# Chapter 1

# Preliminary Concepts

## 1.1 Introduction

In this chapter, we will present some preliminary concepts which will be extensively used throughout this book. In Section 1.2, we shortly discuss the modeling process of a (high performance) communication system. In Section 1.3, we focus on the concept of modulation and its role in communication schemes. In Section 1.4, some introductory material on error correcting codes is presented, with emphasis on *linear* error correcting codes. Section 1.5 is a self-contained introduction to information theory, which should provide the reader not familiar with this field with all the basic tools needed to understand this book. In Section 1.6, a short overview of the following chapters is given.

## 1.2 Modeling a Communication System

Modern point-to-point communication theory is based on a scientific approach to the task of transmitting information from an information source to a remote—either in terms of space or time—destination.

Common to all scientific approaches are the following steps:

- observation of the nature;
- mathematical modeling of the observed phenomena;
- model validation.

Engineering, i.e., designing a communication system, adds a fourth step:

- derivation, on the basis of a mathematical model, of a practical solution to accomplish a given task.

M. Franceschini, G. Ferrari and R. Raheli, *LDPC Coded Modulations,*
DOI: 10.1007/978-3-540-69457-1_1, © Springer-Verlag Berlin Heidelberg 2009

The process of modeling is, therefore, a fundamental and central step in the analysis and design of advanced communication systems. In particular, whenever the design target is to obtain a system performance close to the theoretical (ultimate) limits, the quality of the model has a direct impact on the obtainable performance. This can be rigorously quantified with information-theoretic tools [1, 2].

In this book, we will focus on the transmission of a digital stream of data, i.e., a discrete-time digital process $\mathcal{A}$ consisting of a sequence of symbols $\{a_k\}$ belonging to a discrete set of possible values $\mathscr{A}$. At the output of the receiver, a corresponding sequence of detected symbols $\hat{\mathcal{A}} = \{\hat{a}_k\}$, $\hat{a}_k \in \mathscr{A}$ is found. The quality of the transmission scheme can be determined by analyzing the difference between the sequence $\mathcal{A}$ and the sequence $\hat{\mathcal{A}}$ and may be expressed through several distinct parameters. Under the assumption of binary symbols, the most used indicator of the transmission quality is the bit error probability, namely, the probability of error for a generic bit in the digital stream, i.e.,

$$P_{\mathrm{e}} \triangleq P\{a_k \neq \hat{a}_k\} .$$

In contexts where the probability $P\{a_k \neq \hat{a}_k\}$ depends on the epoch $k$, an average error probability can be defined as

$$P_{\mathrm{e}} = \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} P\{a_k \neq \hat{a}_k\} . \tag{1.1}$$

Depending on the system structure and other assumptions, other quality indicators may be useful, such as the symbol error probability, if the communication system involves the transmission of non-binary symbols, or the frame error probability, if the data stream is partitioned into frames. The error probability is a measure of the rate of occurrence of decision errors over a sufficiently large number of transmission acts according to the interpretation of the concept of probability as the relative frequency of error occurrences. The error rate is a practical performance indicator which is suitable to be measured. In particular the bit error rate (BER), symbol error rate (SER) and frame error rate (FER), represent the rates corresponding to the relevant probabilities. In the following, for brevity, we will not distinguish between error probability and error rate and we will use the two notions interchangably.

Information theory suggests to accomplish reliable transmission by mapping the process $\mathcal{A}$ into a process $\mathcal{C}$, which might be discrete-time or continuous-time, analog or digital, whose statistical properties are suitable for the particular communication channel. This operation is referred to as *channel coding and modulation.* In Figure 1.1, a general point-to-point communication system

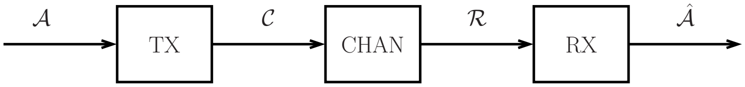

Figure 1.1: Point-to-point communication system model.

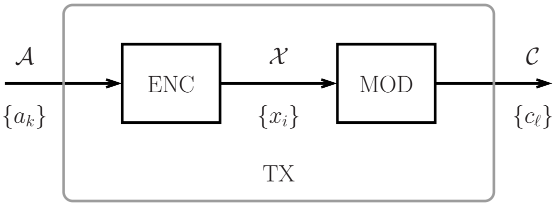

Figure 1.2: Schematic diagram of the transmitter (TX) which performs error correcting coding followed by modulation.

model is shown. Note the presence of the transmitter (TX), whose task is the construction of the signal $\mathcal{C}$ to be transmitted through the channel (CHAN). The receiver (RX) performs detection of the transmitted data sequence based on the received process $\mathcal{R}$. A short introduction to information theory will be given in Section 1.5. For more details, we refer the interested reader to [1, 3], which provide a thorough treatment of this subject.

From a practical point of view, in most applications the process $\mathcal{C}$ may be effectively represented by a discrete-time sequence $\{c_\ell\}$. The operation leading from $\mathcal{A}$ to $\mathcal{C}$ is usually split into two separate actions:

- the first consists in performing discrete algebraic operations on $\mathcal{A}$ which result in a new discrete-time digital process $\mathcal{X}$;

- the second consists in mapping the corresponding sequence $\{x_i\}$ into the final sequence $\{c_\ell\}$, to be transmitted through the considered channel.

The first operation corresponds to the application of an error correction encoder to the digital sequence to be transmitted, whereas the second operation, which maps the digital sequence into a signal suitable for transmission over the channel, is usually referred to as modulation. An illustrative scheme of the transmitter performing error correction encoding followed by modulation is shown in Figure 1.2. Note that the sequences $\{a_k\}$, $\{x_i\}$, and $\{c_\ell\}$ have different index variables $k$, $i$ and $\ell$, to highlight the fact that they might not have the same rate and that, on average, a symbol in one of the three sequences does not necessarily correspond to a single symbol in the other sequences.

## 1.3 Modulation

As stated above, the term *modulation* denotes an operation that maps a sequence of digital symbols, which may correspond to the data to be transmitted or an encoded version of such data, to a sequence of samples or a waveform suitable for transmission through the considered communication channel.

Modulation is a mathematical model of the physical block, found in virtually every communication system, devoted to transform the digital sequence into the transmitted signal, which is often electrical, but may as well be optical, acoustic, etc.

We assume that all the signals in the communication system can be given a discrete-time representation. Modulators are processing blocks and may or may not have memory, meaning that their current output sample may be a function of only the last input sample or more input samples, respectively. Typical memoryless modulations for baseband transmission, which implies mapping the digital sequence into real samples, belonging to a finite set $\mathscr{C}$, referred to as the modulation constellation, include:

- on-off keying (OOK) or, in general, amplitude-shift keying (ASK);
- pulse amplitude modulation (PAM).

A memoryless modulator for bandpass transmission maps the digital sequence into complex samples. Typical complex constellations include:

- quadrature amplitude modulation (QAM);
- phase shift keying (PSK);
- amplitude and phase shift keying (APSK).

For a survey of the principal modulation formats, their constellations, and their properties, the interested reader is referred to [4]. Examples of modulations with memory include:

- differential PSK (DPSK);
- trellis-coded modulation (TCM);
- continuous phase modulation (CPM).

The introduction of memory into the modulator provides potentially infinite degrees of freedom. In fact, any combination of a processing block with a given modulator may be interpreted as a modulator with memory. In order to take advantage of the separation between coding, i.e., the operation leading from $\mathcal{A}$ to $\mathcal{X}$, and modulation, i.e., the operation leading from $\mathcal{X}$ to $\mathcal{C}$, the complexity of the modulation process should be kept low. Nonetheless, it is possible to consider, as part of the tasks carried out by a modulator with memory, operations which usually are considered separately, such as, for example

- pilot symbols insertion;
- a line coding block for spectral shaping.

The separation of coding and modulation is the starting point for the derivation of low-density parity-check (LDPC)-coded modulations. In particular, it allows to separately focus on modulation, whose role is to provide an effective “interface” to the channel, and coding, whose role is to improve the efficiency of information transmission. This will be investigated in depth in Chapter 5.

## 1.4 Error Correcting Codes

Error correcting codes (ECC) are functions which map a sequence $\mathcal{A}$ of discrete values, i.e., the data sequence, into a (usually longer) sequence $\mathcal{X}$ of discrete values, i.e., the code sequence. As their name suggests, ECC admit inverse functions capable to recover the data sequence $\mathcal{A}$, regardless of possible transmission errors in the code sequence $\mathcal{X}$, provided that the amount of erroneous symbols does not exceed the error correction capability.

Typically, the encoding of a data stream can be performed in two different ways: (i) block coding and (ii) stream coding. The first one consists in partitioning the data sequence into blocks of length $K$ and applying to each block a coding function returning blocks, i.e., the *codewords*, of length $N$. The obtained codewords are joined to form the code sequence. In stream coding, the data sequence is fed to a finite state machine (FSM), which outputs one or more code symbols for each input data symbol.

### 1.4.1 Block Codes

In general, a block code $C$ is a vector function associating a vector of $K$ elements, belonging to a set $\mathscr{A}$, to a vector of $N$ elements belonging to a set $\mathscr{X}$. Assuming that the function is injective, the cardinality of the set of all codewords is equal to $[\mathrm{Card}(\mathscr{A})]^K$, where $\mathrm{Card}(\mathscr{A})$ is the cardinality of $\mathscr{A}$. The code rate $R_C$, expressed in bits per output symbol, is defined as

$$R_C = \frac{K \log_2 \mathrm{Card}(\mathscr{A})}{N} \, .$$

A linear block code over a $q$-ary Galois finite field—denoted as GF$(q)$—is a vector linear function which associates a $K$-dimensional vector $\boldsymbol{a}$ of elements in GF$(q)$, i.e., the data to be encoded, to a $N$-dimensional vector $\boldsymbol{x}$ of elements in GF$(q)$, i.e., the codeword, according to the following rule:

$$\boldsymbol{x} = G\boldsymbol{a}$$

where $G$ is a matrix in GF$(q)$ referred to as code generation matrix [5, 6]. It can be shown that the set of all codewords can be characterized by the so-called parity check matrix, usually denoted with $H$, which can be obtained by proper manipulation of $G$. A vector $\boldsymbol{x}$ is a codeword if and only if

$$H\boldsymbol{x} = 0$$

i.e., the set of all codewords is defined as the null space of the parity check matrix $H$ [5, 6]. In this book, only binary codes will be considered; hence, in all cases data and codeword elements will be in GF(2), i.e., the Boolean algebra.

Several important code families belong to the set of linear block codes, among which:

- cyclic codes, comprising Bose-Chaudhuri-Hocquenghem (BCH) codes, Reed-Solomon codes, Hamming codes [6];
- turbo codes [7, 8];
- LDPC codes [9–13].

Several techniques exist for decoding linear block codes. In particular, two approaches are possible: (i) hard-output decoding, which outputs estimates of the codewords or the data symbols, and (ii) soft-output decoding, which outputs the *likelihood* of each codeword or data symbol, i.e., an estimate of how much a codeword or a data symbol is likely to assume each possible value. In this book, we are interested in soft-output decoding only, since, as it will be clear in the next chapters, this leads to the construction of complex receivers based on simpler soft-input soft-output (SISO) detection/decoding blocks.

### 1.4.2 Stream Codes

Stream coding is an error correcting coding technique which applies to streams of data as opposed to blocks of data. The codeword length is unbounded, so that a stream encoder typically operates on a symbol-by-symbol basis, outputting an encoded symbol for each data symbol at its input. Practical stream encoders are implemented by means of FSMs. In particular, an FSM is defined by two functions: (i) the *output function* and (ii) the *next state function*. These two functions are mathematically described as follows:

$$c_k = f_{\mathrm{o}}(a_k, s_k) \tag{1.2}$$

$$s_{k+1} = f_{\mathrm{ns}}(a_k, s_k) \tag{1.3}$$

where $c_k$ denotes the output symbol at epoch $k$, $s_k$ denotes the state at epoch $k$, $a_k$ is the input data symbol at epoch $k$, and $f_{\mathrm{o}}(\cdot,\cdot)$ and $f_{\mathrm{ns}}(\cdot,\cdot)$ denote the output function and the next state function, respectively. The output stream is formed by the sequence $\{c_k\}$. Without loss of generality, it is assumed that the code sequence has the same length of the information sequence—the redundancy introduced by the encoder is accounted for by expanding the symbol cardinality.

Linear stream coding is the most popular stream coding technique for error correction. Such codes are also known as *convolutional codes* since the output stream can be seen as the convolution of the input data stream with an “impulse response” stream. The convolution is carried out using finite field algebra.

The decoding of stream codes, like that of block codes, can be performed by means of hard-output or soft-output decoding algorithms. The most important hard decoding algorithm for stream coding is the well known Viterbi algorithm (VA) [4, 14, 15]. The VA is optimum, in the sense that it computes the most likely transmitted data sequence given the received observable sequence. Soft decoding/detection techniques for stream codes comprise the soft-output Viterbi algorithm (SOVA) [16].

In general, in stream coding it is not possible to compute exact *a posteriori* probability (APP) of the (possibly infinite) transmitted data symbols. Nevertheless, APP computation is possible, *if the stream encoder is used to encode a finite-length sequence*, by means of the famous Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm, also known as forward backward (FB) algorithm [17].

## 1.5 Information Theory Basics

In this book, only a basic set of tools drawn from classical information theory will be used. In this section, the main information-theoretic concepts will be concisely presented, to provide the unfamiliar reader with a sufficient (minimum) background. For more details, we refer the interested reader to fundamental information theory textbooks [1,3,18,19] and to the vast scientific literature referenced therein.

While studying communication problems, we are interested in the information transfer capabilities of communication systems. In order to set a mathematical framework, the first problem is to define the concept of information measure. A discrete *information source* $\mathcal{C}$ is an entity emitting a stochastic sequence of symbols $\{C_k\}$, each belonging to a finite set $\mathscr{C}$. Let us assume to describe the sequence as an *ergodic* random sequence. We define the concept of *entropy rate* $\mathcal{H}(\mathcal{C})$ to measure the expected rate of information emitted by the information source $\mathcal{C}$ as follows:

$$\mathcal{H}(\mathcal{C}) = \lim_{n\to\infty} -\frac{1}{n}\mathrm{E}\{\log p(C_1,\ldots,C_n)\} \tag{1.4}$$

where $\mathrm{E}\{\cdot\}$ denotes the expectation, $p(c_1,\ldots,c_n)$ denotes the probability that the first $n$ elements of the random sequence $\{C_k\}$ are equal to $c_1,\ldots,c_n$, and log is the logarithm to a given base. For base 2, information is measured in *bits*. If the source is memoryless, i.e., the output values are independent and identically distributed (i.i.d.), the entropy rate is usually referred to as *entropy* and its definition is the following:

$$\mathcal{H}(\mathcal{C}) = \mathsf{H}(C_k) \triangleq -\mathrm{E}\{\log p(C_k)\} \tag{1.5}$$

which, since $\{C_k\}$ are i.i.d., does not depend on $k$.

The entropy of a source can be interpreted as a measure of its unpredictability. Sources emitting samples which can be predicted with high probability are characterized by low entropy whereas sources emitting samples which can be predicted with low probability have higher entropy. The entropy rate of a discrete information source is always higher than or equal to zero. In particular, $\mathcal{H}(\mathcal{C}) = 0$ if and only if there exist a sequence $\{c_k^*\}$ such that $P\{C_k = c_k^*\} = 1$, $\forall k$.

In general, any stream operation—deterministic or stochastic—on the output of a source $\mathcal{C}$, leads to a new source $\mathcal{Y}$ with different statistical properties. The *conditional entropy rate* of $\mathcal{Y}$ given $\mathcal{C}$ is defined as follows:

$$\mathcal{H}(\mathcal{Y}|\mathcal{C}) \triangleq \lim_{n\to\infty} -\frac{1}{n}\mathrm{E}\{\log p(Y_1,\ldots,Y_n|C_1,\ldots,C_n)\} \tag{1.6}$$

and measures the "residual" entropy rate of $\mathcal{Y}$ given $\mathcal{C}$.

The *(mutual) information rate* between $\mathcal{C}$ and $\mathcal{Y}$ is defined as follows:

$$\mathsf{I}(\mathcal{C};\mathcal{Y}) = \mathcal{H}(\mathcal{Y}) - \mathcal{H}(\mathcal{Y}|\mathcal{C}) \tag{1.7}$$

and measures the average per-sample amount of information carried by $\mathcal{Y}$ regarding $\mathcal{C}$.

In the case of i.i.d. sources, the conditional entropy rate and the information rate are referred to as *conditional entropy* and *mutual information*, respectively.

Entropy and information rate can be extended to sources emitting continuously distributed samples. In this case, the entropy rate is referred to as *differential entropy rate* and is defined as follows:

$$\mathsf{h}(\mathcal{C}) \triangleq \lim_{n\to\infty} -\frac{1}{n}\mathrm{E}\{\log p(C_1,\ldots,C_n)\} \tag{1.8}$$

where $p(C_1,\ldots,C_n)$ denotes the probability density function (pdf) of the first $n$ elements of the random sequence computed using as argument the (random) vector $(C_1,\ldots,C_n)$. The differential entropy rate may take any real value, i.e., it is not limited to positive or zero values.

If $\mathcal{C}$ is a generic source (with discrete or continuously distributed alphabet), $\mathcal{Y}$ is a source emitting symbols in a continuously distributed alphabet, and $\mathcal{Y}$ and $\mathcal{C}$ are jointly ergodic, the information rate between $\mathcal{C}$ and $\mathcal{Y}$ can be defined in terms of the differential entropy rate as follows:

$$\mathsf{I}(\mathcal{C};\mathcal{Y}) = \mathsf{h}(\mathcal{Y}) - \mathsf{h}(\mathcal{Y}|\mathcal{C}) \tag{1.9}$$

or, equivalently, in terms of the entropy rate as

$$\mathsf{I}(\mathcal{C};\mathcal{Y}) = \mathcal{H}(\mathcal{C}) - \mathcal{H}(\mathcal{C}|\mathcal{Y})\,. \tag{1.10}$$

Unlike the differential entropy rate, the information rate is larger than or equal to zero, regardless of the type of source. As a consequence, if $\mathcal{C}$ in (1.10) is a discrete source, given that

$$\mathsf{I}(\mathcal{C};\mathcal{Y}) = \mathcal{H}(\mathcal{C}) - \mathcal{H}(\mathcal{C}|\mathcal{Y}) \geq 0$$

and, since the entropy rate (of a discrete source) is non-negative,

$$\mathcal{H}(\mathcal{C}|\mathcal{Y}) \geq 0$$

it follows that

$$\mathsf{I}(\mathcal{C};\mathcal{Y}) \leq \mathcal{H}(\mathcal{C})\,.$$

In other words, the information rate between a discrete source and a generic source is upper bounded by the entropy rate of the discrete source.

In [20], C. E. Shannon showed that, if the input and the output of a channel have the same joint statistical properties of two sources $\mathcal{C}$ and $\mathcal{Y}$, respectively, $\mathsf{I}(\mathcal{C};\mathcal{Y})$ is the supremum of the achievable data rates through that channel (considering the input distribution imposed by $\mathcal{C}$).

For a given channel, the corresponding *channel capacity* is a quantity defined as follows:

$$\mathsf{C} = \max_{\mathcal{C}\in\mathcal{K}} \mathsf{I}(\mathcal{C};\mathcal{Y}) \tag{1.11}$$

where the maximization is carried out over all possible input distributions, i.e., input sources, belonging to a set of sources $\mathcal{K}$ satisfying a given constraint or set of constraints. The channel capacity thus represents the supremum of the achievable information rates that can be reliably transmitted through the considered channel.

The following examples will clarify the role of the information theory quantities described above.

**Example 1.1** *Binary input AWGN channel.*

Assume to transmit a sequence of i.i.d. binary symbols $\{C_k\}$, where the random variable (RV) $C_k$ may take on values in the set $\mathscr{C} = \{1, -1\}$ and $P\{C_k = 1\} = P\{C_k = -1\} = 1/2$, through a discrete additive white Gaussian noise (AWGN) channel. We will refer to this configuration as *binary input AWGN* (BIAWGN) channel.

The output samples $\{Y_k\}$ can be expressed as:

$$Y_k = C_k + W_k \tag{1.12}$$

where $\{W_k\}$ are zero-mean i.i.d. Gaussian samples with variance $\sigma_w^2$. The signal-to-noise ratio ($\mathsf{SNR}$) can be defined as follows:

$$\mathsf{SNR} \triangleq \frac{\mathrm{E}\{|C_k|^2\}}{\mathrm{E}\{|W_k|^2\}} = \frac{1}{\sigma_w^2}\,. \tag{1.13}$$

The entropy of the input source is

$$\mathcal{H}(\mathcal{C}) = \mathsf{H}(C_k) = -\mathrm{E}\{\log P(C_k)\} = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = 1 \text{ bit}\,. \tag{1.14}$$

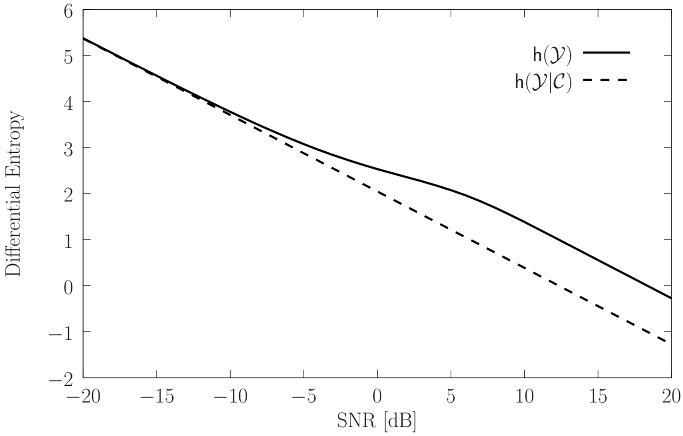

Figure 1.3: Example 1.1: differential entropy rates.

The differential entropy of the output is:

$$\begin{aligned}\mathsf{h}(\mathcal{Y}) &= -\mathrm{E}\{\log p(Y_k)\} \\ &= -\int_{-\infty}^{\infty} p(y)\log p(y)\mathrm{d}y \\ &= -\int_{-\infty}^{\infty} \frac{e^{-\frac{(y-1)^2}{2\sigma_w^2}} + e^{-\frac{(y+1)^2}{2\sigma_w^2}}}{2\sqrt{2\pi\sigma_w^2}} \log \frac{e^{-\frac{(y-1)^2}{2\sigma_w^2}} + e^{-\frac{(y+1)^2}{2\sigma_w^2}}}{2\sqrt{2\pi\sigma_w^2}} \mathrm{d}y \qquad (1.15)\end{aligned}$$

and the conditional differential entropy of the output given the input is:

$$\begin{aligned}\mathsf{h}(\mathcal{Y}|\mathcal{C}) &= \mathsf{h}(\mathcal{W}+\mathcal{C}|\mathcal{C}) \\ &= \mathsf{h}(\mathcal{W}) \\ &= -\mathrm{E}\{\log p(W)\} \\ &= -\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma_w^2}} e^{-\frac{(w)^2}{2\sigma_w^2}} \log \frac{e^{-\frac{(w)^2}{\sigma_w^2}}}{2\sqrt{2\pi\sigma_w^2}} \mathrm{d}w \\ &= \frac{1}{2}\log 2\pi e\sigma_w^2 \,. \qquad (1.16)\end{aligned}$$

In Figure 1.3, $\mathsf{h}(\mathcal{Y}|\mathcal{C})$ and $\mathsf{h}(\mathcal{Y})$ are shown as functions of the SNR. The curves are monotonically decreasing, owing to the fact that at higher SNR $\mathcal{Y}$ becomes more predictable. The difference between $\mathsf{h}(\mathcal{Y})$ and $\mathsf{h}(\mathcal{Y}|\mathcal{C})$, however, is a monotonically increasing function of the SNR. In Figure 1.4, the information rate (IR) $\mathsf{I}(\mathcal{Y};\mathcal{C}) = \mathsf{h}(\mathcal{Y}) - \mathsf{h}(\mathcal{Y}|\mathcal{C})$ is shown as a function of the SNR. At very low

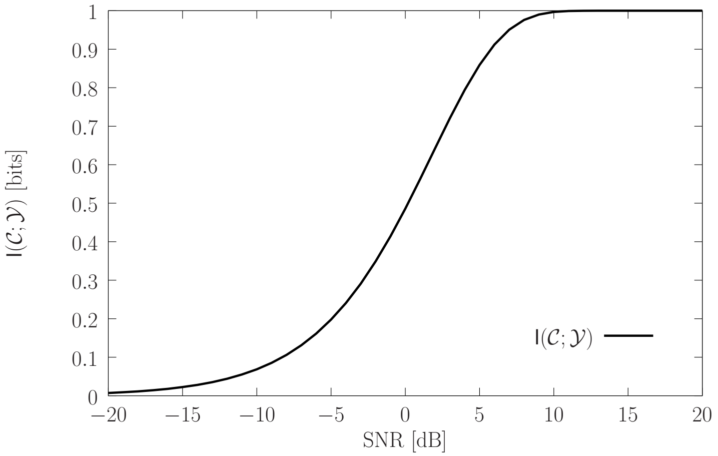

Figure 1.4: Example 1.1: information rate.

SNR, the IR is close to zero and increases as the SNR increases. Its asymptotic value is given by $\mathcal{H}(\mathcal{C}) = 1$. For each SNR value, the IR gives the maximum attainable average information rate (per transmitted binary symbol) assuming the given statistical distribution for $\mathcal{C}$. For example, at 0 dB the IR is equal to 0.486 bit: therefore, the maximum attainable data rate that can be sent through the channel is 0.486 bit per channel use (or bit per sample). In other words, codes with code rate $R_C > 0.486$ shall not achieve arbitrarily low BER at a SNR equal to 0 dB. On the other hand, for every code rate $R_C < 0.486$ there exists a code that will achieve a specified arbitrarily low BER provided that the codeword length is sufficiently large.

Another useful point of view is to interpret the IR theoretical limit in terms of the SNR required to enable achievability of a given code rate $R$. For example, from the IR curve in Figure 1.4, one can conclude that it is not possible to achieve an arbitrarily small BER with codes whose rate is equal to 0.5 if SNR$<$ 0.18 dB, regardless of the codeword length. This interpretation of the theoretical performance limits suggested by information theory will be used in this book while discussing numerical performance results for LDPC coded modulated schemes.

**Example 1.2** *Binary symmetric channel and correction of residual errors.*

Assume to transmit a sequence of i.i.d. binary symbols $\{C_k\}$, where the RV $C_k$ takes a value in the set $\{0, 1\}$ and $P\{C_k = 1\} = P\{C_k = 0\} = 1/2$. The channel outputs a binary symbol $Y_k \in \{0, 1\}$, which may be equal to $C_k$

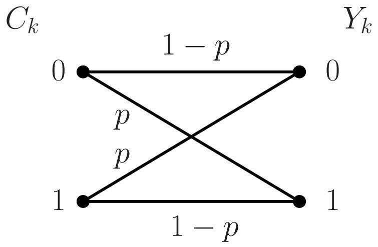

Figure 1.5: Example 1.2: binary symmetric channel.

or $1 - C_k$ according to the following rule:

$$Y_k = \begin{cases} C_k & \text{with probability } 1-p \\ 1-C_k & \text{with probability } p. \end{cases} \tag{1.17}$$

Hence, $p$ represents the probability of error in a single use of the channel. The channel is memoryless, in the sense that the outputs are conditionally independent, i.e.,

$$P(y_1, \ldots, y_n | c_1, \ldots, c_n) = \prod_{i=1}^{n} P(y_i|c_i) \, .$$

This channel is commonly known as a binary symmetric channel (BSC). In Figure 1.5, a diagram of a BSC is shown. Although this channel is very simple, it may be very useful since almost any digital communication system may be ultimately interpreted as a BSC characterized by $p$ equal to the BER, as defined in (1.1). In fact, if the processes $\mathcal{A}$ and $\hat{\mathcal{A}}$ in Figure 1.1 are binary, they could be seen as the input and the output of a binary-input binary-output channel. To obtain a memoryless binary-input binary-output channel, one may add an interleaving block in front of the input of the transmitter and a corresponding de-interleaving block at the output of the receiver. To make the overall system symmetric, i.e., to obtain equal probability of error for a bit equal to either 0 and to 1, it is sufficient to input the transmitted bit sequence to a "pseudo-random" bit flipper placed before the interleaver and to use a corresponding de-flipper of the bit sequence at the output of the de-interleaver at the receiver side.[1]

[1] Here, *pseudo-random flipping* of a bit sequence denotes flipping of the bits of a subset of the bit sequence, followed, at the receiver, by an identical operation to recover the original bit sequence. For example, one could flip all bits in even positions. In random flipping, each bit is flipped with probability 1/2. This operation is not intended for encryption purposes, but, rather, to shape the statistical characteristics of the bit sequence.

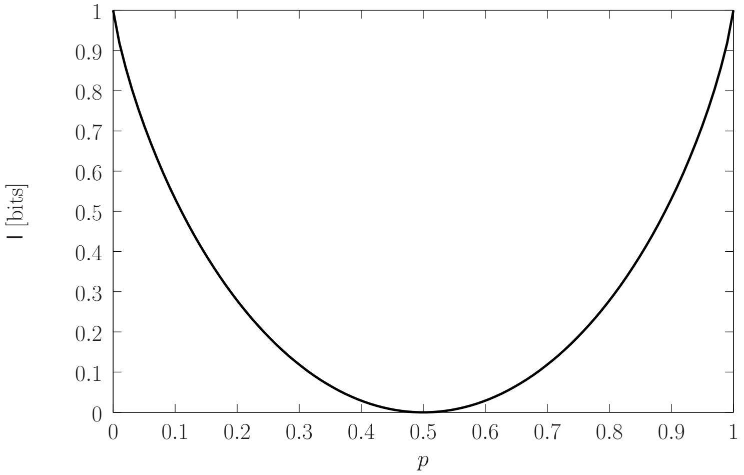

Figure 1.6: Example 1.2: information rate of a binary symmetric channel as a function of the channel error probability $p$.

The IR of a BSC with the considered input sequence of i.i.d. symbols can be expressed as

$$\begin{aligned} \mathsf{I}(\mathcal{C};\mathcal{Y}) &= \mathcal{H}(\mathcal{Y}) - \mathcal{H}(\mathcal{Y}|\mathcal{C}) \\ &= 1 - H(p) \end{aligned}$$

where

$$H(p) \triangleq -p\log_2 p - (1-p)\log_2(1-p)\,.$$

In Figure 1.6, the IR of a BSC with the considered input sequence of i.i.d. symbols is shown as a function of the BSC transition probability $p$. By interpreting the IR as the supremum of the achievable code rates, one can obtain interesting insights on systems characterized by a small, but finite, BER.

In particular, consider a communication scheme with a binary process $\mathcal{A}$ at its input and a binary process $\hat{\mathcal{A}}$, consisting of the sequence of decided bits, at its output. The system is assumed to be characterized by a low probability of error $p < 0.5$. By introducing, as discussed earlier, (i) an ideal interleaver before the system input and its corresponding de-interlever at the output of the receiver, and, if necessary, (ii) a bit-flipper and a de-flipper at the transmitter and the receiver, respectively, it is possible to transform the considered communication system into a BSC with transition probability $p$ with arbitrary accuracy. The obtained system is completely characterized by

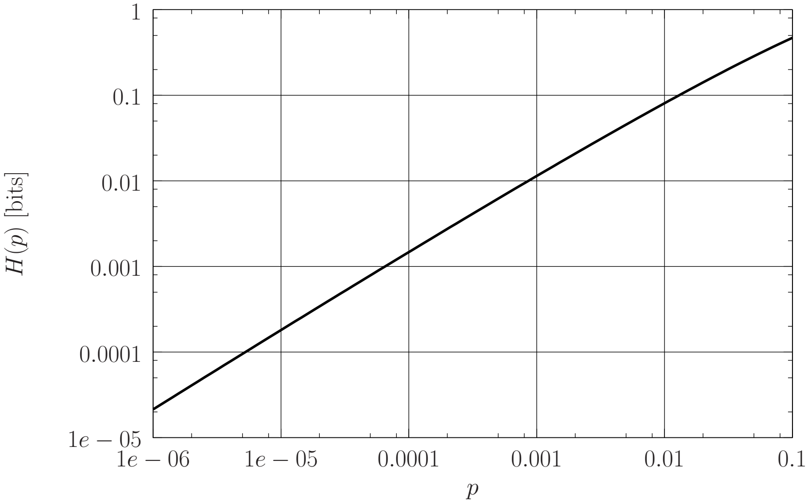

Figure 1.7: Example 2: $H(p)$, i.e., the fraction of bits that must be devoted to outer error correction to obtain arbitrarily small error probability from a communication system characterized by BER equal to $p$.

its IR, since the function $H(p)$ is invertible for $0 \leq p \leq 0.5$. In particular, by means of a proper outer error correction code for a BSC, with a code rate lower than the IR, it is possible to obtain arbitrarily small BER. In this sense, $1 - \mathsf{I}(\mathcal{C};\mathcal{Y}) = H(p)$ is the minimum fraction of bits that must be "wasted" to guarantee arbitrarily small error probability. In Figure 1.7, the minimum fraction of bits that must be devoted to redundancy for error correction coding of the BSC obtained from a generic communication system is shown considering a BER range from $10^{-6}$ to $10^{-1}$.

For instance, if a communication system achieves a BER equal to $10^{-3}$, a fraction at least equal to $H(10^{-3}) \simeq 0.011 \simeq 1\%$ of bits in the transmitted stream must be devoted to guaranteeing an arbitrarily small error rate. In other words, a system with BER equal to $10^{-3}$ may be used to build a system with arbitrarily low error probability and effective data rate only 1% lower than that of an uncoded system.

## 1.6 The Following Chapters

This chapter introduced some of the concepts that will be used in the remainder of this book. In particular, basic concepts on the detection techniques of interest will be presented in Chapter 2. The structure and characterization of LDPC codes will be discussed in more detail in Chapter 3. The basic information-theoretic concepts introduced in the current chapter are useful for understanding the theoretical limits that one is facing when designing an advanced communication system. In particular, the extrinsic information transfer (EXIT) chart-based analysis, described in Chapter 4, is based on the concept of mutual information. Bounds, based on mutual information, on the BER of LDPC coded modulated schemes will be given in Chapter 5. Chapter 6 deals with the design of LDPC coded modulations for memoryless channels, whereas Chapter 7 focuses on differentially encoded LDPC coded modulations. Finally, in Chapter 8 some final remarks are provided.