Error-Correcting Coding Theory

Error-Correcting Coding Theory

Man Young Rhee

Hanyang University
Seoul, Korea

McGraw-Hill Publishing Company

New York St. Louis San Francisco Auckland Bogotá
Caracas Hamburg Lisbon London Madrid Mexico
Milan Montreal New Delhi Oklahoma City
Paris San Juan São Paulo Singapore
Sydney Tokyo Toronto

Library of Congress Cataloging-in-Publication Data

Rhee, Man Young
Error-correcting coding theory

Bibliography: p.
Includes index.
1. Error correcting codes (Information theory)
2. Data transmission systems. I. Title.
TK5102.5.R46 1989 005.7'2 88-37735
ISBN 0-07-052061-5

1234567890 DOC/DOC 89432109

ISBN 0-07-052061-5

The editors for this book were Theron Shreve and Rita T. Margolies, the designer was Naomi Auerbach, and the production supervisor was Dianne L. Walber. It was set in Century Schoolbook. It was composed by the McGraw-Hill Publishing Company Professional & Reference Division composition unit.

Printed and bound by R. R. Donnelley & Sons Company.

Information contained in this work has been obtained by McGraw-Hill, Inc., from sources believed to be reliable. However, neither McGraw-Hill nor its authors guarantees the accuracy or completeness of any information published herein and neither McGraw-Hill nor its authors shall be responsible for any errors, omissions, or damages arising out of use of this information. This work is published with the understanding that McGraw-Hill and its authors are supplying information but are not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought.

Preface

The digital coded systems for either data transmission or data storage have much in common. Since the channel or storage medium is subject to various types of noise, distortion, and interference, the output of the channel or storage medium differs from its input because they are both sensitive to the errors that can result from impaired transmission. The theory and practice of error-correction coding is concerned with protection of digital information against the errors that occur during data transmission or storage. Many ingenious error-correction techniques based on a vigorous mathematical theory have been developed and have many important and frequent applications. The current problem with any high-speed data communication system is how to control the errors that occur during data transmission through a noisy channel. In order to achieve reliable communications, designers should develop good codes and efficient .decoding algorithms. Knowledge of error-correction coding is in great demand and becomes an important asset for many practicing engineers and computer scientists who are involved in the design of large digital systems. The amount of coding research directed at error control will continue to grow because the recent developments in integrated circuit chip technology have advanced so much.

The history of error-correction coding begins in 1948 with the publication of a landmark paper by Claude Shannon. Since Shannon's work, a great deal of effort has been devoted to the encoder-decoder implementation for controlling errors in noisy environments. The pioneering work of coding in the early 1950s concentrated on a theory that required extensive mathematical rigor in the domain of abstract algebra and probability theory. During the 1960s much effort was devoted to finding structures for classes of good codes that could produce an arbitrarily small probability of error, but little progress was made. In the 1970s, coding research began to focus on designing families of codes with larger code lengths and better performance. However, further advances are still required to overcome practical limitations. As the era of the 1980s opened, the emphasis in coding research shifted dramatically from theory to practical applications.

This book is intended to serve as an introduction to error-correction coding theory. The primary objective is to present how the basic concepts and techniques of error control are applied to digital transmission and storage systems so that the underlying theory can be better understood. The book progresses systematically from block codes to convolutional codes, through material that lies at the forefront of modern coding theory. In studying the construction and properties of error-correcting codes, one needs to make extensive use of the notions associated with the theory of vector spaces and to understand the concepts and operations of abstract algebra.

The mathematical prerequisite introduced in Chapter 2 provides a theoretical format for the remainder of the text. Each succeeding chapter then builds upon this theory, applying it to a specific area within coding research. However, mathematical sophistication has been kept at the lowest level possible. More than 134 examples with detailed solutions and many theorems with proofs appear in the text, both to provide a greater understanding of the subject matter and to induce a more flexible approach to the solution of the problems at the end of each chapter. A solution manual is available to instructors through the publisher.

Chapter 1 presents an introduction and overview of error-control coding for digital comunications and computer storage systems. Since coding is an extremely mathematical subject, Chapter 2 presents a brief survey of the powerful structure of abstract algebra. This algebraic framework provides the tools needed by the reader to understand the theory of error-correcting codes. Although the mathematical treatment in this chapter is somewhat rigorous, it is limited in scope to materials that will be useful in succeeding chapters.

Chapters 3 to 8 are devoted to the analysis and design of block codes that control errors in a noisy environment. The fundamentals of linear block codes are thoroughly presented in Chapter 3. Cyclic codes, as they are used in practice, form an important subclass of block codes. These codes are attractive because encoding and syndrome computation can easily be implemented by employing shift-register circuits or digital logic circuits. The basic structure and properties of cyclic codes are presented in Chapter 4.

Chapter 5 is mainly concerned with coding techniques for noisy channels on which transmission errors occur independent of digit position, i.e., random errors. A simple way of decoding for some cyclic codes, known as error-trapping decoding, is covered in this chapter. Chapter 6 covers cyclic codes that are very effective for burst-error correction. Many theorems and proofs that have relevance to burst-error correction are also included. There are extensive discussions of burst-error-trapping decoding using Fire codes.

Chapter 7 provides detailed coverage of BCH codes for multiple error correction. Iterative algorithms for finding the error-location polynomial are presented in great detail. Numerous examples are presented, and various hardware configurations are discussed. Hardware implementations for the syndrome computation circuit and the searching unit for error-location numbers are presented. Software for the error-location polynomial is included, and discussion of nonbinary Reed-Solomon codes for concatenated coding systems is also presented. Chapter 8 presents the subject of majority-logic decodable codes. Majority-logic decoding techniques for correcting random errors are fully covered.

Chapters 9 through 11 are devoted to the principles, structures, and encoding-decoding methods of convolutional codes. Chapter 9 introduces the fundamental properties of convolutional codes, coupled with the encoder state diagram that serves as the basis for studying tree or trellis code structure and distance properties. Chapter 10 presents probabilistic decoding—that is, maximum-likelihood decoding and sequential decoding. Maximum-likelihood decoding using the Viterbi algorithm for hard and soft decisions, as well as sequential decoding using both the Fano and the ZJ stack algorithms, are intensively covered in this chapter. Performance analysis based on code distance properties is also included.

Chapter 11 presents threshold decoding of convolutional codes by Massey's majority-logic decodable principle. Convolutional codes for correcting longer bursts can be obtained by interleaving. The interleaved codes will correct not only burst errors but also many patterns of random errors. In this chapter, a code developed specifically for correcting burst errors is covered in detail. Since this book presents the application of error-correction coding in the context of communications system design, the emphasis throughout is placed on the design and implementation of encoders and decoders.

The scope of the book is adequate to span a two-semester graduate-level sequence, and the material has been organized with such a course in mind. The book can also be used as reference for electronic engineers and computer scientists in industry who are interested in studying the fundamentals of coding and how to apply the principles to the design of error-control digital systems.

A natural division of the material is to cover Chapters 1 through 7 in a first-semester graduate course, and Chapters 8 through 11 in the second. Alternatively, portions of the book can be used for a one-semester course, at the instructor's choice, but it is recommended that the instructor cover Chapters 3, 4, 6, 7, 9, and 10, which include the fundamentals of both block codes and convolutional codes. Chapters 3, 4, 5, 7(partly), 9, and 10 should be studied by senior students who are seriously interested in coding theory.

In most books, the notation for block codes often differs from the notation for convolutional codes. Therefore, I have given careful consideration to choosing notation that brings clarity and internal consistency to the book. The symbol **d** is used for the information data; **c** for the code word or transmitting vector; **e** for the error pattern; and **r** for the received word. This notation is used consistently not only with block codes, but with convolutional codes as well.

This book is the outgrowth of my teaching and research efforts in coding over the last 20 years at the university and in industry, coupled with my supervision of many graduate students at both the M.S. and Ph.D. levels. I thank them all, even if not by name. The books and papers that had a large influence on this book, either directly or indirectly, are listed in the bibliography at the end of the text. However, it is impossible to mention all those people who influenced me a great deal in writing this book. My sincere gratitude goes to them.

I am grateful to the Ministry of Communications, Republic of Korea, Korea Telecommunication Authority, the Electronics and Telecommunications Research Institute, and the Korea Science and Engineering Foundation for their continuing support of my interest and research in the coding field. Some of the research papers for those organizations are contained in this book.

I would like to express my special appreciation to Rita Margolies, Editing Supervisor, who made numerous corrections and suggestions for improvements of this book. I also wish to thank composition and production people for their dedication and perseverance in preparing this beautiful book.

Man Y. Rhee

Contents

Preface vii

Chapter 1. Introduction and Overview 1

Chapter 2. Elements of Abstract Algebra for Coding 15

Chapter 3. Linear Binary Block Codes 35

Chapter 4. Cyclic Codes 89

Chapter 5. Short-Random-Error Correction by Error-Trapping Techniques 115

Chapter 6. Burst-Error Correction for Block Codes 145

Chapter 7. Bose-Chaudhuri-Hocquenghem Codes and Reed-Solomon Codes 167

Chapter 8. Majority-Logic Decodable Codes 255

Chapter 9. Convolutional Codes and Their Structures 311

Chapter 10. Maximum-Likelihood Decoding and Sequential Decoding for Convolutional Codes 351

Chapter 11. Threshold Decoding for Convolutional Codes 387

Bibliography 427

List of Symbols 435

Glossary of Terms 441

Index 453

Chapter 1

Introduction and Overview

Digital data transmission through a physical channel in a communication system and data recording on a storage medium in a computer system have much in common. In both cases, digital data are transferred from an information source to a destination. Since the channel or storage medium is subject to various types of noise, distortion, and interference, the output of the channel or storage medium differs from its input because of the errors that can result from impaired transmission. Therefore, the need for error control arises from the massive amount of data processed in communication and storage systems.

Good codes and coding algorithms are available to meet this need. In addition, the rapid advances in gate array integrated circuit chip technology have made possible the design and implementation of encoder-decoder pairs which use these coding algorithms.

Error-control coding is a special subject with its own history and its own arithmetic systems. The theory and the practice of error-control coding relate to protection of digital information against the errors that occur during data transmission or storage. Many ingenious error-correction techniques have been developed based on rigorous mathematical theory and have then become important subjects with frequent applications. A major concern of the designer is the control of errors so that reliable reproduction of information and data can be obtained.

1.1 Model of Digital-Coded Systems

The block diagram in Fig. 1.1 illustrates the basic elements for transmission or storage of digital information through a coded system. For this system, all information transferred between blocks must be in digital form if error control is to be utilized.

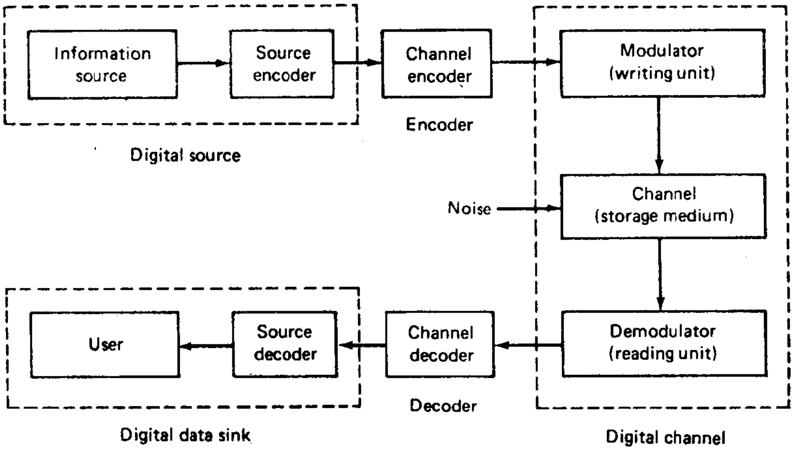

Figure 1.1 Block diagram for digital-coded system.

Data, which enter the communication (or storage) system from the information source, are first processed by a source encoder that is designed to convert the source information into coded form. The source encoder usually transforms the source output into a sequence of binary digits (bits) called the information sequence **d**. The source output can be either a continuous waveform or a sequence of discrete symbols. In the case of analog output, the source encoder must possess an analog-to-digital (A/D) conversion capability, for example, pulse-code modulation (PCM).

The channel encoder transforms the information sequence **d** into a binary-coded sequence **c** called a code word. The channel code word is a new, longer sequence that contains redundancy of parity-check symbols. Each symbol in the code word might be represented by a bit or, perhaps, by a group of bits. In general, binary digits are not suitable for transmission over the waveform channel or for recoding on a digital storage medium.

The binary digits in a code word from the channel encoder are fed into a modulator (or writing unit) that transforms each bit into an elementary signal waveform. Thus, the modulator must convert each bit of the channel code word into a suitable waveform of duration T seconds in order for the bits to transmit. Binary phase shift keying (PSK) or frequency shift keying (FSK) are commonly used as signaling waveforms for transmitting the code word. This waveform enters the channel (or storage medium) and is corrupted by noise. The waveform channel consists of all the hardware and physical media that the waveform passes through in going from the output of the modulator (or writing unit) to the input of the demodulator (or reading unit). In coherent systems, the binary modulation scheme of PSK is often used. With this binary transmission, a 1 is represented by the waveform $s_1(t) = \sqrt{2P}\cos\omega_0 t$, while a 0 is represented by the antipodal signal $s_0(t) = -s_1(t) = \sqrt{2P}\cos(\omega_0 t + 180°)$, where the power in the waveform is $P = E_b/T$. In noncoherent PSK systems, the demodulation of the waveform sign is not possible, so typically a pair of signaling tones, $s_0(t) = \sqrt{2P}\sin(\omega_0 t + \theta)$ and $s_1(t) = \sqrt{2P}\sin(\omega_1 t + \theta)$, is used to represent the transmitted digits 0 and 1, respectively, and to pass the received signal through parallel bandpass matched filters. An alternate approach is to use a correlator with the received signal multiplied by a locally generated version of the transmitted waveform. Typical examples of waveform channels are telephone lines, microwave links, high-frequency radio links, telemetry links, and satellite links. Typical storage media include core memories, magnetic tapes, drums, disk files, and optical memory units. Each of these examples is subject to various types of noise disturbances. Some random or burst noise is usually added to the channel waveform during transmission. Distortion may be present as a result of heavy filtering or multiple signal paths. The disturbance may cause signal suppression which could in turn cause the amplitude of the received signal to vary, or the channel itself may be time-varying. The disturbance may generally be modeled as an additive gaussian process, or it may be urban noise of various kinds, or it may result from intentional jamming by an unfriendly party. Channels are also defined in another way. For memoryless channels, the noise affects each transmitted bit independently. Hence transmission errors occur randomly in the received word $\mathbf{r}$. Typical examples of memoryless random-error channels are most line-of-sight transmission channels, many satellite channels, and deep-space channels. For memory channels, the noise does not occur independently in the channel and the transmission errors occur in bursts. Examples of burst-error channels are (1) radio channels where the error bursts are caused by signal fading as a result of multipath transmission, (2) telephone lines which are affected by impulsive switching noise and crosstalk from other lines, and (3) magnetic recordings, which are subject to tape dropouts due to surface defects and dust particles. Of course, there are some channels which contain a combination of both random and burst errors.

Next, the resulting received signal is processed first by the demodulator and then by the channel decoder. The demodulator (or reading unit) makes a decision for each received signal of duration T seconds to determine which of the two possible digits, 1 or 0, was transmitted. This is called a hard decision. The output of the demodulator is called the received word $\mathbf{r}$. This $\mathbf{r}$ may not match the transmitted code word

c as a result of transmission error. Each demodulated digit is a best estimate of the transmitted digit, but the demodulator makes some errors because of channel noise. The probability that this estimate is correct depends upon the signal-to-noise ratio in the data bandwidth, the amount of signal distortion due to filtering and nonlinear effects, and the detection scheme being used.

The channel decoder transforms the received sequence **r** into a binary sequence $\hat{\mathbf{d}}$, or the estimated information sequence. Since the noise may cause some decoding errors, the channel decoder must be implemented to minimize the probability of decoding error. The channel decoder uses the syndrome of a received code word **r** to correct the errors in the received word and then produces an estimate of the information sequence **d**. If all errors are corrected, the estimated information sequence $\hat{\mathbf{d}}$ matches the original source information **d**.

The source decoder transforms the estimated sequence $\hat{\mathbf{d}}$ into an estimate of the source output and delivers this estimate to the user. Thus the source decoder performs the inverse operation of the source encoder and delivers its output to the data sink.

This book aims to present the analysis, design, and implementation of the channel encoder and decoder, which is a subject known as error-correction coding. The data compression or compaction functions performed by the source encoder and the source decoder are not discussed here. The information source and source encoder are combined into a digital source with output **d**; the modulator (or writing unit), the waveform channel (or storage medium), and the demodulator (or reading unit) are combined into a coding channel with input **c** and output **r**; and the source decoder and user are combined into a digital sink with input $\hat{\mathbf{d}}$. Thus, in this book, the channel encoder and decoder will be referred to simply as the encoder and decoder.

1.2 Coding Classification

Coding is largely classified into two different types: (1) coding for block codes and (2) coding for convolutional codes. All data of interest in a binary block code can be represented as a binary information sequence consisting of 1s and 0s. The encoder for a block code divides the information sequence into information blocks of k bits. Thus, each information block is represented by the binary k-tuple $\mathbf{d} = (d_0, d_1, \ldots, d_{k-1})$ simply called an information sequence. There are a total of 2^k different possible kinds of information sequences. The encoder transforms each k-bit information sequence **d** independently into an n-tuple $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$ called a code word. Thus, a binary block code consists of a set of code words of length n. The elements of a code word are selected from an alphabet of q elements. When $q = 2$ for 0 and 1, the code is called a binary code, whereas when $q > 2$, the code is nonbinary. There are 2^n possible code words in a binary block code of length n $(k < n)$. However, from these 2^n code words, we select only 2^k code words to form a code. Thus a block of k information bits is mapped into a code word of length n 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 known as the code rate. Since the n-bit code word **c** depends only on the corresponding k-bit information sequence, **d**, the encoder is memoryless and can be implemented with a combinational logic circuit. Since $R \leq 1$ for a binary code, $n - k$ redundant bits can be added to each k-bit information sequence to form a code word. These redundant bits provide the code with the capability of combating the channel noise. These $n - k$ redundant bits are called parity-check bits.

Besides its code rate R, an important parameter of a code word is its weight, which is simply the number of nonzero elements that it contains. In general, each code word has its own weight. The set of all weights in a code constitutes the weight distribution of the code. When all the 2^k code words have equal weight, the code is called a constant-weight code. The distance between the two code words is defined as the number of corresponding elements or positions in which they differ. This measure is called the Hamming distance. The smallest value of the set of Hamming distances for the 2^k code words is called the minimum distance and is denoted by $d_{\min}$. An (n, k) block code with the minimum distance $d_{\min}$ guarantees the correction of all the error patterns of $t = \lfloor (d_{\min} - 1)/2 \rfloor$ or fewer errors ($\lfloor x \rfloor$ denotes the largest integer contained in x). The performance characteristics obtained by coding will depend on a number of code parameters such as the code rate, the number of code words in the code, the distance properties of the code, the coding bound, and the coding gain.

A number of elementary concepts from linear algebra play a central role in coding theory. Expressed in terms appropriate for (n, k) block codes, the vector space V consists of the 2^n distinct n-tuples over the field of two elements $\{0, 1\}$. An (n, k) linear code is a set of 2^k n-tuples called code words which forms a subspace S of vector space V over Galois field $GF(2)$. If we select a set of k linearly independent vectors from V and from this set construct the set of all linear combinations of these vectors, the resulting set forms a subspace S of dimension k. If the set of vectors in V is orthogonal to every vector in S, this set of vectors is also a subspace of V and is called the null space of S. Since the dimension of S is k, the dimension of the null space is $n - k$. The null space of S is another linear code which consists of 2^{n-k} code words of length n and $n - k$ information bits. We call it an $(n, n - k)$ dual code. Furthermore, some topics that are related to the theory of error-correcting codes, such as groups, rings, and Galois fields in abstract algebra, will be discussed in Chap. 2.

The usual figure of merit for a coded system is the ratio of energy per information bit to noise spectral density E_b/N_0 that is required to achieve a given probability of error. The coding gain is defined for the amount of improvement that is achieved when a particular coding scheme is used. The usual method of determining coding gain is to plot the probability of error $P(E)$ versus E_b/N_0 for both coded and uncoded operations and to read the difference in required E_b/N_0 at a specified error rate. There are essentially two kinds of coding bounds, i.e., bounds on minimum distance and bounds on performance. The bounds on minimum distance are the Hamming bound and the Plotkin bound, which indicate the maximum possible minimum distance for a given code length and code rate, while the achievable lower bound on the minimum distance of the best code is the Gilbert-Varsharmov bound. When new codes are being developed, minimum distance bounds are often used to determine how close the code is to the best possible one. These bounds are fully discussed in Chap. 3. The bounds on performance indicate that the average performance of all block codes exhibits a probability of error that decreases exponentially with code length. These random coding bounds imply the existence of specific codes which do better than average. However, these bounds are not very useful for estimating the absolute performance of a code because good codes exhibit a probability of error that is considerably lower than that predicted by the bound. Chapters 5 through 8 are devoted to the analysis, design, and implementation of block codes for controlling errors in a noisy environment.

Modern coded systems are often designed to transmit at very high data rates. To protect such systems from error, designers often use an alternate coding scheme using convolutional codes in addition to block codes. A binary convolutional code can be generated by a linear finite-state machine connected to an m-stage shift register, n modulo-2 adders connected to some of the shift register stages, and a multiplexer that scans the output of the modulo-2 adders. This machine is called an (n, k) convolutional encoder with memory order m, sometimes denoted as an (n, k, m) convolutional encoder as a matter of convenience.

The input data to the convolutional encoder, called the information sequence, are shifted into and along the shift register k bits at a time, and the encoder output sequences are obtained by taking the convolution of the information sequence with the generator sequences of the code. Thus the multiplexer terminals use a process that scans in a serial fashion for the code word that is to be transmitted over the channel. The number of output bits for each k-bit input sequence is n. Since a convolutional encoder generates n encoded bits for every k information bits, the code rate is $R = k/n$, which is consistent with the definition of the code rate for a block code. Generally, k and n are small integers.

The generator matrix G of the code is a semi-infinite matrix which has an infinite number of rows and columns. This corresponds to the fact that the information and code sequences can be arbitrarily very large. However, for any practical application, there is a maximum allowable length L for which we often define the Lth truncation of a convolutional code. Thus the information sequence consists of kL bits, and the code sequence is represented by $n(m + L)$ bits. Therefore, the Lth truncation of an (n, k) convolutional code with memory order m can be viewed as an $(n(m + L), kL)$ linear block code. In this sense, convolutional codes may be thought of as a special class of linear block codes with superior properties which both facilitate decoding and improve performance. The constraint length is defined as $n_A = (m + 1)n$ because it is the maximum number of encoder outputs that can be affected by a single information bit.

With block codes, algebraic properties have been very important in constructing good classes of codes and in developing decoding algorithms. However, this is not the case with convolutional codes in general. Decoding of convolutional codes is probabilistic decoding. Generally, maximum-likelihood decoding by the Viterbi algorithm and sequential decoding by the Fano algorithm or the stack algorithm are called probabilistic decoding. On the other hand, threshold or majority-logic decoding is classified as algebraic decoding.

The Viterbi algorithm was proposed in 1967 and has been recognized as an attractive solution to maximum-likelihood decoding. Maximum-likelihood decoding is characterized as the determination of the shortest path through a topological structure called a code trellis; an efficient solution for that determination is the Viterbi algorithm. However, Viterbi decoding of convolutional codes is not practical for a long code with a larger constraint length n_A because the error probability increases exponentially with n_A. Since both complexity and decoding effort increase exponentially with n_A, convolutional codes with small constraint lengths must be used for Viterbi decoding because of the limitation of the encoder memory m. Another point related to the Viterbi algorithm is that 2^m computations must be performed per data sequence even when the noise effect is negligible, which results in wasted decoding effort.

In 1961, Wozencraft devised a decoding technique called sequential decoding, which has been the subject of research interest for over 20 years. In 1963, Fano introduced a new version of sequential decoding which is referred to as the Fano algorithm. Another version of sequential decoding, known as the stack or ZJ algorithm, was discovered independently by Zigangirov in 1966 and Jelinek in 1969. A major problem with sequential decoding is that the number of computations required to advance one node deeper into the code tree is a random variable. This characteristic strongly affects the complexity required to achieve a given level of performance. The performance of sequential decoding is slightly suboptimum, but its decoding effort is independent of the code constraint length. Thus sequential decoding can be used with codes of long constraint lengths. But its major drawback is that noise frames require a great deal of computation. Consequently, decoding times occasionally exceed some upper limit, causing data to be lost or erased.

In 1963, Massey introduced a less efficient but simpler-to-implement decoding method, called majority-logic or threshold decoding, which would be applicable to convolutional codes. The structure of convolutional codes is not algebraic but topological. But an algebraic approach can also be used for decoding convolutional codes. Conceptually and practically threshold decoding is closest to majority-logic decoding of block codes. Thus threshold decoding can eliminate the search aspects of Viterbi decoding and sequential decoding. Although threshold decoding is inferior in performance when compared to Viterbi or sequential decoding, decoder implementation is somewhat simpler.

1.3 Development of Coding Theory

The history of error-correction coding began in 1948 with the publication of a landmark paper by Claude Shannon. Since Shannon's work, a great deal of effort has been devoted to the problem of implementing encoder-decoder pairs for error control in noisy communications and digital computers. There are two different types of codes in common use today, i.e., block codes and convolutional codes.

During most of the 1950s, coding research was devoted primarily to block codes with a strong emphasis on algebraic approaches. In spite of diligent research, no better class of codes was found until the end of the decade. Hamming codes (1950) were the first class of linear block codes devised for single-error correction.

During the 1960s much effort was devoted to finding structures for classes of good codes that could produce an arbitrarily small probability of error, but not much progress was made. Meggitt (1961) devised a decoder that was applicable to any cyclic code, but refinements were necessary for practical implementation; the technique is known as error-trapping decoding. Although the idea of error-trapping decoding was introduced by Meggitt, it was refined by Kasami, Mitchel, Rudolph, and others. However, error-trapping decoding becomes ineffective if it is applied to long, high-rate codes that have a greater error-correcting capability. Fire (1959) discovered a large class of burst-error-correcting cyclic codes. Fire codes comprise the first class of cyclic codes that can be used for correcting burst errors and that can be decoded by the error-trapping technique. Burton (1969) introduced a class of phased-burst-error-correcting cyclic codes. Bose and Chaudhuri (1960) and Hocquenghem (1959) found a large class of multiple-error-correcting codes, called the BCH codes. The discovery of these powerful BCH codes led to a search for a practical decoding algorithm, which was devised by Peterson (1960), refined by Gorenstein and Zierler (1961), and extended by Berlekamp, Massey, Chien, Forney, and Lin. In addition, Reed and Solomon (1960) found an extremely important and practical class of nonbinary BCH codes that could cope with bursts of errors. The Reed-Solomon codes are at present coming into widespread use in many communication and computer storage systems. The majority-logic decoder is another effective device for decoding certain classes of cyclic block codes. Massey (1963) was the first to present a unified treatment for majority-logic decoding algorithms. The maximum-length codes and the difference-set codes are two small subclasses of one-step majority decodable cyclic codes. Finite-geometry codes (euclidean geometry codes and projective geometry codes) have their own structure and rules of orthogonalization for the multiple-step majority decodable codes. Rudolph (1967) investigated finite geometry codes which were extended and generalized by many coding researchers.

Another remarkable achievement made in the 1960s was the discovery of convolutional codes. Since 1955 when convolutional codes were first introduced by Elias, Wozencraft (1961) proposed sequential decoding as a practical decoding method, Fano (1963) introduced the Fano algorithm as a new version of sequential decoding, and another version of sequential decoding, called the ZJ stack algorithm, was independently discovered by Zigangirov (1966) and Jelinek (1969). In 1967, Viterbi proposed the Viterbi algorithm for maximum-likelihood decoding. Since then Omura (1969) and Forney (1972–1974) proved that the Viterbi algorithm is the best maximum-likelihood decoding algorithm for convolutional codes with shorter constraint lengths. In 1963, Massey proposed a less powerful but easy to implement decoding method called threshold decoding. It differs from Viterbi decoding and sequential decoding in that the final decision made on a given information block is based on only one constraint length of received blocks rather than on the entire received sequence. Since Hagelbarger (1959) introduced the concept of correcting burst errors of recurrent codes, many coding theorists have developed techniques for providing more