3.98	4.88
(3,2,6) m=4	7	$\frac{7}{10}$	56 ($m\to\infty$)	-	-	-	-	-	-	-

Note that the parameters of the TCM (3,2,6) code are taken from [12] whereas those of the (3,2,3) code were computed and given in chapter 4. This makes them slightly different since some use the complete and exact values of $N_e(d_{free})$, $N_b(d_{free})$ and $N_S(d_{free})$ for the TCM (3,2,3) code, while the other, $N_{e\infty}(d_{free})$, only represents the limit of $N_e(d_{free})$ when the number of parallel transitions tends towards infinity, that is when m tends towards infinity. Also, to the author's knowledge, none of $N_b(d_{free})$ or $N_S(d_{free})$ for the TCM (3,2,6) code are known. As a result, the TCM (3,2,3) code will be fully used to compare theoretical and simulation results, while the second, (3,2,6) code, will be mainly used to compare simulation performances. It is expected that differences will exist between the performances of the two systems using the two codes with different memory lengths. These are depicted in Figures 7.2 and 7.3.

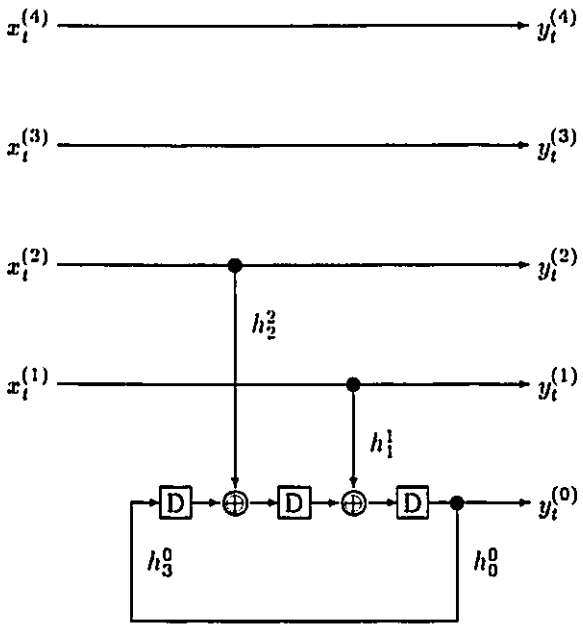

Figure 7.2: Structure of the TCM-32QAM (3,2,3) trellis code.

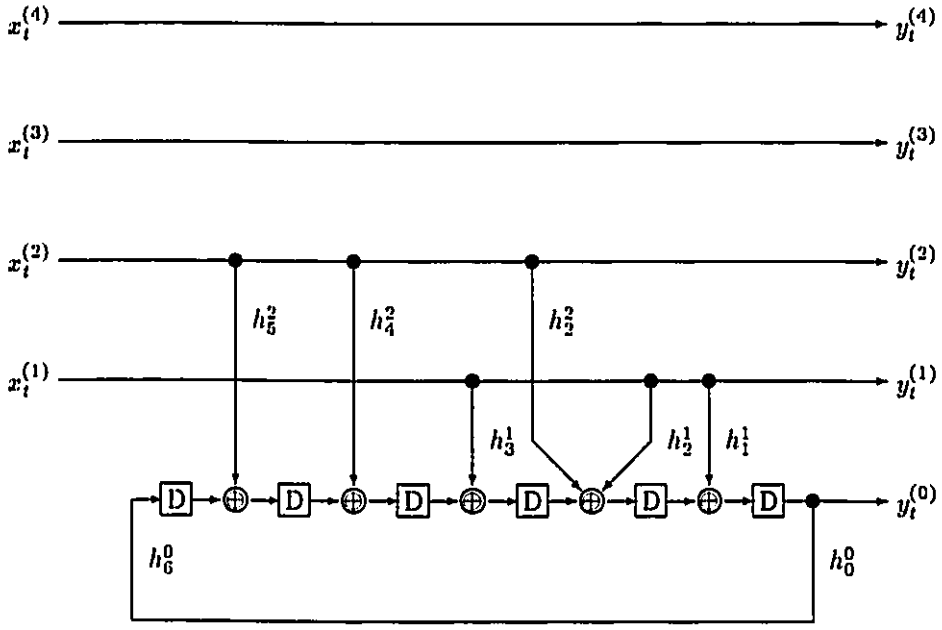

Figure 7.3: Structure of the TCM-32QAM (3,2,6) trellis code.

7.2.2 Constellations used for both transmission models

The signal constellation used is the one of Figure 7.4 and is derived from Figures 2.12 and 2.13 under the condition that there are $m - k = 2$ uncoded bits, which makes 4 parallel transitions on the trellis code and requires 8 subsets of 4 points each. Figure 7.4 represents the signal constellation for TCM 32-QAM constellation. The three bits $y_t^{(2)}$, $y_t^{(1)}$ and $y_t^{(0)}$ determine one of the eight subsets while the last two ones, $y_t^{(4)}$ and $y_t^{(3)}$, select one of the four signals in the corresponding subset, at time t:

$$
\begin{aligned}
y_t^{(2)}, y_t^{(1)}, y_t^{(0)} &= 000 \Longrightarrow \square \\
y_t^{(2)}, y_t^{(1)}, y_t^{(0)} &= 001 \Longrightarrow \star \\
y_t^{(2)}, y_t^{(1)}, y_t^{(0)} &= 010 \Longrightarrow \diamondsuit \\
y_t^{(2)}, y_t^{(1)}, y_t^{(0)} &= 011 \Longrightarrow \spadesuit \\
y_t^{(2)}, y_t^{(1)}, y_t^{(0)} &= 100 \Longrightarrow \heartsuit \\
y_t^{(2)}, y_t^{(1)}, y_t^{(0)} &= 101 \Longrightarrow \bullet \\
y_t^{(2)}, y_t^{(1)}, y_t^{(0)} &= 110 \Longrightarrow \bowtie \\
y_t^{(2)}, y_t^{(1)}, y_t^{(0)} &= 111 \Longrightarrow \clubsuit
\end{aligned}
$$

No optimization has been made inside each sub-constellation. For the four of them, which form a square, some improvement could have been made. However, it could not have any effect on the first order term of the probability of error expressions. The four others are optimized as shown in Figure 7.4.

In Figure 7.4, the most critical distance is $\Delta_3 = \sqrt{8}\Delta_0$, which is the minimum distance between points of a given sub-constellation. For both TCM (3,2,3) and (3,2,6) codes, this distance is less than the free distances of the codes $d_{free}(3,2,3) = \sqrt{5}\Delta_0$ and $d_{free}(3,2,6) = \sqrt{7}\Delta_0$: the value of d_{free} is obtained on two paths of the trellis that are not parallel transitions. As a consequence, the probability of error event is not exactly equal to the probability of symbol error but is only an upperbound for it: an event error will consist of several symbol errors. Thus, the computed values of N_S and N_b are particularly useful here. At the second order term, the TCM

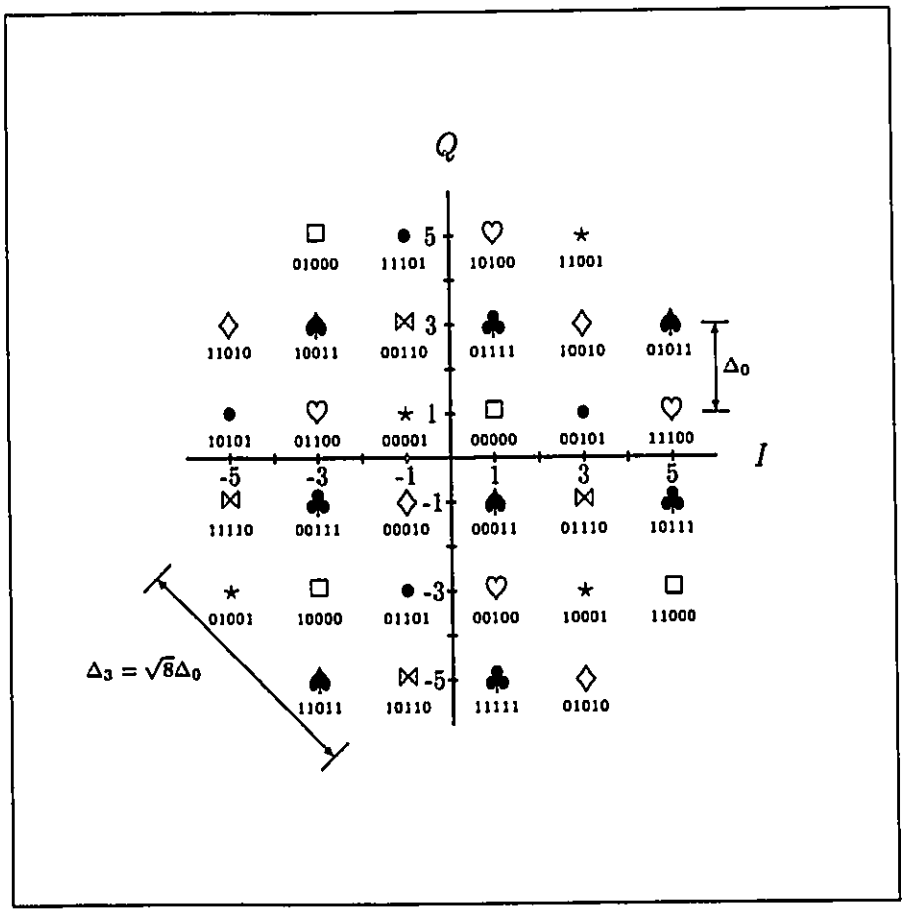

Figure 7.4: Signal constellation for the TCM 32-QAM trellis code.

(3,2,3) code is ruled by the second next minimum distance d_{next} on the trellis, where $d_{next} = \sqrt{6}\Delta_0 < \Delta_3$. This distance d_{next} is unknown at second order term for the TCM (3,2,6) code. It is likely that $d_{next}(3,2,6)$ is $\sqrt{8}\Delta_0$ or $\sqrt{9}\Delta_0$. In the first case, at the second order, both on an error on parallel transitions and error of distance d_{next} on the trellis are dominant. For the second one, only d_{next}-long distances on the trellis are dominant.

For the 64-QAM 2/3-2/3 bi-unidimensional TCM scheme, the constellation for each in-phase and quadrature component is as shown in Figure 5.5. There is no parallel transition on the corresponding trellis since the total number of bits in the TCM encoder, that is $m = 2$, is also the number of bits entering the convolutional encoder (i.e. $k = 2$).

7.2.3 Simulation results with AWGN only

Figures 7.5 and 7.6 represent both theoretical and simulation probability of symbol and bit error respectively for TCM 32-QAM and 64-QAM 2/3-2/3 with (3,2,3) code for an AWGN channel.

Closeness between theoretical and experimental curves for both $\mathcal{P}_S$ and $\mathcal{P}_b$ is satisfactory for medium and high SNR values. It is likely that for longer simulations, and higher SNR, their relative variation would have been smaller. The main constraint remains as the computation time.

Figure 7.7 depicts the symbol error rates obtained by simulation for TCM 32-QAM and 64-QAM 2/3-2/3 with (3,2,6) code. It also contains the asymptotic curve of probability of event error at first order for 32-QAM. The relatively large difference between the two 32-QAM curves, i.e. the theoretical and experimental ones, has three main reasons. First, an event error of distance $d_{free}(\nu = 7) = \sqrt{7}\Delta_0$ certainly carries at least 5 symbols, and probably around 10. Secondly, the asymptotic parameters given by Ungerboeck in [12] stands when the number of in-bits tends towards infinity. For instance N_{free}, according to [12], is equal to 56 but may not be reached for a TCM 32-QAM code. Figure 7.7 therefore emphasizes the importance of exact curves

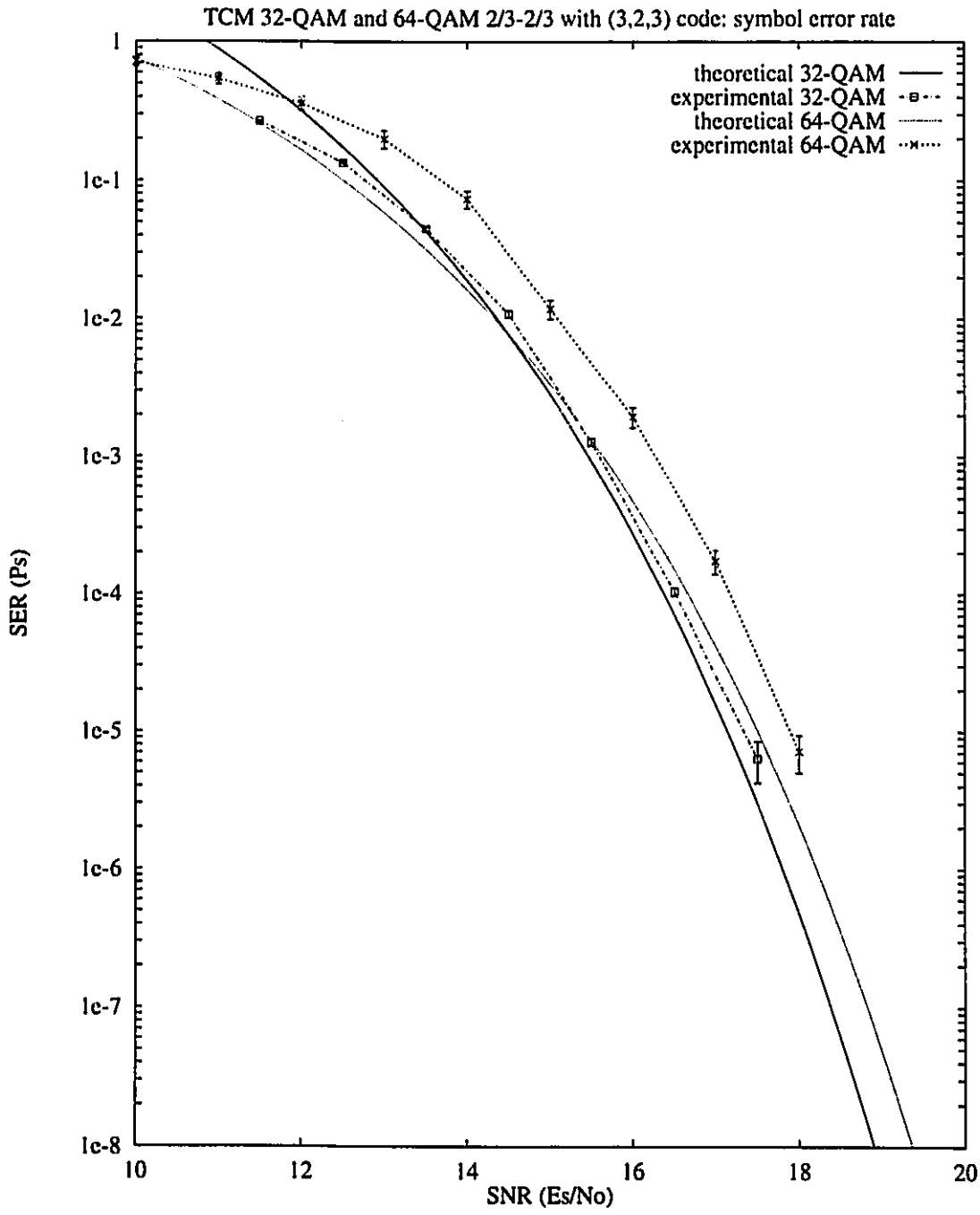

Figure 7.5: Probability of symbol error with (3,2,3) code for TCM 32-QAM and 64-QAM 2/3-2/3 transmission models.

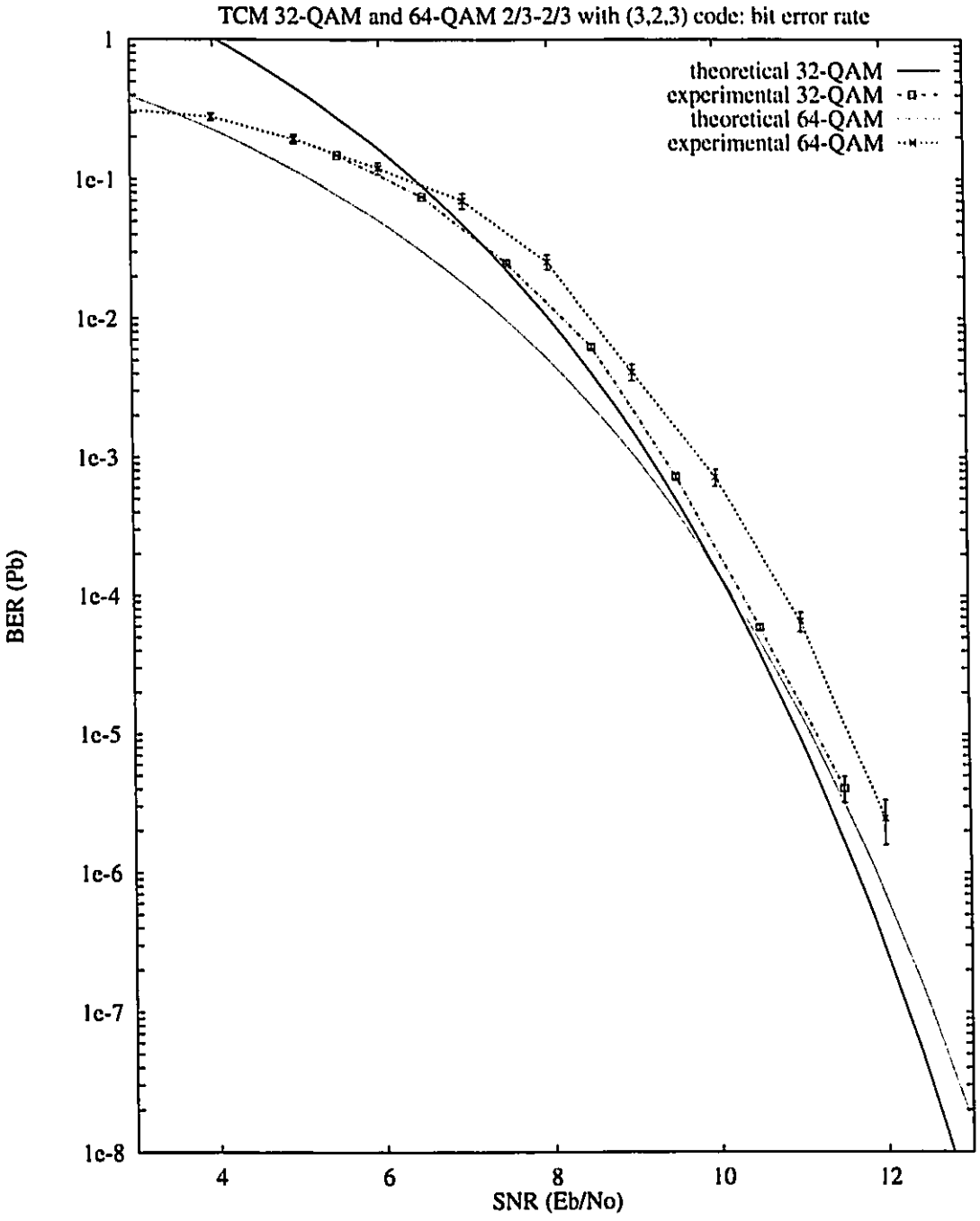

Figure 7.6: Probability of bit error with (3,2,3) code for TCM 32-QAM and 64-QAM 2/3-2/3 transmission models.

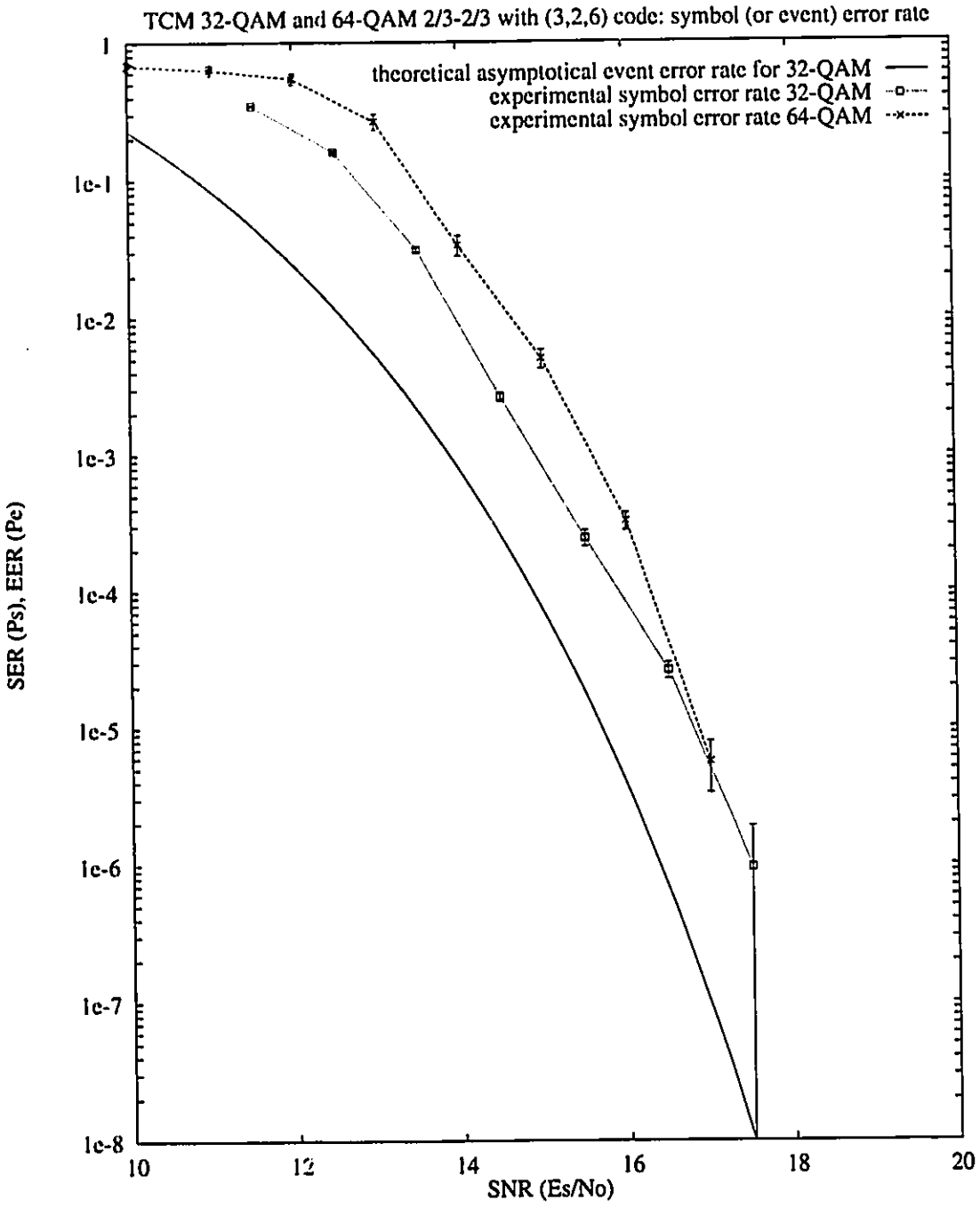

Figure 7.7: Probability of symbol error rate for both TCM 32-QAM and 64-QAM 2/3-2/3 along with theoretical asymptotic event error rate for 32-QAM, with TCM (3,2,6) code.

of probability, that is symbol rather than event error ones, and demonstrates the importance of the knowledge of exact factors rather than of the asymptotic ones. Finally, the event error probability is only known at first order, and this expands the gap with the simulation results, especially for medium and low SNR.

Figure 7.8 shows simulation curves of probability of bit error rates for both (3,2,3) and (3,2,6) codes, and both 32-QAM and 64-QAM 2/3-2/3 TCM modulation schemes. It shows the effective performance improvement when the TCM encoder complexity increases from 8 states to 64 states.

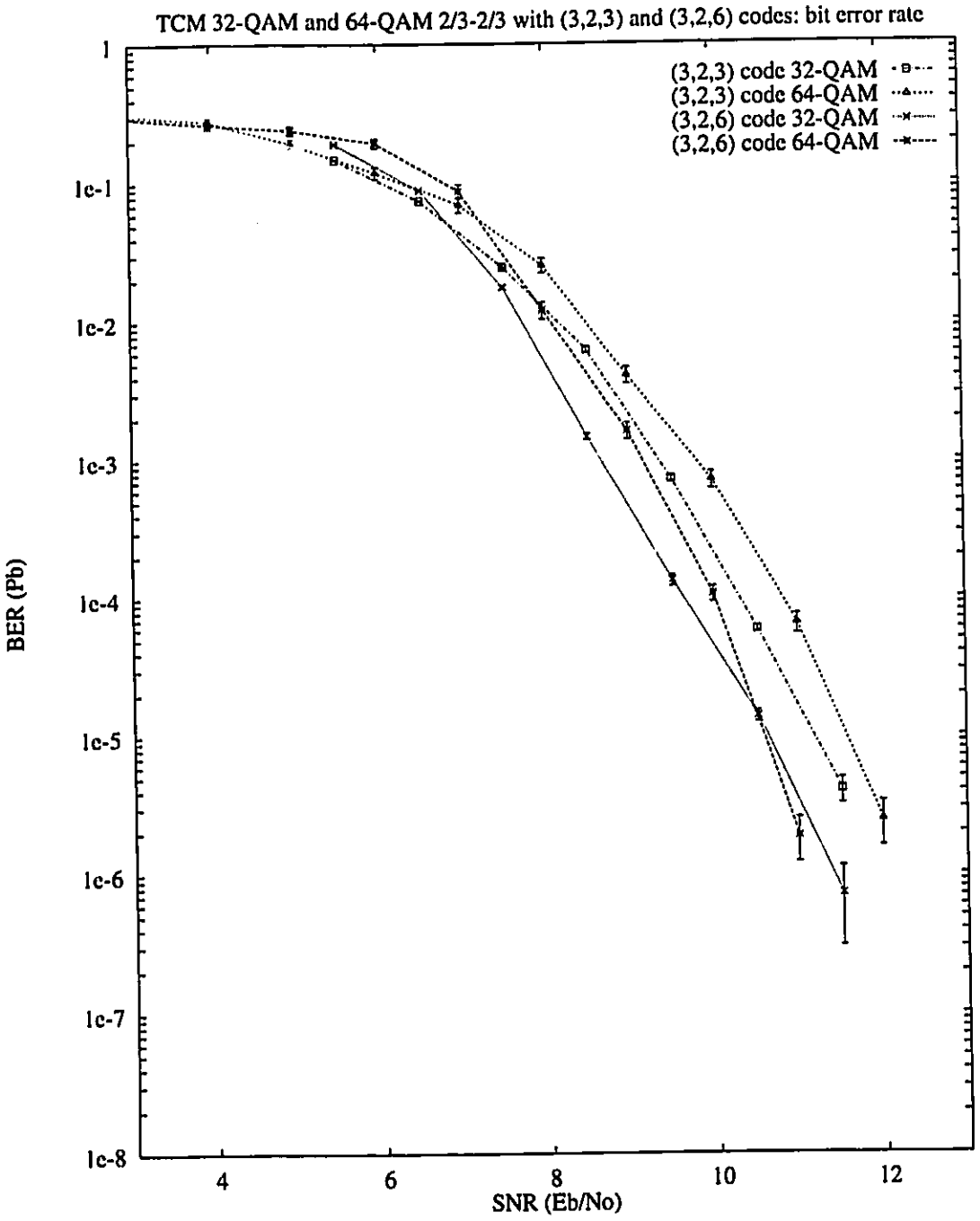

Figure 7.8: Probability of bit error for the TCM (3,2,3) and (3,2,6) codes and 32-QAM and 64-QAM 2/3-2/3 TCM schemes.

7.2.4 Simulation results with AWGN and co-channel interference

Another common impairment phenomenon consists in co-channel interference. Simulation results are provided on Figure 7.9 for the symbol error rate. Similar curves have been obtained for bit error rates. It clearly shows the drastic alteration of the received signal and the increase of the error rates caused by the co-channel signal interference. It comes out that a signal-to-noise-plus-interference ratio SNI

$$\rho_i = \frac{E_S}{E_{Si}}$$

larger than 12 dB is sufficient to cancel the improvement of a coded modulation over an uncoded one. A SNI of 15 dB leads to a loss of 10 dB on the SNR for the TCM scheme with (3,2,3) code.

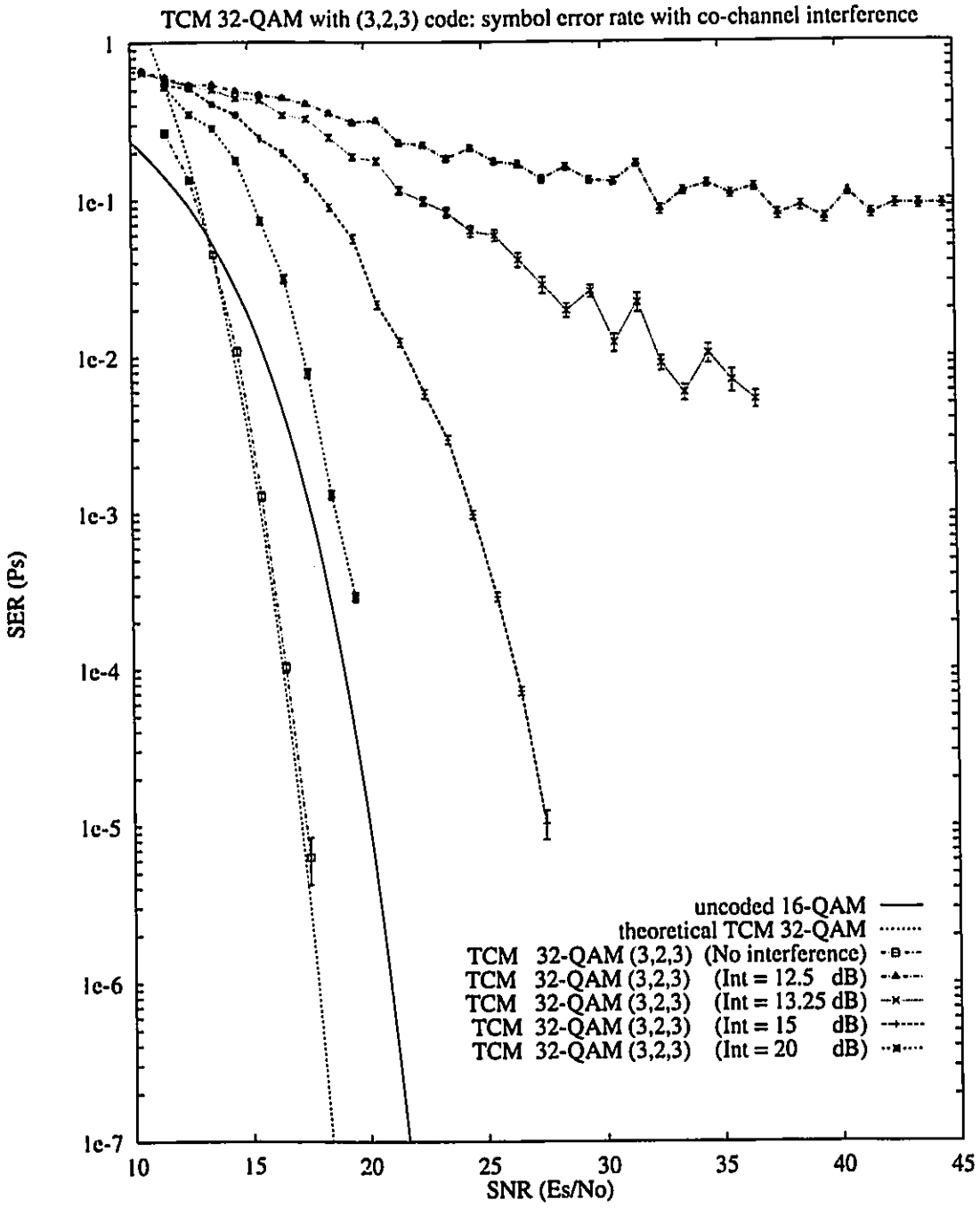

Figure 7.9: Probability of bit error for TCM (3,2,3) and (3,2,6) codes, 32-QAM and 64-QAM 2/3-2/3.

7.3 Concatenated codes: RS and TCM codes

7.3.1 Introduction

At the output of the Viterbi decoder, symbols in error often results in error-bursts and error control coding techniques are necessary. Reed-Solomon (RS) codes are powerful for decoding error-bursts.

The RS symbols are obtained by grouping m_{RS} information bits and matched to a Galois field $GF[q] = GF[2^{m_{RS}}]$. A set of k_{RS} symbols are concatenated with $2t$ redundant ones to form a codeword of n_{RS} symbols. If n_{RS} is less than $N_{RS} = q - 1 = 2^{m_{RS}} - 1$, the set of symbols is considered as filled by the required number of zero symbols. In the considered concatenated coding scheme, the codewords are then decomposed to form TCM symbols which in turn are sent through the noisy channel. At the output of the Viterbi decoder, the symbols are recombined back to their original RS codeword format. Any possible error-burst is then distributed over several consecutive RS symbols of a codeword. When t or less symbols from the set of n_{RS} of a codeword are in error, it is possible to correct them and recover the exact original k_{RS} RS symbols.

7.3.2 Description of the RS codes used

Three RS codes have been mainly dealt with in the simulations. The RS(15,9,3) code was used to test and verify the Galois field construction, the RS coding and decoding procedures.

$RS(15,9,3)$ **code:**

- Galois field $GF[16]$; symbol size: $m_{RS} = 4$. (hexadecimal symbols)
- Codeword length $N_{RS} = 15$; information words length $K_{RS} = 9$
- Error correcting capacity $t = 3$
- Effective words lengths: $n_{RS} = N_{RS}$ and $k_{RS} = K_{RS}$
- Code rate $R_c = 60\%$ (redundancy: $\frac{n-k}{n} = 40\%$)

Generator polynomial $g_{RS(15,9,3)}(x)$ of $RS(15,9,3)$ code:

$$g_{RS(15,9,3)}(x) = x^6 + \alpha^{10}\, x^5 + \alpha^{14}\, x^4 + \alpha^4\, x^3 + \alpha^6\, x^2 + \alpha^9\, x + \alpha^6$$

Table 7.2 lists the elements of GF[16], with the different formats that the program had to deal with them.

Table 7.2: Galois field $GF[q] = GF[16]$

element of $GF[16]$	binary	decimal
$\alpha^0 = 1$	0001	1
$\alpha^1 = \alpha$	0010	2
$\alpha^2 = \alpha^2$	0100	4
$\alpha^3 = \alpha^3$	1000	8
$\alpha^4 = \alpha+1$	0011	3
$\alpha^5 = \alpha^2+\alpha$	0110	6
$\alpha^6 = \alpha^3+\alpha^2$	1100	12
$\alpha^7 = \alpha^3+\alpha+1$	1011	11
$\alpha^8 = \alpha^2+1$	0101	5
$\alpha^9 = \alpha^3+\alpha$	1010	10
$\alpha^{10} = \alpha^2+\alpha+1$	0111	7
$\alpha^{11} = \alpha^3+\alpha^2+\alpha$	1110	14
$\alpha^{12} = \alpha^3+\alpha^2+\alpha+1$	1111	15
$\alpha^{13} = \alpha^3+\alpha^2+1$	1101	13
$\alpha^{14} = \alpha^3+1$	1001	9

The two following RS codes have been used to compare performances of codes of different code rate. These are the RS(208,188,10) and RS(255,239,8) codes.

$RS(208, 188, 10)$ **code:**

- Galois field $GF[q] = GF[2^{m_{RS}}] = GF[2^8]$; symbol size: $m_{RS} = 8$. (bytes)
- Codeword length $N_{RS} = 255$; information words length $K_{RS} = 235$
- Error correcting capacity $t = 10$
- Effective words lengths: $n_{RS} = 208$ and $k_{RS} = 188$
- Code rate $R_c = 90\%$ (redundancy: $\frac{n-k}{n} = 10\%$)

Generator polynomial $g_{RS(208,108,10)}(x)$ of $RS(208, 108, 10)$ code:

$$\begin{aligned} g_{RS(208,188,10)}(x) \;=\; & x^{20} + \alpha^{18}\, x^{19} + \alpha^{62}\, x^{18} + \alpha^{82}\, x^{17} + \alpha^{54}\, x^{16} + \alpha^{66}\, x^{15} + \\ & \alpha^{169}\, x^{14} + \alpha^{33}\, x^{13} + \alpha^{195}\, x^{12} + \alpha^{211}\, x^{11} + \alpha^{190}\, x^{10} + \alpha^{232}\, x^{9} + \\ & \alpha^{237}\, x^{8} + \alpha^{96}\, x^{7} + \alpha^{253}\, x^{6} + \alpha^{171}\, x^{5} + \alpha^{180}\, x^{4} + \alpha^{229}\, x^{3} + \\ & \alpha^{230}\, x^{2} + \alpha^{207}\, x + \alpha^{210} \end{aligned}$$

Code $RS(255, 239, 8)$**:**

- Galois field $GF[q] = GF[2^{m_{RS}}] = GF[2^8]$; symbol size: $m_{RS} = 8$. (bytes)
- Codeword length $N_{RS} = 255$; information words length $K_{RS} = 239$
- Error correcting capacity $t = 8$
- Effective words lengths: $n_{RS} = 255$ and $k_{RS} = 239$
- Code rate $R_c \approx 94\%$ (redundancy: $\frac{n-k}{n} \approx 6\%$)

Generator polynomial $g_{RS(255,239,8)}(x)$ of $RS(255, 239, 8)$ code:

$$\begin{aligned} g_{RS(255,239,8)}(x) \;=\; & x^{16} + \alpha^{121}\, x^{15} + \alpha^{106}\, x^{14} + \alpha^{110}\, x^{13} + \alpha^{113}\, x^{12} + \alpha^{107}\, x^{11} + \\ & \alpha^{167}\, x^{10} + \alpha^{83}\, x^{9} + \alpha^{11}\, x^{8} + \alpha^{100}\, x^{7} + \alpha^{201}\, x^{6} + \alpha^{158}\, x^{5} + \\ & \alpha^{181}\, x^{4} + \alpha^{195}\, x^{3} + \alpha^{208}\, x^{2} + \alpha^{240}\, x + \alpha^{136} \end{aligned}$$

The Table of Galois field GF[256] is given in Appendix E.

7.3.3 Simulation results with 32-QAM

Figures 7.10 and 7.11 represent the experimental results obtained with RS(208,188,10) on a 32-QAM for RS symbol error rate and bit error rate respectively. It should be pointed out that the size of a RS symbol is a byte, say $m_{RS(208,188,10)} = 8$ since they are elements of the Galois field $GF[256]$, which consists of two TCM symbols at the input and output of the convolutional encoder. Therefore, no comparison is possible between the older TCM symbols and the new RS ones.

In order to point out the improvement made by the RS code, Figure 7.12 shows the bit error rate for the TCM 32-QAM (3,2,3) code alone and concatenated with the RS(208,188,10) outer code. At the order of 10^{-5} bit error rate, a gain of $3dB$ ($2dB$) has been obtained for TCM (3,2,3) code (TCM (3,2,6) code). The performances of both TCM codes reduces to two identical curves of probability, but this is partly due to the random feature of the simulation: for TCM (3,2,6), some irregularity is observed.

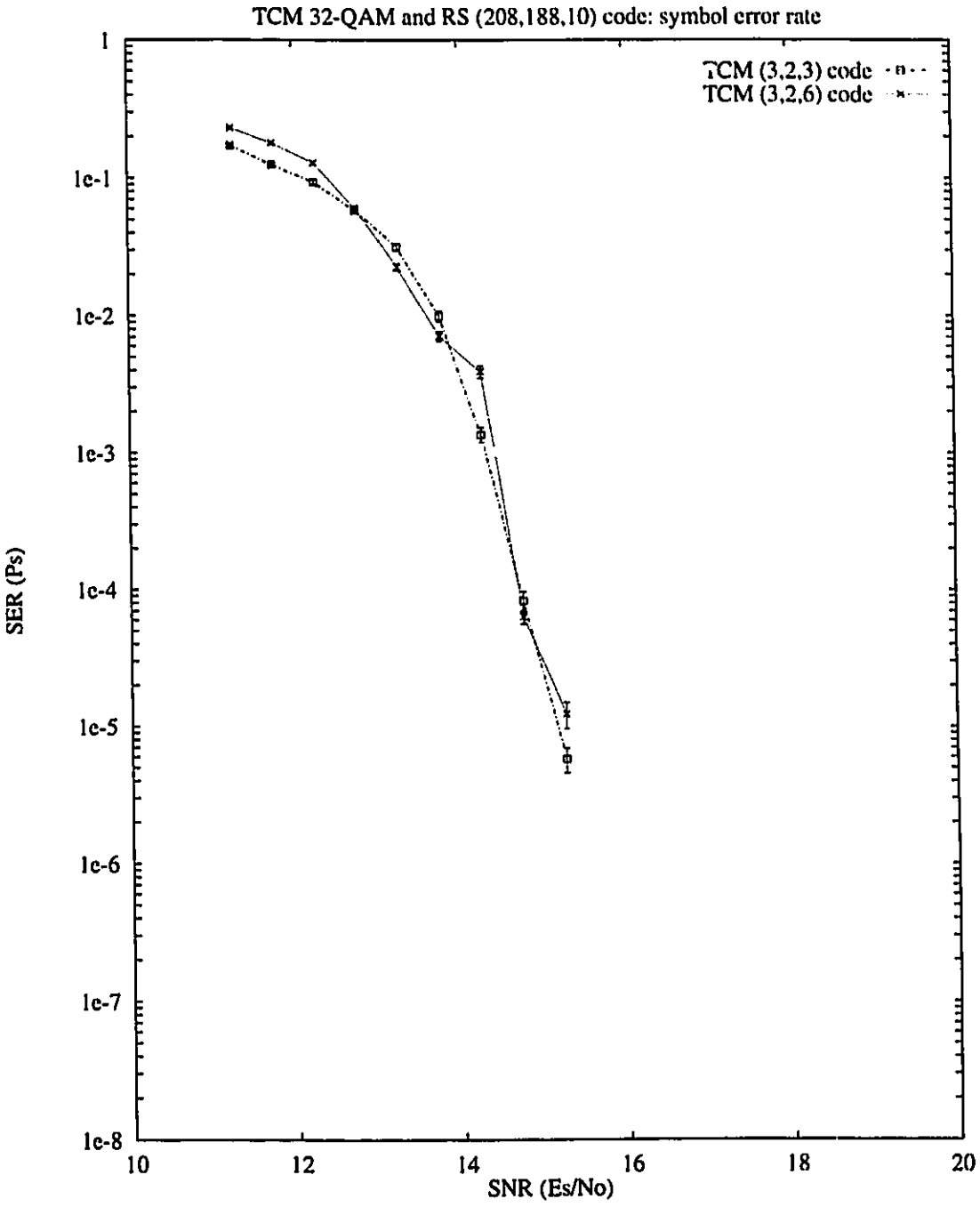

Figure 7.10: Probability of RS symbol error with TCM (3,2,3) and (3,2,6) codes through a 32-QAM transmission model.

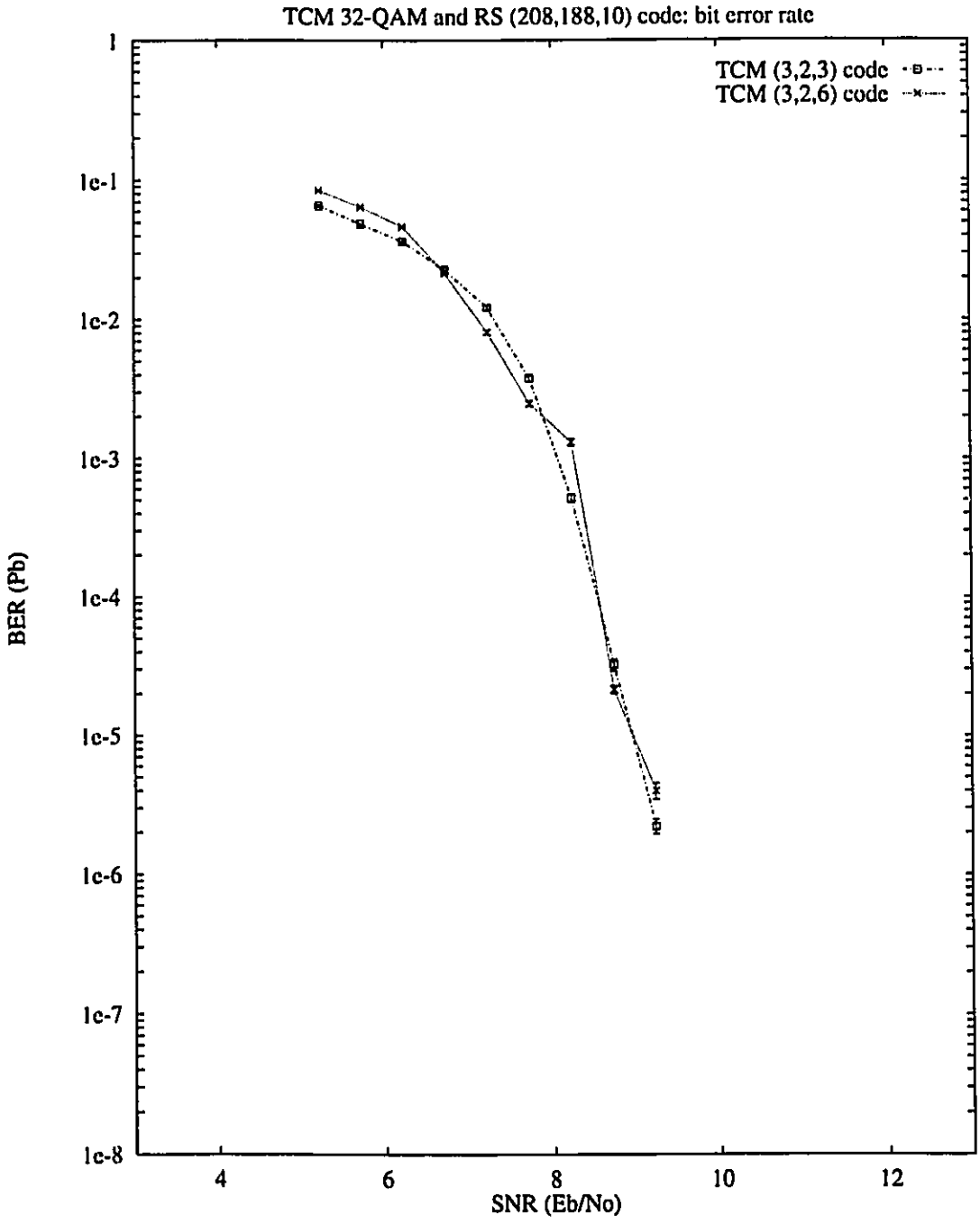

Figure 7.11: Probability of bit error with TCM (3,2,3) and (3,2,6) codes through a 32-QAM transmission model.

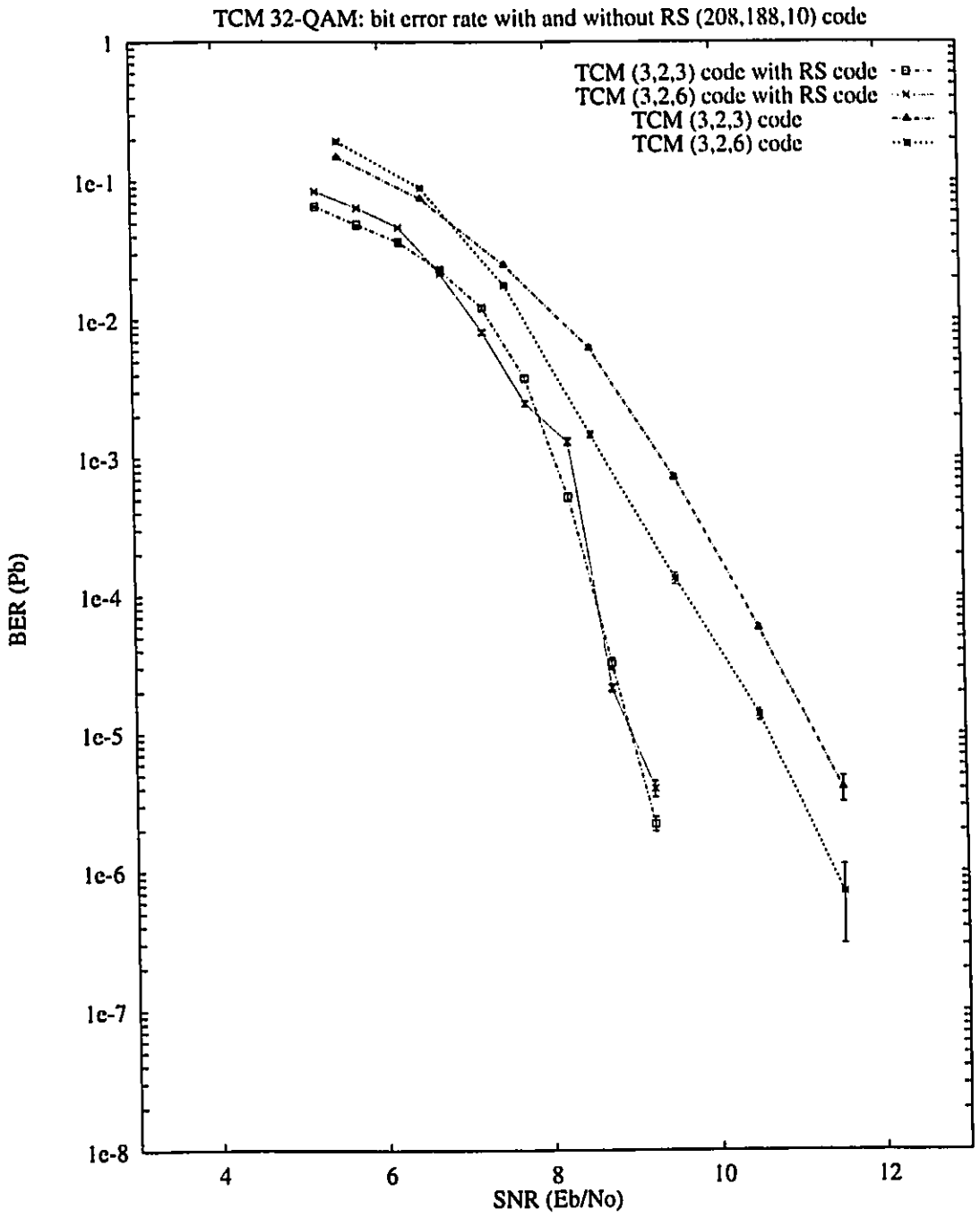

Figure 7.12: Probability of bit error with TCM (3,2,3) and (3,2,6) codes through a 32-QAM transmission model.

7.4 Interleaving improvement

Since two RS codes have been used to perform simulations, performances of the less powerful of the two are provided first. Figures 7.13 and 7.14 respectively show the symbol and bit error rate for the transmission scheme composed of the concatenation of TCM (3,2,3) or (3,2,6) codes used on a classic 32-CROSS constellation or the 2/3-2/3 rate 64-QAM model and the RS(255,239,8) code with interleaving.

Figures 7.15 and 7.16 also show the symbol and bit rate of these concatenated schemes with the RS(208,188,10) code. It should be noted that the two transmission models, say ordinary 32-TCM QAM and 2/3-2/3 64-TCM QAM present similar performances in both cases. It is also observed that the RS(208,188,10) code is far more powerful for low and medium signal-to-noise ratios. Its redundancy largely exceeds the other's. Finally, it may be observed that a gain of about 0.5 dB is obtained from the RS(255,239,8) to the RS(208,188,10) code. It corresponds to an increase in the redundancy of the RS code.

It is worth measuring the improvement provided by the interleaving. Figures 7.17 and 7.18 provide the comparison. The order of the interleaving is set to 8. It comes out that this is far enough to improve the performances of the transmission model: a gain of 1 dB is reached. These results confirm that interleaving is desirable.

7.5 Conclusion

The required bit rate of the order of 10^{-9} has not been reached through computer simulations. However, at the order of 10^{-5}, the curves of error probability are almost vertical. There is only less than one 1 dB missing. This has been done with both TCM schemes, that are, conventional 32-QAM and rate 2/3-2/3 64-QAM, concatenated with interleaved Reed-Solomon codes.

This establishes that the transmission scheme composed of TCM code combined

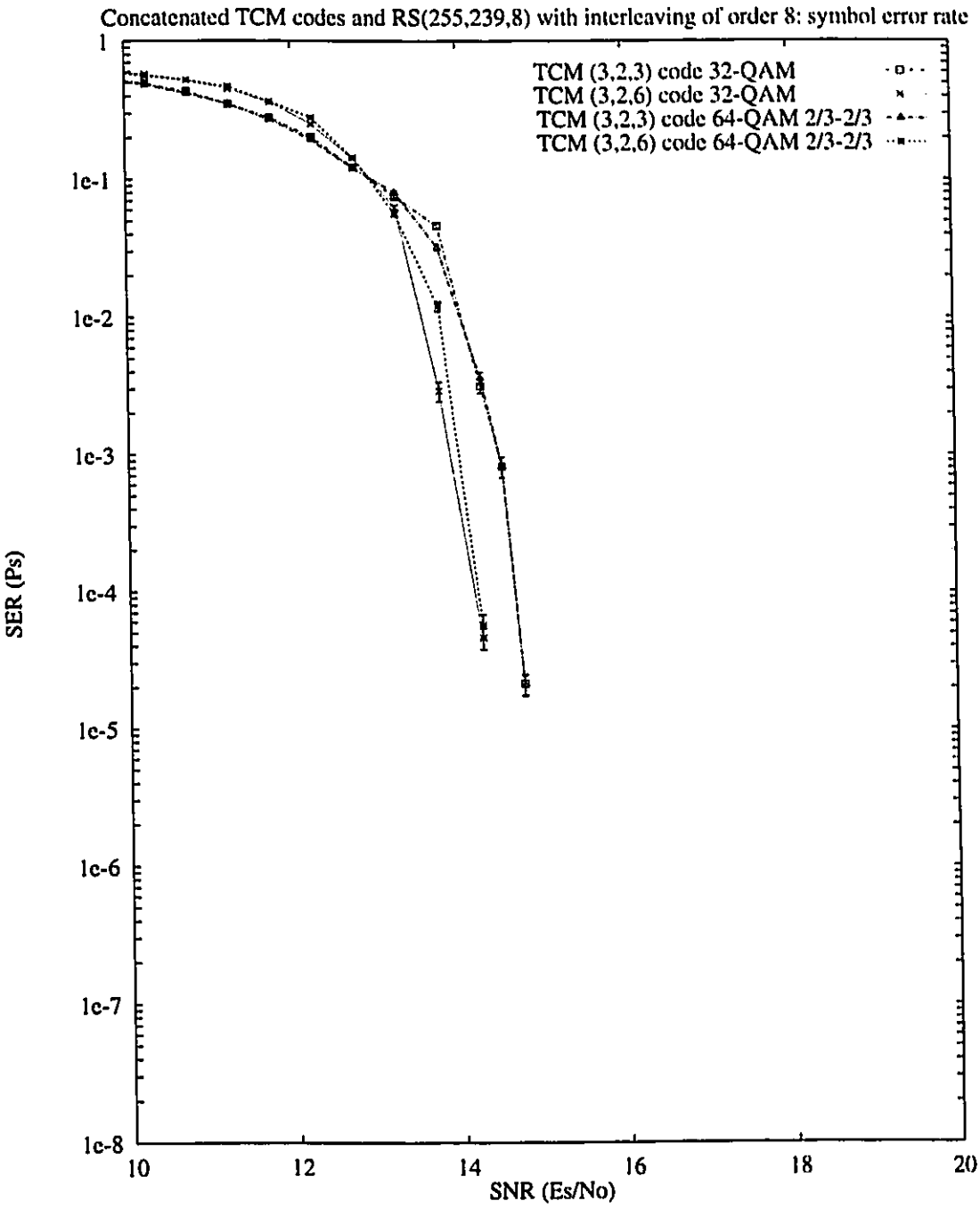

Figure 7.13: Probability of RS symbol error with TCM (3,2,3) and (3,2,6) codes through a 32-QAM and RS(255,239,8) transmission model.

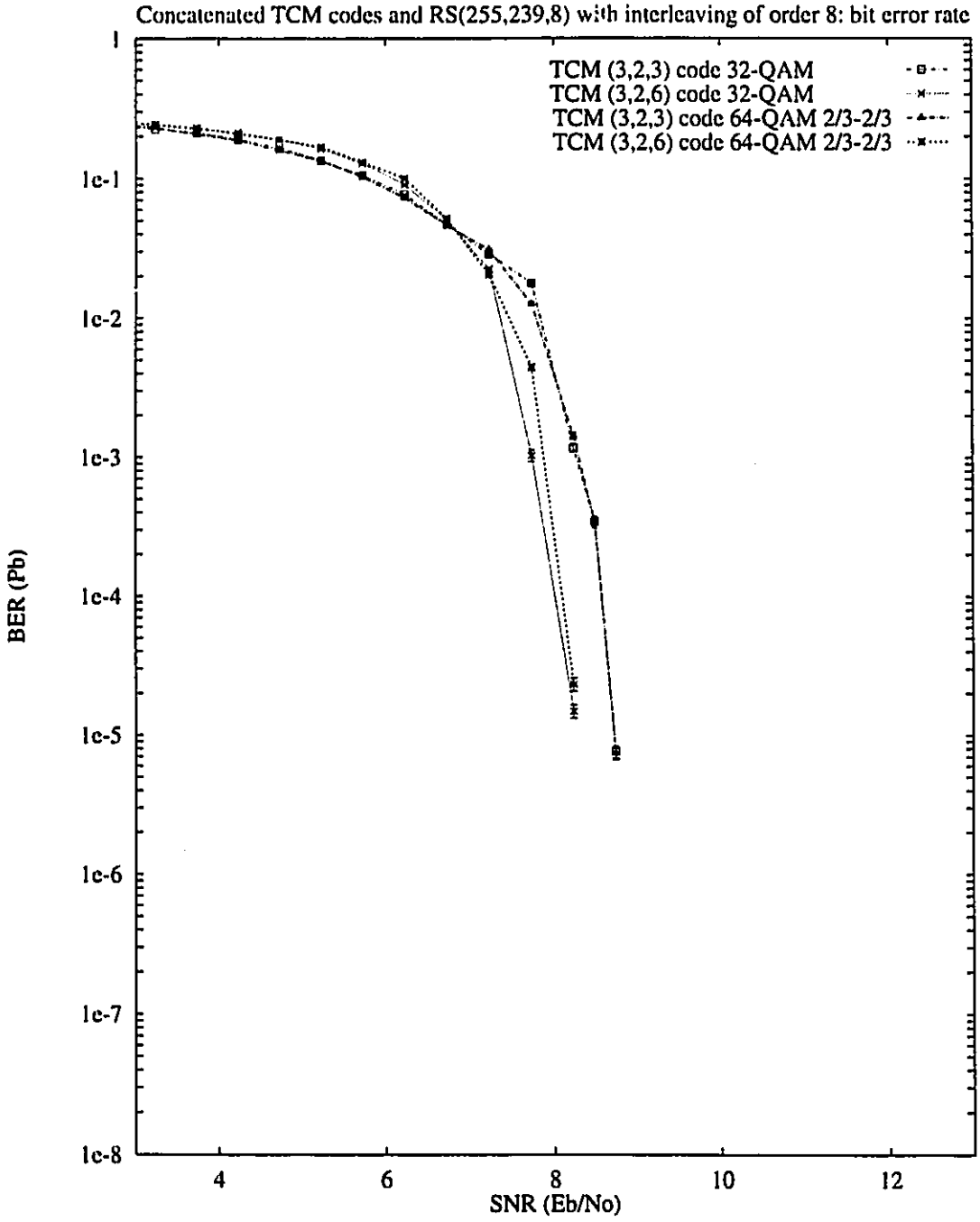

Figure 7.14: Probability of bit error with TCM (3,2,3) and (3,2,6) codes through a 32-QAM and RS(255,239,8) transmission model.

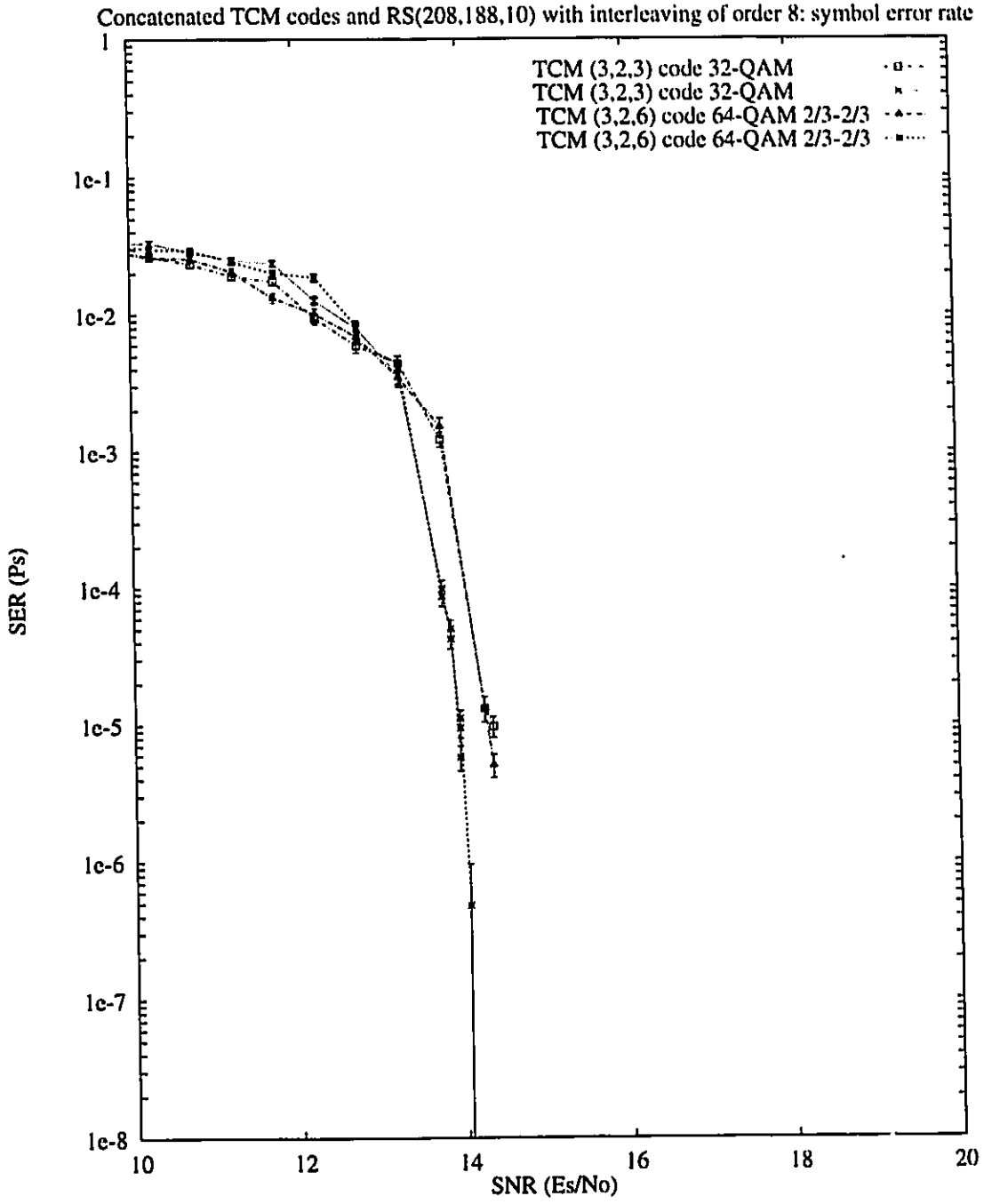

Figure 7.15: Probability of RS symbol error with TCM (3,2,3) and (3,2,6) codes through a 32-QAM and RS(208,188,10) transmission model.

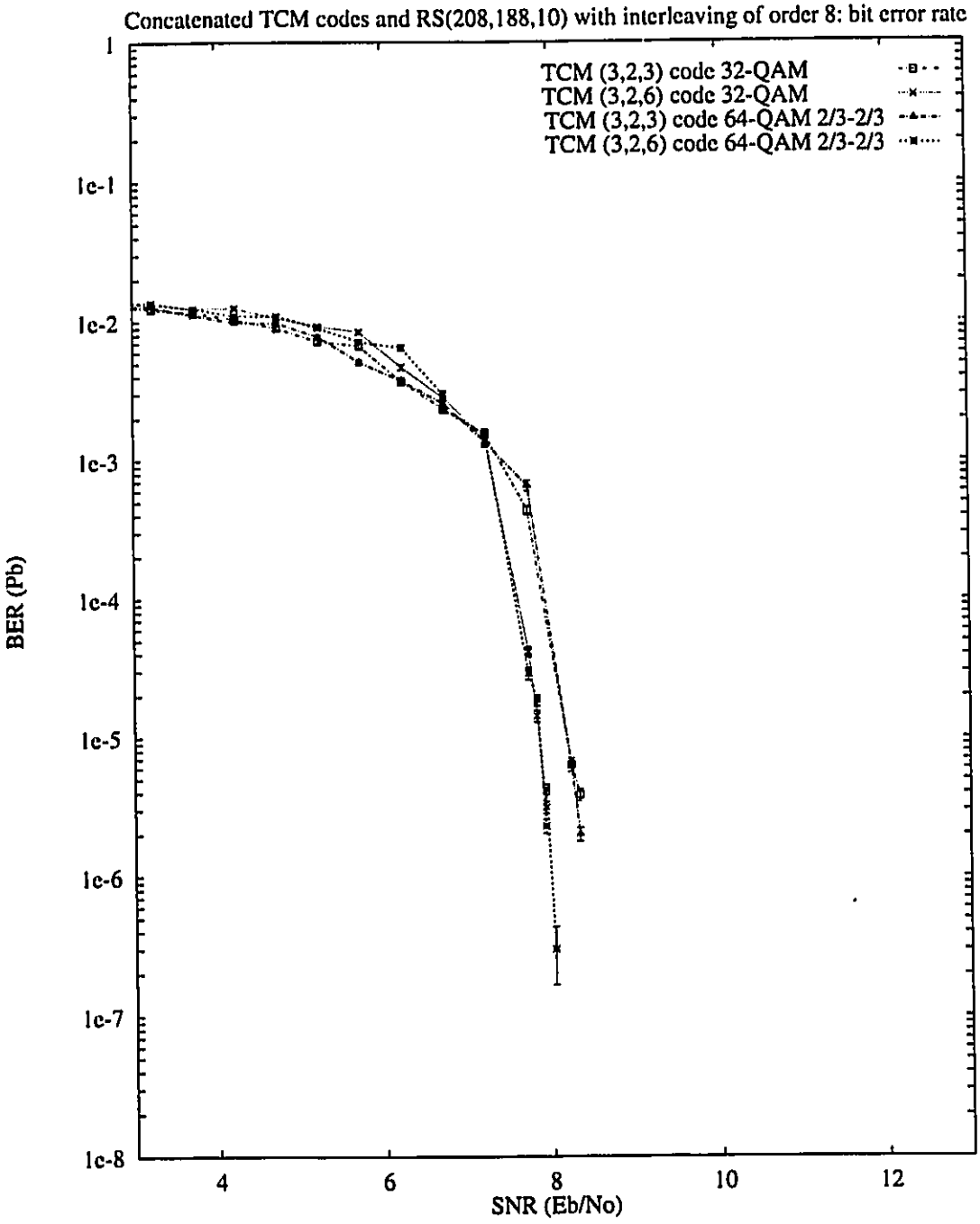

Figure 7.16: Probability of bit error with TCM (3,2,3) and (3,2,6) codes through a 32-QAM and RS(208,188,10) transmission model.

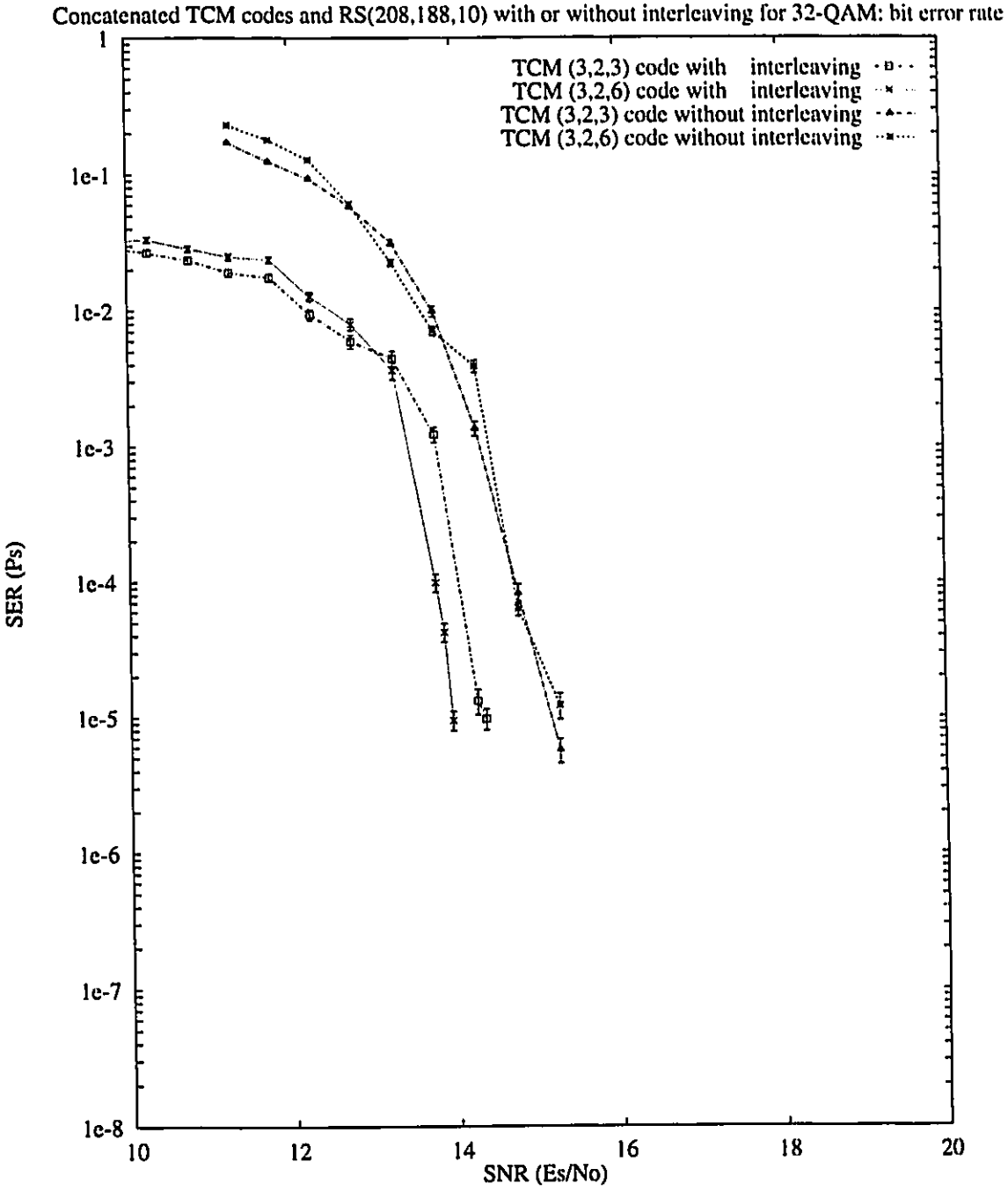

Figure 7.17: Probability of RS symbol error with TCM (3,2,3) code through a 32-QAM, RS(208,188,10) and interleaving of order 8.

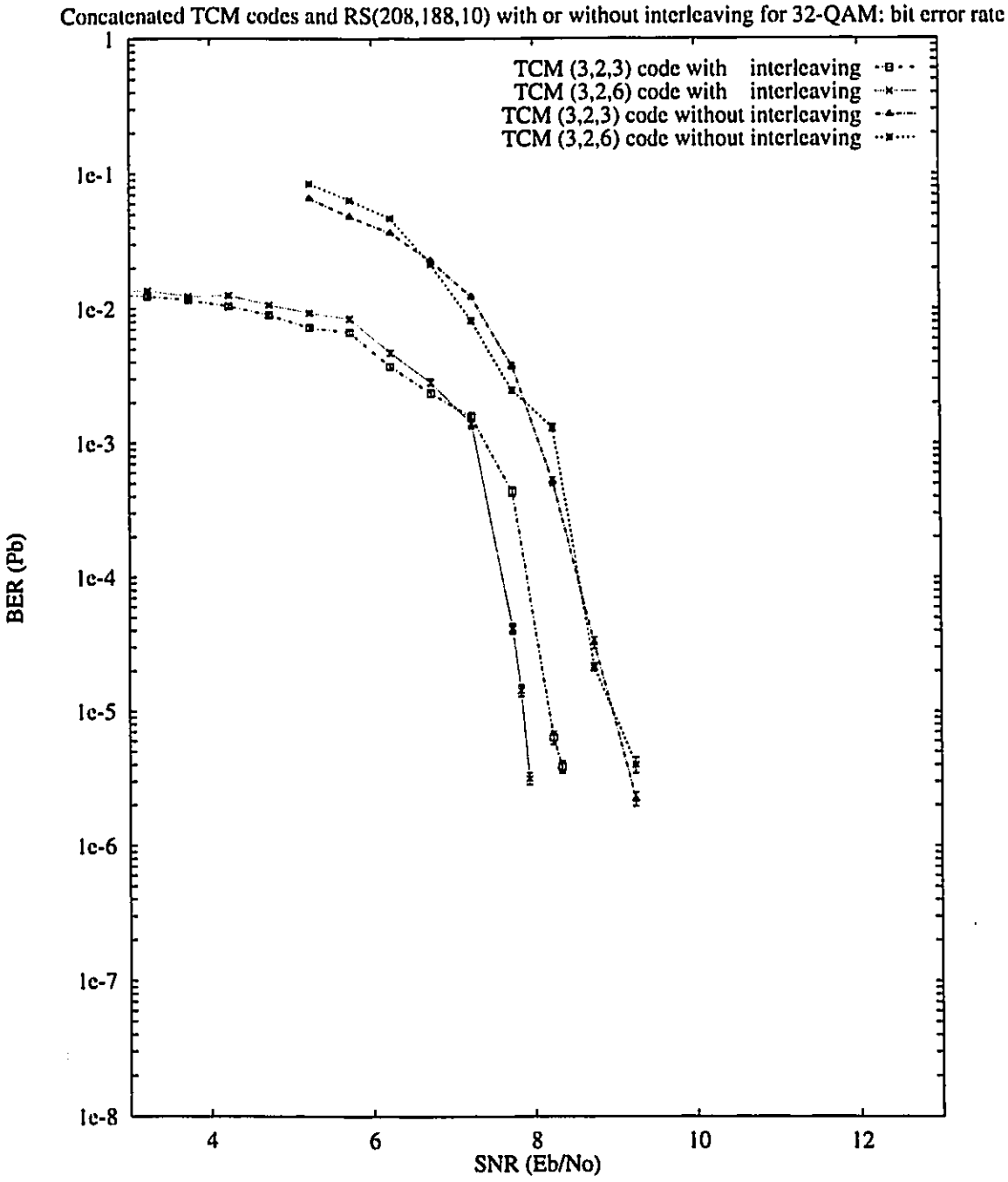

Figure 7.18: Probability of bit error with TCM (3,2,3) code through a 32-QAM, RS(208,188,10) and interleaving of order 8.

with Reed-Solomon codes and interleaving is a potential candidate for the model that will be adopted to broadcast HDTV in North-America within the next decades.

Chapter 8

Conclusions

8.1 Thesis summary

The study presented in this thesis covers several different aspects of Trellis Coded Modulation for Quadrature Amplitude Modulation. In chapter 2, a systematic search for optimum mappings has been explored for up to a 64-QASK constellation. These mappings were to be used with the Ungerboeck recursive convolutional codes. The method is expanded in detail, and it requires some optimization inside each sub-constellation, which may be very different depending on the number of parallel transitions of the trellis. It has been shown that under this constraint, puncturing a high-leveled constellation does not generally result in an optimum smaller one.

Chapter 3 explored the performance evaluation of Trellis Coded Modulation. It has been shown that under some geometric properties of the mappings, the *uniformity condition* could be satisfied. This allows for far less elaborative calculations of the transfer functions of the TCM codes. The mappings derived from chapter 2 were found to meet this condition.

With some computer support, the transfer functions of TCM codes have been calculated and used to express the theoretical probabilities of event, symbol and bit error. The results of the programs gave parameters of the TCM codes to second order and were reported in chapter 4. A synthetic expression for error probability

has also been stated. Analysis of different types of error upperbounds resulted in the observation that the second order upperbound was more reliable, especially for lower signal-to-noise ratios. Chapter 6 gives tables of parameters and presents curves of error probability using the calculations of the transfer functions. These curves emphasize how useful the second order expression for error probability is, especially for low and medium signal-to-noise ratios. They are useful since simulation results are difficult to obtain for high signal-to-noise ratios, and allow for comparison with theoretical results.

Some experimental results had also been obtained through simulations over AWGN, as reported in chapter 7. The simulations have been run using an equivalent base-band system model, under the assumption of a perfect square pulse shape to ensure the Nyquist criteria and prevent any intersymbol interference. They are close to the theoretical results and powerful enough to consider an HDTV broadcasting implementation using such a transmission scheme. Error control coding was employed using RS codes with interleaving to improve the performance of the system.

In addition, another scheme for TCM QAM, namely rate 2/3-2/3 64 QAM, has been studied with separate coding processes for the in-phase and quadrature components of the signal in chapter 5. The performances of this new scheme is comparable to the previously discussed 32-QAM TCM scheme. This new scheme has the advantage of reducing the complexity of the decoder applications, which may be very interesting for high bit rate applications, such as HDTV.

8.2 Suggestions for further research

Although optimization within the sub-constellation was suggested on a 32-CROSS constellation, it has still not been done. It is desirable to evaluate by simulation the expected improvement of this part of the optimization of mappings. As far as theoretical evaluation is concerned, it would be possible to use the same kind of program to provide expression of the transfer functions of the code, under the condition that

the code remains *uniform*.

However, as reported in chapters 4 and 5, the calculation of second order terms require much more computer memory space, and this means that a powerful algorithm with efficient memory management should be first designed.

Finally, since HDTV transmission has been considered in this thesis, it would be worthwhile to provide comparative performances of different models. For instance, in Europe, an OFDM is being dealt with for HDTV signals, and Japan still suggest other transmission schemes.

Appendix A

Matrix expressions for the probability of error events

In section 3.1, the probability of event errors has been expressed in terms of a weight function $\mathbf{W}$ of an error event $\mathbf{E}^L$ as a sum over a symbol sequences $\mathbf{Y}^L$. From Equation (3.6), half the sum over the possible event errors $\mathbf{E}^L$ is an upperbound for the probability of event error and Equation (3.5) is the union bound for the occurrence of all possible symbol sequences corrupted by such an event error $\mathbf{E}^L$. This event error weight function may also be expressed in terms of matrices of dimension $N_\nu \times N_\nu$ where N_ν denotes the number of states 2^ν of the code, as declared on Equation (3.7). The proof of this "obscure" statement follows.

A symbol sequence $\mathbf{Y}^L$ is described by:

$$\mathbf{Y}^L = [Y_1, Y_2, \ldots, Y_L]$$

or a sequence of states:

$$\mathbf{Y}^L \equiv (s_1, s_2, \ldots, s_{L+1})$$

such that the symbol Y_i is the output of one of the 2^{m-k} transitions $t_{s_i \to s_{i+1}}$, where k denotes the number of bits convolutionally encoded. Thus, in Equation (3.5), the summation over $\mathbf{Y}^L$ may be expressed as:

$$\begin{aligned}
\sum_{\mathbf{Y}^L} &= \sum_{(Y_1,Y_2,\ldots,Y_L)} \\
&= \sum_{(s_1,s_2,\ldots,s_{L+1})} \left\{ \sum_{Y_1 \in \left\{Y_{t_{s_1-s_2}}\right\}} + \sum_{Y_2 \in \left\{Y_{t_{s_2-s_3}}\right\}} + \cdots + \sum_{Y_L \in \left\{Y_{t_{s_L-s_{L+1}}}\right\}} \right\} \\
&= \sum_{(s_1,s_2,\ldots,s_{L+1})} \left\{ \sum_{(Y_1,Y_2,\ldots,Y_L) \in \left\{Y_{t_{s_1-s_2}}\right\} \bowtie \left\{Y_{t_{s_2-s_3}}\right\} \bowtie \cdots \bowtie \left\{Y_{t_{s_L-s_{L+1}}}\right\}} \right\} \qquad \text{(A.1)}
\end{aligned}$$

where the symbol $\bowtie$ represents the cartesian product. Combined with Equation (A.2) which expresses nothing but the distributivity of the multiplication over the addition:

$$\prod_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}(i)} a_{i,j} = \sum_{\left(j_1,j_2,\cdots,j_{\dim(\mathcal{I})}\right) \in \mathcal{J}(1) \bowtie \mathcal{J}(2) \bowtie \cdots \bowtie \mathcal{J}(\dim(\mathcal{I}))} \prod_{i \in \mathcal{I}} a_{i,j_i} \qquad \text{(A.2)}$$

and applied onto Equation (3.5), Equation (A.1) becomes:

$$\mathbf{W}(\mathbf{E}^L) = \sum_{(s_1,s_2,\ldots,s_{L+1})} \prod_{l=1}^{L} \sum_{Y \in \left\{Y_{t_{s_l-s_{l+1}}}\right\}} \left(\mathcal{P}\{Y\} Z^{\|J(Y)-J(Y \oplus E_l)\|^2} \right) \qquad \text{(A.3)}$$

Let $\mathbf{G}(E)$ be an *error weight matrix* defined according to Equation (3.8). Then the term

$$\sum_{Y \in \left\{Y_{t_{s_l-s_{l+1}}}\right\}} \left(\frac{1}{2^m} Z^{\|J(Y)-J(Y \oplus E_l)\|^2} \right) = \frac{1}{2^m} \sum_{Y \in \left\{Y_{t_{s_l-s_{l+1}}}\right\}} Z^{\|J(Y)-J(Y \oplus E_l)\|^2} \qquad \text{(A.4)}$$

is the entry (s_l, s_{l+1}) of $\mathbf{G}(E_l)$. Hence, Equation (A.3) is exactly:

$$\begin{aligned}
\mathbf{W}(\mathbf{E}^L) &= \frac{1}{N_\nu} \sum_{(s_1,s_2,\ldots,s_{L+1})} \prod_{l=1}^{L} [\mathbf{G}(E_l)]_{s_l,s_{l+1}} \Big|_{D=Z} \\
&= \frac{1}{N_\nu} \sum_{(s_1,s_{L+1})} \left(\sum_{(s_2,\ldots,s_L)} [\mathbf{G}(E_1)]_{s_1,s_2} \times [\mathbf{G}(E_2)]_{s_2,s_3} \times \cdots \times [\mathbf{G}(E_L)]_{s_L,s_{L+1}} \Big|_{D=Z} \right) \qquad \text{(A.5)}
\end{aligned}$$

The last sum has the pattern of the expression of the entry (s_1, s_{L+1}) of the product of the L matrices $\mathbf{G}(E_l)$, and Equation (A.5) becomes:

$$\mathbf{W}(\mathbf{E}^L) = \frac{1}{N_\nu} \sum_{(s_1, s_{L+1})} \left[\prod_{l=1}^{L} \mathbf{G}(E_l) \right]_{s_1, s_{L+1}} \Bigg|_{D=Z} \tag{A.6}$$

The sum over the states (s_1, s_{L+1}) represents the sum of the entries of the matrix $\prod_{l=1}^{L} \mathbf{G}(E_l)$. Therefore, if $\mathbf{1}$ denotes a vector of N_ν elements equal to 1:

$$\mathbf{W}(\mathbf{E}^L) = \frac{1}{N_\nu} {}^t\mathbf{1} \left(\prod_{l=1}^{L} \mathbf{G}(E_l) \right) \mathbf{1} \Bigg|_{D=Z} \tag{A.7}$$

Appendix B

General case for the Uniformity condition

B.1 Definition of $\mathcal{Y}_0$

We go back to section 3.2 and reconsider the introduction of symbols subset $\mathcal{Y}_0$. Because of the structure of the convolutional code, the coded out-bit $y_t^{(0)}$ was simply equal to r_0, the very first register of the code. All other coded out-bits were equal to the corresponding in-bits. Thus, it was a particular and simple case to make a distinction between the even and odd states, as simple as letting $\mathcal{Y}_0$ be the symbols subset of even outputs! Using our notation, we here recall the steps of the very general study of linear codes, provided by Zehavi and Wolf in [14]. Our $\mathcal{Y}_0$ would be defined according to *Lemma 1*, which states that it is the set of outputs of a coder from the all-zero state: $\mathcal{Y}_0$ is a commutative group, and the output of any other state belongs either to $\mathcal{Y}_0$ or $\overline{\mathcal{Y}_0}$. As a corollary, the existence of a partition of the set of all possible states, into two subsets $\mathcal{S}_0$ and $\overline{\mathcal{S}_0}$, such that the outputs of any state from $\mathcal{S}_0$ (resp. $\overline{\mathcal{S}_0}$) are in $\mathcal{Y}_0$ (resp. $\overline{\mathcal{Y}_0}$) is derived. Then *Lemma 2* states that an error E is either in $\mathcal{Y}_0$ or in $\overline{\mathcal{Y}_0}$ whether it came from a state from $\mathcal{S}_0$ or $\overline{\mathcal{S}_0}$. Finally, [14] defined a class of codes that are based upon a binary linear convolutional code with a nonlinear

mapping from the encoder, and satisfy:

$$\forall E \in \mathcal{Y}\ ,\ \forall s \in \mathcal{S}_0\ ,\ \forall \overline{s} \in \overline{\mathcal{S}_0}\ ,\ \sum_{X \in \mathcal{X}} D^{\|f(Y_{s,X}) - f(Y_{s,X} \oplus E)\|^2} = \sum_{X \in \mathcal{X}} D^{\|f(Y_{\overline{s},X}) - f(Y_{\overline{s},X} \oplus E)\|^2} \tag{B.1}$$

where $Y_{s,X}$ denotes the output of the transition from state s with input X, and $\mathcal{X}$ denotes the set of all possible inputs. For any code from this class, *Theorem 1* [14] states the possible computation of the Euclidean distances between correct and incorrect paths, through the trellis code. This is what has been done in chapter 4. The above second condition, (B.1), is nothing but what [1] defined as the *uniformity condition*, as developed in chapter 3.

For the Ungerboeck codes used in this thesis, as seen in chapter 3, $\mathcal{S}_0$ is the subset of even states and $\mathcal{Y}_0$ the subset of even outputs. They are related through the equality: $y^{(0)} = r_0$, where r_0 is the first register of the recursive Ungerboeck codes.

B.2 Satisfying the *uniformity condition*

Consider Equation (B.1). According to what has just been recalled above, $\mathcal{Y}_0$ is defined as:

$$\begin{aligned} \mathcal{Y}_0 &= \{Y_{s,X} \in \mathcal{Y}\ ,\ X \in \mathcal{X}\ (s \in \mathcal{S}_0)\} \\ &= \{Y_{s,X} \in \mathcal{Y}\ ,\ X \in \mathcal{X}\ (\forall s \in \mathcal{S}_0)\} \end{aligned} \tag{B.2}$$

This implies that:

$$\forall E \in \mathcal{Y}\ ,\ \forall s_1\ ,\ s_2 \in \mathcal{S}_0, \sum_{X \in \mathcal{X}} D^{\|f(Y_{s_1,X}) - f(Y_{s_1,X} \oplus E)\|^2} = \sum_{X \in \mathcal{X}} D^{\|f(Y_{s_2,X}) - f(Y_{s_2,X} \oplus E)\|^2}$$

Similarly, its complementary $\overline{\mathcal{Y}_0}$ is equal to:

$$\begin{aligned} \overline{\mathcal{Y}_0} &= \{Y_{\overline{s},X} \in \mathcal{Y}\ ,\ X \in \mathcal{X}\ (\overline{s} \in \overline{\mathcal{S}_0})\} \\ &= \{Y_{\overline{s},X} \in \mathcal{Y}\ ,\ X \in \mathcal{X}\ (\forall \overline{s} \in \overline{\mathcal{S}_0})\} \end{aligned} \tag{B.3}$$

and consequently:

$$\forall E \in \mathcal{Y}\ ,\ \forall \overline{s_1}, \overline{s_2} \in \overline{\mathcal{S}_0}, \sum_{X \in \mathcal{X}} D^{\|f(Y_{\overline{s_1},X}) - f(Y_{\overline{s_1},X} \oplus E\|^2} = \sum_{X \in \mathcal{X}} D^{\|f(Y_{\overline{s_2},X}) - f(Y_{\overline{s_2},X} \oplus E)\|^2}$$

Therefore, the condition given in Equation (B.1) reduces to:

$$\exists s \in \mathcal{S}_0 \ , \ \exists \overline{s} \in \overline{\mathcal{S}_0} \ , \ \forall E \in \mathcal{Y} \ , \ \sum_{X \in \mathcal{X}} D^{\|f(Y_s,X)-f(Y_s,X\oplus E)\|^2} = \sum_{X \in \mathcal{X}} D^{\|f(Y_{\overline{s}},X)-f(Y_{\overline{s}},X\oplus E)\|^2} \tag{B.4}$$

Let s and $\overline{s}$ be respectively two states from $\mathcal{S}_0$ and $\overline{\mathcal{S}_0}$. Using Equation (B.2), the summations from left and right hand sides of Equation (B.4) may be considered as:

$$\forall E \in \mathcal{Y} \ , \ \sum_{Y_i \in \mathcal{Y}_0} D^{\|f(Y_i)-f(Y_i\oplus E)\|^2} = \sum_{Y_c \in \overline{\mathcal{Y}_0}} D^{\|f(Y_c)-f(Y_c\oplus E)\|^2} \tag{B.5}$$

Both Equation (B.5) and *Lemma 2* point out the existence of two cases:

first case: $E \in \overline{\mathcal{Y}_0}$

Then, for all Y from $\mathcal{Y}_0$, $Y \oplus E$ belongs to $\overline{\mathcal{Y}_0}$. Let Y_{i0} be in $\mathcal{Y}_0$. Then there exists a unique symbol Y_{c0} in $\overline{\mathcal{Y}_0}$ such that: $Y_{i0} \oplus E = Y_{c0}$, and more: $Y_{c0} \oplus E = Y_{i0}$. The left hand side term corresponding to Y_{i0} is:

$$D^{\|f(Y_{i0})-f(Y_{i0}\oplus E)\|^2}$$

and the right hand side one corresponding to Y_{c0} is:

$$D^{\|f(Y_{c0})-f(Y_{c0}\oplus E)\|^2} = D^{\|f(Y_{i0}\oplus E)-f(Y_{i0})\|^2}$$

which points out a one-to-one correspondence between the terms of each side of the Equality (B.5). With no restriction, Equation (B.5) stands.

second case: $E \in \mathcal{Y}_0$

Then, for all Y from $\mathcal{Y}_0$, $Y \oplus E$ belongs to $\mathcal{Y}_0$ and for all Y in $\overline{\mathcal{Y}_0}$, $Y \oplus E$ belongs to $\overline{\mathcal{Y}_0}$. A simple sufficient condition may easily be the existence of an isometry that maps:

$$\{Y_i \in \mathcal{Y}_0\} \overset{\sigma}{\longmapsto} \{Y_c \in \overline{\mathcal{Y}_0}\}$$

Then we have:

$$\forall Y_i \in \mathcal{Y}_0 \ , \ D^{\|f(Y_{i0}\oplus E)-f(Y_{i0})\|^2} = D^{\|\sigma(f(Y_{i0}\oplus E))-\sigma(f(Y_{i0}))\|^2}$$

and

$$\sum_{Y_i \in \mathcal{Y}_0} D^{\|f(Y_i)-f(Y_i\oplus E\|^2} = \sum_{Y_i \in \mathcal{Y}_0} D^{\|\sigma(f(Y_i))-\sigma(f(Y_i\oplus E))\|^2} = \sum_{Y_c \in \overline{\mathcal{Y}_0}} D^{\|f(Y_c)-f(Y_c\oplus E)\|^2}$$

where the last equality simply consists of an index permutation.

In section 3.2, isometry was mathematically expressed as:

$$\sigma(f(Y)) = f(Y \oplus \hat{Y}) = f(\hat{Y} \oplus 1)$$

and in section 3.3 it practically consisted of a rotation by $-\frac{\pi}{2}$ on the constellation.

Thus, a sufficient condition we may use in order to satisfy the *uniformity condition* is the existence of an isometry on the constellation which maps a subset of symbols to its complementary subset, this subset being strictly related to the code structure.

Appendix C

Average energy per symbol for one and two-dimensional mappings

C.1 Introduction

When TCM and uncoded modulation performances are compared to each other, the average energy per symbol is an important factor which has to be at least canceled by the power of the free distance of the code. Since coded modulation increases the size of the constellation, it is necessary to evaluate the exact values of the ratio of coded to uncoded energy per symbol. In chapter 6, these values were reported with no cited references, nor any justification. This Appendix provides the calculation. It should be noted that the 8-CROSS constellation doesn't match the general formula derived later for higher CROSS constellations. This is the reason why in Tables 6.2, 6.4, and 6.6, the 8-CROSS constellation is shown separately from the higher CROSS constellations.

Based on the observation that the polynomial:

$$P(x) = \frac{4}{3}x^3 - \frac{1}{3}x$$

is of the form:

$$P(x) - P(x-1) = (2x-1)^2$$

we derive:

$$\sum_{i=1}^{n} (2i-1)^2 = P(n) - P(0) = \frac{(2n-1)(2n)(2n+1)}{6} \qquad \text{(C.1)}$$

C.2 Average energy per symbol for one dimensional mapping

For a 2^m-ASK constellation, the average energy per symbol is expressed in terms of $d_0 = \frac{\Delta_0}{2}$ as:

$$E_S = \frac{1}{2^m}(2)\sum_{i=1}^{2^{m-1}} [(2i-1)d_0]^2$$

Using Equation (C.1), it reduces to:

$$E_S = \frac{2^{2m}-1}{3} d_0^2$$

Or when normalized to Δ_0:

$$\frac{E_{S_{ASK}}^{(2^m)}}{\Delta_0^2} = \frac{2^{2m}-1}{12} \qquad \text{(C.2)}$$

C.3 Average energy per symbol for two dimensional mappings

The non square shape of CROSS constellations make the calculation of the average energy a bit more complicate. We first provide it for square constellation, that is for 2^{2m}-QASK.

C.3.1 Average energy per symbol for QASK constellation

The total square is composed of $2^m \times 2^m$ points and shared by the x and y-axis into 4 smaller square of $2^{m-1} \times 2^{m-1}$ squares. On each of these 4 squares, all points

located on the i^{th} vertical (resp. horizontal) line have equal abscissa (resp. ordinate): $(2i-1)d_0$. Therefore:

$$E_S = \frac{1}{2^{2m}}(4)(2)\sum_{i=1}^{2^{m-1}}\left(2^{m-1}\right)\left[(2i-1)d_0\right]^2 = 2\frac{2^{2m}-1}{3}$$

And:

$$\frac{E_{SQASK}^{(2^{2m})}}{\Delta_0^2} = \frac{2^{2m}-1}{6} \tag{C.3}$$

C.3.2 Average energy per symbol for CROSS constellation

The shape of a 2^{2m+1}-CROSS constellation is as depicted in Figure C.1.

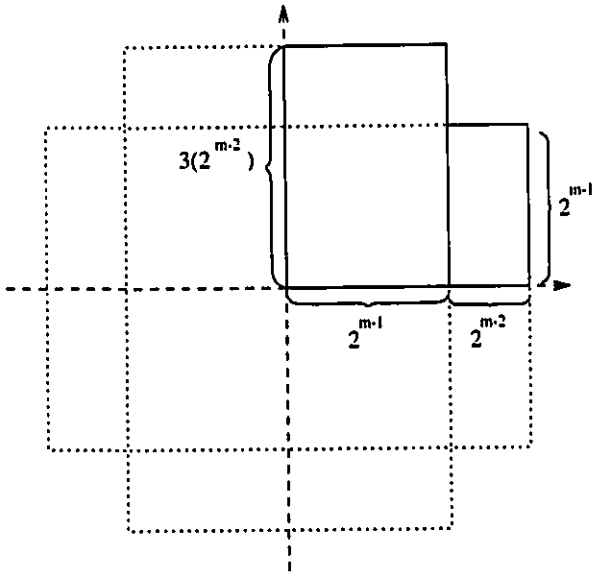

Figure C.1: Shape of a 2^{2m+1}-CROSS constellation.

It appears that m must be greater than 2, or equal to 2, for the number a points on the smaller edge of the rectangle is 2^{m-2}. As a result, the average energy per symbol for the 8-CROSS has to be calculated differently.

The square and rectangle pointed out on Figure C.1 are used to proceed calculation of the sum of the square of the abscissas, the same horizontal share the ordinates. This leads to:

$$E_S = \frac{1}{2^{2m-1}}[4][2]\left\{2^{m-1}\sum_{i=1}^{3\times 2^{m-2}}\left[(2i-1)\,d_0\right]^2 + 2^{m-2}\sum_{i=1}^{2^{m-1}}\left[(2i-1)\,d_0\right]^2\right\}$$

Combined with Equation (C.1), it simplifies to:

$$E_S = \frac{1}{6}\left[3\left(9 \times 2^{2m-2} - 1\right) + 4 \times 2^{2m-2} - 1\right] d_0^2$$

and finally:

$$\frac{E_{S_{CROSS}}^{(2^{2m+1})}}{\Delta_0^2} = \frac{\frac{31}{32}2^{2m+1} - 1}{6} \qquad \text{(C.4)}$$

Appendix D

Review of some Galois fields properties and their use in RS codes

A Galois field $GF[q = 2^{m_{RS}}]$ is constructed from a unique minimal primitive polynomial of degree m_{RS}. If α is a root of such a polynomial, $\alpha^{q-1} = 1$ and the set $\{1, \alpha, \ldots, \alpha^{q-2}\}$ forms a cyclic group. With the primitive polynomial, any power of α can be expressed in terms of the m_{RS} first ones. The additive law is XOR ($x + x = 0$), and a conventional multiplication over the set $\{0, 1, \alpha, \ldots, \alpha^{q-1}\}$ which matches the Galois field is used. Each codeword has a polynomial format $\mathbf{c}(x)$, with coefficients which are Galois field elements. A generator polynomial is used to code the information polynomial and defined as:

$$g_{RS(n_{RS},k_{RS},t)}(x) = \prod_{i=1}^{2t} \left(x - \alpha^i\right) \tag{D.1}$$

D.1 RS coding

If $\mathbf{i}(x)$ is a polynomial formed by the k_{RS} input RS symbols, let the Euclidean division of $\mathbf{i}(x)$ by the generator polynomial $g_{RS}(x)$ be:

$$\begin{cases} x^{2t}\ \mathbf{i}(x) = \mathbf{q}(x)g_{RS}(x) + \mathbf{r}(x) \\ 0 \leq \deg(\mathbf{r}) \leq 2t - 1 \end{cases} \tag{D.2}$$

then the corresponding codeword is defined by:

$$\mathbf{c}(x) = x^{2t}\ \mathbf{i}(x) + \mathbf{r}(x) = \mathbf{q}(x)\ g(x) \tag{D.3}$$

This way, $\mathbf{c}(x)$ is the polynomial of degree n_{RS} with at least $2t$ roots $\{\alpha, \ldots, \alpha^{2t}\}$ and which is the closest to $x^{2t}\ \mathbf{i}(x)$.

The RS codes are not perfect codes: there exist some n_{RS}-degree polynomials that have more than t symbols in error, and for which classic decoding scheme is not able to recover the closest codeword, even though it surely detects that the number of symbols in error with respect to a possible codeword is t, or at least t. Some improved algorithms have been developed to avoid the failures of RS decoding [4, 9]. This has not been covered by the simulations reported in this thesis.

As far as classic decoding is concerned, there exist at least two algorithms to process the received corrupted word $\mathbf{v}(x) = \mathbf{c}(x) + \mathbf{e}(x)$. In this thesis, the Peterson-Gorenstein-Zierler algorithm [2, 7] has been implemented rather than the Berlekamp-Massey one [6, 3]. It is described in section D.2.

D.2 RS decoding: the Peterson-Gorenstein-Zierler algorithm

We assume here that the received codeword $\mathbf{v}(\mathbf{x})$ has less than t symbols in errors. If it has more, step 2 of the algorithm detects $\nu = t$ errors. Then, it is possible that step 4 fails: the error locating polynomial has at most t roots in the Galois field but

may have strictly less or even none. In such a case, the received codeword is decoded with no correction.

Peterson-Gorenstein-Zierler algorithm for RS decoding

1. Calculate the syndromes $S_i = \mathbf{v}(\alpha^i)$, for $i = 1, \ldots, 2t$

2. Determine the highest integer ν such that:

$$\begin{cases} \nu < \mu \leq 2t, \det(M_\mu) = 0 \\ \det(M_\nu) \neq 0 \end{cases} \text{ with } M_\mu = \begin{pmatrix} S_1 & S_2 & \cdots & S_\mu \\ S_2 & S_3 & \cdots & S_{\mu+1} \\ \vdots & \vdots & \ddots & \vdots \\ S_\mu & S_{\mu+1} & \cdots & S_{2\mu-1} \end{pmatrix}$$

3. Calculate the coefficients of the error locating polynomial $\Lambda(x)$ according to:

$$\begin{pmatrix} \Lambda_\nu \\ \Lambda_{\nu-1} \\ \vdots \\ \Lambda_1 \end{pmatrix} = (M_\nu)^{-1} \begin{pmatrix} S_{\nu+1} \\ S_{\nu+2} \\ \vdots \\ S_{2\nu} \end{pmatrix}$$

4. Find the roots of Λ with a Chien search, that is calculate $\Lambda(\alpha^i)$ for all i until ν roots are found. The roots $X_l = \alpha^{p_l}$ for $l = 1, \ldots, \nu$ determine the position of the errors.

5. Calculate the weight of the ν errors by solving the system:

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_\nu \end{pmatrix} = \begin{pmatrix} X_1 & X_2 & \cdots & X_\nu \\ X_1^2 & X_2^2 & \cdots & X_\nu^2 \\ \vdots & \vdots & \ddots & \vdots \\ X_1^\nu & X_2^\nu & \cdots & X_\nu^\nu \end{pmatrix}^{-1} \begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_\nu \end{pmatrix}$$

6. The error correcting polynomial is $\hat{\mathbf{e}}(x)$ defined by:

$$\hat{\mathbf{e}}(x) = \sum_{l=1}^{\nu} Y_l x^{p_l}$$

and the corrected codeword:

$$\hat{\mathbf{c}}(x) = \mathbf{v}(x) + \hat{\mathbf{e}}(x)$$

Appendix E

Galois field $GF[256]$

Table E.1: Galois field $GF[q] = GF[256]$

element of $GF[256]$	binary	decimal
$\alpha^0 = 1$	00000001	1
$\alpha^1 = \alpha$	00000010	2
$\alpha^2 = \alpha^2$	00000100	4
$\alpha^3 = \alpha^3$	00001000	8
$\alpha^4 = \alpha^4$	00010000	16
$\alpha^5 = \alpha^5$	00100000	32
$\alpha^6 = \alpha^6$	01000000	64
$\alpha^7 = \alpha^7$	10000000	128
$\alpha^8 = \alpha^4+\alpha^3+\alpha^2+1$	00011101	29
$\alpha^9 = \alpha^5+\alpha^4+\alpha^3+\alpha$	00111010	58
$\alpha^{10} = \alpha^6+\alpha^5+\alpha^4+\alpha^2$	01110100	116
$\alpha^{11} = \alpha^7+\alpha^6+\alpha^5+\alpha^3$	11101000	232
$\alpha^{12} = \alpha^7+\alpha^6+\alpha^3+\alpha^2+1$	11001101	205
$\alpha^{13} = \alpha^7+\alpha^2+\alpha+1$	10000111	135
$\alpha^{14} = \alpha^4+\alpha+1$	00010011	19
$\alpha^{15} = \alpha^5+\alpha^2+\alpha$	00100110	38
$\alpha^{16} = \alpha^6+\alpha^3+\alpha^2$	01001100	76
$\alpha^{17} = \alpha^7+\alpha^4+\alpha^3$	10011000	152
$\alpha^{18} = \alpha^5+\alpha^3+\alpha^2+1$	00101101	45
$\alpha^{19} = \alpha^6+\alpha^4+\alpha^3+\alpha$	01011010	90
$\alpha^{20} = \alpha^7+\alpha^5+\alpha^4+\alpha^2$	10110100	180
$\alpha^{21} = \alpha^6+\alpha^5+\alpha^4+\alpha^2+1$	01110101	117
$\alpha^{22} = \alpha^7+\alpha^6+\alpha^5+\alpha^3+\alpha$	11101010	234
$\alpha^{23} = \alpha^7+\alpha^6+\alpha^3+1$	11001001	201
$\alpha^{24} = \alpha^7+\alpha^3+\alpha^2+\alpha+1$	10001111	143
$\alpha^{25} = \alpha+1$	00000011	3
$\alpha^{26} = \alpha^2+\alpha$	00000110	6
$\alpha^{27} = \alpha^3+\alpha^2$	00001100	12
$\alpha^{28} = \alpha^4+\alpha^3$	00011000	24
$\alpha^{29} = \alpha^5+\alpha^4$	00110000	48
$\alpha^{30} = \alpha^6+\alpha^5$	01100000	96

Table E: Galois field $GF[q] = GF[256]$ (part 2)

element of $GF[256]$	binary	decimal
$\alpha^{31} = \alpha^7+\alpha^6$	11000000	192
$\alpha^{32} = \alpha^7 +\alpha^4+\alpha^3+\alpha^2 +1$	10011101	157
$\alpha^{33} = \alpha^5 +\alpha^2+\alpha+1$	00100111	39
$\alpha^{34} = \alpha^6 +\alpha^3+\alpha^2+\alpha$	01001110	78
$\alpha^{35} = \alpha^7 +\alpha^4+\alpha^3+\alpha^2$	10011100	156
$\alpha^{36} = \alpha^5 +\alpha^2 +1$	00100101	37
$\alpha^{37} = \alpha^6 +\alpha^3 +\alpha$	01001010	74
$\alpha^{38} = \alpha^7 +\alpha^4 +\alpha^2$	10010100	148
$\alpha^{39} = \alpha^5+\alpha^4 +\alpha^2 +1$	00110101	53
$\alpha^{40} = \alpha^6+\alpha^5 +\alpha^3 +\alpha$	01101010	106
$\alpha^{41} = \alpha^7+\alpha^6 +\alpha^4 +\alpha^2$	11010100	212
$\alpha^{42} = \alpha^7 +\alpha^5+\alpha^4 +\alpha^2 +1$	10110101	181
$\alpha^{43} = \alpha^6+\alpha^5+\alpha^4 +\alpha^2+\alpha+1$	01110111	119
$\alpha^{44} = \alpha^7+\alpha^6+\alpha^5 +\alpha^3+\alpha^2+\alpha$	11101110	238
$\alpha^{45} = \alpha^7+\alpha^6 +1$	11000001	193
$\alpha^{46} = \alpha^7 +\alpha^4+\alpha^3+\alpha^2+\alpha+1$	10011111	159
$\alpha^{47} = \alpha^5 +\alpha+1$	00100011	35
$\alpha^{48} = \alpha^6 +\alpha^2+\alpha$	01000110	70
$\alpha^{49} = \alpha^7 +\alpha^3+\alpha^2$	10001100	140
$\alpha^{50} = \alpha^2 +1$	00000101	5
$\alpha^{51} = \alpha^3 +\alpha$	00001010	10
$\alpha^{52} = \alpha^4 +\alpha^2$	00010100	20
$\alpha^{53} = \alpha^5 +\alpha^3$	00101000	40
$\alpha^{54} = \alpha^6 +\alpha^4$	01010000	80
$\alpha^{55} = \alpha^7 +\alpha^5$	10100000	160
$\alpha^{56} = \alpha^6 +\alpha^4+\alpha^3+\alpha^2 +1$	01011101	93
$\alpha^{57} = \alpha^7 +\alpha^5+\alpha^4+\alpha^3 +\alpha$	10111010	186
$\alpha^{58} = \alpha^6+\alpha^5 +\alpha^3 +1$	01101001	105
$\alpha^{59} = \alpha^7+\alpha^6 +\alpha^4 +\alpha$	11010010	210
$\alpha^{60} = \alpha^7 +\alpha^5+\alpha^4+\alpha^3 +1$	10111001	185
$\alpha^{61} = \alpha^6+\alpha^5 +\alpha^3+\alpha^2+\alpha+1$	01101111	111
$\alpha^{62} = \alpha^7+\alpha^6 +\alpha^4+\alpha^3+\alpha^2+\alpha$	11011110	222
$\alpha^{63} = \alpha^7 +\alpha^5 +1$	10100001	161
$\alpha^{64} = \alpha^6 +\alpha^4+\alpha^3+\alpha^2+\alpha+1$	01011111	95
$\alpha^{65} = \alpha^7 +\alpha^5+\alpha^4+\alpha^3+\alpha^2+\alpha$	10111110	190
$\alpha^{66} = \alpha^6+\alpha^5 +1$	01100001	97
$\alpha^{67} = \alpha^7+\alpha^6 +\alpha$	11000010	194
$\alpha^{68} = \alpha^7 +\alpha^4+\alpha^3 +1$	10011001	153
$\alpha^{69} = \alpha^5 +\alpha^3+\alpha^2+\alpha+1$	00101111	47
$\alpha^{70} = \alpha^6 +\alpha^4+\alpha^3+\alpha^2+\alpha$	01011110	94
$\alpha^{71} = \alpha^7 +\alpha^5+\alpha^4+\alpha^3+\alpha^2$	10111100	188
$\alpha^{72} = \alpha^6+\alpha^5 +\alpha^2 +1$	01100101	101
$\alpha^{73} = \alpha^7+\alpha^6 +\alpha^3 +\alpha$	11001010	202
$\alpha^{74} = \alpha^7 +\alpha^3 +1$	10001001	137
$\alpha^{75} = \alpha^3+\alpha^2+\alpha+1$	00001111	15
$\alpha^{76} = \alpha^4+\alpha^3+\alpha^2+\alpha$	00011110	30
$\alpha^{77} = \alpha^5+\alpha^4+\alpha^3+\alpha^2$	00111100	60
$\alpha^{78} = \alpha^6+\alpha^5+\alpha^4+\alpha^3$	01111000	120
$\alpha^{79} = \alpha^7+\alpha^6+\alpha^5+\alpha^4$	11110000	240
$\alpha^{80} = \alpha^7+\alpha^6+\alpha^5+\alpha^4+\alpha^3+\alpha^2 +1$	11111101	253
$\alpha^{81} = \alpha^7+\alpha^6+\alpha^5 +\alpha^2+\alpha+1$	11100111	231
$\alpha^{82} = \alpha^7+\alpha^6 +\alpha^4 +\alpha+1$	11010011	211
$\alpha^{83} = \alpha^7 +\alpha^5+\alpha^4+\alpha^3 +\alpha+1$	10111011	187
$\alpha^{84} = \alpha^6+\alpha^5 +\alpha^3 +\alpha+1$	01101011	107
$\alpha^{85} = \alpha^7+\alpha^6 +\alpha^4 +\alpha^2+\alpha$	11010110	214
$\alpha^{86} = \alpha^7 +\alpha^5+\alpha^4 +1$	10110001	177

Table E: Galois field $GF[q] = GF[256]$ (part 3)

element of $GF[256]$	binary	decimal
$\alpha^{87} = \alpha^6+\alpha^5+\alpha^4+\alpha^3+\alpha^2+\alpha+1$	01111111	127
$\alpha^{88} = \alpha^7+\alpha^6+\alpha^5+\alpha^4+\alpha^3+\alpha^2+\alpha$	11111110	254
$\alpha^{89} = \alpha^7+\alpha^6+\alpha^5+1$	11100001	225
$\alpha^{90} = \alpha^7+\alpha^6+\alpha^4+\alpha^3+\alpha^2+\alpha+1$	11011111	223
$\alpha^{91} = \alpha^7+\alpha^5+\alpha+1$	10100011	163
$\alpha^{92} = \alpha^6+\alpha^4+\alpha^3+\alpha+1$	01011011	91
$\alpha^{93} = \alpha^7+\alpha^5+\alpha^4+\alpha^2+\alpha$	10110110	182
$\alpha^{94} = \alpha^6+\alpha^5+\alpha^4+1$	01110001	113
$\alpha^{95} = \alpha^7+\alpha^6+\alpha^5+\alpha$	11100010	226
$\alpha^{96} = \alpha^7+\alpha^6+\alpha^4+\alpha^3+1$	11011001	217
$\alpha^{97} = \alpha^7+\alpha^5+\alpha^3+\alpha^2+\alpha+1$	10101111	175
$\alpha^{98} = \alpha^6+\alpha+1$	01000011	67
$\alpha^{99} = \alpha^7+\alpha^2+\alpha$	10000110	134
$\alpha^{100} = \alpha^4+1$	00010001	17
$\alpha^{101} = \alpha^5+\alpha$	00100010	34
$\alpha^{102} = \alpha^6+\alpha^2$	01000100	68
$\alpha^{103} = \alpha^7+\alpha^3$	10001000	136
$\alpha^{104} = \alpha^3+\alpha^2+1$	00001101	13
$\alpha^{105} = \alpha^4+\alpha^3+\alpha$	00011010	26
$\alpha^{106} = \alpha^5+\alpha^4+\alpha^2$	00110100	52
$\alpha^{107} = \alpha^6+\alpha^5+\alpha^3$	01101000	104
$\alpha^{108} = \alpha^7+\alpha^6+\alpha^4$	11010000	208
$\alpha^{109} = \alpha^7+\alpha^5+\alpha^4+\alpha^3+\alpha^2+1$	10111101	189
$\alpha^{110} = \alpha^6+\alpha^5+\alpha^2+\alpha+1$	01100111	103
$\alpha^{111} = \alpha^7+\alpha^6+\alpha^3+\alpha^2+\alpha$	11001110	206
$\alpha^{112} = \alpha^7+1$	10000001	129
$\alpha^{113} = \alpha^4+\alpha^3+\alpha^2+\alpha+1$	00011111	31
$\alpha^{114} = \alpha^5+\alpha^4+\alpha^3+\alpha^2+\alpha$	00111110	62
$\alpha^{115} = \alpha^6+\alpha^5+\alpha^4+\alpha^3+\alpha^2$	01111100	124
$\alpha^{116} = \alpha^7+\alpha^6+\alpha^5+\alpha^4+\alpha^3$	11111000	248
$\alpha^{117} = \alpha^7+\alpha^6+\alpha^5+\alpha^3+\alpha^2+1$	11101101	237
$\alpha^{118} = \alpha^7+\alpha^6+\alpha^2+\alpha+1$	11000111	199
$\alpha^{119} = \alpha^7+\alpha^4+\alpha+1$	10010011	147
$\alpha^{120} = \alpha^5+\alpha^4+\alpha^3+\alpha+1$	00111011	59
$\alpha^{121} = \alpha^6+\alpha^5+\alpha^4+\alpha^2+\alpha$	01110110	118
$\alpha^{122} = \alpha^7+\alpha^6+\alpha^5+\alpha^3+\alpha^2$	11101100	236
$\alpha^{123} = \alpha^7+\alpha^6+\alpha^2+1$	11000101	197
$\alpha^{124} = \alpha^7+\alpha^4+\alpha^2+\alpha+1$	10010111	151
$\alpha^{125} = \alpha^5+\alpha^4+\alpha+1$	00110011	51
$\alpha^{126} = \alpha^6+\alpha^5+\alpha^2+\alpha$	01100110	102
$\alpha^{127} = \alpha^7+\alpha^6+\alpha^3+\alpha^2$	11001100	204
$\alpha^{128} = \alpha^7+\alpha^2+1$	10000101	133
$\alpha^{129} = \alpha^4+\alpha^2+\alpha+1$	00010111	23
$\alpha^{130} = \alpha^5+\alpha^3+\alpha^2+\alpha$	00101110	46
$\alpha^{131} = \alpha^6+\alpha^4+\alpha^3+\alpha^2$	01011100	92
$\alpha^{132} = \alpha^7+\alpha^5+\alpha^4+\alpha^3$	10111000	184
$\alpha^{133} = \alpha^6+\alpha^5+\alpha^3+\alpha^2+1$	01101101	109
$\alpha^{134} = \alpha^7+\alpha^6+\alpha^4+\alpha^3+\alpha$	11011010	218
$\alpha^{135} = \alpha^7+\alpha^5+\alpha^3+1$	10101001	169
$\alpha^{136} = \alpha^6+\alpha^3+\alpha^2+\alpha+1$	01001111	79
$\alpha^{137} = \alpha^7+\alpha^4+\alpha^3+\alpha^2+\alpha$	10011110	158
$\alpha^{138} = \alpha^5+1$	00100001	33
$\alpha^{139} = \alpha^6+\alpha$	01000010	66
$\alpha^{140} = \alpha^7+\alpha^2$	10000100	132
$\alpha^{141} = \alpha^4+\alpha^2+1$	00010101	21
$\alpha^{142} = \alpha^5+\alpha^3+\alpha$	00101010	42

Table E: Galois field $GF[q] = GF[256]$ (part 4)

element of $GF[256]$	binary	decimal
$\alpha^{143} = \alpha^6 + \alpha^4 + \alpha^2$	01010100	84
$\alpha^{144} = \alpha^7 + \alpha^5 + \alpha^3$	10101000	168
$\alpha^{145} = \alpha^6 + \alpha^3 + \alpha^2 + 1$	01001101	77
$\alpha^{146} = \alpha^6 + \alpha^4 + \alpha^2$	01010100	84
$\alpha^{147} = \alpha^7 + \alpha^5 + \alpha^3$	10101000	168
$\alpha^{148} = \alpha^7 + \alpha^4 + \alpha^3 + \alpha$	10011010	154
$\alpha^{149} = \alpha^5 + \alpha^3 + 1$	00101001	41
$\alpha^{150} = \alpha^6 + \alpha^4 + \alpha$	01010010	82
$\alpha^{151} = \alpha^7 + \alpha^5 + \alpha^2$	10100100	164
$\alpha^{152} = \alpha^6 + \alpha^4 + \alpha^2 + 1$	01010101	85
$\alpha^{153} = \alpha^7 + \alpha^5 + \alpha^3 + \alpha$	10101010	170
$\alpha^{154} = \alpha^6 + \alpha^3 + 1$	01001001	73
$\alpha^{155} = \alpha^7 + \alpha^4 + \alpha$	10010010	146
$\alpha^{156} = \alpha^5 + \alpha^4 + \alpha^3 + 1$	00111001	57
$\alpha^{157} = \alpha^6 + \alpha^5 + \alpha^4 + \alpha$	01110010	114
$\alpha^{158} = \alpha^7 + \alpha^6 + \alpha^5 + \alpha^2$	11100100	228
$\alpha^{159} = \alpha^7 + \alpha^6 + \alpha^4 + \alpha^2 + 1$	11010101	213
$\alpha^{160} = \alpha^7 + \alpha^5 + \alpha^4 + \alpha^2 + \alpha + 1$	10110111	183
$\alpha^{161} = \alpha^6 + \alpha^5 + \alpha^4 + \alpha + 1$	01110011	115
$\alpha^{162} = \alpha^7 + \alpha^6 + \alpha^5 + \alpha^2 + \alpha$	11100110	230
$\alpha^{163} = \alpha^7 + \alpha^6 + \alpha^4 + 1$	11010001	209
$\alpha^{164} = \alpha^7 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1$	10111111	191
$\alpha^{165} = \alpha^6 + \alpha^5 + \alpha + 1$	01100011	99
$\alpha^{166} = \alpha^7 + \alpha^6 + \alpha^2 + \alpha$	11000110	198
$\alpha^{167} = \alpha^7 + \alpha^4 + 1$	10010001	145
$\alpha^{168} = \alpha^5 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1$	00111111	63
$\alpha^{169} = \alpha^6 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha$	01111110	126
$\alpha^{170} = \alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha^2$	11111100	252
$\alpha^{171} = \alpha^7 + \alpha^6 + \alpha^5 + \alpha^2 + 1$	11100101	229
$\alpha^{172} = \alpha^7 + \alpha^6 + \alpha^4 + \alpha^2 + \alpha + 1$	11010111	215
$\alpha^{173} = \alpha^7 + \alpha^5 + \alpha^4 + \alpha + 1$	10110011	179
$\alpha^{174} = \alpha^6 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha + 1$	01111011	123
$\alpha^{175} = \alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^2 + \alpha$	11110110	246
$\alpha^{176} = \alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + 1$	11110001	241
$\alpha^{177} = \alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1$	11111111	255
$\alpha^{178} = \alpha^7 + \alpha^6 + \alpha^5 + \alpha + 1$	11100011	227
$\alpha^{179} = \alpha^7 + \alpha^6 + \alpha^4 + \alpha^3 + \alpha + 1$	11011011	219
$\alpha^{180} = \alpha^7 + \alpha^5 + \alpha^3 + \alpha + 1$	10101011	171
$\alpha^{181} = \alpha^6 + \alpha^3 + \alpha + 1$	01001011	75
$\alpha^{182} = \alpha^7 + \alpha^4 + \alpha^2 + \alpha$	10010110	150
$\alpha^{183} = \alpha^5 + \alpha^4 + 1$	00110001	49
$\alpha^{184} = \alpha^6 + \alpha^5 + \alpha$	01100010	98
$\alpha^{185} = \alpha^7 + \alpha^6 + \alpha^2$	11000100	196
$\alpha^{186} = \alpha^7 + \alpha^4 + \alpha^2 + 1$	10010101	149
$\alpha^{187} = \alpha^5 + \alpha^4 + \alpha^2 + \alpha + 1$	00110111	55
$\alpha^{188} = \alpha^6 + \alpha^5 + \alpha^3 + \alpha^2 + \alpha$	01101110	110
$\alpha^{189} = \alpha^7 + \alpha^6 + \alpha^4 + \alpha^3 + \alpha^2$	11011100	220
$\alpha^{190} = \alpha^7 + \alpha^5 + \alpha^2 + 1$	10100101	165
$\alpha^{191} = \alpha^6 + \alpha^4 + \alpha^2 + \alpha + 1$	01010111	87
$\alpha^{192} = \alpha^7 + \alpha^5 + \alpha^3 + \alpha^2 + \alpha$	10101110	174
$\alpha^{193} = \alpha^6 + 1$	01000001	65
$\alpha^{194} = \alpha^7 + \alpha$	10000010	130
$\alpha^{195} = \alpha^4 + \alpha^3 + 1$	00011001	25
$\alpha^{196} = \alpha^5 + \alpha^4 + \alpha$	00110010	50
$\alpha^{197} = \alpha^7 + \alpha^3 + \alpha^2 + 1$	10001101	141
$\alpha^{198} = \alpha^2 + \alpha + 1$	00000111	7

Table E: Galois field $GF[q] = GF[256]$ (part 5)

element of $GF[256]$	binary	decimal
$\alpha^{195} = \alpha^6+\alpha^5 \quad +\alpha^2$	01100100	100
$\alpha^{196} = \alpha^7+\alpha^6 \quad +\alpha^3$	11001000	200
$\alpha^{199} = \quad \alpha^3+\alpha^2+\alpha$	00001110	14
$\alpha^{200} = \quad \alpha^4+\alpha^3+\alpha^2$	00011100	28
$\alpha^{201} = \quad \alpha^5+\alpha^4+\alpha^3$	00111000	56
$\alpha^{202} = \alpha^6+\alpha^5+\alpha^4$	01110000	112
$\alpha^{203} = \alpha^7+\alpha^6+\alpha^5$	11100000	224
$\alpha^{204} = \alpha^7+\alpha^6 \quad +\alpha^4+\alpha^3+\alpha^2 \quad +1$	11011101	221
$\alpha^{205} = \alpha^7 \quad +\alpha^5 \quad +\alpha^2+\alpha+1$	10100111	167
$\alpha^{206} = \quad \alpha^6 \quad +\alpha^4 \quad +\alpha+1$	01010011	83
$\alpha^{207} = \alpha^7 \quad +\alpha^5 \quad +\alpha^2+\alpha$	10100110	166
$\alpha^{208} = \quad \alpha^6 \quad +\alpha^4 \quad +1$	01010001	81
$\alpha^{209} = \alpha^7 \quad +\alpha^5 \quad +\alpha$	10100010	162
$\alpha^{210} = \quad \alpha^6 \quad +\alpha^4+\alpha^3 \quad +1$	01011001	89
$\alpha^{211} = \alpha^7 \quad +\alpha^5+\alpha^4 \quad +\alpha$	10110010	178
$\alpha^{212} = \quad \alpha^6+\alpha^5+\alpha^4+\alpha^3 \quad +1$	01111001	121
$\alpha^{213} = \alpha^7+\alpha^6+\alpha^5+\alpha^4 \quad +\alpha$	11110010	242
$\alpha^{214} = \alpha^7+\alpha^6+\alpha^5+\alpha^4+\alpha^3 \quad +1$	11111001	249
$\alpha^{215} = \alpha^7+\alpha^6+\alpha^5 \quad +\alpha^3+\alpha^2+\alpha+1$	11101111	239
$\alpha^{216} = \alpha^7+\alpha^6 \quad +\alpha+1$	11000011	195
$\alpha^{217} = \alpha^7 \quad +\alpha^4+\alpha^3 \quad +\alpha+1$	10011011	155
$\alpha^{218} = \quad \alpha^5 \quad +\alpha^3 \quad +\alpha+1$	00101011	43
$\alpha^{219} = \quad \alpha^6 \quad +\alpha^4 \quad +\alpha^2+\alpha$	01010110	86
$\alpha^{220} = \alpha^7 \quad +\alpha^5 \quad +\alpha^3+\alpha^2$	10101100	172
$\alpha^{221} = \quad \alpha^6 \quad +\alpha^2 \quad +1$	01000101	69
$\alpha^{222} = \alpha^7 \quad +\alpha^3 \quad +\alpha$	10001010	138
$\alpha^{223} = \quad \alpha^3 \quad +1$	00001001	9
$\alpha^{224} = \quad \alpha^4 \quad +\alpha$	00010010	18
$\alpha^{225} = \quad \alpha^5 \quad +\alpha^2$	00100100	36
$\alpha^{226} = \quad \alpha^6 \quad +\alpha^3$	01001000	72
$\alpha^{227} = \alpha^7 \quad +\alpha^4$	10010000	144
$\alpha^{228} = \quad \alpha^5+\alpha^4+\alpha^3+\alpha^2 \quad +1$	00111101	61
$\alpha^{229} = \quad \alpha^6+\alpha^5+\alpha^4+\alpha^3 \quad +\alpha$	01111010	122
$\alpha^{230} = \alpha^7+\alpha^6+\alpha^5+\alpha^4 \quad +\alpha^2$	11110100	244
$\alpha^{231} = \alpha^7+\alpha^6+\alpha^5+\alpha^4 \quad +\alpha^2 \quad +1$	11110101	245
$\alpha^{232} = \alpha^7+\alpha^6+\alpha^5+\alpha^4 \quad +\alpha^2+\alpha+1$	11110111	247
$\alpha^{233} = \alpha^7+\alpha^6+\alpha^5+\alpha^4 \quad +\alpha+1$	11110011	243
$\alpha^{234} = \alpha^7+\alpha^6+\alpha^5+\alpha^4+\alpha^3 \quad +\alpha+1$	11111011	251
$\alpha^{235} = \alpha^7+\alpha^6+\alpha^5 \quad +\alpha^3 \quad +\alpha+1$	11101011	235
$\alpha^{236} = \alpha^7+\alpha^6 \quad +\alpha^3 \quad +\alpha+1$	11001011	203
$\alpha^{237} = \alpha^7 \quad +\alpha^3 \quad +\alpha+1$	10001011	139
$\alpha^{238} = \quad \alpha^3 \quad +\alpha+1$	00001011	11
$\alpha^{239} = \quad \alpha^4 \quad +\alpha^2+\alpha$	00010110	22
$\alpha^{240} = \quad \alpha^5 \quad +\alpha^3+\alpha^2$	00101100	44
$\alpha^{241} = \quad \alpha^6 \quad +\alpha^4+\alpha^3$	01011000	88
$\alpha^{242} = \alpha^7 \quad +\alpha^5+\alpha^4$	10110000	176
$\alpha^{243} = \quad \alpha^6+\alpha^5+\alpha^4+\alpha^3+\alpha^2 \quad +1$	01111101	125
$\alpha^{244} = \alpha^7+\alpha^6+\alpha^5+\alpha^4+\alpha^3 \quad +\alpha$	11111010	250
$\alpha^{247} = \alpha^7 \quad +\alpha+1$	10000011	131
$\alpha^{248} = \quad \alpha^4+\alpha^3 \quad +\alpha+1$	00011011	27
$\alpha^{249} = \quad \alpha^5+\alpha^4 \quad +\alpha^2+\alpha$	00110110	54
$\alpha^{250} = \quad \alpha^6+\alpha^5 \quad +\alpha^3+\alpha^2$	01101100	108
$\alpha^{251} = \alpha^7+\alpha^6 \quad +\alpha^4+\alpha^3$	11011000	216
$\alpha^{252} = \alpha^7 \quad +\alpha^5 \quad +\alpha^3+\alpha^2 \quad +1$	10101101	173
$\alpha^{253} = \quad \alpha^6 \quad +\alpha^2+\alpha+1$	01000111	71
$\alpha^{254} = \alpha^7 \quad +\alpha^3+\alpha^2+\alpha$	10001110	142

Bibliography

[1] E. BIGLIERI, D. DIVSALAR, P. J. MCLANE, and S. SIMON. *Introduction to Trellis-Coded Modulation with Applications*, chapter Performance Evaluation, pages 99–125. Macmillan Publishing Company, 1991.

[2] R.E. BLAHUT. *Theory and Practice of Error Control Codes.* Addison-Wesley, Reading, Mass., 1984.

[3] G COHEN, J.-L. DORNSTETTER and P. GODLEWSKI. *Codes correcteurs d'erreurs: une introduction au codage algébrique.* Masson, Paris, 1992. Collection Technique et Scientifique des Télécommunications *CNET-ENST.*

[4] Arne DÜR. Avoiding Decoder Malfunction in the Peterson-Gorenstein-Zierler Decoder. *IEEE Trans. Inform. Theory*, IT-39:640–643, Mar. 1993.

[5] C. HEEGARD, S.A. LERY, and W.H. PAIK. Practical Coding for QAM Transmission of HDTV. *IEEE J. Select. Areas Commun.*, JSAC-11(1):111–118, Jan. 1993.

[6] S. LIN and D. COSTELLO. *Error Control Coding: Fundamentals and Applications.* Prentice-Hall, Inc., Englewood Cliffs, N.J., New-Jersey, 1983.

[7] V. PLESS. *Introduction to the Theory of Error-Correcting Codes.* Wiley-Interscience Series, New-York, 1989.

[8] J.-R. PORATH. Algorithms for converting convolutional codes from feedback to feedforward form and vice versa. *Electron. Lett.*, EL-25(15):1008-1009, Jul. 1989.

[9] Meera SRINIVASAN and Dilip V. SARWATE. Malfunction in the Peterson-Gorenstein-Zierler Decoder. *IEEE Trans. Inform. Theory*, IT-40:1649–1653, Sept. 1994.

[10] G. UNGERBOECK. Channel Coding with Multilevel/Phase signals. *IEEE Trans. Inform. Theory*, IT-28:55–67, Jan. 1982.

[11] G. UNGERBOECK. Trellis-Coded Modulation with Redundant Signals Sets. Part I: Introduction. *IEEE Commun. Magazine*, MAG-25:5–11, Feb. 1987.

[12] G. UNGERBOECK. Trellis-Coded Modulation with Redundant Signals Sets. Part II: State of the art. *IEEE Commun. Magazine*, MAG-25:12–21, Feb. 1987.

[13] Y. WU and B. CARON. Digital Television Terrestrial Broadcasting. *IEEE Commun. Magazine*, MAG-32(5):46–52, May 1994.

[14] E. ZEHAVI and J. K. WOLF. On the performance evaluation of trellis codes. *IEEE Trans. Inform. Theory*, IT-33:196–201, Mar. 1987.