FUNDAMENTALS OF
CONVOLUTIONAL CODING
To Jim Massey, 1934 - 2013
our friend and mentor

To our clans\textsuperscript{1}

\textbullet\quad Regina
\quad \quad \quad \quad Katrin
\quad \quad \quad \quad Hugo Simon Liv
\quad \quad Peter
\quad \quad Oliver Oskar Stella Emma

\quad Hanna

\quad Dima
\quad \quad Valja
\quad Natalia Alex Mischa

\textsuperscript{1}See Figure 8.2.
CONTENTS

Preface xi

Acknowledgement xiv

1 Introduction 1
  1.1 Why error control? 1
  1.2 Block codes—a primer 8
  1.3 Codes on graphs 21
  1.4 A first encounter with convolutional codes 28
  1.5 Block codes versus convolutional codes 35
  1.6 Capacity limits and potential coding gain revisited 36
  1.7 Comments 39
  Problems 41

2 Convolutional encoders—Structural properties 49
  2.1 Convolutional codes and their encoders 49
## CONTENTS

2.2 The Smith form of polynomial convolutional generator matrices 58  
2.3 Encoder inverses 67  
2.4 Encoder and code equivalences 76  
2.5 Basic encoding matrices 79  
2.6 Minimal-basic encoding matrices 82  
2.7 Minimal encoding matrices and minimal encoders 90  
2.8 Canonical encoding matrices* 109  
2.9 Minimality via the invariant-factor theorem* 127  
2.10 Syndrome formers and dual encoders 131  
2.11 Systematic convolutional encoders 139  
2.12 Some properties of generator matrices—an overview 150  
2.13 Comments 150  
Problems 152

3 Distance properties of convolutional codes 161  
3.1 Distance measures—a first encounter 161  
3.2 Active distances 171  
3.3 Properties of convolutional codes via the active distances 179  
3.4 Lower bound on the distance profile 181  
3.5 Upper bounds on the free distance 186  
3.6 Time-varying convolutional codes 191  
3.7 Lower bound on the free distance 195  
3.8 Lower bounds on the active distances* 200  
3.9 Distances of cascaded concatenated codes* 207  
3.10 Path enumerators 213  
3.11 Comments 220  
Problems 221

4 Decoding of convolutional codes 225  
4.1 The Viterbi algorithm revisited 226  
4.2 Error bounds for time-invariant convolutional codes 235  
4.3 Tighter error bounds for time-invariant convolutional codes 250  
4.4 Exact bit error probability for Viterbi decoding 255  
4.5 The BCJR algorithm for APP decoding 271  
4.6 The one-way algorithm for APP decoding 283  
4.7 A simple upper bound on the bit error probability for extremely noisy channels 288  
4.8 Tailbiting trellises 293
4.9  Decoding of tailbiting codes  302
4.10 BEAST decoding of tailbiting codes  308
4.11 Comments  323
Problems  324

5  Random ensemble bounds for decoding error probability  333
5.1  Upper bounds on the output error burst lengths  333
5.2  Bounds for periodically time-varying convolutional codes  345
5.3  Lower error probability bounds for convolutional codes  355
5.4  General bounds for time-varying convolutional codes  363
5.5  Bounds for finite back-search limits  375
5.6  Quantization of channel outputs  379
5.7  Comments  384
Problems  384

6  List decoding  387
6.1  List decoding algorithms  388
6.2  List decoding—performance  391
6.3  The list minimum weight  397
6.4  Upper bounds on the probability of correct path loss  407
6.5  Lower bound on the probability of correct path loss  416
6.6  Correct path loss for time-invariant convolutional codes  418
6.7  Comments  422
Problems  423

7  Sequential decoding  425
7.1  The Fano metric  426
7.2  The stack algorithm  431
7.3  The Fano algorithm  433
7.4  The Creeper algorithm*  436
7.5  Simulations  448
7.6  Computational analysis of the stack algorithm  450
7.7  Error probability analysis of the stack algorithm  460
7.8  Analysis of the Fano algorithm  471
7.9  Analysis of Creeper*  477
7.10 Comments  480
Problems  481
## 8 Low-density parity-check codes

8.1 LDPC block codes  
8.2 LDPC convolutional codes  
8.3 Block and convolutional permutors  
8.4 Lower bounds on distances of LDPC codes  
8.5 Iterative decoding of LDPC codes  
8.6 Iterative limits and thresholds  
8.7 Braided block codes*  
8.8 Comments  
Problems  

## 9 Turbo coding

9.1 Parallel concatenation of two convolutional codes  
9.2 Distance bounds of turbo codes  
9.3 Parallel concatenation of three and more convolution codes  
9.4 Iterative decoding of turbo codes  
9.5 Braided convolutional codes*  
9.6 Comments  
Problems  

## 10 Convolutional codes with good distance properties

10.1 Computing the Viterbi spectrum using FAST  
10.2 The magnificent BEAST  
10.3 Some classes of rate $R = 1/2$ convolutional codes  
10.4 Low rate convolutional codes  
10.5 High rate convolutional codes  
10.6 Tailbiting trellis encoders  
10.7 Comments  

Appendix A: Minimal encoders  
Appendix B: Wald’s identity  
References  
Index
Our goal with this book is to present a comprehensive treatment of convolutional codes, their construction, their properties, and their performance. The purpose is that the book could serve as a graduate-level textbook, be a resource for researchers in academia, and be of interest to industry researchers and designers.

This book project started in 1989 and the first edition was published in 1999. The work on the second edition began in 2009. By now the material presented here has been maturing in our minds for more than 40 years, which is close to our entire academic lives. We believed that the appearance of some of David Forney’s important structural results on convolutional encoders in a textbook was long overdue. For us, that and similar thoughts on other areas generated new research problems. Such interplays between research and teaching were delightful experiences. This second edition is the final result of those experiences.

Chapter 1 provides an overview of the essentials of coding theory. Capacity limits and potential coding gains, classical block codes, convolutional codes, Viterbi decoding, and codes on graphs are introduced. In Chapter 2, we give formal definitions of convolutional codes and convolutional encoders. Various concepts of minimality are discussed in-depth using illuminative examples. Chapter 3 is devoted to a flurry of distances of convolutional codes. Time-varying convolutional codes are introduced.
and upper and lower distance bounds are derived. An in-depth treatment of Viterbi decoding is given in Chapter 4, including both error bounds and tighter error bounds for time-invariant convolutional codes as well as a closed-form expression for the exact bit error probability. Both the BCJR (Bahl-Cocke-Jelinek-Raviv) and the one-way algorithms for a posteriori probability decoding are discussed. A simple upper bound on the bit error probability for extremely noisy channels explains why it is important that the constituent convolutional encoders are systematic when iterative decoding is considered. The chapter is concluded by a treatment of tailbiting codes, including their BEAST (Bidirectional Efficient Algorithm for Searching Trees) decoding. In Chapter 5, we derive various random ensemble bounds for the decoding error probability. As an application we consider quantization of channel outputs. Chapter 6 is devoted to list decoding of convolutional codes, which is thoroughly analyzed. Once again we make the important conclusion that there are situations when it is important that a convolutional encoder is systematic. In Chapter 7, we discuss a subject that is close to our hearts, namely sequential decoding. Both our theses were on that subject. We describe and analyze the stack algorithm, the Fano algorithm, and Creeper. James Massey regarded the Fano algorithm as being the most clever algorithm among all algorithms! Chapters 8 and 9 rectify the lack of a proper treatment of low-density parity-check (LDPC) codes and turbo codes in the first edition, where these important areas got a too modest section. These codes revolutionized the world of coding theory at the end of the previous millennium. In Chapter 8, the LDPC block codes, which were invented by Robert Gallager and appeared in his thesis, are discussed. Then they are generalized to LDPC convolutional codes, which were invented by the second author and his graduate student Alberto Jiménez-Felström. They are discussed in-depth together with bounds on their distances. Iterative decoding is introduced and iterative limits and thresholds are derived. The chapter is concluded by the introduction of the related braided block codes. Turbo codes are treated in Chapter 9 together with bounds on their distances and iterative decoding. Moreover, the braided block codes are generalized to their convolutional counterpart. In Chapter 10, we introduce two efficient algorithms, FAST (Fast Algorithm for Searching Trees) and BEAST, for determining code distances. FAST was designed to determine the Viterbi spectrum for convolutional encoders while using BEAST we can determine the spectral components for block codes as well. Extensive lists are given of the best known code generators with respect to free distance, numbers of nearest neighbors and of information bit errors, Viterbi spectrum, distance profile, and minimum distance. These lists contain both nonsystematic and systematic generators. In Appendix A we demonstrate how to minimize two examples of convolutional encoders and in Appendix B we present Wald’s identity and related results that are necessary for our analyses in Chapters 3–7.

For simplicity’s sake, we restricted ourselves to binary convolutional codes. In most of our derivations of the various bounds we only considered the binary symmetric channel (BSC). Although inferior from a practical communications point of view, we believe that its pedagogical advantages outweigh that disadvantage.
Each chapter ends with some comments, mainly historical in nature, and sets of problems that are highly recommended. Many of those were used by us as exam questions. Note that sections marked with asterisk (*) can be skipped at the first reading without loss of continuity.

There are various ways to organize the material into an introductory course on convolutional coding. Chapter 1 should always be read first. Then one possibility is to cover the following sections, skipping most of the proofs found there: 2.1–2.7, 2.10, 2.13, 3.1, 3.5, 3.10, 3.11, 4.1–4.2, 4.5, 4.7–4.11, 5.6, 6.1–6.2, 7.1–7.3, 7.5, 7.10, 8.1–8.6, 8.8, 9.1–9.4, 9.6, and perhaps also 10.1, 10.2, and 10.7. With our younger students (seniors), we emphasize explanations, and discussions of algorithms and assign a good deal of the problems found at the end of the chapters. With the graduate students we stress proving theorems, because a good understanding of the proofs is an asset in advanced engineering work.

Finally, we do hope that some of our passion for convolutional coding has worked its way into these pages.

Rolf Johannesson

Kamil Sh. Zigangirov
Rolf is particularly grateful to Göran Einarsson, who more than 40 years ago not only suggested convolutional codes as Rolf’s thesis topic but also recommended that he spend a year of his graduate studies with James Massey at the University of Notre Dame. This year was the beginning of a lifelong devotion to convolutional codes and a lifelong genuine friendship which lasted until Jim passed away in 2013. The influence of Jim cannot be overestimated. Rolf would also like to acknowledge David Forney’s outstanding contributions to the field of convolutional codes; without his work convolutional codes would have been much less exciting.

Kamil would like to thank his colleagues at the Institute for Problems of Information Transmission in Moscow. In particular, he would like to mention the “daytime” seminars organized by the late Roland Dobrushin and the late Mark Pinsker and the “evening” seminars organized by Leonid Bassalygo. During these seminars, Kamil had the opportunity to use the participants as guinea pigs when he wanted to test many of the fundamental ideas presented in this book. Special thanks go to Mark Pinsker and Victor Zyablov. Mark introduced Kamil to convolutional codes, and he inspired Kamil for more than 30 years. Over the years, Kamil has also benefited from numerous discussions with Victor.
In our research we have benefited considerably from our cooperation—sometimes long lasting, sometimes a single joint paper—with colleagues John Anderson, Irina Bocharova, Daniel Costello, David Forney, Boris Kudryashov, James Massey, Zhe-Xian Wan, and Victor Zyablov.

Needless to say, we would not have reached our view of convolutional codes without our work with former graduate students: Gunilla Bratt, Mats Cedervall, Karin Engdahl, Marc Handlery, Florian Hug, Stefan Höst, Alberto Jiménez-Felström, Michael Lentmaier, Per Ljungberg, Maja Lončar, Johan Nyström, Harro Osthoff, Victor Pavlushkov, Joakim Persson, Ben Smeets, Per Ståhl, Dmitry Trukhachev, Kristian Wahlgren, Ola Winzell, and Emma Wittenmark, who have all contributed in many ways. They inspired us, produced research results, did simulations, provided us with various figures, suggested improvements, saved us from many blunders, and much more. We are grateful to Florian Hug, Alberto Jiménez-Felström, Michael Lentmaier, Maja Lončar, Johan Nyström, Harro Osthoff, and Dmitry Trukhachev for permitting us to use pieces of LaTeX source code taken from their theses. We are greatly indebted to Lena Månsson, who with great enthusiasm and outstanding skill typed the first edition and partly the second edition and above all to LaTeX-wizard Florian Hug who volunteered to finalize the second edition. Without Florian’s editing and composition expertise this edition would not have been completed. Thanks Florian!

Special thanks go to Mary Hatcher, Danielle LaCourciere, and Brady Chin at Wiley for their support. We are grateful to Beatrice Ruberto who did an outstanding job and saved us from making embarrassing blunders. In particular, we appreciate that Mary turned out to have the patience of an angel when Rolf numerous times failed to meet deadlines.
CHAPTER 1

INTRODUCTION

1.1 WHY ERROR CONTROL?

The fundamental idea of information theory is that all communication is essentially digital—it is equivalent to generating, transmitting, and receiving randomly chosen binary digits, bits. When these bits are transmitted over a communication channel—or stored in a memory—it is likely that some of them will be corrupted by noise. In his 1948 landmark paper “A Mathematical Theory of Communication” [Sha48] Claude E. Shannon recognized that randomly chosen binary digits could (and should) be used for measuring the generation, transmission, and reception of information. Moreover, he showed that the problem of communicating information from a source over a channel to a destination can always be separated—without sacrificing optimality—into the following two subproblems: representing the source output efficiently as a sequence of binary digits (source coding) and transmitting binary, random, independent digits over the channel (channel coding). In Fig. 1.1 we show a general digital communication system. We use Shannon’s separation principle and split the encoder and decoder into two parts each as shown in Fig. 1.2. The channel coding parts can be designed independently of the source coding parts, which simplifies the use of the same communication channel for different sources.
To a computer specialist, “bit” and “binary digit” are entirely synonymous. In information theory, however, “bit” is Shannon’s unit of information [Sha48, Mas82]. For Shannon, *information* is what we receive when uncertainty is reduced. We get exactly 1 bit of information from a binary digit when it is drawn in an experiment in which successive outcomes are independent of each other and both possible values, 0 and 1, are equiprobable; otherwise, the information is less than 1. In the sequel, the intended meaning of “bit” should be clear from the context.

![Figure 1.1](Image1.png)

**Figure 1.1** Overview of a digital communication system.

![Figure 1.2](Image2.png)

**Figure 1.2** A digital communication system with separate source and channel coding.

Shannon’s celebrated channel coding theorem states that every communication channel is characterized by a single parameter $C_t$, the *channel capacity*, such that $R_t$ randomly chosen bits per second can be transmitted arbitrarily reliably over the channel if and only if $R_t \leq C_t$. We call $R_t$ the *data transmission rate*. Both $C_t$ and $R_t$ are measured in bits per second. Shannon showed that the specific value of the *signal-to-noise ratio* is not significant as long as it is large enough, that is, so large that $R_t \leq C_t$ holds; what matters is how the information bits are encoded. The information should not be transmitted one information bit at a time, but long information sequences should be *encoded* such that each information bit has some influence on many of the bits transmitted over the channel. This radically new idea gave birth to the subject of *coding theory*.

*Error control coding* should protect digital data against errors that occur during transmission over a noisy communication channel or during storage in an unreliable memory. The last decades have been characterized not only by an exceptional increase in data transmission and storage but also by a rapid development in micro-electronics,
providing us with both a need for and the possibility of implementing sophisticated algorithms for error control.

Before we study the advantages of coding, we shall consider the digital communication channel in more detail. At a fundamental level, a channel is often an analog channel that transfers waveforms (Fig. 1.3). Digital data $u_0u_1u_2$, where $u_i = 0 \ 1$, must be modulated into waveforms to be sent over the channel.

![Figure 1.3](image-url) A decomposition of a digital communication channel.

In communication systems where carrier phase tracking is possible (coherent demodulation), phase-shift keying (PSK) is often used. Although many other modulation systems are in use, PSK systems are very common and we will use one of them to illustrate how modulations generally behave. In binary PSK (BPSK), the modulator generates the waveform

$$s_1(t) = \frac{2E_s}{T} \cos \left\{ \begin{array}{ll} \frac{2E_s}{T} \cos t & 0 < t < T \\ 0 & \text{otherwise} \end{array} \right. \quad (1.1)$$

for the input 1 and $s_0(t) = -s_1(t)$ for the input 0. This is an example of antipodal signaling. Each symbol has duration $T$ seconds and energy $E_s = ST$, where $S$ is the power and $\omega = \frac{2\pi}{T}$. The transmitted waveform is

$$v(t) = \sum_{i=0}^{\infty} s_{u_i}(t - iT) \quad (1.2)$$

Assume that we have a waveform channel such that additive white Gaussian noise (AWGN) $n(t)$ with zero mean and two-sided power spectral density $N_0 / 2$ is added to the transmitted waveform $v(t)$, that is, the received waveform $r(t)$ is given by

$$r(t) = v(t) + n(t) \quad (1.3)$$

where

$$E[n(t)] = 0 \quad (1.4)$$
and
\[ E[n(t + \tau)n(t)] = \frac{N_0}{2} \]  
(1.5)

where \( E[ \cdot ] \) and \( (\cdot) \) denote the mathematical expectation and the delta function, respectively.

Based on the received waveform during a signaling interval, the demodulator produces an estimate of the transmitted symbol. The optimum receiver is a matched filter with impulse response
\[ h(t) = \begin{cases} \frac{2T}{\sqrt{2}} \cos t & 0 < t < T \\ 0 & \text{else} \end{cases} \]  
(1.6)

which is sampled each \( T \) seconds (Fig. 1.4). The matched filter output \( Z_i \) at the sample time \( iT \),
\[ Z_i = \int_{iT}^{iT+1} r(\tau) h(iT - \tau) d\tau \]  
(1.7)

is a Gaussian random variable \( N(\mu, \sigma^2) \) with mean
\[ = \int_{0}^{T} \frac{2E_s}{T} \cos \frac{\sqrt{2}}{T} \cos \left( T \cos t \right) \cos \left( T \cos t \right) d\tau = \frac{E_s}{2} \]  
(1.8)

where the sign is + or − according to whether the modulator input was 1 or 0, respectively, and variance
\[ 2 = \frac{N_0}{2} \int_{0}^{T} \frac{2}{T} \cos \frac{\sqrt{2}}{T} \cos \left( T \cos t \right) \cos \left( T \cos t \right) d\tau = \frac{N_0}{2} \]  
(1.9)

After the sampler we can make a hard decision, that is, a binary quantization with threshold zero, of the random variable \( Z_i \). Then we obtain the simplest and most important binary-input and binary-output channel model, the binary symmetric channel (BSC) with crossover probability \( \epsilon \) (Fig. 1.5). The crossover probability is of course closely related to the signal-to-noise ratio \( E_s/N_0 \). Since the channel output for a given signaling interval depends only on the transmitted waveform and noise during that interval and not on other intervals, the channel is said to be memoryless.

Because of symmetry, we can without loss of generality assume that a 0, that is, \( \frac{2E_s}{T} \cos \ t \), is transmitted over the channel. Then we have a channel “error”
WHY ERROR CONTROL?

0 0
1 1

\[ \epsilon = 1 - \sqrt{\frac{2}{\pi N_0}} \int_{\infty}^{0} e^{-\left(\frac{z+\sqrt{E_s}}{\sqrt{N_0}}\right)^2} dz \]

where

\[ f_{Z_i}(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \]

we have

\[ Q(x) = \frac{1}{2} \sqrt{2\pi} e^{-\frac{x^2}{4}} \int_{x}^{\infty} e^{-\frac{y^2}{2}} dy \]

is the complementary error function of Gaussian statistics (often called the Q-function).

When coding is used, we prefer measuring the energy per information bit, \( E_b \), rather than per symbol. For uncoded BPSK, we have \( E_b = E_s \). Letting \( P_b \) denote the bit error probability (or bit error rate), that is, the probability that an information bit is erroneously delivered to the destination, we have for uncoded BPSK

\[ P_b = Q\left(\frac{\sqrt{2E_s N_0}}{2E_b N_0}\right) \]

How much better can we do with coding?

It is clear that when we use coding, it is a waste of information to make hard decisions. Since the influence of each information bit will be spread over several channel symbols, the decoder can benefit from using the value of \( Z_i \) (hard decisions use only the sign of \( Z_i \)) as an indication of how reliable the received symbol is. The demodulator can give the analog value of \( Z_i \) as its output, but it is often more
practical to use, for example, a three-bit quantization—a soft decision. By introducing seven thresholds, the values of $Z_i$ are divided into eight intervals and we obtain an eight-level soft-quantized discrete memoryless channel (DMC) as shown in Fig. 1.6.

Shannon [Sha48] showed that the capacity of the bandlimited AWGN channel with bandwidth $W$ is\footnote{Here and hereafter we write log for log$_2$.}

$$C_t^W = W \log_2 1 + \frac{S}{N_0 W} \text{ bits/s}$$

(1.15)

where $N_0$ and $S$ denote the two-sided noise spectral density and the signaling power, respectively. If the bandwidth $W$ goes to infinity, we have

$$C_t \overset{\text{def}}{=} \lim_{W \to \infty} W \log_2 1 + \frac{S}{N_0 W} = \frac{S}{N_0 \ln 2} \text{ bits/s}$$

(1.16)

If we transmit $K$ information bits during $\tau$ seconds, where $\tau$ is a multiple of bit duration $T$, we have

$$E_b = \frac{S}{K}$$

(1.17)

Since the data transmission rate is $R_t = K \text{ bits/s}$, the energy per bit can be written

$$E_b = \frac{S}{R_t}$$

(1.18)

Combining (1.16) and (1.18) gives

$$\frac{C_t}{R_t} = \frac{E_b}{N_0 \ln 2}$$

(1.19)
Figure 1.7  Capacity limits and regions of potential coding gain.

From Shannon’s celebrated channel coding theorem [Sha48] it follows that for reliable communication we must have $R_t \leq C_t$. Hence, from this inequality and (1.19) we have

$$\frac{E_b}{N_0} \ln 2 = 0.69 = 1.6 \text{ dB}$$

(1.20)

which is the fundamental Shannon limit.

In any system that provides reliable communication in the presence of additive white Gaussian noise the signal-to-noise ratio $E_b/N_0$ cannot be less than the Shannon limit, 1.6 dB!

On the other hand, as long as $E_b/N_0$ exceeds the Shannon limit, 1.6 dB, Shannon’s channel coding theorem guarantees the existence of a system—perhaps very complex—for reliable communication over the channel.

In Fig. 1.7, we have plotted the fundamental limit of (1.20) together with the bit error rate for uncoded BPSK, that is, equation (1.14). At a bit error rate of $10^{-5}$, the infinite-bandwidth additive white Gaussian noise channel requires an $E_b/N_0$ of at least 9.6 dB. Thus, at this bit error rate we have a potential coding gain of 11.2 dB!
For the bandlimited AWGN channel with BPSK and hard decisions, that is, a BSC with crossover probability $\epsilon$ (Fig. 1.5) Shannon [Sha48] showed that the capacity is

$$C_{BSC}^t = 2W(1 - h(\epsilon)) \text{ bits/s}$$ \hspace{1cm} (1.21)

where

$$h(\epsilon) = -\epsilon \log_2 \epsilon - (1 - \epsilon) \log_2 (1 - \epsilon)$$ \hspace{1cm} (1.22)

is the binary entropy function. If we restrict ourself to hard decisions, we can use (1.21) and show (Problem 1.2) that for reliable communication we must have

$$\frac{E_b}{N_0} \geq \frac{\pi}{2} \log_2 2 = 1.09 = 0.4 \text{ dB}$$ \hspace{1cm} (1.23)

In terms of capacity, soft decisions are about 2 dB more efficient than hard decisions.

Although it is practically impossible to obtain the entire theoretically promised 11.2 dB coding gain, communication systems that pick up 2–8 dB are routinely in use. During the last decade iterative decoding has been used to design communication systems that operate only tenths of a dB from the Shannon limit.

We conclude this section, which should have provided some motivation for the use of coding, with an adage from R. E. Blahut [Bla92]: “To build a communication channel as good as we can is a waste of money”—use coding instead!

### 1.2 BLOCK CODES—A PRIMER

For simplicity, we will deal only with binary block codes. We consider the entire sequence of information bits to be divided into blocks of $K$ bits each. These blocks are called messages and denoted $u = u_0 u_1 \ldots u_{K-1}$. In block coding, we let $u$ denote a message rather than the entire information sequence as is the case in convolutional coding to be considered later.

A binary $(N,K)$ block code $B$ is a set of $M = 2^K$ binary $N$-tuples (or row vectors of length $N$) $v = v_0 v_1 \ldots v_{N-1}$ called codewords. $N$ is called the block length and the ratio

$$R = \frac{\log M}{N} = \frac{K}{N}$$ \hspace{1cm} (1.24)

is called the code rate and is measured in bits per (channel) use. The data transmission rate in bits/s is obtained by multiplying the code rate (1.24) by the number of transmitted channel symbols per second:

$$R_t = R \cdot T$$ \hspace{1cm} (1.25)

If we measure the channel capacity for the BSC in bits/channel use (bits c.u.), then the capacity of the BSC equals

$$C = 1 - h(\epsilon) \text{ (bits c.u.)}$$ \hspace{1cm} (1.26)

According to Shannon’s channel coding theorem, for reliable communication, we must have $R \leq C$ and the block length $N$ should be chosen sufficiently large.
Figure 1.8 A binary symmetric channel (BSC) with (channel) encoder and decoder.

**EXAMPLE 1.1**

The set  \( B = \{000, 011, 101, 110\} \) is a \((3, 2)\) code with four codewords and rate \( R = \frac{2}{3} \).

An encoder for an \((N, K)\) block code \( B \) is a one-to-one mapping from the set of \( M = 2^K \) binary messages to the set of codewords \( B \).

**EXAMPLE 1.2**

\[
\begin{array}{c|ccc}
   & v_0 & v_1 & v_2 \\
\hline
00 & 000 & & \\
01 & 011 & & \\
10 & 101 & & \\
11 & 110 & &
\end{array}
\quad
\begin{array}{c|ccc}
   & v_0 & v_1 & v_2 \\
\hline
00 & 101 & & \\
01 & 011 & & \\
10 & 110 & & \\
11 & 000 & &
\end{array}
\]

are two different encoders for the code \( B \) given in the previous example.

The rate \( R = K/N \) is the fraction of the digits in the codeword that are necessary to represent the information; the remaining fraction, \( 1 - R = (N - K)/N \), represents the redundancy that can be used to detect or correct errors.

Suppose that a codeword \( v \) corresponding to message \( u \) is sent over a BSC (see Fig. 1.8). The channel output \( r = r_0 r_1 \ldots r_{N-1} \) is called the received sequence. The decoder transforms the received \( N\)-tuple \( r \), which is possibly corrupted by noise, into the \( K\)-tuple \( u \), called the estimated message \( \hat{u} \). Ideally, \( u \) will be a replica of the message \( u \), but the noise may cause some decoding errors. Since there is a one-to-one correspondence between the message \( u \) and the codeword \( v \), we can, equivalently, consider the corresponding \( N\)-tuple \( \hat{v} \) as the decoder output. If the codeword \( v \) was transmitted, a decoding error occurs if and only if \( \hat{v} \neq v \).

Let \( P_E \) denote the block (or word) error probability, that is, the probability that the decision \( v \) for the codeword differs from the transmitted codeword \( v \). Then we have

\[
P_E = P(v = v | r) P(r)
\]

where the probability that we receive \( r \), \( P(r) \), is independent of the decoding rule and \( P(v = v | r) \) is the conditional probability of decoding error given the received sequence \( r \). Hence, in order to minimize \( P_E \), we should specify the decoder such
that $P(v = v \mid r)$ is minimized for a given $r$ or, equivalently, such that $P(v = v \mid r) \overset{\text{def}}{=} P(v = v \mid r)$ is maximized for a given $r$. Thus the block error probability $P_E$ is minimized by the decoder, which as its output chooses $v$ such that the corresponding $v = v$ maximizes $P(v = v \mid r)$. That is, $v$ is chosen as the most likely codeword given that $r$ is received. This decoder is called a maximum a posteriori probability (MAP) decoder.

Using Bayes' rule we can write

$$P(v = v \mid r) = \frac{P(r \mid v) P(v)}{P(r)}$$

The code carries the most information possible with a given number of codewords when the codewords are equally likely. It is reasonable to assume that a decoder that is designed for this case also works satisfactorily—although not optimally—when the codewords are not equally likely, that is, when less information is transmitted. When the codewords are equally likely, maximizing $P(v = v \mid r)$ is equivalent to maximizing $P(r \mid v)$. The decoder that makes its decision $v = v$ such that $P(r \mid v)$ is maximized is called a maximum-likelihood (ML) decoder.

Notice that in an erroneous decision for the codeword some of the information digits may nevertheless be correct. The bit error probability, which we introduced in the previous section, is a better measure of quality in most applications. However, it is in general more difficult to calculate. The bit error probability depends not only on the code and on the channel, like the block error probability, but also on the encoder and on the information symbols!

The use of block error probability as a measure of quality is justified by the inequality

$$P_b \leq P_E$$

When $P_E$ can be made very small, inequality (1.29) implies that $P_b$ can also be made very small.

The Hamming distance between the two $N$-tuples $r$ and $v$, denoted $d_H(r \mid v)$, is the number of positions in which their components differ.

**EXAMPLE 1.3**

Consider the 5-tuples 10011 and 11000. The Hamming distance between them is 3.

The Hamming distance, which is an important concept in coding theory, is a metric; that is,

(i) $d_H(x \mid y) = 0$, with equality if and only if $x = y$ (positive definiteness)

(ii) $d_H(x \mid y) = d_H(y \mid x)$ (symmetry)

(iii) $d_H(x \mid y) = d_H(x \mid z) + d_H(z \mid y)$, all $z$ (triangle inequality)

The Hamming weight of an $N$-tuple $x = x_0 x_1 \ldots x_{N-1}$, denoted $w_H(x)$, is defined as the number of nonzero components in $x$. 
For the BSC, a transmitted symbol is erroneously received with probability where \( \epsilon \) is the channel crossover probability. Thus, assuming ML decoding, we must make our decision \( \hat{v} \) for the codeword \( v \) to maximize \( P(r | v) \); that is,

\[
\hat{v} = \arg \max_v P(r | v) \tag{1.30}
\]

where

\[
P(r | v) = d_H(r, v)(1 - \epsilon)^N d_H(r, v) = (1 - \epsilon)^N \frac{d_H(r, v)}{1} \tag{1.31}
\]

Since \( 0 < \epsilon < \frac{1}{2} \) for the BSC, we have

\[
0 < \epsilon < 1 \tag{1.32}
\]

and, hence, maximizing \( P(r | v) \) is equivalent to minimizing \( d_H(r, v) \). Clearly, ML decoding is equivalent to minimum (Hamming) distance (MD) decoding, that is, choosing as the decoder output the message \( \hat{u} \) whose corresponding codeword \( \hat{v} \) is (one of) the closest codeword(s) to the received sequence \( r \).

In general, the decoder must compare the received sequence \( r \) with all \( M = 2^K = 2^{RN} \) codewords. The complexity of ML or MD decoding grows exponentially with the block length \( N \). Thus it is infeasible to decode block codes with large block lengths. But to obtain low decoding error probability we have to use codes with relatively large block lengths. One solution of this problem is to use codes with algebraic properties that can be exploited by the decoder. Other solutions are to use codes on graphs (see Section 1.3) or convolutional codes (see Section 1.4).

In order to develop the theory further, we must introduce an algebraic structure.

A field is an algebraic system in which we can perform addition, subtraction, multiplication, and division (by nonzero numbers) according to the same associative, commutative, and distributive laws as we use with real numbers. Furthermore, a field is called finite if the set of numbers is finite. Here we will limit the discussion to block codes whose codewords have components in the simplest, but from a practical point of view also the most important, finite field, the binary field, \( \mathbb{F}_2 \), for which the rules for addition and multiplication are those of modulo-two arithmetic, namely

\[
\begin{array}{c|cc}
+ & 0 & 1 \\
0 & 0 & 1 \\
1 & 1 & 0 \\
\end{array}
\quad
\begin{array}{c|cc}
0 & 1 \\
0 & 0 & 0 \\
1 & 0 & 1 \\
\end{array}
\]

We notice that addition and subtraction coincide in \( \mathbb{F}_2 \)!

The set of binary \( N \)-tuples are the vectors in an \( N \)-dimensional vector space, denoted \( \mathbb{F}_2^N \), over the field \( \mathbb{F}_2 \). Vector addition is component-by-component addition in \( \mathbb{F}_2 \). The scalars are the elements in \( \mathbb{F}_2 \). Scalar multiplication of scalar \( a \in \mathbb{F}_2 \) and vector \( x_0 x_1 \ldots x_{N-1} \in \mathbb{F}_2^N \) is carried out according to the rule

\[
a(x_0 x_1 \ldots x_{N-1}) = ax_0 ax_1 \ldots ax_{N-1} \tag{1.33}
\]

Since \( a \) is either 0 or 1, scalar multiplication is trivial in \( \mathbb{F}_2^N \).
Hamming weight and distance are clearly related:

\[ d_H(x, y) = w_H(x) = w_H(x + y) \]  

(1.34)

where the arithmetic is in the vector space \( \mathbb{F}_2^N \) and where the last equality follows from the fact that subtraction and addition coincide in \( \mathbb{F}_2 \).

The minimum distance, \( d_{\text{min}} \), of a code \( B \) is defined as the minimum value of \( d_H(v, v') \) over all \( v \) and \( v' \) in \( B \) such that \( v \neq v' \).

**Example 1.4**

The code \( B \) in Example 1.1 has \( d_{\text{min}} = 2 \). It is a single-error-detecting code (see Problem 1.3).

Let \( v \) be the actual codeword and \( r \) the possibly erroneously received version of it. The error pattern \( e = e_0 e_1 \ldots e_{N-1} \) is the \( N \)-tuple that satisfies

\[ r = v + e \]  

(1.35)

The number of errors is

\[ w_H(e) = d_H(r, v) \]  

(1.36)

Let \( E_t \) denote the set of all error patterns with \( t \) or fewer errors, that is,

\[ E_t = \{ e | w_H(e) \leq t \} \]  

(1.37)

We will say that a code \( B \) corrects the error pattern \( e \) if for all \( v \) the decoder maps \( r = v + e \) into \( \hat{v} = v \). If \( B \) corrects all error patterns in \( E_t \) and there is at least one error pattern in \( E_{t+1} \) which the code cannot correct, then \( t \) is called the error-correcting capability of \( B \).

**Theorem 1.1** The code \( B \) has error-correcting capability \( t \) if and only if \( d_{\text{min}} > 2t \).

**Proof:** Suppose that \( d_{\text{min}} > 2t \). Consider the decoder which chooses \( v \) as (one of) the codeword(s) closest to \( r \) in Hamming distance (MD decoding). If \( r = v + e \) and \( e \in E_t \), then \( d_H(r, v) \leq t \). The decoder output \( v \) must also satisfy \( d_H(r, v) \leq t \) since \( v \) must be at least as close to \( r \) as \( v \) is. Thus,

\[ d_H(v, v) = d_H(v, r) + d_H(r, v) \leq 2t < d_{\text{min}} \]  

(1.38)

which implies that \( v = v \) and thus the decoding is correct.

Conversely, suppose that \( d_{\text{min}} \leq 2t \). Let \( v \) and \( v' \) be two codewords such that \( d_H(v, v') = d_{\text{min}} \), and let the components of \( r \) be specified as

\[ r_i = \begin{cases} v_i & \text{all } i \text{ such that } v_i = v_i \\ v_i & \text{the first } t \text{ positions with } v_i = v_i \text{ (if } t < d_{\text{min}}) \text{ or } \\ v_i & \text{all positions with } v_i = v_i \text{ (otherwise)} \end{cases} \]  

(1.39)