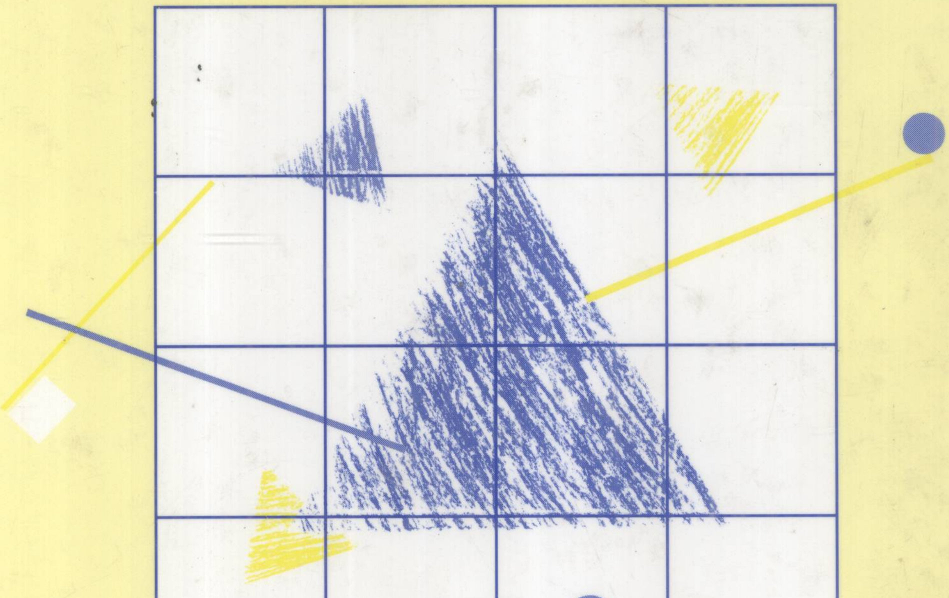
ERROR CONTROL CODING

AN INTRODUCTION



PETER SWEENEY

Error control coding
An introduction

Peter Sweeney

Department of Electronic and Electrical Engineering
University of Surrey

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

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

1
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

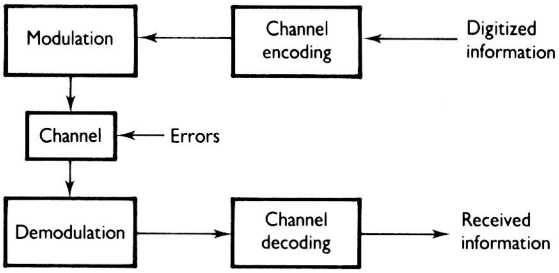


Figure 1.1 Coding system

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.

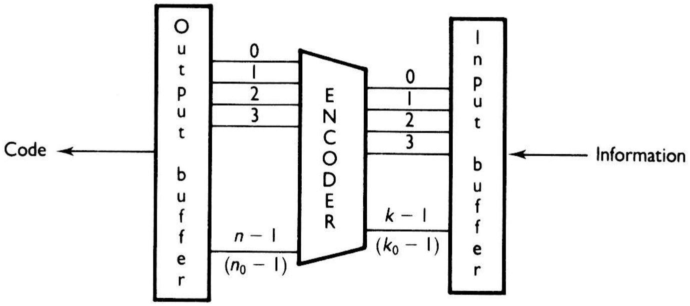


Figure 1.2 Encoder