

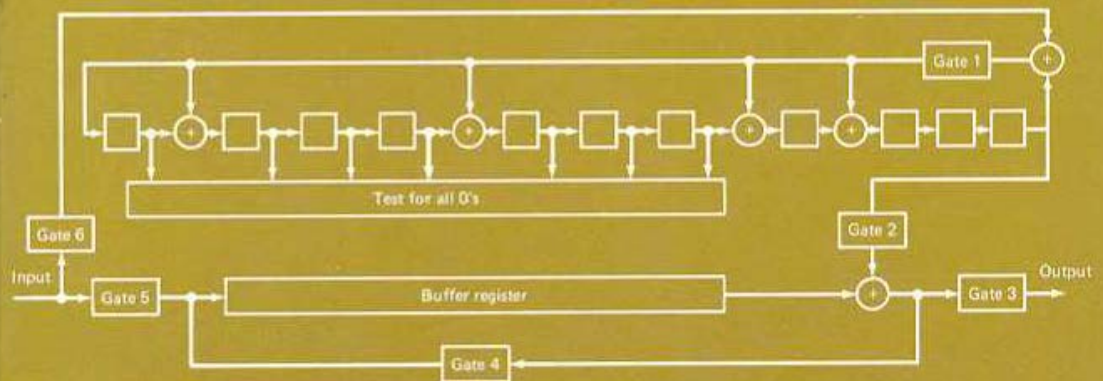
IN/COSTELLO

SHU LIN / DANIEL J. COSTELLO, Jr.

Error Control Coding:

Fundamentals and Applications

Error Control Coding:
Fundamentals and Applications



PRENTICE-HALL SERIES IN COMPUTER APPLICATIONS IN ELECTRICAL ENGINEERING

Franklin F. Kuo, Editor

QA
268
L55
C.1

ERROR CONTROL CODING
Fundamentals and Applications

**PRENTICE-HALL COMPUTER APPLICATIONS
IN ELECTRICAL ENGINEERING SERIES**

FRANKLIN F. KUO, editor

ABRAMSON and KUO, Computer-Communication Networks
BOWERS and SEDORE, Sceptre: A Computer Program for Circuit and Systems Analysis
CADZOW, Discrete Time Systems: An Introduction with Interdisciplinary Applications
CADZOW and MARTENS, Discrete-Time and Computer Control Systems
DAVIS, Computer Data Displays
FRIEDMAN and MENON, Fault Detection in Digital Circuits
HUELSMAN, Basic Circuit Theory
JENSEN and LIEBERMAN, IBM Circuit Analysis Program: Techniques and Applications
JENSEN and WATKINS, Network Analysis: Theory and Computer Methods
KLINE, Digital Computer Design
KOCHENBURGER, Computer Simulation of Dynamic Systems
KUO, (ed.) Protocols and Techniques for Data Communication Networks
KUO and MAGNUSON, Computer Oriented Circuit Design
LIN, An Introduction to Error-Correcting Codes
LIN and COSTELLO, Error Control Coding: Fundamentals and Applications
NAGLE, CARROLL, and IRWIN, An Introduction to Computer Logic
RHYNE, Fundamentals of Digital Systems Design
SIFFERLEN and VARTANIAN, Digital Electronics with Engineering Applications
STAUDHAMMER, Circuit Analysis by Digital Computer
STOUTEMYER, PL/1 Programming for Engineering and Science

ERROR CONTROL CODING

Fundamentals and Applications

SHU LIN

University of Hawaii
Texas A&M University

DANIEL J. COSTELLO, JR.

Illinois Institute of Technology

Prentice-Hall, Inc. Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging in Publication Data

LIN, SHU.
Error control coding.

(Prentice-Hall computer applications in
electrical engineering series)
Includes bibliographical references and index.

I. Error-correcting codes (Information theory)

I. Costello, Daniel J. . II. Title.

III. Series.

QA268.L55 .001.53'9 82-5255

ISBN 0-13-283796-X AACR2



Editorial/production supervision and interior design by Anne Simpson

Cover design by Marvin Warshaw

Manufacturing buyer: Joyce Levatino

© 1983 by Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632

All rights reserved. No part of this book
may be reproduced in any form or
by any means without permission in writing
from the publisher.

Printed in the United States of America

10 9 8 7

ISBN 0-13-283796-X

PRENTICE-HALL INTERNATIONAL, INC., *London*
PRENTICE-HALL OF AUSTRALIA PTY. LIMITED, *Sydney*
EDITORIA PRENTICE-HALL DO BRAZIL, LTDA, *Rio de Janeiro*
PRENTICE-HALL CANADA INC., *Toronto*
PRENTICE-HALL OF INDIA PRIVATE LIMITED, *New Delhi*
PRENTICE-HALL OF JAPAN, INC., *Tokyo*
PRENTICE-HALL OF SOUTHEAST ASIA PTE. LTD., *Singapore*
WHITEHALL BOOKS LIMITED, *Wellington, New Zealand*

will be a replica of the information sequence $\mathbf{u}$, although the noise may cause some *decoding errors*. Another major topic of this book is the design and implementation of channel decoders that minimize the probability of decoding error.

The *source decoder* transforms the *estimated sequence* $\hat{\mathbf{u}}$ into an *estimate* of the source output and delivers this estimate to the *destination*. When the source is continuous, this involves digital-to-analog (D/A) conversion. In a well-designed system, the estimate will be a faithful reproduction of the source output except when the channel (or storage medium) is very noisy.

To focus attention on the channel encoder and channel decoder, (1) the information source and source encoder are combined into a *digital source* with output $\mathbf{u}$; (2) the modulator (or writing unit), the channel (or storage medium), and the demodulator (or reading unit) are combined into a *coding channel* with input $\mathbf{v}$ and output $\mathbf{r}$; and (3) the source decoder and destination are combined into a *digital sink* with input $\hat{\mathbf{u}}$. A simplified block diagram is shown in Figure 1.2.

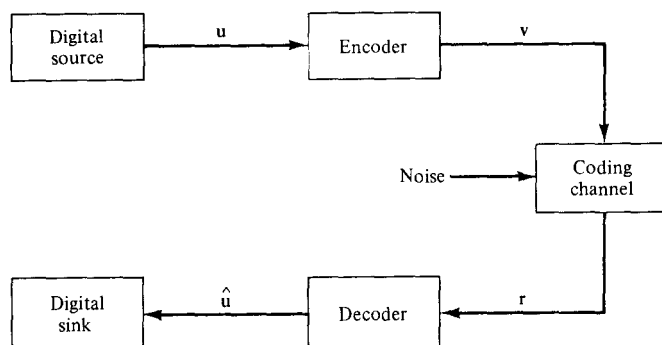


Figure 1.2 Simplified model of a coded system.

The major engineering problem that is addressed in this book is to design and implement the channel encoder/decoder pair such that (1) information can be transmitted (or recorded) in a noisy environment as fast as possible, (2) reliable reproduction of the information can be obtained at the output of the channel decoder, and (3) the cost of implementing the encoder and decoder falls within acceptable limits.

1.2 TYPES OF CODES

There are two different types of codes in common use today, block codes and convolutional codes. The encoder for a block code divides the information sequence into message blocks of k information bits each. A message block is represented by the binary k -tuple $\mathbf{u} = (u_1, u_2, \dots, u_k)$ called a *message*. (In block coding, the symbol $\mathbf{u}$ is used to denote a k -bit message rather than the entire information sequence.) There are a total of 2^k different possible messages. The encoder transforms each message $\mathbf{u}$ independently into an n -tuple $\mathbf{v} = (v_1, v_2, \dots, v_n)$ of discrete symbols called a *code word*. (In block coding, the symbol $\mathbf{v}$ is used to denote an n -symbol block rather than the entire encoded sequence.) Therefore, corresponding to the 2^k different possible messages, there are 2^k different possible code words at the encoder

output. This set of 2^k code words of length n is called an (n, k) *block code*. The ratio $R = k/n$ is called the *code rate*, and can be interpreted as the number of information bits entering the encoder per transmitted symbol. Since the n -symbol output code word depends only on the corresponding k -bit input message, the encoder is memoryless, and can be implemented with a combinational logic circuit.

In a binary code, each code word $\mathbf{v}$ is also binary. Hence, for a binary code to be useful (i.e., to have a different code word assigned to each message), $k \leq n$ or $R \leq 1$. When $k < n$, $n - k$ redundant bits can be added to each message to form a code word. These redundant bits provide the code with the capability of combating the channel noise. For a fixed code rate R , more redundant bits can be added by increasing the block length n of the code while holding the ratio k/n constant. How to choose these redundant bits to achieve reliable transmission over a noisy channel is the major problem in designing the encoder. An example of a binary block code with $k = 4$ and $n = 7$ is shown in Table 1.1. Chapters 3 through 9 are devoted to the analysis and design of block codes for controlling errors in a noisy environment.

TABLE 1.1 BINARY BLOCK CODE WITH
 $k = 4$ AND $n = 7$

Messages	Code words
(0 0 0 0)	(0 0 0 0 0 0 0)
(1 0 0 0)	(1 1 0 1 0 0 0)
(0 1 0 0)	(0 1 1 0 1 0 0)
(1 1 0 0)	(1 0 1 1 1 0 0)
(0 0 1 0)	(1 1 1 0 0 1 0)
(1 0 1 0)	(0 0 1 1 0 1 0)
(0 1 1 0)	(1 0 0 0 1 1 0)
(1 1 1 0)	(0 1 0 1 1 1 0)
(0 0 0 1)	(1 0 1 0 0 0 1)
(1 0 0 1)	(0 1 1 1 0 0 1)
(0 1 0 1)	(1 1 0 0 1 0 1)
(1 1 0 1)	(0 0 0 1 1 0 1)
(0 0 1 1)	(0 1 0 0 0 1 1)
(1 0 1 1)	(1 0 0 1 0 1 1)
(0 1 1 1)	(0 0 1 0 1 1 1)
(1 1 1 1)	(1 1 1 1 1 1 1)

The encoder for a convolutional code also accepts k -bit blocks of the information sequence $\mathbf{u}$ and produces an encoded sequence (code word) $\mathbf{v}$ of n -symbol blocks. (In convolutional coding, the symbols $\mathbf{u}$ and $\mathbf{v}$ are used to denote sequences of blocks rather than a single block.) However, each encoded block depends not only on the corresponding k -bit message block at the same time unit, but also on m previous message blocks. Hence, the encoder has a *memory order* of m . The set of encoded sequences produced by a k -input, n -output encoder of memory order m is called an (n, k, m) *convolutional code*. The ratio $R = k/n$ is called the *code rate*. Since the encoder contains memory, it must be implemented with a sequential logic circuit.

In a binary convolutional code, redundant bits for combating the channel noise

can be added to the information sequence when $k < n$ or $R < 1$. Typically, k and n are small integers and more redundancy is added by increasing the memory order m of the code while holding k and n , and hence the code rate R , fixed. How to use the memory to achieve reliable transmission over a noisy channel is the major problem in designing the encoder. An example of a binary convolutional encoder with $k = 1$, $n = 2$, and $m = 2$ is shown in Figure 1.3. As an illustration of how code words are generated, consider the information sequence $\mathbf{u} = (1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ \dots)$, where the leftmost bit is assumed to enter the encoder first. Using the rules of exclusive-or addition, and assuming that the multiplexer takes the first encoded bit from the top output, it is easy to see that the encoded sequence is $\mathbf{v} = (1 \ 1, \ 1 \ 0, \ 1 \ 0, \ 0 \ 0, \ 0 \ 1, \ 1 \ 1, \ 0 \ 0, \ 0 \ 0, \ 0 \ 0, \ \dots)$. Chapters 10 through 14 are devoted to the analysis and design of convolutional codes for controlling errors in a noisy environment.

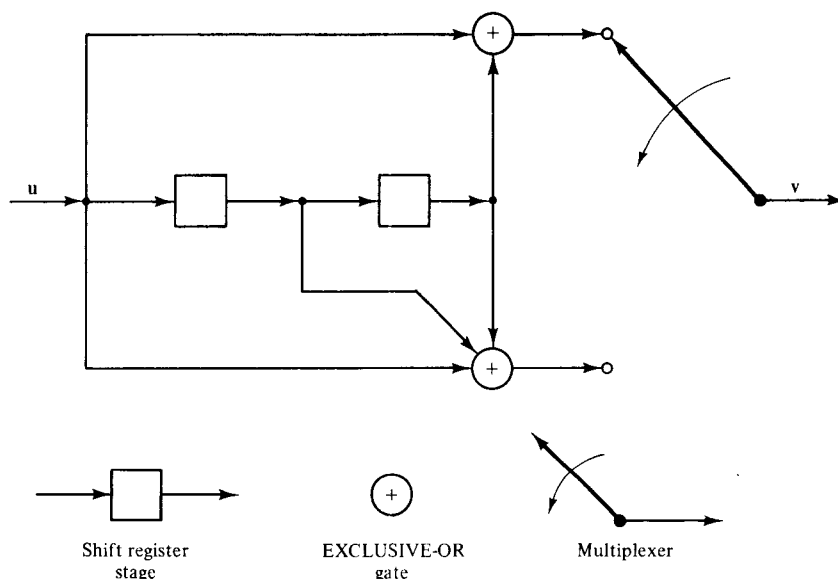


Figure 1.3 Binary convolutional encoder with $k = 1$, $n = 2$, and $m = 2$.

1.3 MODULATION AND DEMODULATION

The modulator in a communication system must select a waveform of duration T seconds, which is suitable for transmission, for each encoder output symbol. In the case of a binary code, the modulator must generate one of two signals, $s_0(t)$ for an encoded "0" or $s_1(t)$ for an encoded "1." For a wideband channel, the optimum choice of signals is

$$\begin{aligned} s_0(t) &= \sqrt{\frac{2E}{T}} \sin \left(2\pi f_0 t + \frac{\pi}{2} \right), & 0 \leq t \leq T \\ s_1(t) &= \sqrt{\frac{2E}{T}} \sin \left(2\pi f_0 t - \frac{\pi}{2} \right), & 0 \leq t \leq T, \end{aligned} \quad (1.1)$$