The Art of Error Correcting Coding

Second Edition

Robert H. Morelos-Zaragoza
San Jose State University, USA

John Wiley & Sons, Ltd

The Art of Error Correcting Coding

The Art of Error Correcting Coding

Second Edition

Robert H. Morelos-Zaragoza
San Jose State University, USA

John Wiley & Sons, Ltd

John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester,
West Sussex PO19 8SQ, England

Telephone (+44) 1243 779777

Email (for orders and customer service enquiries): cs-books@wiley.co.uk
Visit our Home Page on www.wiley.com

Other Wiley Editorial Offices

John Wiley & Sons Inc., 111 River Street, Hoboken, NJ 07030, USA

Jossey-Bass, 989 Market Street, San Francisco, CA 94103-1741, USA

Wiley-VCH Verlag GmbH, Boschstr. 12, D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 42 McDougall Street, Milton, Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01, Jin Xing Distripark, Singapore 129809

John Wiley & Sons Canada Ltd, 6045 Freemont Blvd, Mississauga, ONT, L5R 4J3, Canada

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN-13: 978-0-470-01558-2 (HB)
ISBN-10: 0-470-01558-6 (HB)

Typeset in 10/12pt Times by Laserwords Private Limited, Chennai, India.
Printed and bound in Great Britain by Antony Rowe Ltd, Chippenham, England.
This book is printed on acid-free paper responsibly manufactured from sustainable forestry in which at least two trees are planted for each one used for paper production.

Contents

Preface

The first edition of this book was the result of hundreds of emails from all over the world with questions on the theory and applications of error correcting coding (ECC), from colleagues from both academia and industry. Most of the questions have been from engineers and computer scientists needing to select, implement or simulate a particular coding scheme. The questions were sparked by a popular web site[1] initially set up at Imai Laboratory at the Institute of Industrial Science, University of Tokyo, in early 1995. An important aspect of this text is the absence of theorems and proofs. The approach is to teach basic concepts using simple examples. References to theoretical developments are made when needed. This book is intended to be a reference guide to error correcting coding techniques for graduate students and professionals interested in learning the basic techniques and applications of ECC. Computer programs that implement the basic encoding and decoding algorithms of practical coding schemes are available on a companion web site. This site is referred to as the "ECC web site" throughout the text and is located at:

`http://the-art-of-ecc.com`

This book is unique in that it introduces the basic concepts of error correcting codes with simple illustrative examples. Computer programs written in C language and new Matlab[2] scripts are available on the ECC web site and help illustrate the implementation of basic encoding and decoding algorithms of important coding schemes, such as convolutional codes, Hamming codes, BCH codes, Reed–Solomon codes and turbo codes, and their application in digital communication systems. There is a rich theory of ECC that will be touched upon, by referring to the appropriate material. There are many good books dealing with the theory of ECC, for example, references (Lin and Costello 2005), (MacWilliams and Sloane 1977), (Peterson and Weldon 1972), (Blahut 1984), (Bossert 1999), (Wicker 1995), just to cite a few. Readers may wish to consult them before, during or after going through the material in this book. Each chapter describes, using simple and easy-to-follow numerical examples, the basic concepts of a particular coding or decoding scheme, rather than going into the detail of the theory behind it. Basic analysis tools are given to help in the assessment of the error performance of a particular ECC scheme.

The book deals with the *art* of error correcting coding, in the sense that it addresses the need for selecting, implementing and simulating algorithms for encoding and decoding of codes for error correction and detection. New features of the second edition include additional in-text examples as well as new problems at the end of each chapter, intended for use in a course on ECC. A comprehensive bibliography is included, for readers who wish

[1] `http://www.eccpage.com`

[2] Matlab is a registered trademark of The Mathworks, Inc.

to learn more about the beautiful theory that makes it all work. The book is organized as follows. In Chapter 1, the basic concepts of error correction and coding and decoding techniques are introduced. Chapter 2 deals with important and simple-to-understand families of codes, such as the Hamming, Golay and Reed–Muller codes. In Chapter 3, cyclic codes and the important family of BCH codes are described. Finite-field arithmetic is introduced and basic decoding algorithms, such as Berlekamp–Massey, Euclidean and PGZ, are described, and easy to follow examples are given to understand their operation. Chapter 4 deals with Reed–Solomon codes and errors-and-erasures decoding. A comprehensive treatment of the available algorithms is given, along with examples of their operation. In Chapter 5, binary convolutional codes are introduced. Focus in this chapter is on the understanding of the basic structure of these codes, along with a basic explanation of the Viterbi algorithm with Hamming metrics. Important implementation issues are discussed. In Chapter 6, several techniques for modifying a single code or combining several codes are given and illustrated by simple examples. Chapter 7 deals with soft-decision decoding algorithms, some of which have not yet received attention in the literature, such as a soft-output ordered-statistics decoding algorithm. Moreover, Chapter 8 presents a unique treatment of turbo codes, both parallel concatenated and serial concatenated, and block product codes, from a coding theoretical perspective. In the same chapter, low-density parity-check codes are examined. For all these classes of codes, basic decoding algorithms are described and simple examples are given. Finally, Chapter 9 deals with powerful techniques that combine error correcting coding with digital modulation, and several clever decoding techniques are described.

I would like to express my gratitude to the following persons for inspiring this work. Professor Francisco Garcia Ugalde, Universidad Nacional Autónoma de México, for introducing me to the exciting world of error correcting codes. Parts of this book are based on my Bachelor's thesis under his direction. Professor Edward Bertram, University of Hawaii, for teaching me the basics of abstract algebra. Professor David Muñoz, Instituto Technológico y de Estudios Superiores de Monterrey, México, for his kindness and support. Professors Tadao Kasami, Hiroshima City University, Toru Fujiwara, University of Osaka, and Hideki Imai, University of Tokyo, for supporting my stay as a visiting academic researcher in Japan. Dan Luthi and Advait Mogre, LSI Logic Corporation, for many stimulating discussions and the opportunity to experience the process of putting ideas into silicon. Marc P. C. Fossorier of University of Hawaii for his kind help. My former colleague Dr. Misa Mihaljević of Sony Computer Science Laboratories, for pointing out connections between decoding and cryptoanalysis. I would also like to thank wholeheartedly Dr. Mario Tokoro, President of Sony Computer Science Laboratories, and Professor Ryuji Kohno, Yokohama National University, for making it possible for me to have a fine environment in which to write the first edition of this book. In particular, I want to express my eternal gratitude to Professor Shu Lin of University of California at Davis. I am also grateful to the graduate students of San Jose State University who took my course and helped in designing and testing some of the problems in the second edition.

I dedicate this book to Richard W. Hamming, Claude Shannon and Gustave Solomon, three extraordinary gentlemen who greatly impacted the way people live and work today.

Robert H. Morelos-Zaragoza
San Jose, California, USA

Foreword

In modern digital communication and storage systems design, information theory is becoming increasingly important. The best example of this is the appearance and quick adoption of turbo and block product codes in many practical satellite and wireless communication systems. I am pleased to recommend this new book, authored by Dr. Robert Morelos-Zaragoza, to those who are interested in error correcting codes or have to apply them. The book introduces key concepts of error correcting coding (ECC) in a manner that is easy to understand. The material is logically well structured and presented using simple illustrative examples. This, together with the computer programs available on the web site, is a novel approach to teaching the basic techniques used in the design and application of error correcting codes.

One of the best features of the book is that it provides a natural introduction to the principles and decoding techniques of turbo codes, LDPC codes, and product codes, from an algebraic channel coding perspective. In this context, turbo codes are viewed as punctured product codes. With simple examples, the underlying ideas and structures used in the construction and iterative decoding of product codes are presented in an unparalleled manner. The detailed treatment of various algebraic decoding techniques for the correction of errors and erasures using Reed–Solomon codes is also worth a mention. On the applications of ECC in combined channel coding and digital modulation, or coded modulation, the author does a good job in introducing the basic principles that are used in the construction of several important classes of coded modulation systems.

I believe that practitioner engineers and computer scientists will find this book to be both a good learning tool and a valuable reference. The companion ECC web site is a unique feature that is not found anywhere else. Incidentally, this web site was born in my laboratory at the University of Tokyo in 1995, where Dr. Morelos-Zaragoza worked until June of 1997 and did a very good job as my associate researcher, writing many high-quality papers. Robert is polite, modest and hard-working, and is always friendly. In summary, I strongly recommend *The Art of Error Correcting Coding* as an excellent introductory and reference book on the principles and applications of error correcting codes.

Professor Hideki Imai
The University of Tokyo
Tokyo, Japan

The ECC web site

A companion web site for the book *The Art of Error Correcting Coding* has been set up and is located permanently at the following URL address:

`http://the-art-of-ecc.com`

The **ECC web site** contains computer programs written in both C and Matlab[3] to implement algorithms for encoding and decoding of important families of error correcting codes. New scripts to analyze the performance of error correcting coding schemes have been added. Also, an instructor's solutions manual is now available containing the answers to the problems at the end of each chapter. The web site is maintained by the author, to ensure that the domain name remains unchanged. An important advantage of having a companion web site is that it allows the author to post update notes, new computer programs and simulation results relevant to the contents of the book.

The computer programs in the ECC web site are organized in two ways: by topic and by function. In the topical organization of the programs, the logical structure of the book is closely followed, going from simple syndrome-based decoding of linear block codes to more elaborate algebraic decoding over finite fields of BCH and Reed-Solomon codes, passing through Viterbi decoding of convolutional codes and decoding of combinations and constructions of codes, to iterative decoding of turbo and product codes, belief-propagation decoding of low-density parity-check codes and applications in coded modulation techniques. The functional organization of the programs in the ECC web site is intended for readers who already know exactly what they are looking for. In particular, this classification of the programs is followed with respect to the decoding algorithms.

[3]Matlab is a registered trademark of The Mathworks, Inc.

1

Introduction

The history of error correcting coding (ECC) started with the introduction of the Hamming codes (Hamming 1974), at or about the same time as the seminal work of Shannon (1948). Shortly after, Golay codes were invented (Golay 1974). These two first classes of codes are optimal, and will be defined in a subsequent section.

Figure 1.1 shows the block diagram of a canonical digital communications/storage system. This is the famous *Figure 1* in most books on the theory of ECC and digital communications (Benedetto and Biglieri 1999). The information source and destination will include any source coding scheme matched to the nature of the information. The ECC encoder takes as input the information symbols from the source and adds redundant symbols to it, so that most of the errors – introduced in the process of modulating a signal, transmitting it over a noisy medium and demodulating it – can be corrected (Massey 1984; McEliece 1977; Moon 2005).

Usually, the channel is assumed to be such that samples of an additive noise process are added to the modulated symbols (in their equivalent complex baseband representation). The noise samples are assumed to be independent from the source symbols. This model is relatively easy to track mathematically and includes additive white Gaussian noise (AWGN) channels, flat Rayleigh fading channels, and binary symmetric channels (BSC). The case of frequency-selective channels can also be included, as techniques such as spread-spectrum and multicarrier modulation (MCM) effectively transform them into either AWGN channels or flat Rayleigh fading channels.

At the receiver end, the ECC decoder utilizes the redundant symbols and their relationship with the information symbols in order to correct channel errors. In the case of error detection, the ECC decoder can be best thought of as a reencoder of the received information, followed by a check that the redundant symbols generated are the same as those received.

In classical ECC theory, the combination of modulation, noisy medium and demodulation was modeled as a *discrete memoryless channel* with input $\bar{v}$ and output $\bar{r}$. An example of this is binary transmission over an AWGN channel, which is modeled as a *BSC*. This is illustrated in Figure 1.2. The BSC has a probability of channel error p – or transition probability – equal to the probability of a bit error for binary

The Art of Error Correcting Coding Second Edition Robert H. Morelos-Zaragoza

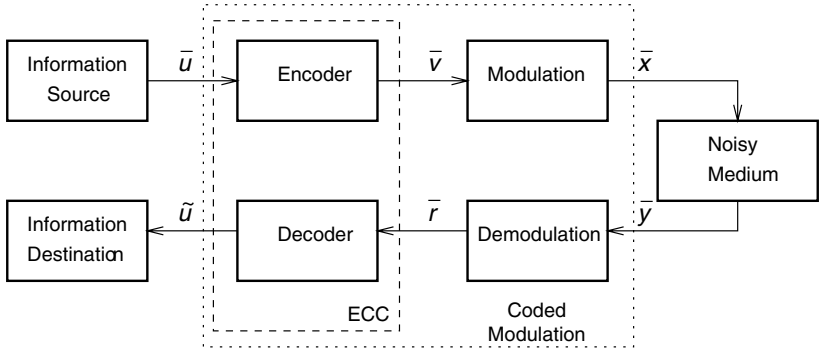

Figure 1.1 A canonical digital communications system.

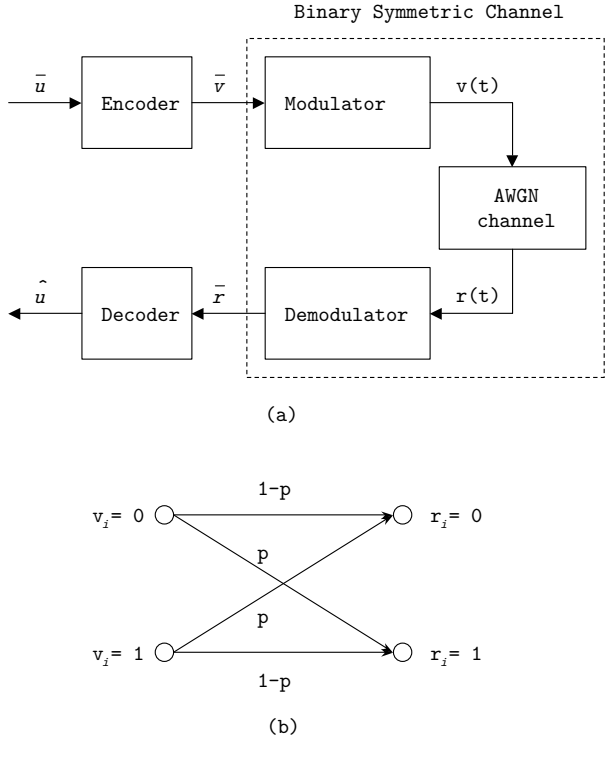

Figure 1.2 A binary communication system over an AWGN channel and corresponding BSC.

signaling over an AWGN channel,

$$p = Q\left(\sqrt{\frac{2E_b}{N_0}}\right), \tag{1.1}$$

where E_b/N_0 is the energy-per-bit-to-noise ratio – also referred to as the bit signal-to-noise ratio (SNR) or SNR per bit – and

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-z^2/2}\, dz, \tag{1.2}$$

is the Gaussian Q-function. In terms of the *complementary error function*, the Q-function can be written as

$$Q(x) = \frac{1}{2}\ \text{erfc}\left(\frac{x}{\sqrt{2}}\right). \tag{1.3}$$

Equation (1.2) is useful in analytical derivations and Equation (1.3) is used in the computation with C programs or Matlab scripts of performance bounds and approximations.

Massey (1974) suggested considering ECC and modulation as a single entity, known in modern literature as *coded modulation*. This approach provides a higher efficiency and coding gain[1] rather than the serial concatenation of ECC and modulation, by joint design of codes and signal constellations. Several methods of combining coding and modulation are covered in this book, including the following: trellis-coded modulation (TCM) (Ungerboeck 1982) and multilevel coded modulation (MCM) (Imai and Hirakawa 1977). In a coded modulation system, the (soft-decision) channel outputs are directly processed by the decoder. In contrast, in a classical ECC system, the hard-decision bits from the demodulator are fed to a binary decoder.

Codes can be combined in several ways. An example of *serial concatenation* (that is, concatenation in the classical sense) is the following. For years, the most popular concatenated ECC scheme has been the combination of an outer Reed–Solomon (RS) code, through intermediate interleaving, and an inner binary convolutional code. This scheme has been used in numerous applications, ranging from space communications to digital broadcasting of high definition television. The basic idea is that the soft-decision decoder of the convolutional code produces bursts of errors that can be broken into smaller pieces by the deinterleaving process and handled effectively by the RS decoder. RS codes are nonbinary codes that work with symbols composed of several bits, and can deal with multiple bursts of errors. Serial concatenation has the advantage that it requires two separate decoders, one for the inner code and one for the outer code, instead of a single but very complex decoder for the overall code.

This book examines these types of ECC systems. First, basic code constructions and their decoding algorithms, in the Hamming space (that is, dealing with bits), are presented. In the second part of the book, important soft-decision decoding (SDD) algorithms for binary transmission are introduced. These algorithms work over the Euclidean space and achieve a reduction in the required transmitted power per bit of at least 2 dB, compared with Hamming-space (hard-decision) decoders. Several kinds of soft-decision decoders are

[1]Coding gain is defined as the difference in SNR between the coded system and an uncoded system with the same bit rate.

considered, with attention given to their algorithmic aspects (the "how" they work), rather than to their theoretical aspects (the 'why' they work). Finally, combinations of codes and interleaving for iterative decoding and of coding and modulation for bandwidth-efficient transmission are the topic of the last part of the book.

1.1 Error correcting coding: Basic concepts

All error correcting codes are based on the same basic principle: *redundancy* is added to information in order to correct any errors that may occur in the process of transmission or storage. In a basic (and practical) form, redundant symbols are appended to information symbols to obtain a coded sequence or *code word*. For the purpose of illustration, a code word obtained by encoding with a *block code* is shown in Figure 1.3. Such an encoding is said to be *systematic*. Systematic encoding means that the information symbols always appear in the first (leftmost) k positions of a code word. The remaining (rightmost) $n-k$ symbols in a code word are a function of the information symbols, and provide redundancy that can be used for error correction and/or detection purposes[2]. The set of all code sequences is called an *error correcting code*, and will henceforth be denoted by C.

1.1.1 Block codes and convolutional codes

According to the manner in which redundancy is added to messages, ECC can be divided into two classes: block and convolutional. Both types of coding schemes have found practical applications. Historically, convolutional codes have been preferred, apparently because of the availability of the soft-decision Viterbi decoding algorithm and the belief over many years that block codes could not be efficiently decoded with soft-decisions. However, recent developments in the theory and design of SDD algorithms for linear block codes have helped to dispel this belief. Moreover, the best ECC known to date remain block codes (long irregular low-density parity-check (LDPC) codes).

Block codes process the information on a block-by-block basis, treating each block of information bits independently from others. In other words, block coding is a memoryless operation, in the sense that code words are independent from each other. In contrast, the output of a convolutional encoder depends not only on the current input information, but also on previous inputs or outputs, either on a block-by-block or a bit-by-bit basis. For simplicity of exposition, we begin with a study of the structural properties of block codes. Many of these properties are common to both types of codes.

It should be noted that block codes have, in fact, memory, when encoding is thought of as a bit-by-bit process and within a code word. Most recently, the difference between block

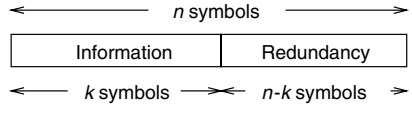

Figure 1.3 A systematic block encoding for error correction.

[2]Note: Without loss of generality, the order of the information and redundancy can be reversed (i.e., the first $n-k$ positions in a code word for redundant symbols and the remaining k positions for information symbols).

and convolutional codes has become less and less well defined, especially after recent advances in the understanding of the trellis structure of block codes, and the tail-biting structure of some convolutional codes. Indeed, colleagues working on convolutional codes sometimes refer to block codes as "codes with time-varying trellis structure." Similarly, researchers working with block codes may consider convolutional codes as "codes with a regular trellis structure."

1.1.2 Hamming distance, Hamming spheres and error correcting capability

Consider an error correcting code C with binary elements. As mentioned above, block codes are considered for simplicity of exposition. In order to achieve error correcting capabilities, not all the 2^n possible binary vectors of length n are allowed to be transmitted. Instead, C is a subset of the n-dimensional binary vector space $V_2 = \{0, 1\}^n$, such that its elements are as far apart as possible.

Consider two vectors $\bar{x}_1 = (x_{1,0}, x_{1,1}, \ldots, x_{1,n-1})$ and $\bar{x}_2 = (x_{2,0}, x_{2,1}, \ldots, x_{2,n-1})$ in V_2. Then the *Hamming distance* between $\bar{x}_1$ and $\bar{x}_2$, denoted $d_H(\bar{x}_1, \bar{x}_2)$, is defined as the number of elements in which the vectors differ,

$$d_H(\bar{x}_1, \bar{x}_2) \stackrel{\Delta}{=} \left| \left\{ i : x_{1,i} \neq x_{2,i}, \quad 0 \leq i < n \right\} \right| = \sum_{i=0}^{n-1} x_{1,i} \oplus x_{2,i}, \tag{1.4}$$

where $|A|$ denotes the number of elements in (or the cardinality of) a set A and $\oplus$ denotes addition modulo-2 (exclusive-OR).

Given a code C, its *minimum Hamming distance*, $d_{\min}$, is defined as the minimum Hamming distance among all possible distinct pairs of code words in C,

$$d_{\min} = \min_{\bar{v}_1, \bar{v}_2 \in C} \{ d_H(\bar{v}_1, \bar{v}_2) | \bar{v}_1 \neq \bar{v}_2 \}. \tag{1.5}$$

Throughout the book, the array $(n, k, d_{\min})$ is used to denote the parameters of a block code of length n, that encodes messages of length k bits and has a minimum Hamming distance $d_{\min}$. The assumption is made that the size of the code is $|C| = 2^k$.

Example 1.1.1 *The simplest error correcting code is a binary repetition code of length 3. It repeats each bit three times, so that a "0" is encoded onto the vector (000) and a "1" onto the vector (111). Since the two code words differ in all three positions, the Hamming distance between them is equal to three. Figure 1.4 is a pictorial representation of this code. The three-dimensional binary space corresponds to the set of* $2^3 = 8$ *vertices of the three-dimensional unit-volume cube. The Hamming distance between code words (000) and (111) equals the number of edges in a path between them. This is equivalent to the number of coordinates that one needs to change to convert (000) into (111), or vice versa. Thus* $d_H((000), (111)) = 3$. *There are only two code words in this case, as a result,* $d_{\min} = 3$.

The binary vector space V_2 is also known as a *Hamming space*. Let $\bar{v}$ denote a code word of an error correcting code C. A *Hamming sphere* $S_t(\bar{v})$, of *radius* t and centered around $\bar{v}$, is the set of vectors in V_2 at a distance less than or equal to t from the center $\bar{v}$,

$$S_t(\bar{v}) = \{ \bar{x} \in V_2 | d_H(\bar{x}, \bar{v}) \leq t \}. \tag{1.6}$$

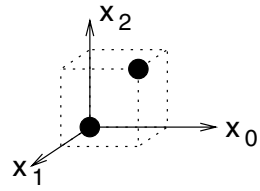

Figure 1.4 A (3,1,3) repetition code in a three-dimensional binary vector space.

Note that the size of (or the number of code words in) $S_t(\bar{v})$ is given by the following expression

$$\left|S_t(\bar{v})\right| = \sum_{i=0}^{t} \binom{n}{i}. \tag{1.7}$$

Example 1.1.2 *Figure 1.5 shows the Hamming spheres of radius* $t = 1$ *around the code words of the* $(3, 1, 3)$ *binary repetition code.*

Note that the Hamming spheres for this code are disjoint, that is, there is no vector in V_2 *(or vertex in the unit-volume three-dimensional cube) that belongs to both* $S_1(000)$ *and* $S_1(111)$. *As a result, if there is a change in any one position of a code word* $\bar{v}$, *then the resulting vector will still lie inside a Hamming sphere centered at* $\bar{v}$. *This concept is the basis of understanding and defining the error correcting capability of a code* C.

The *error correcting capability*, t, of a code C is the largest radius of Hamming spheres $S_t(\bar{v})$ around all the code words $\bar{v} \in C$, such that for all different pairs $\bar{v}_i, \bar{v}_j \in C$, the corresponding Hamming spheres are disjoint, that is,

$$t = \max_{\bar{v}_i, \bar{v}_j \in C} \left\{ \ell | S_\ell(\bar{v}_i) \cap S_\ell(\bar{v}_j) = \emptyset, \quad \bar{v}_i \neq \bar{v}_j \right\}. \tag{1.8}$$

In terms of the minimum distance of C, $d_{\min}$, an equivalent and more common definition is

$$t = \lfloor (d_{\min} - 1)/2 \rfloor, \tag{1.9}$$

where $\lfloor x \rfloor$ denotes the largest integer less than or equal to x.

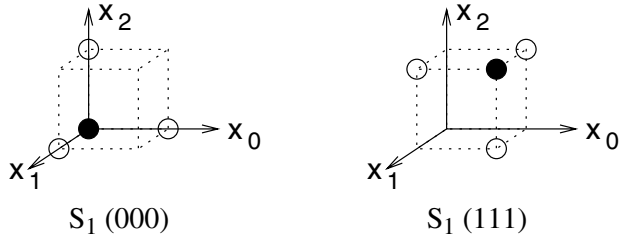

Figure 1.5 Hamming spheres of radius $t = 1$ around the code words of the (3,1,3) binary repetition code.

Note that in order to compute the minimum distance $d_{\min}$ of a block code C, in accordance with Equation (1.5), a total of $2^{k-1}(2^k - 1)$ distances between distinct pairs of code words are needed. This is practically impossible even for codes of relatively modest size, say, $k = 50$ (for which approximately 2^{99} code word pairs need to be examined). One of the advantages of *linear* block codes is that the computation of $d_{\min}$ requires to know the *Hamming weight* of all $2^k - 1$ nonzero code words.

1.2 Linear block codes

As mentioned above, finding a good code means finding a subset of V_2 with elements as far apart as possible. This is very difficult. In addition, even when such a set is found, there is still the problem of *how to assign code words to information messages*.

Linear codes are *vector subspaces* of V_2. This means that encoding can be accomplished by matrix multiplications. In terms of digital circuitry, simple encoders can be built using exclusive-OR gates, AND gates and D flip-flops. In this chapter, the binary vector space operations of sum and multiplication are meant to be the output of exclusive-OR (or modulo 2 addition) and AND gates, respectively. The tables of addition and multiplication for the binary elements in $\{0, 1\}$ are as follows:

a	b	$a + b$	$a \cdot b$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

which are seen to correspond to the outputs of a binary exclusive-OR gate and an AND gate, respectively.

1.2.1 Generator and parity-check matrices

Let C denote a binary linear $(n, k, d_{\min})$ code. Now, C is a k-dimensional vector subspace, and therefore it has a *basis*, say $\{\bar{v}_0, \bar{v}_1, \ldots, \bar{v}_{k-1}\}$, such that any code word $\bar{v} \in C$ can be represented as a linear combination of the elements on the basis of:

$$\bar{v} = u_0\bar{v}_0 + u_1\bar{v}_1 + \cdots + u_{k-1}\bar{v}_{k-1}, \tag{1.10}$$

where $u_i \in \{0, 1\}$, $1 \leq i < k$. Equation (1.10) can be written in terms of a *generator matrix* G and a *message vector*, $\bar{u} = (u_0, u_1, \ldots, u_{k-1})$, as follows:

$$\bar{v} = \bar{u}G, \tag{1.11}$$

where

$$G = \begin{pmatrix} \bar{v}_0 \\ \bar{v}_1 \\ \vdots \\ \bar{v}_{k-1} \end{pmatrix} = \begin{pmatrix} v_{0,0} & v_{0,1} & \cdots & v_{0,n-1} \\ v_{1,0} & v_{1,1} & \cdots & v_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ v_{k-1,0} & v_{k-1,1} & \cdots & v_{k-1,n-1} \end{pmatrix}. \tag{1.12}$$

Due to the fact that C is a k-dimensional vector space in V_2, there is an $(n-k)$-dimensional *dual space* $C^\top$, generated by the rows of a matrix H, called the *parity-check matrix*, such that $GH^\top = 0$, where $H^\top$ denotes the transpose of H. In particular, note that for any code word $\bar{v} \in C$,

$$\bar{v}H^\top = \bar{0}. \tag{1.13}$$

Equation (1.13) is of fundamental importance in *decoding of linear codes*, as will be shown in section 1.3.2.

A linear code $C^\perp$ that is generated by H is a binary linear $(n, n-k, d_{\min}^\perp)$ code, called the *dual code* of C.

1.2.2 The weight is the distance

As mentioned in section 1.1.2, a nice feature of linear codes is that computing the minimum distance of the code amounts to computing the minimum Hamming weight of its nonzero code words. In this section, this fact is shown. The *Hamming weight*, $\mathrm{wt}_H(\bar{x})$, of a vector $\bar{x} = (x_0, x_1, \ldots, x_{n-1}) \in V_2$ is defined as the number of nonzero elements in $\bar{x}$, which can be expressed as the sum

$$\mathrm{wt}_H(\bar{x}) = \sum_{i=0}^{n-1} x_i. \tag{1.14}$$

From the definition of the Hamming distance, it is easy to see that $\mathrm{wt}_H(\bar{x}) = d_H(\bar{x}, \bar{0})$. For a binary linear code C, note that the distance

$$d_H(\bar{v}_1, \bar{v}_2) = d_H(\bar{v}_1 + \bar{v}_2, \bar{0}) = \mathrm{wt}_H(\bar{v}_1 + \bar{v}_2) = \mathrm{wt}_H(\bar{v}_3), \tag{1.15}$$

where, by linearity, $\bar{v}_1 + \bar{v}_2 = \bar{v}_3 \in C$. As a consequence, the minimum distance of C can be computed by finding the minimum Hamming weight among the $2^k - 1$ nonzero code words. This is simpler than the brute force search among all the pairs of code words, although still a considerable task even for codes of modest size (or dimension k).

1.3 Encoding and decoding of linear block codes

1.3.1 Encoding with G and H

Equation (1.11) gives an encoding rule for linear block codes that can be implemented in a straightforward way. If encoding is to be systematic, then the generator matrix G of a linear block $(n, k, d_{\min})$ code C can be brought to a *systematic form*, G_{sys}, by elementary row operations and/or column permutations. G_{sys} is composed of two submatrices: The k-by-k identity matrix, denoted I_k, and a k-by-$(n-k)$ *parity submatrix* P, such that

$$G_{\mathrm{sys}} = \left(I_k | P\right), \tag{1.16}$$

where

$$P = \begin{pmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,n-k-1} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,n-k-1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} \end{pmatrix}. \tag{1.17}$$

Since $GH^\top = 0_{k,n-k}$, where $0_{k,n-k}$ denotes the k-by-$(n-k)$ all-zero matrix, it follows that the systematic form, H_{sys}, of the parity-check matrix is

$$H_{\text{sys}} = \left(P^\top | I_{n-k}\right). \tag{1.18}$$

Example 1.3.1 *Consider a binary linear* $(4, 2, 2)$ *code with generator matrix*

$$G = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

To bring G into systematic form, permute (exchange) the second and fourth columns and obtain

$$G_{\text{sys}} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}.$$

Thus, the parity-check submatrix is given by

$$P = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}.$$

It is interesting to note that in this case, the relation $P = P^\top$ *holds*[3]*. From (1.18) it follows that the systematic form of the parity-check matrix is*

$$H_{\text{sys}} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

In the following, let $\bar{u} = (u_0, u_1, \ldots, u_{k-1})$ denote an information message to be encoded and $\bar{v} = (v_0, v_1, \ldots, v_{n-1})$ the corresponding code word in C.

If the parameters of C are such that $k < (n-k)$, or equivalently the *code rate* $k/n < 1/2$, then encoding with the generator matrix is the most economical. The cost considered here is in terms of binary operations. In such a case

$$\bar{v} = \bar{u} G_{\text{sys}} = (\bar{u}, \bar{v}_p), \tag{1.19}$$

where $\bar{v}_p = \bar{u} P = (v_k, v_{k+1}, \ldots, v_{n-1})$ represents the parity-check (redundant) part of the code word.

However, if $k > (n-k)$, or $k/n > 1/2$, then alternative encoding with the parity-check matrix H requires less number of computations. In this case, we have encoding based on Equation (1.13), $(\bar{u}, \bar{v}_p)H^\top = 0$, such that the $(n-k)$ parity-check positions $v_k, v_{k+1}, \ldots, v_{n-1}$ are obtained as follows:

$$v_j = u_0 p_{0,j} + u_1 p_{1,j} + \cdots + u_{k-1} p_{k-1,j}, \qquad k \le j < n. \tag{1.20}$$

Stated in other terms, the systematic form of a parity-check matrix of a linear code has as entries of its rows the coefficients of the parity-check equations, from which the values of the redundant positions are obtained. This fact will be used when LDPC codes are presented, in Section 8.3.

[3] In this case, the code in question is referred to as a *self-dual code*. See also Section 2.2.3.

Example 1.3.2 *Consider the binary linear* $(4, 2, 2)$ *code from Example 1.3.1. Let messages and code words be denoted by* $\bar{u} = (u_0, u_1)$ *and* $\bar{v} = (v_0, v_1, v_2, v_3)$*, respectively. From (1.20) it follows that*

$$\begin{aligned} v_2 &= u_0 + u_1 \\ v_3 &= u_0 \end{aligned}$$

The correspondence between the $2^2 = 4$ *two-bit messages and code words is as follows:*

$$\begin{aligned} (00) &\mapsto (0000) \\ (01) &\mapsto (0110) \\ (10) &\mapsto (1011) \\ (11) &\mapsto (1101) \end{aligned} \tag{1.21}$$

1.3.2 Standard array decoding

In this section, a decoding procedure is presented that finds the closest code word $\bar{v}$ to a received noisy word $\bar{r} = \bar{v} + \bar{e}$. The *error vector* $\bar{e} \in \{0, 1\}^n$ is produced by a BSC, as depicted in Figure 1.6. It is assumed that the crossover probability (or BSC parameter) p is such that $p < 1/2$.

As shown in Table 1.1 a *standard array* (Slepian 1956) for a binary linear $(n, k, d_{\min})$ code C is a table of all possible received vectors $\bar{r}$ arranged in such a way that the closest code word $\bar{v}$ to $\bar{r}$ can be read out. The standard array contains 2^{n-k} rows and $2^k + 1$ columns. The entries of the rightmost 2^k columns of the array contain all the vectors in $V_2 = \{0, 1\}^n$.

In order to describe the decoding procedure, the concept of *syndrome* is needed. The syndrome of a word in V_2 is defined from Equation (1.13) as

$$\bar{s} = \bar{r} H^{\top}, \tag{1.22}$$

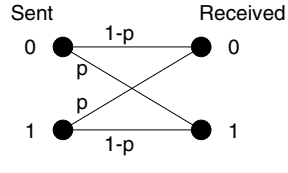

Figure 1.6 A binary symmetric channel model.

Table 1.1 The standard array of a binary linear block code.

$\bar{s}$	$\bar{u}_0 = \bar{0}$	$\bar{u}_2$	$\cdots$	$\bar{u}_{k-1}$
$\bar{0}$	$\bar{v}_0 = \bar{0}$	$\bar{v}_1$	$\cdots$	$\bar{v}_{2^k-1}$
$\bar{s}_1$	$\bar{e}_1$	$\bar{e}_1 + \bar{v}_1$	$\cdots$	$\bar{e}_1 + \bar{v}_{2^k-1}$
$\bar{s}_2$	$\bar{e}_2$	$\bar{e}_2 + \bar{v}_1$	$\cdots$	$\bar{e}_2 + \bar{v}_{2^k-1}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\bar{s}_{2^{n-k}-1}$	$\bar{e}_{2^{n-k}-1}$	$\bar{e}_{2^{n-k}-1} + \bar{v}_1$	$\cdots$	$\bar{e}_{2^{n-k}-1} + \bar{v}_{2^k-1}$

where H is the parity-check matrix of C. That $\bar{s}$ is indeed a set of symptoms that indicate errors is shown as follows. Suppose that a code word $\bar{v} \in C$ is transmitted over a BSC and received as $\bar{r} = \bar{v} + \bar{e}$. The syndrome of $\bar{r}$ is

$$\bar{s} = \bar{r}H^{\top} = (\bar{v} + \bar{e})H^{\top} = \bar{e}H^{\top}, \tag{1.23}$$

where to obtain the last equality Equation (1.13) has been used. Therefore, the computation of the syndrome can be thought of as a *linear transformation* of an error vector.

Standard array construction procedure

1. As the first row, in the positions corresponding to the 2^k rightmost columns, enter all the code words of C, beginning with the all-zero code word in the leftmost position. In the position corresponding to the first column, enter the all-zero syndrome. Let $j = 0$.

2. Let $j = j + 1$. Find the smallest Hamming weight word $\bar{e}_j$ in V_2, not in C, and not included in previous rows. The corresponding syndrome $\bar{s}_j = \bar{e}_j H^{\top}$ is the first (rightmost) entry of the row. The 2^k remaining entries in that row are obtained by adding $\bar{e}_j$ to all the entries in the first row (the code words of C).

3. Repeat the previous step until all vectors in V_2 are included in the array. Equivalently, let $j = j + 1$. If $j < 2^{n-k}$, then repeat previous step, otherwise stop.

Example 1.3.3 *The standard array of the binary linear* $(4, 2, 2)$ *code is the following:*

$\bar{s}$	00	01	10	11
00	0000	0110	1011	1101
11	1000	1110	0011	0101
10	0100	0010	1111	1001
01	0001	0111	1010	1100

Decoding with the standard array proceeds as follows. Let $\bar{r} = \bar{v} + \bar{e}$ be the received word. Find the word in the array and output as decoded message $\bar{u}$ the header of the column in which $\bar{r}$ lies. Conceptually, this process requires storing the entire array and matching the received word to an entry in the array.

However, a simplified decoding procedure can be found by noticing that every row in the array has the same syndrome. Each row of the array, denoted Row_i, with $0 \leq i < 2^{n-k}$, a *coset* of C, is such that $\text{Row}_i = \left\{\bar{e}_i + \bar{v} \middle| \bar{v} \in C\right\}$. The vector $\bar{e}_i$ is known as the *coset leader*.

The syndrome of the elements in the i-th row is given by

$$\bar{s}_i = (\bar{e}_i + \bar{v})H^{\top} = \bar{e}_i H^{\top}, \tag{1.24}$$

which is *independent* of the particular choice of $\bar{v} \in C$. The simplified decoding procedure is: compute the syndrome of the received word $\bar{r} = \bar{e}_{i'} + \bar{v}$,

$$\bar{s}_{i'} = (\bar{e}_{i'} + \bar{v})\, H^{\top} = \bar{e}_{i'} H^{\top},$$

and find $\bar{s}_{i'}$ in the leftmost column of the standard array. Then read out the value of $\bar{e}_{i'}$, from the second column, and add it to the received word to obtain the closest code word $\bar{v}' \in C$ to $\bar{r}$. Therefore instead of $n \times 2^n$ bits, standard array decoding can be implemented with an array of $n \times 2^{n-k}$ bits.

Example 1.3.4 *Consider again the binary linear* $(4, 2, 2)$ *code from Example 1.3.1. Suppose that the code word* $\bar{v} = (0110)$ *is transmitted and that* $\bar{r} = (0010)$ *is received. Then the syndrome is*

$$\bar{s} = \bar{r}H^\top = (0010)\begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \end{pmatrix}.$$

From the standard array of the code, the corresponding coset leader $\bar{e}' = (0100)$ *is found, and therefore, the estimated code word is* $\bar{v}' = \bar{r} + \bar{e}' = (0010) + (0100) = (0110)$. *One error has been corrected! This may sound strange, since the minimum distance of the code is only two, and thus according to (1.9) single error, correction is impossible. However, this can be explained by looking again at the standard array of this code (Example 1.3.3 above). Note that the third row of the array contains two distinct binary vectors of weight one. This means that only three out of a total of four single-error patterns can be corrected. The error above is one of those correctable single-error patterns.*

It turns out that this $(4, 2, 2)$ *code is the simplest instance of a linear* Linear unequal error protection (LUEP) *code (van Gils 1983; Wolf and Viterbi 1996). This LUEP code has a separation vector* $\bar{s} = (3, 2)$, *which means that the minimum distance between any two code words for which the* first *message bit differs is at least three, and that for the* second *message bit is at least two.*

If encoding is *systematic*, then the above procedure gives the estimated message $\bar{u}'$ in the first k positions of $\bar{v}'$. This is a plausible reason for having a systematic encoding.

1.3.3 Hamming spheres, decoding regions and the standard array

The standard array is also a convenient way of understanding the concept of Hamming sphere and error correcting capability of a linear code C, introduced in Section 1.1.2.

By construction, note that the 2^k rightmost columns of the standard array, denoted Col_j, for $1 \le j \le 2^k$, contain a code word $\bar{v}_j \in C$ and a set of $2^{n-k} - 1$ words at the smallest Hamming distance from $\bar{v}_j$, that is,

$$\mathrm{Col}_j = \left\{ \bar{v}_j + \bar{e}_i \middle| \bar{e}_i \in \mathrm{Row}_i, \quad 0 \le i < 2^{n-k} \right\}. \tag{1.25}$$

The sets Col_j are known as the *decoding regions*, in the Hamming space, around each code word $\bar{v}_j \in C$, for $0 \le j \le 2^k - 1$. This is to say that if code word $\bar{v}_j \in C$ is transmitted over a BSC and the received word $\bar{r}$ lies in the set Col_j, then it will be successfully decoded into $\bar{v}_j$.

Hamming bound

The set Col_j and the error correcting capability t of code C are related by the Hamming sphere $S_t(\bar{v}_j)$: A binary linear $(n, k, d_{\min})$ code C has decoding regions Col_j that properly contain Hamming spheres $S_t(\bar{v}_j)$, that is, $S_t(\bar{v}_j) \subseteq \mathrm{Col}_j$.