Chapter 3

Theoretical limits

The recent invention of turbo codes and the rediscovery of LDPC codes have brought back into favour the theoretical limits of transmission which were reputed to be inaccessible until now. This chapter provides the conceptual bases necessary to understand and compute these limits, in particular those that correspond to real transmission situations with messages of finite length and binary modulations.

3.1 Information theory

3.1.1 Transmission channel

A *channel* is any environment where symbols can be propagated (telecommunications) or recorded (mass memories). For example, the symbols 0 and 1 of the binary alphabet can be represented by the polarity of a voltage applied to one end of a pair of conducting wires, stipulating for example that $+V$ corresponds to 1 and $-V$ to 0. Then, the polarity measure at the other end will show which binary symbol was emitted. At the emitter side, the polarity is changed at regularly spaced intervals to represent the bits of a message and will enable this message to be reconstituted at the receiver side. This scheme is far too simple to illustrate modern telecommunications systems but, generally, it is the sign of a real physical value that represents a binary symbol at the output of the channel. Usually, a binary symbol is represented by a certain waveform and the operation that associates a sequence of waveforms with the sequence of symbols of the message is the modulation. Modulation was the subject of the previous chapter.

We consider a situation where the channel is not very reliable, that is, where the observation at the receiving end does not enable the bit really emitted to be identified with certitude because an interference value, noise, independent of the emitted message and random, is added to the useful value (spurious effects of attenuation can also be added, like on the Rayleigh channel). Thermal noise is well represented by a Gaussian random process. When demodulation is performed in an optimal way, it results in a random Gaussian variable whose sign represents the best hypothesis concerning the binary symbol emitted. The channel is then characterized by its *signal to noise ratio*, defined as the ratio of the power of the useful signal to that of the perturbing noise. For a given signal to noise ratio, the decisions taken on the binary symbols emitted are assigned a constant error probability, which leads to the simple model of the binary symmetric channel.

3.1.2 An example: the binary symmetric channel

This is the simplest channel model, and it has already been mentioned in Section 1.3. A channel can generally be described by the probabilities of the transition of the symbols that are input, towards the symbols that are output. A binary symmetric channel is thus represented in Figure 3.1. This channel is *memoryless*, in the sense that it operates separately on the successive input bits. Its input symbol X and its output symbol Y are both binary. If $X = 0$ (respectively $X = 1$), there exists a probability p that $Y = 1$ (resp. $Y = 0$). p is called the error probability of the channel.

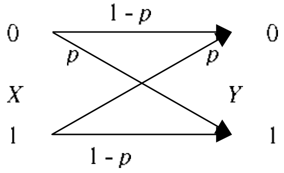


Figure 3.1 – Binary symmetric channel with error probability p. The transition probabilities of an input symbol towards an output symbol are equal two by two.

Another description of the same channel can be given in the following way: let E be a binary random variable taking value 1 with a probability $p < 1/2$ and value 0 with the probability $1-p$. The hypothesis that $p < 1/2$ does not restrict the generality of the model because changing the arbitrary signs 0 and 1 leads to replacing an initial error probability $p > 1/2$ by $1 - p < 1/2$. The behaviour of the channel can be described by the algebraic expression $Y = X \oplus E$, where X and Y are the binary variables at the input and at the output of the channel respectively, E a binary error variable, and $\oplus$ represents the modulo 2 addition.

Configurations of errors on the binary symmetric channel

Let us now suppose that we no longer consider a particular single symbol, but a set of n symbols (consecutive or not) making up a *word*, denoted

$\mathbf{x} = (x_1 x_2 \ldots x_n)$. The operation of the channel is described by the vector addition modulo 2 of $\mathbf{x}$ and of an error vector $\mathbf{e} = (e_1 e_2 \ldots e_n)$:

$$\mathbf{y} = \mathbf{x} \oplus \mathbf{e} \tag{3.1}$$

with $\mathbf{y} = (y_1 y_2 \ldots y_n)$, the notation $\oplus$ now designating the modulo 2 addition of two words, symbol to symbol. The hypothesis that the binary symmetric channel is memoryless means that the random variables e_i, $i = 1...n$, are mutually independent. The number of configurations of possible errors is 2^n, and their probability, for an error probability p of the given channel, depends only on the weight $w(e)$ of the configuration of errors e realized, defined as the number of 1 symbols that it contains. Thus, the probability of the appearance of a particular configuration of errors of weight $w(e)$ affecting a word of length n equals:

$$P_{\mathbf{e}} = p^{w(\mathbf{e})}(1-p)^{n-w(\mathbf{e})} \tag{3.2}$$

As p was assumed to be lower than 1/2, probability P_e is a decreasing function of the weight $w(\mathbf{e})$, whatever n.

The probability of the appearance of any configuration of errors of weight w equals:

$$P_w = \left(\begin{array}{c} n \\ w \end{array} \right) p^w (1-p)^{n-w} \tag{3.3}$$

The weight of the error configurations thus follows a Bernoulli distribution whose mathematical expectation (or mean) is np and the variance (the expectation of the square of the difference between its effective value and its mean) is $np(1-p)$.

Mutual information and capacity of the binary symmetric channel

To characterize a channel, we first have to measure the quantity of information that a symbol Y leaving a channel provides, on average, about the corresponding symbol that enters, X. This value called *mutual information* and whose unit is the Shannon (Sh), is defined for a discrete input and output channel by:

$$I(X;Y) = \sum_X \sum_Y \Pr(X,Y) \log_2 \frac{\Pr(X\,|Y\,)}{\Pr(X)} = \sum_X \sum_Y \Pr(X,Y) \log_2 \frac{\Pr(X,Y)}{\Pr(X)\Pr(Y)} \tag{3.4}$$

In this expression, the sums are extended to all the discrete values that X and Y can take in a given alphabet. $\Pr(X,Y)$ denote the joint probability of X and Y, $\Pr(X|Y)$ the probability of X conditionally to Y (that is, when Y is given), $\Pr(X)$ and $\Pr(Y)$ are the marginal probabilities of X and Y (that is, of each of the variables X and Y whatever the value taken by the other: $\Pr(X) = \sum_Y \Pr(X,Y)$ and $\Pr(Y) = \sum_X \Pr(X,Y)$). These different probabilities are linked according to Bayes' law:

$$\Pr(X,Y) = \Pr(X\,|Y\,)\Pr(Y) = \Pr(Y\,|X\,)\Pr(X) \tag{3.5}$$

The first equality in (3.4) defined $I(X;Y)$ as the logarithmic increase of the probability of X that results on average from the data Y, that is, the average quantity of information that the knowledge of Y provides about that of X. The second equality in (3.4), deduced from the first using (3.5), shows that this value is symmetric in X and in Y. The quantity of information that Y provides about X is therefore equal to what X provides about Y, which justifies the name of mutual information.

Mutual information is not sufficient to characterize the channel because the former also depends on the *entropy* of the source, that is, the quantity of information that it produces on average per emitted symbol. Entropy, that is, in practice the average number of bits necessary to represent each symbol, is defined by:

$$H(X) = \sum_{X} \Pr(X) \log_2 (\Pr(X))$$

The *capacity* of a channel is defined as the maximum of the mutual information of its input and output random variables with respect to all the possible probability distributions of the input variables, and it could be demonstrated that this maximum is reached for a symmetric memoryless channel when the input variable of the channel, X, has equiprobable values (which also causes the entropy of the source to be maximum). For example, for the binary symmetric channel, the capacity is given by:

$$C = 1 + p \log_2 (p) + (1-p) \log_2 (1-p) \ (\mathrm{Sh}) \tag{3.6}$$

This capacity is maximum for $p = 0$ (then it equals 1 Sh, like the entropy of the source: the channel is then "transparent") and null for $p = 1/2$, which is what we could expect since then there is total incertitude.

3.1.3 Overview of the fundamental coding theorem

The simplest code that we can imagine is the repetition code that involves emitting information bits in the form of several identical symbols. Hard decoding is performed according to the principle of a majority vote, and soft decoding by the simple addition of the samples received. If the channel is Gaussian, repetition coding provides no gain in the case of soft decoding. For example, transmitting the same symbol twice, allocating each of them half of the available energy and then reconstituting the emitted symbol by addition does not give a better result than transmitting a single symbol with all the energy available. As for the majority vote, it can only be envisaged from a triple emission and in all cases, on a Gaussian channel this procedure degrades the budget link in relation to the non-coded solution. It should however be noted that repeating messages is a widespread technique, not as a procedure for error correction coding, but as a technique for recovering packets of erroneous messages or messages lost during transmission. This technique called ARQ (Automatic Repeat reQuest) cannot be implemented in all systems, in particular in point to multipoint links (e.g. television broadcasting).

The codes are ideally efficient only if their codewords are long, in the sense that the error probability can be made arbitrarily small only if the length of these codewords tends towards infinity. In addition, a good code must keep an emission or coding rate $R = k/n$ non-null when the number k of information bits tends towards infinity. That an error-free communication is effectively possible asymptotically for a non-null emission rate is a major result of information theory, called the *fundamental theorem of channel coding*, which preceded attempts to construct practical codes, thus of finite length. This theorem was a powerful incentive in the search for ever more efficient new codes. Moreover, it presented engineers with a challenge, insofar as the proof of the theorem was based on *random coding*, whose decoding is far too complex to be envisaged in practice.

Although the mathematical proof of the fundamental theorem in its most general form contains fairly difficult mathematics, we believe that it can be easily understood with the help of the *law of large numbers*. This law simply says that experimental realizations have frequencies, defined as the ratio of the number of occurrences noted to the total number of attempts, which tend towards the probabilities of the corresponding events when the number of attempts tend towards infinity. Let us consider, for example, the game of heads and tails. With an "honest" coin, after 10000 throws we could theoretically arrive at the sequence consisting exclusively of all heads (or all tails), but with a probability that is only $2^{-10000} \approx 10^{-3010}$ (in comparison, one second represents about 10^{-18} of the time that has elapsed since the creation of the universe). In stark contrast, the probability that the frequency of the heads (or tails) is close to the mean 1/2 (belonging for example to the interval 0,47-0,53) is in the neighbourhood of 1. In a similar way, an error configuration with a weight close to np when n symbols are emitted on a binary symmetric channel of error probability p is very likely, on condition that the message sent is sufficiently long.

3.1.4 Geometrical interpretation

Consider the finite space S_n of the codewords of n bits having the minimum Hamming distance d. It contains 2^n elements that are said to be its points. In geometrical terms, saying that the number of errors is close to np with high probability means that the received word is represented by a point that, with high propability it is very close to the surface of a hypersphere with n dimensions in S_n, centred on the emitted word and whose radius is equal to the expected mean number of errors np. If the minimum distance d of the code is higher than twice this number, the point on the surface of this hypersphere is closer to the word effectively emitted than to any other codeword and therefore identifies it without ambiguity. The optimal decoding rule, which was presented in Chapter 1, can therefore be stated thus:

"*Choose the codeword closest to the received word*"

The larger n is, the smaller the probability that this rule has an erroneous result is, and this probability tends towards 0 (assuming that p is kept constant) when n tends towards infinity, provided that $d > 2np$. So d has also to tend towards infinity.

Still in geometrical terms, the construction of the best possible code can therefore be interpreted as involving choosing $M < 2^n$ points belonging to S_n in such a way that they are as far away as possible from each other (note that the inequality $M < 2^n$ implies that the code is necessarily redundant). For a given value of the error probability p of the channel (still assumed to be binary symmetric) it is clear that there is a limit to the number M of points that can be placed in S_n while maintaining the distance between these points higher than $2np$. Let $M_{\max}$ be this number. The value

$$C = \lim_{n \to \infty} \frac{\log_2 (M_{\max})}{n}$$

measures in shannons the greatest quantity of information per symbol that can be communicated without any errors through the channel, and it happens to coincide with the capacity of the channel defined in Section 3.1. No explicit procedure making it possible to determine $M_{\max}$ points in S_n while maintaining the distance between these points higher than $2np$ is generally known, except in a few simple, not very useful, cases.

3.1.5 Random coding

Random coding, that is, the construction of a code by randomly choosing its elements, is a way of choosing M scattered points in the space S_n. This method is optimal for the distribution of distances, when n tends towards infinity. Random coding enables the points to be, on average, equally distributed in all the n dimensions of S_n and it reaches a mean emission rate equal to the capacity of the channel. For a code containing M codewords of length n, it means randomly drawing each bit of a codeword independently of the others with the probability $1/2$ that it is 0 or 1, the M codewords that make up the code being drawn in the same way independently from each other. The probability of a particular codeword $\mathbf{c}$ is $P_c = 2^{-n}$. We thus obtain codewords whose weights follow a Bernoulli distribution and the probability of obtaining any codeword of weight w is given by (3.3) for $p = 1/2$, that is:

$$P_w = \left(\begin{array}{c} n \\ w \end{array} \right) 2^{-n} \tag{3.7}$$

The mathematical expectation, or mean, of this weight is $n/2$ and its variance equals $n/4$. For very large n, a good approximation of the weight distribution of the codewords obtained by random coding is a Gaussian distribution. If we replace w/n by the continuous random variable X, the probability that $X \in (x, x + \mathrm{d}x)$ is $p_X(x)\mathrm{d}x$, where:

$$p_X(x) = \sqrt{\frac{2n}{\pi}} \exp\left[2n(x - 1/2)^2\right] \tag{3.8}$$

This function has a maximum at $x = 1/2$, therefore for $w = n/2$, and takes symmetric decreasing values when x diverges from $1/2$. It is centred around its maximum $x = 1/2$ and the width of the region where it takes non-negligible values decreases as $1/\sqrt{n}$, and therefore tends towards 0 when n tends towards infinity.

Unfortunately, decoding a code obtained by random coding is impossible in practice since decoding a *single* received word would imply comparing it to *all* the codewords. Since long words are necessary for good performance, the number of codewords (2^{Rn}), and therefore the number of necessary combinations, is considerable if Rn is large, which is the case in practice. This is why research on error correcting codes has been directed towards non-random coding rules offering the path to decoding with reasonable complexity.

No general way is known for constructing a code having $M_{\max}$ codewords, for an arbitrary value of n and a given error probability p. We know with certitude, or we conjecture, that a small number of schemes are optimal for given values of M and n, for a few simple channels. In the absence of a general rule for building optimal codes, research has focused on codes satisfying a simpler criterion: that of minimum distance, that is, the greater a code's minimum distance, the better it is. The pertinence of this criterion was not questioned until the end of the 1980's. This criterion does not take into account the number of codewords at the minimum distance from a given word (or multiplicity), whereas a large value for this number leads to a degradation in performance. Turbo codes, which will be examined in the following chapters, were not initially built to satisfy this criterion. Their minimum distance can be small (at least if we compare it to the known bounds on the largest minimum distance possible and in particular the Gilbert-Varshamov bound which we shall see later) but their multiplicity is also very small. These properties mean that there can be an *error floor*, that is, a far less rapid decrease in the error probability of their decoding as a function of the signal to noise ratio when the latter is large, than when it is small. This error floor phenomenon can also be visible with LDPC codes, although the latter can be designed on the criterion of minimum distance. Be that as it may, in the case of turbo codes like in that of LDPC, since the finality of correction coding is to improve communications when the channel is bad, we could say that these codes are good when they are useful and mediocre when they are less useful.

Codes imitating random coding

A simple idea is to try to build codes "imitating" random coding, in a certain sense. Since the performance of a code depends essentially on the distribution of its distances, and that of a linear code on the distribution of its weights, we can undertake to build a linear code having a weight distribution close to that of random coding. This idea has not been much exploited directly, but we can interpret turbo codes as being a first implementation. Before returning to the design of coding procedures, we will make an interesting remark concerning codes that imitate random coding.

The probability of obtaining a codeword of length n and weight w by randomly drawing the bits 0 and 1 each with a probability of 1/2, independently of each other, is given by (3.7). Drawing a codeword 2^k times, we obtain an average number of words of weight w equal to:

$$N_{w,k} = \left(\begin{array}{c} n \\ w \end{array} \right) 2^{-(n-k)}$$

Assuming that n, k and w are large, we can express $\left(\begin{array}{c} n \\ w \end{array} \right)$ approximately, using the Stirling formula:

$$\left(\begin{array}{c} n \\ w \end{array} \right) \approx \frac{1}{\sqrt{2\pi}} \frac{n^{n+1/2}}{w^{w+1/2}\left(n-w\right)^{n-w+1/2}}$$

The minimal weight obtained on average, that is $w_{\min}$, is the largest number such that $N_{w_{\min},k}$ has value 1 for the best integer approximation. The number $N_{w_{\min},k}$ is therefore small. It will be sufficient for us to put it equal to a constant λ close to 1, which it will not be necessary to detail further because it will be eliminated from the calculation. We must therefore have:

$$2^{-(n-k)} \frac{1}{\sqrt{2\pi}} \frac{n^{n+1/2}}{w_{\min}^{w_{\min}+1/2}\left(n-w_{\min}\right)^{n-w_{\min}+1/2}} = \lambda$$

Taking the base 2 logarithms and ignoring the constant in relation to n, k and $w_{\min}$ that tend towards infinity, we obtain:

$$1 - \frac{k}{n} \approx H_2\left(w_{\min}/n\right)$$

where $H_2\left(\cdot\right)$ is the *binary entropy* function:

$$\begin{array}{rcll} H_2(x) & = & -x\log_2(x) - (1-x)\log_2(1-x) & \text{for } 0 < x < 1 \\ & = & 0 & \text{for } x = 0 \text{ or } x = 1 \end{array}$$

The weight $w_{\min}$ is the average minimal weight of a code obtained by drawing at random. Among the set of all the linear codes thus obtained (weights and distances therefore being merged), there is at least one whose minimum distance d is higher than or equal to the average weight $w_{\min}$, so that we have:

$$1 - \frac{k}{n} \leq H_2\left(d/n\right) \qquad (3.9)$$

This is the asymptotic form of the Gilbert-Varshamov bound that links the minimum distance d of the code having the greatest minimum distance possible, given the parameters k and n. It is a lower bound but, in its asymptotic form, it is very close to equality. A code whose minimum distance verifies this bound with equality is considered to be good for the minimum distance criterion. This shows that a code built with a weight distribution close to that of random coding is also good for this criterion.

3.2 Theoretical limits to performance

3.2.1 Binary input and real output channel

Only the case of the binary symmetric channel, with constant error probability p, has been considered so far. Instead of admitting a constant error probability, we can consider that the error probability in fact varies from one symbol to another because the noise sample that affects the received value varies randomly. Thus, in the presence of Gaussian noise, the value leaving the optimal demodulator is a Gaussian random variable whose sign represents the optimal decision. We will consider the channel that has this real random variable as its output value, that we denote a. It can be shown that this value is linked to the optimal decision $\hat{x}$, that is, to the best hypothesis concerning the emitted bit x, and to the "instantaneous" error probability p_a, according to the relation:

$$a = -\left(-1\right)^{\hat{x}} \ln\left(\frac{1-p_a}{p_a}\right) \tag{3.10}$$

which means, assuming p_a lower than 1/2:

$$p_a = \frac{1}{\exp\left(-(-1)^{\hat{x}}a\right)+1} = \frac{1}{\exp\left(|a|\right)+1} \tag{3.11}$$

We mean by instantaneous error probability the error probability p_a that affects the received symbol when the real value measured at the output of the channel is a. The inequality $p_a < 1/2$ makes $\ln\left(\frac{1-p_a}{p_a}\right)$ positive and then the best decision is $\hat{x} = 1$ when a is positive and $\hat{x} = 0$ when a is negative. In addition, the absolute value $|a| = \ln\left(\frac{1-p_a}{p_a}\right)$ is a decreasing function of the error probability of the decision, and it therefore measures its reliability. It is null for $p_a = 1/2$ and tends towards infinity when error probability p_a tends towards 0 (the decision then becomes absolutely reliable). The real quantity that (3.10) defines is called *relative value* or more often *log likelihood ratio* (LLR) of the corresponding binary symbol.

The capacity of the channel thus defined can be calculated as the maximum with respect to X of the mutual information $I(X;Y)$, defined by generalizing (3.4) to real $Y = a$. This generalization is possible but the expression of the capacity thus obtained will not be given here. We merely note that this capacity is higher than that of the binary symmetric channel that is deduced from it by taking a hard decision, that is, restricted to the binary symbol $Y = \hat{x}$, by a factor that increases when the signal to noise ratio of the channel decreases. It reaches $\pi/2$ when we make this ratio tend towards 0, if the noise is Gaussian. For a given signal to noise ratio, the binary input continuous output channel is therefore better than the binary symmetric channel that can be deduced from it by taking hard decisions. This channel is also simpler than the hard decision channel, since it does not have any means to take a binary decision according to the received real value. Taking a hard decision means losing the information carried by the individual variations of this value, which explains that the capacity of the soft output channel is higher.

3.2.2 Capacity of a transmission channel

Here we will consider the most general case where the input and the output of the channel are no longer only scalar values but can be vectors whose dimension N is a function of the modulation system. For example, we will have $N = 1$ for an amplitude modulation and $N = 2$ for a phase modulation with a 4-point constellation. X and Y are therefore replaced by $\mathbf{X}$ and $\mathbf{Y}$.

Capacity was introduced in Section 3.1 for a discrete input and output channel, and is defined as the maximum of the mutual information of its input and output variables, with respect to all the possible probability distributions of the variables. For any dimension of the signal space, the law remains:

$$C = \max_{p(\mathbf{X})} I\left(\mathbf{X};\mathbf{Y}\right) \tag{3.12}$$

where $I(\mathbf{X};\mathbf{Y})$ is the mutual information between $\mathbf{X}$ and $\mathbf{Y}$. When the input and the output of the channel are real values, and no longer discrete values, the probabilities are replaced by probability densities and the sums in relation (3.4) become integrals. For realizations $\mathbf{x}$ and $\mathbf{y}$ of the random variables $\mathbf{X}$ and $\mathbf{Y}$, we can write the mutual information as a function of the probabilities of $\mathbf{x}$ and $\mathbf{y}$:

$$I\left(\mathbf{X};\mathbf{Y}\right) = \underbrace{\int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty}}_{2N\text{ times}} p\left(\mathbf{x}\right) p\left(\mathbf{y}\left|\mathbf{x}\right.\right) \log_2 \frac{p\left(\mathbf{y}\left|\mathbf{x}\right.\right)}{p\left(\mathbf{y}\right)} \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} \tag{3.13}$$

To determine C, we therefore have to maximize (3.13) which is valid for all types of inputs (continuous, discrete) of any dimension N. In addition, the maximum is reached for equiprobable inputs (see Section 3.1), for which we have:

$$p\left(\mathbf{y}\right) = \frac{1}{M}\sum_{i=1}^{M} p\left(\mathbf{y}\left|\mathbf{x}_i\right.\right)$$

where M is the number of symbols or modulation order. (3.13) can then be written in the form:

$$C = \log_2(M) - \frac{1}{M}\sum_{i=1}^{M} \underbrace{\int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty}}_{N\,\text{times}} p\left(\mathbf{y}\left|\mathbf{x}_i\right.\right) \log_2\left(\frac{\sum_{j=1}^{M} p\left(\mathbf{y}\left|\mathbf{x}_j\right.\right)}{p\left(\mathbf{y}\left|\mathbf{x}_i\right.\right)}\right) \mathrm{d}\mathbf{y} \qquad (3.14)$$

According to the additional information available about the transmission, such as the type of noise on the channel, possible fading, the type of input and output (continuous, discrete) and the modulation used, (3.14) can be particularized.

Shannon limit of a band-limited continuous input and output Gaussian channel

Consider the case of a Gaussian channel, with continuous input and output. The Shannon bound [3.3] giving the maximum capacity C of such a channel is reached taking at its input a white Gaussian noise of null mean and variance σ^2, described by independent probabilities on each dimension, that is, such that:

$$p(\mathbf{x}) = \prod_{n=1}^{N} p(x_n)$$

where $\mathbf{x} = [x_1 x_2 \ldots x_N]$ is the input vector and $p(x_n) = N(0, \sigma^2)$. The mutual information is reached for equiprobable inputs, and denoting $N_0/2$ the variance of the noise, (3.14) after development gives:

$$C = \frac{N}{2}\log_2\left(1 + \frac{2\sigma^2}{N_0}\right).$$

This relation is modified to make the mean energy E_b of each of the bits and consequently the signal to noise ratio $\frac{E_b}{N_0}$. For N=2, we have:

$$C_b = \log_2\left(1 + R\frac{E_b}{N_0}\right) \qquad (3.15)$$

the capacity being expressed in bit per second per Hertz and per couple dimension. Taking $R = 1$, this leads to the ratio E_b/N_0 being limited by the normalized Shannon limit, as shown in Figure 3.2.

Capacity of a discrete input Gaussian channel

The discrete input, denoted $\mathbf{x} = \mathbf{x}_i\,, i = 1, \cdots, M$, is typically the result of a modulation performed before transmission. The inputs $\mathbf{x}_i$ belong to a set of M

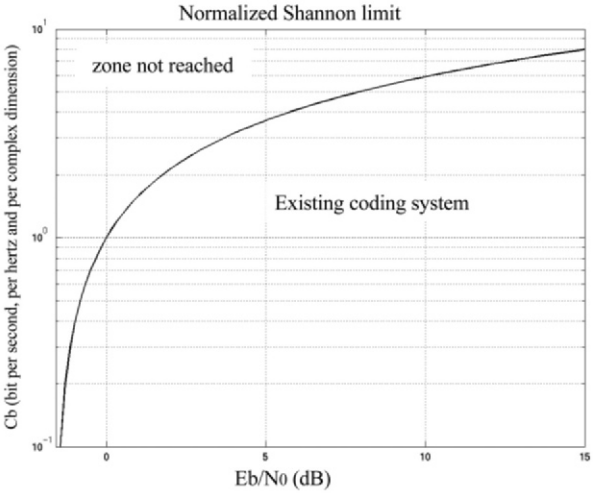


Figure 3.2 – Normalized Shannon limit.

discrete values, M being the modulation order, and have dimension N, that is $\mathbf{x}_i = [x_{i1}, x_{i2}, \cdots, x_{iN}]$. The transition probability of the Gaussian channel is:

$$p\left(\mathbf{y}\left|\mathbf{x}_i\right.\right) = \prod_{n=1}^{N} p\left(y_n\left|x_{in}\right.\right) = \prod_{n=1}^{N} \frac{1}{\sqrt{\pi N_0}} \exp\left(\frac{-\left(y_n - x_{in}\right)^2}{N_0}\right)$$

and we assume inputs taking the M different possible values equiprobably. Denoting $\mathbf{d}_{ij} = (\mathbf{x}_i - \mathbf{x}_j)/\sqrt{N_0}$ the vector of dimension N relative to the distance between the symbols $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\mathbf{t}$ an integration vector of dimension N, we obtain a simplified expression of (3.14), representing the capacity of a discrete input Gaussian channel, for any type of modulation:

$$\begin{aligned} C &= \log_2(M) \\ &- \frac{(\sqrt{\pi})^{-N}}{M} \sum_{i=1}^{M} \underbrace{\int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty}}_{N\,\text{times}} \exp\left(-\left|\mathbf{t}\right|^2\right) \log_2\left[\sum_{j=1}^{M} \exp\left(-2\mathbf{t}\mathbf{d}_{ij} - \left|\mathbf{d}_{ij}\right|^2\right)\right] \mathrm{d}\mathbf{t} \end{aligned} \tag{3.16}$$

C being expressed in bit/symbol. We note that $\mathbf{d}_{ij}$ increases when the signal to noise ratio increases (N_0 decreases) and the capacity tends towards $\log_2(M)$. The different possible modulations only appear in the expression of $\mathbf{d}_{ij}$. The discrete sums from 1 to M represent the possible discrete inputs. For the final calculation, we express $\mathbf{d}_{ij}$ as a function of E_s/N_0 according to the modulation, E_s being the energy per symbol, and the capacity of the channel can be determined using a computer. Figure 3.3 gives the result of the calculation for some PSK and QAM modulations.

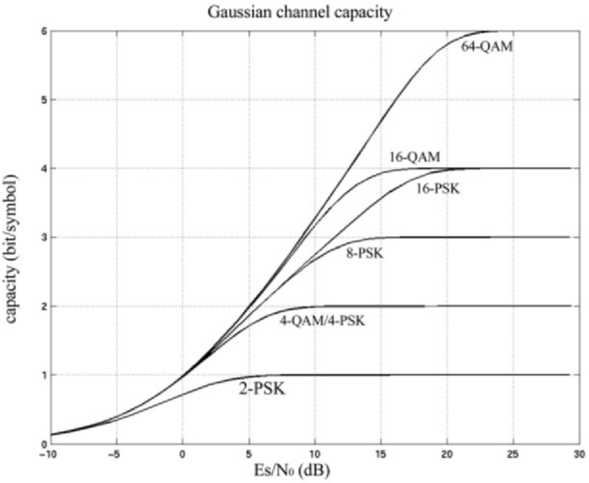


Figure 3.3 – Capacity of some modulations.

Capacity of the Rayleigh channel

Let there be a Rayleigh channel whose attenuation is denoted α. For discrete equiprobable inputs (a case similar to the Gaussian channel treated above), (3.14) is always applicable. There are two cases, conditioned by the knowledge of the attenuation α of the channel, or not.

In the case where α is not known *a priori*, we write the conditional probability density of the Rayleigh channel in the form:

$$\begin{aligned} p(\mathbf{y}\,|\mathbf{x}_i\,) &= \int\limits_{-\infty}^{+\infty} p(\alpha)p(\mathbf{y}\,|\mathbf{x}_i,\alpha\,)\mathrm{d}\alpha \\ &= \int\limits_{0}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}} 2\alpha\exp\left(-\alpha^2\right)\exp\left(-\frac{|\mathbf{y}-\alpha\mathbf{x}_i|^2}{2\sigma^2}\right)\mathrm{d}\alpha \end{aligned}$$

One development of this expression means that we can explicitly write this conditional probability density that turns out to be independent of α:

$$\begin{aligned} p(\mathbf{y}\,|\mathbf{x}_i\,) &= \sqrt{\frac{2}{\pi}}\frac{\sigma e^{-\frac{|\mathbf{y}|^2}{2\sigma^2}}}{|\mathbf{x}_i|^2+2\sigma^2} + 2\frac{\mathbf{x}_i\mathbf{y}e^{-\frac{|\mathbf{y}|^2}{|\mathbf{x}_i|^2+2\sigma^2}}}{\left(|\mathbf{x}_i|^2+2\sigma^2\right)^{3/2}} \\ &\quad\times\left[1-\frac{1}{2}\mathrm{erfc}\left(\frac{\mathbf{x}_i\mathbf{y}}{\sigma\sqrt{2\left(|\mathbf{x}_i|^2+2\sigma^2\right)}}\right)\right] \end{aligned}$$

which is sufficient to enable the capacity to be evaluated by using (3.14).
In the case where the attenuation is known, the probability density for a particular realization of α can be written:

$$p\left(\mathbf{y}\left|\mathbf{x}_i\right.,\alpha\right)=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\left|\mathbf{y}-\alpha\mathbf{x}_i\right|^2}{2\sigma^2}\right)$$

The instantaneous capacity C_α for this particular realization of α is first calculated and then we have to average C_α over the set of realizations of α in order to obtain the capacity of the channel:

$$C_\alpha=\frac{1}{M}\sum_{i=1}^{M}\int_{-\infty}^{+\infty}p\left(\mathbf{y}\left|\mathbf{x}_i,\alpha\right.\right)\log_2\left(\frac{p\left(\mathbf{y}\left|\mathbf{x}_i,\alpha\right.\right)}{p\left(\mathbf{y}\left|\alpha\right.\right)}\right)\mathrm{d}\mathbf{y}$$

$$C=\int_{0}^{+\infty}C_\alpha p\left(\alpha\right)\mathrm{d}\alpha=E\left[C_\alpha\right]$$

3.3 Practical limits to performance

In the sections above, we obtained the theoretical limits for performance which are subject to certain hypotheses that are not realistic in practice, in particular the transmission of infinite length data blocks. In the great majority of communication systems today, it is a sequence of data blocks that is transmitted, these blocks being of very variable size depending on the system implemented. Logically, limited size block transmission leads to a loss of performance compared to infinite size block transmission, because the quantity of redundant information contained in the codewords is lower.

Another parameter used to specify the performance of real transmission systems is the packet error rate (PER), which corresponds to the proportion of blocks of wrong data (containing at least one binary error after decoding).

What follows contains some results on the Gaussian channel, for two cases: the binary input and continuous output Gaussian channel, and the continuous input and output Gaussian channel. The case of the continuous input can be assimilated to that of a modulation with an infinite number of states M. The fewer states we have to describe the input, the less efficient the communication. Consequently, a binary input channel gives a lower bound on the practical performance of the set of modulations, whereas a continuous input channel gives its higher limit.

3.3.1 Gaussian binary input channel

Initial work on this channel was done by Gallager [3.2]. We denote again $p(\mathbf{y}|\mathbf{x})$ the probability of transition on the channel, and we consider information messages of size k. Assuming that a message, chosen arbitrarily and equiprobably among 2^k, is encoded then transmitted through the channel, and assuming that we use maximum likelihood decoding, then the coding theorem provides a bound on the mean error probability of decoding the correct codeword. In [3.2], it is shown that it is possible to limit the PER in the following way, for whatever value of variable ρ, $0 \leq \rho \leq 1$:

$$\mathrm{PER} \leq (2^k - 1)^\rho \sum_{\mathbf{y}} \left[\sum_{\mathbf{x}} \Pr(\mathbf{x}) p(\mathbf{y}|\mathbf{x})^{1/1+\rho} \right]^{1+\rho} \tag{3.17}$$

In the case of a channel with equiprobable binary inputs, the probability of each of the inputs is 1/2 and the vectors $\mathbf{x}$ and $\mathbf{y}$ can be treated independently in x and y scalar values. Considering that (3.17) is valid for any ρ, in order to obtain the closest upper bound to the PER, we must minimize the right-hand side of (3.17) as a function of ρ. Introducing the rate $R = k/n$, it therefore means minimizing for $0 \leq \rho \leq 1$, the expression :

$$\left\{ 2^{\rho R} \int_{-\infty}^{+\infty} \frac{1}{2} \left(\frac{1}{\sigma\sqrt{2\pi}} \right)^{\frac{1}{1+\rho}} \times \left[\exp\left(-\frac{(y-1)^2}{2\sigma^2(1+\rho)} \right) + \exp\left(-\frac{(y+1)^2}{2\sigma^2(1+\rho)} \right) \right] \mathrm{d}y \right\}^k$$

The explicit value of σ is known for binary inputs (2-PSK and 4-PSK modulations): $\sigma = (2RE_b/N_0)^{-1/2}$. An exploitable expression of Gallager's upper bound on the PER of a binary input channel is then:

$$e^{-k\frac{E_b}{N_0}} \min_{0 \leqslant \rho \leqslant 1} \left\{ \int_0^{+\infty} 2^{\rho R+1} \frac{\exp^{-y^2}}{\sqrt{\pi}} \left(\cosh\left(\frac{y\sqrt{4RE_b/N_0}}{1+\rho} \right) \right)^{1+\rho} \mathrm{d}y \right\}^k \tag{3.18}$$

This expression links the PER, the rate, the size k of the messages and the signal to noise ratio E_b/N_0, for a Gaussian binary input channel. It gives an upper bound of the PER and not an equality. This equation is not very well adapted to all cases. In particular, simulations show that for a rate close to 1, the bound is far too lax and does not give really useful results.

If we want to determine the penalty associated with a given packet size, we can compare the result obtained by evaluating (3.18) with the result obtained by computing the capacity that considers infinite size packets

3.3.2 Gaussian continuous input channel

In the case of a continuous input channel, we consider the case contrasting with that of the binary input channel, that is, we will obtain an upper bound on the practical limits of performance (all the modulations show performance lower than a continuous input channel). Any modulation used will give performance lower bounded by the limit obtained by a binary input and upper bounded by a continuous input.

The first results were given by Shannon [3.4] and by the so-called *sphere-packing bound* method which provides a lower bound on the error probability of random codes on a Gaussian channel. We again assume maximum likelihood decoding. A codeword of length n is a sequence of n whole numbers. Geometrically, this codeword can be assimilated to a point in an n-dimensional Euclidean space and the noise can be seen as a displacement of this point towards a neighbouring point following a Gaussian distribution (see Section 3.1.4). Denoting P the power of the emitted signal, all the codewords are situated on the surface of a sphere of radius $\sqrt{nP}$.

Observing that we have a code with 2^k points (codewords), each at a distance $\sqrt{nP}$ from the origin in n-dimensional space, any two points are equidistant from the origin, and consequently, the bisector of these two points (a hyperplane of dimension $n-1$) passes through the origin. Considering the set of 2^k points making up the code, all the hyperplanes pass through the origin and form pyramids with the origin as the summit. The error probability, after decoding, is $\Pr(e) = \frac{1}{2^k}\sum\limits_{i=1}^{2^k}\Pr(e_i)$, where $\Pr(e_i)$ is the probability that the point associated with the codeword i is moved by the noise outside the corresponding pyramid.

The principle of Shannon's sphere-packing bound involves this geometrical vision of coding. However, it is very complex to keep the 'pyramid' approach and the solid angle pyramid Ω_i, around the codeword i, is replaced by a cone with the same summit and the same solid angle Ω_i (Figure 3.4).

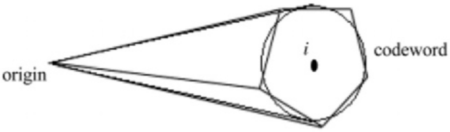


Figure 3.4 – Assimilation of a pyramid with one cone in Shannon's so-called sphere-packing approach.

It can be shown that the probability that the signal remains in the cone is higher than the probability that it remains in the same solid angle pyramid. Consequently, the error probability can be lower-bounded in the following way:

$$\Pr(e) \geq \frac{1}{2^k}\sum_{i=1}^{2^k} Q^*(\Omega_i) \tag{3.19}$$

denoting $Q^*(\Omega_i)$ the probability that the noise moves point i out of the solid angle cone Ω_i (therefore a decoding error is made on this point). We also observe that, if we consider the set of codewords equally distributed on the surface of the sphere of radius $\sqrt{nP}$, the decoding pyramids form a partition of this same sphere, and therefore the solid angle of this sphere Ω_0 is the sum of all the solid angles of the Ω_i pyramids. We can thus replace the solid angles Ω_i by the mean solid angle $\Omega_0/2^k$.

This progression, which leads to a lower bound on the error probability for an optimal decoding of random codes on the Gaussian channel, is called the sphere-packing bound because it involves restricting the coding to an n-dimensional sphere and the effects of the noise to movements on this sphere.

Mathematical simplifications give an exploitable form of the lower bound on the packet error rate (PER):

$$\begin{aligned} &\ln(\mathrm{PER}) \geq \tfrac{k}{R}\left[\ln\left(G(\theta_i, A)\sin\theta_i\right) - \tfrac{1}{2}\left(A^2 - AG(\theta_i, A)\cos\theta_i\right)\right] \\ &\theta_i \approx \arcsin\left(2^{-R}\right) \\ &G(\theta_i, A) \approx \left(A\cos\theta_i + \sqrt{A^2\cos^2\theta_i + 4}\right)/2 \\ &A = \sqrt{2RE_b/N_0} \end{aligned} \tag{3.20}$$

These expressions link the size k of the messages, the signal to noise ratio E_b/N_0 and the coding rate R. For high values of R and for block sizes k lower than a few tens of bits, the lower bound is very far from the real PER.

Asymptotically, for block sizes tending towards infinity, the bound obtained by (3.20) tends towards the Shannon limit for a continuous input and output channel such as presented in Section 3.2. In the same way as for the binary input channel, if we wish to quantify the loss caused by the transmission of finite length packets, we must normalize the values obtained by evaluating (3.20) by removing the Shannon limit (3.15) from them, the penalty having to be null when the packet sizes tend towards infinity. The losses due to the transmission of finite length packets in comparison with the transmission of a continuous flow of data are less in the case of a continuous input channel than in the case of a binary input channel.

3.3.3 Some examples of limits

Figure 3.5 below gives an example of penalties caused by the transmission of blocks of size k lower than 10000 bits, in the case of continuous input and in the case of binary input. These penalty values should be combined with the values of capacities presented in Figure 3.3, in order to obtain the absolute limits. As we have already mentioned, this figure is to be considered with caution for small values of k and high PER.

The results obtained concern the Gaussian channel. It is theoretically possible to consider the case of fading channels (Rayleigh, for example) but the computations become complicated and the results very approximate.

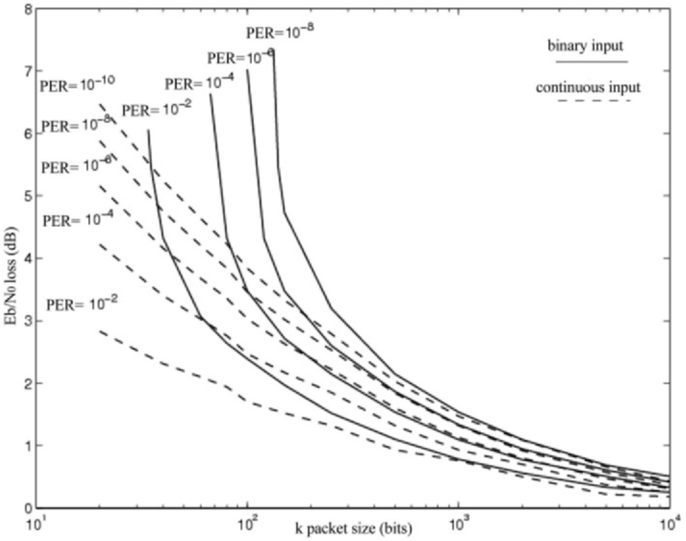


Figure 3.5 – Penalties in $Eb/N0$ for the transmission of finite length packets for the continuous input channel and the binary input channel as a function of size k (information bits), for a coding rate 5/6 and different PER.

3.4 Minimum distances required

So far, we have highlighted the theoretical limits and they have been calculated for the Gaussian channel. These limits determine boundaries, expressed in signal to noise ratio, between transmission channels at the output of which it is possible to correct the errors and channels for which this correction cannot be envisaged. Assuming that codes exist whose decoding can be performed close to these limits, the question now arises about how we can know which minimum Hamming distances (MHD) these codes should have in order to satisfy a given objective of error rates.

Here we present some results for the Gaussian channel and modulations currently used: 4-PSK, 8-PSK and 16-QAM.

3.4.1 MHD required with 4-PSK modulation

With maximum likelihood decoding after transmission on a Gaussian channel, the PER has a known upper bound, called the union bound:

$$\text{PER} \leq \frac{1}{2} \sum_{d \geq d_{\min}} N\left(d\right) \operatorname{erfc}\left(\sqrt{dR\frac{E_b}{N_0}}\right) \tag{3.21}$$

where erfc(x) denotes the complementary error function defined by $\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int\limits_{x}^{\infty} \exp\left(-t^2\right)\mathrm{d}t$. $d_{\min}$ is the minimum Hamming distance of the code associated with the modulation considered, 2-PSK or 4-PSK in the present case. $N(d)$ represents the multiplicities of the code (see Section 1.5). In certain cases, these multiplicities can be determined precisely (like for example simple convolutional codes, Reed-Solomon codes, BCH codes, etc. ...), and (3.21) can easily be evaluated. For other codes, in particular turbo codes, it is not possible to determine these multiplicities easily and we have to consider some realistic hypotheses in order to get round the problem. The hypotheses that we adopt for turbo codes and for LDPC codes are the following [3.1]:

- *Hypothesis 1: Uniformity.* There exists at least one codeword of weight[1] $d_{\min}$ having an information bit d_i equal to "1", for any place i of the systematic part $(1 \leq i \leq k)$.
- *Hypothesis 2: Unicity.* There is only one codeword of weight $d_{\min}$ such that $d_i =$"1".
- *Hypothesis 3: Non-overlapping.* The k codewords of weight $d_{\min}$ associated with the k bits of information are distinct.

Using these hypotheses and limiting ourselves to the first term of the sum in (3.21), the upper bound becomes an asymptotic approximation (low PERs):

$$\mathrm{PER} \approx \frac{k}{2}\operatorname{erfc}\left(\sqrt{d_{\min} R \frac{E_b}{N_0}}\right) \tag{3.22}$$

The three hypotheses, taken separately, are more or less realistic. Hypotheses 1 and 3 are somewhat pessimistic as to the quantity of codewords at the minimum distance. As for hypothesis 2, it is slightly optimistic. The three hypotheses together are suitable for an acceptable approximation of the multiplicity, especially since imprecision about the value of this multiplicity does not affect the quality of the final result. Indeed, the targeted minimum distance that we wish to determine from (3.22) appears in an exponential argument, whereas the multiplicity is a multiplying coefficient.

It is then possible to combine (3.22) with the results obtained in Section 3.3 which provide the signal to noise ratio limits. Giving E_b/N_0 the limit value beyond which using a code is not worthwhile, we can extract from (3.22) the MHD sufficient to reach a PER at that limit value. Given, on the one hand, that (3.22) assumes ideal (maximum likelihood) decoding and, on the other hand, that the theoretical limit is not reached in practice, the targeted MHD can be slightly lower than the result of this extraction.

Figure 3.6 presents some results obtained using this method.

[1] The codes being linear, distance and weight have the same meaning.

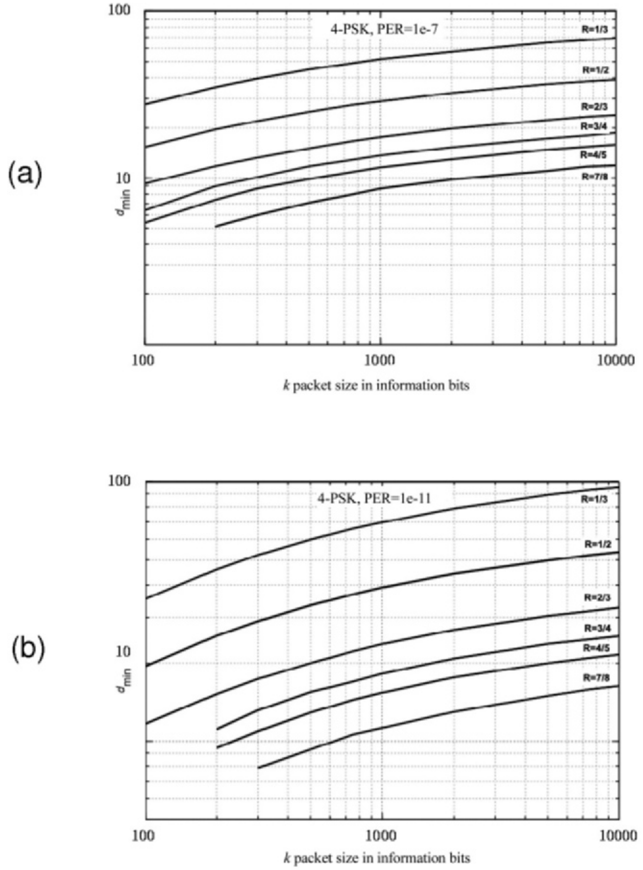


Figure 3.6 – Minimum distances required for 4-PSK modulation and a Gaussian channel as a function of packet size, for some coding rates and $PER = 10^{-7}$ and $PER = 10^{-11}$.

3.4.2 MHD required with 8-PSK modulation

Here we consider an 8-PSK modulation on a Gaussian channel implemented using the principle of the "pragmatic" approach, as presented in Figure 3.7. This approach first involves encoding the data flow in packets to produce codewords that are then randomly permuted by the interleaver Π, with a permutation law drawn randomly. The contents of the permuted codewords are then organized in groups of 3 bits using a Gray coding, before being modulated in 8-PSK. The demodulator provides the received symbols from which we extract the log likelihood ratios (LLRs) for all the bits of the packets. Finally, inverse interleaving and decoding complete the transmission chain.

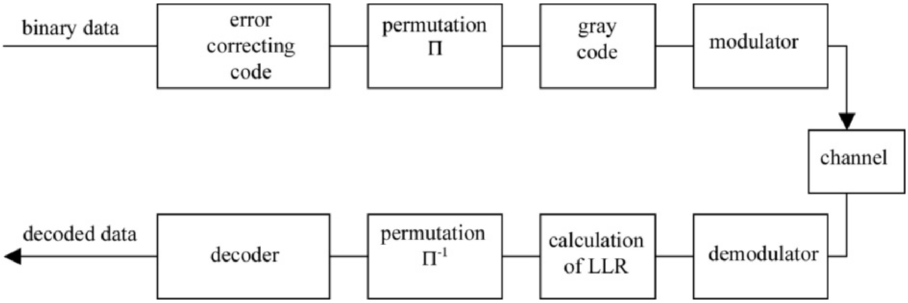


Figure 3.7 – Transmission scheme using the pragmatic approach.

The error probability P_e is the probability of deciding about an incorrect codeword instead of the codeword emitted. Let N_s be the number of modulated symbols that differ between the incorrectly decoded codeword and the codeword emitted. Also let $\{\phi_i\}$ and $\{\phi_i'\}$ $(1 \leq i \leq N_s)$ be the transmitted phase sequences for these symbols that differ. It is possible to express P_e as a function of these phases and of the signal to noise ratio:

$$P_e = \frac{1}{2}\text{erfc}\sqrt{\frac{E_s}{N_0}\left[\sum_{i=1,N_s}\sin^2\left(\frac{\varphi_i' - \varphi_i}{2}\right)\right]} \tag{3.23}$$

where E_s is the energy per symbol emitted and N_0 the monolateral noise power spectral density. It is however not possible to exploit (3.23) in the general case. We require an additional hypothesis, which is then added to the three hypotheses formulated in the previous section, and assume that N_S is much lower than the size of the interleaved codewords:

- *Hypothesis 4:* A symbol does not contain more than one opposite bit in the correct codeword and in the wrong codeword.

This hypothesis allows the following probabilities to be expressed:

$$\Pr\{\varphi_i - \varphi_i' = \pi/4\} = 2/3; \quad \Pr\{\varphi_i - \varphi_i' = 3\pi/4\} = 1/3$$

which means that two times out of three on average, the Euclidean distance between the concurrent symbols is $2\sqrt{\frac{E_s}{T}}\sin(\pi/8)$ and, one time out of three, is raised to $2\sqrt{\frac{E_s}{T}}\sin(3\pi/8)$ (Figure 3.8).

Considering the asymptotic case, that is, putting $N_s = d_{\min}$, yields:

$$\text{PER}_{\text{8-PSK},\Pi\,\text{random}} \approx \\ k\left(\tfrac{2}{3}\right)^{d_{\min}}\sum_{j=0}^{d_{\min}}\binom{d_{\min}}{j}\left(\tfrac{1}{2}\right)^{j+1}\text{erfc}\sqrt{\tfrac{E_s}{N_0}\left[j\sin^2\tfrac{3\pi}{8} + (d_{\min}-j)\sin^2\tfrac{\pi}{8}\right]} \tag{3.24}$$

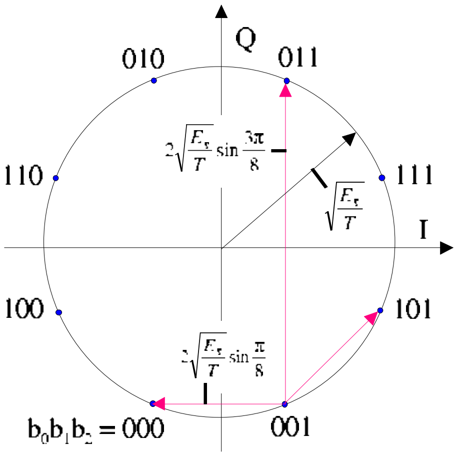

Q
010
011
$2\sqrt{\frac{E_s}{T}}\sin\frac{3\pi}{8}$
$\sqrt{\frac{E_s}{T}}$
110
111
I
100
101
$2\sqrt{\frac{E_s}{T}}\sin\frac{\pi}{8}$
$b_0b_1b_2 = 000$
001

Figure 3.8 – 8-PSK constellation with Gray coding. E_s and T are the energy and the duration of a symbol, respectively.

This relation therefore makes it possible to establish a relation between the signal to noise ratio, the size of the information blocks and the PER. In the same way as in Section 3.4, we can combine this result with the limits on the signal to noise ratio to obtain the MHD targeted for a 8-PSK coded modulation using the pragmatic approach. Figure 3.9 presents some results obtained with this method.

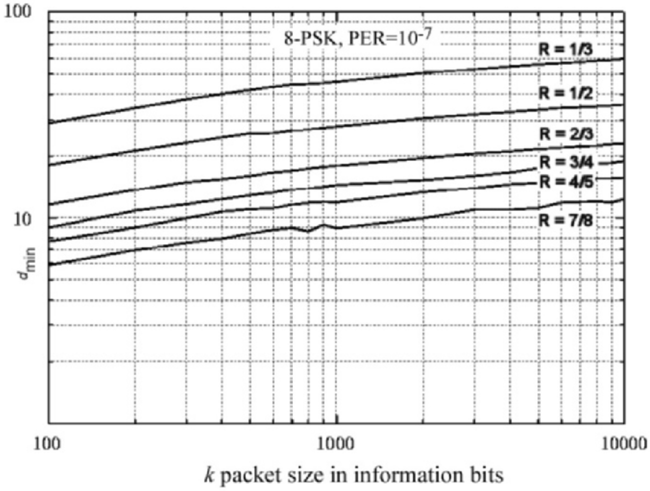
3.4.3 MHD required with 16-QAM modulation

The same method as above, based on the same four hypotheses, can be applied to the case of 16-QAM modulation with pragmatic encoding. The constellation is a standard 16-state Gray constellation. For 75% of the bits making up the symbols, the minimum Euclidean distance is $\sqrt{2E_s/5}$ and for the remaining 25%, this distance is $3\sqrt{2E_s/5}$. Estimating the PER gives:

$$\text{PER} \approx k \left(\frac{3}{4}\right)^{d_{\min}} \sum_{j=0}^{d_{\min}} \binom{d_{\min}}{j} \left(\frac{1}{3}\right)^{j} \frac{1}{2}\text{erfc}\sqrt{(8j + d_{\min})\,\frac{E_s}{10N_0}} \qquad (3.25)$$

Like for 4-PSK and 8-PSK modulations, this relation used jointly with signal to noise ratio limits makes it possible to obtain targeted MHD values for 16-QAM modulation (Figure 3.10).

Some observations can be made from the results obtained in Section 3.4. For example, in the particular case of 4-PSK modulation, for a rate $R = 1/2$, size $k = 4000$ bits and PER of 10^{-11}, Figure 3.6 provides a targeted MHD of 50. From the evaluation that we can make from the Gilbert-Varshamov bound



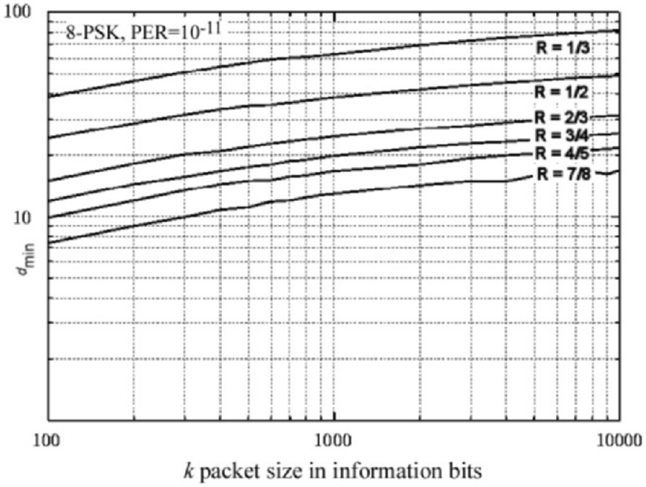


Figure 3.9 – Minimum distances required for 8-PSK modulation and a Gaussian channel as a function of packet size, for some coding rates and $PER = 10^{-7}$ and $PER = 10^{-11}$.

(relation (3.9)), random codes have a minimum distance of about 1000. There is therefore a great difference between what ideal (random) coding can offer and what we really need.

A second aspect concerns the dependency of the required MHD upon the modulation used, a dependency that turns out to be minimum. Thus, a code

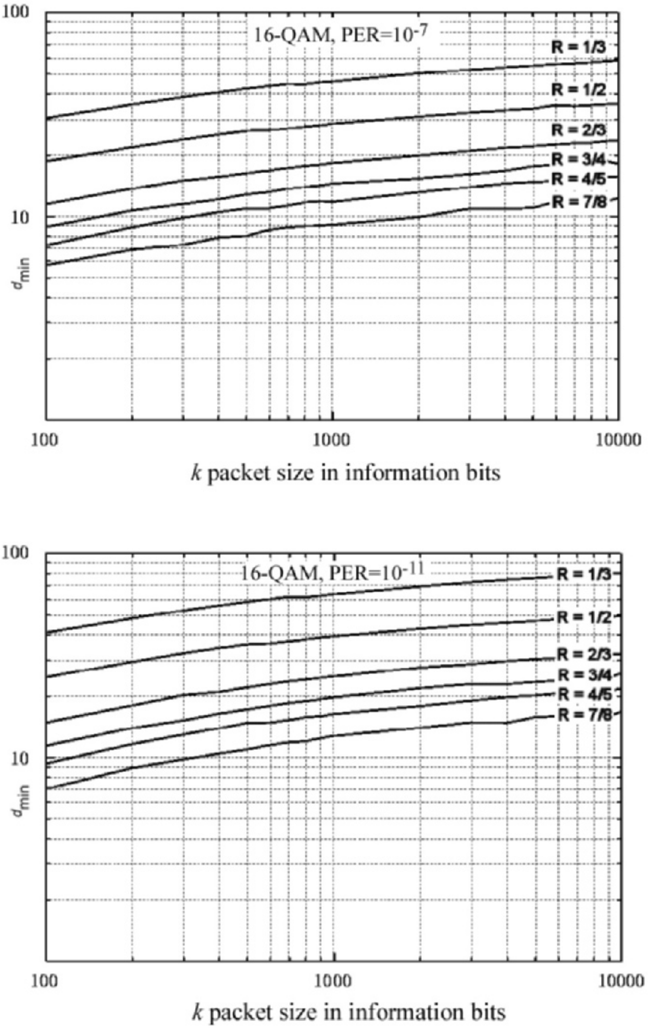


Figure 3.10 – Minimum distances required for 16-QAM modulation and a Gaussian channel as a function of the packet size, for some coding rates and $PER = 10^{-7}$ and $PER = 10^{-11}$.

having a minimum distance sufficient to reach the channel capacity with 4-PSK modulation will also satisfy specifications with the other modulations, for a certain size of message (larger than 1000 bits for $R = 1/2$) or for longer messages (over 5000 bits) if the rate is higher.

Bibliography

[3.1] C. Berrou, E. Maury, and H. Gonzalez. Which minimum hamming distance do we really need? In *Proceedings of the 3rd International Symposium on Turbo Codes & related topics (ISTC 2003)*, pages 141–148, Brest, France, Sept. 2003.

[3.2] R.E. Gallager. *Information Theory and Reliable Communications*. John Wiley & Sons, 1968.

[3.3] C.E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27, July-Oct. 1948.

[3.4] C.E. Shannon. Probability of error for optimal codes in a gaussian channel. *Bell Systems Technical Journal*, 38, 1959.