

The Theory of Error-Correcting Codes

Part I

F.J. MacWilliams
N.J.A. Sloane

The Theory of Error-Correcting Codes

Part I

F.J. MacWilliams
N.J.A. Sloane

Bell Laboratories
Murray Hill
NJ 07974
U.S.A.

上海交通大学图书馆

#E 95650



1977

north-holland publishing company
amsterdam · new york · oxford

© North-Holland Publishing Company 1977

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owner.

LCC Number: 76-41296

ISBN: 0 444 85009 0

Published by:

North-Holland Publishing Company

Amsterdam · New York · Oxford

Sole distributors for the U.S.A. and Canada:

Elsevier/North-Holland Inc.

52 Vanderbilt Avenue

New York, NY 10017

Library of Congress Cataloging in Publication Data

MacWilliams, Florence Jessie, 1917-

The theory of error-correcting codes.

1. Error-correcting codes (Information theory)

I. Sloane, Neil James Alexander, 1939- joint author.

II. Title.

QA268.M3 519.4 76-41296

ISBN 0 444 85009 0

Preface

Coding theory began in the late 1940's with the work of Golay, Hamming and Shannon. Although it has its origins in an engineering problem, the subject has developed by using more and more sophisticated mathematical techniques. It is our goal to present the theory of error-correcting codes in a simple, easily understandable manner, and yet also to cover all the important aspects of the subject. Thus the reader will find both the simpler families of codes—for example, Hamming, BCH, cyclic and Reed-Muller codes—discussed in some detail, together with encoding and decoding methods, as well as more advanced topics such as quadratic residue, Golay, Goppa, alternant, Kerdock, Preparata, and self-dual codes and association schemes.

Our treatment of bounds on the size of a code is similarly thorough. We discuss both the simpler results—the sphere-packing, Plotkin, Elias and Garshamov bounds—as well as the very powerful linear programming method and the McEliece-Rodemich-Rumsey-Welch bound, the best asymptotic result known. An appendix gives tables of bounds and of the best codes presently known of length up to 512.

Having two authors has helped to keep things simple: by the time we both understand a chapter, it is usually transparent. Therefore this book can be used both by the beginner and by the expert, as an introductory textbook and as a reference book, and both by the engineer and the mathematician. Of course this has not resulted in a thin book, and so we suggest the following menus:

An elementary first course on coding theory for mathematicians: Ch. 1, Ch. 2 (§6 up to Theorem 22), Ch. 3, Ch. 4 (§§1–5), Ch. 5 (to Problem 5), Ch. 7 (not §§7, 8), Ch. 8 (§§1–3), Ch. 9 (§§1, 4), Ch. 12 (§8), Ch. 13 (§§1–3), Ch. 14 (§§1–3).

A second course for mathematicians: Ch. 2 (§§1–6, 8), Ch. 4 (§§6, 7 and part of 8), Ch. 5 (to Problem 6, and §§3, 4, 5, 7), Ch. 6 (§§1–3, 10, omitting the

proof of Theorem 33), Ch. 8 (§§5, 6), Ch. 9 (§§2, 3, 5), Ch. 10 (§§1-5, 11), Ch. 11, Ch. 13 (§§4, 5, 9), Ch. 16 (§§1-6), Ch. 17 (§7, up to Theorem 35), Ch. 19 (§§1-3).

An elementary first course on coding theory for engineers: Ch. 1, Ch. 3, Ch. 4 (§§1-5), Ch. 5 (to Problem 5), Ch. 7 (not §7), Ch. 9 (§§1, 4, 6), Ch. 10 (§§1, 2, 5, 6, 7, 10), Ch. 13 (§§1-3, 6, 7), Ch. 14 (§§1, 2, 4).

A second course for engineers: Ch. 2 (§§1-6), Ch. 8 (§§1-3, 5, 6), Ch. 9 (§§2, 3, 5), Ch. 10 (§11), Ch. 12 (§§1-3, 8, 9), Ch. 16 (§§1, 2, 4, 6, 9), Ch. 17 (§7, up to Theorem 35).

There is then a lot of rich food left for an advanced course: the rest of Chapters 2, 6, 11 and 14, followed by Chapters 15, 18, 19, 20 and 21 – a feast!

The following are the principal codes discussed:

Alternant, Ch. 12;

BCH, Ch. 3, §§1, 3; Ch. 7, §6; Ch. 8, §5; Ch. 9; Ch. 21, §8;

Chien-Choy generalized BCH, Ch. 12, §7;

Concatenated, Ch. 10, §11; Ch. 18, §§5, 8;

Conference matrix, Ch. 2, §4;

Cyclic, Ch. 7, Ch. 8;

Delsarte-Goethals, Ch. 15, §5;

Difference-set cyclic, Ch. 13, §8;

Double circulant and quasi-cyclic, Ch. 16, §§6-8;

Euclidean and projective geometry, Ch. 13, §8;

Goethals generalized Preparata, Ch. 15, §7;

Golay (binary), Ch. 2, §6; Ch. 16, §2; Ch. 20;

Golay (ternary), Ch. 16, §2; Ch. 20;

Goppa, Ch. 12, §§3-5;

Hadamard, Ch. 2, §3;

Hamming, Ch. 1, §7, Ch. 7, §3 and Problem 8;

Irreducible or minimal cyclic, Ch. 8, §§3, 4;

Justesen, Ch. 10, §11;

Kerdock, Ch. 2, §8; Ch. 15, §5;

Maximal distance separable, Ch. 11;

Nordstrom-Robinson, Ch. 2, §8; Ch. 15, §§5, 6;

Pless symmetry, Ch. 16, §8;

Preparata, Ch. 2, §8; Ch. 15, §6; Ch. 18, §7.3;

Product, Ch. 18, §§2-6;

Quadratic residue, Ch. 16;

Redundant residue, Ch. 10, §9;

Reed-Muller, Ch. 1, §9; Chs. 13-15;

Reed-Solomon, Ch. 10;

Self-dual, Ch. 19;
Single-error-correcting nonlinear, Ch. 2, §7; Ch. 18, §7.3;
Srivastava, Ch. 12, §6.

Encoding methods are given for:

Linear codes, Ch. 1, §2;
Cyclic codes, Ch. 7, §8;
Reed-Solomon codes, Ch. 10, §7;
Reed-Muller codes, Ch. 13, §§6, 7; Ch. 14, §4.

Decoding methods are given for:

Linear codes, Ch. 1, §§3, 4;
Hamming codes, Ch. 1, §7;
BCH codes, Ch. 3, §3; Ch. 9, §6; Ch. 12, §9;
Reed-Solomon codes, Ch. 10, §10;
Alternant (including BCH, Goppa, Srivastava and Chien-Choy generalized BCH codes) Ch. 12, §9;
Quadratic residue codes, Ch. 16, §9;
Cyclic codes, Ch. 16, §9,

while other decoding methods are mentioned in the notes to Ch. 16.

When reading the book, keep in mind this piece of advice, which should be given in every preface: if you get stuck on a section, skip it, but keep reading! Don't hesitate to skip the proof of a theorem: we often do. Starred sections are difficult or dull, and can be omitted on the first (or even second) reading.

The book ends with an extensive bibliography. Because coding theory overlaps with so many other subjects (computers, digital systems, group theory, number theory, the design of experiments, etc.) relevant papers may be found almost anywhere in the scientific literature. Unfortunately this means that the usual indexing and reviewing journals are not always helpful. We have therefore felt an obligation to give a fairly comprehensive bibliography. The notes at the ends of the chapters give sources for the theorems, problems and tables, as well as small bibliographies for some of the topics covered (or not covered) in the chapter.

Only *block* codes for correcting *random* errors are discussed; we say little about codes for correcting other kinds of errors (bursts or transpositions) or about variable length codes, convolutional codes or source codes (see the

Notes to Ch. 1). Furthermore we have often considered only *binary* codes, which makes the theory a lot simpler. Most writers take the opposite point of view: they think in binary but publish their results over arbitrary fields.

There are a few topics which were included in the original plan for the book but have been reluctantly omitted for reasons of space:

(i) Gray codes and snake-in-the-box codes – see Adelson et al. [5, 6], Buchner [210], Caviar [253], Chien et al. [290], Cohn [299], Danzer and Klee [328], Davies [335], Douglas [382, 383], Even [413], Flores [432], Gardner [468], Gilbert [481], Guy [571], Harper [605], Klee [764–767], Mecklenberg et al. [951], Mills [956], Preparata and Nievergelt [1083], Singleton [1215], Tang and Liu [1307], Vasil'ev [1367], Wyner [1440] and Yuen [1448, 1449].

(ii) Comma-free codes – see Ball and Cummings [60, 61], Baumert and Cantor [85], Crick et al. [316], Eastman [399], Golomb [523, pp. 118–122], Golomb et al. [528], Hall [587, pp. 11–12], Jiggs [692], Miyakawa and Moriya [967], Niho [992] and Redinbo and Walcott [1102]. See also the remarks on codes for synchronizing in the Notes to Ch. 1.

(iii) Codes with unequal error protection – see Gore and Kilgus [549], Kilgus and Gore [761] and Mandelbaum [901].

(iv) Coding for channels with feedback – see Berlekamp [124], Horstein [664] and Schalkwijk et al. [1153–1155].

(v) Codes for the Gaussian channel – see Biglieri et al. [148–151], Blake [155, 156, 158], Blake and Mullin [162], Chadwick et al. [256, 257], Gallager [464], Ingemarsson [683], Landau [791], Ottoson [1017], Shannon [1191], Slepian [1221–1223] and Zetterberg [1456].

(vi) The complexity of decoding – see Bajoga and Walbesser [59], Chaitin [257a–258a], Gelfand et al. [471], Groth [564], Justesen [706], Kolmogorov [774a], Marguinaud [916], Martin-Löf [917a], Pinsker [1046a], Sarwate [1145] and Savage [1149–1152a].

(vii) The connections between coding theory and the packing of equal spheres in n -dimensional Euclidean space – see Leech [803–805], [807], Leech and Sloane [808–810] and Sloane [1226].

The following books and monographs on coding theory are our predecessors: Berlekamp [113, 116], Blake and Mullin [162], Cameron and Van Lint [234], Golomb [522], Lin [834], Van Lint [848], Massey [922a], Peterson [1036a], Peterson and Weldon [1040], Solomon [1251] and Sloane [1227a]; while the following collections contain some of the papers in the bibliography: Berlekamp [126], Blake [157], the special issues [377a, 678, 679], Hartnett [620], Mann [909] and Slepian [1224]. See also the bibliography [1022].

We owe a considerable debt to several friends who read the first draft very carefully, made numerous corrections and improvements, and frequently saved us from dreadful blunders. In particular we should like to thank I.F. Blake, P. Delsarte, J.-M. Goethals, R.L. Graham, J.H. van Lint, G. Longo, C.L. Mallows, J. McKay, V. Pless, H.O. Pollak, L.D. Rudolph, D.W. Sarwate, many other colleagues at Bell Labs, and especially A.M. Odlyzko for

their help. Not all of their suggestions have been followed, however, and the authors are fully responsible for the remaining errors. (This conventional remark is to be taken seriously.) We should also like to thank all the typists at Bell Labs who have helped with the book at various times, our Secretary Peggy van Ness who has helped in countless ways, and above all Marion Messersmith who has typed and retyped most of the chapters. Sam Lomonaco has very kindly helped us check the galley proofs.

Contents of Part I

Preface	v
-------------------	---

Contents of Part I	xi
Abbreviated contents of Part II	xvi

Chapter 1. Linear codes

1. Linear codes	1
2. Properties of a linear code	5
3. At the receiving end	7
4. More about decoding a linear code	15
5. Error probability	18
6. Shannon's theorem on the existence of good codes	22
7. Hamming codes	23
8. The dual code	26
9. Construction of new codes from old (II)	27
10. Some general properties of a linear code	32
11. Summary of Chapter 1	34
Notes on Chapter 1	34

Chapter 2. Nonlinear codes, Hadamard matrices, designs and the Golay code

1. Nonlinear codes	38
2. The Plotkin bound	41
3. Hadamard matrices and Hadamard codes	44

4. Conferences matrices	55
5. t -designs	58
6. An introduction to the binary Golay code	64
7. The Steiner system $S(5, 6, 12)$, and nonlinear single-error correcting codes	70
8. An introduction to the Nordstrom–Robinson code	73
9. Construction of new codes from old (III)	76
Notes on Chapter 2	78

Chapter 3. An introduction to BCH codes and finite fields

1. Double-error-correcting BCH codes (I)	80
2. Construction of the field $GF(16)$	82
3. Double-error-correcting BCH codes (II)	86
4. Computing in a finite field	88
Notes on Chapter 3	92

Chapter 4. Finite fields

1. Introduction	93
2. Finite fields: the basic theory	95
3. Minimal polynomials	99
4. How to find irreducible polynomials	107
5. Tables of small fields	109
6. The automorphism group of $GF(p^m)$	112
7. The number of irreducible polynomials	114
8. Bases of $GF(p^m)$ over $GF(p)$	115
9. Linearized polynomials and normal bases	118
Notes on Chapter 4	124

Chapter 5. Dual codes and their weight distribution

1. Introduction	125
2. Weight distribution of the dual of a binary linear code	125
3. The group algebra	132
4. Characters	134
5. MacWilliams theorem for nonlinear codes	135
6. Generalized MacWilliams theorems for linear codes	141
7. Properties of Krawtchouk polynomials	150
Notes on Chapter 5	153

Chapter 6. Codes, designs and perfect codes

1. Introduction	155
2. Four fundamental parameters of a code	156
3. An explicit formula for the weight and distance distribution	158
4. Designs from codes when $s \leq d'$	160
5. The dual code also gives designs	164
6. Weight distribution of translates of a code	166
7. Designs from nonlinear codes when $s' < d$	174
8. Perfect codes	175
9. Codes over $\text{GF}(q)$	176
10. There are no more perfect codes	179
Notes on Chapter 6	186

Chapter 7. Cyclic codes

1. Introduction	188
2. Definition of a cyclic code	188
3. Generator polynomial	190
4. The check polynomial	194
5. Factors of $x^n - 1$	196
6. t -error-correcting BCH codes	201
7. Using a matrix over $\text{GF}(q^n)$ to define a code over $\text{GF}(q)$	207
8. Encoding cyclic codes	209
Notes on Chapter 7	214

Chapter 8. Cyclic codes (contd.): Idempotents and Mattson–Solomon polynomials

1. Introduction	216
2. Idempotents	217
3. Minimal ideals, irreducible codes, and primitive idempotents	219
4. Weight distribution of minimal codes	227
5. The automorphism group of a code	229
6. The Mattson–Solomon polynomial	239
7. Some weight distributions	251
Notes on Chapter 8	255

Chapter 9. BCH codes

1. Introduction	257
2. The true minimum distance of a BCH code	259

3. The number of information symbols in BCH codes	262
4. A table of BCH codes	266
5. Long BCH codes are bad	269
6. Decoding BCH codes	270
7. Quadratic equations over $GF(2^m)$	277
8. Double-error-correcting BCH codes are quasi-perfect	279
9. The Carlitz-Uchiyama bound	280
10. Some weight distributions are asymptotically normal	282
Notes on Chapter 9	291

Chapter 10. Reed-Solomon and Justesen codes

1. Introduction	294
2. Reed-Solomon codes	294
3. Extended RS codes	296
4. Idempotents of RS codes	296
5. Mapping $GF(2^m)$ codes into binary codes	298
6. Burst error correction	301
7. Encoding Reed-Solomon codes	301
8. Generalized Reed-Solomon codes	303
9. Redundant residue codes	305
10. Decoding RS codes	306
11. Justesen codes and concatenated codes	306
Notes on Chapter 10	315

Chapter 11. MDS codes

1. Introduction	317
2. Generator and parity check matrices	318
3. The weight distribution of an MDS code	319
4. Matrices with every square submatrix nonsingular	321
5. MDS codes from RS codes	323
6. n -arcs	326
7. The known results	327
8. Orthogonal arrays	328
Notes on Chapter 11	329

Chapter 12. Alternant, Goppa and other generalized BCH codes

1. Introduction	332
2. Alternant codes	333

3. Goppa codes	338
4. Further properties of Goppa codes	346
5. Extended double-error-correcting Goppa codes are cyclic	350
6. Generalized Srivastava codes	357
7. Chien-Choy generalized BCH codes	360
8. The Euclidean algorithm	362
9. Decoding alternant codes	365
Notes on Chapter 12	368

Abbreviated contents of Part II

Abbreviated contents of Part I	v
Contents of Part II	vi
Chapter 13. Reed-Muller codes	370
Chapter 14. First-order Reed-Muller codes	406
Chapter 15. Second-order Reed-Muller, Kerdock and Preparata codes	433
Chapter 16. Quadratic-residue codes	480
Chapter 17. Bounds on the size of a code	523
Chapter 18. Methods for combining codes	567
Chapter 19. Self-dual codes and invariant theory	596
Chapter 20. The Golay codes	634
Chapter 21. Association schemes	651
Appendix A. Tables of the best codes known	673
Appendix B. Finite geometries	692
Bibliography	703
Index	757

Linear codes

§1. Linear codes

Codes were invented to correct errors on noisy communication channels. Suppose there is a telegraph wire from Boston to New York down which 0's and 1's can be sent. Usually when a 0 is sent it is received as a 0, but occasionally a 0 will be received as a 1, or a 1 as a 0. Let's say that on the average 1 out of every 100 symbols will be in error. I.e. for each symbol there is a probability $p = 1/100$ that the channel will make a mistake. This is called a *binary symmetric channel* (Fig. 1.1).

There are a lot of important messages to be sent down this wire, and they must be sent as quickly and reliably as possible. The messages are already written as a string of 0's and 1's—perhaps they are being produced by a computer.

We are going to *encode* these messages to give them some protection against errors on the channel. A block of k message symbols $u = u_1 u_2 \dots u_k$

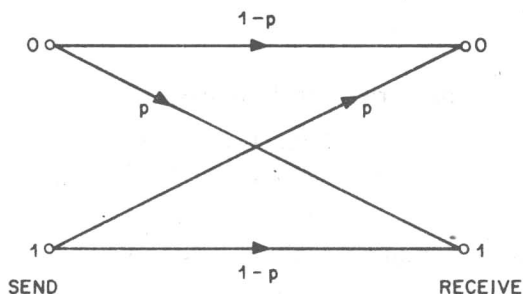


Fig. 1.1. The binary symmetric channel, with error probability p . In general $0 \leq p \leq \frac{1}{2}$.

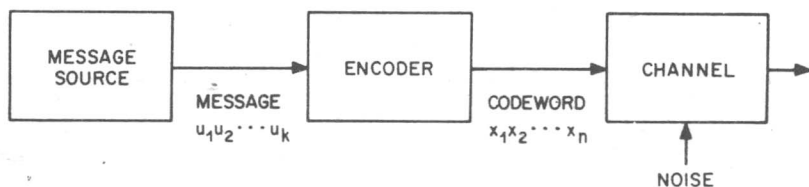


Fig. 1.2.

($u_i = 0$ or 1) will be encoded into a *codeword* $\mathbf{x} = x_1x_2 \dots x_n$ ($x_i = 0$ or 1) where $n \geq k$ (Fig. 1.2); these codewords form a *code*.

The method of encoding we are about to describe produces what is called a *linear code*. The first part of the codeword consists of the message itself:

$$x_1 = u_1, \quad x_2 = u_2, \quad \dots, \quad x_k = u_k,$$

followed by $n - k$ *check* symbols

$$x_{k+1}, \dots, x_n.$$

The check symbols are chosen so that the codewords satisfy

$$H \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = H\mathbf{x}^T = 0, \quad (1)$$

where the $(n - k) \times n$ matrix H is the *parity check matrix* of the code, given by

$$H = [A | I_{n-k}], \quad (2)$$

A is some fixed $(n - k) \times k$ matrix of 0's and 1's, and

$$I_{n-k} = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix}$$

is the $(n - k) \times (n - k)$ unit matrix. The arithmetic in Equation (1) is to be performed *modulo 2*, i.e. $0 + 1 = 1$, $1 + 1 = 0$, $-1 = +1$. We shall refer to this as *binary arithmetic*.

Example. Code #1. The parity check matrix

$$H = \left[\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array} \right] \quad (3)$$

defines a code with $k = 3$ and $n = 6$. For this code

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

The message $u_1u_2u_3$ is encoded into the codeword $\mathbf{x} = x_1x_2x_3x_4x_5x_6$, which begins with the message itself:

$$x_1 = u_1, \quad x_2 = u_2, \quad x_3 = u_3,$$

followed by three check symbols $x_4x_5x_6$ chosen so that $H\mathbf{x}^T = 0$, i.e. so that

$$\begin{aligned} x_2 + x_3 + x_4 &= 0, \\ x_1 + x_3 + x_5 &= 0, \\ x_1 + x_2 + x_6 &= 0. \end{aligned} \tag{4}$$

If the message is $\mathbf{u} = 011$, then $x_1 = 0$, $x_2 = 1$, $x_3 = 1$, and the check symbols are

$$\begin{aligned} x_4 &= -1 - 1 = 1 + 1 = 2 = 0, \\ x_5 &= -1 = 1, \quad x_6 = -1 = 1, \end{aligned}$$

so the codeword is $\mathbf{x} = 011011$.

The Equations (4) are called the *parity check equations*, or simply *parity checks*, of the code.

The first parity check equation says that the 2nd, 3rd and 4th symbols of every codeword must add to 0 modulo 2; i.e. their sum must have even parity (hence the name!).

Since each of the 3 message symbols $u_1u_2u_3$ is 0 or 1, there are altogether $2^3 = 8$ codewords in this code. They are:

000000	011011	110110
001110	100011	111000.
010101	101101	

In the general code there are 2^k codewords.

As we shall see, code # 1 is capable of correcting a single channel error (in any one of the six symbols), and using this code reduces the average probability of error per symbol from $p = .01$ to .00215 (see Problem 24). This is achieved at the cost of sending 6 symbols only 3 of which are message symbols.

We take (1) as our general definition:

Definition. Let H be any binary matrix. The *linear code with parity check matrix H* consists of all vectors \mathbf{x} such that

$$H\mathbf{x}^T = 0.$$

(where this equation is to be interpreted modulo 2).

It is convenient, but not essential, if H has the form shown in (2) and (3), in which case the first k symbols in each codeword are *message* or *information* symbols, and the last $n - k$ are *check* symbols.