Bhargava, V.K. “Forward Error Correction Coding”
Mobile Communications Handbook
Ed. Suthan S. Suthersan
Boca Raton: CRC Press LLC, 1999

Forward Error Correction Coding

V.K. Bhargava
University of Victoria

I.J. Fair
University of Alberta

10.1 Introduction

In 1948, Claude Shannon issued a challenge to communications engineers by proving that communication systems could be made arbitrarily reliable as long as a fixed percentage of the transmitted signal was redundant [9]. He showed that limits exist only on the rate of communication and not its accuracy, and went on to prove that errorless transmission could be achieved in an additive white Gaussian noise (AWGN) environment with infinite bandwidth if the ratio of energy per data bit to noise power spectral density exceeds the **Shannon Limit**. He did not, however, indicate how this could be achieved. Subsequent research has led to a number of techniques that introduce redundancy to allow for correction of errors without retransmission. These techniques, collectively known as forward error correction (FEC) coding techniques, are used in systems where a reverse channel is not available for requesting retransmission, the delay with retransmission would be excessive, the expected number of errors would require a large number of retransmissions, or retransmission would be awkward to implement [10].

A simplified model of a digital communication system which incorporates FEC coding is shown in Fig. 10.1. The FEC code acts on a **discrete data channel** comprising all system elements between the encoder output and decoder input. The encoder maps the source data to q-ary code symbols which are modulated and transmitted. During transmission, this signal can be corrupted, causing errors to arise in the demodulated symbol sequence. The FEC decoder attempts to correct these errors and restore the original source data.

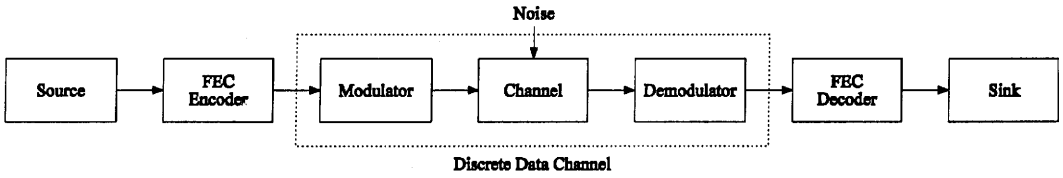


FIGURE 10.1: Block diagram of a digital communication system with forward error correction.

A demodulator which outputs only a value for the q-ary symbol received during each symbol interval is said to make **hard decisions.** In the **binary symmetric channel** (BSC), hard decisions are made on binary symbols and the probability of error is independent of the value of the symbol. One example of a BSC is the coherently demodulated binary phase-shift-keyed (BPSK) signal corrupted by AWGN. The conditional probability density functions which result with this system are depicted in Fig. 10.2. The probability of error is given by the area under the density functions that lies across the decision threshold, and is a function of the symbol energy E_s and the one-sided noise power spectral density N_0.

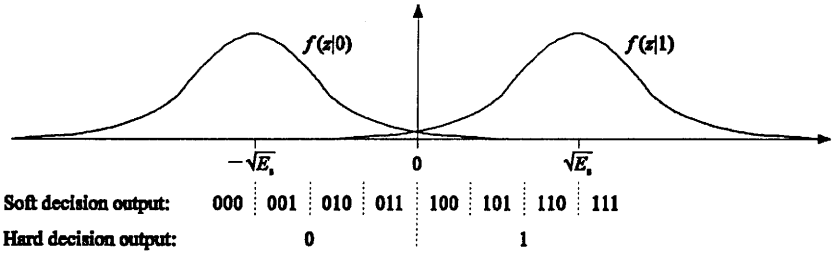


FIGURE 10.2: Hard and soft decision demodulation of a coherently demodulated BPSK signal corrupted by AWGN. $f(z \mid 1)$ and $f(z \mid 0)$ are the Gaussianly distributed conditional probability density functions at the threshold device.

Alternatively, the demodulator can make **soft decisions** or output an estimate of the symbol value along with an indication of its confidence in this estimate. For example, if the BPSK demodulator uses three-bit quantization, the two least significant bits can be taken as a confidence measure. Possible soft-decision thresholds for the BPSK signal are depicted in Fig. 10.2. In practice, there is little to be gained by using many soft-decision quantization levels.

Block and convolutional codes introduce redundancy by adding parity symbols to the message data. They map k source symbols to n code symbols and are said to have **code rate** $R = k/n$. With fixed information rates, this redundancy results in increased bandwidth and lower energy per transmitted symbol. At low signal-to-noise ratios, these codes cannot compensate for these impairments, and performance is degraded. At higher ratios of information symbol energy E_b to noise spectral density N_0, however, there is **coding gain** since the performance improvement offered by coding more than compensates for these impairments. Coding gain is usually defined as the reduction in required E_b/N_0 to achieve a specific error rate in an error-control coded system over one without coding. In contrast to block and convolutional codes, trellis-coded modulation introduces redundancy by expanding the size of the signal set rather than increasing the number of symbols transmitted, and so offers the advantages of coding to band-limited systems.

Each of these coding techniques is considered in turn. Following a discussion of **interleaving** and concatenated coding, this chapter gives an overview of a recent and significant advance in coding, the development of Turbo codes, and concludes with a brief overview of FEC applications.

10.2 Fundamentals of Block Coding

In block codes there is a one-to-one mapping between k-symbol source words and n-symbol codewords. With q-ary signalling, q^k out of the q^n possible n-tuples are valid code vectors. The set of all n-tuples forms a **vector space** in which the q^k code vectors are distributed. The **Hamming distance** between any two code vectors is the number of symbols in which they differ; the **minimum distance** $d_{\min}$ of the code is the smallest Hamming distance between any two codewords.

There are two contradictory objectives of block codes. The first is to distribute the code vectors in the vector space such that the distance between them is maximized. Then, if the decoder receives a corrupted vector, by evaluating the nearest valid code vector it will decode the correct word with high probability. The second is to pack the vector space with as many code vectors as possible to reduce the redundancy in transmission.

When code vectors differ in at least $d_{\min}$ positions, a decoder which evaluates the nearest code vector to each received word is guaranteed to correct up to t random symbol errors per word if

$$d_{\min} \geq 2t + 1 \tag{10.1}$$

Alternatively, all $q^n - q^k$ illegal words can be detected, including all error patterns with $d_{\min} - 1$ or fewer errors. In general, a block code can correct all patterns of t or fewer errors and detect all patterns of u or fewer errors provided that $u \geq t$ and

$$d_{\min} \geq t + u + 1 \tag{10.2}$$

If $q = 2$, knowledge of the positions of the errors is sufficient for their correction; if $q > 2$, the decoder must determine both the positions and values of the errors. If the demodulator indicates positions in which the symbol values are unreliable, the decoder can assume their value unknown and has only to solve for the value of these symbols. These positions are called **erasures.** A block code can correct up to t errors and v erasures in each word if

$$d_{\min} \geq 2t + v + 1 \tag{10.3}$$

10.3 Structure and Decoding of Block Codes

Shannon showed that the performance limit of codes with fixed code rate improves as the block length increases. As n and k increase, however, practical implementation requires that the mapping from message to code vector not be arbitrary but that an underlying structure to the code exist. The structures developed to date limit the error correcting capability of these codes to below what Shannon proved possible, on average, for a code with random codeword assignments. Although Turbo codes have made significant strides towards approaching the Shannon Limit, the search for good constructive codes continues.

A property which simplifies implementation of the coding operations is that of code linearity. A code is **linear** if the addition of any two code vectors forms another code vector, which implies that the code vectors form a subspace of the vector space of n-tuples. This subspace, which contains

the all-zero vector, is spanned by any set of k linearly independent code vectors. Encoding can be described as the multiplication of the information k-tuple by a **generator matrix** $\boldsymbol{G}$, of dimension $k \times n$, which contains these basis vectors as rows. That is, a message vector $\boldsymbol{m}_i$ is mapped to a code vector $\boldsymbol{c}_i$ according to

$$\boldsymbol{c}_i = \boldsymbol{m}_i \boldsymbol{G}, \qquad i = 0, 1, \ldots, q^k - 1 \tag{10.4}$$

where elementwise arithmetic is defined in the **finite field** GF(q). In general, this encoding procedure results in code vectors with nonsystematic form in that the values of the message symbols cannot be determined by inspection of the code vector. However, if $\boldsymbol{G}$ has the form $[\boldsymbol{I}_k, \boldsymbol{P}]$ where $\boldsymbol{I}_k$ is the $k \times k$ identity matrix and $\boldsymbol{P}$ is a $k \times (n-k)$ matrix of parity checks, then the k most significant symbols of each code vector are identical to the message vector and the code has **systematic** form. This notation assumes that vectors are written with their most significant or first symbols in time on the left, a convention used throughout this chapter.

For each generator matrix there is an $(n-k) \times k$ **parity check matrix** $\boldsymbol{H}$ whose rows are orthogonal to the rows in $\boldsymbol{G}$, i.e., $\boldsymbol{G}\boldsymbol{H}^T = \boldsymbol{0}$. If the code is systematic, $\boldsymbol{H} = [-\boldsymbol{P}^T, \boldsymbol{I}_{n-k}]$. Since all codewords are linear sums of the rows in $\boldsymbol{G}$, it follows that $\boldsymbol{c}_i \boldsymbol{H}^T = \boldsymbol{0}$ for all $i, i = 0, 1, \ldots, q^k - 1$, and that the validity of the demodulated vectors can be checked by performing this multiplication. If a codeword $\boldsymbol{c}$ is corrupted during transmission so that the hard-decision demodulator outputs the vector $\hat{\boldsymbol{c}} = \boldsymbol{c} + \boldsymbol{e}$, where $\boldsymbol{e}$ is a nonzero error pattern, the result of this multiplication is an $(n-k)$-tuple that is indicative of the validity of the sequence. This result, called the **syndrome** $\boldsymbol{s}$, is dependent only on the error pattern since

$$\boldsymbol{s} = \hat{\boldsymbol{c}}\boldsymbol{H}^T = (\boldsymbol{c} + \boldsymbol{e})\boldsymbol{H}^T = \boldsymbol{c}\boldsymbol{H}^T + \boldsymbol{e}\boldsymbol{H}^T = \boldsymbol{e}\boldsymbol{H}^T \tag{10.5}$$

If the error pattern is a code vector, the errors go undetected. For all other error patterns, however, the syndrome is nonzero. Since there are $q^{n-k} - 1$ nonzero syndromes, $q^{n-k} - 1$ error patterns can be corrected. When these patterns include all those with t or fewer errors and no others, the code is said to be a **perfect code.** Few codes are perfect; most codes are capable of correcting some patterns with more than t errors. **Standard array decoders** use lookup tables to associate each syndrome with an error pattern but become impractical as the block length and number of parity symbols increases. Algebraic decoding algorithms have been developed for codes with stronger structure. These algorithms are simplified with imperfect codes if the patterns corrected are limited to those with t or fewer errors, a simplification called **bounded distance decoding.**

Cyclic codes are a subclass of linear block codes with an algebraic structure that enables encoding to be implemented with a linear feedback shift register and decoding to be implemented without a lookup table. As a result, most block codes in use today are cyclic or are closely related to cyclic codes. These codes are best described if vectors are interpreted as polynomials and the arithmetic follows the rules for polynomials where the elementwise operations are defined in GF(q). In a cyclic code, all codeword polynomials are multiples of a **generator polynomial** $g(x)$ of degree $n-k$. This polynomial is chosen to be a divisor of $x^n - 1$ so that a cyclic shift of a code vector yields another code vector, giving this class of codes its name. A message polynomial $m_i(x)$ can be mapped to a codeword polynomial $c_i(x)$ in nonsystematic form as

$$c_i(x) = m_i(x)g(x), \qquad i = 0, 1, \ldots, q^k - 1 \tag{10.6}$$

In systematic form, codeword polynomials have the form

$$c_i(x) = m_i(x)x^{n-k} - r_i(x), \qquad i = 0, 1, \ldots, q^k - 1 \tag{10.7}$$

where $r_i(x)$ is the remainder of $m_i(x)x^{n-k}$ divided by $g(x)$. Polynomial multiplication and division can be easily implemented with shift registers [5].

The first step in decoding the demodulated word is to determine if the word is a multiple of $g(x)$. This is done by dividing it by $g(x)$ and examining the remainder. Since polynomial division is a linear operation, the resulting syndrome $s(x)$ depends only on the error pattern. If $s(x)$ is the all-zero polynomial, transmission is errorless or an undetectable error pattern has occurred. If $s(x)$ is nonzero, at least one error has occurred. This is the principle of the **cyclic redundancy check** (CRC). It remains to determine the most likely error pattern that could have generated this syndrome.

Single error correcting binary codes can use the syndrome to immediately locate the bit in error. More powerful codes use this information to determine the locations and values of multiple errors. The most prominent approach of doing so is with the iterative technique developed by Berlekamp. This technique, which involves computing an error-locator polynomial and solving for its roots, was subsequently interpreted by Massey in terms of the design of a minimum-length shift register. Once the location and values of the errors are known, Chien's search algorithm efficiently corrects them. The implementation complexity of these decoders increases only as the square of the number of errors to be corrected [4] but does not generalize easily to accommodate soft-decision information. Other decoding techniques, including Chase's algorithm and threshold decoding, are easier to implement with soft-decision input [6]. Berlekamp's algorithm can be used in conjunction with transform-domain decoding, which involves transforming the received block with a finite field Fourier-like transform and solving for errors in the transform domain. Since the implementation complexity of these decoders depends on the block length rather than the number of symbols corrected, this approach results in simpler circuitry for codes with high redundancy [13].

Other block codes have also been constructed, including codes that are based on transform-domain spectral properties, codes that are designed specifically for correction of burst errors, and codes that are decodable with straightforward threshold or majority logic decoders [5, 6, 7].

10.4 Important Classes of Block Codes

When errors occur independently, Bose–Chaudhuri–Hocquenghem (BCH) codes provide one of the best performances of known codes for a given block length and code rate. They are cyclic codes with $n = q^m - 1$, where m is any integer greater than 2. They are designed to correct up to t errors per word and so have **designed distance** $d = 2t + 1$; the minimum distance may be greater. Generator polynomials for these codes are listed in many texts, including [6]. These polynomials are of degree less than or equal to mt, and so $k \geq n - mt$. BCH codes can be shortened to accommodate system requirements by deleting positions for information symbols.

Some subclasses of these codes are of special interest. Hamming codes are perfect single error correcting binary BCH codes. Full length codes have $n = 2^m - 1$ and $k = n - m$ for any m greater than 2. The duals of these codes are maximal-length codes, with $n = 2^m - 1$, $k = m$, and $d_{\min} = 2^{m-1}$. All $2^m - 1$ nonzero code vectors in these codes are cyclic shifts of a single nonzero code vector. Reed–Solomon (RS) codes are nonbinary BCH codes defined over GF(q), where q is often taken as a power of two so that symbols can be represented by a sequence of bits. In these cases, correction of even a single symbol allows for correction of a burst of bit errors. The block length is $n = q - 1$, and the minimum distance $d_{\min} = 2t + 1$ is achieved using only $2t$ parity symbols. Since RS codes meet the Singleton bound of $d_{\min} \leq n - k + 1$, they have the largest possible minimum distance for these values of n and k and are called **maximum distance separable** codes.

The Golay codes are the only nontrivial perfect codes that can correct more than one error. The (11, 6) ternary Golay code has minimum distance 5. The (23, 12) binary code is a triple error correcting BCH code with $d_{\min} = 7$. To simplify implementation, it is often extended to a (24, 12) code through the addition of an extra parity bit. The extended code has $d_{\min} = 8$.

The (23, 12) Golay code is also a binary quadratic residue code. These cyclic codes have prime length of the form $n = 8m \pm 1$, with $k = (n+1)/2$ and $d_{\min} \geq \sqrt{n}$. Some of these codes are as good as the best codes known with these values of n and k, but it is unknown if there are good quadratic residue codes with large n [5].

Reed-Muller codes are equivalent to binary cyclic codes with an additional overall parity bit. For any m, the rth-order Reed-Muller code has $n = 2^m$, $k = \Sigma_{i=0}^{r} \binom{m}{i}$, and $d_{\min} = 2^{m-r}$. The rth-order and $(m - r - 1)$th-order codes are duals, and the first-order codes are similar to maximal-length codes. These codes, and the closely related Euclidean geometry and projective geometry codes, can be decoded with threshold decoding.

The performance of several of these block codes is shown in Fig. 10.3 in terms of decoded bit error probability vs. E_b/N_0 for systems using coherent, hard-decision demodulated BPSK signalling. Many other block codes have also been developed, including Goppa codes, quasicyclic codes, burst error correcting Fire codes, and other lesser known codes.

10.5 Principles of Convolutional Coding

Convolutional codes map successive information k-tuples to a series of n-tuples such that the sequence of n-tuples has distance properties that allow for detection and correction of errors. Although these codes can be defined over any alphabet, their implementation has largely been restricted to binary signals, and only binary convolutional codes are considered here.

In addition to the code rate $R = k/n$, the **constraint length** K is an important parameter for these codes. Definitions vary; we will use the definition that K equals the number of k-tuples that affect formation of each n-tuple during encoding. That is, the value of an n-tuple depends on the k-tuple that arrives at the encoder during that encoding interval as well as the $K - 1$ previous information k-tuples.

Binary convolutional encoders can be implemented with kK-stage shift registers and n modulo-2 adders, an example of which is given in Fig. 10.4(a) for a rate 1/2, constraint length 3 code. The encoder shifts in a new k-tuple during each encoding interval and samples the outputs of the adders sequentially to form the coded output.

Although connection diagrams similar to that of Fig. 10.4(a) completely describe the code, a more concise description can be given by stating the values of n, k, and K and giving the adder connections in the form of vectors or polynomials. For instance, the rate 1/2 code has the generator vectors $\boldsymbol{g}_1 = 111$ and $\boldsymbol{g}_2 = 101$, or equivalently, the generator polynomials $g_1(x) = x^2 + x + 1$ and $g_2(x) = x^2 + 1$. Alternatively, a convolutional code can be characterized by its impulse response, the coded sequence generated due to input of a single logic-1. It is straightforward to verify that the circuit in Fig. 10.4(a) has the impulse response 111011. Since modulo-2 addition is a linear operation, convolutional codes are linear, and the coded output can be viewed as the convolution of the input sequence with the impulse response, hence the name of this coding technique. Shifted versions of the impulse response or generator vectors can be combined to form an infinite-order generator matrix which also describes the code.

Shift register circuits can be modeled as finite state machines. A Mealy machine description of a convolutional encoder requires $2^{k(K-1)}$ states, each describing a different value of the $K - 1$ k-tuples which have most recently entered the shift register. Each state has 2^k exit paths which correspond to the value of the incoming k-tuple. A state machine description for the rate 1/2 encoder depicted in Fig. 10.4(a) is given in Fig. 10.4(b). States are labeled with the contents of the two leftmost register stages; edges are labeled with information bit values and their corresponding coded output.

The dimension of time is added to the description of the encoder with tree and trellis diagrams.

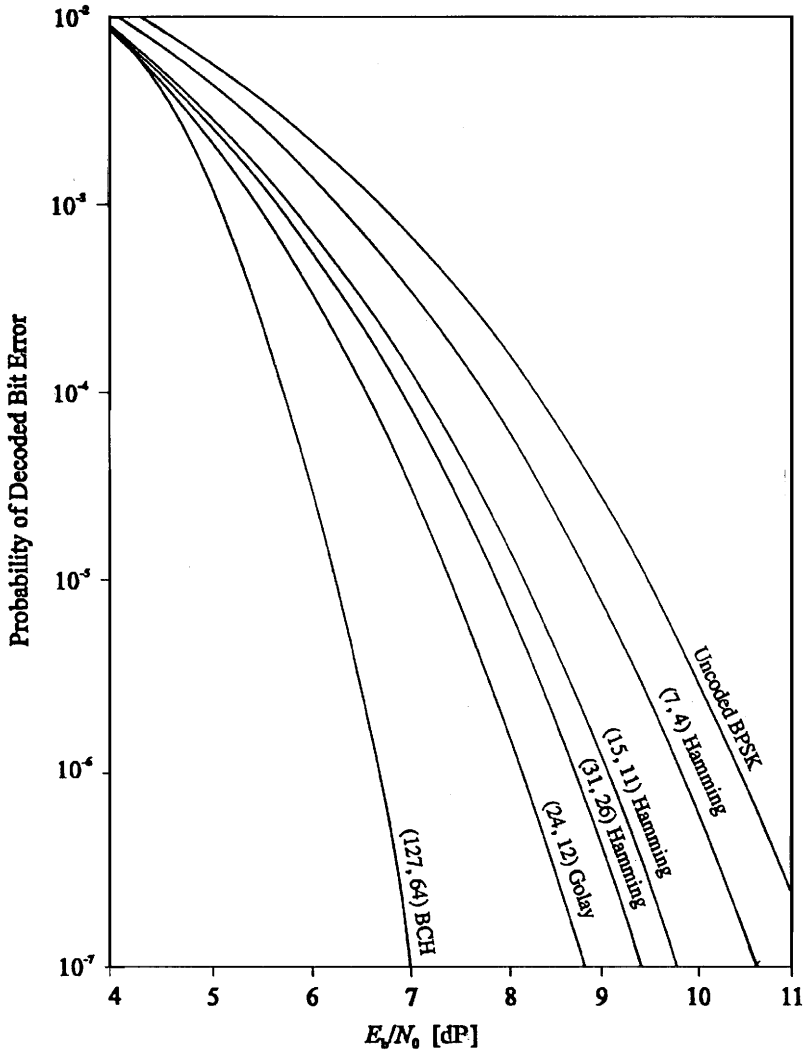


FIGURE 10.3: Block code performance. *Source:* Sklar, B., 1988, *Digital Communications: Fundamentals and Applications*, © 1988, p. 300. Reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, NJ.

The tree diagram for the rate 1/2 convolutional code is given in Fig. 10.4(c), assuming the shift register is initially clear. Each node represents an encoding interval, from which the upper branch is taken if the input bit is a 0 and the lower branch is taken if the input bit is a 1. Each branch is labeled with the corresponding output bit sequence. A drawback of the tree representation is that it grows without bound as the length of the input sequence increases. This is overcome with the trellis diagram depicted in Fig. 10.4(d), Again, encoding results in left-to-right movement, where the upper of the two branches is taken whenever the input is a 0, the lower branch is taken when the input is a 1, and the output is the bit sequence which weights the branch taken. Each level of nodes corresponds to a state of the encoder as shown on the left-hand side of the diagram.

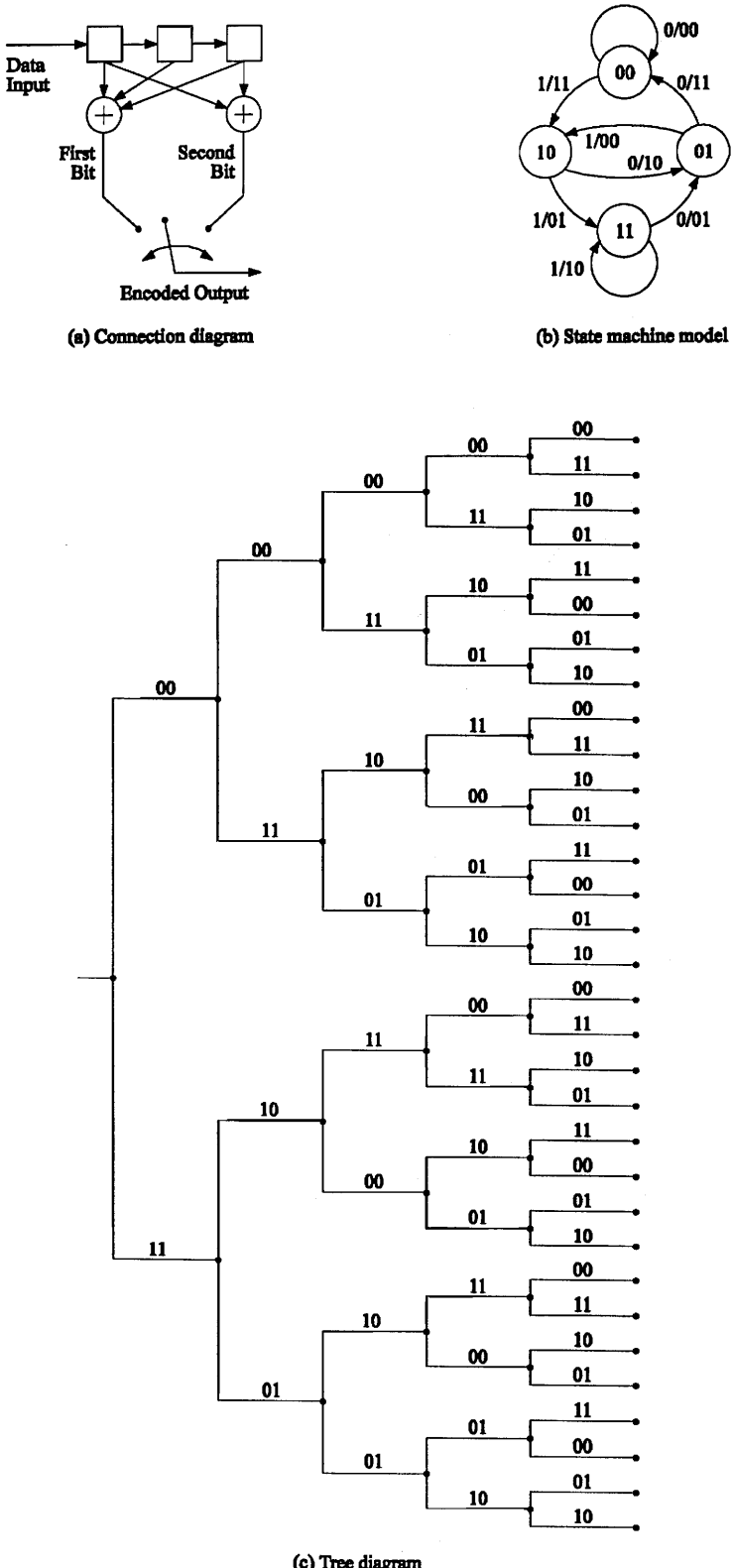
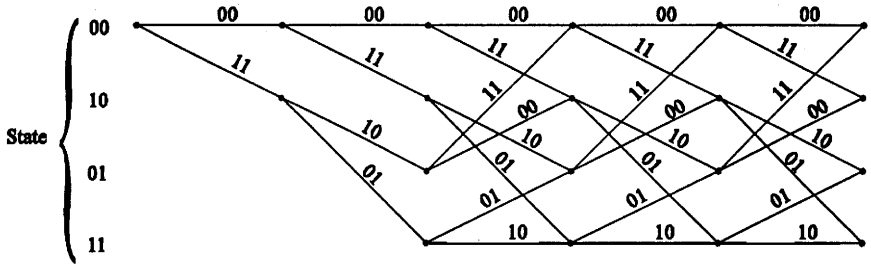


FIGURE 10.4: A rate 1/2, constraint length 3 convolutional code.



(d) Trellis diagram

FIGURE 10.4: (*Continued*).

If the received sequence contains errors, it may no longer depict a valid path through the tree or trellis. It is the job of the decoder to determine the original path. In doing so, the decoder does not so much correct errors as find the closest valid path to the received sequence. As a result, the error correcting capability of a convolutional code is more difficult to quantify than that of a block code; it depends on how valid paths differ. One measure of this difference is the **column distance** $d_c(i)$, the minimum Hamming distance between all coded sequences generated over i encoding intervals which differ in the first interval. The nondecreasing sequence of column distance values is the **distance profile** of the code. The column distance after K intervals is the minimum distance of the code and is important for evaluating the performance of a code that uses threshold decoding. As i increases, $d_c(i)$ approaches the **free distance** of the code, d_{free}, which is the minimum Hamming distance in the set of arbitrarily long paths that diverge and then remerge in the trellis.

With maximum likelihood decoding, convolutional codes can generally correct up to t errors within three to five constraint lengths, depending on how the errors are distributed, where

$$d_{\text{free}} \geq 2t + 1 \tag{10.8}$$

The free distance can be calculated by exhaustively searching for the minimum-weight path that returns to the all-zero state, or evaluating the term of lowest degree in the generating function of the code.

The objective of a convolutional code is to maximize these distance properties. They generally improve as the constraint length of the code increases, and nonsystematic codes generally have better properties than systematic ones. Good codes have been found by computer search and are tabulated in many texts, including [6]. Convolutional codes with high code rate can be constructed by **puncturing** or periodically deleting coded symbols from a low rate code. A list of low rate codes and perforation matrices that result in good high rate codes can be found in many sources, including [13]. The performance of good punctured codes approaches that of the best convolutional codes known with similar rate, and decoder implementation is significantly less complex.

Convolutional codes can be **catastrophic,** having the potential to generate an unlimited number of decoded bit errors in response to a finite number of errors in the demodulated bit sequence. Catastrophic error propagation is avoided if the code has generator polynomials with a greatest common divisor of the form x^a for any a or, equivalently, if there are no closed-loop paths in the state diagram with all-zero output other than the one taken with all-zero input. **Systematic codes** are not catastrophic.

10.6 Decoding of Convolutional Codes

In 1967, Viterbi developed a maximum likelihood decoding algorithm that takes advantage of the trellis structure to reduce the complexity of the evaluation. This algorithm has become known as the **Viterbi algorithm.** With each received n-tuple, the decoder computes a **metric** or measure of likelihood for all paths that could have been taken during that interval and discards all but the most likely to terminate on each node. An arbitrary decision is made if path metrics are equal. The metrics can be formed using either hard or soft decision information with little difference in implementation complexity.

If the message has finite length and the encoder is subsequently flushed with zeros, a single decoded path remains. With a BSC, this path corresponds to the valid code sequence with minimum Hamming distance from the demodulated sequence. Full-length decoding becomes impractical as the length of the message sequence increases. The most likely paths tend to have a common stem, however, and selecting the trace value four or five times the constraint length prior to the present decoding depth results in near-optimum performance. Since the number of paths examined during each interval increases exponentially with the constraint length, the Viterbi algorithm also becomes impractical for codes with large constraint length. To date, Viterbi decoding has been implemented for codes with constraint lengths up to ten. Other decoding techniques, such as sequential and threshold decoding, can be used with larger constraint lengths.

Sequential decoding was proposed by Wozencraft, and the most widely used algorithm was developed by Fano. Rather than tracking multiple paths through the trellis, the sequential decoder operates on a single path while searching the code tree for a path with high probability. It makes tentative decisions regarding the transmitted sequence, computes a metric between its proposed path and the demodulated sequence, and moves forward through the tree as long as the metric indicates that the path is likely. If the likelihood of the path becomes low, the decoder moves backward, searching other paths until it finds one with high probability. The number of computations involved in this procedure is almost independent of the constraint length and is typically quite small, but it can be highly variable, depending on the channel. Buffers must be provided to store incoming sequences as the decoder searches the tree. Their overflow is a significant limiting factor in the performance of these decoders.

Figure 10.5 compares the performance of the Viterbi and sequential decoding algorithms for several convolutional codes operating on coherently demodulated BPSK signals corrupted by AWGN. Other decoding algorithms have also been developed, including syndrome decoding methods such as table look-up feedback decoding and threshold decoding [6]. These algorithms are easily implemented but offer suboptimal performance. Techniques such as the one discussed by [1] have been developed to support both soft input and soft output, but these decoding techniques typically increase decoder complexity.

10.7 Trellis-Coded Modulation

Trellis-coded modulation (TCM) has received considerable attention since its development by Ungerboeck in the late 1970s [11]. Unlike block and convolutional codes, TCM schemes achieve coding gain by increasing the size of the signal alphabet and using multilevel/phase signalling. Like convolutional codes, sequences of coded symbols are restricted to certain valid patterns. In TCM, these patterns are chosen to have large Euclidean distance from one another so that a large number of corrupted sequences can be corrected. The Viterbi algorithm is often used to decode these sequences. Since the symbol transmission rate does not increase, coded and uncoded signals require the same transmis-

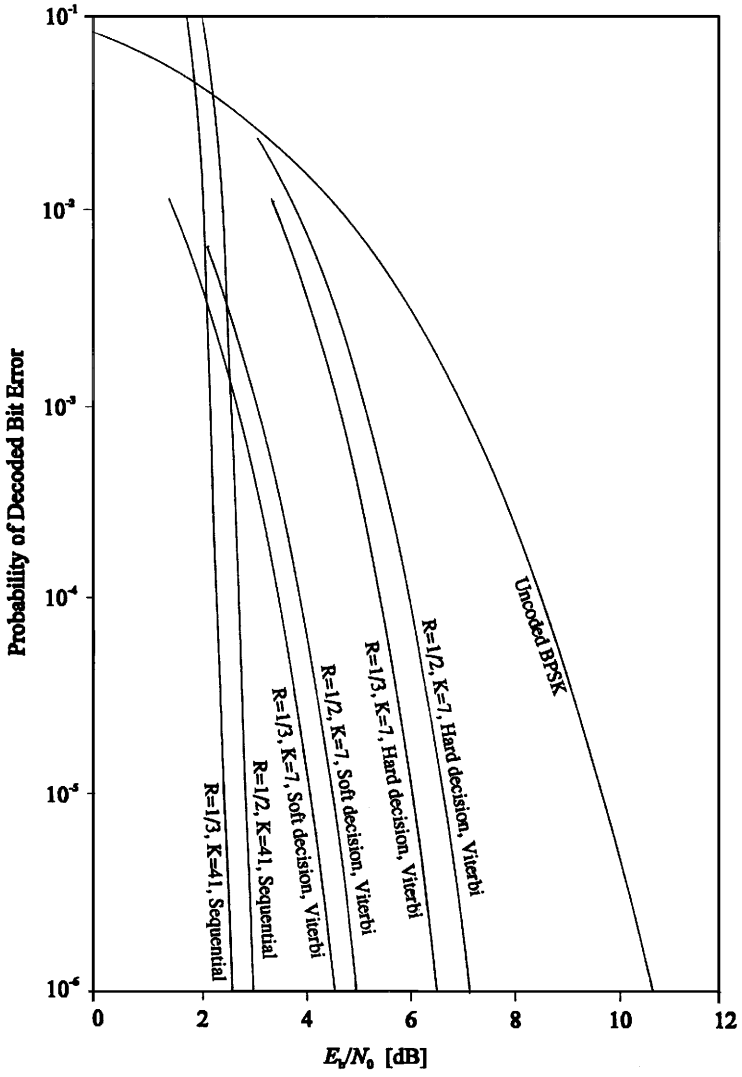


FIGURE 10.5: Convolutional code performance. *Source:* Omura, J.K. and Levitt, B.K., © 1982 IEEE, "Coded Error Probability Evaluation for Antijam Communication Systems," *IEEE Trans. Commun.*, vol. COM-30, no. 5, pp. 896–903. Reprinted by permission of IEEE.

sion bandwidth. If transmission power is held constant, the signal constellation of the coded signal is denser. The loss in symbol separation, however, is more than overcome by the error correction capability of the code.

Ungerboeck investigated the increase in channel capacity that can be obtained by increasing the size of the signal set and restricting the pattern of transmitted symbols, and concluded that almost all of the additional capacity can be gained by doubling the number of points in the signal constellation. This is accomplished by encoding the binary data with a rate $R = k/(k+1)$ code and mapping sequences of $k+1$ coded bits to points in a constellation of 2^{k+1} symbols. For example, the rate $2/3$

encoder of Fig. 10.6(a) encodes pairs of source bits to three coded bits. Figure 10.6(b) depicts one stage in the trellis of the coded output where, as with the convolutional code, the state of the encoder is defined by the values of the two most recent bits to enter the shift register. Note that unlike the trellis for the convolutional code, this trellis contains parallel paths between nodes.

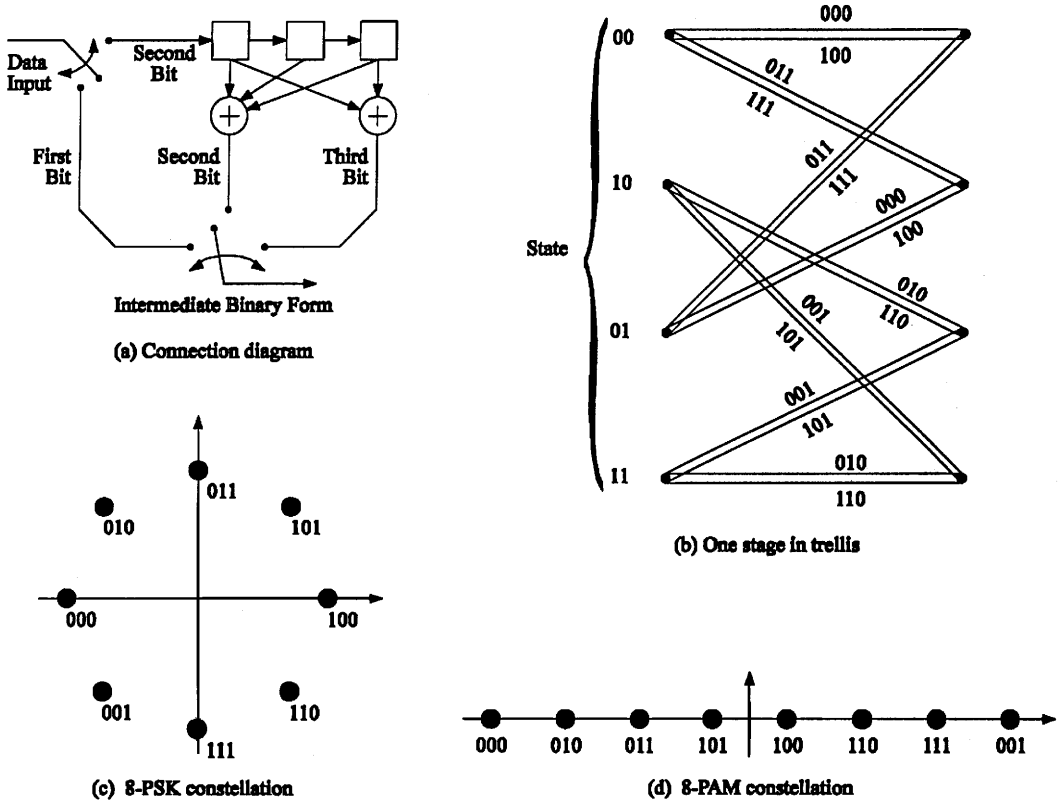


FIGURE 10.6: Rate 2/3 trellis-coded modulation.

The key to improving performance with TCM is to map the coded bits to points in the signal space such that the Euclidean distance between transmitted sequences is maximized. A method that ensures improved Euclidean distance is the method of **set partitioning.** This involves separating all parallel paths on the trellis with maximum distance and assigning the next greatest distance to paths that diverge from or merge onto the same node. Figures 10.6(c) and 10.6(d) give examples of mappings for the rate 2/3 code with 8-PSK and 8-PAM signal constellations, respectively.

As with convolutional codes, the free distance of a TCM code is defined as the minimum distance between paths through the trellis, where the distance of concern is now Euclidean distance rather than Hamming distance. The free distance of an uncoded signal is defined as the distance between the closest signal points. When coded and uncoded signals have the same average power, the coding

gain of the TCM system is defined as

$$\text{coding gain} = 20 \log_{10} \left(\frac{d_{\text{free, coded}}}{d_{\text{free, uncoded}}} \right) \tag{10.9}$$

It can be shown that the simple, rate $2/3$ 8 phase-shift keying (PSK) and 8 pulse-amplitude modulation (PAM) TCM systems provide gains of 3 dB and 3.3 dB, respectively, [6]. More complex TCM systems yield gains up to 6 dB. Tables of good codes are given in [11].

10.8 Additional Measures

When the demodulated sequence contains bursts of errors, the performance of codes designed to correct independent errors improves if coded sequences are **interleaved** prior to transmission and deinterleaved prior to decoding. Deinterleaving separates the burst errors, making them appear more random and increasing the likelihood of accurate decoding. It is generally sufficient to interleave several block lengths of a block coded signal or several constraint lengths of a convolutionally encoded signal. Block interleaving is the most straightforward approach, but delay and memory requirements are halved with convolutional and helical interleaving techniques. Periodicity in the way sequences are combined is avoided with pseudorandom interleaving.

Serially **concatenated codes,** first investigated by Forney, use two levels of coding to achieve a level of performance with less complexity than a single coding stage would require. The inner code interfaces with the modulator and demodulator and corrects the majority of the errors; the outer code corrects errors that appear at the output of the inner-code decoder. A convolutional code with Viterbi decoding is usually chosen as the inner code, and an RS code is often chosen as the outer code due to its ability to correct the bursts of bit errors which can result with incorrect decoding of trellis-coded sequences. Interleaving and deinterleaving outer-code symbols between coding stages offers further protection against the burst error output of the inner code.

Product codes effectively place the data in a two dimensional array and use FEC techniques over both the rows and columns of this array. Not only do these codes result in error protection in two dimensions, but the manner in which the array is constructed can offer advantages similar to those achieved through interleaving.

10.9 Turbo Codes

The most recent significant achievement in FEC coding is the development of **Turbo codes** [3]. The principle of this coding technique is to encode the data with two or more **constituent codes** concatenated in parallel form. The received sequence is decoded in an iterative, serial approach using soft-input, soft-output decoders. This iterative decoding approach involves feedback of information in a manner similar to processes within the turbo engine, giving this coding technique its name.

Turbo codes effectively result in the construction of relatively long codewords with few codewords being close in terms of Hamming distance, while at the same time constraining the implementation complexity of the decoder to practical limits. The first Turbo codes developed used recursive systematic convolutional codes as the constituent codes, and punctured them to improve the code rate. The use of other constituent codes has since been considered. Two or more of these codes are concatenated in parallel,where code concatenation is combined with interleaving in order to increase the independence of the data sequences encoded by the constituent encoders. This apparent increase in randomness, implemented with simple interleavers, is an important contributing factor to the excellent performance of the decoders.

As in other multi-stage coding techniques, the complexity of the decoder is limited through use of separate decoding stages for each constituent code. The input to the first stage is the soft output of the demodulator for a finite-length received symbol sequence. Subsequent stages use both the demodulator output and an output of the previous decoding stage which is indicative of the reliability of the symbols. This information, gleaned from soft-output decoders, is called **extrinsic information**.

Decoding proceeds by iterating through constituent decoders, each forwarding updated extrinsic information to the next decoder, until a predefined number of iterations has been completed or the extrinsic information indicates that high reliability has been achieved. This approach results in very good performance at low values of E_b/N_0. Simulations have demonstrated error rates of 10^{-5} at signal-to-noise ratios appreciably less than 1 dB. At higher values of E_b/N_0, however, the performance curves can exhibit flattening if constituent codes are chosen in a manner that results in an overall small Hamming distance for the code.

Although this coding technique has shown great promise, there remains considerable work with regard to optimizing code parameters. Great strides have been made over the last few years in understanding the structure of these codes and relating them to serially concatenated and product codes, but many researchers are still examining these codes in order to advance their development. With this research will come optimization of the Turbo code process and application of these codes in various communication systems.

10.10 Applications

FEC coding remained of theoretical interest until advances in digital technology and improvements in decoding algorithms made their implementation possible. It has since become an attractive alternative to improving other system components or boosting transmission power. FEC codes are commonly used in digital storage systems, deep-space and satellite communication systems, terrestrial radio and band limited wireline systems, and have also been proposed for fiber optic transmission. Accordingly, the theory and practice of error correcting codes now occupies a prominent position in the field of communications engineering.

Deep-space systems began using forward error correction in the early 1970s to reduce transmission power requirements, and used multiple error correcting RS codes for the first time in 1977 to protect against corruption of compressed image data in the Voyager missions [12]. The Consultative Committee for Space Data Systems (CCSDS) has since recommended use of a concatenated coding system which uses a rate 1/2, constraint length 7 convolutional inner code and a (255, 223) RS outer code.

Coding is now commonly used in satellite systems to reduce power requirements and overall hardware costs and to allow closer orbital spacing of geosynchronous satellites [2]. FEC codes play integral roles in the VSAT, MSAT, INTELSAT, and INMARSAT systems [13]. Further, a (31, 15) RS code is used in the joint tactical information distribution system (JTIDS), a (7, 2) RS code is used in the air force satellite communication system (AFSATCOM), and a (204, 192) RS code has been designed specifically for satellite time division multiple access (TDMA) systems. Another code designed for military applications involves concatenation of a Golay and RS code with interleaving to ensure an imbalance of 1's and 0's in the transmitted symbol sequence and enhance signal recovery under severe noise and interference [2].

TCM has become commonplace in transmission of data over voiceband telephone channels. Modems developed since 1984 use trellis coded QAM modulation to provide robust communication at rates above 9.6 kb/s. Various coding techniques are used in the new digital cellular and

personal communication standards, with an emphasis on convolutional and cyclic redundancy check codes [8].

FEC codes have also been widely used in digital recording systems, most prominently in the compact disc digital audio system. This system uses two levels of coding and interleaving in the cross-interleaved RS coding (CIRC) system to correct errors that result from disc imperfections and dirt and scratches which accumulate during use. Steps are also taken to mute uncorrectable sequences [12].

Defining Terms

Binary symmetric channel: A memoryless discrete data channel with binary signalling, hard-decision demodulation, and channel impairments that do not depend on the value of the symbol transmitted.

Bounded distance decoding: Limiting the error patterns which are corrected in an imperfect code to those with t or fewer errors.

Catastrophic code: A convolutional code in which a finite number of code symbol errors can cause an unlimited number of decoded bit errors.

Code rate: The ratio of source word length to codeword length, indicative of the amount of information transmitted per encoded symbol.

Coding gain: The reduction in signal-to-noise ratio required for specified error performance in a block or convolutional coded system over an uncoded system with the same information rate, channel impairments, and modulation and demodulation techniques. In TCM, the ratio of the squared free distance in the coded system to that of the uncoded system.

Column distance: The minimum Hamming distance between convolutionally encoded sequences of a specified length with different leading n-tuples.

Constituent codes: Two or more FEC codes that are combined in concatenated coding techniques.

Cyclic code: A block code in which cyclic shifts of code vectors are also code vectors.

Cyclic redundancy check: When the syndrome of a cyclic block code is used to detect errors.

Designed distance: The guaranteed minimum distance of a BCH code designed to correct up to t errors.

Discrete data channel: The concatenation of all system elements between FEC encoder output and decoder input.

Distance profile: The minimum Hamming distance after each encoding interval of convolutionally encoded sequences which differ in the first interval.

Erasure: A position in the demodulated sequence where the symbol value is unknown.

Extrinsic information: The output of a constituent soft decision decoder that is forwarded as input to the next decoding stage in iterative decoding of Turbo codes.

Finite field: A finite set of elements and operations of addition and multiplication that satisfy specific properties. Often called Galois fields and denoted GF(q), where q is the number of elements in the field. Finite fields exist for all q which are prime or the power of a prime.

Free distance: The minimum Hamming weight of convolutionally encoded sequences that diverge and remerge in the trellis. Equals the maximum column distance and the limiting value of the distance profile.

Generator matrix: A matrix used to describe a linear code. Code vectors equal the information vectors multiplied by this matrix.

Generator polynomial: The polynomial that is a divisor of all codeword polynomials in a cyclic block code; a polynomial that describes circuit connections in a convolutional encoder.

Hamming distance: The number of symbols in which codewords differ.

Hard decision: Demodulation that outputs only a value for each received symbol.

Interleaving: Shuffling the coded bit sequence prior to modulation and reversing this operation following demodulation. Used to separate and redistribute burst errors over several codewords (block codes) or constraint lengths (trellis codes) for higher probability of correct decoding by codes designed to correct random errors.

Linear code: A code whose code vectors form a vector space. Equivalently, a code where the addition of any two code vectors forms another code vector.

Maximum distance separable: A code with the largest possible minimum distance given the block length and code rate. These codes meet the Singleton bound of $d_{\min} \leq n - k + 1$.

Metric: A measure of goodness against which items are judged. In the Viterbi algorithm, an indication of the probability of a path being taken given the demodulated symbol sequence.

Minimum distance: In a block code, the smallest Hamming distance between any two codewords. In a convolutional code, the column distance after K intervals.

Parity check matrix: A matrix whose rows are orthogonal to the rows in the generator matrix of a linear code. Errors can be detected by multiplying the received vector by this matrix.

Perfect code: A t error correcting (n, k) block code in which $q^{n-k} - 1 = \Sigma_{i=1}^{t} \binom{n}{i}$.

Puncturing: Periodic deletion of code symbols from the sequence generated by a convolutional encoder for purposes of constructing a higher rate code. Also, deletion of parity bits in a block code.

Set partitioning: Rules for mapping coded sequences to points in the signal constellation that always result in a larger Euclidean distance for a TCM system than an uncoded system, given appropriate construction of the trellis.

Shannon Limit: The ratio of energy per data bit E_b to one-sided noise power spectral density N_0 in an AWGN channel above which errorless transmission is possible when bandwidth limitations are not placed on the signal and transmission is at channel capacity. This limit has the value $\ln 2 = 0.693 = -1.6$ dB.

Soft decision: Demodulation that outputs an estimate of the received symbol value along with an indication of the reliability of this value. Usually implemented by quantizing the received signal to more levels than there are symbol values.

Standard array decoding: Association of an error pattern with each syndrome by way of a lookup table.

Syndrome: An indication of whether or not errors are present in the demodulated symbol sequence.

Systematic code: A code in which the values of the message symbols can be identified by inspection of the code vector.

Vector space: An algebraic structure comprised of a set of elements in which operations of vector addition and scalar multiplication are defined. For our purposes, a set of n-tuples consisting of symbols from GF(q) with addition and multiplication defined in terms of elementwise operations from this finite field.

Viterbi algorithm: A maximum-likelihood decoding algorithm for trellis codes that discards low-probability paths at each stage of the trellis, thereby reducing the total number of paths that must be considered.

References

[1] Bahl, L.R., Cocke, J., Jelinek, F., and Raviv, J., Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate. *IEEE Transactions on Information Theory*, 20, 248–287, 1974.

[2] Berlekamp, E.R., Peile, R.E., and Pope, S.P., The application of error control to communications. *IEEE Commun. Mag.*, 25(4), 44–57, 1987.

[3] Berrou, C., Glavieux, A., and Thitimajshima, P., Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes. *Proceedings of ICC'93*, Geneva, Switzerland, 1064–1070, 1993. Later expanded and published as: Berrou, C., Glavieux, A., 1996. Near Optimum Error Correcting Coding and Decoding. *IEEE Transactions on Communications*, 44(10), 1261–1271, 1996.

[4] Bhargava, V.K., Forward error correction schemes for digital communications. *IEEE Commun. Mag.*, 21(1), 11–19, 1983.

[5] Blahut, R.E., *Theory and Practice of Error Control Codes*, Addison-Wesley, Reading, MA, 1983.

[6] Clark, G.C. Jr. and Cain, J.B., *Error Correction Coding for Digital Communications*, Plenum Press, New York, 1981.

[7] Lin, S. and Costello, D.J. Jr., *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[8] Rappaport, T.S., *Wireless Communications, Principles and Practice*, Prentice-Hall and IEEE Press, NJ, 1996.

[9] Shannon, C.E., A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3), 379–423 and 623–656, 1948.

[10] Sklar, B., *Digital Communications: Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[11] Ungerboeck, G., Trellis-coded modulation with redundant signal sets. *IEEE Commun. Mag.*, 25(2), 5–11 and 12–21, 1987.

[12] Wicker, S.B. and Bhargava, V.K., *Reed-Solomon Codes and Their Applications*, IEEE Press, NJ, 1994.

[13] Wu, W.W., Haccoun, D., Peile, R., and Hirata, Y., Coding for satellite communication. *IEEE J. Selected Areas in Commun.*, SAC-5(4), 724–748, 1987.

Further Information

There is now a large amount of literature on the subject of FEC coding. An introduction to the philosophy and limitations of these codes can be found in the second chapter of Lucky's book *Silicon Dreams: Information, Man, and Machine*, St. Martin's Press, New York, 1989. More practical introductions can be found in overview chapters of many communications texts. The number of texts devoted entirely to this subject also continues to grow. Although these texts summarize the algebra underlying block codes, more in-depth treatments can be found in mathematical texts. Survey papers appear occasionally in the literature, but the interested reader is directed to the seminal papers by Shannon, Hamming, Reed and Solomon, Bose and Chaudhuri, Hocquenghem, Wozencraft, Fano, Forney, Berlekamp, Massey, Viterbi, Ungerboeck, Berrou and Glavieux, among others. The most recent advances in the theory and implementation of error control codes are published in *IEEE*

Transactions on Information Theory, IEEE Transactions on Communications, and special issues of *IEEE Journal on Selected Areas in Communications.*