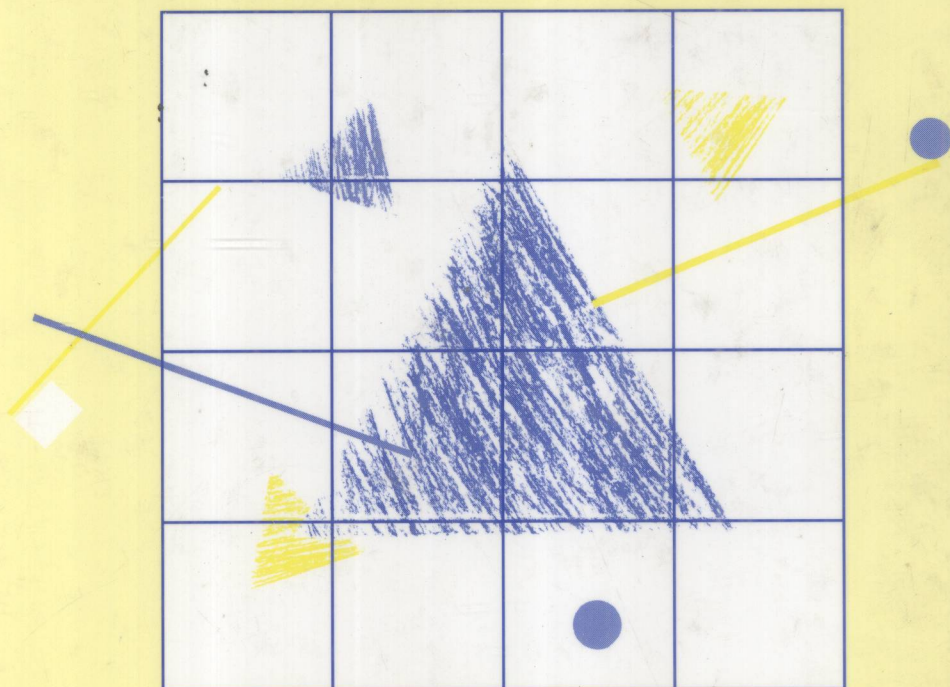
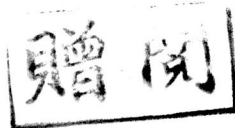


**ERROR
CONTROL
CODING**
AN INTRODUCTION



PETER SWEENEY

7P14
5974
9761352



Error control coding

An introduction

Peter Sweeney

Department of Electronic and Electrical Engineering
University of Surrey



E9761352



Prentice Hall

New York London Toronto Sydney Tokyo Singapore



First published 1991 by
Prentice Hall International (UK) Ltd
66 Wood End, Hemel Hempstead
Hertfordshire HP2 4RG
A division of
Simon & Schuster International Group

© P. Sweeney 1991

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 prior permission, in writing, from the publisher.

For permission within the United States of America contact Prentice Hall Inc., Englewood Cliffs, NJ 07632.

Typeset in 10/12 pt Times by Columns Design and Production Services Ltd, Reading

Printed and bound in Great Britain
by Dotesios Printers Ltd, Trowbridge, Wiltshire.

Library of Congress Cataloging-in-Publication Data

Sweeney, Peter, 1950–

Error control coding : an introduction / by Peter Sweeney.

p. cm.

Includes bibliographical references and index.

ISBN (invalid) 0–13–248119–5

1. Coding theory. 2. Error-correcting codes (Information theory)

I. Title

TK5102.5.S958 1991

629.8'315–dc20

90–7522

CIP

British Library Cataloguing in Publication Data

Sweeney, Peter 1950–

Error control coding : an introduction.

1. Error-correcting codes

I. Title

005.72

ISBN 0–13–284118–5

ISBN 0–13–284126–6 pbk

1 2 3 4 5 95 94 93 92 91

Error control coding

An introduction

Preface

All sciences and technologies contain subjects that are generally reckoned to be strictly for the *cognoscenti*, being both difficult and of marginal importance. In the early days of my exposure to the subject of error correcting codes, that was certainly the image that it presented to me. I was concerned that my efforts to understand would at best be useful for the one system on which I was working. Secretly I was afraid that whole subject might be beyond my mental capabilities. I went as far into the subject as was strictly necessary and no further. It was some years before the wider applicability became obvious and I was brave enough to delve a little deeper.

Having been closely concerned with the design and assessment of error control schemes for some seven years now, I can see that my image of the subject was completely wrong. Error control methods can deliver acceptable results more cheaply than can be achieved without them. Moderately successful attempts to teach at advanced undergraduate and postgraduate levels have shown that the subject is not even that difficult, even though some of the students are no more intelligent than I am! The old image dies hard, but good treatments of coding are starting to appear in the better text books on digital communications and there are at least four reference works from the 1980s that can play some part in making the subject more accessible.

The intention of this book is to help fill a gap in the middle market, providing a specialized text whose approach is designed for students and which is not beyond their pockets. Once the fundamentals have been grasped they can then make good use of the reference books and research papers to study particular matters in greater depth. This does not mean that there is no coverage of advanced material; such subjects as finite field arithmetic, Reed Solomon codes, the Berlekamp–Massey algorithm and the Viterbi algorithm are all treated in some detail, albeit through the medium of simple examples. Nor does it mean (I hope) that there is little of interest to the practicing engineer, since I have tried to point out the practical implications of the theory, including some that I have not seen treated elsewhere. Indeed I hope that the removal of certain specialist aspects will allow important principles to emerge more clearly.

The approach that I have adopted seems to work reasonably well for my own students. It has been tested out on several classes and has been improved in the process. In essence it is to start from simple examples and proceed from the particular to the general. Although my own difficulties with the subject are largely in the past, I can still remember what those difficulties were and I know that working through examples has helped me to understand more clearly. With the passage of time my own understanding has improved but my memory of the learning process has faded, so that in a few years time I am not sure that I will still be in touch with the needs of beginners. I hope I have captured roughly the right moment.

There is enough material in this book easily to fill thirty hours of lectures. I imagine that is more than enough for most courses on error control coding and some selection will have to be made. I have used the material to teach three different, although partially overlapping, courses to postgraduates and final-year undergraduates. I have tried to point out at the start of each chapter whether there are any prerequisites for its understanding. I have also occasionally included similar material in two different places to avoid too much cross-referencing. The example that comes to mind is that a simple description of Reed Solomon codes has been given in the chapter on burst error correction, even though they have been extensively described in an earlier chapter; that earlier treatment would not form a part of a good many courses.

Chapter 1 is intended to show in general terms the uses and benefits of coding and to convince through simple examples that error correction is possible. It would be possible to use this treatment at a relatively early stage of an undergraduate course, particularly if it was intended to take the subject further in a later year. From there the next step would be to cover some or all of Chapter 2 (on block codes) before covering cyclic codes in Chapter 3. Chapter 4 would be appropriate for an in-depth course on block codes including finite field arithmetic, transform techniques and algebraic decoding. If the course is aimed towards communications systems, then Chapter 5, covering convolutional codes and concentrating on Viterbi decoding, will be essential. Further chapters cover the various methods of coding for bursty channels, concatenated codes, coding for bandwidth limited channels (Ungerboeck codes), and methods not using forward error correction, particularly detecting errors and requesting retransmission. In the final chapter I have tried to approach the subject from the problem rather than the code, to meet the needs of the engineer who knows the nature of the problems faced and wants to know what solutions are worth considering. I am sure that this chapter could be a lot better than it is, but it is rare to have it at all and I think it does have something useful to say.

As well as defining the prerequisites, each chapter has suggestions for further reading at the end. The references are not intended to be exhaustive but should be seen as the initial directions along the paths of discovery. All but the last chapter have a selection of exercises at the end to test understanding. Some of the exercises are straightforward, some are more difficult and some are intended

to make the reader think about issues that will arise later in the book. The ones marked with an asterisk are considered to be the most difficult, bearing in mind the stage of understanding that the student will have reached; often they are the ones intended to provoke thought about later issues. There is a teachers' manual too, intended to make sure that the exercises were sensible and that I could do all of them. I hope it will also help teachers to stay one step ahead of the class.

At this point I would like to record my gratitude to a number of people without whom this book might never have happened. Firstly, my initial learning of the subject relied heavily on the authors of books and papers mentioned at various points in the text; I have tried to give credit where it is due, although many of the approaches are so widely adopted that it is impossible to know the original source. My thanks to Bob Harris of ESTEC and Paddy Farrell of Manchester University who were instrumental in persuading me to look closely at coding in the first place, and to Barry Evans who let me use the students on our M.Sc. in Satellite Communication Engineering as guinea-pigs.

Many of the ideas in this book have come from, or been improved by, students here at the Department of Electronic and Electrical Engineering, University of Surrey. Toufic Bechnati, David Tran and Javed Mirza provided ideas on finite field arithmetic and algebraic decoding. John Jackson and Mun-Kein Chang improved my understanding of product codes. Rob Jeffrey and Maher Tarabah contributed to the treatment of convolutional codes. A lecture by Ray Hill of Salford University brought the Griesmer bound to my attention, and I have also adopted a suggestion made by Graham Norton of Bristol University regarding the treatment of linearity. Thanks also to the anonymous reviewers, conscripted by Prentice Hall, whose constructive criticisms have been so helpful. If I have forgotten anyone, as surely I must, my thanks and apologies go to them too.

Finally I want to thank members of my family for their patience and encouragement. My wife, Gillian, had to put up with my getting home late when my real work could not be reasonably accommodated with the preparation of the manuscript. My parents were not affected by such problems, nevertheless the book is a small testimony to their encouragement of my childhood studies. For all the work's deficiencies, however, the responsibility is mine.

Peter Sweeney

Contents

List of figures	xi
Preface	xiii
I The principles of coding	I
1.1 Error control schemes	1
1.2 Coding in communication systems	2
1.3 Measures of distance	6
1.4 Soft-decision metrics for a Gaussian channel	7
1.5 Example of block coding	9
1.6 Random error detection and correction capability of block codes	12
1.7 Output error rates using block codes	13
1.8 Code performance and coding gain	15
1.9 General considerations affecting coding gain	17
1.10 Information theory	18
1.11 Further reading	21
1.12 Exercises	21
2 Linear block codes	23
2.1 Introduction	23
2.2 Mathematics of binary codes	23
2.3 Linearity	24
2.4 Parity checks	26
2.5 Systematic codes	27
2.6 Minimum Hamming distance of a linear block code	27
2.7 How to encode – generator matrix	28
2.8 Encoding with the parity check matrix	29
2.9 Decoding with the parity check matrix	31
2.10 Decoding by standard array	33
2.11 Codec design for linear block codes	34

2.12	Modifications to block codes	36
2.13	Bounds on block codes	39
2.14	Further reading	43
2.15	Exercises	44
3	Cyclic codes	46
3.1	Introduction	46
3.2	Definition of a cyclic code	46
3.3	Example of a cyclic code	47
3.4	Polynomial representation	48
3.5	Encoding by convolution	48
3.6	Establishing the cyclic property	50
3.7	Deducing the properties of a cyclic code	50
3.8	Primitive polynomials	51
3.9	Systematic encoding of cyclic codes	52
3.10	Syndrome of a cyclic code	54
3.11	Implementation of encoding	54
3.12	Decoding	57
3.13	Decoder operation	60
3.14	Multiple error correction	61
3.15	Example of multiple error correction	62
3.16	Shortened cyclic codes	64
3.17	Expurgated cyclic codes	65
3.18	BCH codes	67
3.19	Further reading	68
3.20	Exercises	68
4	Block codes based on finite field arithmetic	70
4.1	Introduction	70
4.2	Definition of a finite field	70
4.3	Prime size finite field $\text{GF}(p)$	72
4.4	Extensions to the binary field – finite field $\text{GF}(2^m)$	73
4.5	Zech logarithms	75
4.6	Implementation of finite field representation	75
4.7	Properties of polynomials and finite field elements	77
4.8	Specifying codes by roots	80
4.9	Hamming codes	81
4.10	BCH codes	81
4.11	Fourier transform in a finite field	84
4.12	Roots and spectral components	86
4.13	BCH codes in the frequency domain	86
4.14	BCH decoding and the BCH bound	87
4.15	Decoding in the frequency domain	88

4.16	Encoding and decoding example	90
4.17	Further examples of Reed Solomon decoding	92
4.18	Decoding of binary BCH codes	95
4.19	Polynomial form of the key equation	96
4.20	Euclid's algorithm	97
4.21	The Berlekamp–Massey algorithm	99
4.22	Extended Reed Solomon codes	101
4.23	Erasure decoding of BCH codes	104
4.24	Example of erasure decoding of Reed Solomon codes	105
4.25	Further reading	107
4.26	Exercises	108
5	Convolutional codes	110
5.1	Introduction	110
5.2	General properties of convolutional codes	111
5.3	Generator polynomials	112
5.4	Terminology	112
5.5	Encoder state diagram	113
5.6	Distance structure of convolutional codes	114
5.7	Evaluating distance and weight structure	116
5.8	Catastrophic error propagation	117
5.9	Maximum likelihood decoding	118
5.10	The Viterbi algorithm	119
5.11	Example of Viterbi decoding	120
5.12	Applicability of convolutional codes	123
5.13	Performance of convolutional codes	125
5.14	Punctured convolutional codes	126
5.15	Sequential decoding	127
5.16	Syndrome decoding	129
5.17	Further reading	130
5.18	Exercises	131
6	Coding for bursty channels	133
6.1	Introduction	133
6.2	Description of bursty errors	134
6.3	Block codes for single-burst error correction	135
6.4	Fire codes	136
6.5	Convolutional codes for burst error correction	138
6.6	Phased-burst error correcting codes	139
6.7	Techniques for compound channels	140
6.8	Interleaving	141
6.9	Product codes	144

6.10	Further reading	148
6.11	Exercises	149
7	Concatenated codes	150
7.1	Introduction	150
7.2	General principle of concatenation	150
7.3	Concatenation using inner block code	152
7.4	Maximal length codes	152
7.5	Orthogonal codes	153
7.6	Reed Muller codes	153
7.7	Block codes with soft-decision decoding	154
7.8	Concatenation using inner convolutional code	155
7.9	Performance of concatenated codes	156
7.10	Further reading	157
7.11	Exercises	158
8	Coding for bandwidth-limited channels	159
8.1	Introduction	159
8.2	M -ary phase shift keying (MPSK)	160
8.3	Quadrature amplitude modulation	161
8.4	Ungerboeck codes	162
8.5	Set partitioning	162
8.6	Integration of coding with partitioning	165
8.7	Performance of Ungerboeck codes	166
8.8	Further reading	167
8.9	Exercises	168
9	Error detection methods	169
9.1	Introduction	169
9.2	Random error detection performance of block codes	170
9.3	Weight distributions	171
9.4	Worst case undetected error rate	173
9.5	Burst error detection	173
9.6	Examples of error detection codes	173
9.7	Synchronization using block codes	175
9.8	Automatic retransmission request	176
9.9	Stop-and-wait ARQ	177
9.10	Go-back-N ARQ	177
9.11	Selective repeat ARQ	179
9.12	ARQ in communications protocols	180
9.13	Hybrid ARQ/FEC	181

9.14	Error concealment	182
9.15	Further reading	183
9.16	Exercises	183
10	Selection of a coding scheme	185
10.1	Introduction	185
10.2	General considerations	186
10.3	Data structure	187
10.4	Information type	187
10.5	Data rate	188
10.6	Real-time processing	189
10.7	Power and bandwidth constraints	189
10.8	Channel error mechanisms	190
10.9	Cost	190
10.10	Applications	191
	References	193
	Index	196

List of figures

1.1	Coding system	2
1.2	Encoder	3
1.3	Binary symmetric channel	5
1.4	Soft decision detection thresholds	8
1.5	Performance curve for (255,191) block code	17
2.1	Encoder structure	35
2.2	Error correction for linear block code	36
3.1	Encoder for cyclic (7,4) Hamming code	55
3.2	Improved version of encoder for cyclic (7,4) Hamming code	56
3.3	General encoder for cyclic codes	56
3.4	Syndrome formation for shortened code	64
3.5	Syndrome formation for expurgated code	67
4.1	Shift register generation of transform of error locator polynomial	88
4.2	Frequency domain encoding and decoding of Reed Solomon codes	90
4.3	Shift register for recursive extension	91
4.4	Recursive extension with single error	93
4.5	Attempted recursive extension after triple error	94
4.6	Encoding of extended Reed Solomon code	102
4.7	Syndrome formation for extended Reed Solomon code	103
4.8	Recursive extension with one error and two erasures	107
5.1	Convolutional encoder	111
5.2	Encoder state diagram	114
5.3	Modified encoder state diagram	116
5.4	Code trellis	119
5.5	Survivor paths in Viterbi decoding	123
6.1	Error burst of length 10	134
6.2	End-around burst of length 6	135
6.3	Split syndrome formation for (105,94) Fire code	137
6.4	Block interleaving	142
6.5	Rate 1/2 convolutional encoder with interleaving to degree 3	143
6.6	Convolutional interleaving	143
6.7	Product code	144
6.8	Quadruple-error correcting product code with four errors	146
6.9	Product code after row decoding	146

6.10	Product code after decoding of two rows	146
6.11	Product code after row erasure	147
6.12	Cyclic ordering for product code	148
7.1	Code concatenation	151
8.1	Demodulator decision boundaries for MPSK	160
8.2	A 16-point QAM constellation	161
8.3	A 32-point QAM constellation	161
8.4	A 64-point constellation partitioned into two sets	163
8.5	Set A of a partitioned 64-point constellation	164
8.6	Set B of a partitioned 64-point constellation	164
8.7	A 64-point constellation partitioned twice	165
8.8	A 64-point constellation partitioned three times	165
8.9	Trellis for rate 1/2 code	167
8.10	Partitioned 8-PSK constellation	167
9.1	Syndrom calculation for synchronization	175
9.2	Go-back-5 ARQ	178
9.3	Selective repeat ARQ	179
9.4	Selective repeat ARQ with five-message buffer	180

I

The principles of coding

1.1 Error control schemes

Error control coding is concerned with methods of delivering information from a source to a destination with a minimum of errors. As such it can be seen as a branch of information theory and certainly traces its origins to Shannon's work in the late 1940s. The early theoretical work indicates what is possible and provides some insights into the general principles of error control. On the other hand, the problems involved in finding and implementing codes have meant that the practical effects of employing coding are at present somewhat different from what was originally expected.

Shannon's work showed that any communication channel could be characterized by a capacity at which information could be reliably transmitted. At any rate of information transmission up to the channel capacity it should be possible to transfer information at error rates that can be reduced to any desired level. Error control can be provided by introducing redundancy into transmissions. This means that more symbols are included in the message than are strictly needed just to convey the information, with the result that only certain patterns at the receiver correspond to valid transmissions. Once an adequate degree of error control has been introduced, the error rates can be made as low as required by extending the length of the code, thus averaging the effects of noise over a longer period.

Experience has shown that to find good long codes is more easily said than done. Present-day practice is not to use codes as a way of obtaining the theoretical channel capacity but to concentrate on the improvements that can be obtained compared with uncoded communications. Thus the use of coding may increase the operational range of a communications system, reduce the error rates, reduce the transmitted power requirements or obtain a blend of all these benefits. Apart from the variety of codes that are available, there are several general techniques for the control of errors and the choice will depend on the nature of the data and the user's requirements for error-free reception. The most complex techniques fall into the category of forward error correction, where it is

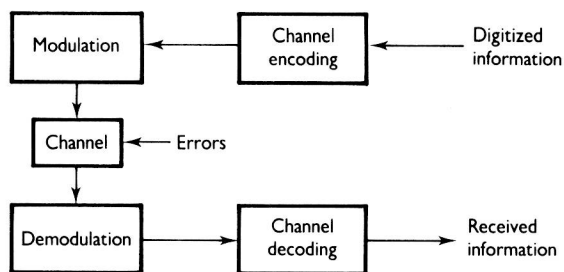


Figure 1.1 Coding system

assumed that errors will occur and a code capable of correcting the assumed errors is applied to the messages. Alternatives are to detect errors and request retransmission, which is known as retransmission error control, or to use inherent redundancy to process the erroneous data in a way that will make the errors subjectively unimportant. This latter technique of error concealment is essentially a signal-processing task and is not considered here. This book concentrates on forward error correction, but the performance of error detection codes and their use in conjunction with retransmission error control are considered in Chapter 9.

This chapter first looks at the place of coding within telecommunications systems. Sections 1.3 and 1.4 then address the question of what coding is in principle trying to achieve and the interface between the demodulator and the decoder; these sections may be omitted on first reading without seriously weakening the rest of the chapter. Next is found a simple example of a code, from a class known as block codes, which is used to show how error detection and correction may in principle be achieved. A number of general results for error detection and correction capabilities and output error rates are then obtained for block codes. Sections 1.8 and 1.9 introduce the concept of coding gain and explain many of the commonly found features of code performance. Finally the practical considerations are put into perspective by a summary of the results of information theory, which provide some insights into the possibilities for advancements in coded systems.

1.2 Coding in communication systems

A typical communication system incorporating coding is shown in Figure 1.1. The important elements of the system are as follows:

Source encoding

Information is given a digital representation, possibly in conjunction with techniques for removal of any inherent redundancy within the data. Such techniques, although an important subject in their own right, are not considered in this text. The most important point for our purposes is that the error control techniques to be described will all operate on a digital form of information.

Error control coding

The encoder is represented in Figure 1.2. The information is formed into frames to be presented to the encoder, each frame consisting of a fixed number of symbols. In most cases the symbols at the input of the encoder are bits; in a very few cases symbols consisting of several bits are required by the encoder. The term *symbol* will be used to maintain generality.

The symbols in the input frame, and possibly a number of previous frames, are used by the encoder to produce its output. The output generally contains more symbols than the input, i.e. redundancy has been added. A commonly used descriptor of a code is the *code rate* (R), which is the ratio of input to output symbols in one frame. A low code rate indicates a high degree of redundancy, which is likely to provide more effective error control than a higher rate at the expense of reducing the information throughput.

If the decoder uses only the current frame to produce its output, then the code is called a (n, k) block code, with the number of input symbols per frame designated k and the corresponding number of output symbols n . If the encoder remembers a number of previous frames and uses them in its algorithm, then the code is called a tree code and is usually a member of a subset known as convolutional codes. In this case the number of symbols in the input frame will be designated k_0 with n_0 symbols in the output frame.

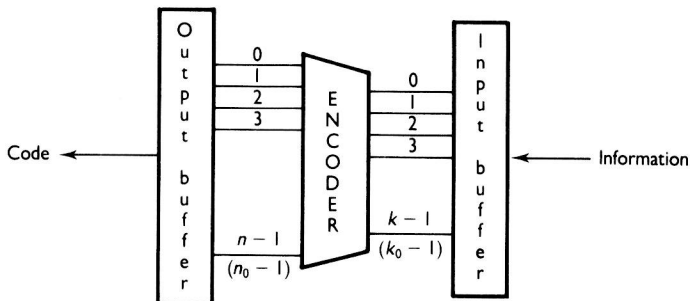


Figure 1.2 Encoder