29• Information Theory

} { { } }

Algebraic Coding Theory
Standard Article

Tor Helleseth Torleiv KlØve[1]
[1]University of Bergen, Bergen, Norway

DOI: 10.1002/047134608X.W4205
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (222K)

Abstract

The sections in this article are

Linear Codes

Some Bounds on Codes

Galois Fields

Cyclic Codes

BCH Codes

Automorphisms

The Weight Distribution of a Code

The Binary Golay Code

Decoding

Reed–Solomon Codes

Nonlinear Codes from Codes over Z_4

ALGEBRAIC CODING THEORY

In computers and digital communication systems, information is almost always represented in a binary form as a sequence of bits each having the values 0 or 1. This sequence of bits is transmitted over a *channel* from a sender to a receiver. In some applications the channel is a storage medium like a CD, where the information is written to the medium at a certain time and retrieved at a later time. Due to physical limitations of the channel, some of the transmitted bits may be corrupted (the channel is *noisy*) and thus make it difficult for the receiver to reconstruct the information correctly.

In algebraic coding theory we are mainly concerned with developing methods for detecting and correcting errors that typically occur during transmission of information over a noisy channel. The basic technique to detect and correct errors is by introducing redundancy in the data that are to be transmitted. This is similar to communicating in a natural language in daily life. One can understand the information while listening to a noisy radio or talking on a bad telephone line due to the redundancy in the language.

For an example, suppose the sender wants to communicate one of 16 different messages to a receiver. Each message m can then be represented as a binary quadruple $m = (c_0, c_1, c_2, c_3)$. If the message (0101) is transmitted and the first position is corrupted such that (1101) is received, this leads to an

uncorrectable error since this quadruple represents a different valid message than the message that was sent across the channel. The receiver will have no way to detect and correct a corrupted message in general, since any quadruple represents a valid message.

Therefore, to combat errors the sender *encodes* the data by introducing redundancy into the transmitted information. If M messages are to be transmitted, the sender selects a subset of M binary n-tuples, where $M < 2^n$. Each of the M messages is encoded into one of the selected n-tuples. The set consisting of the M n-tuples obtained after encoding is called a binary (n, M) *code* and the elements are called *codewords.* The codewords are sent over the channel.

It is customary for many applications to let $M = 2^k$, such that each message can be represented uniquely by a k-tuple of information bits. To encode each message the sender can append $n - k$ parity bits depending on the message bits and use the resulting n bit codeword to represent the corresponding message.

A binary code C is called a linear code if the sum (modulo 2) of two codewords is again a codeword. This is always the case when the parity bits are linear combinations of the information bits. In this case, the code C is a vector space of dimension k over the binary field of two elements, containing $M = 2^k$ codewords, and is called an $[n, k]$ code. The main reason for using linear codes is that these codes have more algebraic structure and are therefore often easier to analyze and decode in practical applications.

The simplest example of a linear code is the $[n, n - 1]$ *even-weight code* (or parity-check code). The encoding consists of appending a single parity bit to the $n - 1$ information bits so that the codeword has an even number of ones. Thus the code consists of all 2^{n-1} possible n-tuples of even weight, where the *weight* of a vector is the total number of ones in its components. This code can detect all errors in an odd number of positions, since if such an error occurs the received vector will also have odd weight. The even-weight code, however, can only detect errors. For example, if (000 . . . 0) is sent and the first bit is corrupted, then (100 . . . 0) is received. Also, if (110 . . . 0) was sent and the second bit was corrupted, then (100 . . . 0) is received. Hence, there is no way the receiver can correct this single error or, in fact, any other error.

An illustration of a code that can correct any single error is shown in Fig. 1. The three circles intersect and divide the plane into seven finite areas and one infinite area. Each finite area contains a bit c_i for $i = 0, 1, \ldots, 6$. Each of the 16 possible messages, denoted by (c_0, c_1, c_2, c_3), is encoded into a codeword $(c_0, c_1, c_2, c_3, c_4, c_5, c_6)$, in such a way that the sum of the bits in each circle has an even parity.

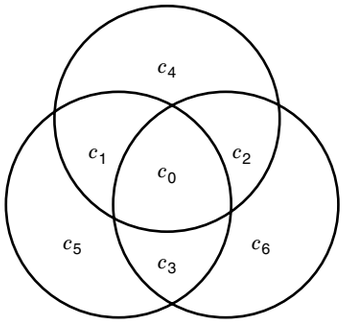


Figure 1. The message (c_0, c_1, c_2, c_3) is encoded into the codeword $(c_0, c_1, c_2, c_3, c_4, c_5, c_6)$, where c_4, c_5, c_6 are chosen such that there is an even number of ones within each circle.

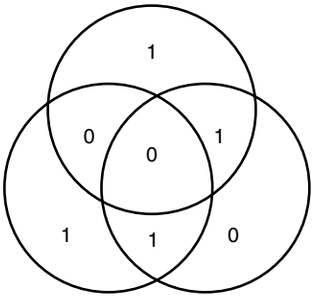


Figure 2. Example of the encoding procedure given in Fig. 1. The message (0011) is encoded into (0011110). Note that there is an even number of ones within each circle.

In Fig. 2, an example is shown of encoding the message (0011) into the codeword (0011110). Since the sum of two codewords also obeys the parity checks and thus is a codeword, the code is a linear [7, 4] code.

Suppose, for example, that the transmitted codeword is corrupted in the bit c_1 such that the received word is (0111110). Then, calculating the parity of each of the three circles, we see that the parity fails for the upper circle as well as for the leftmost circle while the parity of the rightmost circle is correct. Hence, from the received vector we can indeed conclude that bit c_1 is in error and should be corrected. In the same way, any single error can be corrected by this code.

LINEAR CODES

An (n, M) code is simply a set of M vectors of length n with components from a finite field $F_2 = \{0, 1\}$, where addition and multiplication are done modulo 2. For practical applications it is desirable that the code is provided with more structure. Therefore, linear codes are often preferred. A linear $[n, k]$ code C is a k-dimensional subspace C of F_2^n, where F_2^n is the vector space of n-tuples with coefficients from the finite field F_2.

A linear code C is usually described in terms of a generator matrix or a parity-check matrix. A *generator matrix* G of C is a $k \times n$ matrix whose row space is the code C. That is,

$$C = \{\boldsymbol{x}G | \boldsymbol{x} \in F_2^k\}$$

A *parity-check matrix* H is an $(n - k) \times n$ matrix such that

$$C = \{\boldsymbol{c} \in F_2^n | \boldsymbol{c}H^{\text{tr}} = \boldsymbol{0}\}$$

where H^{tr} denotes the transpose of H.

Example. The codewords in the code in the previous section are the vectors $(c_0, c_1, c_2, c_3, c_4, c_5, c_6)$ that satisfy the following

system of *parity-check equations:*

$$c_0 + c_1 + c_2 + c_4 = 0$$
$$c_0 + c_1 + c_3 + c_5 = 0$$
$$c_0 + c_2 + c_3 + c_6 = 0$$

where all additions are modulo 2. Each of the three parity-check equations correspond to one of the three circles.

The coefficient matrix of the parity-check equations is the parity-check matrix

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

The code C is therefore given by

$$C = \{\boldsymbol{c} = (c_0, c_1, \cdots, c_6) | \boldsymbol{c}H^{\text{tr}} = \boldsymbol{0}\}$$

A generator matrix for the code in the previous example is given by

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Two codes are *equivalent* if the codewords in one of the codes can be obtained by a fixed permutation of the positions in the codewords in the other code. If G (respectively, H) is a generator (respectively, parity-check) matrix of a code, then the matrices obtained by permuting the columns of these matrices in the same way give the generator matrix (respectively, parity-check) matrix of the permuted code.

The *Hamming distance* between $\boldsymbol{x} = (x_0, x_1, \ldots, x_{n-1})$ and $\boldsymbol{y} = (y_0, y_1, \ldots, y_{n-1})$ in F_2^n is the number of positions in which they differ. That is,

$$d(\boldsymbol{x}, \boldsymbol{y}) = |\{i | x_i \neq y_i, 0 \leq i \leq n-1\}|$$

The Hamming distance has the properties required to be a metric:

1. $d(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ for all $\boldsymbol{x}, \boldsymbol{y} \in F_2^n$ and equality holds if and only if $\boldsymbol{x} = \boldsymbol{y}$.
2. $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$ for all $\boldsymbol{x}, \boldsymbol{y} \in F_2^n$.
3. $d(\boldsymbol{x}, \boldsymbol{z}) \geq d(\boldsymbol{x}, \boldsymbol{y}) + d(\boldsymbol{y}, \boldsymbol{z})$ for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in F_2^n$.

For any code C one of the most important parameters is its *minimum distance,* defined by

$$d = \min\{d(\boldsymbol{x}, \boldsymbol{y}) | \boldsymbol{x} \neq \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{y} \in C\}$$

The *Hamming weight* of a vector $\boldsymbol{x}$ in F_2^n is the number of nonzero components in $\boldsymbol{x} = (x_0, x_1, \ldots, x_{n-1})$. That is,

$$w(\boldsymbol{x}) = |\{i | x_i \neq 0, 0 \leq i \leq n-1\}| = d(\boldsymbol{x}, \boldsymbol{0})$$

Note that since $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{x} - \boldsymbol{y}, \boldsymbol{0}) = w(\boldsymbol{x} - \boldsymbol{y})$ for a linear code C, it follows that

$$d = \min\{w(\boldsymbol{z}) | \boldsymbol{z} \in C, \boldsymbol{z} \neq \boldsymbol{0}\}$$

Therefore, finding the minimum distance of a linear code is equivalent to finding the minimum nonzero weight among all codewords in the code.

If $w(\boldsymbol{c}) = i$, then $\boldsymbol{c}H^{\text{tr}}$ is the sum of i columns of H. Hence, an alternative description of the minimum distance of a linear code is as follows: the smallest d such that there exists d linearly dependent columns in the parity-check matrix. In particular, to obtain a linear code of minimum distance at least three, it is sufficient to select the columns of a parity-check matrix to be distinct and nonzero.

Sometimes we include d in the notation and refer to an $[n, k]$ code with minimum distance d as an $[n, k, d]$ code. If t components are corrupted during transmission of a codeword, we say that t errors have occurred or that an error $\boldsymbol{e}$ of weight t has occurred [where $\boldsymbol{e} = (e_0, e_1, \ldots, e_{n-1}) \in F_2^n$, where $e_i = 1$ if and only if the ith component was corrupted—that is, if $\boldsymbol{c}$ was sent, $\boldsymbol{c} + \boldsymbol{e}$ was received].

The *error-correcting capability* of a code is defined as

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor$$

where $\lfloor x \rfloor$ denotes the largest integer $\leq x$.

A code with minimum distance d can correct all errors of weight t or less. This is due to the fact that if a codeword $\boldsymbol{c}$ is transmitted and an error $\boldsymbol{e}$ of weight $e \leq t$ occurs, the received vector $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$ is closer in Hamming distance to the transmitted codeword $\boldsymbol{c}$ than to any other codeword. Therefore, decoding any received vector to the closest codeword corrects all errors of weight $\leq t$.

The code can also be used for error detection only. The code is able to detect all errors of weight $< d$ since if a codeword is transmitted and the error has weight $< d$, then the received vector is not another codeword.

The code can also be used for a combination of error correction and error detection. For a given $e \leq t$, the code can correct all errors of weight $\leq e$ and in addition detect all errors of weight at most $d - e - 1$. This is due to the fact that no vector in F_2^n can be at distance $\leq e$ from one codeword and at the same time at a distance $\leq d - e - 1$ from another codeword. Hence, the algorithm in this case is to decode a received vector to a codeword at distance $\leq e$ if such a codeword exists and otherwise detect an error.

If C is an $[n, k]$ code, the *extended code* C^{ext} is the $[n + 1, k]$ code defined by

$$C^{\text{ext}} = \left\{ (c_{\text{ext}}, c_0, c_1, \ldots, c_{n-1}) \,\middle|\, (c_0, c_1, \ldots, c_{n-1}) \in C, \; c_{\text{ext}} = \sum_{i=0}^{n-1} c_i \right\}$$

That is, each codeword in C is extended by one parity bit such that the Hamming weight of each codeword becomes even. In particular, if C has odd minimum distance d, then the minimum distance of C^{ext} is $d + 1$. If H is a parity-check matrix for C, then a parity-check matrix for C^{ext} is

$$\begin{pmatrix} 1 & 1 \\ \boldsymbol{0}^{\text{tr}} & H \end{pmatrix}$$

where $\boldsymbol{1} = (11 \ldots 1)$.

For any linear $[n, k]$ code C, the *dual code* $C^\perp$ is the $[n, n-k]$ code defined by

$$C^\perp = \{\boldsymbol{x} \in F_2^n | (\boldsymbol{x}, \boldsymbol{c}) = 0 \text{ for all } \boldsymbol{c} \in C\}$$

where $(\boldsymbol{x}, \boldsymbol{c}) = \sum_{i=0}^{n-1} x_i c_i$. We say that $\boldsymbol{x}$ and $\boldsymbol{c}$ are *orthogonal* if $(\boldsymbol{x}, \boldsymbol{c}) = 0$. Therefore, $C^\perp$ consists of all n-tuples that are orthogonal to all codewords in C and vice versa—that is, $(C^\perp)^\perp = C$. It follows that $C^\perp$ has dimension $n - k$ since it consists of all vectors that are solutions of a system of equations with coefficient matrix G of rank k. Hence, the parity-check matrix of $C^\perp$ is a generator matrix of C, and similarly the generator matrix of $C^\perp$ is a parity-check matrix of C. In particular, $GH^{\text{tr}} = O$ [the $k \times (n-k)$ matrix of all zeros].

Example. Let C be the $[n, n-1, 2]$ even-weight code where

$$G = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 1 & \cdots & 0 & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 & 1 \end{pmatrix}$$

and

$$H = (1 \quad 1 \quad \cdots \quad 1 \quad 1 \quad 1)$$

Then $C^\perp$ has H and G as its generator and parity-check matrices, respectively. It follows that $C^\perp$ is the $[n, 1, n]$ *repetition code* consisting of the two codewords $(00 \cdots 000)$ and $(11 \cdots 111)$.

Example. Let C be the $[2^m - 1, 2^m - 1 - m, 3]$ code, where H contains all nonzero m-tuples as its columns. This is known as the *Hamming code*. In the case when $m = 3$, a parity-check matrix is already described in Eq. (1). Since all columns of the parity-check matrix are distinct and nonzero, the code has minimum distance at least 3. The minimum distance is indeed 3 since there exist three columns whose sum is zero, in fact the sum of any two columns of H equals another column in H for this particular code.

The dual code $C^\perp$ is the $[2^m - 1, m, 2^{m-1}]$ *simplex code* all of whose nonzero codewords have weight 2^{m-1}. This follows since the generator matrix has all nonzero vectors as its columns. In particular, taking any linear combination of rows, the number of columns with odd parity in the corresponding subset of rows equals 2^{m-1} (and the number with even parity is $2^{m-1} - 1$).

The extended code of the Hamming code is a $[2^m, 2^m - 1 - m, 4]$ code. Its dual code is a $[2^m, m+1, 2^{m-1}]$ code that is known as the first-order Reed-Muller code.

SOME BOUNDS ON CODES

The *Hamming bound* states that for any (n, M, d) code we have

$$M \sum_{i=0}^{e} \binom{n}{i} \le 2^n$$

where $e = \lfloor (d-1)/2 \rfloor$. This follows from the fact that the M spheres

$$S_{\boldsymbol{c}} = \{\boldsymbol{x} | d(\boldsymbol{x}, \boldsymbol{c}) \le e\}$$

centered at the codewords $\boldsymbol{c} \in C$ are disjoint and that each sphere contains

$$\sum_{i=0}^{e} \binom{n}{i}$$

vectors.

If the spheres fill the whole space, that is,

$$\bigcup_{\boldsymbol{c} \in C} S_{\boldsymbol{c}} = F_2^n$$

then C is called *perfect*. The binary *linear* perfect codes are as follows:

- The $[n, 1, n]$ repetition codes for all odd n
- The $[2^m - 1, 2^m - 1 - m, 3]$ Hamming codes H_m for all $m \ge 2$
- The $[23, 12, 7]$ Golay code G_{23}

We will return to the Golay code later.

GALOIS FIELDS

There exist finite fields, also known as Galois fields, with p^m elements for any prime p and any positive integer m. A Galois field of a given order p^m is unique (up to isomorphism) and is denoted by F_{p^m}.

For a prime p, let $F_p = \{0, 1, \ldots, p-1\}$ denote the integers modulo p with the two operations addition and multiplication modulo p.

To construct a Galois field with p^m elements, select a polynomial $f(x)$ with coefficients in F_p that is irreducible over F_p; that is, $f(x)$ cannot be written as a product of two polynomials with coefficients from F_p of degree ≥ 1 (irreducible polynomials of any degree m over F_p exist).

Let

$$F_{p^m} = \{a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_0 | a_0, \ldots, a_{m-1} \in F_p\}$$

Then F_{p^m} is a finite field when addition and multiplication of the elements (polynomials) are done modulo $f(x)$ and modulo p. To simplify the notations let α denote a zero of $f(x)$, that is, $f(\alpha) = 0$. If such an α exists, it can formally be defined as the equivalence class of x modulo $f(x)$. For coding theory, $p = 2$ is by far the most important case, and we assume this from now on. Note that for any $a, b \in F_{2^m}$,

$$(a+b)^2 = a^2 + b^2$$

Example. The Galois field F_{2^4} can be constructed as follows. Let $f(x) = x^4 + x + 1$ that is an irreducible polynomial over F_2. Then $\alpha^4 = \alpha + 1$ and

$$F_{2^4} = \{a_3\alpha^3 + a_2\alpha^2 + a_1\alpha + a_0 | a_0, a_1, a_2, a_3 \in F_2\}$$

Computing the powers of α, we obtain

$$\alpha^5 = \alpha \cdot \alpha^4 = \alpha(\alpha + 1) = \alpha^2 + \alpha,$$
$$\alpha^6 = \alpha \cdot \alpha^5 = \alpha(\alpha^2 + \alpha) = \alpha^3 + \alpha^2,$$
$$\alpha^7 = \alpha \cdot \alpha^6 = \alpha(\alpha^3 + \alpha^2) = \alpha^4 + \alpha^3 = \alpha^3 + \alpha + 1$$

and, similarly, all higher powers of α can be expressed as a linear combination of α^3, α^2, α, and 1. In particular, $\alpha^{15} = 1$. We get the following table of the powers of α. In the table the polynomial $a_3\alpha^3 + a_2\alpha^2 + a_1\alpha + a_0$ is represented as $a_3a_2a_1a_0$.

i	α^i	i	α^i	i	α^i
0	0001	5	0110	10	0111
1	0010	6	1100	11	1110
2	0100	7	1011	12	1111
3	1000	8	0101	13	1101
4	0011	9	1010	14	1001

Hence, the elements 1, α, α^2, . . ., α^{14} are all the nonzero elements in F_{2^4}. Such an element α that generates the nonzero elements of F_{2^m} is called a *primitive element* in F_{2^m}. An irreducible polynomial $g(x)$ with a primitive element as a root is called a primitive polynomial. Every finite field has a primitive element, and therefore the multiplicative subgroup of a finite field is cyclic.

All elements in F_{2^m} are roots of the equation $x^{2^m} + x = 0$. Let β be an element in F_{2^m}. It is important to study the polynomial $m(x)$ of smallest degree with coefficients in F_2 that has β as a zero. This polynomial is called the *minimal polynomial* of β over F_2.

First, observe that if $m(x) = \sum_{i=0}^{\kappa} m_i x^i$ has coefficients in F_2 and β as a zero, then

$$m(\beta^2) = \sum_{i=0}^{\kappa} m_i \beta^{2i} = \sum_{i=0}^{\kappa} m_i^2 \beta^{2i} = \left(\sum_{i=0}^{\kappa} m_i \beta^i \right)^2 = (m(\beta))^2 = 0$$

Hence, $m(x)$ has β, β^2, . . ., $\beta^{2^{\kappa-1}}$, as zeros, where κ is the smallest integer such that $\beta^{2^\kappa} = \beta$. Conversely, the polynomial with exactly these zeros can be shown to be a binary irreducible polynomial.

Example. We will find the minimal polynomial of all the elements in F_{2^4}. Let α be a root of $x^4 + x + 1 = 0$; that is, $\alpha^4 = \alpha + 1$. The minimal polynomials over F_2 of α^i for $0 \le i \le 14$ are denoted $m_i(x)$. Observe by the preceding argument that $m_{2i}(x) = m_i(x)$, where the indices are taken modulo 15. It follows that

$$m_0(x) = (x + \alpha^0) = x + 1,$$
$$m_1(x) = (x + \alpha)(x + \alpha^2)(x + \alpha^4)(x + \alpha^8) = x^4 + x + 1,$$
$$m_3(x) = (x + \alpha^3)(x + \alpha^6)(x + \alpha^{12})(x + \alpha^9) = x^4 + x^3 + x^2 + x + 1,$$
$$m_5(x) = (x + \alpha^5)(x + \alpha^{10}) = x^2 + x + 1,$$
$$m_7(x) = (x + \alpha^7)(x + \alpha^{14})(x + \alpha^{13})(x + \alpha^{11}) = x^4 + x^3 + 1,$$
$$m_9(x) = m_3(x),$$
$$m_{11}(x) = m_7(x),$$
$$m_{13}(x) = m_7(x)$$

To verify this, one simply computes the coefficients and uses the preceding table of F_{2^4} in the computations. For example,

$$m_5(x) = (x + \alpha^5)(x + \alpha^{10}) = x^2 + (\alpha^5 + \alpha^{10})x + \alpha^5 \cdot \alpha^{10} = x^2 + x + 1$$

This also leads to a factorization into irreducible polynomials:

$$x^{2^4} + x = x \prod_{j=0}^{14} (x + \alpha^j) = x(x + 1)(x^2 + x + 1)(x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)(x^4 + x^3 + 1) = x m_0(x) m_1(x) m_3(x) m_5(x) m_7(x)$$

In fact, it holds in general that $x^{2^m} + x$ is the product of all irreducible polynomials over F_2 of degree that divides m.

Let $C_i = \{i2^j \pmod n \mid j = 0, 1, \ldots\}$, which is called the *cyclotomic coset* of $i \pmod n$. Then the elements of the cyclotomic coset $C_i \pmod{2^m - 1}$ correspond to the exponents of the zeros of $m_i(x)$. That is,

$$m_i(x) = \prod_{j \in C_i} (x - \alpha^j)$$

The cyclotomic cosets $\pmod n$ are important in the next section when cyclic codes of length n are discussed.

CYCLIC CODES

Many good linear codes that have practical and efficient decoding algorithms have the property that a cyclic shift of a codeword is again a codeword. Such codes are called cyclic codes.

We can represent the set of n-tuples over F_2^n as polynomials of degree $< n$ in a natural way. The vector $\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-1})$ is represented as the polynomial $c(x) = c_0 + c_1x + c_2x^2 + \cdots + c_{n-1}x^{n-1}$. A *cyclic shift*

$$\sigma(\boldsymbol{c}) = (c_{n-1}, c_0, c_1, \ldots, c_{n-2})$$

of $\boldsymbol{c}$ is then represented by the polynomial

$$\sigma(c(x)) = c_{n-1} + c_0x + c_1x^2 + \cdots + c_{n-2}x^{n-1} = x(c_{n-1}x^{n-1} + c_0 + c_1x + \cdots + c_{n-2}x^{n-2}) + c_{n-1}(x^n + 1) \equiv xc(x) \pmod{x^n + 1}$$

Example. Rearranging the columns in the parity-check matrix of the [7, 4] Hamming code in Eq. (1), an equivalent code is obtained with parity-check matrix

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} \tag{2}$$

This code contains 16 codewords, which are represented next in polynomial form:

$$\begin{aligned}
1000110 &\leftrightarrow & x^5+x^4+1 &= & (x^2+x+1)g(x)\\
0100011 &\leftrightarrow & x^6+x^5+x &= & (x^3+x^2+x)g(x)\\
1010001 &\leftrightarrow & x^6+x^2+1 &= & (x^3+x+1)g(x)\\
1101000 &\leftrightarrow & x^3+x+1 &= & g(x)\\
0110100 &\leftrightarrow & x^4+x^2+x &= & xg(x)\\
0011010 &\leftrightarrow & x^5+x^3+x^2 &= & x^2g(x)\\
0001101 &\leftrightarrow & x^6+x^4+x^3 &= & x^3g(x)\\
0010111 &\leftrightarrow & x^6+x^5+x^4+x^2 &= & (x^3+x^2)g(x)\\
1001011 &\leftrightarrow & x^6+x^5+x^3+1 &= & (x^3+x^2+x+1)g(x)\\
1100101 &\leftrightarrow & x^6+x^4+x+1 &= & (x^3+1)g(x)\\
1110010 &\leftrightarrow & x^5+x^2+x+1 &= & (x^2+1)g(x)\\
0111001 &\leftrightarrow & x^6+x^3+x^2+x &= & (x^3+x)g(x)\\
1011100 &\leftrightarrow & x^4+x^3+x^2+1 &= & (x+1)g(x)\\
0101110 &\leftrightarrow & x^5+x^4+x^3+x &= & (x^2+x)g(x)\\
0000000 &\leftrightarrow & 0 &= & 0\\
1111111 &\leftrightarrow & x^6+x^5+\cdots+x+1 &= & (x^3+x^2+1)g(x)
\end{aligned}$$

By inspection it is easy to verify that any cyclic shift of a codeword is again a codeword. Indeed, the 16 codewords in the code are $\mathbf{0}$, $\mathbf{1}$ and all cyclic shifts of (1000110) and (0010111). The unique nonzero polynomial in the code of lowest possible degree is $g(x) = x^3 + x + 1$, and $g(x)$ is called the *generator polynomial* of the cyclic code. The code consists of all polynomials $c(x)$ that are multiples of $g(x)$. Note that the degree of $g(x)$ is $n - k = 3$ and that $g(x)$ divides $x^7 + 1$ since $x^7 + 1 = (x + 1)(x^3 + x + 1)(x^3 + x^2 + 1)$.

The code therefore has a simple description in terms of the set of code polynomials as

$$C = \{c(x) | c(x) = u(x)(x^3 + x + 1), \deg(u(x)) < 4\}$$

This situation holds in general for any cyclic code.

For any cyclic $[n, k]$ code C, we have

$$C = \{c(x) | c(x) = u(x)g(x), \deg(u(x)) < k\}$$

for a polynomial $g(x)$ of degree $n - k$ that divides $x^n + 1$.

We can show this as follows: Let $g(x)$ be the generator polynomial of C, which is the nonzero polynomial of smallest degree r in the code C. Then the cyclic shifts $g(x)$, $xg(x)$, $\cdots$, $x^{n-r-1}g(x)$ are codewords as well as any linear combination $u(x)g(x)$, where $\deg(u(x)) < n - r$. These are the only 2^{n-r} codewords in the code C, since if $c(x)$ is a codeword then

$$c(x) = u(x)g(x) + s(x), \text{where} \quad \deg(s(x)) < \deg(g(x))$$

By linearity, $s(x)$ is a codeword and therefore $s(x) = 0$ since $\deg(s(x)) < \deg(g(x))$ and $g(x)$ is the nonzero polynomial of smallest degree in the code. It follows that C is as described previously. Since C has 2^{n-r} codewords, it follows that $n - r = k$; that is, $\deg(g(x)) = n - k$.

Finally, we show that $g(x)$ divides $x^n + 1$. Let $c(x) = c_0 + c_1x + \cdots + c_{n-1}x^{n-1}$ be a nonzero codeword shifted such that $c_{n-1} = 1$. Then a cyclic shift of $c(x)$ given by $\sigma(c(x)) = c_{n-1} + c_0x + c_1x + \cdots + c_{n-2}x^{n-1}$ is also a codeword and

$$\sigma(c(x)) = xc(x) + \vartheta(x^n + 1)$$

Since both of the codewords $c(x)$ and $\sigma(c(x))$ are divisible by $g(x)$, it follows that $g(x)$ divides $x^n + 1$.

Since the generator polynomial of a cyclic code divides $x^n + 1$, it is important to know how to factor $x^n + 1$ into irreducible polynomials. Let n be odd. Then there is an integer m such that $2^m \equiv 1 \pmod{n}$ and there is an element $\alpha \in F_{2^m}$ of order n [if ω is a primitive element of F_{2^m}, then α can be taken to be $\alpha = \omega^{(2^m-1)/n}$].

We have

$$x^n + 1 = \prod_{i=0}^{n-1}(x + \alpha^i)$$

Let $m_i(x)$ denote the minimal polynomial of α^i; that is, the polynomial of smallest degree with coefficients in F_2 and having α^i as a zero. The generator polynomial $g(x)$ can be written as

$$g(x) = \prod_{i \in I}(x + \alpha^i)$$

where I is a subset of $\{0, 1, \ldots, n-1\}$, called the *defining set* of C with respect to α. Then $m_i(x)$ divides $g(x)$ for all $i \in I$. Further, $g(x) = \prod_{j=1}^{l} m_{i_j}(x)$ for some $i_1, i_2, \ldots, i_l$.

We can therefore describe the cyclic code in alternative equivalent ways as

$$\begin{aligned}
&C = \{c(x) | m_i(x) \text{ divides } c(x), \text{ for all } i \in I\},\\
&C = \{c(x) | c(\alpha^i) = 0, \text{ for all } i \in I\},\\
&C = \{\boldsymbol{c} \in F_2^n | \boldsymbol{c}H^{\text{tr}} = \mathbf{0}\}
\end{aligned}$$

where

$$H = \begin{pmatrix}
1 & \alpha^{i_1} & \alpha^{2i_1} & \cdots & \alpha^{(n-1)i_1}\\
1 & \alpha^{i_2} & \alpha^{2i_2} & \cdots & \alpha^{(n-1)i_2}\\
\vdots & \vdots & \vdots & \ddots & \vdots\\
1 & \alpha^{i_l} & \alpha^{2i_l} & \cdots & \alpha^{(n-1)i_l}
\end{pmatrix}$$

The encoding for cyclic codes is usually done in one of two ways. Let $u(x)$ denote the information polynomial of degree $< k$. The two ways are as follows:

1. Encode into $u(x)g(x)$.
2. Encode into $c(x) = x^{n-k}u(x) + s(x)$, where $s(x)$ is the polynomial such that
   - $s(x) \equiv x^{n-k}u(x) \pmod{g(x)}$ [thus $g(x)$ divides $c(x)$]
   - $\deg(s(x)) < \deg(g(x))$

The last of these two methods is systematic; that is, the last k bits of the codeword are the information bits.

BCH CODES

An important task in coding theory is to design codes with a guaranteed minimum distance d that correct all errors of Hamming weight $\lfloor (d-1)/2 \rfloor$. Such codes were designed independently by Bose and Ray-Chaudhuri (1960) and by Hocquenghem (1959) and are known as BCH codes. To construct a BCH code of *designed distance* d, the generator polynomial

is chosen to have $d-1$ consecutive powers of α as zeros

$$\alpha^b, \alpha^{b+1}, \cdots, \alpha^{b+d-2}$$

That is, the defining set I with respect to α contains a set of $d-1$ consecutive integers (mod n). The parity-check matrix of the BCH code is

$$H = \begin{pmatrix} 1 & \alpha^b & \alpha^{2b} & \cdots & \alpha^{(n-1)b} \\ 1 & \alpha^{b+1} & \alpha^{2(b+1)} & \cdots & \alpha^{(n-1)(b+1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{b+d-2} & \alpha^{2(b+d-2)} & \cdots & \alpha^{(n-1)(b+d-2)} \end{pmatrix}$$

To show that this code has minimum distance at least d, it is sufficient to show that any $d-1$ columns are linear independent. Suppose there is a linear dependency between the $d-1$ columns corresponding to $\alpha^{i_1 b}, \alpha^{i_2 b}, \ldots, \alpha^{i_{d-1} b}$. In this case the $(d-1) \times (d-1)$ submatrix obtained by retaining these columns in H has determinant

$$\begin{vmatrix} \alpha^{i_1 b} & \alpha^{i_2 b} & \cdots & \alpha^{i_{d-1} b} \\ \alpha^{i_1 (b+1)} & \alpha^{i_2 (b+1)} & \cdots & \alpha^{i_{d-1} (b+1)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{i_2 (b+d-2)} & \alpha^{i_2 (b+d-2)} & \cdots & \alpha^{i_{d-1} (b+d-2)} \end{vmatrix}$$

$$= \alpha^{b(i_1+i_2+\ldots+i_{d-1})} \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \alpha^{i_1} & \alpha^{i_2} & \cdots & \alpha^{i_{d-1}} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha^{(d-2)i_1} & \alpha^{(d-2)i_2} & \cdots & \alpha^{(d-2)i_{d-1}} \end{vmatrix}$$

$$= \alpha^{b(i_1+i_2+\ldots+i_{d-1})} \prod_{k<r} (\alpha^{i_k} - \alpha^{i_r}) \neq 0$$

since the elements $\alpha^{i_1}, \alpha^{i_2}, \cdots, \alpha^{i_{d-1}}$ are distinct (the last equality follows from the fact that the last determinant is a Vandermonde determinant). It follows that the BCH code has minimum Hamming distance at least d.

If $b = 1$, which is often the case, the code is called a *narrow-sense* BCH code. If $n = 2^m - 1$, the BCH code is called a *primitive* BCH code. A binary single error-correcting primitive BCH code is generated by $g(x) = m_1(x)$. The zeros of $g(x)$ are α^{2^i}, $i = 0, 1, \ldots, m-1$. The parity-check matrix is

$$H = (1 \quad \alpha^1 \quad \alpha^2 \quad \ldots \quad \alpha^{2^m-2})$$

This code is equivalent to the Hamming code since α is a primitive element of F_{2^m}.

To construct a binary double error-correcting primitive BCH code, we let $g(x)$ have α, α^2, α^3, α^4 as zeros. Therefore, $g(x) = m_1(x)m_3(x)$ is a generator polynomial of this code. The parity-check matrix of a double error-correcting BCH code is

$$H = \begin{pmatrix} 1 & \alpha^1 & \alpha^2 & \cdots & \alpha^{2^m-2} \\ 1 & \alpha^3 & \alpha^6 & \cdots & \alpha^{3(2^m-2)} \end{pmatrix}$$

In particular, a binary double-error correcting BCH code of length $n = 2^4 - 1 = 15$ is obtained by selecting

$$\begin{aligned} g(x) &= m_1(x)m_3(x) \\ &= (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1) \\ &= x^8 + x^7 + x^6 + x^4 + 1 \end{aligned}$$

Similarly, a binary triple-error correcting BCH code of the same length is obtained by choosing the generator polynomial

$$\begin{aligned} g(x) &= m_1(x)m_3(x)m_5(x) \\ &= (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)(x^2 + x + 1) \\ &= x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1 \end{aligned}$$

The main interest in BCH codes is due to the fact that they have a very fast and efficient decoding algorithm. We describe this later.

AUTOMORPHISMS

Let C be a binary code of length n. Consider a permutation π of the set $\{0, 1, \ldots, n-1\}$; that is, π is a one-to-one function of the set of coordinate positions onto itself.

For a codeword $\boldsymbol{c} \in C$, let

$$\pi(\boldsymbol{c}) = (c_{\pi(0)}, c_{\pi(1)}, \ldots, c_{\pi(n-1)})$$

That is, the coordinates are permuted by the permutation π. If

$$\{\pi(\boldsymbol{c}) | \boldsymbol{c} \in C\} = C$$

then π is called an *automorphism* of the code C.

Example. Consider the following (nonlinear code):

$$C = \{101, 011\}$$

The actions of the six possible permutations on three elements are given in the following table. The permutations that are automorphisms are marked by a star.

$\pi(0)$	$\pi(1)$	$\pi(2)$	$\pi((101))$	$\pi((011))$	
0	1	2	101	011	★
0	2	1	110	011	
1	0	2	011	101	★
1	2	0	011	110	
2	0	1	110	101	
2	1	0	101	110	

In general, the set of automorphisms of a code C is a group, the *Automorphism group* Aut(C). We note that

$$\sum_{i=0}^{n-1} x_i y_i = \sum_{i=0}^{n-1} x_{\pi(i)} y_{\pi(i)}$$

and so $(\boldsymbol{x}, \boldsymbol{y}) = 0$ if and only if $(\pi(\boldsymbol{x}), \pi(\boldsymbol{y})) = 0$. In particular, this implies that

$$\text{Aut}(C) = \text{Aut}(C^\perp)$$

That is, C and $C^\perp$ have the same automorphism group.

For a *cyclic* code C of length n, we have by definition $\sigma(\boldsymbol{c}) \in C$ for all $\boldsymbol{c} \in C$, where $\sigma(i) \equiv i - 1 \pmod{n}$. In particular, $\sigma \in \text{Aut}(C)$. For n odd, the permutation δ defined by $\delta(j) =$

$2j \pmod n$ is also contained in the automorphism group. To show this it is easier to show that $\delta^{-1} \in \mathrm{Aut}(C)$. We have

$$\begin{aligned} \delta^{-1}(2j) &= j && \text{for } j = 0, 1, \ldots, (n-1)/2, \\ \delta^{-1}(2j+1) &= (n+1)/2 + j && \text{for } j = 0, 1, \ldots, (n-1)/2 - 1 \end{aligned}$$

Let $g(x)$ be a generator polynomial for C, and let $\sum_{i=0}^{n-1} c_i x^i = a(x)g(x)$. Since $x^n \equiv 1 \pmod{x^n + 1}$, we have

$$\begin{aligned} \sum_{i=0}^{n-1} c_{\delta^{-1}(i)} x^i &\equiv \sum_{j=0}^{(n-1)/2} c_j x^{2j} + \sum_{j=0}^{(n-1)/2-1} c_{(n+1)/2+j} x^{2j+1+n} \\ &= \sum_{j=0}^{(n-1)/2} c_j x^{2j} + \sum_{j=(n+1)/2}^{n-1} c_j x^{2j} \\ &= a(x^2)g(x^2) = (a(x^2)g(x))g(x), \quad (\mathrm{mod}\, x^n + 1) \end{aligned}$$

and so $\delta^{-1}(\boldsymbol{c}) \in C$; that is, $\delta^{-1} \in \mathrm{Aut}(C)$ and so $\delta \in \mathrm{Aut}(C)$.

The automorphism group $\mathrm{Aut}(C)$ is *transitive* if for each pair (i, j) there exists a $\pi \in \mathrm{Aut}(C)$ such that $\pi(i) = j$. More general, $\mathrm{Aut}(C)$ is t-fold transitive if, for distinct $i_1, i_2, \ldots, i_t$ and distinct $j_1, j_2, \ldots, j_t$, there exists a $\pi \in \mathrm{Aut}(C)$ such that $\pi(i_1) = j_1, \pi(i_2) = j_2, \ldots, \pi(i_t) = j_t$.

Example. Any cyclic $[n, k]$ code has a transitive automorphism group since σ repeated s times, where $s \equiv i - j \pmod n$, maps i to j.

Example. The (nonlinear) code $C = \{101, 011\}$ was considered previously. Its automorphism group is not transitive since there is no automorphism π such that $\pi(0) = 2$.

Example. Let C be the $[9, 3]$ code generated by the matrix

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

This is a cyclic code and we will determine its automorphism group. The all zero and the all one vectors in C are transformed into themselves by any permutation. The vectors of weight 3 are the rows of the generator matrix and the vectors of weight 6 are the complements of these vectors. Hence, we see that π is an automorphism if and only if it leaves the set of the three rows of the generator matrix invariant, that is, if and only if the following conditions are satisfied:

$$\begin{aligned} \pi(0) \equiv \pi(3) \equiv \pi(6) \quad (mod\ 3), \\ \pi(1) \equiv \pi(4) \equiv \pi(7) \quad (mod\ 3), \\ \pi(2) \equiv \pi(5) \equiv \pi(8) \quad (mod\ 3). \end{aligned}$$

Note that the two permutations σ and δ defined previously satisfy these conditions, as they should. They are listed explicitly in the following table

i	0	1	2	3	4	5	6	7	8
$\sigma(i)$	8	0	1	2	3	4	5	6	7
$\delta(i)$	0	2	4	6	8	1	3	5	7

The automorphism group is transitive since the code is cyclic, but not doubly transitive. For example, there is no automorphism π such that $\pi(0) = 0$ and $\pi(3) = 1$ since 0 and 1 are not equivalent modulo 3. A simple counting argument shows that $\mathrm{Aut}(C)$ has order 1296: First choose $\pi(0)$; this can be done in 9 ways. There are then 2 ways to choose $\pi(3)$ and $\pi(6)$. Next choose $\pi(1)$; this can be done in 6 ways. There are again 2 ways to choose $\pi(4)$ and $\pi(7)$. Finally, there are $3 \cdot 2$ ways to choose $\pi(2)$, $\pi(5)$, $\pi(8)$. Hence, the order is $9 \cdot 2 \cdot 6 \cdot 2 \cdot 3 \cdot 2 = 1296$.

Example. Consider the extended Hamming code H_m^{ext}. The positions of the codewords correspond to the elements of F_{2^m} and are permuted by the affine group

$$\mathrm{AG} = \{\pi \,|\, \pi(x) = ax + b, a, b \in F_{2^m}, a \neq 0\}$$

This is the automorphism group of H_m^{ext}. It is double transitive.

THE WEIGHT DISTRIBUTION OF A CODE

Let C be a binary linear $[n, k]$ code. As we noted before,

$$d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{x} - \boldsymbol{y}, \boldsymbol{0}) = w(\boldsymbol{x} - \boldsymbol{y})$$

If $\boldsymbol{x}, \boldsymbol{y} \in C$, then $\boldsymbol{x} - \boldsymbol{y} \in C$ by the linearity of C. In particular, this means that the set of distances from a fixed codeword to all the other codewords is independent of which codeword we fix; that is, the code looks the same from any codeword. In particular, the set of distances from the codeword $\boldsymbol{0}$ is the set of *Hamming weights* of the codewords. For $i = 0, 1, \ldots, n$, let A_i denote the number of codewords of weight i. The sequence

$$A_0, A_1, A_2, \ldots, A_n$$

is called the *weight distribution* of the code C. The corresponding polynomial

$$A_C(z) = A_0 + A_1 z + A_2 z^2 + \cdots + A_n z^n$$

is known as the *weight enumerator polynomial* of C.

The polynomials $A_C(z)$ and $A_{C^\perp}(z)$ are related by the fundamental MacWilliams identity:

$$A_{C^\perp}(z) = 2^{-k}(1+z)^n A_C\left(\frac{1-z}{1+z}\right)$$

Example. The $[2^m - 1, m]$ simplex code has the weight enumerator polynomial $1 + (2^m - 1)z^{2^{m-1}}$. The dual code is the $[2^m - 1, 2^m - 1 - m]$ Hamming code with weight enumerator polynomial

$$\begin{aligned} &2^{-m}(1+z)^{2^m-1}\left(1 + (2^m - 1)\left(\frac{1-z}{1+z}\right)^{2^{m-1}}\right) \\ &\quad = 2^{-m}(1+z)^{2^m-1} + (1 - 2^{-m})(1-z)^{2^{m-1}}(1+z)^{2^{m-1}-1} \end{aligned}$$

For example, for $m = 4$, we get the weight distribution of the [15, 11] Hamming code:

$$1 + 35z^3 + 105z^4 + 168z^5 + 280z^6 + 435z^7 + 435z^8 + 280z^9 + 168z^{10} + 105z^{11} + 35z^{12} + z^{15}$$

Consider a binary linear code C that is used purely for error detection. Suppose a codeword $\boldsymbol{c}$ is transmitted over a binary symmetric channel with bit error probability p. The probability of receiving a vector $\boldsymbol{r}$ at distance i from $\boldsymbol{c}$ is $p^i(1-p)^{n-i}$, since i positions are changed (each with probability p) and $n - i$ are unchanged (each with probability $1 - p$). If $\boldsymbol{r}$ is not a codeword, then this will be discovered by the receiver. If $\boldsymbol{r} = \boldsymbol{c}$, then no errors have occurred. However, if $\boldsymbol{r}$ is another codeword, then an undetectable error has occurred. Hence, the *probability of undetected error* is given by

$$\begin{aligned} P_{\text{ue}}(C, p) &= \sum_{\boldsymbol{c}' \neq \boldsymbol{c}} p^{d(\boldsymbol{c}', \boldsymbol{c})}(1-p)^{n-d(\boldsymbol{c}', \boldsymbol{c})} \\ &= \sum_{\boldsymbol{c}'' \neq \boldsymbol{0}} p^{w(\boldsymbol{c}'')}(1-p)^{n-w(\boldsymbol{c}'')} \\ &= \sum_{i=1}^{n} A_i p^i (1-p)^{n-i} \\ &= (1-p)^n A_C\left(\frac{p}{1-p}\right) - (1-p)^n \end{aligned}$$

From the MacWilliams identity we also get

$$P_{\text{ue}}(C^\perp, p) = 2^{-k} A_C(1-2p) - (1-p)^n$$

Example. For the $[2^m - 1, 2^m - 1 - m]$ Hamming code H_m, we get

$$P_{\text{ue}}(H_m, p) = 2^{-m}(1 + (2^m - 1)(1-2p)^{2^{m-1}}) - (1-p)^{2^{m}-1}$$

More information on the use of codes for error detection can be found in the book by Kløve and Korzhik (see Reading List).

THE BINARY GOLAY CODE

The Golay code G_{23} has received much attention. It is practically useful and has a number of interesting properties. The code can be defined in various ways. One definition is that G_{23} is the cyclic code generated by the irreducible polynomial

$$x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1$$

which is a factor of $x^{23} + 1$ over F_2. Another definition is the following: Let H denote the [7, 4] Hamming code and let H^* be the code whose codewords are the reversed of the codewords of H. Let

$$C = \{(\boldsymbol{u} + \boldsymbol{x}, \boldsymbol{v} + \boldsymbol{x}, \boldsymbol{u} + \boldsymbol{v} + \boldsymbol{x}) | \boldsymbol{u}, \boldsymbol{v} \in H^{\text{ext}}, \boldsymbol{x} \in (H^*)^{\text{ext}}\}$$

where H^{ext} is the [8, 4] extended Hamming code and $(H^*)^{\text{ext}}$ is the [8, 4] extended H^*. The code C is a [24, 12, 8] code. Puncturing the last position, we get a [23, 12, 7] code that is (equivalent to) the Golay code.

The weight distribution of G_{23} is given by the following table:

i	A_i
0, 23	1
7, 16	253
8, 15	506
11, 12	1288

The automorphism group $\text{Aut}(G_{23})$ of the Golay code is the Mathieu group M_{23}, a simple group of order $10200960 = 2^7 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11 \cdot 23$, which is four-fold transitive.

Much information about G_{23} can be found in the book by MacWilliams and Sloane (see Reading List).

DECODING

Suppose that a codeword $\boldsymbol{c}$ from the $[n, k]$ code C was sent and that an error $\boldsymbol{e}$ occurred during the transmission over the noisy channel. Based on the received vector $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$, the receiver has to make an estimate of what was the transmitted codeword. Since error patterns of lower weight are more probable than error patterns of higher weight, the problem is to estimate an error $\hat{\boldsymbol{e}}$ such that the weight of $\hat{\boldsymbol{e}}$ is as small as possible. He will then decode the received vector $\boldsymbol{r}$ into $\hat{\boldsymbol{c}} = \boldsymbol{r} + \hat{\boldsymbol{e}}$.

If H is a parity-check matrix for C, then $\boldsymbol{c}H^{\text{tr}} = \boldsymbol{0}$ for all codewords $\boldsymbol{c}$. Hence,

$$\boldsymbol{r}H^{\text{tr}} = (\boldsymbol{c} + \boldsymbol{e})H^{\text{tr}} = \boldsymbol{c}H^{\text{tr}} + \boldsymbol{e}H^{\text{tr}} = \boldsymbol{e}H^{\text{tr}} \tag{3}$$

The vector

$$\boldsymbol{s} = \boldsymbol{e}H^{\text{tr}}$$

is known as the *syndrome* of the error $\boldsymbol{e}$; Eq. (3) shows that $\boldsymbol{s}$ can be computed from $\boldsymbol{r}$. We now have the following outline of a decoding strategy:

1. Compute the syndrome $\boldsymbol{s} = \boldsymbol{r}H^{\text{tr}}$.
2. Estimate an error $\hat{\boldsymbol{e}}$ of smallest weight corresponding to the syndrome $\boldsymbol{s}$.
3. Decode to $\hat{\boldsymbol{c}} = \boldsymbol{r} + \hat{\boldsymbol{e}}$.

The hard part is, of course, step 2.

For any vector $\boldsymbol{x} \in F_2^n$, the set $\{\boldsymbol{x} + \boldsymbol{c} \mid \boldsymbol{c} \in C\}$ is a *coset* of C. All the elements of the coset have the same syndrome—namely, $\boldsymbol{x}H^{\text{tr}}$. There are 2^{n-k} cosets, one for each syndrome in F_2^{n-k}, and the set of cosets is partition of F_2^n. We can rephrase step 2 as follows: Find a vector $\boldsymbol{e}$ of smallest weight in the coset with syndrome $\boldsymbol{s}$.

Example. Let C be the [6, 3, 3] code with parity-check matrix

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

A *standard array* for C is the following array (the eight columns to the right):

000	111000	001011	010101	011110	100110	101101	110011	111000
110	100000	101011	110101	111110	000110	001101	010011	011000
101	010000	011011	000101	001110	110110	111101	100011	101000
011	001000	000011	011101	010110	101110	100101	111011	110000
100	000100	001111	010001	011010	100010	101001	110111	111100
010	000010	001001	010111	011100	100100	101111	110001	111010
001	000001	001010	010100	011111	100111	101100	110010	111001
111	100001	101010	110100	111111	000111	001100	010010	011001

Each row in the array is a listing of a coset of C; the first row is a listing of the code itself. The vectors in the first column have minimal weight in their cosets and are known as *coset leaders*. The choice of coset leader may not be unique. For example, in the last coset there are three vectors of minimal weight. Any entry in the array is the sum of the codeword at the top of the column and the coset leader (at the left in the row). Each vector of F_2^6 is listed exactly once in the array. The standard array can be used for decoding: Locate $\boldsymbol{r}$ in the array and decode to the codeword at the top of the corresponding column (that is, the coset leader is assumed to be the error pattern). However, this is not a practical method; except for small n, the standard array of 2^n entries is too large to store (also locating $\boldsymbol{r}$ may be a problem). A step in simplifying the method is to store a table of coset leaders corresponding to the 2^{n-k} syndromes. In the preceding table this is illustrated by listing the syndromes at the left. Again this is a possible alternative only if $n - k$ is small. For carefully designed codes, it is possible to *compute* $\boldsymbol{e}$ from the syndrome. The simplest case is single errors: If $\boldsymbol{e}$ is an error pattern of weight 1, where the 1 is in the ith position, then the corresponding syndrome is the ith column of H; hence, from H and the syndrome we can determine i.

Example. Let H be the $m \times (2^m - 1)$ parity-check matrix where the ith column is the binary expansion of the integer i for $i = 1, 2, \ldots, 2^m - 1$. The corresponding $[2^m - 1, 2^m - 1 - m, 3]$ Hamming code corrects all single errors. Decoding is done as follows: Compute the syndrome $\boldsymbol{s} = (s_0, s_1, \ldots, s_{m-1})$. If $\boldsymbol{s} \neq \boldsymbol{0}$, then correct position $i = \sum_{j=0}^{m-1} s_j 2^j$.

Example. Let

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^3 & \alpha^6 & \cdots & \alpha^{3(n-1)} \end{pmatrix}$$

where $\alpha \in F_{2^m}$ and $n = 2^m - 1$. This is the parity-check matrix for the double error-correcting BCH code. It is convenient to have a similar representation of the syndromes:

$$\boldsymbol{s} = (S_1, S_3) \quad \text{where} \quad S_1, S_3 \in F_{2^m}$$

Depending on the syndrome, there are several cases:

1. If no errors have occurred, then clearly $S_1 = S_3 = 0$.
2. If a single error has occurred in the ith position (that is, the position corresponding to α^i), then $S_1 = \alpha^i$ and $S_3 = \alpha^{3i}$. In particular, $S_3 = S_1^3$.
3. If two errors have occurred in positions i and j, then

$$S_1 = \alpha^i + \alpha^j, \quad S_3 = \alpha^{3i} + \alpha^{3j}$$

This implies that $S_1^3 = S_3 + \alpha^i\alpha^j S_1 \neq S_3$. Furthermore, $x_1 = \alpha^{-i}$ and $x_2 = \alpha^{-j}$ are roots of the equation

$$1 + S_1 x + \frac{S_1^3 + S_3}{S_1} x^2 = 0 \tag{4}$$

This gives the following procedure to correct two errors:

- Compute S_1 and S_3.
- If $S_1 = S_3 = 0$, then assume that no errors have occurred.
- Else, if $S^3 = S_1^3 \neq 0$, then one error has occurred in the ith position determined by $S_1 = \alpha^i$.
- Else (if $S_3 \neq S_1^3$), consider the equation

$$1 + S_1 x + (S_1^3 + S_3)/S_1 x^2 = 0$$

If the equation has two roots α^{-i} and α^{-j}, then errors have occurred in positions i and j.

Else (if the equation has no roots in F_{2^m}), then more than two errors have occurred.

Similar explicit expressions (in terms of the syndrome) for the coefficients of an equation with the error positions as roots can be found for t error-correcting BCH codes when $t = 3$, $t = 4$, etc., but they become increasingly complicated. However, there is an efficient algorithm for determining the equation, and we describe this is some detail next.

Let α be a primitive element in F_{2^m}. A parity-check matrix for the primitive t error-correcting BCH code is

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^3 & \alpha^6 & \cdots & \alpha^{3(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{2t-1} & \alpha^{2(2t-1)} & \cdots & \alpha^{(2t-1)(n-1)} \end{pmatrix}$$

where $n = 2^m - 1$. Suppose errors have occurred in positions $i_1, i_2, \ldots, i_\tau$, where $\tau \leq t$. Let $X_j = \alpha^{i_j}$ for $j = 1, 2, \ldots, \tau$. The *error locator polynomial* $\Lambda(x)$ is defined by

$$\Lambda(x) = \prod_{j=1}^{\tau} (1 + X_j x) = \sum_{l=0}^{\tau} \lambda_l x^l$$

The roots of $\Lambda(x) = 0$ are X_j^{-1}. Therefore, if we can determine $\Lambda(x)$, then we can determine the locations of the errors. Expanding the expression for $\Lambda(x)$, we get

$$\begin{aligned} \lambda_0 &= 1, \\ \lambda_1 &= X_1 + X_2 + \cdots + X_\tau, \\ \lambda_2 &= X_1X_2 + X_1X_3 + X_2X_3 + \cdots + X_{\tau-1}X_\tau, \\ \lambda_3 &= X_1X_2X_3 + X_1X_2X_4 + X_2X_3X_4 \\ &\quad + \cdots + X_{\tau-2}X_{\tau-1}X_\tau, \\ &\vdots \\ \lambda_\tau &= X_1X_2 \cdots X_\tau, \\ \lambda_l &= 0 \text{ for } l > \tau \end{aligned}$$

Hence λ_l is the lth elementary symmetric function of $X_1, X_2, \ldots, X_\tau$.

From the syndrome we get $S_1, S_3, \ldots, S_{2t-1}$, where

$$\begin{aligned} S_1 &= X_1 + X_2 + \cdots + X_\tau, \\ S_2 &= X_1^2 + X_2^2 + \cdots + X_\tau^2, \\ S_3 &= X_1^3 + X_2^3 + \cdots + X_\tau^3, \\ &\vdots \\ S_{2t} &= X_1^{2t} + X_2^{2t} + \cdots + X_\tau^{2t} \end{aligned}$$

Further,

$$S_{2r} = X_1^{2r} + X_2^{2r} + \cdots + X_\tau^{2r} = (X_1^r + X_2^r + \cdots + X_\tau^r)^2 = S_r^2$$

for all r. Hence, from the syndrome we can determine the polynomial

$$S(x) = 1 + S_1 x + S_2 x^2 + \cdots + S_{2t} x^{2t}$$

The *Newton equations* are a set of relations between the power sums S_r and the symmetric functions λ_l—namely,

$$\sum_{j=0}^{l-1} S_{l-j}\lambda_j + l\lambda_l = 0 \text{ for } l \geq 1$$

Let

$$\Omega(x) = S(x)\Lambda(x) = \sum_{l \geq 0} \omega_l x^\ell \tag{5}$$

Since $\omega_l = \sum_{j=0}^{l-1} S_{l-j}\lambda_j + \lambda_l$, the Newton equations imply that

$$\omega_l = 0 \text{ for all odd } l, 1 \leq l \leq 2t-1 \tag{6}$$

The *Berlekamp–Massey algorithm* is an algorithm that, given $S(x)$, determines the polynomial $\Lambda(x)$ of smallest degree such that Eq. (6) is satisfied, where the ω_l are defined by Eq. (5). The idea is, for $r = 0, 1, \ldots, t$, to determine polynomials $\Lambda^{(r)}$ of lowest degree such that

$$\omega_l^{(r)} = 0 \text{ for all odd } l, 1 \leq l \leq 2r-1$$

where

$$\sum_{l \geq 0} \omega_l^{(r)} x^l = S(x)\Lambda^{(r)}(x)$$

For $r = 0$, we can clearly let $\Lambda^{(0)}(x) = 1$. We proceed by induction. Let $0 \leq r < t$, and suppose that polynomials $\Lambda^{(\rho)}(x)$ have been constructed for $0 \leq \rho \leq r$. If $\omega_{2r+1}^{(r)} = 0$, then we can choose

$$\Lambda^{(r+1)}(x) = \Lambda^{(r)}(x)$$

If, on the other hand, $\omega_{2r+1}^{(r)} \neq 0$, then we modify $\Lambda^{(r)}(x)$ by adding another suitable polynomial. There are two cases to consider. First, if $\Lambda^{(r)}(x) = 1$ [in which case $\Lambda^{(r)}(x) = 1$ for $0 \leq \tau \leq r$], then

$$\Lambda^{(r+1)}(x) = 1 + \omega_{2r+1}^{(r)} x^{2r+1}$$

will have the required property. If $\Lambda^{(r)}(x) \neq 1$, then there exists a maximal positive integer $\rho < r$ such that $\omega_{2\rho+1}^{(\rho)} \neq 0$ and we add a suitable multiple of $\Lambda^{(\rho)}$:

$$\Lambda^{(r+1)}(x) = \Lambda^{(r)}(x) + \omega_{2r+1}^{(r)} (\omega_{2\rho+1}^{\rho})^{-1} x^{2r-2\rho} \Lambda^{(\rho)}(x)$$

We note that this implies that

$$\Lambda^{(r+1)}(x)S(x) = \sum_{l \geq 0} w_l^{(r)} x^l + \omega_{2r+1}^{(r)} (\omega_{2\rho+1}^{(\rho)})^{-1} \sum_{l \geq 0} \omega_l^{(\rho)} x^{l+2r-2\rho}$$

Hence for odd l we get

$$\omega_l^{(r+1)} = \begin{cases} \omega_l^{(r)} = 0 \\ \qquad \text{for } 1 \leq l \leq 2r-2\rho-1, \\ \omega_l^{(r)} + \omega_{2r+1}^{(r)} (\omega_{2\rho+1}^{(\rho)})^{-1} \omega_{l-2r+2\rho}^{(\rho)} = 0 + 0 = 0 \\ \qquad \text{for } 2r-2\rho+1 \leq l \leq 2r-1, \\ \omega_{2r+1}^{(r)} + \omega_{2r+1}^{(r)} (\omega_{2\rho+1}^{(\rho)})^{-1} \omega_{2\rho+1}^{(\rho)} = \omega_{2r+1}^{(r)} + \omega_{2r+1}^{(r)} = 0 \\ \qquad \text{for } l = 2r+1 \end{cases}$$

We now formulate these ideas as an algorithm (in a Pascal-like syntax). In each step we keep the present $\Lambda(x)$ [the superscript (r) is dropped] and the modifying polynomial [$x^{2r-2\rho-1}$ or $(\omega_{2\rho+1}^{(\rho)})^{-1} x^{2r-2\rho-1}\Lambda^{(\rho)}(x)$], which we denote by $B(x)$.

Berlekamp–Massey algorithm in the binary case

```
Input: t and S(x).

Λ(x) := 1;   B(x) := 1;
for r := 1 to t do
begin
      ω := coefficient of x^(2r-1) in S(x)Λ(x);
      if ω = 0  then B(x) := x²B(x)
                else [Λ(x), B(x)] := [Λ(x) +
                 ωxB(x), xΛ(x)/ω]
end;
```

The assignment following the `else` is two assignments to be done in parallel; the new $\Lambda(x)$ and $B(x)$ are computed from the old ones.

The Berlekamp–Massey algorithm determines the polynomial $\Lambda(x)$. To find the roots of $\Lambda(x) = 0$, we try all possible elements of F_{2^m}. In practical applications, this can be efficiently implemented using shift registers (usually called the *Chien search*).

Example. We consider the [15, 7, 5] double-error correcting BCH code; that is, $m = 4$ and $t = 2$. As a primitive element, we choose α such that $\alpha^4 = \alpha + 1$. Suppose that we have received a vector with syndrome $(S_1, S_3) = (\alpha^4, \alpha^5)$. Since $S_3 \neq S_1^3$, at least two errors have occurred. Equation (4) becomes

$$1 + \alpha^4 x + \alpha^{10} x^2 = 0$$

which has the zeros α^{-3} and α^{-7}. We conclude that the received vector has two errors (namely, in positions 3 and 7).

Now consider the Berlekamp–Massey algorithm for the same example. First we compute $S_2 = S_1^2 = \alpha^8$ and $S_4 = S_2^2 =$

α. Hence

$$S(x) = 1 + \alpha^4 x + \alpha^8 x^2 + \alpha^5 x^3 + \alpha x^4$$

The values of r, ω, $\Lambda(x)$, and $B(x)$ after each iteration of the for-loop in the Berlekamp–Massey algorithm are shown in the following table:

r	ω	$\Lambda(x)$	$B(x)$
		1	1
1	α^4	$1 + \alpha^4 x$	$\alpha^{11} x$
2	α^{14}	$1 + \alpha^4 x + \alpha^{10} x^2$	$\alpha x + \alpha^5 x^2$

Hence, $\Lambda(x) = 1 + \alpha^4 x + \alpha^{10} x^2$ (as before).

Now consider the same code with syndrome of received vector $(S_1, S_3) = (\alpha, \alpha^9)$. Since $S_3 \neq S_1^3$, at least two errors have occurred. We get

$$\Lambda(x) = 1 + \alpha x + x^2$$

However, the equation $1 + \alpha x + x^2 = 0$ does not have any roots in F_{2^4}. Hence, at least three errors have occurred, and the code is not able to correct them.

REED–SOLOMON CODES

In the previous sections we have considered binary codes where the components of the codewords belong to the finite field $F_2 = \{0, 1\}$. In a similar way we can consider codes with components from any finite field F_q.

The *Singleton bound* states that for any $[n, k, d]$ code with components from F_q, we have

$$d \leq n - k + 1$$

A code for which $d = n - k + 1$ is called *maximum distance separable* (MDS). The only binary MDS codes are the trivial $[n, 1, n]$ repetition codes and $[n, n - 1, 2]$ even-weight codes. However, there are important nonbinary MDS codes (in particular, the Reed–Solomon codes, which we now will describe).

Reed–Solomon codes are t error-correcting cyclic codes with symbols from a finite field F_q, even though they can be constructed in many different ways. They can be considered as the simplest generalization of BCH codes. Since the most important case for applications is $q = 2^m$, we consider this case here. Each symbol is then an element in F_{2^m} and can be considered as an m-bit symbol.

The construction of a cyclic Reed–Solomon code is as follows: Let α be a primitive element of F_{2^m}. Since $\alpha^i \in F_{2^m}$ for all i, the minimal polynomial of α^i over F_{2^m} is just $x + \alpha^i$. The generator polynomial of a (primitive) t error-correcting Reed–Solomon code of length $2^m - 1$ has $2t$ consequtive powers of α as zeros:

$$\begin{aligned} g(x) &= \prod_{i=0}^{2t-1} (x + \alpha^{b+i}) \\ &= g_0 + g_1 x + \cdots + g_{2t-1} x^{2t-1} + x^{2t} \end{aligned}$$

The code has the following parameters:

Block length: $n = 2^m - 1$
Number of parity-check symbols: $n - k = 2t$
Minimum distance: $d = 2t + 1$

Thus, the Reed–Solomon codes satisfy the Singleton bound with equality $n - k = d - 1$. That is, they are MDS codes.

The weight distribution of the Reed–Solomon code is (for $i \geq d$)

$$A_i = \binom{n}{i} \sum_{j=0}^{i-d} (-1)^j \binom{i}{j} (2^{m(i-d-j+1)} - 1)$$

The encoding of Reed–Solomon codes is similar to the encoding of binary cyclic codes. The decoding is similar to the decoding of binary BCH codes with one added complication. Using a generalization of the Berlekamp–Massey algorithm, we determine the polynomials $\Lambda(x)$ and $\Omega(x)$. From $\Lambda(x)$ we can determine the locations of the errors. In addition, we have to determine the value of the errors (in the binary case the values are always 1). The value of the error at location X_j can easily be determined using $\Omega(x)$ and $\Lambda(x)$; we omit further details.

NONLINEAR CODES FROM CODES OVER Z_4

In the previous sections we have mainly considered binary *linear* codes; that is, codes where the sum of two codewords is again a codeword. The main reason has been that the linearity greatly simplified construction and decoding of the codes.

A binary nonlinear (n, M, d) code C is simply a set of M binary n-tuples with pairwise distance at least d, but without any further imposed structure. In general, to find the minimum distance of a nonlinear code one has to compute the distance between all pairs of codewords. This is, of course, more complicated than for linear codes, where it suffices to find the minimum weight among all the nonzero codewords. The lack of structure in a nonlinear code also makes it quite difficult to decode in an efficient manner.

There are, however, some advantages to nonlinear codes. For given values of length n and minimum distance d, it is sometimes possible to construct nonlinear codes with more codewords than is possible for linear codes. For example, for $n = 16$ and $d = 6$ the best linear code has dimension $k = 7$ (i.e., it contains 128 codewords). The code of length 16 obtained by extending the double-error-correcting primitive BCH code has these parameters.

In 1967, Nordstrom and Robinson found a nonlinear code with parameters $n = 16$ and $d = 6$ containing $M = 256$ codewords, which has twice as many codewords as the best linear code for the same values of n and d.

In 1968, Preparata generalized this construction to an infinite family of codes having parameters

$$(2^{m+1}, 2^{2^{m+1}-2m-2}, 6), m \text{ odd}, m \geq 3$$

A few years later, in 1972, Kerdock gave another generalization of the Nordstrom–Robinson code and constructed another infinite class of codes with parameters

$$(2^{m+1}, 2^{2m+2}, 2^m - 2^{(m-1)/2}), m \text{ odd}, m \geq 3$$

The Preparata code contains twice as many codewords as the extended double-error-correcting BCH code and is optimal in the sense of having the largest possible size for the given length and minimum distance. The Kerdock code has twice as

many codewords as the best known linear code. In the case $m = 3$ the Preparata code and the Kerdock codes both coincide with the Nordstrom–Robinson code.

The Preparata and Kerdock codes are *distance invariant.* This means that the distance distribution from a given codeword to all the other codewords is independent of the given codeword. In particular, since they contain the all-zero codeword, their weight distribution equals their distance distribution.

In general, there is no natural way to define the dual code of a nonlinear code, and thus the MacWilliams identities have no meaning for nonlinear codes. However, one can define the weight enumerator polynomial $A(z)$ of a nonlinear code in the same way as for linear codes and compute its *formal dual* $B(z)$ from the MacWilliams identities:

$$B(z) = \frac{1}{M}(1+z)^n A\left(\frac{1-z}{1+z}\right)$$

The polynomial $B(z)$ obtained in this way has no simple interpretation. In particular, it may have coefficients that are noninteger or even negative. For example, if $C = \{(110), (101), (111)\}$, then $A(z) = 2z^2 + z^3$ and $B(z) = (3 - 5z + z^2 + z^3)/3$.

An observation that puzzled the coding theory community for a long time was that the weight enumerator of the Preparata code $A(z)$ and the weight enumerator of the Kerdock code $B(z)$ satisfied the MacWilliams identities, and in this sense these nonlinear codes behaved like dual linear codes.

Hammons, Kumar, Calderbank, Sloane, and Solé (*IEEE Trans. Information Theory* **40**: 301–319, 1994) gave a significantly simpler description of the family of Kerdock codes. They constructed a linear code over $Z_4 = \{0, 1, 2, 3\}$, which is an analog of the binary first-order Reed–Muller code. This code is combined with a mapping called the Gray map that maps the elements of Z_4 into binary pairs. The Gray map ϕ is defined by

$$\phi(0) = 00, \phi(1) = 01, \phi(2) = 11, \phi(3) = 10$$

The Lee weight of an element in Z_4 is defined by

$$w_L(0) = 0, w_L(1) = 1, w_L(2) = 2, w_L(3) = 1$$

Extending ϕ in a natural way to a map ϕ: $Z_4^n \to Z_2^{2n}$, one observes that ϕ is a distance preserving map from Z_4^n (under the Lee metric) to Z_2^{2n}, (under the Hamming metric).

A linear code over Z_4 is a subset of Z_4^n such that any linear combination of two codewords is again a codeword. From a linear code $\mathscr{C}$ of length n over Z_4, one obtains a binary code $C = \phi(\mathscr{C})$ of length $2n$ by replacing each component in a codeword in $\mathscr{C}$ by its image under the Gray map. This code is usually nonlinear.

The minimum Hamming distance of C equals the minimum Lee distance of $\mathscr{C}$ and is equal to the minimum Lee weight of $\mathscr{C}$ since $\mathscr{C}$ in linear over Z_4.

Example. To obtain the Nordstrom–Robinson code, we will construct a code over Z_4 of length 8 and then apply the Gray map.

Let $f(x) = x^3 + 2x^2 + x + 3 \in Z_4[x]$. Let β be a zero of $f(x)$; that is, $\beta^3 + 2\beta^2 + \beta + 3 = 0$. Then we can express all powers of β in terms of 1, β, and β^2, as follows:

$$\begin{aligned}
\beta^3 &= 2\beta^2 + 3\beta + 1 \\
\beta^4 &= 3\beta^2 + 3\beta + 2 \\
\beta^5 &= \beta^2 + 3\beta + 3 \\
\beta^6 &= \beta^2 + 2\beta + 1 \\
\beta^7 &= 1
\end{aligned}$$

Consider the code $\mathscr{C}$ over Z_4 with generator matrix given by

$$\begin{aligned}
G &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & \beta & \beta^2 & \beta^3 & \beta^4 & \beta^5 & \beta^6 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 2 & 3 & 1 \\ 0 & 0 & 1 & 0 & 3 & 3 & 3 & 2 \\ 0 & 0 & 0 & 1 & 2 & 3 & 1 & 1 \end{bmatrix}
\end{aligned}$$

where the column corresponding to β^i is replaced by the coefficients in its expression in terms of 1, β, and β^2. Then the Nordstrom–Robinson code is the Gray map of $\mathscr{C}$.

The dual code $\mathscr{C}^\perp$ of a code $\mathscr{C}$ over Z_4 is defined similarly as for binary linear codes, except that the inner product of the vectors $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ with components in Z_4 is defined by

$$(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} x_i y_i \pmod 4$$

The dual code $\mathscr{C}^\perp$ of $\mathscr{C}$ is then

$$\mathscr{C}^\perp = \{\boldsymbol{x} \in Z_4^n (\boldsymbol{x}, \boldsymbol{c}) = 0 \text{ for all } \boldsymbol{c} \in Z_4^n\}$$

For a linear code $\mathscr{C}$ over Z_4, there is a MacWilliams relation that determines the Lee weight distribution of the dual code $\mathscr{C}^\perp$ from the Lee weight distribution of $\mathscr{C}$. Therefore, one can compute the relation between the Hamming weight distributions of the nonlinear codes $C = \phi(\mathscr{C})$ and $C_\perp = \phi(\mathscr{C}^\perp)$, and it turns out that the MacWilliams identities hold.

Hence, to find nonlinear binary codes related by the MacWilliams identities, one can start with a pair of Z_4-linear dual codes and apply the Gray map. For any odd integer $m \geq 3$, the Gray map of the code $\mathscr{K}_m$ over Z_4 with generator matrix

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & \beta & \beta^2 & \cdots & \beta^{2^m-2} \end{bmatrix}$$

is the binary nonlinear $(2^{m+1}, 2^{2m+2}, 2^m - 2^{(m-1)/2})$ Kerdock code. The Gray map of $\mathscr{K}_m^\perp$ has the same weight distribution as the $(2^{m+1}, 2^{2^{m+1}-2m-2}, 6)$ Preparata code. It is, however, not identical to the Preparata code and is therefore denoted the "Preparata" code. Hence the Kerdock code and the "Preparata" code are the Z_4-analogy of the first-order Reed–Muller code and the extended Hamming code, respectively.

Hammons, Kumar, Calderbank, Sloane, and Solé also showed that the binary code defined by $C = \phi(\mathscr{C})$, where $\mathscr{C}$ is

the quaternary code with parity-check matrix given by

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & \beta & \beta^2 & \cdots & \beta^{2^m-2} \\ 0 & 2 & 2\beta^3 & 2\beta^6 & \cdots & 2\beta^{3(2^m-2)} \end{bmatrix}$$

is a binary nonlinear $(2^{m+1}, 2^{2^{m+1}-3m-2}, 8)$ code whenever $m \geq 3$ is odd. This code has the same weight distribution as the Goethals code, which is a nonlinear code that has four times as many codewords as the comparable linear extended triple-error-correcting primitive BCH code. The code $C_\perp = \phi(\mathscr{C}^\perp)$ is identical to a binary nonlinear code that was constructed in a much more complicated way by Delsarte and Goethals more than 20 years ago.

To analyze codes obtained from codes over Z_4 in this manner, one is led to study Galois rings instead of Galois fields. Similar to a Galois field, a Galois ring can be defined as $Z_{p^e}[x]/(f(x))$, where $f(x)$ is a monic polynomial of degree m that is irreducible modulo p. The richness in structure of the Galois rings has led to several recently discovered good nonlinear codes that have an efficient and fast decoding algorithm.

Reading List

R. Blahut, *The Theory and Practice of Error Control Codes.* Reading, MA: Addison-Wesley, 1983.

R. Hill, *A First Course in Coding Theory.* Oxford: Clarendon Press, 1986.

T. Kløve and V. I. Korzhik, *Error-Detecting Codes,* Boston, MA: Kluwer Academic, 1995.

R. Lidl and H. Niederreiter, *Finite Fields,* vol. 20 of *Encyclopedia of Mathematics and Its Applications.* Reading, MA: Addison-Wesley, 1983.

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

F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes,* Amsterdam: North-Holland, 1977.

W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes.* Cambridge, MA: MIT Press, 1972.

M. Purser, *Introduction to Error-Correcting Codes,* Boston, MA: Artech House, 1995.

J. H. van Lint, *Introduction to Coding Theory,* New York: Springer-Verlag, 1982.

H. van Tilborg, *Error-Correcting Codes—A First Course,* Lund: Studentlitteratur, 1993.

S. A. Vanstone and P. C. van Oorschot, *An Introduction to Error-Correcting Codes with Applications,* Boston, MA: Kluwer Academic, 1989.

TOR HELLESETH TORLEIV KLØVE
University of Bergen

ALGEBRA, LINEAR. See LINEAR ALGEBRA.

ALGEBRA, PROCESS. See PROCESS ALGEBRA.

ALGORITHMIC DIFFERENTIATION. See AUTOMATIC DIFFERENTIATION.

ALGORITHMS. See DIVIDE AND CONQUER METHODS.

ALGORITHMS AND DATA STRUCTURES. See DATA STRUCTURES AND ALGORITHMS.

ALGORITHMS FOR BACKTRACKING. See BACKTRACKING.

ALGORITHMS FOR RECURSION. See RECURSION.

ALGORITHMS, GENETIC. See GENETIC ALGORITHMS.

ALGORITHMS, MULTICAST. See MULTICAST ALGORITHMS.

ALGORITHMS, ONLINE. See ONLINE OPERATION.

} { { } }

Wiley Encyclopedia of Electrical and Electronics Engineering

Cryptography
Standard Article
Yvo G. Desmedt
[1]University of Wisconsin—Milwaukee, Milwaukee, WI
[2]University of London, UK

DOI: 10.1002/047134608X.W4202
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (116K)

Abstract

The sections in this article are

Fundamentals

Tools

Algorithms Based on Number Theory and Algebra

Conclusion

Reading List

CRYPTOGRAPHY

Cryptography is the science and study of the security aspects of communications and data in the presence of a malicious adversary. Cryptanalysis is the study of methods used to break cryptosystems. Cryptographic schemes and protocols are being and have been developed to protect data. Until 1974, only privacy issues were studied, and the main users were diplomats and the military (1). Systems are also being deployed to guarantee integrity of data, as well as different aspects of authenticity and to identify individuals or computers (called entity authenticity). Emerging topics of study include anonymity and traceability, authorized wiretapping (called law enforcement), copyright, digital contracts, freedom of speech, revocation of rights, timestamping, witnessing, etc. Related disciplines are computer security, network security, physical security (including tempest), spread spectrum, and steganography.

Fast computers and advances in telecommunications have made high-speed, global, widespread computer networks possible, in particular the Internet, which is an open network. It has increased the access to databases, such as the open World Wide Web. To decrease communication cost and to be user-friendly, private databases containing medical records, proprietary information, tax information, etc., are often accessible via the Internet by using a low-security password scheme.

The privacy of data is obviously vulnerable during communication, and data in transit can be modified, in particular in

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*.

open networks. Because of the lack of secure computers, such concerns extend to stored data. Data communicated and/or accessible over such networks include bank and other financial transactions, love letters, medical records, proprietary information, etc., whose privacy must be protected. The authenticity of (the data in) contracts, databases, electronic commerce, etc. must be protected against modifications by an outsider or by one of the parties involved in the transaction. Modern cryptography provides the means to address these issues.

FUNDAMENTALS

To protect data, one needs to know what type of attacks the untrusted party (enemy) can use. These depend on the security needs. The two main goals of modern cryptography are privacy and authenticity. The issue of protecting privacy is discussed now.

Privacy

The threat undermining privacy is eavesdropping. The untrusted party, called the eavesdropper, will have access to the transmitted or stored data, for example, by tapping the line or capturing (even rather minimal) electromagnetic interference from a screen. To protect the data, called the plaintext or cleartext, it is transformed into ciphertext. This transformation is called encryption. To achieve security, it should be difficult for the eavesdropper to cryptanalyze, that is, to recover the plaintext from the ciphertext. However, to guarantee usefulness, the legitimate receiver should be able to recover the plaintext. Such an operation is called decryption and uses a key k. To guarantee that only the legitimate receiver is able to decrypt, obviously this key must remain secret. If the sender wants to send data to different receivers, the encryption algorithm must use a parameter k', specifying the receiver, as extra input. For historical reasons this parameter has been called a (encryption) key, which is discussed in more detail later.

The person who attempts a cryptanalysis, called a cryptanalyst, may in some circumstances know a previously encrypted plaintext when trying to break the current ciphertext. Such an attack is called a known-plaintext attack, distinguishing it from the more basic ciphertext-only attack in which only the ciphertext is available to the cryptanalyst. Even more powerful attacks, especially in the commercial world, are feasible, such as a chosen-plaintext attack, in which the cryptanalyst chooses one (or more) plaintext(s). A company achieves this by sending a ciphertext to a local branch of a competing company that will most likely send the corresponding plaintext to its headquarters and encrypt it with a key the first party wants to break (1). In a variant of this type of attack the cryptanalyst sends a chosen ciphertext to the receiver. The plaintext is likely to be garbled and thrown in the bin. If the garbage collectors collaborate with the cryptanalyst, the latter has started a chosen-ciphertext attack. In the strongest subtype of chosen-text attacks the text chosen may depend on (previous or) other texts, and therefore it is called adaptive.

Authenticity

A document is authentic if it originated from the claimed source and if its content has not been modified. So, now the cryptanalyst is allowed to try to inject fraudulent messages and attempt to alter the data. Therefore one calls the cryptanalyst an active eavesdropper. To protect the data, one appends a message authentication code, abbreviated as MAC. If there is no concern for privacy, the message itself is sent in the clear. Only the legitimate sender should be allowed to generate a MAC. Therefore the sender needs to know a secret key k. If the key were not secret, anybody could impersonate the sender. So, the authenticator generation algorithm has the message and the sender's secret key as input. To check the authenticity of a message, the receiver runs a verification algorithm. If the algorithm's outputs "false," then the message is definitely not authentic and must be rejected and discarded. If the output is "satisfactory," very likely the message is authentic and is accepted. One cannot give a 100% guarantee that the message is authentic because the active eavesdropper could be very lucky, but one can approach the 100% margin as closely as desired. If the receiver wants to verify the authenticity of messages originating from different senders, the verification algorithm must use a parameter k', specifying the sender, as extra input. For historical reasons this parameter has been called a key, which is discussed in more detail later.

In all types of attacks the active eavesdropper is allowed to see one (or more) authenticated message(s). In chosen-text attacks, the cryptanalyst can choose a text which the sender will authenticate and/or send messages with a (fictitious) MAC(s). In the latter case, it is assumed that the active eavesdropper can find out whether the message was accepted or rejected.

Public Key Systems

One can wonder whether k' must remain secret, which is discussed now. If it is easy to compute k from k', it is obvious that k' must also remain secret. Then the key must be unique to a sender–receiver pair. This introduces a key management problem, since this key has to be transmitted in a secure way. In this case, the cryptosystem is called a conventional or symmetric cryptosystem and k, k' usually coincide.

On the other hand, if it is hard to compute k from k' and hard to compute a $\overline{k}$, which allows partial cryptanalysis, then the key k' can be made public. This concept was invented by Diffie and Hellman (2) and independently by Merkle (3). Such a system is called a public key (or sometimes an asymmetric cryptosystem). This means that for privacy protection each receiver R publishes a personal k_R, and for authentication, the sender S makes k_S public. In the latter case the obtained authenticator is called a digital signature because anyone who knows the correct public key k_S can verify the correctness.

Note that the sender can claim that the secret key was stolen or that k_S was published without consent. That would allow a denial of ever having sent a message (4). Such situations must be dealt with by an authorized organization. If high security is desired, the MAC of the message must be deposited with a notary public. Another solution is digital time stamping (5) based on cryptography (the signer needs to alert an authority that his public key must have been stolen or lost).

If the public key is not authentic, the one who created the fake public key can decrypt messages intended for the legitimate receiver or can sign claiming to be the sender (6). So

then the security is lost. In practice, this problem is solved as follows. A known trusted entity(ies), for example, an authority, certifies that the key K_S corresponds to S, and therefore signs (S, K_S). This signature is called a certificate.

Security Levels

There are different levels of security in modern cryptography, depending on whether information theory, physics (in particular quantum physics), computational complexity theory, or heuristics, has been used. To be more precise, when the computer power of the opponent is allowed to be unbounded and one can mathematically prove that a formal definition of security is satisfied, then one is speaking about unconditional security. Information theory and probability theory is used to achieve this level of security. Evidently the formal definition of security must sufficiently model real-world security need(s).

In quantum cryptography one assumes the correctness of the laws of quantum physics (7).

A system or protocol is proven secure, relative to an assumption, when one can mathematically prove the following statement. The latter being, if the assumption is true, then a formal security definition is satisfied for that system or protocol. Such an assumption is typically an unproven claim in computational complexity theory, such as the presumed hardness of factoring large integers, or to compute discrete logarithm in finite groups. In this model the users and the opponent have only a computer power bounded by a polynomial in function of the length of a security parameter and one states that a system is secure if it requires superpolynomial (that is, growing faster to infinity than any polynomial) time to break it. One should note that this model is limited. Indeed, when using a cryptosystem, one needs to choose a security parameter which fixes the length.

In practice, a system is secure if the enemy needs the computer time of all computers on earth working in parallel, and the user needs, varying from application to application, 1 nanosecond up to a few minutes. However, modern theoretical computer science cannot guarantee that a certain number of basic operations are needed to break a cryptosystem. So, new algorithms may be developed that break cryptosystems faster than the previously best algorithms. Moreover, new technology makes computers faster each day. The impact of new algorithms and new hardware is clear from the following example. In 1977, it was estimated that factoring a 129 digit integer (product of two primes) would take 40 quadrillion (that is 4×10^{16}) years, whereas it was actually factored in 1993–1994 using the idle time of approximately 1600 computers on the Internet for 234 days (8).

A cryptosystem or protocol is as secure as another if one can mathematically prove that a new attack on the first scheme implies a new attack against the other and vice versa.

Finally, the weakest form of security is called heuristic. A system is heuristically secure if no (significant) attack has been found. Many modern but practical cryptosystems have such a level of security.

TOOLS

Many tools are used to achieve the desired security properties. These are based on discrete mathematics from several disciplines (mainly algebra, combinatorics, number theory, and probability theory) and our state-of-the-art knowledge of computer science (in particular, the study of (efficient) algorithms, algorithmic number theory, and computational complexity). Software engineering is used to design software implementations. Electrical engineering plays a role in hardware implementations, and information theory is also used, in particular, to construct unconditionally secure cryptosystems. Some of the main tools are explained briefly now.

The One-Time Pad

The one-time pad (9), also called the Vernam scheme, was originally designed to achieve privacy. Shannon (10), who invented information theory to study cryptography, proved the unconditional security of the scheme when used for privacy. The scheme has become a cornerstone of cryptography and is used as a principle in a wide range of seemingly unrelated contexts.

Shannon defined an encryption system as perfect when, for a cryptanalyst not knowing the secret key, the message m is independent of the ciphertext c.

In the original scheme the plaintext is represented in binary. Before encrypting the binary message, the sender and receiver have obtained a secret key, a binary string chosen uniformly at random. When m_i is the ith plaintext bit, k_i the ith key bit and c_i the ith ciphertext bit, in the Vernam scheme $c_i = m_i \oplus k_i$, where $\oplus$ is the exclusive-or, also known as exor. To decrypt, the receiver computes $m_i = c_i \oplus k_i^{-1}$, where in the case of the exclusive-or $k^{-1} = k$. The key is used only once. This implies that if the sender needs to encrypt a new message, then a new key is chosen, which explains the terminology: one-time pad. In modern applications, the exor is often replaced by a group operation.

Secret Sharing

A different interpretation of the one-time pad has recently been given (11–13). Suppose that one would like to make a backup of a secret m with bits m_i. If it is put into only one safe, a thief who breaks open the safe will find it. So, it is put in two safes so that a thief who breaks open one safe is unable to recover the secret.

The solution to this problem is to choose a uniformly random string of bits k_i (as many as there are bits in the message). One stores the bits k_i in the first safe and the bits $c_i = m_i \oplus k_i$ in the second. Given the content of both safes, one can easily recover the secret.

In the previous discussion, it is assumed that two safes would not be broken into, but only one at the most. If one fears that the thief may succeed in opening more, one could proceed as follows. Choose uniformly random $(t - 1)$ elements $s_1, s_2, \ldots, s_{t-1}$ in a finite group $S(+)$ and (assuming $m \in S$) construct $s_t = m - (s_1 + s_2 + \cdots + s_{t-1})$. Put s_i $(1 \leq i \leq t)$ in safe i. An example of such a group is $GF(2^n)(+)$ where n is the length of the message m. When $t = 2$, this corresponds to the one-time pad. One calls s_i $(1 \leq i \leq t)$ a share of the secret m, and the one who knows the share is called a shareholder or participant. Then it is easy to prove that the eavesdropper who opens $(t - 1)$ safes learns nothing about the secret. Only by opening all the safes is one able to recover the secret m.

A major disadvantage of this scheme is that it is unreliable. Indeed if one share is destroyed, for example, by an earthquake, the secret m is lost. A t-out-of-l secret sharing scheme is the solution. In such a scheme, one has l shares, but only t are required to recover the secret, whereas $(t - 1)$ are useless. An example of such a secret sharing scheme is discussed later on.

The concept of secret sharing was generalized, allowing one to specify in more detail who can recompute the secret and who cannot (14). Although previous secret sharing schemes protect reliability and privacy, they do not protect correctness and authenticity. Indeed, a shareholder could reveal an incorrect share, which (very likely) implies the reconstruction of an incorrect secret. When one can demonstrate the correctness of the shares, it is called verifiable secret sharing.

One-Way Functions

Cryptography based on computational complexity relies on one-way functions. A function(s) f is one-way if it is easy to compute f, and, given an image y, it is hard to find an x such that $y = f(x)$.

The state-of-the-art of computational complexity does not allow one to prove that one-way functions exist. For some functions f no efficient algorithm has been developed so far to invert f, and in modern cryptography it is often assumed that such functions *are* one-way.

One-way functions have many applications in modern cryptography. For example, it has been proven that a necessary and sufficient condition for digital signatures is a one-way function(15,16).

Block Ciphers

A blockcipher is a cryptosystem in which the plaintext and ciphertext are divided into strings of equal length, called blocks, and each block is encrypted one at a time with the same key.

To obtain acceptable security, a block cipher requires a good mode (17). Indeed, patterns of characters are very common. For example, subsequent spaces are often used in text processors. Common sequences of characters are also not unusual. For example, " from the " corresponds to 10 characters, which is 80 bits. In the Electronic Code Book (ECB) mode, the plaintext is simply divided into blocks that are then encrypted. Frequency analysis of these blocks allows one to find such very common blocks. This method allows one to find a good fraction of the plaintext and often the complete plaintext if the plaintext that has been encrypted is sufficiently long. Good modes have been developed based on feedback and feedforward.

Many block ciphers have been designed. Some of the most popular ones are the US Data Encryption Standard (DES), the Japanese NTT (Nippon Telegraph and Telephone Corporation), Fast Encipherment ALgorithm (FEAL), the "International Data Encryption Algorithm" (IDEA) designed by Lai (Switzerland), RC2, and RC5. DES (18), an ANSI (American National Standards Institute) and NIST (National Institute of Standards and Technology, US) standard for roughly 20 years, is being replaced by the Advanced Encryption Standard (AES), currently under development.

Hash Function

A hash function h is a function with n bits of input and m bits of output, where $m < n$. A cryptographic hash function needs to satisfy the following properties:

1. It is a one-way function.
2. Given x, it is hard to find an $x' \neq x$ such that $h(x) = h(x')$.
3. It is hard to find an x and an $x' \neq x$ such that $h(x) = h(x')$.

Note that the second property does *not* necessarily imply the third.

Several modes of block ciphers allow one to make cryptographic hash functions. A cryptographic hash function is an important tool for achieving practical authentication schemes. When signing a message digitally, first one pads it, and then one uses a cryptographic hash function before using the secret key to sign.

Universal hash functions are another type of hash function. These are used in unconditionally secure settings.

When referring to a hash function in applied cryptography, one means a cryptographic hash function.

Pseudonoise Generators and Stream Ciphers

A problem with the one-time pad is that the key can be used only once. The key must be transported by a secure path. In the military and diplomatic environment, this is often done by a trusted courier (using secret sharing, trust in the courier can be reduced). However, these requirements are unrealistic commercially.

The goal of a pseudonoise (or pseudorandom) generator is to output a binary string whose probability distribution is (computationally) indistinguishable from a uniformly random binary string. The pseudonoise generator starts from a *seed,* which is a relatively short binary string chosen uniformly random.

When one replaces the one-time key in the Vernam scheme by the output of a pseudorandom generator, this is called a stream cipher. Then the sender and receiver use the seed as the secret key. It has been proven that if the pseudonoise is (computationally) indistinguishable from uniform, the privacy protection obtained is proven secure. This means that if an unproven computational complexity hypothesis is satisfied, no modern computer can find information about the plaintext from the ciphertext. It has also been demonstrated that a one-way function is needed to build a pseudorandom generator. Moreover, given any one-way function, one can build a pseudorandom generator. Unfortunately, the latter result is too theoretical to be used for building efficient pseudorandom generators.

Linear-feedback shift-register sequences are commonly used in software testing. However, these are too predictable to be useful in cryptography and do not satisfy the previous definition. Indeed, using linear algebra and having observed a sufficient number of outputs, one can compute the seed and predict the next outputs.

Many practical pseudorandom generators have been presented. Some of these have been based on nonlinear combinations of linear-feedback shift-registers others on recurrent lin-

ear congruences. Many of these systems have been broken. Using the output feedback (OFB) mode (17) of a block cipher one can also obtain pseudonoise generators. An example of a pseudonoise generator based on number theory is discussed later on.

Key Distribution

Public key systems, when combined with certificates, solve the key distribution problem. In many applications, however, replaying old but valid signatures should be impossible. Indeed, for example, one should not allow a recorded and replayed remote authenticated login to be accepted in the future. A solution to this problem is to require a fresh session key, used only for a particular session. Another reason to use session keys is that public key systems are slow, and so sender and receiver need to agree on a common secret key.

When conventional cryptography is used, the problem of key management is primary. Freshness remains important. The problem is how two parties who may have never communicated with each other can agree on a common secret key.

Many protocols have been presented. Designing secure ones is very tricky. Different security levels exist. A key distribution protocol based on number theory is discussed further on.

Zero-Knowledge

In many practical protocols one must continue using a key without endangering its security. Zero-knowledge (19) has been invented to prevent a secret(s) which has been used in a protocol by party (parties) A to leak to other parties B.

If B is untrusted, one gives the dark side of B the name B′. More scientifically, machines B adhere to their specified protocol. To specify parties that will interact with A, but behave differently, we need to speak about B′.

When untrusted parties (or a party), let us say specified by B′, are involved in a protocol, they see data being communicated to them and they also know the randomness they have used in this protocol. This data pulled together is called the view of B′. To this view corresponds a probability distribution (a random variable), because of the randomness used in the protocol. When both parties A and B have x as common input, this random variable is called $\text{View}_{A,B'}(x)$. If x is indeterminate, we have a family of such random variables, denoted $\{\text{View}_{A,B'}(x)\}$. One says that the protocol is zero-knowledge (does not leak anything about the secret of A) if one can simulate the view of B. This means that there is a computer (polynomial-time machine) without access to the secret that can generate strings with a distribution that is indistinguishable from $\{\text{View}_{A,B'}(x)\}$. One form of indistinguishability is called perfect, meaning that the two distributions are identical. There is also statistical and computational indistinguishability.

So, zero-knowledge says that whatever party B′ learned could be simulated off-line. So party B did not receive any information it can use after the protocol terminated. This is an important tool when designing proven secure protocols.

Commitment and Interactive Proofs

In many cryptographic settings, a prover A needs to prove to a verifier B that something has been done correctly, for example, demonstrate that a public key was chosen following the specifications. A straightforward, but unacceptable solution, would be to reveal the secret key used.

The solution to this problem is to use interaction (19). In many of these interactive protocols, the prover *commits* to something. The verifier asks a question [if the question is chosen randomly then the protocol is called an Arthur–Merlin game (20)]. Then the prover replies and may be asked to open the commitment. This may be repeated.

To be a (interactive) proof, it is necessary that the verifier will accept if the statement is true and the prover and verifier follow the described protocol. This property is called completeness. It is also necessary that the verifier will reject the proof if the statement is false, even if the prover behaves differently than specified and the dishonest prover A' has infinite computer power. This requirement is known as soundness. In a variant of interactive proofs, called arguments, the last condition has been relaxed.

An important subset of interactive proofs are the zero-knowledge ones. Then the view of a possibly dishonest verifier can be simulated, so the verifier does not learn any information that can be used off-line. Zero-knowledge interactive proofs have been used toward secure identification (entity authentication) protocols. An example of such a protocol is discussed later.

Note that several mechanisms for turning interactive zero-knowledge proofs into noninteractive ones have been studied both from a theoretical and practical viewpoint.

Cryptanalysis

Cryptanalysis uses its own tools. The classical tools include statistics and discrete mathematics.

Even if a cryptographic scheme is secure (that is, has not been broken), an inappropriate use of it may create a security breach. A mode or protocol may allow a cryptanalyst to find the plaintext, impersonate the sender, etc. Such problems are called “protocol failures.” An incorrect software implementation often enables a hacker to make an attack, and a poor hardware implementation may imply, for example, that the plaintext or the key leaks due to electromagnetic radiation or interference.

The most popular modern cryptanalytic tool against asymmetric cryptosystems, based on the geometry of numbers, is the Lenstra–Lenstra–Lovasz (LLL) lattice reduction algorithm (21). It has, for example, been used to break several knapsack public key systems and many protocols (22). When analyzing the security of block ciphers, the differential (23) and linear cryptanalytic (24) methods are very important. Specially developed algorithms to factor and compute discrete log have been developed, for example, the quadratic sieve method (25).

ALGORITHMS BASED ON NUMBER THEORY AND ALGEBRA

Although many of these algorithms are rather slow, they are becoming very popular. Attempts to break them have allowed scientists to find better lower bounds on the size of keys for which no algorithm exists and unlikely will be invented in the near future to break these cryptosystems. However, if a true quantum computer can be built, the security of many of these schemes is in jeopardy.

When writing $a \in_R S$, one means that a is chosen uniformly random in the set S.

We assume that the reader is familiar with basic knowledge of number theory and algebra.

RSA

RSA is a very popular public key algorithm invented by Rivest, Shamir, and Adleman (26).

To generate a public key, one chooses two random and different primes p and q which are large enough (512 bits at least). One computes their product $n := p \cdot q$. Then one chooses $e \in_R Z^*_{\phi(n)}$, where $\phi(n) = (p - 1)(q - 1)$, computes $d := e^{-1} \bmod \phi(n)$ and publishes (e, n) as a public key. The number d is the secret key. The numbers p, q, and $\phi(n)$ must also remain secret or be destroyed.

To encrypt a message $m \in Z_n$, one finds the authentic public key (e, n) of the receiver. The ciphertext is $c := m^e \bmod n$. To decrypt the ciphertext, the legitimate receiver computes $m' := c^d \bmod n$ using the secret key d. The Euler–Fermat theorem (and the Chinese Remainder theorem) guarantees that $m' = m$.

To sign with RSA, one processes the message M, hashes it with h to obtain m, computes $s := m^d \bmod n$, and sends (M, s), assuming that h has been agreed upon in advance. The receiver, who knows the correct public key (e, n) of the sender, can verify the digital signature. Given (M', s'), one computes m' from M', using the same preprocessing and hash function as in the signing operation, and accepts the digital signature if $m' = (s')^e \bmod n$. If this fails, the receiver rejects the message.

Many popular implementations use $e = 3$, which is not recommended at all for encryption. Other special choices for e are popular, but extreme care with such choices is called for. Indeed many signature and encryption schemes have suffered severe protocol failures.

Diffie–Hellman Key Distribution

Let $\langle g \rangle$ be a finite cyclic group of large enough order generated by g. We assume that q, a multiple of the order of the ord(g) (not necessarily a prime), is public.

The first party, let us say A, chooses $a \in_R Z_q$, computes $x := g^a$ in this group, and sends x to the party with which it wants to exchange a key, say B. Then B chooses $a \in_R Z_q$, computes $y := g^b$ in this group, and sends y to A. Now both parties can compute a common key. Indeed, A computes $z_1 := y^a$ in this group, and B computes $z_2 := x^b$ in this group. Now $z_2 = z_1$, as is easy to verify.

It is very important to observe that this scheme does not provide authenticity. A solution to this very important problem has been described in Ref. 27.

The cryptanalyst needs to compute $z = g^{\log_g(x) * \log_g(y)}$ in $\langle g \rangle$. This is believed to be difficult and is called the Diffie–Hellman search problem.

An example of a group which is considered suitable is a subgroup of Z^*_p, the Abelian group for the multiplication of elements modulo a prime p. Today it is necessary to have at least a 1024 bit value for p, and q should have a prime factor of at least 160 bits. Other groups being used include elliptic curve groups.

ElGamal Encryption

The ElGamal scheme (28) is a public key scheme. Let g and q be as in the Diffie–Hellman scheme. If g and q differ from user to user, then these should be extra parts of the public key.

To make a public key, one chooses $a \in_R Z_q$, computes $y := g^a$ in this group, and makes y public. To encrypt $m \in \langle g \rangle$, knowing the public key y_A, one chooses $k \in_R Z_q$, computes $(c_1, c_2) := (g^k, m \cdot y_A^k)$ in the group, and sends $c = (c_1, c_2)$. To decrypt, the legitimate receiver (using the secret key a) computes $m' := c_2 \cdot (c_1^a)^{-1}$ in this group.

The security of this scheme is related to the Diffie–Hellman problem.

ElGamal Signatures

The public and secret key are similar as in the ElGamal encryption scheme. The group used is Z^*_p, where p is a prime.

Let M be the message and m the hashed and processed version of M. To sign, the sender chooses $k \in_R Z^*_{p-1}$, computes $r := g^k \bmod p$, computes $s := (m - ar)k^{-1} \bmod (p - 1)$, and sends (M, r, s). To verify the signature, the receiver computes m from M and accepts the signature if $g^m = r^s \cdot y^r \bmod p$; otherwise rejects.

Several variants of this scheme have been proposed, for example, the US Digital Signature Standard (29).

Pseudonoise Generator

Several pseudorandom generators have been presented, but we discuss only one. In the Blum–Blum–Shub (30) generator, a large enough integer $n = pq$ is public, where p and q have secretly been chosen. One starts from a seed $s \in Z^*_n$ and sets $x := s$, and the first output bit b_0 of the pseudorandom generator is the parity bit of s. To compute the next output bit, compute $x := x^2 \bmod n$ and output the parity bit. More bits can be produced in a similar manner.

More efficient pseudorandom generators have been presented (31).

Shamir's Secret Sharing Scheme

Let t be the threshold, m be the secret, and l the number of shareholders.

In this scheme (12), one chooses $a_1, a_2, \ldots, a_{t-1} \in_R GF(q)$, and lets $f(0) = a_0 = m$, where $f(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \cdots + a_{t-1} \cdot x^{t-1}$ is a polynomial over $GF(q)$ and $q \geq l + 1$. The share $s_i = f(x_i)$ where $x_i \neq 0$ and the x_i are distinct. This corresponds to a Reed–Solomon code in which the message contains the secret and $(t - 1)$ uniformly chosen elements. Given t shares it is easy to compute $f(0)$, the secret, using Lagrange interpolation. One can easily prove that given $(t - 1)$ (or less shares), one has perfect secrecy, that is, any $(t - 1)$ shares are independent of the secret m.

GQ

Fiat and Shamir (32) suggested using zero-knowledge proofs to achieve identification. We discuss a variant of their scheme invented by Guillou and Quisquater (33).

Let $n = pq$, where p and q are distinct primes and v is a positive integer. To each prover one associates a number I, relatively prime to n which has a vth root. The prover, usually

called Alice, will prove that I has a vth root and will prove that she knows a vth root s such that $s^v I \equiv 1 \bmod n$. If she can prove this, then a receiver will conclude that the person in front must be Alice. One has to be careful with such a conclusion (34). The zero-knowledge interactive proof is as follows.

The verifier first checks whether I is relatively prime to n. The prover chooses $r \in_R Z_n^*$, computes $z := r^v \bmod n$, and sends z to the verifier. The verifier chooses $q \in_R Z_v$ and sends it to the prover. If $q \notin Z_v$, the prover halts. Else, the prover computes $y := rs^q \bmod n$ and sends y to the verifier. The verifier checks that $y \in Z_n^*$ and that $z = y^v I^q \bmod n$. If one of these tests fails, the protocol is halted.

This protocol must be repeated to guarantee soundness. Avoiding such repetitions is a practical concern, addressed in Ref. 35. If the protocol did not halt prematurely, the verifier accepts the prover's proof.

CONCLUSION

More encryption schemes and many more signature schemes exist than we were able to survey. The tools we discussed are used in a broad range of applications, such as electronic funds transfer (36), electronic commerce, threshold cryptography (37,38) (which allows companies to have public keys and reduce the potential of abuse by insiders), private e-mail. Cryptography has evolved from a marginally important area in electrical engineering and computer science to a crucial component.

READING LIST

Several books on practical cryptography have appeared in the last few years. The book by Menezes et al. (39) can be considered the best technical survey on the topic of applied cryptography printed so far. A more academic book, although not so exclusive, is Stinson's (40). Several interesting chapters, in particular the one on cryptanalysis (22) have appeared in the book edited by Simmons (41). Several chapters on cryptography will appear in Ref. 42.

Unfortunately, no good book on theoretical cryptography has appeared so far. Books which have appeared in this area are only readable by experts in the area, or their authors have only written about their own contributions.

Although outdated, the tutorial by Brassard (43) balances theory and practical aspects and is still worth reading. The book by Kahn (1) overviews historical cryptosystems. Although the new edition discusses modern cryptography, there are too few pages on the topic to justify buying the new edition if one has the old one. Other more general books are available (44,45).

The main conferences on the topic of cryptography are Eurocrypt and Crypto. Nowadays, there are many specialized or more local conferences, such as *Asiacrypt* (which absorbed *Auscrypt*), the *Workshop on Fast Software Encryption,* the *Workshop on Cryptographic Protocols,* the *ACM Conference on Computer and Communications Security,* and the *IMA Conference on Cryptography and Coding* in Britain. The proceedings of many of these conferences are published by Springer Verlag. Many conferences have sessions on cryptography, such as *IEEE-ISIT, IEEE-FOCS, ACM-STOC.* Articles that have appeared in journals are scattered. Unfortunately, some prestigious journals have accepted several articles of poor quality.

BIBLIOGRAPHY

1. D. Kahn, *The Codebreakers,* New York: Macmillan, 1967.
2. W. Diffie and M. E. Hellman, New directions in cryptography, *IEEE Trans. Inf. Theory,* **IT-22**: 644–654, 1976.
3. R. C. Merkle, Secure communications over insecure channels, *Commun. ACM,* **21**: 294–299, 1978.
4. J. Saltzer, On digital signatures, *ACM Oper. Syst. Rev.,* **12** (2): 12–14, 1978.
5. S. Haber and W. S. Stornetta, How to time-stamp a digital document, *J. Cryptol.,* **3** (2): 99–111, 1991.
6. G. J. Popek and C. S. Kline, Encryption and secure computer networks, *ACM Comput. Surv.,* **11** (4): 335–356, 1979.
7. C. H. Bennett and G. Brassard, An update on quantum cryptography, *Lect. Notes Comput. Sci.,* **196**: 475–480, 1985.
8. D. Atkins et al., The magic words are squeamish ossifrage, *Lect. Notes Comput. Sci.,* **917**: 263–277, 1995.
9. G. S. Vernam, Cipher printing telegraph systems for secret wire and radio telegraphic communications, *J. Amer. Inst. Electr. Eng.,* **45**: 109–115, 1926.
10. C. E. Shannon, Communication theory of secrecy systems, *Bell Syst. Tech. J.,* **28**: 656–715, 1949.
11. G. R. Blakley, Safeguarding cryptographic keys, *AFIPS Conf. Proc.,* **48**: 313–317, 1979.
12. A. Shamir, How to share a secret, *Commun. ACM,* **22**: 612–613, 1979.
13. G. R. Blakley, One-time pads are key safeguarding schemes, not cryptosystems, *Proc. IEEE Symp. Security Privacy,* CA, 1980, pp. 108–113.
14. M. Ito, A. Saito, and T. Nishizeki, Secret sharing schemes realizing general access structures, *Proc. IEEE Global Telecommun. Conf. (GLOBECOM '87),* 1987, pp. 99–102.
15. M. Naor and M. Yung, Universal one-way hash functions and their cryptographic applications, *Proc. 21st Annu. ACM Symp. Theory Comput. (STOC),* 1989, pp. 33–43.
16. J. Rompel, One-way functions are necessary and sufficient for secure signatures, *Proc. 22nd Annu. ACM Symp. Theory Comput. (STOC),* 1990, pp. 387–394.
17. National Bureau of Standards, *DES Modes of Operation,* FIPS Publ. No. 81 (Fed. Inf. Process. Stand.), Washington, DC: US Department of Commerce, 1980.
18. National Bureau of Standards, *Data Encryption Standard,* FIPS Publ. No. 46 (Fed. Inf. Process. Stand.), Washington, DC: US Department of Commerce, 1977.
19. S. Goldwasser, S. Micali, and C. Rackoff, The knowledge complexity of interactive proof systems, *SIAM J. Comput.,* **18** (1): 186–208, 1989.
20. L. Babai, Trading group theory for randomness, *Proc. 17th Annu. ACM Symp. Theory Comput. (STOC),* 1985, pp. 421–429.
21. A. K. Lenstra, Jr., H. W. Lenstra, and L. Lovasz, Factoring polynomials with rational coefficients, *Math. Ann.,* **261**: 515–534, 1982.
22. E. F. Brickell and A. M. Odlyzko, Cryptanalysis: A survey of recent results, in G. J. Simmons (ed.), *Contemporary Cryptology,* New York: IEEE Press, 1992, pp. 501–540.
23. E. Biham and A. Shamir, Differential cryptanalysis of DES-like cryptosystems, *J. Cryptol.,* **4** (1): 3–72, 1991.
24. M. Matsui, Linear cryptanalysis method for DES cipher, *Lect. Notes Comput. Sci.,* **765**: 386–397, 1994.

25. C. Pomerance, The quadratic sieve factoring algorithm, *Lect. Notes Comput. Sci.,* **209**: 169–182, 1985.
26. R. L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public key cryptosystems, *Commun. ACM,* **21**: 294–299, 1978.
27. P. C. van Oorschot, W. Diffie, and M. J. Wiener, Authentication and authenticated key exchanges, *Des., Codes Cryptogr.,* **2**: 107–125, 1992.
28. T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Trans. Inf. Theory,* **31**: 469–472, 1985.
29. National Institute of Standards and Technology, *Digital Signature Standard,* FIPS Publ. No. 186 (Fed. Inf. Process. Stand.), Springfield, VA: U.S. Department of Commerce, 1994.
30. L. Blum, M. Blum, and M. Shub, A simple unpredictable pseudo-random number generator, *SIAM J. Comput.,* **15** (2): 364–383, 1986.
31. A. W. Schrift and A. Shamir, The discrete log is very discreet, *Proc. 22nd Annu. ACM Symp. Theory Comput. (STOC),* 1990, pp. 405–415.
32. A. Fiat and A. Shamir, How to prove yourself: Practical solutions to identification and signature problems, *Lect. Notes Comput. Sci.,* **263**: 186–194, 1987.
33. L. C. Guillou and J.-J. Quisquater, A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory, *Lect. Notes Comput. Sci.,* **330**: 123–128, 1988.
34. S. Bengio et al., Secure implementations of identification systems, *J. Cryptol.,* **4** (3): 175–183, 1991.
35. M. V. D. Burmester, An almost-constant round interactive zero-knowledge proof, *Inf. Process. Lett.,* **42** (2): 81–87, 1992.
36. S. Brands, Electronic money, in M. Atallah (ed.), *Handbook of Algorithms and Theory of Computation,* Boca Raton, FL: CRC Press, in press, 1998.
37. Y. G. Desmedt, Threshold cryptography, *Eur. Trans. Telecommun.,* **5** (4): 449–457, 1994.
38. Y. Desmedt, Some recent research aspects of threshold cryptography, *Lect. Notes Comput. Sci.,* **1396**: 158–173, 1997.
39. A. Menezes, P. van Oorschot, and S. Vanstone, *Applied Cryptography,* Boca Raton, FL: CRC Press, 1996.
40. D. R. Stinson, *Cryptography: Theory and Practice,* Boca Raton, FL: CRC Press, 1995.
41. G. J. Simmons (ed.), *Contemporary Cryptology,* New York: IEEE Press, 1992.
42. M. Atallah (ed.), *Handbook of Algorithms and Theory of Computation,* Boca Raton, FL: CRC Press, in press, 1998.
43. G. Brassard, Modern Cryptology, *Lect. Notes Comput. Sci.,* Springer-Verlag, New York, 1988, p. 325.
44. B. Schneier, *Applied Cryptography. Protocols, Algorithms, and Source Code in C,* 2nd ed. New York: Wiley, 1996.
45. C. P. Schnorr, Efficient signature generation for smart cards, *J. Cryptol.,* **4** (3): 239–252, 1991.

YVO G. DESMEDT
University of
Wisconsin—Milwaukee
University of London

} { { } }

Data Compression Codes, Lossy
Standard Article

Ken Chu[1]
[1]Apple Computers Inc., Cupertino, CA

DOI: 10.1002/047134608X.W4203
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (190K)

Abstract

The sections in this article are

- Lossy Versus Lossless
- Why Lossy?
- Periodic Sampling
- Aliasing
- Quantization
- Vector Quantization
- Transform Coding
- Discrete Cosine Transform
- Subband Coding
- Predictive Coding
- Rate Distortion Theory

Acknowledgments

DATA COMPRESSION CODES, LOSSY

In this article we introduce lossy data compression. We consider the overall process of converting from analog data to digital so that the data are processed in digital form. Our goal is to achieve the most compression while retaining the highest possible fidelity. First we consider the requirements of signal sampling and quantization. Then we introduce several effective and popular lossy data compression techniques. At the end of this article we describe the theoretical limits of lossy data compression performance.

Lossy compression is a process of transforming data into a more compact form in order to reconstruct a close approximation to the original data. Let us start with a description using a classical information coding system model. A common and general data compression system is illustrated in Fig. 1.

As shown in Fig. 1, the information *source data*, S, is first transformed by the *compression process* to *compressed signal,* which usually is a more compact representation of the source data. The compact form of data offers tremendous advantages in both communication and storage applications. For example, in communication applications, the compressed signal is transmitted to a receiver through a communication channel with lower communication bandwidth. In storage applications, the compressed signal takes up less space. The stored data can be retrieved whenever they are needed. After received (or retrieved) signal is received (retrieved), it is pro-

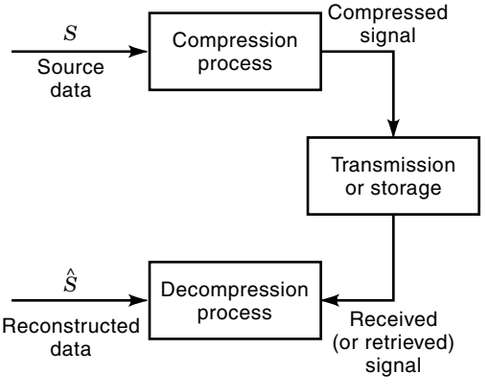


Figure 1. General data compression system.

cessed by the *decompression process,* which reconstructs the original data with the greatest possible fidelity. In lossy compression systems, the original signal, S, cannot be perfectly retrieved from the reconstructed signal, $\hat{S}$, which is only a close approximation.

LOSSY VERSUS LOSSLESS

In some applications, such as in compressing computer binary executables, database records, and spreadsheet or word processor files, the loss of even a single bit of data can be catastrophic. For such applications, we use *lossless data compression* techniques so that an exact duplicate of the input data is generated after the compress/decompress cycle. In other words, the reconstructed signal, $\hat{S}$, is identical to the original signal, S,

$$S = \hat{S}$$

Lossless data compression is also known as *noiseless data compression.* Naturally, it is always desirable to recreate perfectly the original signal after the transmission or storage process. Unfortunately, this requirement is difficult, costly, and sometimes infeasible for some applications. For example, for audio or visual applications, the original source data are analog data. The digital audio or video data we deal with are already an approximation of the original analog signal. After the compress/decompress cycle, there is no way to reconstruct an exact duplicate of the original continuous analog signal. The best we can do is to minimize the loss of fidelity during the compress/decompress process. In reality we do not need the requirement of $S=\hat{S}$ for audio and video compression other than for some medical or military applications. The *International Standards Organization* (ISO) has published the JPEG (Joint Photographic Experts Group) standard for still image compression (1) and the MPEG (Moving Pictures Expert Group) standard for moving picture audio and video compression (2, 3). Both JPEG and MPEG standards concern lossy compression, even though JPEG also has a lossless mode. The International Telecommunication Union (ITU) has published the H-series video compression standards, such as *H.261* (4) and *H.263* (5), and the G-series speech compression standards, such as *G.723* (6) and *G.728* (7). Both the H-series and G-series standards are also for lossy compression.

WHY LOSSY?

Lossy compression techniques involve some loss of source information, so data cannot be reconstructed in the original form after they are compressed by lossy compression techniques. However, we can generally get a much higher compression ratio and possibly a lower implementation complexity.

For many applications, a better compression ratio and a lower implementation complexity are more desirable than the ability to reconstruct perfectly the original data. For example, in audio-conferencing applications, it is not necessary to reconstruct perfectly the original speech samples at the receiving end. In general, telephone quality speech is expected at the receiver. By accepting a lower speech quality, we can achieve a much higher compression ratio with a moderated implementation complexity. Because of this, the conferencing speech signals can be transmitted to the destination through a lower bandwidth network at a reasonable cost. For music centric entertainment applications that require near CD-quality audio, the amount of information loss that can be tolerated is significantly lower. However, it is still not necessary to restrict compression to lossless techniques. The European MUSICAM and ISO MPEG digital audio standards both incorporate lossy compression yet produce high-fidelity audio. Similarly a perfect reconstruction of the original sequence is not necessary for most of the visual applications as long as the distortion does not result in annoying artifacts.

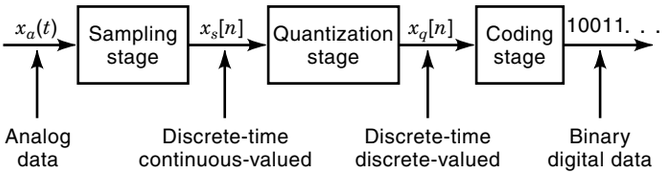


Figure 2. Analog-to-digital converter.

Most signals in our environment, such as speech, audio, video, radio, and sonar emissions, are analog signals. We have just discussed how lossy compression techniques are especially useful for compressing digital representations of analog data. Now let us discuss how to effectively convert an analog signal to digital data.

Theoretically converting an analog signal to the desired digital form is a three-stage process, as illustrated in Fig. 2. In the first stage, the analog data (continuous-time and continuous-valued) are converted to discrete-time and continuous-valued data by taking samples of the continuous-time signal at regular instants, $t=nT_1$,

$$x_s[n] = x_a(nT_1) \qquad \text{for } n = 0, \pm 1, \pm 2, \ldots$$

where T_1 is the *sampling interval.* In the *quantization stage,* the discrete-time continuous-valued signals are further converted to discrete-time discrete-valued signals by representing the value of each sample with one of a finite set of possible values. The difference between the unquantized sample $x_s[n]$ and the quantizer output $x_q[n]$ is called the *quantization error.* In reality quantization is a form of lossy data compression. Finally, in the *coding stage*, the quantized value, $x_q[n]$, is coded to a binary sequence, which is transmitted through the communication channel to the receiver. From a compression point of view, we need an analog-to-digital conversion system that generates the shortest possible binary sequence while still maintaining required fidelity. Let us discuss the signal sampling stage first.

PERIODIC SAMPLING

The typical method of converting a continuous-time signal to its discrete-time representation is through *periodic sampling,* with a sequence of samples,$x_s[n]$, obtained from the continuous-time signal $x_a(t)$ according to the following relationship

$$x_s[n] = x_a(nT_1) \qquad \text{for all integers } n$$

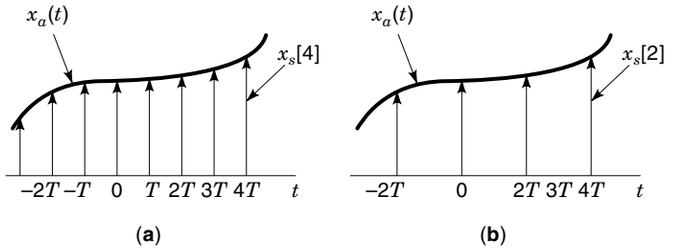


Figure 3. Continuous-time signal $x_a(t)$ sampled to discrete-time signals at the sampling period of (a) T, and (b) $2T$.

where n is an integer, T_1 is the *sampling period,* and its reciprocal $n_1=1/T_1$ is the *sampling frequency,* in samples per second. To visualize this process, consider embedding the samples in an idealized impulse train to form an idealized continuous time sampled waveform $x_s(t) = \sum_{n=-\infty}^{\infty} x_s[n]\delta(t-nT_1)$, where each impulse or Dirac δ function can be thought of as an infinitesimally narrow pulse of unit area at time $t = nT_1$ which is depicted as an arrow with height 1 corresponding to the area of the impulse. Then $x_s(t)$ can be drawn as a sequence of arrows of height $x_s[n]$ at time $t = nT_1$, as shown with the original signal $x_a(t)$ in Fig. 3 for sampling periods of T and $2T$.

The sampling process usually is not an invertible process. In other words, given a discrete-time sequence, $x_s[n]$, it is not always possible to reconstruct the original continuous-time input of the sampler, $x_a(t)$. It is very clear that the sampling process is not a one-to-one mapping function. There are many continuous-time signals that may produce the same discrete-time sequence output unless they have same bandwidth and sampled at Nyquist rate.

ALIASING

In order to get better understanding of the periodic sampler, let us look at it from frequency domain. First, consider the idealized sampling function, a periodic unit impulse train signal, $s(t)$:

$$s(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT_1)$$

where T_1 is the period of $s(t)$. The properties of impulse functions imply that the idealized sampled waveform is easily expressed as

$$\begin{aligned} x_s(t) &= x_a(t)s(t) \\ &= x_a(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_1) \\ &= \sum_{n=-\infty}^{\infty} x_a(nT_1)\delta(t - nT_1) \end{aligned} \tag{1}$$

To summarize, the idealized sampled data signal is defined as a product of the original signal and a samping function and is composed of a series of equally spaced impulses weighted by the values of the original continuous-time signal at the sampling instants, as depicted in Fig. 4.

Now let us make a Fourier analysis of $x_s(t)$. The Fourier transform pair (8) is defined as

$$x(t) = \int_{-\infty}^{+\infty} X(f)e^{j2\pi ft}\,df \tag{2}$$

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft}\,dt \tag{3}$$

where $X(f)$ is the *Fourier transform* of $x(t)$, or symbolically, $X(f) = \mathcal{F}(x(t))$, and $x(t)$ is the *inverse Fourier transform* of $X(f)$, $x(t) = \mathcal{F}^{-1}(X(f))$. A standard result of generalized Fourier analysis is that

$$s(t) = \frac{1}{T_1} \sum_{n=-\infty}^{+\infty} e^{j2n\pi f_1 t} \tag{4}$$

After substitution of Eq. (4) into Eq. (1), the sampled data, $\chi_{s(t)}$, yield

$$\begin{aligned} x_s(t) &= x_a(t)s(t) \\ &= \frac{1}{T_1} \sum_{n=-\infty}^{\infty} x_a(t)e^{j2n\pi f_1 t} \end{aligned} \tag{5}$$

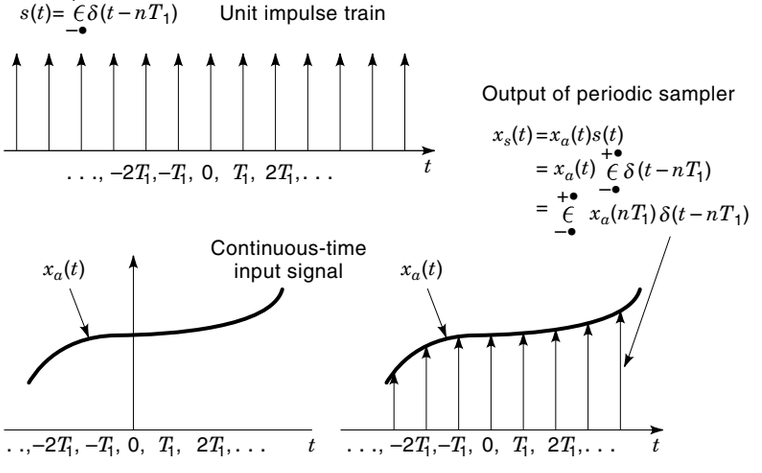


Figure 4. Periodic sampled continuous-time signal $x_a(t)$.

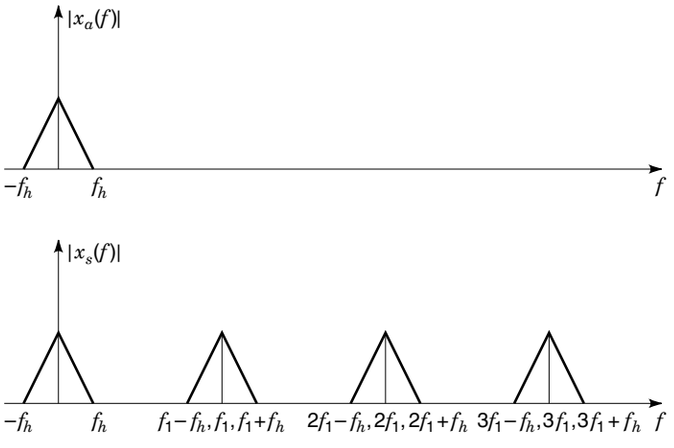


Figure 5. Spectrum of the sampled data sequence $x_s(t)$.

Now, taking the Fourier transform of $x_s(t)$ in Eq. (5), the result is

$$\begin{aligned} X_s(f) &= \int_{-\infty}^{+\infty} \left(\frac{1}{T_1} \sum_{n=-\infty}^{+\infty} x_a(t) e^{j2n\pi f_1 t} \right) e^{-j2\pi ft}\, dt \\ &= \frac{1}{T_1} \sum_{n=-\infty}^{+\infty} \int_{-\infty}^{+\infty} x_a(t) e^{-j2\pi (f-nf_1)t}\, dt \\ &= \frac{1}{T_1} \sum_{n=-\infty}^{+\infty} X_a(f - nf_1) \end{aligned} \tag{6}$$

We see from Eq. (6) that the spectrum of a sampled-data signal consists of the periodically repeated copies of the original signal spectrum. Each copy is shifted by integer multiples of the sampling frequency. The magnitudes are multiplied by $1T_1$.

Let us assume that the original continuous-time signal $x_a(t)$ is *bandlimited* to $0 \leq |f| \leq f_h$, then the spectrum of the sampled data sequence $x_s[n]$takes the form illustrated in Fig. 5. In the case where $f_h > f_1 - f_h$, or $f_1 < 2f_h$, there is an overlap between two adjacent copies of the spectrum as illustrated in Fig. 6. Now the overlapped portion of the spectrum is different from the original spectrum, and therefore it becomes impossible to recover the original spectrum. As a result the reconstructed output is distorted from the original continuous-time input signal. This type of the distortion is usually referred to as *aliasing*.

To avoid aliasing a bandlimited continuous-time input, it is necessary to sample the input at the sampling frequency $f_1 \geq 2f_h$. This is stated in the famous *Nyquist sampling theorem* (10).

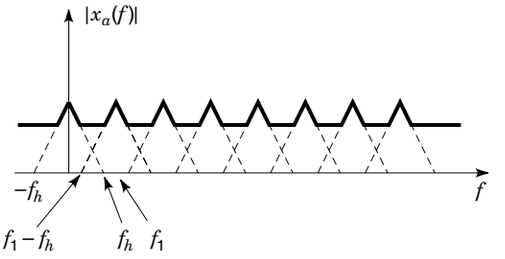


Figure 6. Spectrum of the sampled data sequence $x_s(t)$ for the case of $f_h > f_1 - f_h$.

Nyquist Sampling Theorem. If $x_a(t)$ is a bandlimited continuous-time signal with $X(f) = 0$ for $|f| > f_h$, then $x_a(t)$ can be uniquely reconstructed from the periodically sampled sequence $x_a(nT)$, $-\infty < n < \infty$, if $1/T > 2f_h$.

On the other hand, if the signal is not bandlimited, theoretically there is no avoiding the aliasing problem. All real-life continuous-time signals, such as audio, speech, or video emissions, are approximately bandlimited. A common practice is to get a close approximation of the original signals by filtering the continuous-time input signal with a low-pass filter before the sampling stage. This low-pass filter ensures that the filtered continuous-time signal meets the bandlimited criterion. With this *presampling filter* and a proper sampling rate, we can ensure that the spectral components of interest are within the bounds for which the signal can be recovered, as illustrated in Fig. 7.

QUANTIZATION

In the quantization stage discrete-time continuous-valued signals are converted to discrete-time discrete-valued signals. In the quantization process, amplitudes of the samples are quantized by dividing the entire amplitude range into a finite set of amplitude ranges. Each amplitude range has a representative amplitude value. The representative amplitude value for the range is assigned to all samples falling into the given range. Quantization is the most important step to removing the irrelevant information during lossy compression process. Therefore the performance of the quantizer plays a major role of overall performance of a lossy compression system.

There are many different types of quantizers. The simplest and most popular one is the *uniform quantizer,* in which the quantization levels and ranges are distributed uniformly. In general, a signal with amplitude x is specified by index k if x falls into the interval

$$I_k : \{x : x_k \leq x < x_{k+1}\}, \qquad k = 1, 2, 3, \ldots, L \tag{7}$$

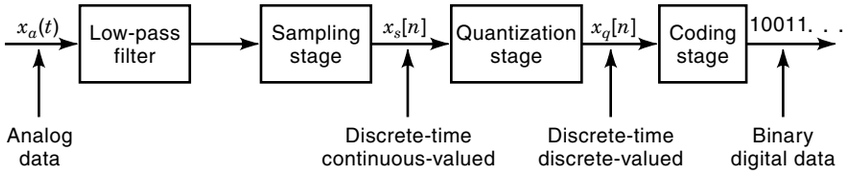


Figure 7. Sampling a continuous-time signal that is not bandlimited.

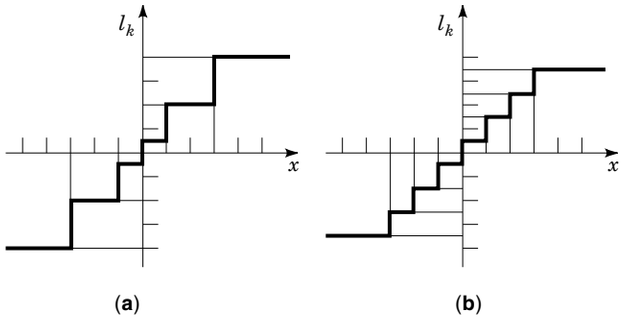


Figure 8. Examples of (a) a nonuniform quantizer, (b) an 8-level uniform quantizer.

In this process, the continuous valued signal with amplitude x is mapped into an *L-ary* index k. In most cases the *L-ary* index, k, is coded into binary numbers at the *coding stage* and transmitted to the receiver. Often, at the coding stage, efficient entropy coding is incorporated to generate variable length codewords in order to reach the entropy rate of quantized signals. Figure 8(a) and 8(b) gives examples of a nonuniform quantizer and an 8-level ($L = 8$) uniform quantizer.

At the receiver, the index k is translated into an amplitude I_k that represents all the amplitudes of signals fall into the interval I_k, namely

$$\hat{x}_k = l_k \qquad \text{if } x \in I_k \tag{8}$$

where $\hat{x}_k$ is the output of the decoder. The amplitude l_k is called the *representation level,* and the amplitude x_k is called the *decision level.* The difference between the input signal and the decoded signal, $x_k - x$, is called the *quantization error,* or *quantization noise.* Figure 9 gives an example of a quantized waveform and the corresponding quantization noise.

Quantization steps and ranges can be changed adaptively during the compression process. As an example, for video conferencing application, the compressed audio and video bit streams are transmitted through a network to the destination. Under the condition that the network is out of bandwidth, one cannot possibly transmit all the compressed data to the decoder in a timely manner. One easy solution is to increase the quantization step, such that quantizer generates a lower-quality output, and the bandwidth requirement is lower accordingly. This quantizer which changes adaptively is called an *adaptive quantizer.*

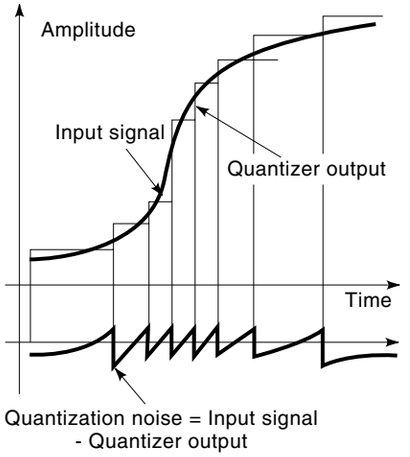


Figure 9. Quantization and quantization noise.

VECTOR QUANTIZATION

We have just introduced different ways of quantizing the output of a source. In all cases we discussed, the quantizer inputs were scalar values. In other words, the quantizer takes a single output sample of the source at a time and converts it to a quantized output. This type of quantizer is called a *scalar quantizer.*

Consider a case where we want to encode a consecutive sequence of samples from a stationary source. It is well-known from Shannon information theory that encoding a block of samples is more efficient than encoding each individual sample separately. In other words, during the quantization stage we wish to generate a representative index for a block of samples instead of for each separate sample. The basic concept is to generalize the idea from quantizing one sample at a time to quantizing a set of samples at a time. The set of the samples is called a *vector,* and this type of quantization process is called *vector quantization.*

Vector quantization is one of the most popular lossy data compression techniques. It is widely used in image, audio, and speech compression applications. The most popular vector quantization is *fixed-length vector quantization.* In the quantization process, consecutive input samples are grouped into fixed-length vectors first. As an example, we can group L samples of input speech as one L-dimensional vector, which forms the input vector to the vector quantizer. For a typical vector quantizer, both the encoder and the decoder share a common codebook, $\boldsymbol{C} = \{\boldsymbol{c}_i; i = 1, \ldots, N\}$, which can be predefined, fixed, or changed adaptively. Each entry of the codebook, $\boldsymbol{c}_i$, is called a *code-vector,* which is carefully selected as one of N representatives of the input vectors. Each code vector, $\boldsymbol{c}_i$, is also assigned an index, i. During the quantization stage the input vector, $\boldsymbol{x}$, is compared against each code-vector, $\boldsymbol{c}_i$, in the codebook. The "closest" code-vector, $\boldsymbol{c}_k$, is then selected as the representative code-vector for the input vector, and the corresponding index, k, is transmitted to the receiver. In other words, $\boldsymbol{c}_k$ is selected as the representative code-vector if

$$d(\boldsymbol{x}, \boldsymbol{c}_k) \le d(\boldsymbol{x}, \boldsymbol{c}_i) \qquad \text{for all } \boldsymbol{c}_i \in \boldsymbol{C} \tag{9}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_L)$ is the L-ary input vector and $\boldsymbol{C} = \{\boldsymbol{c}_i; i = 1, \ldots, N\}$ is the shared codebook, with ith code-vector, $\boldsymbol{c}_i$. The idea of vector quantization is identical to that of scalar quantization, except the distortion is measured on an L-dimensional vector basis. In Fig. 10 we show an example of a two-dimensional vector space quantized by a vector quantizer with $L = 2$, and $N = 16$. The code-vector $\boldsymbol{c}_k$ represents the input vector if it falls into the shaded vector space where Eq. (9) is satisfied. Since the receiver shares the same codebook with the encoder, and with received index, k, the decoder can easily retrieve the same representative code-vector, $\boldsymbol{c}_k$.

How do we measure the closeness, $d(\boldsymbol{x}, \boldsymbol{y})$, or distortion, between two L-ary vectors, $\boldsymbol{x}$ and $\boldsymbol{y}$, during the vector quantization process? The answer is dependent on the application. A *distortion measure* usually quantifies how well a vector quantizer can perform. It is also critical to the implementa-

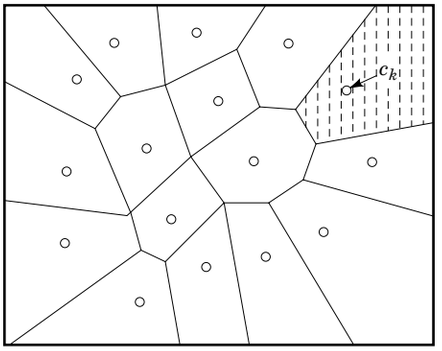


Figure 10. Two-dimensional vector space quantized by a vector quantizer.

tion of the vector quantizer, since measuring the distortion between two L-dimensional vectors is one of the most computationally intensive parts of the vector quantization algorithm. There are several ways of measuring the distortion. The most widely used distortion measure is the *mean square error* (MSE), which is defined as

$$d(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{L} \sum_{i=1}^{L} (x_i - y_i)^2$$

Another popular distortion measure is the *mean absolute difference* (MAD), or *mean absolute error* (MAE), and it is defined as

$$d(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{L} \sum_{i=1}^{L} |x_i - y_i|$$

There are various ways of generating the vector quantization codebook. Each method generates the codebook with different characteristics. The *LBG algorithm* (11) or the generalized Lloyd algorithm, computes a codebook with minimum average distortion for a given training set and a given codebook size. Tree-structured VQ (vector quantitization) imposes a tree structure on the codebook such that the search time is reduced (12,13,14). *Entropy-constrained vector quantization* (ECVQ) minimizes the distortion for a given average codeword length rather than a given codebook size (15). *Finite-state vector quantization* (FSVQ) can be modeled as a finite-state machine where each state represents a separate VQ codebook (16). *Mean / residual VQ* (M/RVQ) predicts the original image based on a limited data set, and then forms a residual by taking the difference between the prediction and the original image (17). Then the data used for prediction are coded with a scalar quantizer, and the residual is coded with a vector quantizer.

TRANSFORM CODING

We just considered the vector quantization, which effectively quantizes a block of data called a vector. Suppose that we have a reversible orthogonal transform, T, that transforms a block of data to a transform domain with the transform pair as

$$\boldsymbol{y} = T(\boldsymbol{x})$$
$$\boldsymbol{x} = T^{-1}(\boldsymbol{y})$$

where $\boldsymbol{x}$ is the original data block and T^{-1} is the inverse transform of T. In the transform domain we refer to the components of $\boldsymbol{y}$ as the transform coefficients. Suppose that the transform T has the characteristic that most of the transform coefficients are very small. Then the insignificant transform coefficients need not to be transmitted to decoder and can be eliminated during the quantization stage. As a result very good compression can be achieved with the transform coding approach. Figure 11 shows a typical lossy transform coding data compression system.

In Fig. 11 the input data block, $\boldsymbol{x}$, passes through the forward transform, T, with transform coefficients, $\boldsymbol{y}$, as its output. T has the characteristics that most of its output, $\boldsymbol{y}$, are small and insignificant and that there is little statistical correlation among the transform coefficients, which usually results in efficient compression by simple algorithms. The transform coefficients, $\boldsymbol{y}$, are quantized by the quantizer, Q. Small and insignificant coefficients have a zero quantized value; therefore only few nonzero coefficients need to be coded and transmitted to the decoder. For the best compression ratio, efficient entropy coding can be applied to the quantized coefficients at the coding stage. After receiving the signal from the network, the decoder decodes and inverse quantizes the received signal and reconstructs the transform coefficients, $\hat{\boldsymbol{y}}$. The reconstructed transform coefficients passes through the inverse transform, T^{-1}, which generates the reconstructed signal, $\hat{\boldsymbol{x}}$.

In general, transform coding takes advantage of the linear dependency of adjacent input samples. The linear transform actually converts the input samples to the transform domain for efficient quantization. In the quantization stage the transform coefficients can be quantized with a scalar quantizer or a vector quantizer. However, bit allocation among transform coefficients is crucial to the performance of the transform coding. A proper bit allocation at the quantization stage can achieve the output with a good fidelity as well as a good compression ratio.

There are quite a few transform coding techniques. Each has its characteristics and applications. The discrete Fourier transform (DFT) is popular and is commonly used for spectral analysis and filtering (18). Fast implementation of the DFT, also known as fast Fourier transform (FFT), reduces the transform operation to $n(n \log_2 n)$ for an n-point transform (19). The Karhunen–Loeve transform (KLT) is an optimal transform in the sense that its coefficients contain a larger fraction of the total energy compared to any other transform (20). There is no fast implementation of the KLT, however,

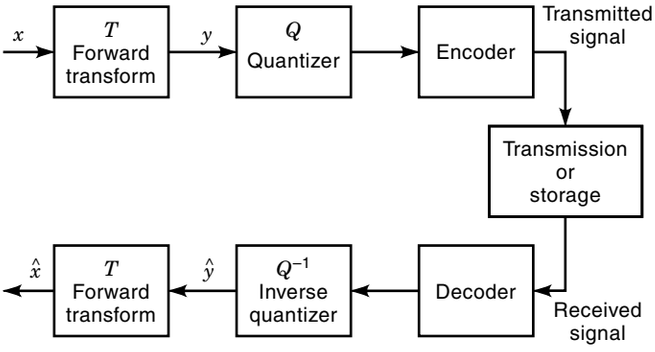


Figure 11. Basic transform coding system block diagram.

and its basis functions are target dependent. Because of this the KLT is not widely used. The Walsh–Hadamard transform (WHT) offers a modest decorrelating capability, but it has a very simple implementation (21). It is quite popular, especially for hardware implementation.

Transform coding plays a very important role in the recent lossy compression history. In the next section we will introduce the discrete cosine transform (DCT), which is the most popular transform for transform coding techniques.

DISCRETE COSINE TRANSFORM

The most important transform for transform coding is the discrete cosine transform (DCT) (22). The one-dimensional DCT F of a signal f is defined as follows (23,24):

$$F(k) = \sqrt{\frac{2}{N}}\, c(k) \sum_{j=0}^{N-1} f(j) \cos\left[\frac{(2j+1)k\pi}{2N}\right], \qquad k - 0,\ 1,\ 2,\ 3,\ \ldots,\ N-1$$

where $c(0) = 1/\sqrt{2}$ and $c(k) = 1$ for $k \neq 0$. The inverse DCT (IDCT) is given by

$$f(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} c(k) F(k) \cos\left[\frac{(2n+1)k\pi}{2N}\right], \qquad n = 0,\ 1,\ 2,\ 3,\ \ldots,\ N-1$$

A two-dimensional DCT for an image is formed by first taking the one-dimensional DCT of all rows of an image, and then taking the one-dimension DCT of all columns of the resulting image.

The DCT has fast implementations with a computational complexity of $O(n \log n)$ for an n-point transform. It has higher compression efficiency, since it avoids the generation of spurious spectral components. The DCT is the most widely used transform in transform coding for many reasons. It has superior energy compaction characteristics for most correlated source (25), especially for Markov sources with high correlation coefficient ρ,

$$\rho = \frac{E[\mathbf{x}_n \mathbf{x}_{n+1}]}{E[\mathbf{x}_n^2]}$$

where E denotes expectation. Since many sources can be modeled as Markov sources with a high correlation coefficient value, the superior energy compaction capability has made the DCT the most popular transform coding technique in the field of data compression. The DCT also tends to reduce the statistical correlation among coefficients. These properties make DCT-based lossy compression schemes very efficient. In addition the DCT can be implemented with reasonably low complexity. Because of this the DCT transform coding technique is widely used for both image and audio compression applications. The JPEG (1) and MPEG (2,3) published by ISO, and H.261 (4) and H.263 (5) published by ITU, are based on DCT transform coding compression techniques.

SUBBAND CODING

In the last section we introduced transform coding, which converts the input samples to the transform domain. Quantization and bit allocation are applied to the transform coefficients in the transform domain. One of the drawbacks of transform coding is that it has high computational complexity. Now we introduce another compression technique—*subband coding,* which usually has lower complexity than transform coding.

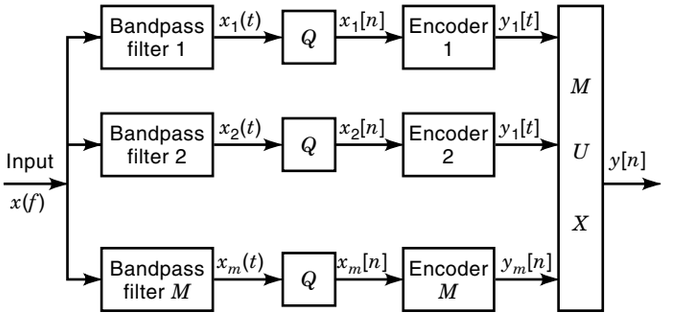


Figure 12. Block diagram of a typical subband coder.

Just like transform coding, subband coding uses a frequency domain approach. The block diagram of a typical subband encoder is illustrated in Fig. 12. The input signal, $x(t)$, is first filtered by a bank of M bandpass filters. Each bandpass filter produces a signal, $x_k(t)$, with limited ranges of spatial frequencies. Each filtered signal is followed by a quantizer and a bandpass encoder, which encodes the signal, $x_k(t)$, with different encoding techniques according to the properties of the subband. It may be encoded with different bit rates, quantization steps, entropy codings, or error distributions. The coding techniques we introduced in the previous sections, such as the vector quantization and entropy coding, are often used at the encoder. Finally the multiplexer combines all the subband coder output, $y_k[n]$, together and sends it through the communication channel to the decoder.

A subband decoder has the inverse stages of its encoder, as shown in Fig. 13. When a signal, $\hat{y}[n]$, is received from the communication channel, it goes through demultiplexing, decoding, and bandpass filtering prior to subband addition.

Subband coding has many advantages over other compression techniques. By controlling the bit allocations, quantization levels, and entropy coding separately for each subband, we can fully control the quality of the reconstructed signal. For this reason we can fully utilize the bandwidth of the communication channel. With an appropriate subband coding technique, we can achieve a good reconstruction signal quality, along with good compression. To take an example, for

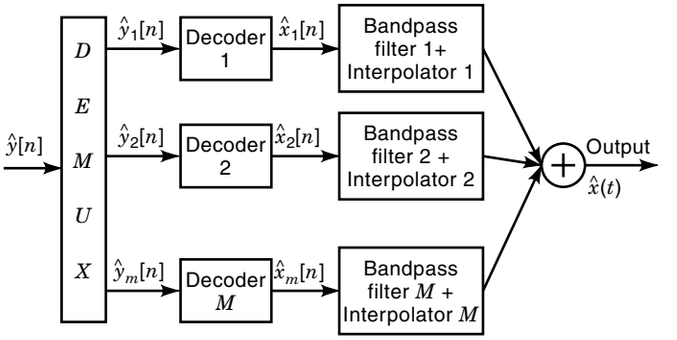


Figure 13. Subband decoder.

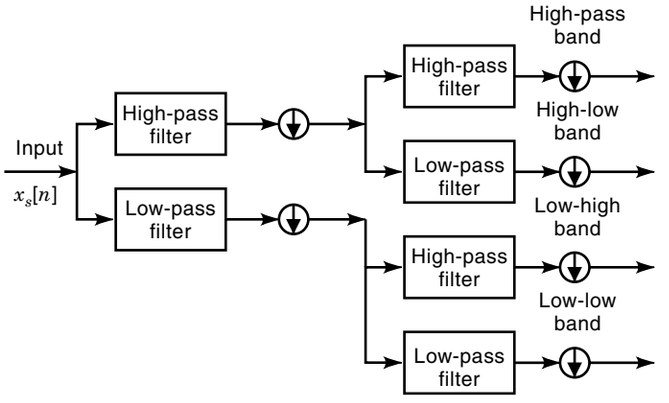


Figure 14. Four-band filter bank for uniform subband coding.

audio and speech applications low-frequency components are usually critical to the reconstructed sound quality. The subband coding technique enables the encoder to allocate more bits to lower subbands, and to quantize them with finer quantization steps. As a result the reconstructed data retains higher fidelity and higher signal-to-noise ratio (SNR).

A critical part of subband coding implementation is the filter bank. Each filter in the filter bank isolates certain frequency components from the original signal. Traditionally the most popular bandpass filter used in subband coding consisted of cascades of *low-pass filters* (LPFs) and *high-pass filters* (HPFs). A four-band filter bank for uniform subband coding is shown in Fig. 14. The filtering is usually accomplished digitally, so the original input is the sampled signal. The circled arrows denote down sampled by 2, since only half the samples from each filter are needed. The total number of samples remains the same. An alternative to a uniform subband decomposition is to decompose only the low-pass outputs, as in Fig. 15. Here the subbands are not uniform in size. A decomposition of this type is an example of a critically sampled *pyramid decomposition* or *wavelet decomposition* (26). Two-dimensional wavelet codes are becoming increasingly popular for image coding applications and include some of the best performing candidates for JPEG-2000.

Ideally the filter bank in the encoder would consist of a low-pass and a high-pass filter set with nonoverlapping, but contiguous, unit gain frequency responses. In reality the ideal filter is not realizable. Therefore, in order to convert the full spectrum, it is necessary to use filters with overlapping frequency response. As described earlier, the overlapping frequency response will cause aliasing. The problem is resolved by using exact reconstruction filters such as the *quadrature mirror filters* (QMF), as was suggested by Princey and Bradley (27), Croisier, Easteban, and Galand (28), Johnson (29), and Smith and Barnwell (30).

The idea of QMF is to allow the aliasing caused by overlapping filters in the encoder (analysis filter) canceled exactly by the filter banks in the decoder (synthesis filter). The filters are designed such that the overall amplitude and phase distortion is minimized. Then overall subband coding system with QMF filter bank is almost aliasing-free.

PREDICTIVE CODING

In this section we introduce another interesting compression technique—predictive coding. In the predictive coding systems, we assume a strong correlation between adjacent input data, which can be scalar, vector, or even block samples. There are many types of predictive coding systems. The most popular one is the linear predictive coding system based on the following linear relationship:

$$\hat{x}[k] = \sum_{i=0}^{k-1} \alpha_i x[i] \tag{10}$$

where the $x[i]$ are the input data, the α_i are the prediction coefficients, and $\hat{x}[k]$ is the predicted value of $x[k]$. The difference between the predicted value and the actual value, $e[k]$, is called the *prediction error:*

$$e[k] = x[k] - \hat{x}[k] \tag{11}$$

It is found that the prediction error usually has a much lower variance than the original signal, and is significantly less correlated. It has a stable histogram that can be approximated by a Laplacian distribution (31). With linear predictive coding, one can achieve a much higher SNR at a given bit rate. Equivalently, with linear predictive coding, one can reduce the bit rate for a given SNR. There are three basic components in the predictive coding encoder. They are predictor, quantizer, and coder, as illustrated in Fig. 16.

As shown in Fig. 16, the predicted signal, $\hat{x}[k]$, is subtracted from the input data, $x[k]$. The result of the subtraction is the prediction error, $e[k]$, according to Eq. (11). The prediction error is quantized, coded, and sent through communication channel to the decoder. In the mean time the predicted signal is added back to quantized prediction error, $e_q[k]$, to create reconstructed signal, $\tilde{x}$. Notice that the predictor makes the prediction according to Eq. (10), with previously reconstructed signal, $\tilde{x}$'s.

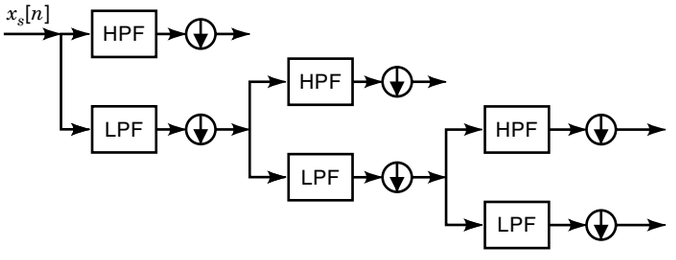


Figure 15. Filter bank for nonuniform subband coding.

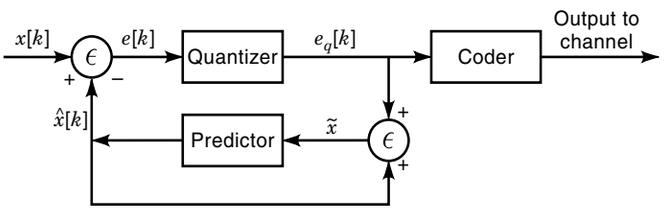


Figure 16. Block diagram of a predictive coder.

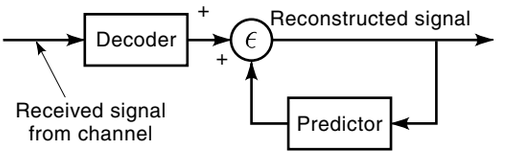


Figure 17. Predictive coding decoder.

Just like the encoder, the predictive coding decoder has a predictor, as shown in Fig. 17, which also operates in the same way as the one in the encoder. After receiving the prediction error from the encoder, the decoder decodes the received signal first. Then the predicted signal is added back to create the reconstructed signal. Even though linear prediction coding is the most popular predictive coding system, there are many variations. If the predictor coefficients remain fixed, then it is called *global prediction.* If the prediction coefficients change on each frame basis, then it is called *local prediction.* If they change adaptively, then it is called *adaptive prediction.* The main criterion of a good linear predictive coding is to have a set of prediction coefficients that minimize the mean-square prediction error.

Linear predictive coding is widely used in both audio and video compression applications. The most popular linear predictive codings are the *differential pulse code modulation* (DPCM) and the *adaptive differential pulse code modulation* (ADPCM).

RATE DISTORTION THEORY

In the previous sections we have briefly introduced several lossy data compression techniques. Each of them has some advantages for a specific environment. In order to achieve the best performance, one often combines several techniques. For example, in the MPEG-2 video compression, the encoder includes a predictive coder (motion estimation), a transform coder (DCT), an adaptive quantizer, and an entropy coder (run-length and Huffman coding). In this section we consider how well a lossy data compression can perform. In other words, we explore the theoretical performance trade-offs between fidelity and bit rate.

The limitation for lossless data compression is straightforward. By definition, the reconstructed data for lossless data compression must be identical to the original sequence. Therefore lossless data compression algorithms need to preserve all the information in the source data. From the lossless source coding theorem of Shannon information theory, we know that the bit rate can be made arbitrarily close to the entropy rate of the source that generated the data. So the entropy rate, defined as the entropy per source symbol, is the lower bound of size of the compressed data.

For lossy compression, distortion is allowed. Suppose that a single output X of a source is described by a probability density source function $f_x(x)$ and that X is quantized by a quantizer q into an approximate reproduction $\hat{x} = q(x)$. Suppose also that we have a measure of distortion $d(x, \hat{x}) \geq 0$ such as a square error $|x - \hat{x}|^2$ that measures how bad $\hat{x}$ is as a reproduction of x. Then the quality of the quantizer q can be quantized by the *average distortion*

$$D(q) = Ed(x, q(x)) = \int f_x(x) d(x, q(x))\, dx$$

The *rate* of the quantizer $R(q)$ has two useful definitions. If a fixed number of bits is sent to describe each quantizer level, then

$$R(q) = \log_2 M$$

where M is the number of possible quantizer outputs. On the other hand, if we are allowed to use a varying number of bits, then Shannon's lossless coding theorem says that

$$R(q) = H(q(x))$$

The entropy of the discrete quantizer output is the number of bits required on the average to recover $q(x)$. Variable length codes can provide a better trade-off of rate and distribution, since more bits can be used on more complicated data and fewer bits on low-complexity data such as silence or background. Whichever definition is used, we can define the *optimal performance* at a given bit rate by

$$\Delta(r) = \min_{q\,:\,R(q) \leq r} D(q)$$

By the *operational distortion-rate function,* or by the dual function,

$$R(d) = \min_{q\,:\,D(q) \leq d} R(q)$$

That is, a quantizer is *optimal* if it minimizes the distortion for a given rate, and vice versa. In a similar fashion we could define the optimal performance $\Delta_k(r)$ or $\mathbf{R}_k(d)$ using vector quantizers of dimension k as providing the optimal rate-distortion trade-off. Last we could ask for the optimal performance, say $\Delta_\infty(r)$ or $\mathbf{R}_\infty(d)$, when one is allowed to use quantizers of arbitrary length and complexity:

$$\Delta_\infty(r) = \min_k \Delta_k(r)$$
$$\mathbf{R}_\infty(d) = \min_k \mathrm{R}_\mathrm{k}(d)$$

where the Δ_k and $\mathbf{R}_k$ are normalized to distortion per sample (pixel) and bits per sample (pixel). Why study such optimizations? Because they give an unbeatable performance bound to all real codes and they provide a benchmark for comparison. If a real code is within 0.25 dB of $\Delta_\infty(r)$, it may not be worth any effort to further improve the code.

Unfortunately, Δ_∞ and $\mathbf{R}_\infty$ are not computable from these definitions, the required optimization is too complicated. Shannon rate-distortion theory (32) shows that in some cases Δ_∞ and $\mathbf{R}_\infty$ can be found. Shannon defined the (Shannon) rate-distortion function by replacing actual quantizers by random mappings. For example, a first-order rate-distortion function is defined by

$$R(d) = \min I(X, Y)$$

where the minimum is over all conditional probability density functions $f_{Y|X}(y|x)$ such that

$$Ed(X, Y) = \int \int f_{Y|X}(y|x) f_X(x) d(x, y)\, dxdy \leq d$$

The dual function, the Shannon distortion-rate function $D(r)$ is defined by minimizing the average distortion subject to a constraint on the mutual information. Shannon showed that for a memoryless source that

$$\mathbf{R}_\infty(d) = R(d)$$

That is, $R(d)$ provides an unbeatable performance bound over all possible code, and the bound can be approximately achieved using vector quantization of sufficiently large dimension.

For example, if the source is a memoryless Gaussian source with zero mean and variance σ^2, then

$$R(d) = \frac{1}{2}\log\frac{\sigma^2}{d}, \qquad 0 \le d \le \sigma^2$$

or equivalently,

$$D(r) = \sigma^2 e^{-2R}$$

which provides an optimal trade-off with which real systems can be compared. Shannon and others extended this approach to sources with memory and a variety of coding structures.

The Shannon bounds are always useful as lower bounds, but they are often over conservative because they reflect only in the limit of very large dimensions and hence very complicated codes. An alternative approach to the theory of lossy compression fixes the dimension of the quantizers but assumes that the rate is large and hence that the distortion is small. The theory was developed by Bennett (33) in 1948 and, as with Shannon rate-distortion theory, has been widely extended since. It is the source of the "6 dB per bit" rule of thumb for performance improvement of uniform quantizers with bit rate, as well as of the common practice (which is often misused) of modeling quantization error as white noise.

For example, the Bennett approximation for the optimal distortion using fixed rate scalar quantization on a Gaussian source is (34)

$$\delta(r) \cong \frac{1}{12}6\pi\sqrt{3}\sigma^2 2^{-2R}$$

which is strictly greater than the Shannon distortion-rate function, although the dependence of R is the same. Both the Shannon and Bennett theories have been extremely useful in the design and evaluation of lossy compression systems.

ACKNOWLEDGMENTS

The author wishes to thank Professor Robert M. Gray of Stanford University for providing valuable information and enlightening suggestions. The author also wishes to thank Allan Chu, Chi Chu, and Dr. James Normile for reviewing his manuscript.

BIBLIOGRAPHY

1. W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Data Compression Standard,* New York: Van Nostrand Reinhold, 1993.
2. J. L. Mitchell, et al., *MPEG Video Compression Standard,* London: Chapman & Hall, 1997.
3. B. B. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An Introduction to MPEG-2,* London: Chapman & Hall, 1997.
4. Recommendation H.261: *Video Codec for Audiovisual Services at* $p \times 64$ *kbits/s.* ITU-T (CCITT), March 1993.
5. Draft Recommendation H.263: *Video Coding for Low Bitrate Communication,* ITU-T (CCITT), December 1995.
6. Draft Recommendation G.723: *Dual Rate Speech Coder for Multimedia Communication Transmitting at 5.3 and 6.3 Kbits/s,* ITU-T (CCITT), October 1995.
7. Draft Recommendation G.728: *Coding of Speech at 16 Kbit/s Using Low-Delay Code Excited Linear Prediction (LD-CELP),* ITU-T (CCITT), September 1992.
8. R. N. Bracewell, *The Fourier Transform and Its Applications,* 2nd ed., New York: McGraw-Hill, 1978, pp.6–21.
9. R. N. Bracewell, *The Fourier Transform and Its Applications,* 2nd ed., New York: McGraw-Hill, 1978, pp. 204–215.
10. H. Nyquest, Certain topics in telegraph transmission theory, *Trans. AIEE,* **47**: 617–644, 1928.
11. Y. Linde, A. Buzo, and R. M. Gray, An algorithm for vector quantizer design, *IEEE Trans. Commun.,* **28**: 84–95, 1980.
12. R. M. Gray, Vector quantization, *IEEE Acoust. Speech Signal Process.,* **1** (2): 4–29, 1984.
13. J. Makhoul, S. Roucos, and H. Gish, Vector quantization in speech coding, *Proc. IEEE,* **73**: 1551–1588, 1985.
14. A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression,* Norwell, MA: Kluwer, 1992.
15. P. A. Chou, T. Lookabaugh, and R. M. Gray, Entropy-constrained vector quantization, *IEEE Trans. Acoust., Speech Signal Process.,* **37**: 31–42, 1989.
16. J. Foster, R. M. Gray, and M. O. Dunham, Finite-state vector quantization for waveform coding, *IEEE Trans. Inf. Theory,* **31**: 348–359, 1985.
17. R. L. Baker and R. M. Gray, Differential vector quantization of achromatic imagery, *Proc. Int. Picture Coding Symp.,* 1983, pp. 105–106.
18. W. K. Pratt, *Digital Image Processing,* New York: Wiley-Interscience, 1978.
19. E. O. Brigham, *The Fast Fourier Transform,* Englewood Cliffs, NJ: Prentice-Hall, 1974.
20. P. A. Wintz, Transform Picture Coding, *Proc. IEEE,* **60**: 809–820, 1972.
21. W. K. Pratt, *Digital Image Processing,* New York: Wiley-Interscience, 1978.
22. W. H. Chen and W. K. Pratt, Scene adaptive coder, *IEEE Trans. Commun.,* **32**: 224–232, 1984.
23. N. Ahmed, T. Natarajan, and K. R. Rao, Discrete cosine transform, *IEEE Trans. Comput.,* **C-23**: 90–93, 1974.
24. N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing,* New York: Springer-Verlag, 1975.
25. N. S. Jayant and P. Noll, *Digital Coding of Waveforms,* Englewood Cliffs, NJ: Prentice-Hall, 1984.
26. M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding,* Upper Saddle River, NJ: Prentice-Hall PTR, 1995.
27. J. Princey and A. Bradley, Analysis/synthesis filter bank design based on time domain aliasing cancellation, *IEEE Trans. Acoust. Speech Signal Process.,* **3**: 1153–1161, 1986.
28. A. Croisier, D. Esteban, and C. Galand, Perfect channel splitting by use of interpolation/decimation techniques, *Proc. Int. Conf. Inf. Sci. Syst.,* Piscataway, NJ: IEEE Press, 1976.
29. J. D. Johnson, A filter family designed for use in quadrature mirror filter banks, *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.,* Piscataway, NJ: IEEE Press, 1980, pp. 291–294.

30. M. J. T. Smith and T. P. Barnwell III, A procedure for designing exact reconstruction filter banks for tree structured subband coders, *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.,* Piscataway, NJ: IEEE Press, 1984.
31. A. Habibi, Comparison of *n*th-order DPCM encoder with linear transformation and block quantization techniques, *IEEE Trans. Commun. Technol.,* **19**: 948–956, 1971.
32. C. E. Shannon, Coding theorems for a discrete source with a fidelity criterion, *IRE Int. Convention Rec.,* pt. 4, **7**: 1959, 142–163.
33. A. Gersho, Asymptotically optimal block quantization, *IEEE Trans. Inf. Theory,* **25**: 373–380, 1979.
34. A. Gersho, *Principles of Quantization, IEEE Trans. Circuits Syst.,* **25**: 427–436, 1978.

Reading List

1. R. M. Gray and D. L. Neuhoff, Quantization, *IEEE Trans. Inf. Theory,* 1998.
2. M. Rabbani and P. W. Jones, *Digital Image Compression Techniques,* Tutorial Texts in Optical Engineering, vol. 7, Bellingham, WA: SPIE Optical Eng. Press, 1991.
3. J. L. Mitchell, et al., *MPEG Video Compression Standard,* London: Chapman & Hall, 1997.
4. B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An Introduction to MPEG-2,* London: Chapman & Hall, 1997.
5. W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Data Compression Standard,* New York: Van Nostrand Reinhold, 1993.
6. T. M. Cover and J. A. Thomas, *Elements of Information Theory,* New York: Wiley, 1991.

KEN CHU
Apple Computers Inc.

DATA CONVERTERS. See ANALOG-TO-DIGITAL CONVERSION.

DATA DETECTION. See DEMODULATORS.

} { { } }

Wiley Encyclopedia of Electrical and Electronics Engineering

Estimation Theory
Standard Article
Anatoli Juditsky[1]
[1]INRIA Rhone-Alpes, Montbonnot Saint Martin, France

DOI: 10.1002/047134608X.W4213
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (161K)

Abstract

The sections in this article are

Basic Concepts

Asymptotic Behavior of Estimators

Methods of Producing Estimators

Nonparametric Estimation

Model Selection

ESTIMATION THEORY

It is often the case in control and communication systems that the mathematical model describing a particular phenomenon is completely specified, except some unknown quantities. These quantities must be estimated. Identification, adaptive control, learning systems, and the like, provide examples. Exact answers are often difficult, expensive, or merely impossible to obtain. However, approximate answers that are likely to be close to the exact answers may be fairly easily obtainable. Estimation theory provides a general guide for obtaining such answers; above all, it makes mathematically precise such phrases as "likely to be close," "this estimator is optimal (better than others)," and so forth.

Though estimation theory originated from certain practical problems, only the mathematical aspects of the subject are considered below. The practical usefulness of the concepts used is not comprehensively discussed. One can refer to the treatises (3) and (9) for thorough motivations of these concepts from the application point of view.

What follows considers only the *statistical* framework, that is, it is supposed that the noisy environment, where observations are taken, is of a *stochastic* (random) nature. Situations when this assumption does not hold are addressed by *minimax estimation* methods.

Depending on how much prior information about the system to be identified is available, one may distinguish between two cases:

1. The system can be specified up to an unknown parameter of finite dimension. Then the problem is called the *parametric* estimation problem. For instance, such a problem arises when the parameters of a linear system of bounded dimension are to be estimated.
2. However, rather often, one has to infer relationships between input and output data of a system, when very little prior knowledge is available. In engineering practice, this problem is known as black-box modeling. Linear system of infinite dimension and general nonlinear systems, when the input/output relation cannot be defined in terms of a fixed number of parameters, provide examples. In estimation theory, these problems are referred to as those of *nonparametric* estimation.

Consider now some simple examples of mathematical statements of estimation problems.

Example 1. Let $X_1, \ldots, X_n$ be independent random variables (or observations) with a common unknown distribution $\mathscr{P}$ on the real line. One can consider several estimates (i.e., functions of the observations (X_i), $i = 1, \ldots, n$) of the mean $\theta = \int x d\mathscr{P}$:

1. The empirical mean
$$\tilde{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{1}$$
2. The empirical median $m = \text{median}\,(X_1, \ldots, X_n)$, which is constructed as follows: Let $Z_1, \ldots, Z_n$ be an increasing rearrangement of $X_1, \ldots, X_n$. Then $m = Z_{[(n+1)/2]}$ for n odd and $m = (Z_{n/2} + Z_{n/2+1})/2$ for n even (here $[x]$ stands for the integer part of x).
3. $g = (\max_{1\le i\le n}(X_i) + \min_{1\le i\le n}(X_i))/2$

Example 2. The (linear regression model). The variables y_i, X_i^k, $i = 1, \ldots, n$, $k = 1, \ldots, d$ are observed, where
$$y_i = \theta_1 X_i^1 + \cdots + \theta_d X_i^d + e_i$$
The e_i are random disturbances and $\theta_1, \ldots, \theta_d$ should be estimated. Let us denote $X_i = (X_i^1, \ldots, X_i^d)^T$, $\theta = (\theta_1, \ldots, \theta_d)^T$. The estimate
$$\hat{\theta}_n = \arg\min_{\theta} \sum_{i=1}^{n} (y_i - \theta^T X_i)^2$$
of θ is referred to as the *least squares estimate.*

Example 3. Let $f(x)$ be an unknown signal, observed at the points, $X_i = i/n$, $i = 1, \ldots, n$ with additive noise:

$$y_i = f(X_i) + e_i, \qquad i = 1, \ldots, n \tag{2}$$

This problem is referred to as nonparametric regression. Suppose that f is square-integrable and periodic on [0,1]. Then one can develop f into Fourier series

$$f(x) = \sum_{k=0}^{\infty} c_k \phi_k(x)$$

where, for instance, $\phi_0(x) = 0$, $\phi_{2l-1}(x) = \sqrt{2}\sin(2\pi lx)$, and $\phi_{2l}(x) = \sqrt{2}\cos(2\pi lx)$ for $l = 1, 2, \ldots$. Then one can compute the empirical coefficients

$$\hat{c}_k = \frac{1}{n} \sum_{k=1}^{n} y_i \phi_k(X_i) \tag{3}$$

and construct an estimate $\hat{f}_n$ of f by substituting the estimates of the coefficients in the Fourier sum of the length M:

$$\hat{f}_n(x) = \sum_{k=1}^{M} \hat{c}_k \phi_k(x) \tag{4}$$

Examples 1 and 2 above are simple parametric estimation problems. Example 3 is a nonparametric problem. Typically, one chooses the order M of the Fourier approximation as a function of total number of observations n. This way, the problem of function estimation can be seen as that of parametric estimation, though the number of parameters to be estimated is not bounded beforehand and can be large.

The basic ideas of estimation theory will now be illustrated, using parametric estimation examples. Later, it shall be seen how they can be applied in the nonparametric estimation.

BASIC CONCEPTS

Note the abstract statement of the estimation problem. It is assumed an observation of X is a random element, whose unknown distribution belongs to a given family of distributions P. The family can always be parametrized and written in the form $\{\mathscr{P}_\theta\colon \theta \in \Theta\}$. Here the form of dependence on the parameter and the set Θ are assumed to be known. The problem of estimation of an unknown parameter θ or of the value $g(\theta)$ of a function g at the point θ consists of constructing a function $\hat{\theta}(X)$ from the observations, which gives a sufficiently good approximation of θ (or of $g(\theta)$).

A commonly accepted approach to *comparing estimators,* resulting from A. Wald's contributions, is as follows: consider a quadratic *loss function* $q(\hat{\theta}(X) - \theta)$ (or, more generally, a nonnegative function $w(\hat{\theta}(X), \theta)$), and given two estimators $\hat{\theta}_1(X)$ and $\hat{\theta}_1(X)$, the estimator for which the expected loss (risk) $Eq(\hat{\theta}_i(X) - \theta)$, $i = 1, 2$ is the smallest is called the better, with respect to the quadratic loss function q (or to w).

Obviously, such a method of comparison is not without its defects. For instance, the estimator that is good for one value of the parameter θ may be completely useless for other values. The simplest example of this kind is given by the "estimator" $\hat{\theta}_0 \equiv \theta_0$, for some fixed θ_0 (independent of observations). Evidently, the estimator $\hat{\theta}^*$ possessing the property

$$Eq(\hat{\theta}^*(X) - \theta) \le Eq(\hat{\theta}(X) - \theta), \qquad \text{for any } \theta \in \Theta$$

for any estimate $\hat{\theta}$ may be considered as optimal. The trouble is that such estimators do not exist (indeed, any "reasonable" estimator cannot stand the comparison with the "fixed" estimator $\hat{\theta}_0$ at θ_0). Generally, in this method of comparing the quality of estimators, many estimators prove to be incomparable. Estimators can be compared by their behavior at "worst" points: an estimator $\hat{\theta}^*$ of θ is called *minimax estimator* relative to the quadratic loss function $q(\cdot)$ if

$$\sup_{\theta \in \Theta} Eq(\hat{\theta}^*(X) - \theta) = \inf_{\hat{\theta}} \sup_{\theta \in \Theta} Eq(\hat{\theta}(X) - \theta)$$

where the lower bound is taken over all estimators $\hat{\theta}$ of θ.

In the Bayesian formulation of the problem the unknown parameter is considered to represent values of the random variables with prior distribution Q on Θ. In this case, the best estimator $\hat{\theta}^*$ relative to the quadratic loss is defined by the relation

$$\begin{aligned} Eq(\hat{\theta}^*(X) - \theta) &= \int_{\Theta} Eq(\hat{\theta}^*(X) - \theta) Q(d\theta) \\ &= \inf_{\hat{\theta}} \int_{\Theta} Eq(\hat{\theta}(X) - \theta) Q(d\theta) \end{aligned}$$

and the lower bound is taken over all estimators $\hat{\theta}$.

As a rule, it is assumed that in parametric estimation problems the elements of the parametric family $\{\mathscr{P}_\theta\colon \theta \in \Theta\}$ possess the density $p(x, \theta)$. If the density is sufficiently smooth function of θ and the *Fisher information* matrix

$$I(\theta) = \int \frac{dp}{d\theta}(x, \theta) \left(\frac{dp}{d\theta}(x, \theta)\right)^T \frac{dx}{p(x, \theta)}$$

exists. In this case, the estimation problem is said to be *regular,* and the accuracy of estimation can be bounded from below by the *Cramér-Rao* inequality: if $\theta \in \mathbf{R}$, then for any estimator $\hat{\theta}$,

$$E|\hat{\theta} - \theta|^2 \ge \frac{\left[1 + \left(\frac{db}{d\theta}\right)(\theta)\right]^2}{I(\theta)} + b^2(\theta) \tag{5}$$

where $b(\theta) = E\hat{\theta} - \theta$ is the bias of the estimate $\hat{\theta}$. An analogous inequality holds in the case of multidimensional parameter θ. Note that if the estimate θ is *unbiased,* that is, $E\hat{\theta} = \theta$, then

$$E|\hat{\theta} - \theta|^2 \ge I^{-1}(\theta)$$

Moreover, the latter inequality typically holds asymptotically, even for biased estimators when $I(\theta) = I$ does not depend on θ. It can be easily verified that for independent observations $X_1, \ldots, X_n$ with common regular distribution $\mathscr{P}_\theta$, if $I(\theta)$ is the Fisher information on one observation, then the Fisher infor-

mation on the whole sample $I_n(\theta) = nI(\theta)$, and the Cramér–Rao inequality takes the form

$$E|\hat{\theta} - \theta|^2 \geq \frac{\left[1 + \left(\frac{db}{d\theta}\right)(\theta)\right]^2}{nI(\theta)} + b^2(\theta)$$

where $\hat{\theta} = \hat{\theta}(X_1, \ldots, X_n)$.

Return to Example 1. Let X_i be normal random variables with distribution density

$$\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\theta)^2}{2\sigma^2}\right)$$

If σ^2 is known, then the estimator $\bar{X}$ is an unbiased estimator of θ, and $E(\bar{X} - \theta)^2 = \sigma^2/n$. On the other hand, the Fisher information of the normal density $I(\theta) = \sigma^{-2}$. Thus $\bar{X}$ is in this situation the best unbiased estimator of θ.

If, in the same example, the distribution $\mathscr{P}$ possesses the Laplace density

$$\frac{1}{2a} \exp\left(-\frac{|x-\theta|}{a}\right)$$

then the Fisher information on one observation $I(\theta) = a^{-1}$. In this case $E(\bar{X} - \theta)^2 = 2a/n$. However, the median estimator m, as n grows to infinity, satisfies $nE(m - \theta)^2 \to a$. Therefore, one can suggest that m is an asymptotically better estimator of θ, in this case.

The error $\hat{\theta}_n - \theta$ of the least-squares estimator $\hat{\theta}$ in Example 2, given the observations $y_1, X_1, \ldots, y_n, X_n$, has the covariance matrix

$$E(\hat{\theta}_n - \theta)(\hat{\theta} - \theta)^T = \sigma^2 \left(\sum_{i=1}^{n} X_i X_i^T\right)^{-1}$$

This estimator is the best unbiased estimator of θ if the disturbances e_i obey normal distribution with zero mean and variance σ^2.

Note that, if the Fisher information $I(\theta)$ is infinite, the estimation with the better rate than $1/n$ is possible. For instance, if in Example 1 the distribution $\mathscr{P}$ is uniform over $[\theta - 1/2, \theta + 1/2]$, then the estimate g satisfies

$$E(g - \theta)^2 = \frac{1}{2(n+1)(n+2)}$$

ASYMPTOTIC BEHAVIOR OF ESTIMATORS

Accepting the stochastic model in estimation problems makes it possible to use the power of limit theorems (the law of large numbers, the central limit theorem, etc.) of probability theory, in order to study the properties of the estimation methods. However, these results holds *asymptotically,* that is, when certain parameters of the problem tend to limiting values (e.g., when the sample size increases indefinitely, the intensity of the noise approaches zero, etc.). On the other hand, the solution of nonasymptotic problems, although an important task in its own right, cannot be a subject of a sufficiently general mathematical theory: the correspondent solution depends heavily on the specific noise distribution, sample size, and so on. As a consequence, for a long time there have been attempts to develop a general procedure of constructing estimates which are not necessarily optimal for a given finite amount of data, but which approach optimality asymptotically (when the sample size increases or the signal-to-noise ratio goes to zero).

For the sake of being explicit, a problem such as in Example 2 is examined, in which $\Theta \in \mathbf{R}^d$. It is to be expected that, when $n \to \infty$, "good" estimators will get infinitely close to the parameter being estimated. Let P_θ denote the distribution of observations $y_1, X_1, \ldots$ for a fixed parameter θ. A sequence of estimators $\hat{\theta}_n$ is called *a consistent sequence of estimators* of θ, if $\hat{\theta}_n \to \theta$ in the probability P_θ for all $\theta \in \Theta$. Note that the estimators, proposed for Examples 1 and 2 above, are consistent.

Note that the notion of the minimax estimator can be refined when the asymptotic framework is concerned. An estimator $\hat{\theta}_n$, for which the quantity

$$\lim_{n\to\infty} \sup_{\theta\in\Theta} Eq(\hat{\theta}_n - \theta)$$

is minimized is referred to as the *asymptotically minimax* estimator in Θ, relative to the quadratic loss q. At first glance, this approach seems to be excessively "cautious": if the number n of observations n is large, a statistician can usually localize the value of parameter θ with sufficient reliability in a small interval around some θ_0. In such a situation, it would seem unnecessary to limit oneself to the estimators that "behave nicely" for values θ that are far away from θ_0. Thus one may consider locally asymptotic minimax estimators at a given point θ_0, that is, estimators that become arbitrarily close to the asymptotically minimax estimators in a small neighborhood of θ_0. However, it is fortunate that, in all interesting cases, asymptotically minimax estimators in Θ are also asymptotically minimax in any nonempty open subset of Θ. Detailed study of asymptotic properties of statistical estimators is a subject of *asymptotic theory* of estimation. Refer to (7) and (10) for exact statements and thorough treatment of correspondent problems.

METHODS OF PRODUCING ESTIMATORS

Let $p(X, \theta)$ stand for the density of the observation measure $\mathscr{P}_\theta$. The most widely used maximum-likelihood method recommends that the estimator $\hat{\theta}(X)$ be defined as the maximum point of the random function $p(X, \theta)$. Then $\hat{\theta}(X)$ is called the *maximum-likelihood estimator.* When the parameter set $\Theta \leq \mathbf{R}^d$, the maximum-likelihood estimators are to be found among the roots of the *likelihood equation*

$$\frac{d}{d\theta} \ln p(X, \theta) = 0$$

if these roots are inner points of Θ and $p(X, \theta)$ is continuously differentiable. In Example 1, $\bar{X}$ in (1) is the maximum-likelihood estimator if the distribution $\mathscr{P}$ is Gaussian. In Example 2, if the disturbances e_i have Laplace density, the maximum-likelihood estimator m_n satisfies

$$m_n = \arg \min_m \sum_{i=1}^{n} |y_i - m^T X_i|$$

Another approach consists to suppose that the parameter θ obeys a prior distribution Q on Θ. Then one can take a *Bayesian estimator* $\hat{\theta}$ relative to Q, although the initial formulation is not Bayesian. For example, if $\Theta = \mathbf{R}^d$, it is possible to estimate θ by means

$$\frac{\int_{\mathbf{R}^d} \theta p(X, \theta)\, d\theta}{p(X, \theta)\, d\theta}$$

This is a Bayesian estimator relative to the uniform prior distribution.

The basic merit of maximum-likelihood and Bayesian estimators is that, given certain general conditions, they are consistent, asymptotically efficient, and asymptotically normally distributed. The latter means that is $\hat{\theta}$ is an estimator, then the normalized error $I(\theta)^{1/2}(\hat{\theta} - \theta)$ converges in distribution to a Gaussian random variable with zero mean, and the identity covariance matrix.

The advantages of the maximum-likelihood estimators justify the amount of computation involved in the search for the maximum of the *likelihood function* $p(X, \theta)$. However, this can be a hard task. In some situations, the *least-squares method* can be used instead. In Example 1, it recommends that the minimum point of the function

$$\sum_{i=1}^{n} (X_i - \theta)^2$$

be used as the estimator. In this case, $\bar{X}$ in Eq. (1) is the least-squares estimate. In Example 2, the least squares estimator $\hat{\theta}_n$ coincides with the maximum-likelihood solution if the noises e_i are normally distributed.

Often, the exact form of density $p(X, \theta)$ of observations is unknown. However, the information that $p(X, \theta)$ belongs to some convex class P is available. The *robust approach* estimation recommends to find the density $p^*(X, \theta)$, which maximizes the risk of the least-squares estimate on P, and then to take

$$\hat{\theta}^*(X) = \arg \min_{\theta} p^*(X, \theta)$$

as the estimator. The $\hat{\theta}^*$ is referred to as the *robust estimate.* Suppose, for instance, that in Example 1 the distribution $\mathscr{P}$ satisfies $\int (x - \theta)\mathscr{P}(dx) \leq \sigma^2$. Then the empirical mean $\bar{X}$ is the robust estimate. If $p(x - \theta)$ is the density of $\mathscr{P}$, and it is known that $p(\cdot)$ is unimodal and for some $a > 0$ $p(0) \geq (2a)^{-1}$, then the median m is the robust estimator of θ [for more details, refer to (5)].

NONPARAMETRIC ESTIMATION

Consider the problem of nonparametric estimation. To be concrete, consider Eq. (2) in Example 3 above. There are two factors that limit the accuracy with which the signal f can be recovered. First, only a finite number of observation points $(X_i)_{i=1}^n$ are available. This suggests that $f(x)$, at other points x than those which are observed, must be obtained from the observed points by interpolation or extrapolation. Second, as in the case of parametric estimation, at the points of observation, X_i, $i = 1, \ldots, n$, $f(X_i)$ is observed with an additive noise $e_i = y_i - f(X_i)$. Clearly, the observation noises e_i introduce a random component in the estimation error. A general approach to the problem is the following: one first chooses an approximation method, that is, substitutes the function in question by its approximation. For instance, in Example 3, the approximation with a Fourier sum is chosen (it is often referred to as the projection approximation, since the function f is approximated by its projection on a final-dimensional subspace, generated by M first functions in the Fourier basis). Then one estimates the parameters involved in this approximation. This way the problem of function estimation is reduced to that of parametric estimation, though the number of parameters to be estimated is not fixed beforehand and can be large. To limit the number of parameters some *smoothness* or *regularity* assumptions have to be stated concerning f. Generally speaking, smoothness conditions require that the unknown function f belongs to a restricted class, such that, given an approximation technique, any function from the class can be "well" approximated, using a limited number of parameters. The choice of the approximation method is crucial for the quality of estimation and heavily depends on the prior information available about the unknown function f [refer to Ref. (8) for a more extensive discussion]. Now see how the basic ideas of estimation theory can be applied to the nonparametric estimation problems.

Performance Measures for Nonparametric Estimators

The following specific issues are important:

1. What plays the role of Cramér-Rao bound and Fisher Information Matrix in this case? Recall that the Cramér-Rao bound [Eq. (5)] reveals the best performance one can expect in identifying the unknown parameter θ from sample data arising from some parameterized distribution P_θ, $\theta \in \Theta$, where Θ is the domain over which the unknown parameter θ ranges. In the nonparametric (as well as in the parametric) case, lower bounds for the best achievable performance are provided by *minimax risk functons.* These lower bounds will be introduced and associated notions of optimality will be discussed.
2. For parametric estimation problems, a quadratic loss function is typical to work with. In functional estimation, however, the choice is much wider. One can be interested in the behavior of the estimate at one particular point x_0, or in the global behavior of the estimate. Different distance measures should be used in these two different cases.

In order to compare different nonparametric estimators, it is necessary to introduce suitable figures of merit. It seems first reasonable to build on the mean square deviation (or mean absolute deviation) of some semi-norm of the error, it is denoted by $\|\hat{f}_N - f\|$. A semi-norm is a norm, except it does not satisfy the condition: $\|f\| = 0$ implies $f = 0$. The following semi-norms are commonly used: $\|f\| = (\int f^2(x)dx)^{1/2}$, ($L_2$-norm), $\|f\| = \sup_x |f(x)|$ (uniform norm, C- or L_∞-norm), $\|f\| = |f(x_0)|$ (absolute value at a fixed point x_0). Then consider the *risk function*

$$R_{a_N}(\hat{f}_N, f) = E[a_N^{-1} \|\hat{f}_N - f\|]^2 \tag{6}$$

where a_N is a normalizing positive sequence. Letting a_N decrease as fast as possible so that the risk still remains bounded yields a notion of a convergence rate. Let $\mathcal{F}$ be a set of functions that contains the "true" regression function f, then the maximal risk $r_{a_N}(\hat{f}_N)$ of estimator $\hat{f}_N$ on $\mathcal{F}$ is defined as follows:

$$r_{a_N}(\hat{f}_N) = \sup_{f \in \mathcal{F}} R_{a_N}(\hat{f}_N, f)$$

If the maximal risk is used as a figure of merit, the optimal estimator $\hat{f}^*_N$ is the one for which the maximal risk is minimized, that is, such that

$$r_{a_N}(\hat{f}^*_N) = \min_{\hat{f}_N} \sup_{f \in \mathcal{F}} R_{a_N}(\hat{f}_N, f)$$

$\hat{f}^*_N$ is called *the minimax estimator* and the value

$$\min_{\hat{f}_N} \sup_{f \in \mathcal{F}} R_{a_N}(\hat{f}_N, f)$$

is called *the minimax risk* on $\mathcal{F}$. Notice that this concept is consistent with the minimax concept used in the parametric case.

The construction of minimax nonparametric regression estimators for different sets $\mathcal{F}$ is a difficult problem. However, letting a_N decrease as fast as possible so that the minimax risk still remains bounded yields a notion of a best achievable convergence rate, similar to that of parametric estimation. More precisely, one may state the following definition:

1. The positive sequence a_N is a lower rate of convergence for the set $\mathcal{F}$ in the semi-norm $\|\cdot\|$ if

$$\liminf_{N \to \infty} r_{a_N}(\hat{f}^*_N) = \liminf_{N \to \infty} \inf_{\hat{f}_N} \sup_{f \in \mathcal{F}} E[a_N^{-1} \|\hat{f}_N - f\|]^2 \geq C_0 \tag{7}$$

   for some positive C_0.

2. The positive sequence a_N is called minimax rate of convergence for the set $\mathcal{F}$ in semi-norm $\|\cdot\|$, if it is a lower rate of convergence, and if, in addition, there exists an estimator $\hat{f}^*_N$ achieving this rate, that is, such that

$$\limsup_{N \to \infty} r_{a_N}(\hat{f}^*_N) < \infty$$

The inequality [Eq. (7)] is a kind of negative statement that says that no estimator of function f can converge to f faster than a_N. Thus, a coarser, but easier approach consists in assessing the estimators by their convergence rates. In this setting, by definition, optimal estimators reach the lower bound as defined in Eq. (7) (recall that the minimax rate is not unique: it is defined to within a constant).

It holds that the larger the class of functions, the slower the convergence rate. Generally, it can be shown that no "good" estimator can be constructed on too rich functional class which is "too rich" [refer to (4)]. Note, however, that convergence can sometimes be proved without any smoothness assumption, though the convergence can be arbitrary slow, depending on the unknown function f to be estimated.

Consider Example 3. The following result can be acknowledged; refer to (7): Consider the Sobolev class $W^s(L)$ on [0, 1], which is the family of periodic functions $f(x)$, $x \in [0, 1]$, such that

$$\sum_{j=0}^{\infty} (1 + j^{2s}) |c_j|^2 \leq L^2 \tag{8}$$

(here c_j are the Fourier coefficients of f). If

$$\|g\| = \left(\int |g(x)|^2 \, dx \right)^{1/2}, \qquad \text{or} \quad \|g\| = |g(x_0)|$$

then $n^{-s/2s+d}$ is a lower rate of convergence for the class $W^s(L)$ in the semi-norm $\|\cdot\|$.

On the other hand, one can construct an estimate $\hat{f}_n$ [refer to (2)], such that uniformly, over $f \in W^s(L)$,

$$E\|\hat{f}_n - f\|_2^2 \leq O(L, \sigma) n^{-2s/(2s+1)} \tag{9}$$

Note that the condition [Eq. (8)] on f means that the function can be "well" approximated by a finite Fourier sum. Indeed, due to the Parseval equality, Eq. (8) implies that if

$$\overline{f}(x) = \sum_{j=1}^{M} c_j \phi_j(x)$$

then $\|\overline{f} - f\|_2^2 = O(M^{-2s})$. The upper bound, Eq. (9), appears rather naturally if one considers the following argument: If one approximates the coefficients c_j by their empirical estimates $\hat{c}_j$ in Eq. (3), the quadratic error in each j is $O(n^{-1})$. Thus, if the sum, Eq. (4) of M terms of the Fourier series is used to approximate f, the "total" stochastic error is order of M/n. The balance between the approximation (the bias) and the stochastic error gives the best choice $M = O(n^{1/(2s+1)})$ and the quadratic error $O(n^{-2s/(2s+1)})$. This simple argument can be used to analyze other nonparametric estimates.

MODEL SELECTION

So far the concern has been with estimation problems when the model structure has been fixed. In the case of parametric estimation, this corresponds to the fixed (a priori known) model order; in functional estimation this corresponds to the known functional class $\mathcal{F}$, which defines the exact approximation order. However, rather often, this knowledge is not accessible beforehand. This implies that one should be able to provide methods to retrieve this information from the data, in order to make estimation algorithms "implementable." One should distinguish between two statements of the model (order) selection problem: the first one arises typically in the *parametric* setting, when one suppose that the exact structure of the model is known up to unknown dimension of the parameter vector; the second one is essentially *nonparametric,* when it is assumed that the true model is of infinite dimension, and the order of a finite-dimensional approximation is to be chosen to minimize a prediction error (refer to the choice of the approximation order M in Eq. (4) of Example 3). These two approaches are illustrated in a simple example.

Example 4. Consider the following problem:

1. Let $\theta = (\theta_0, \ldots \theta_{d-1})^T$ be coefficients of a digital filter of unknown order d, that is,

$$y_i = \sum_{k=0}^{d-1} \theta_i x_{i-k+1} + e_i$$

   We assume that x_i are random variables. The problem is to retrieve θ from the noisy observations (y_i, x_i), $i = 1, \ldots, n$. If one denotes $X_i = (x_i, \ldots, x_{i-d+1})^T$, then the estimation problem can be reformulated as that of the linear regression in Example 2. If the exact order d was known, then the least-squares estimate $\hat{\theta}_n$ could be used to recover θ from the data. If d is unknown, it should be estimated from the data.
2. A different problem arises when the true filter is of infinite order. However, all the components of the vector θ of infinite dimension cannot be estimated. In this case one can approximate the parameter θ of infinite dimension by an estimate $\hat{\theta}_{d,n}$ which has only finite number d of nonvanishing entries:

$$\hat{\theta}_n = (\hat{\theta}_n^{(1)} \ldots, \hat{\theta}_n^{(d)}, 0, 0 \ldots)^T$$

   Then the "estimate order" d can be seen as a nuisance parameter to be chosen, in order to minimize, for instance, the mean prediction error $E[(\hat{\theta}_{d,n} - \theta)^T X_n]^2$.

Suppose that e_i are independent and Gaussian random variables. Denote $S_{d,n}^2 = n^{-1}\sum_{i=1}^n (y_i - \hat{\theta}_{d,n}^T X_i)^2$. If d is unknown, one cannot minimize $S_{d,n}^2$ with respect to d directly: the result of such a brute-force procedure would give an estimate $\hat{\theta}_{d,n}(x)$, which perfectly fits the noisy data (this is known as "overfitting" in the neural network literature). The reason is that $S_{d,n}^2$ is a biased estimate of $E(y_i - \hat{\theta}_{d,n}^T X_i)^2$. The solution rather consists to modify $S^2(d, n)$ to obtain an unbiased estimate of the prediction error. This can be achieved by introducing a penalty which is proportional to the model order d:

$$AIC(d, n) = \left(S_{d,n}^2 + \frac{2\sigma_e^2 d}{n} \right)$$

which is an unbiased (up to the terms of the higher order) estimate of the error up to terms that do not depend on d, $AIC(d, n)$. One can look for d_n such that

$$d_n = \arg \min_{d<n} AIC(d, n)$$

This technique leads to the celebrated Mallows–Akaike criterion (1, 11):

Unfortunately, d_n is not a consistent estimate of d. Thus it does not give a solution to the first problem of Example 4 above. On the other hand, it is shown in (6) that minimization over d of the criterion

$$HIC(d, n) = \left(S_{d,n}^2 + \frac{2\sigma_e^2 \lambda(n) d}{n} \right)$$

where

$$\liminf_n \frac{\lambda(n)}{\log \log n} > 1 \qquad \text{and} \qquad \frac{\lambda(n)}{n} \to 0$$

gives a consistent estimate of the true dimension d in the problem 1 of Example 4.

Another approach is proposed in (12) and (14). It consists to minimize, with respect to d, the total length of the incoding of the sequence y_i, X_i (MML—minimum message length, or MDL—minimum description length). This code length should also take into account the incoding of $\hat{\theta}_{d,n}$. This leads to the criterion (the first-order approximation)

$$d_n = \arg \min_{d \le n} BIC(d, n)$$

where

$$BIC(d, n) = \left(S_{d,n}^2 + \frac{2\sigma_e^2 d \log(n)}{n} \right)$$

As was shown in (13), the Bayesian approach (MAP—maximum a posteriori probability) leads to the minimization of $BIC(d, n)$, independently of the distribution of the parameter d.

BIBLIOGRAPHY

1. H. Akaike, Statistical predictor identification, *Ann. Inst. Math. Statist.,* **22**: 203–217, 1970.
2. N. Cencov, Statistical decision rules and optimal inference, *Amer. Math. Soc. Transl.,* **53**: 1982.
3. H. Cramer, *Mathematical Methods of Statistics,* Princeton, NJ: Princeton University Press, 1946.
4. L. Devroye and L. Györfi, *Nonparametric Density Estimation L_1 View,* New York: Wiley, 1985.
5. P. Huber, *Robust Statistics,* New York: Wiley, 1981.
6. E. J. Hannan, Estimation of the order of an ARMA process, *Ann. Stat.,* **8**: 339–364, 1980.
7. I. A. Ibragimov and R. Z. Khas'minskii, *Statistical Estimation: Asymptotic Theory,* New York: Springer, 1981.
8. A. Juditsky et al., Nonlinear black-box modelling in system identification: Mathematical foundations, *Automatica,* **31** (12): 1725–1750, 1995.
9. M. G. Kendall and A. Stuart, *The Advanced Theory of Statistics,* Griffin, 1979.
10. L. LeCam, *Asymptotic Methods in Statistical Decision Theory,* Springer Series in Statistics, Vol. 26, New York: Springer-Verlag, 1986.
11. C. Mallows, Some comments on Cp, *Technometrics,* **15**: 661–675, 1973.
12. J. Rissanen, *Stochastic Complexity in Statistical Inquiry,* Series in Computer Science, Vol. 15, World Scientific, 1989.
13. G. Schwartz, Estimating the dimension of a model, *Ann. Stat.,* **6** (2): 461–464, 1978.
14. C. S. Wallace and P. R. Freeman, Estimation and inference by compact coding, *J. Royal Stat. Soc., Ser. B,* **49** (3): 240–265, 1987.

ANATOLI JUDITSKY
INRIA Rhone-Alpes

ESTIMATION THEORY. See CORRELATION THEORY; KALMAN FILTERS.

} { { } }

Wiley Encyclopedia of Electrical and Electronics Engineering

Image Codes
Standard Article

Yung-Kai Lai[1] and C.-C. Jay Kuo[1]
[1]University of Southern California, Los Angeles, CA

DOI: 10.1002/047134608X.W4209
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (399K)

Abstract

The sections in this article are

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*

IMAGE CODES

Although usually acquired from analog optical devices, images are often sampled into the digital form, because they are more easily stored, transmitted, and processed digitally. The major difficulty with digital image storage and transmission, however, is the size of bits required to record the image data. For a 512×512 gray-scale image with eight-bit resolution, 256 kbytes of storage space are required. With color images or digital video data, the amount of data is enormously greater. For example, the bit rate is 8.70 Mbytes/s for a color video sequence with 352×288 pixels per picture, eight bits per color channel, and 30 pictures/s. For a 30 s video clip at such a bit rate, the total data takes up 261 Mbytes of storage space, or 21.12 h of transmission time with a 28800-baud modem. Therefore it is desirable to use data compression techniques to reduce the amount of data for digital images and video.

There are some important differences between digital image/video compressions and other digital data compression. First, for most other data compression applications, it is desirable to have the data themselves unaltered. In digital image compression, on the other hand, some information loss is allowed as long as the visual appearance of the image or video is not severely impaired. In some cases, though, lossless image compression is required. For example, it may be preferred that medical images be losslessly compressed, because small deviations from the original image may affect the doctor's diagnosis. The second difference is that natural images contain much redundant information that is very useful for compression. The background of a natural image, for instance, contains a lot of pixels with similar luminance or color components. These background pixels are represented more efficiently by using various image compression techniques.

Generally speaking, digital image/video compression techniques are classified into two categories: lossless compression and lossy compression. Lossless image/video compression uses many lossless compression techniques mentioned in Data compression, lossy. Lossy image/video compression is more important in image/video compression because the compression ratio is more flexibly adjusted without having to preserve every detail in the image/video. This section primarily focuses on this category, and so do many international standards to be introduced later in this section.

General Lossy Image/Video Compression Framework

The most important issue in image/video compression is reducing the redundancy in the image/video. Most of state-of-the-art lossy image/video compression techniques use transform coding for this purpose. A general image/video compression framework using transform coding includes four major parts: color space conversion, transform, quantization, and entropy coding, as shown in Fig. 1.

Color Coordinates and Chrominance Subsampling. Images are often displayed by the cathode ray tube (*CRT*) using red (R), green (G), and blue (B) phosphor emissions. In compression, however, the RGB color coordinate is not the most efficient for representing the color components of images or video. It is known that the luminance (the intensity of the light, the gray-scale projection of the image) is more important than the chrominance (colors hue and saturation) components in human visual perception. Therefore it is

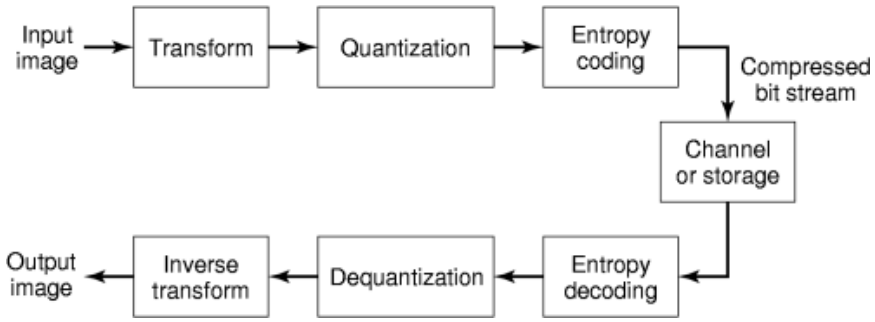


Fig. 1. Block diagram of a general image transform coder. The decoder performs the inverse steps to reconstruct the image.

preferable to transform the color components from the RGB color coordinate to some luminance-chrominance representation so that we put more emphasis on the luminance and discard more unimportant information from the chrominance components without affecting much of the visual perception of compressed images/video. Three often used luminance-chrominance color coordinate systems are YUV, YIQ, and YCbCr color spaces.

The YUV color coordinate was developed by National Television Systems Committee (*NTSC*) and now used in Phase Alternation Line (*PAL*) and Sequentiel Couleur Avec Mémoire (*SECAM*) color television systems. NTSC later developed the YIQ coordinate by rotating the U and V components in YUV space to further reduce the color component bandwidths. The luminance Y and the color components U, V and I, Q in their respective coordinates can be transformed via

$$\begin{bmatrix} \mathrm{Y} \\ \mathrm{U} \\ \mathrm{V} \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.148 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} \mathrm{R} \\ \mathrm{G} \\ \mathrm{B} \end{bmatrix}$$

and

$$\begin{bmatrix} \mathrm{Y} \\ \mathrm{I} \\ \mathrm{Q} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\sin(33^\circ) & \cos(33^\circ) \\ 0 & \cos(33^\circ) & \sin(33^\circ) \end{bmatrix} \begin{bmatrix} \mathrm{Y} \\ \mathrm{U} \\ \mathrm{V} \end{bmatrix}$$

The YCbCr color coordinate was developed as the standard color coordinate system for digital video, as described in ITU-R Recommendation 601 (1). It is an offset and scaled version of the YUV coordinate to limit the dynamic range of luminance and chrominance components within the dynamic ranges of the original RGB components:

$$\begin{bmatrix} \mathrm{Y} \\ \mathrm{Cb} \\ \mathrm{Cr} \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} \mathrm{R} \\ \mathrm{G} \\ \mathrm{B} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.500 \\ 0.500 \end{bmatrix}$$

Because the human visual system is less sensitive to chrominance, the two chrominance components Cb and Cr are usually subsampled to reduce the data volume before compression. Several subsampling formats are commonly used. Two of them, the 4:2:2 format and 4:2:0 format, are used in image/video coding standards, such

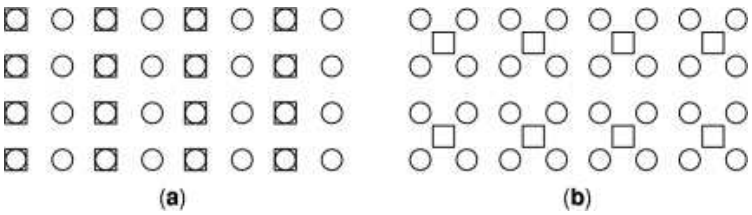


Fig. 2. Different subsampling formats differ in both the ratio of luminance and chrominance samples and their relative positions. Two YCbCr subsampling formats are shown in this figure: (a) 4:2:2, (b) 4:2:0 format. Luminance (Y) and chrominance (Cb, Cr) samples are represented by circles and squares, respectively.

as *JPEG* (Joint Photographic Experts Group) and *MPEG* (Motion Picture Experts Group). These two formats are shown in Fig. 2.

Transform. Transform coding is the most popular coding scheme in scholarly research and industry standards. The purpose of the transformation is to map the digital image from the spatial domain to some transform domain so that its total energy is packed in a small portion of transform coefficients, whereas most other transform coefficients are very close to zero. We can coarsely quantize these unimportant coefficients or simply throw them away in later steps to achieve the goal of compression.

There are several other desirable properties for the transforms used in transform coding. First, a unique inverse transform is required because we have to recover the image from its transform domain. Second, the transform should conserve the total energy in the image. Unitary transforms satisfy these requirements. But not all unitary transforms are suitable for image compression. The energy compaction property and the computational complexity of the transforms are always as important in practical implementation. The optimal transform for energy compaction is known as the Karhunen–Loève transform (*KLT*), but the computational complexity is too high to be practical. Most of the compression standards use the discrete cosine transform (*DCT*). It has a good energy compaction property, and fast algorithms for its forward and inverse transforms are available. Wavelet transform is another promising transform for transform coding and is described in a later section.

Quantization. Most of the transforms convert integral pixel values into floating-point transform coefficients. Encoding these coefficients as floating-point numbers is not economic for lossy compression. Quantization is the process of converting continuous numbers to discrete-value samples. Most transform coding techniques use scalar quantization. The principles of quantization are described in Data compression codes—Lossy and are not discussed in detail here. The output of the scalar quantizer is the index of the reconstruction level. Because quantization is a many-to-one mapping, this is the primary step that causes loss of information in the whole transform coding process.

Entropy Coding. The final stage of the transform coder is to losslessly compress the quantization indexes using an entropy coder for further reduction of compressed bit-stream volume. Two often used entropy coders are the Huffman coder and the arithmetic coder. The details of entropy coders are described in Data compression, lossy.

Image Error Measures. To evaluate the performance of image compression techniques, proper image error measures, which evaluate the difference between the original and compressed images, are necessary. A commonly used image error measure is the mean square error (*MSE*), defined as

$$\mathrm{MSE} = E\{[X(i,j) - X'(i,j)]^2\}$$

where $\mathrm{E}\{\cdot\}$ is the expectation operator, X and X' represent the original and compressed images, respectively, and i, j are the image coordinates of the pixel. The peak signal-to-noise ratio (*PSNR*) is more frequently used

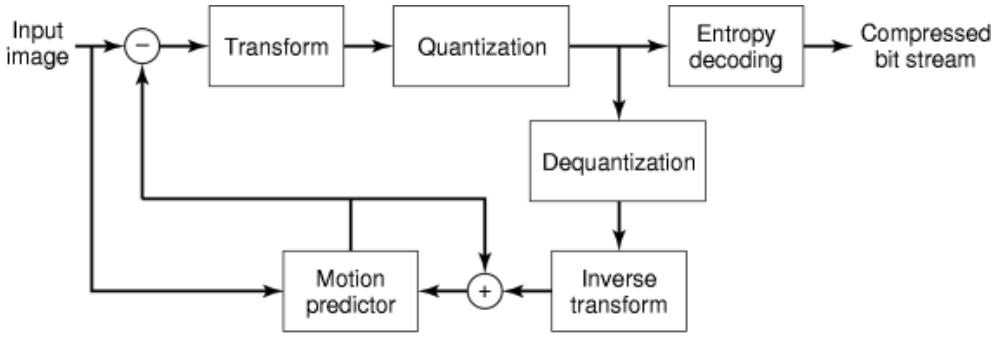


Fig. 3. Block diagram of a general video transform coder using motion-compensated predictive coding. A motion predictor is used to find the motion vector and the estimation error is also transform coded.

in image compression:

$$\text{PSNR} = 10 \log_{10} \left(\frac{P^2}{\text{MSE}} \right) \text{dB}$$

where P is the peak input amplitude. For example, for an eight-bit gray-scale image, $P = 255$. A lower MSE or higher PSNR value means that the compressed image has higher fidelity to the original image. Both MSE and PSNR are conventionally used for gray-scale image error evaluation. There are no consensual error measures for color image compression yet.

Motion-Compensated Predictive Coding. The temporal domain is involved in digital video compression. A digital video is a series of images, called pictures or frames, to be played sequentially. A straightforward way of compressing digital video by image compression techniques is to treat each frame as independent images and compress them separately. However, digital video has redundancies in the temporal domain that are exploited for further compression. Unless there are scene changes, videos usually have many of the same objects in adjacent frames, though the spatial locations on each frame may differ because of object motion. It is a waste to code the same objects on different frames repeatedly. We can encode the object on the first frame and only the direction and distance of object motion in subsequent frames. At the decoder end, after the first frame is decoded, the object on subsequent frames is reconstructed by pasting the object at different spatial locations according to the object motion information. The objection motion direction and distance information are called the motion vector (*MV*), the process to estimate the motion vector between adjacent frames is called motion estimation (*ME*), and the scheme to perform ME and paste the object with respect to the motion vector is called motion compensation (*MC*).

The same object appearing on adjacent frames, however, may appear differently because of light reflection, object rotation, cameras panning or zooming, and so on. Furthermore, new objects may appear which cannot be well estimated with other objects on the previous frame. Therefore motion compensation is only a prediction from previous frames. The difference between the prediction and the actual pixel values has to be computed and encoded. The prediction error, however, would be quite small as long as the ME algorithm finds the minimal-error match from previous frames. The error histogram usually has its peak around zero with small probabilities at large values. The prediction error can be quantized and encoded very efficiently. The block diagram of a general video coder using motion-compensated predictive coding is shown in Fig. 3.

Compression Standards

JPEG Standard. The Joint Photographic Experts Group (*JPEG*) is a working group formed in 1982 under the International Organization for Standardization (*ISO*). This group joined the International Organization for Standardization Consultative Committee (*CCITT*) Special Rapporteur Group (*SRG*) on New Forms of Image Communication to establish an international standard for digital color image compression. After evaluating numerous proposals, they completed the draft technical specification in 1990, and the draft became an international standard in 1992. Some further extensions were developed in 1994. The resulting standard (2), also called JPEG, is now used worldwide for still, continuous-tone, monochrome, and color image compression.

The original JPEG quality requirement is to have indistinguishable images when compressed at 1.50 bits to 2.00 bits per pixel (bpp), excellent image quality at 0.75 bpp to 1.50 bpp, good to very good quality at 0.50 bpp to 0.75 bpp, and moderate to good quality at 0.25 bpp to 0.50 bpp. There are four modes of JPEG operation. They are the sequential baseline DCT-based mode, the progressive DCT-based mode, the sequential lossless mode, and the hierarchical mode. These four modes provide different compression techniques for applications with different requirements. The baseline mode, however, is the mode most often used. The three other modes are rarely used so that many JPEG decode software programs do not even support them.

Baseline Mode. The block diagram of the sequential baseline DCT-based mode JPEG coder and decoder is similar to that shown in Fig. 1. The color image is first converted into the YCbCr coordinates, and then the three components are compressed separately. The core transform used is the discrete cosine transform (DCT), which transforms spatial-domain pixel values into frequency-domain coefficients. To represent the DCT coefficients with 11-bit precision for eight-bit input image (and 15-bit precision for 12-bit input), the three color components in the YCbCr space are level shifted by subtracting 128 (or 2048 for 12-bit input) before performing the DCT. For computational efficiency, the whole input image is partitioned into square blocks of 8×8 pixels each. Then the two-dimensional 8×8 DCT is performed on each block separately:

$$S_{uv} = \frac{C_u}{2}\frac{C_v}{2}\sum_{x=0}^{7}\sum_{y=0}^{7} s_{yx}\cos[(2x+1)u\pi/16]\cos[(2y+1)v\pi/16]$$

$$C_u = \begin{cases} 1/\sqrt{2} & \text{for } u = 0 \\ 1 & \text{for } u > 0, \end{cases} \quad C_v = \begin{cases} 1/\sqrt{2} & \text{for } v = 0 \\ 1 & \text{for } v > 0 \end{cases}$$

where s and S are the 2-D spatial-domain pixel values and the 2-D DCT coefficients, respectively. The subscripts yx and vu are the spatial-domain and frequency-domain coordinates, respectively. S_{00} is called the DC coefficient, and the rest of the 63 coefficients are called AC coefficients. The 8×8 inverse discrete cosine transform (IDCT) used at the decoder end is given by

$$s_{yx} = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} C_u C_v S_{vu}\cos[(2x+1)u\pi/16]\cos[(2y+1)v\pi/16]$$

and 128 (or 2048 for 12-bit input) is added back to restore the original pixel value levels. Numerous fast DCT and IDCT algorithms are available but are not discussed in detail here.

After the DCT operation, the 64 DCT coefficients are quantized by using a lookup table, called the quantization matrix. The default quantization matrices for luminance and chrominance are different because the Human Visual System (*HVS*) has different luminance and color perception characteristics. These two default quantization matrices Q_{vu} are given in Table 1, where the lowest frequency components Q_{00}'s are in the upper left corners. The encoder is also free to define its own quantization matrices, but they have to be

Table 1. JPEG Default Quantization Matrices

(a) Luminance Quantization Matrix

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

(b) Chrominance Quantization Matrix

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

included in the compressed data for the decoder to reconstruct the DCT coefficients. The quantization indexes Sq_{vu} are obtained by dividing the floating-point DCT coefficients by the quantization matrix and rounding the quotient to the nearest integer:

$$Sq_{vu} = \mathbf{R}\text{ound}(S_{vu}/Q_{vu})$$

The reconstructed DCT coefficients R_{vu} are obtained at the decoder side by multiplying the quantization indexes by the quantization matrix:

$$R_{vu} = Sq_{vu}Q_{vu}$$

Because the DC coefficients represent the mean values of the partitioned 8×8 blocks, these coefficients among adjacent blocks are usually quite close to each other in natural images. Thus they are encoded with differential coding in the raster-scan order to take advantage of this property. With the DCT energy compaction property, most of the energy of each 8×8 block is concentrated in low frequencies. Therefore the 63 AC coefficients are encoded in a zigzag order so that the significant coefficients are likely to be encoded first, and in most cases, there are consecutive zero AC coefficients near the end of the block so that they are encoded very efficiently. The differential coding of DC coefficients and the zigzag coding order of AC coefficients is shown in Fig. 4.

Two DC and two AC Huffman tables are used for entropy coding the DCT coefficients. The DC Huffman table for eight-bit resolution is shown in Table 2. The differential (*DIFF*) values range from -2047 to 2047, are classified into 12 categories, denoted as SSSS, and are coded by variable-length codewords. The AC coefficients range from -1023 to 1023 and are classified into 10 nonzero SSSS categories. Because runs of zeros are likely at the high frequencies along the zigzag scan, the lengths of zero runs are encoded with 15 four-bit categories, denoted as RRRR. The combination of RRRRSSSS is encoded using the AC Huffman table with 162 possible

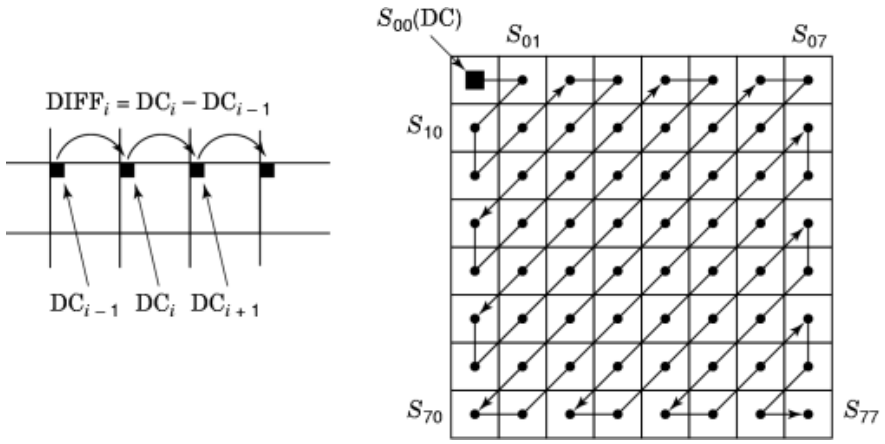


Fig. 4. Differential coding of DC coefficients (left) and zigzag scan order of AC coefficients (right). The differential coding of DC coefficients takes advantage of the cross-block correlation of DC values, whereas the zigzag scan order takes advantage of the energy compaction property so that it is very likely to have consecutive zeros toward the end of the block.

Table 2. Huffman Encoding for DC Coefficients

		Luminance		Chrominance	
DIFF Values	SSSS	Code Length	Code-word	Code Length	Code-word
0	0	2	00	2	00
21, 1	1	3	010	2	01
23, 22, 2, 3	2	3	011	2	10
27 . . . 24, 4 . . . 7	3	3	100	3	110
215 . . . 28, 8 . . . 15	4	3	101	4	1110
231 . . . 216, 16 . . . 31	5	3	110	5	11110
263 . . . 232, 32 . . . 63	6	4	1110	6	111110
2127 . . . 264, 64 . . . 127	7	5	11110	7	1111110
2255 . . . 2128, 128 . . . 255	8	6	111110	8	11111110
2511 . . . 2256, 256 . . . 511	9	7	1111110	9	111111110
21023 . . . 2512, 512 . . . 1023	10	8	11111110	10	1111111110
22047 . . . 21024, 1024 . . . 2047	11	9	111111110	11	11111111110

codes, which are not listed here. A particular RRRRSSSS code is used for zero runs with lengths exceeding 15, and another particular code is used to denote the end of block (*EOB*) when all remaining quantized coefficients in the block are zero.

Progressive Mode. In the baseline mode, the blocks in an image are encoded and decoded in the raster-scan order, that is, from left to right and from top to bottom. The decoder has to receive all the details for one block, decode it, and then proceed to the next block. If, for some reason, the bit stream is cut midway during

the transmission, there is no way the decoder-end user would know the content of the rest of the image. The progressive DCT-based mode uses multiple scans through the image. The DC coefficients of all blocks from the whole image are transmitted first, then the first several AC coefficients of the whole images are transmitted, and so on. In this way, even if the bit stream is cut midway during transmission, it is possible that the whole image with coarser resolution is already available for the decoder-end user to perceive the image content. Progressive image transmission is particularly preferable for image browsing over transmission channels with limited bandwidth. The user can decode the rough image to see if this image carries the required information. If it does, the user can continue the transmission to add more and more details to the image. Otherwise, the user can stop the transmission.

In practice, all DCT coefficients are computed as in the baseline mode and stored in a buffer. The encoder is free to choose the scan number and the coefficients to be transmitted in each scan. For example, the encoder may choose to send the DC coefficients S_{00} in the first scan, S_{01} and S_{10} in the second scan, S_{20}, S_{11}, S_{02} in the third run, S_{03}, S_{12}, S_{21}, S_{30} in the fourth run, and the rest of AC coefficients in the fifth. This choice, called spectral selection, is up to the encoder and can be specified explicitly in the scan header of the compressed bit stream.

In addition to intercoefficient spectral selection, intracoefficient successive approximation is also used for progressive transmission. In short, the successive approximation scheme quantizes the coefficient with lower precision so that a shorter bit stream is transmitted. In every subsequent stage, one more truncated bit is added back to improve the precision of the coefficients by one bit until full precision is reached.

Lossless Mode. Although primarily focused on lossy image compression, JPEG also provides a lossless compression mode. Rather than using the float-point DCT process that introduces error with integral quantization, the lossless mode uses 2-D predictive coding. This predictive coding method uses the upper, left, and upper left neighbor pixels to predict the present pixel value. One of the seven prediction types is chosen and specified in the scan header. The pixels are encoded according to the predictor selected. The lossless mode allows input precision from 2 to 16 bits/sample. The difference between the prediction value and the input is computed modulo 2^{16} and encoded using the Huffman table in Table 2, except that extra entries are added at the end of the table to code the SSSS value from 0 to 16. Arithmetic coding of the modulo difference is also allowed but not discussed in detail here.

Hierarchical Mode. The last mode, the hierarchical mode, is used to generate subimages of smaller size and coarser resolution. First the original image is successively downsampled by a factor of 2 horizontally, vertically, or both. The subimages are smaller versions of the original image with lower resolution. The smallest subimage is transmitted. Then it is upsampled and interpolated bilinearly to form the prediction of the next higher resolution image. The prediction error is encoded and transmitted. The process is repeated until the original image size and resolution are achieved. At the decoder end, a similar process is used to reconstruct the original image by upsampling and adding the prediction error to form multiresolution images. The encoding method can be one of the other three modes: sequential, progressive, or lossless. The hierarchical mode is preferable for platforms with a lower resolution display device or with limited computational power insufficient for reconstructing full-sized images.

JPEG 2000 Standard. New image compression techniques have emerged in recent years since the development of JPEG standard. In addition, JPEG either does not support or does not perform well for some recent applications, such as side-channel information and very low bit-rate coding. All of these encourage the creation of second-generation image compression standards. JPEG 2000, aiming to become an International Standard (*IS*) in year 2000, is the ongoing project for this purpose (3).

The goal of JPEG 2000 is to create a unified system for compressing different types of images (bilevel, gray-scale, or color) with various characteristics and conditions. The purpose of this standard is to complement but not replace the current JPEG standard. Therefore it will focus mainly on the applications for which JPEG fails to provide acceptable quality or performance. The new features of JPEG 2000 will likely include

- High-performance, low bit-rate compression: JPEG performs poorly at a low bitrate, where apparent blocking artifacts appear. JPEG 2000 intends to improve the rate-distortion performance at low bit rates, for example, below 0.25 bpp for gray-scale images, while keeping the excellent performance at higher bit rates. This is the most important function of JPEG 2000.
- Various-type image compression: JPEG focuses mainly on lossy gray-scale and color image compression. The standard for bilevel (such as text) image compression is currently the Joint Bilevel Image Experts Group (*JBIG*) standard. In addition, although providing a lossless mode, JPEG does not perform particularly well in that aspect. JPEG 2000 aims to provide efficient lossless and lossy compression of images with a wide dynamic range, such as bilevel, gray-scale, and color images, all within a unified architecture.
- Robustness to errors: JPEG performs poorly if there is bit error in the bit stream during transmission and storage, for example, when compressed data is transmitted through a noisy channel. JPEG 2000 will incorporate error-resilience or error-correction capabilities into the standard so that the compression bit stream is robust to unexpected errors.
- Content-based description and MPEG-4 interface: one of the most challenging topics in image compression is extracting the semantic content of images and its objects. It benefits applications, such as image retrieval and object-based compression. JPEG 2000 intends to shed light on this problem and hopefully to find some solution for it. In addition, the new video compression method MPEG-4 will use a descriptive language to describe the objects and provide methods to code them. JPEG 2000 is expected to provide an interface for object description and compression for objects.

Other features, including (but not limited to) fixed bit-rate compression, image security, such as image watermarking and encryption, side channel (such as alpha channel and transparency plane) information, random access and processing on arbitrary regions, are also expected to be incorporated into this standard. Though the transform, even the whole framework of JPEG 2000 is likely to be quite different from that used in the existing JPEG, it is desirable that JPEG 2000 should be backward-compatible for JPEG.

MPEG-1 and MPEG-2. The Moving Picture Experts Group (*MPEG*), another working group under ISO, was formed in 1988. It developed the original video coding standard, which was also commonly called MPEG later, for video and associated audio compression. Most of the MPEG parts became an international standard in 1992 (4). Different from JPEG, the second-generation project of MPEG started right after MPEG was completed. To distinguish among generations of MPEGs, the first generation of MPEG is often called MPEG-1. The three essential parts of MPEGs are : video, audio, and systems. We focus only on the video part of this coding standard.

The objective of MPEG-1 is to provide approximately VHS quality of compressed video with a medium bandwidth for 1 to 1.8 Mbps (Mbits/s). It is used for strictly noninterlaced (progressively scanned) video and is optimized for CD-ROM, video CD, and CD-interactive (CD-i) applications. The dimension limits for parameter-constrained video are 768 (h) $\times$ 576 (v) $\times$ 30 (frames/s, fps).

An MPEG-1 video stream includes several hierarchical layers. The whole video sequence is partitioned into at least one *group of pictures* (*GOP*), intended to allow random access into the sequence. Each GOP consists of a certain number of *pictures.* The picture, also called a frame, is the primary coding unit of a video sequence. Each picture consists of three rectangular matrices representing luminance (Y) and two chrominance (Cb, Cr) values. In MPEG-1, the YCbCr matrices are sampled by the 4:2:0 format, that is, the Cb and Cr matrices are subsampled by two horizontally and vertically, therefore their sizes are one-quarter of the Y matrix. Each picture consists of one or more *slices.* The major purpose of slices is to isolate the error in the transmitted bit stream. In the case of a noisy channel, the decoder can skip the erroneous slice and start decoding with the next slice. In an error-free environment, the encoder usually assigns the whole picture to one slice. Each slice is composed of one or more *macroblocks.* The macroblock is the basic coding unit in MPEG. The size of each macroblock is 16 $\times$ 16. In the 4:2:0 format, it consists of six 8 $\times$ 8 *blocks,* four of which are luminance (Y)

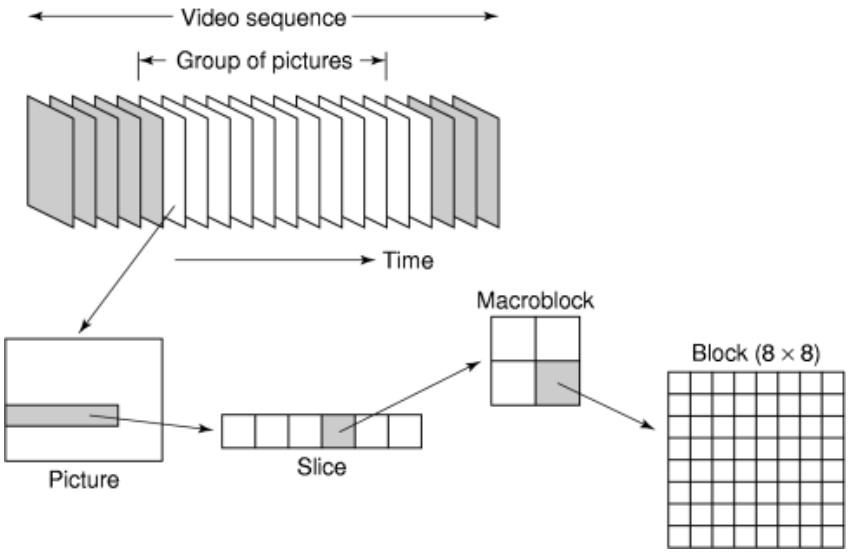


Fig. 5. Video stream data hierarchical layers in MPEG. The four blocks in the macroblocks represent the luminance (Y) blocks in the macroblock. The two 8 × 8 chrominance (Cb and Cr) blocks are downsampled from the 16 × 16 macroblock and are not shown in this figure.

blocks, and the other two are downsampled chrominance (Cb and Cr) blocks. This video stream data hierarchy is shown in Fig. 5.

MPEG-1 uses a DCT-based transform coding scheme to reduce the spatial redundancy, and the motion-compensation technique to reduce the temporal redundancy. MPEG defines four types of pictures: intra (I), predicted (P), bidirectional (B), and DC (D). The D pictures are used only for the fast-forward mode, in which only the DC value of each block is encoded for fast decoding. This type of picture cannot be used in conjunction with the other three types of pictures. It is seldom used thus is not discussed in detail here. A GOP must contain at least one I picture, and may be followed by any number of I, P, B pictures. The I picture is intracoded, which means that it is coded using a still image-coding technique without any temporal dependency on other pictures. The P picture is predicted from a previous I or P picture with its motion information. The B picture is inserted between two I or P pictures (or one of each) and is bidirectionally interpolated from both pictures. With this dependency of I, P, and B pictures, B pictures must be encoded after I or P pictures even though they are displayed before them. Figure 6 shows the picture dependency and the difference between video stream order and display order. Note that this figure only serves as an example of the way the three types of pictures are organized. The actual number of I, P, B pictures in a GOP can be specified arbitrarily by the encoder.

The I pictures are encoded using a JPEG-like image coding scheme. Each 8 × 8 block is level-shifted and transformed by using the 8 × 8 DCT and then quantized by using a default or user-defined intraquantization matrix. The default intraquantization matrix is shown in Table 3(a). The intraquantization matrix can be multiplied by a quantizer scale factor from 1 to 31 from macroblock to macroblock to achieve different bit rates, but the quantization step of DC coefficient is always set to eight. The DC coefficients are differential-coded, whereas the AC coefficients are zigzag scanned and encoded by using run-length coding and Huffman coding, which is similar (but not identical) to JPEG.

The motion-compensation technique is used in coding P pictures. For each 16 × 16 macroblock in the P picture, the most similar macroblock is found in the preceding I or P picture as the prediction or estimation

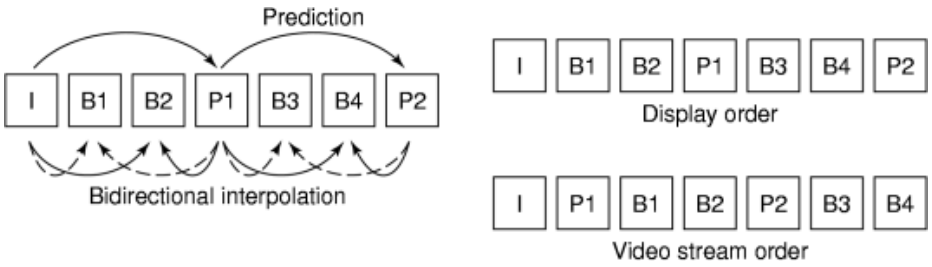


Fig. 6. Interpicture dependency (left) and picture order (right) in MPEG. While the encoder transmits the P frames before the B frames in order to provide interpolation information, it is necessary for the decoder to put the first decoded P frames in a buffer until the subsequent B frames are decoded and to rearrange them for display.

Table 3. MPEG-1 Default Quantization Matrices

(a) Intraquantization Matrix

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

(b) Nonintraquantization Matrix

16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16

of the target macroblock. A motion vector is used to record the spatial displacement between the original and estimated macroblocks in their respective pictures.

The MPEG-1 standard does not define the similarity between two macroblocks and the method of searching for the *most similar* macroblock in the reference picture. The encoder is free to develop its own similarity criterion and motion vector searching algorithm. Two distortion definitions are commonly used to provide the similarity measure:

- Mean squared error (MSE):

$$\mathrm{MSE}(k,l) = \frac{1}{16 \times 16} \sum_{i=0}^{15} \sum_{j=0}^{15} [S_1(u_1+i, v_1+j) - S_2(u_1+i+k, v_1+j+l)]^2$$

- Mean absolute error (MAE):

$$\mathrm{MAE}(k,l) = \frac{1}{16 \times 16} \sum_{i=0}^{15} \sum_{j=0}^{15} \left| S_1(u_1 + i, v_1 + j) - S_2(u_1 + i + k, v_1 + j + l) \right|$$

where S_1 and S_2 are the target and reference pictures, u_1, v_1 are the upper left coordinates in S_1 and S_2, and (k, l) is the MV.

When comparing two MSEs or MAEs, the division of 16×16 is a common factor and thus can be dropped. The smaller the MSE or MAE, the more similar the two macroblocks. The MAE has lower computational complexity and therefore is used more often in MPEG coders. The purpose of the motion vector searching algorithm is to find the MV with the smallest MSE or MAE, and choose it as the best MV representing the motion of the macroblock.

MPEG-1 allows the motion vector to take a large range of displacements in the picture from the reference macroblock. The computational cost to search the whole range, however, is too high to be practical. An efficient encoder usually limits its search to a reasonable range, say, in a 32×32 neighborhood region. This 32×32 region is conventionally called a [−16,15] searching window because the horizontal and vertical displacements for the MV are confined to the [−16, 15] range. The simplest searching algorithm is the full search, that is, to sweeping this searching window pixel-by-pixel and finding the macroblock with the least error. The computational cost is greatly reduced with the logarithmic searching algorithm. In the first step of the logarithmic searching algorithm, eight MVs with large displacement from the starting pixel are selected. The eight errors with respect to these eight MVs are computed and compared to find the smallest one. In the second step, the starting point is taken as the coordinate associated with the smallest error, and the searching displacement is halved. A similar process is repeated until the smallest displacement (one pixel) is met. The process of the full search and the logarithmic search are shown in Fig. 7. MPEG-1 allows half-pixel MV precision if it gives better matching results. The pixel values are bilinearly interpolated to achieve this half-pixel precision. The searching algorithms have to be modified accordingly and the searching range is four times as large with half-pixel precision. Generally speaking, MV search is the most computationally expensive part of the whole MPEG coding process. Many new and efficient searching algorithms have been developed and adopted by commercial encoders.

Depending on the magnitude of motion vector and the prediction error, various coding types can be selected for P picture coding based on the following decisions. How to make these selection decisions is left to the encoder and not explicitly defined in MPEG-1.

- Intra/nonintra: if the intracoding of the macroblock takes less bits than coding the motion vector and the prediction error, we may simply use intracoding as used in I pictures, else nonintracoding of the MV's and prediction errors is used.
- MC/no MC: if the motion vector is very close to zero, we may avoid using MC to save the bits for encoding MVs.
- New/old quantizer scale: if the currently used quantizer scale is not adequate for coding, for example, unable to satisfy the current bit-rate constraint, it may be changed to a new value.
- Coded/not coded: in the nonintracoding case, if the coefficients of a *block* are all zero, the whole block is not coded. In this way a significant number of bits is saved. Because a *macroblock* includes six blocks, if at least one block in a macroblock has to be coded, a coded block pattern has to be included in the bit stream to inform the decoder which blocks in the macroblock are actually encoded.

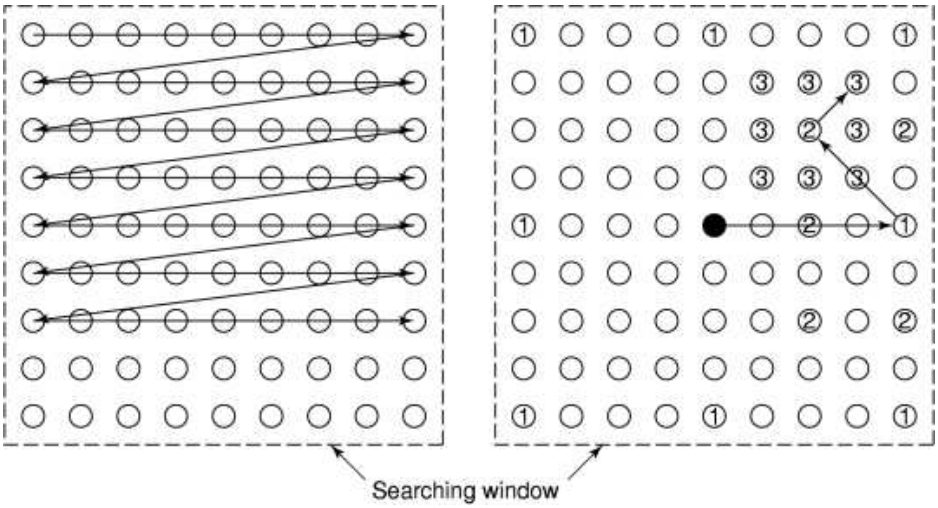


Fig. 7. Two MV searching algorithms: full search (left) and logarithmic search (right), where 1, 2, and 3 indicate the step numbers. In this illustrative example, it takes only 21 MV computations to obtain the best motion vector. On the other hand, all the 81 possible MVs have to be computed in the full search algorithm.

These four decisions generate eight different macroblock types and each is coded differently. A noteworthy macroblock type is the skipped block type, which is not intracoded, having all zero coefficients after quantization, and no MV and quantizer scale change is needed. In other words, the macroblock is identical to the macroblock in the previous I or P picture at exactly the same location. No variable-length code (*VLC*) is needed for this type of macroblocks

In most cases, both the MVs and the prediction errors have to be coded. The MVs of adjacent macroblocks are likely to have similar values because adjacent macroblocks are likely to move coherently in the same direction. Therefore the MVs are encoded with differential coding and further entropy-coded by a variable-length Huffman code. The 16×16 prediction error is encoded by using the transform coding technique similar to that used in encoding I pictures but with a nonintraquantization matrix. The default nonintraquantization matrix is shown in Table 3(b). Another difference from intrablock coding is that the DC coefficients are encoded together with all AC coefficients rather than using a separate differential coder.

The B pictures are obtained by bidirectionally interpolating I or P pictures. At first, one of three MC modes (forward, backward, interpolated) is selected. If the forward mode is selected, a macroblock from the previous I or P picture and the forward MV is used as the prediction. If the backward mode is selected, a macroblock from the future I or P picture and the backward MV is used. If the interpolated mode is selected, one macroblock from the previous and one from the future pictures are bilinearly interpolated to yield the prediction, and both the forward and backward MVs are transmitted.

Similar to P picture coding, three decisions (except for the MC/no MC decision, because all macroblocks have to be motion compensated) have to be made. There are a total of 11 macroblock types in coding B pictures. Different from P picture coding, a skipped block in a B picture uses the same motion vector and same macroblock type as its previous block.

The DCT, quantization, and variable-length codes for B pictures are the same as those of P pictures.

MPEG-2, the second generation of MPEG, focuses primarily on high-quality compressed video for broadcasting. The typical bit rate is 4 Mbps to 15 Mbps, compared to the 1.5 Mbps for MPEG-1. The major applications include digital video disk (*DVD*), digital video broadcasting (*DVB*), and TV systems, such as NTSC and PAL. There was a plan to develop MPEG-3 for high definition television (*HDTV*) applications, but it was later merged

Table 4. MPEG-2 Levels and Profiles

(a)

Level	Max. Dimensions, h × v × fps	Pixels/s	Max. Bit Rate	Significance
Low	352 × 240 × 30	3.04 M	4 Mbps	SIF, consumer tape equiv.
Main	720 × 480 × 30	10.40 M	15 Mbps	CCIR 601, studio TV
High 1440	1440 × 1152 × 30	47.00 M	60 Mbps	4 × CCIR 601, consumer HDTV
High	1920 × 1080 × 30	62.70 M	80 Mbps	Production SMPTE 240 Standard

(b)

Profile	Comments
Simple	Same as Main, only without B-pictures. Intended for software applications, perhaps CATV.
Main	Most decoder chips, CATV satellite. 95% of users.
Main1	Main with spatial and SNR scalability.
High	Main1 with 4:2:2 chroma format.

(c)

	Profile			
Level	Simple	Main	Main1	High
Low	Illegal	∪	Main with SNR scalability	Illegal
Main	∪	∪ (90% of users)	Main with SNR scalability	4:2:2 chroma
High 1440	Illegal	∪	With spatial scalability	4:2:2 chroma
High	Illegal	∪	Illegal	4:2:2 chroma

with MPEG-2. Three parts (systems, video, and audio) of MPEG-2 became IS in 1994, and the other parts were adopted from 1995 to 1997.

Instead of simply providing video compression as in MPEG-1, MPEG-2 focuses on providing more functionality to various applications. MPEG-2 supports both interlaced and noninterlaced video and has a wider range of picture sizes, called *levels.* MPEG-2 also includes several *profiles* to provide different functionality. The combination of levels and profiles for different applications is shown in Table 4. MPEG-2 also provides four scalable modes in Main+ and High profiles for decoders with different capabilities. The *spatial scalable mode* provides two spatial resolution video layers. The *SNR scalable mode* provides two video layers of the same resolution but different quality. The *temporal scalable mode* has one lower layer coded at the basic temporal rate, and the second enhancement layer uses the lower layer as its prediction. The *data partitioning mode* uses progressive transmission which is similar to JPEG's progressive mode.

The MPEG-2 coding scheme (5) is similar to that of MPEG-1 with some modifications and additions for the extra functionality not handled in MPEG-1. The MPEG-2 video syntax (for the main profile) is a superset of that of MPEG-1.

MPEG-4 and MPEG-7. MPEG-4 is an ongoing project of the MPEG family. MPEG-4 (6) focuses mainly on very low bit-rate (64 kbps or less) applications. The major applications are mobile or other telecommunication video applications (5 kbps to 64 kbps) and TV/film applications (up to 2 Mbps). It is expected to be finalized by November 1998.

The three major features of MPEG-4 are *content-based interactivity, compression,* and *universal access.* MPEG-4 plans to provide the functionality of these features to bridge among the TV/film audiovisual data, wireless telecommunication, and computer interactivity.

Topics involved in content-based interactivity include the following:

- Content-based multimedia data access tools: the objects in the coding sequence are segmented and called audio-visual objects (*AVO*). The video is separated into video object planes (*VOP*). Each VOP may have different spatial and temporal resolutions, may have sub-VOPs, and can be associated with different degrees of accessibility, and may be either separated or overlapping.
- Content-based manipulation and bit-stream editing: manipulation and/or editing are allowed to be performed on a VOP, such as spatial position change, spatial scaling, moving speed change, VOP insertion and deletion, etc.
- Synthetic and natural hybrid coding (SNHC): MPEG-4 is intended to compress both natural and synthetic (cartoon, graphics, etc.) video. Issues include the processing of synthetic data in geometry and texture, real-time synchronization and control, integration of mixed media types, and temporal modeling.
- Improved temporal random access.

In compression, several important issues are addressed. The coding efficiency is improved to reduce the bit rate under 64 kbps for mobile applications and 2 Mbps for high-quality TV/film applications. In addition to the objective error measures, such as PSNR, the subjective quality should also be higher compared with existing standards. The ability to encode multiple concurrent data streams, such as the multiple views of a stereo video scene, is also provided. The most important breakthrough in compression, however, should be the object-based coding support. The encoder should be able to encode VOPs with arbitrary shapes, transmit the shape and transparency information of each VOP, support I, P, and B frames of VOPs, and encode the input VOP sequences at fixed and variable frame rates. The coding scheme of MPEG-4 is still block-based DCT coding. To support content-based coding, a square bounding box for an object is first found from the segmentation result. Motion compensation is performed on the macroblocks inside the bounding box. If the macroblock is inside the object, conventional block matching, as in MPEG-1, is used. If the macroblock is complete outside the object, no matching is performed. If the macroblock is partly outside and partly inside the object (that is, the macroblock is on the boundary of the object), some reference pixels have to be padded onto the nonobject area in the macroblock for block matching. Another method is to approximate the boundary macroblocks with polygons and perform polygon matching rather than square block matching.

Two issues are stressed in the universal access feature. One is the robustness in error-prone environments. Because one of MPEG-4's major applications is telecommunication, the channel error over wired and wireless networks has to be considered. The bit stream should be robust for severe error conditions, such as long bursty errors. The other issue is content-based scalability. Scalability in content (spatial, temporal, etc.), quality, and complexity should be allowed.

Because MPEG-4 is intended to provide compression for various types of applications with different bit rates, quality, source material, algorithms, and so on, a *toolbox* approach is adopted for the purpose of integration. In the toolbox approach, a proper *profile* is chosen to satisfy the requirements of the application. The coder selects a compression *algorithm* according to the profile and picks suitable *tools* to realize this

Table 5. Comparison Between H.320 and H.324

Recommendation	H.320 (1990)	H.324 (1995)
Commu. framing and demultiplexing	H.221	H.223
Control	H.230	H.245
Call setup: point-to-point	H.242	H.245
Call setup: multipoint	H.243	H.243
Video coding	H.261	H.263
Audio coding	G.711/G.722/G.728	H.723
Data	T.120	T.120
Network	Above 64 kbps	Below 64 kbps
Network interface	I.400	V.34
Typical network	ISDN-BRI	POTS(GSTN)

algorithm. The MPEG-4 system description language (*MSDL*) allows the transmission in the bit stream of the object structure, rules for the decoder, and the tools not available at the decoder. MPEG-4 also has a close relationship with the virtual reality modeling language (*VRML*) to transmit the description of a 2-D or 3-D scene.

MPEG-7 (7), the newest standard, is currently under development. It will focus mainly on multimedia database management, such as image query, indexing, and retrieval. It is expected to be closely tied with MPEG-4 content-based coding techniques to serve database management purposes.

H.263 Standard. In the 1980s, the International Telephone and Telegraph Consultative Committee (*CCITT*) started its research on low bit-rate videophone and videoconferencing, intended to be transmitted by communication channels with very low bandwidth, such as the telephone lines. The resulting ITU-T H-series recommendations include two H.32X recommendations and their subsystems. The first H.32X system Recommendation H.320, "Narrow-band visual telephone systems and terminal equipment," was finalized in 1990. It targets the bit rate $p \times 64$ kbps ($p = 1$–30). The more recent Recommendation H.324, "Multimedia terminal for low bit-rate visual telephone services over the PSTN," was finalized in 1995. It focuses on bit rates below 64 kbps. The comparison of these two Recommendations is shown in Table 5. We focus here only on the video coding standards H.263 (8).

The acceptable picture formats for H.263 are listed in Table 6. The 4:2:0 YCbCr format is used to represent color frames. Similar to the MPEG family, hierarchical layer representation is also used in H.263. Four layers are used in H.263: the picture layer, the group of blocks (*GOB*) layer, the macroblock layer, and the block layer. Each picture is divided into GOBs. The height of a GOB is defined as 16 pixels for SQCIF, QCIF, and CIF formats, 32 pixels for the 4CIF format, and 64 pixels for the 16CIF format. Each GOB is divided into macroblocks of size 16×16. Each macroblock includes four 8×8 luminance (Y) blocks and two 8×8 chrominance (Cb and Cr) blocks. The building blocks of the encoder include motion-compensation, transform-coding, quantization, and variable-length codes. There are two prediction modes. The *intermode* uses the information in a previous frame for MC. The *intramode* uses only information present in the picture itself. Similar to the picture types in the MPEG family, there are also I, P, and B pictures in the H.263 standard. However, because there is no GOP layer in H.263, the sequence is allowed to extend to an arbitrary number of frames without recurring patterns. However, because the prediction error propagates with each P picture coding, H.263 requires the insertion of at least one I picture in every 132 pictures. H.263 supports half-pixel precision MVs with the searching window of $[-16,15.5]$.

Table 6. Picture Formats Accepted by H.263

Picture Format	Luminance Pixels	Luminance Lines	H.261 Support	H.263 Support	Uncompressed Bit Rate, Mbps, at 10 Frames/s		Uncompressed Bit Rate, Mbps, at 30 Frames/s	
					Gray	Color	Gray	Color
SQCIF	128	96	No	Yes	1.0	1.5	3.0	4.4
QCIF	176	144	Yes	Yes	2.0	3.0	6.1	9.1
CIF	352	288	Optional	Optional	8.1	12.2	24.3	36.5
4CIF	704	576	No	Optional	32.4	48.7	97.3	146.0
16CIF	1408	1152	No	Optional	129.8	194.6	389.3	583.9

Except for the basic mode described above, there are four negotiable modes in H.263, which make the major differences between H.263 and H.261 (and the MPEG family): the *syntax-based arithmetic coding mode,* the *unrestricted motion-vector mode,* the *advanced-prediction mode,* and the *PB-frame mode.*

In the syntax-based arithmetic coding mode, an arithmetic code is used instead of VLC. Arithmetic codes usually provide better entropy-coding performance. The average gain for interframes is about 3% to 4%. For intrablocks and frames, the gain averages about 10%.

The unrestricted MV mode allows the MVs to point outside the picture. This mode is very useful when an object moves along or beyond the edge of the picture, especially for the smaller picture formats. The pixels on the edge row (or column) are replicated to cover the area outside the picture for block matching. This mode includes an extension of the motion-vector range from [−16,15.5] to [−31.5,31.5] so that larger motion vectors can be used.

The advanced prediction mode is used in conjunction with the unrestricted MV mode to achieve better motion prediction. This mode turns on the *four MV option* and the *overlapped-block motion-compensation* (*OBMC*) *option.* With the four MV option, four 8×8 vectors instead of one 16×16 vector are used for some macroblocks in the picture. The MV for the two chrominance blocks is obtained by averaging the four MVs then further dividing the average by two. The OBMC option is used for the luminance component of P pictures. For each 8×8 luminance prediction block, the MV of the current luminance block and two remote MVs are chosen. One remote MV is selected from the two MVs of the blocks to the left and right of the current luminance block and the other from the two MVs of the blocks above and below the current luminance block. Each pixel value in the prediction block is a weighted sum of the three predicted values obtained from the three MVs. The remote MV selection depends on the pixel position in the current block. If the pixel is on the left half of the block, the left-block MV is chosen, otherwise the right-block MV is chosen. The same is true for the top/bottom selection. If one of the surrounding blocks was not coded or was coded in intramode, the corresponding remote MV is set to zero. If the current block is at the border of the picture and therefore a surrounding block is not present, the corresponding remote MV is replaced by the current MV. The advantage of OBMC is that the blocking artifact is greatly reduced since every pixel is predicted by three overlapped blocks.

In the PB-frame mode, two pictures are coded as one unit, called a PB-frame. One P picture is predicted from the last decoded P picture. It is followed by one B-picture predicted from both the last decoded P picture and the P picture currently being decoded. Because most of the information transmitted is for the P picture in the PB-frame, the frame rate can be doubled with this mode without increasing the bit rate much for relatively simple sequences. For sequences with a lot of motion, however, PB-frames do not work as well as the B pictures in MPEG.

With H.263, it is possible to achieve the same quality as H.261 with 30–50% of the bit usage due to the half-pixel prediction and negotiable options in H.263. There are also less overhead and improved VLC tables in H.263. H.263 also outperforms MPEG-1/MPEG-2 for low resolution and low bit rates due to the use of the

four negotiable options. Another reason is that H.263 is less flexible than MPEG thus much less overhead is needed.

More enhancements were made for H.263 and summarized in the new H.263+, a near-term version of H.263. H.263+ supports more picture formats and provides more coding options. The advanced intracoding mode allows predicting an intrablock using neighboring intrablocks. The deblocking filter mode further reduces the blocking artifact by postfiltering the 8×8 block edges with a low-pass filter. The slice-structured mode provides MPEG-like slice structures which act as resynchronization points for bit error and packet loss recovery. The PB-frame mode is also improved to enhance predictive performance. It also adds temporal, SNR, and spatial scalability to H.263. Other enhancements are also made.

H.263L, another refinement of H.263, will be completed in a longer time frame than H.263+. A large change from H.263 and H.263+ is expected, and H.263L might be aligned with the MPEG-4 development and carry similar functionalities.

Other Compression Techniques

Wavelet Compression. Wavelet transform recently attracted a lot of attention in the transform coding field. It provides better performance than DCT-based coding techniques, both in high and low bit rates. Both JPEG 2000 and MPEG-4 are likely to adopt wavelet transforms in their codec.

The wavelet theory states that a signal can be represented by a series of translations and dilations of a basis function that meets certain mathematical requirements. Instead of using the global transformation as in DCT, the wavelet transform uses finite-impulse-response (*FIR*) filters to capture the space-frequency localization characteristics of the signal. This is usually accomplished by using the filter bank approach. The signal is passed through a quadrature mirror filter (*QMF*) bank consisting of a low- and high-pass filter pair denoted, respectively, by $h(k)$ and $g(k)$ with $g(k) = (-1)^k\, h(1-k)$. The forward transform is written as

$$c_k = \sqrt{2} \sum_n h(n-2k) f(n)$$

and

$$d_k = \sqrt{2} \sum_n g(n-2k) f(n)$$

whereas the inverse transform takes the form

$$f'[n] = \sqrt{2} \left(\sum_n h(n-2k) c_k + \sum_n g(n-2k) d_k \right)$$

The $h(n-2k)$ and $g(n-2k)$ are implemented by filtering followed by downsampling operations, when performing the forward transform, or filtering preceded by upsampling operations when performing the inverse transform. The low- or high-passed signals are called the subband signals. The data amount in each of these is half of that of the original signal because of the downsampling process. The 2-D wavelet transform is performed by cascading a horizontal filtering operation and a vertical filtering operation. Thus each subband after the 2-D transform has one-quarter the number of coefficients. The wavelet transform and the subband representation are shown in Fig. 8(a).

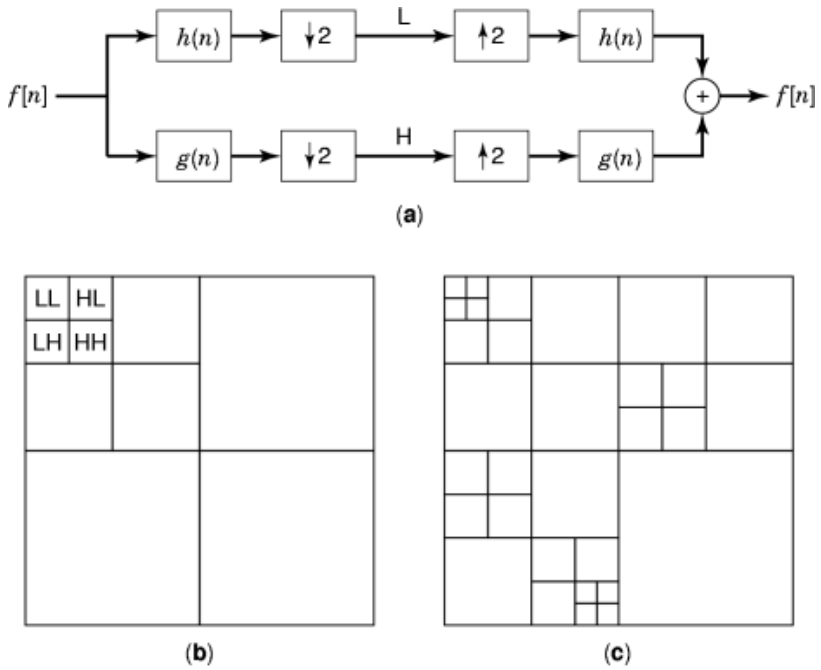


Fig. 8. Wavelet transform: (a) Filter bank implementation. The signal is filtered by high-pass and low-pass filters and downsampled by two. Each of the resulting low-pass and high-pass signals has half of the number of samples. (b) Pyramidal wavelet transforms. Each LL subband is further decomposed to form a regular pyramidal structure. (c) Wavelet packet transform. The subbands are further decomposed according to some criterions, for example, the energy distribution. They do not necessarily form a regular structure, therefore additional structural information has to be coded.

An advantage of wavelet transform in image processing is its flexibility to further decompose the image in the subbands of interest. With the desirable energy compaction property in mind, we can further decompose the subbands with higher energies to refine the bit-allocation strategy in these bands. This approach is called the wavelet packet transform (*WPT*). For most images, however, the low-pass subband usually has the highest energy. Therefore the successive decomposition of the LL band gives fairly good performance. This approach is called the pyramid wavelet transform. These two approaches are shown in Fig. 8(b) and (c).

Wavelet image compression started quite early with performance comparable to DCT compression. The advent of a series of modern wavelet coders, however, boosted the performance while providing a nice embedded bit-stream property. In an embedded bit stream, the transform coefficients are quantized successively so that the most significant coefficients are quantized and transmitted first. More details of the image can be successively added to refine the image if the bit rate allows. In this way, the bit rate is precisely controlled down to the bit level while keeping good performance.

The performance breakthough of modern wavelet coders results from exploiting the correlation between parent and child subbands. Shapiro's embedded zerotree wavelet (*EZW*) coder (9) partitioned the subbands into parent-child groups with same horizontal-vertical wavelet decomposition. If one coefficient in the parent subband is less than some threshold, then the coefficients in the corresponding child subbands are most likely also smaller than this threshold. Therefore only coding the *zerotree* root is enough. After the quantization and grouping procedure, the wavelet coefficients are represented by four symbols, positive, negative, isolated zero, and the zerotree root. They are coded with an arithmetic coder. Both the subjective quality and PSNR are greatly improved.

Several embedded wavelet coders followed the EZW and made more improvements in both the performance and coding complexity. The coefficient representation and prediction scheme were refined by the layer zero coding (*LZC*) technique (10). In LZC, the coefficients were simply represented by zero and one according to their significance, rather than the four-symbol representation of the EZW. The prediction of wavelet coefficients is implemented in the context choice of the adaptive arithmetic coder. The parent-child relationship and the zerotree structure were further exploited by set partitioning in the hierarchical tree (*SPHIT*) algorithm (11), which identified more special classes of tree structures of bits in the significant trees. The multithreshold wavelet coding (*MTWC*) technique (12) uses multiple quantization thresholds for each subband for better bit allocation and rearranges the transmission order of wavelet coefficients to achieve better performance. The latter two have low computational complexity and can be implemented for real-time image compression.

Vector Quantization. Different from scalar quantization, vector quantization (*VQ*) uses a quantization index (codeword) to represent a *vector* to be quantized. Therefore VQ reduces the redundancy if the vectors are closely correlated. VQ is applied to image coding in two ways. One is to use VQ as the replacement of the scalar quantizer in transform coding schemes, and the other is to treat clusters of image pixels as the vectors and perform the VQ.

The first method is useful if the transform coefficients are correlated in some way. For DCT coding, the correlation among DCT coefficients in the same block or across adjacent blocks is not very strong so that VQ cannot improve the performance too much. For wavelet coding, the coefficients are more closely correlated among nearby coefficients in the same subband or among parent-child subbands. Thus using VQ to quantize the wavelet coefficients can improve the performance to a larger extent.

In the second method, the correlation among adjacent pixels in the spatial domain is exploited. To perform VQ, a fixed block size is chosen. The pixel values from the block are chosen as the vector. All of the vectors from several "typical" images are collected as the training vectors, and a training algorithm is chosen. These vectors are trained to form a codebook with a number of representative vectors, called codewords. When compressing an image, every block from the image is extracted as a vector and the nearest codeword from the codebook is found. The index of the codeword is transmitted and the corresponding codeword is used as the reconstruction vector.

The disadvantage of VQ is that the training time required may be too long to form a codebook. Considering the training cost, the proper block size may be about 4×4. In addition, the performance depends on which images are used in the training set and which image is to be compressed. If a universal codebook is used, the performance is not optimal. The performance of traditional VQ is usually a lot worse than the transform coders. The advantage of the VQ coder is that once the training is completed, the encoding speed is quite fast if the size of the codebook is not too large. The decoding speed is extremely fast because no real computation is needed. There are several variants of VQ that improve the performance and reduce the computational complexity (see Data compression, lossy).

Fractal Compression. Fractal compression schemes (13) exploit the spatial redundancy by utilizing the self-similarity in the same image. Given a target region in an image, there could be another region similar to this region with different rotation, scale, and contrast. If we could find this approximation, we could encode the transformation (rotation, scale, contrast, and the displacement) from the target region. This could be very efficient because only a small amount of information needs to be encoded. The coding and decoding of the image is based on the partitioned iterated function system (*PIFS*). The classical way of separating the image into regions is to partition it into fixed-size blocks and find a similar block with some transformations. The transformation information is coded and transmitted. At the decoder end, an initial image (it could even be a random image!) is chosen, and the transformation is applied iteratively to each corresponding block. According to its mathematical theory, the image will converge to the original image after iterations.

The advantage of fractal compression is that it is good for low bit-rate compression because most of the information is included in the image itself and only a small amount of transformation information is needed for encoding a large block. It also alleviates the blocking artifacts in other block-based coding schemes. Another

advantage is that it can be used to enhance the resolution of images even beyond the original resolutions of the images because the iterative process can be extended to subpixel levels. The disadvantage is that the encoding may be too time-consuming in finding a similar block. Rather than pixel-wise matching to find a matched block, fractal compression has to perform all possible transformations to a block in the searching window to find the best match. The decoding time, on the other hand, is relatively fast without too much computation involved. An advantage of the iterative process is that we can cut off the decoding process at an arbitrary number of iterations. But, of course, the result may not be good if too few iterations are performed.

BIBLIOGRAPHY

1. ITU-R Recommendation BT.601, *Encoding parameters of digital television for studios*, 1982.
2. ISO/IEC JTC1 10918-1, *Information technology—digital compression and coding of continuous-tone still images: requirements and guidelines*, 1994.
3. ISO/IEC JTC1/SC29/WG1 N390R, *New work item: JPEG 2000 image coding system*, 1997.
4. ISO/IEC-11172, *Information technology—Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbps*, 1992.
5. ISO/IEC-13818—ITU-T Rec. H.262: *Information technology: Generic coding of moving pictures and associated audio*, 1996.
6. ISO/IEC JTC1/SC29/WG11 N1730, *Overview of the MPEG-4 standard*, 1997.
7. ISO/IEC JTC1/SC29/WG11 N1920, *MPEG-7: Context and objectives*, 1997.
8. ITU-T Recommendation H.263, *Video coding for low bit-rate communication*, 1995.
9. J. Shapiro Embedded image coding using zerotrees of wavelet coefficients, *IEEE Trans. Signal Process.*, **41**: 3445–3462, 1993.
10. D. Taubman A. Zakhor Multirate 3-D subband coding of video, *IEEE Trans. Image Process.*, **3**: 572–588, 1994.
11. A. Said W. A. Pearlman A new, fast, and efficient image codec based on set partitioning in hierarchical trees, *IEEE Trans. Circuits Syst. Video Technol.*, **6**: 243–250, 1996.
12. H.-J. Wang C.-C. J. Kuo A multi-threshold wavelet coder (MTWC) for high fidelity image, *IEEE Signal Process. Soc., 1997 Int. Conf. Image Process.*, 1997.
13. Y. Fisher, ed. *Fractal Image Compression: Theory and Applications*, New York: Springer-Verlag, 1995.

YUNG-KAI LAI
C.-C. JAY KUO
University of Southern California

} { { } }

Wiley Encyclopedia of Electrical and Electronics Engineering

Information Theory
Standard Article

Pramod K. Varshney[1]
[1]Syracuse University, Syracuse, NY

DOI: 10.1002/047134608X.W4206
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (131K)

Abstract

INFORMATION THEORY

The primary goal of a communication system is to convey information-bearing messages from an information source to a destination over a communication channel. All real channels are subject to noise and other channel impairments that limit communication system performance. The receiver attempts to reproduce transmitted messages from the received distorted signals as accurately as possible.

In 1948, Shannon proposed a mathematical theory for the communication process. This theory, known as information theory, deals with the fundamental limits on the representation and transmission of information. Information theory was a remarkable breakthrough in that it provided a quantitative measure for the rather vague and qualitative notion of the amount of information contained in a message. Shannon suggested that the amount of information conveyed by the occurrence of an event is related to the uncertainty associated with it and was defined to be inversely related to the probability of occurrence of that event. Information theory also provides fundamental limits on the transmission of information and on the representation of information. These fundamental limits are employed as benchmarks and are used to evaluate the performance of practical systems by determining how closely these systems approach the fundamental limits.

In his celebrated work, Shannon laid the foundation for the design and analysis of modern communication systems. He proved that nearly error-free information transmission over a noisy communication link is possible by encoding signals prior to transmission over the link and by decoding the received signals. He only provided an existence proof stating that such procedures exist but did not specify an approach to design the best encoders and decoders. Also, he did not discuss the implementation complexity. These results have provided the impetus for researchers to try to design encoding and decoding procedures that approach the fundamental limits given by information theory.

While information theory was primarily developed as a mathematical model for communications, it has had an impact on a wide variety of fields that include physics, chemistry, biology, psychology, linguistics, statistics, economics, and computer science. For example, languages provide a means for communication between human beings, and application of information theory to linguistics arises naturally. Examples of application of information theory to computer science include the design of efficient decision trees and introduction of redundancy in computer systems to attain fault-tolerant computing.

COMMUNICATION SYSTEM MODEL

The main components of a digital communication system are shown in Fig. 1. The source is assumed to be a digital source in that a symbol from a finite alphabet is generated in discrete time. An analog source can be converted to a digital source by sampling and quantization. Data from the source are processed by the source encoder, which represents the source data in an efficient manner. The objective of the source encoding operation is to represent the source output in a compact form with as high fidelity as possible (i.e., with as little information loss as possible). The sequence of source codewords generated by the source encoder is fed to the channel encoder, which yields the sequence of channel codewords. The channel encoder adds redundancy to provide error control capabilities. The goal is to exploit the redundancy in the most effective manner by achieving a high degree of error control capability for a specified amount of redundancy. In some encoding schemes, the input data stream is divided into blocks of fixed length, and then some additional symbols are added to each block to yield channel codewords. These codes are known as block codes. In the class of codes known as tree codes, the encoding process exhibits memory in that a block of input data stream is encoded based on the past blocks also. In either case, the output of the channel encoder is a string of symbols to be transmitted. The modulator converts source codeword symbols to analog waveforms suitable for transmission over the channel. The received waveforms are distorted due to noise and other interference processes present over the channel. The demodulator converts the received waveform into symbols and then furnishes received words to the channel decoder. Due to channel noise, the received word may be in error. The channel decoder exploits the redundancy introduced at the channel encoder to detect and/ or correct errors in the received word. This corrected word is the best estimate of the source codeword, which is delivered to the destination after performing the inverse of the source encoding operation. Information theory is based on a probabilistic model of this communication system.

ENTROPY

Let the discrete random variable S represent the output of a source generating a symbol every signaling interval in a

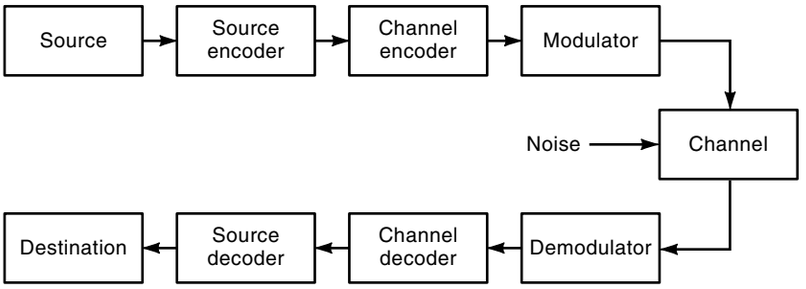


Figure 1. Block diagram of a communication system.

statistically independent manner. This discrete memoryless source (DMS) is assumed to generate symbols from a fixed finite alphabet $\{s_1, \ldots, s_K\}$ with probabilities $P(S = s_k) = p_k$, $k = 1, \ldots, K$. The amount of information gained after observing the symbol s_k is defined by the logarithmic function

$$I(s_k) = \log(1/p_k)$$

It is inversely related to the probability of a symbol occurrence. The base of the logarithm is usually taken to be 2 and the unit is called a bit. In this article, the base of all logarithms is assumed to be 2. Some properties of $I(s_k)$ are as follows:

1. If the outcome of an event is certain, no information gain occurs; that is,

$$I(s_k) = 0 \qquad \text{if} \qquad p_k = 1$$

2. Information gain from the occurrence of an event is nonnegative; that is,

$$I(s_k) \geq 0 \qquad \text{for} \qquad 0 \leq p_k \leq 1$$

3. Occurrence of less probable events results in more information gain; that is,

$$I(s_k) > I(s_\ell) \qquad \text{if} \qquad p_k < p_\ell$$

The average information per source symbol for a DMS is obtained by determining the average of $I(s_1), \ldots, I(s_K)$.

$$H(S) = \sum_{k=1}^{K} p_k \log(1/p_k)$$

This quantity is known as the entropy of the DMS. It characterizes the uncertainty associated with the source and is a function of source symbol probabilities. The entropy is bounded as $0 \leq H(S) \leq \log_2 K$. The lower bound is attained when one of the symbols occurs with probability one and the rest with probability zero. The upper bound is realized when all the symbols are equally likely.

Example: Consider a binary DMS whose output symbols are zero and one with associated probabilities of occurrence given by p_0 and p_1, respectively. The entropy is given by

$$H(S) = -p_0 \log p_0 - p_1 \log p_1$$

It is plotted in Fig. 2 as a function of p_0. Note that $H(S)$ is zero when $p_0 = 0$ or 1. This corresponds to no uncertainty. When $p_0 = \frac{1}{2}$, $H(S) = 1$. This corresponds to maximum uncertainty since symbols 0 and 1 are equally likely.

SOURCE CODING

One of the important problems in communications is an efficient representation of symbols generated by a DMS. Each symbol s_k is assigned a binary codeword of length ℓ_k. For an efficient representation, it is desirable to minimize the average codeword length $\overline{L}$, where

$$\overline{L} = \sum_{k=1}^{K} p_k \ell_k$$

Shannon's first theorem, also known as the source coding theorem, provides a fundamental limit on $\overline{L}$ in terms of the entropy of the source.

Source Coding Theorem: Given a DMS with entropy $H(S)$, the average codeword length $\overline{L}$ for any source encoding scheme is bounded as

$$\overline{L} \geq H(S)$$

Thus, entropy of a DMS provides a fundamental limit on the average number of bits per source symbol necessary to repre-

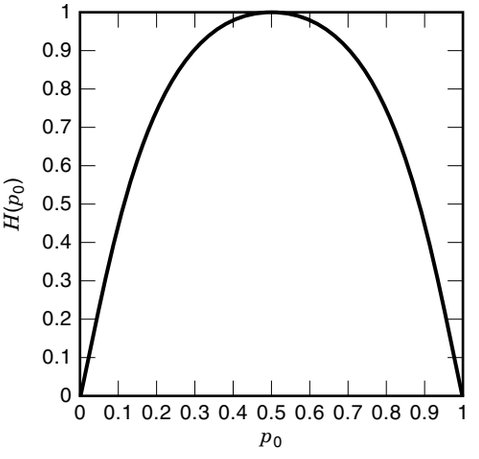
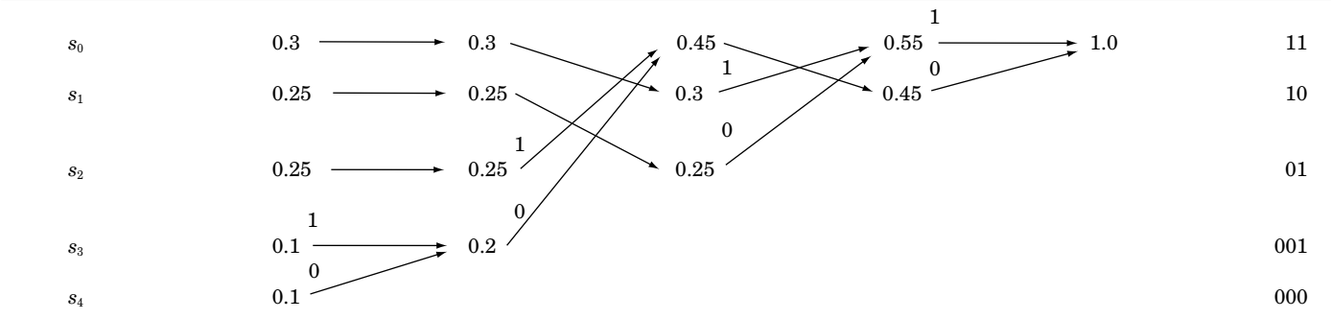


Figure 2. Binary entropy function.

Table 1. Illustration of Huffman Coding Algorithm

Source Symbols	Probabilities at Different Stages 1	2	3	4	5	Codewords
s_0	0.3	0.3	0.45	0.55	1.0	11
s_1	0.25	0.25	0.3	0.45		10
s_2	0.25	0.25	0.25			01
s_3	0.1	0.2				001
s_4	0.1					000



sent the DMS. Based on this lower bound on $\overline{L}$, we can express the coding efficiency of a source encoder as

$$\eta = \frac{H(S)}{\overline{L}}$$

A source encoder that is able to attain the lower bound has an efficiency of one.

An important requirement for source codes is that they be uniquely decodable so that perfect reconstruction is possible from the encoded binary sequence. One class of uniquely decodable codes is the class of prefix-free codes. In these codes, no codeword is a prefix of any other codeword. Huffman code is an example of such a source code in which $\overline{L}$ approaches $H(S)$. This code is optimum in that no other uniquely decodable code has a smaller $\overline{L}$ for a given DMS. The basic procedure for Huffman coding can be summarized as follows:

1. Arrange the source symbols in decreasing order of probabilities.
2. Assign a 0 and a 1 to the two source symbols with lowest probability.
3. Combine the two source symbols into a new symbol with probability equal to the sum of two original probabilities. Place this new symbol in the list according to its probability.
4. Repeat this procedure until there are only two source symbols in the list. Assign a 0 and a 1 to these two symbols.
5. Find the codeword for each source symbol by working backwards to obtain the binary string assigned to each source symbol.

Example: Consider a DMS with an alphabet consisting of five symbols with source probabilities, as shown in Table 1. Different steps of the Huffman encoding procedure and the resulting codewords are also shown. Codewords have been obtained by working backward on the paths leading to individual source symbol.

In this case,

$$\begin{aligned} H(S) = &- 0.3\log 0.3 - 0.25\log 0.25 \\ &- 0.25\log 0.25 - 0.1\log 0.1 - 0.1\log 0.1 \\ &= 2.1855 \text{ bits/symbol} \end{aligned}$$

and $\overline{L} = 2.2$ bits/symbol. Thus, $\overline{L} > H(S)$ and $\eta = 0.9934$.

MUTUAL INFORMATION

Let X and Y be two discrete random variables that take values from $\{x_1, \ldots, x_J\}$ and $\{y_1, \ldots, y_K\}$, respectively. The conditional entropy $H(X|Y)$ is defined as

$$H(X|Y) = \sum_{k=1}^{K}\sum_{j=1}^{J} p(x_j, y_k)\log[1/p(x_j|y_k)]$$

This quantity represents the amount of uncertainty remaining about X after observing Y. Since $H(X)$ represents the original uncertainty regarding X, information gained regarding X by observing Y is obtained by the difference of $H(X)$ and $H(X|Y)$. This quantity is defined as the mutual information $I(X; Y)$.

$$I(X;Y) = H(X) - H(X|Y)$$

Some important properties of $I(X; Y)$ are as follows:

1. The mutual information is symmetric with respect to X and Y; that is,

$$I(X;Y) = I(Y;X)$$

2. The mutual information is nonnegative; that is,

$$I(X;Y) \geq 0$$

3. $I(X; Y)$ is also given as

$$I(X;Y) = H(Y) - H(Y|X)$$

RELATIVE ENTROPY

The relative entropy or discrimination is a measure of the distance between two probability distributions. Let $p(\cdot)$ and $q(\cdot)$ be two probability mass functions. Then relative entropy or Kullback Leibler distance between the two is defined as

$$D(p\|q) = \sum_{k=1}^{K} p(x_k) \log \frac{p(x_k)}{q(x_k)}$$

The relative entropy is always nonnegative and is zero only if p and q are identical.

The mutual information $I(X; Y)$ can be interpreted as the relative entropy between the joint distribution $p(x_j, y_k)$ and the product distribution $p(x_j)\, p(y_k)$. That is,

$$I(X;Y) = D(p(x_j, y_k)\|p(x_j)p(y_k))$$

CHANNEL CAPACITY

Consider a discrete channel with input X and output Y, where X and Y are discrete random variables taking values from $(x_1, \ldots, x_J)$ and $(y_1, \ldots, y_K)$, respectively. This channel is known as a discrete memoryless channel (DMC) if the output symbol at any time depends only on the corresponding input symbol and not on any prior ones. This channel can be completely characterized in terms of channel transition probabilities, $p(y_k|x_j)$; $j = 1, \ldots, J$; $k = 1, \ldots, K$.

Example: An important example of a DMC is the binary symmetric channel (BSC) shown in Fig. 3. In this case, both the input and the output take values from {0, 1} and the two types of errors (receiving a zero when a one is sent, and receiving a one when a zero is sent) are equal.

For a DMC, mutual information $I(X; Y)$ is the amount of input source uncertainty reduced after observing the output. The channel capacity of a DMC is defined as the maximum mutual information for any signaling interval, where the maximization is performed over all possible input probability distributions. That is,

$$C = \max_{\{p(x_j)\}} I(X;Y)$$

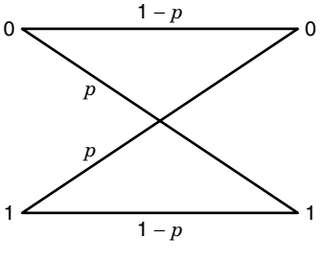


Figure 3. Binary symmetric channel.

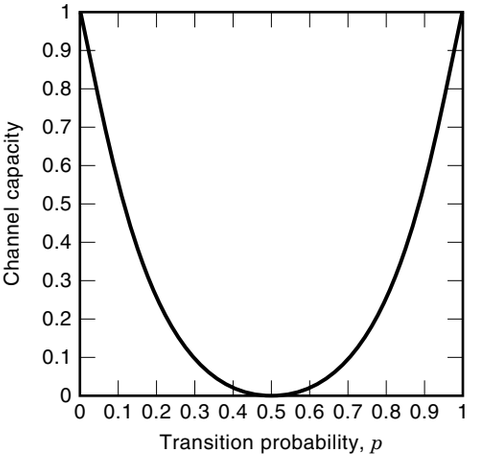


Figure 4. Capacity of a binary symmetric channel.

Channel capacity is a function only of the channel transition probabilities and its units are bits per channel use.

Example: The capacity of a BSC as a function of the error probability p is given by

$$C = 1 - H(p)$$

and is shown in Fig. 4. When $p = 0$ or $p = 1$, the channel capacity is maximum and is equal to 1 bit. Note that $p = 1$ also corresponds to a deterministic channel in that a zero is always received as a one and vice versa. When $p = \frac{1}{2}$, the channel is very noisy and the capacity is zero.

CHANNEL CODING THEOREM

To combat the effects of noise during transmission, the incoming data sequence from the source is encoded into a channel input sequence by introducing redundancy. At the receiver, the received sequence is decoded to reconstruct the data sequence. Shannon's second theorem, also known as the channel coding theorem or the noisy coding theorem, provides the fundamental limits on the rate at which reliable information transmission can take place over a DMC.

Channel Coding Theorem

(i) Let a DMS with entropy $H(S)$ produce a symbol every T_s seconds. Let a DMC have capacity C and be used once every T_c seconds. Then, if

$$\frac{H(S)}{T_s} \leq \frac{C}{T_c}$$

there exists a coding scheme with which source output can be transmitted over the channel and be reconstructed at the receiver with an arbitrarily small probability of error. Here, error refers to the event that a transmitted symbol is reconstructed incorrectly.

(ii) Conversely, if

$$\frac{H(S)}{T_s} > \frac{C}{T_c}$$

it is not possible to transmit data with an arbitrarily small probability of error.

It must be emphasized that the foregoing result only states the existence of "good" codes but does not provide methods to construct such codes. Development of efficient codes has remained an active area of research and is discussed elsewhere in this volume. In error-control coding, redundant symbols are added to the transmitted information at the transmitter to provide error detection and error correction capabilities at the receiver. Addition of redundancy implies increased data rate and thus an increased transmission bandwidth.

DIFFERENTIAL ENTROPY

Thus far, only discrete random variables were considered. Now we define information theoretic quantities for continuous random variables. Consider a continuous random variable X with probability density function $f(x)$. Analogous to the entropy of a discrete random variable, the differential entropy of a continuous random variable X is defined as

$$h(x) = \int_{-\infty}^{\infty} f(x) \log[1/f(x)]\, dx$$

Example: For a Gaussian random variable with probability density function,

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{x^2}{2\sigma^2}\right\}$$

the differential entropy can be computed to be

$$h(x) = \frac{1}{2} \log 2\pi e\sigma^2 \text{ bits}$$

In an analogous manner, mutual information for two continuous random variables X and Y can be defined as

$$I(X;Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) \log \frac{f(x,y)}{f(x)f(y)}\, dx\, dy$$

CAPACITY OF GAUSSIAN CHANNELS

Earlier, the fundamental limit on error-free transmission over a DMC was presented. Here we present the channel capacity theorem for band-limited and power-limited Gaussian channels. This theorem is known as Shannon's third theorem or as the Shannon–Hartley theorem. It is an extremely important result with great practical relevance because it expresses the channel capacity in terms of system parameters channel bandwidth, average signal power, and noise power spectral density.

Channel Capacity Theorem: The capacity of a band-limited additive white Gaussian noise (AWGN) channel is given by

$$C = B \log\left(1 + \frac{P}{N_0 B}\right) \text{ bits/s}$$

where B is the bandwidth of the channel, P is the average transmitted signal power, and the noise power spectral density is equal to $N_0/2$.

The capacity provides a fundamental limit on the rate at which information can be transmitted with arbitrarily small probability of error. Conversely, information cannot be transmitted at a rate higher than C bits/s with arbitrarily small probability of error irrespective of the coding scheme employed.

RATE DISTORTION THEORY

Previously, the problem of source coding that required perfect reconstruction of a DMS was considered. It was seen that the entropy provided the minimum rate at which perfect reconstruction is possible. A question arises as to what happens when the allowed rate is less than the lower bound. Also, what if the source is continuous, because a finite representation of such a source can never be perfect? These questions give rise to rate distortion theory. A distortion measure needs to be defined to quantify the distance between the random variable and its representation. For a given source distribution and distortion measure, the fundamental problem in rate distortion theory is to determine the minimum achievable expected distortion at a given rate. An equivalent problem is to find the minimum rate required to attain a given distortion. This theory is applicable to both continuous and discrete random variables.

Consider a source with alphabet $\mathscr{X}$ that produces a sequence of independent identically distributed random variables $X_1, X_2, \ldots$. Let $\hat{X}_1, \hat{X}_2, \ldots$ be the corresponding reproductions with reproduction alphabet denoted as $\hat{\mathscr{X}}$. The single-letter distortion measure $d(x, \hat{x})$ is a mapping d: $\mathscr{X} \times \hat{\mathscr{X}} \rightarrow R^+$ from the source alphabet-reproduction alphabet pair into the set of nonnegative real numbers. It quantifies the distortion when x is represented by $\hat{x}$. Two commonly used distortion measures are as follows:

Hamming distortion measure:

$$d(x,\hat{x}) = \begin{cases} 0 & \text{if } x = \hat{x} \\ 1 & \text{if } x \neq \hat{x} \end{cases}$$

Squared error distortion measure:

$$d(x,\hat{x}) = (x - \hat{x})^2$$

The single-letter distortion measure can be extended to define the distortion measure for n-tuples as follows:

$$d(x^n, \hat{x}^n) = \frac{1}{n}\sum_{i=1}^{n} d(x_i, \hat{x}_i)$$

This is the average of the per symbol distortion over the elements of the n-tuple.

Now we consider the encoding of the source output sequence of length n, X^n, and then its decoding to yield $\hat{X}^n$. To accomplish this we define a $(2^{nR}, n)$ rate distortion code that consists of an encoding function and a decoding function, as

given by

$$f_n : \mathcal{X}^n \to \{1, 2, \ldots, 2^{nR}\}$$
$$g_n : \{1, 2, \ldots, 2^{nR}\} \to \hat{\mathcal{X}}^n$$

where R is the number of bits available to represent each source symbol. The expected distortion for this rate distortion code is given by

$$D_n = \sum_{x^n} p(x^n) d(x^n, g_n(f_n(x^n)))$$

where $p(\cdot)$ is the probability density function associated with the source.

A rate distortion pair (R, D) is said to be achievable if there exists a rate distortion code with rate R such that

$$\lim_{n \to \infty} D_n \leq D$$

The rate distortion function $R(D)$ is the infimum of rates R such that (R, D) is achievable for a given D. Next, we present the fundamental theorem of rate distortion theory.

Rate Distortion Theorem: The rate distortion function for an independent identically distributed source X with distribution $p(x)$ and bounded distortion function $d(x, \hat{x})$ is given by

$$R(D) = \min_{p(\hat{x}|x):\ \sum_{(x,\hat{x})} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X})$$

Thus, $R(D)$ is the minimum achievable rate at distortion D. Conversely, if R is less than $R(D)$, we cannot achieve a distortion less than or equal to D.

Example: Consider a binary source that produces an output of 1 with probability p. For the Hamming distortion measure, its $R(D)$ is given by

$$R(D) = \begin{cases} H(p) - H(D) & 0 \leq D \leq \min(p, 1-p) \\ 0 & D > \min(p, 1-p) \end{cases}$$

It is illustrated in Fig. 5 for $p = 0.5$.

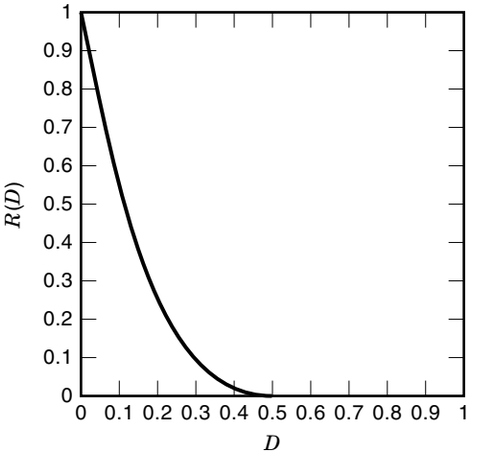


Figure 5. Rate distortion function for the binary source.

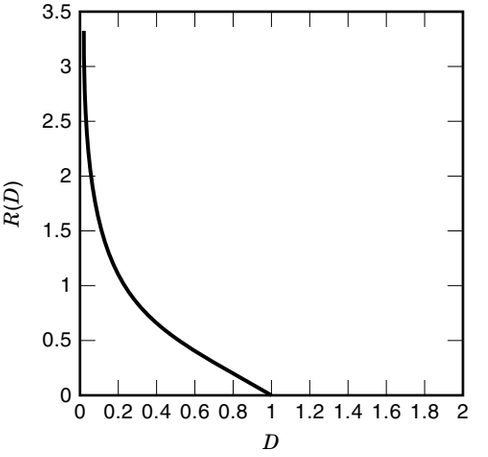


Figure 6. Rate distortion function for the Gaussian source.

Example: Consider a zero-mean Gaussian source with variance σ^2. For the squared error distortion measure, the rate distortion function is given by

$$R(D) = \begin{cases} \frac{1}{2} \log \frac{\sigma^2}{D} & 0 \leq D \leq \sigma^2 \\ 0 & D > \sigma^2 \end{cases}$$

It is plotted in Fig. 6.

The rate distortion function $R(D)$ is a nonincreasing convex function of D. For the binary source, when $D = 0$, the minimum rate required for perfect reconstruction is given by $H(p)$. As D increases, minimum required rate R decreases. Similar observations can also be made for the Gaussian source.

ACKNOWLEDGMENT

I would like to thank Qian Zhang for his help in the preparation of this article. This article was written while the author was a Visiting Scientist at the Air Force Research Laboratory at Rome Research Site, AFRL/IFG, 525 Brooks Road, Rome, NY 13441-4505.

BIBLIOGRAPHY

For a detailed discussion of information theory and its applications, the reader is referred to the sources listed below. Recent results on this topic are published in the *IEEE Transactions on Information Theory.*

T. Berger, *Rate Distortion Theory,* Englewood Cliffs, NJ: Prentice-Hall, 1971.

R. E. Blahut, *Principles and Practice of Information Theory,* Reading, MA: Addison-Wesley, 1987.

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

T. M. Cover and J. A. Thomas, *Elements of Information Theory,* New York: Wiley, 1991.

R. G. Gallagher, *Information Theory and Reliable Communication,* New York: Wiley, 1968.

R. W. Hamming, *Coding and Information Theory,* Englewood Cliffs, NJ: Prentice-Hall, 1980.

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

M. Mansuripur, *Introduction to Information Theory,* Englewood Cliffs, NJ: Prentice-Hall, 1987.

R. J. McEliece, *The Theory of Information Theory and Coding,* Reading, MA: Addison-Wesley, 1977.

C. E. Shannon, A Mathematical Theory of Communication, *Bell Syst. Techn. J.,* vol. 27, pp. 379–423 (part I), and pp. 623–656 (Part II), 1949.

PRAMOD K. VARSHNEY
Syracuse University

INFORMATION THEORY. See MAXIMUM LIKELIHOOD DETECTION.

} { { } }

Wiley Encyclopedia of Electrical and Electronics Engineering

Information Theory of Data Transmission Codes
Standard Article

George Thomas[1]
[1]University of Southwestern Louisiana, Lafayette, LA

DOI: 10.1002/047134608X.W4201
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (265K)

Abstract

The sections in this article are

INFORMATION THEORY OF DATA TRANSMISSION CODES

The basic task of a communication system is to extract relevant information from a source, transport the information through a channel and to reproduce it at a receiver. Shannon, in his ground-breaking *A Mathematical Theory of Communications* (1), quantified the notions of the information rate of a source and the capacity of a channel. He demonstrated the highly non-intuitive result that the fundamental restrictive effect of noise in the channel is not on the *quality* of the information, but only on the *rate* at which information can be transmitted with perfect quality. Shannon considered *coding* schemes, which are mappings from source outputs to transmission sequences. His random-coding arguments established the existence of excellent codes that held the promise of nearly zero error rates over noisy channels while transmitting data at rates close to the *channel capacity*. Shannon's existence proofs did not, however, provide any guidelines toward actually constructing any of these excellent codes. A major focus of research in *information theory* (as Shannon's theory came to be known) over the past 50 years following Shannon's seminal work has been on constructive methods for channel coding. A number of later books (e.g., Refs. 2–9) journalize the development of information theory and coding. In this article we present a broad overview of the state of the art in such data transmission codes.

DATA SOURCES AND CHANNELS

A very general schematic representation of a communication link consists of the following cascade: the source, a source encoder, a channel encoder, a modulator, the channel, the demodulator, the channel decoder, the source decoder, and the receiver. The *source encoder* typically converts the source information into an appropriate format taking into account the quality or *fidelity* of information required at the receiver. Sampling, quantization and analog-to-digital conversion of an analog source, followed possibly by coding for redundancy removal and data compression, is an example. The codes used here are usually referred to as *data compaction* and *data compression* codes. Both strive to minimize the number of bits transmitted per unit of time, the former without loss of fidelity and the latter with possible, controlled reduction in fidelity. This source encoder is followed by the *channel encoder,* which uses *data transmission codes* to control the detrimental effects of channel noise. Controlled amounts of redundancy is introduced into the data stream in a manner that affords error correction. These data transmission codes are the focus of this article. Further down in the cascade, we have the *modulator* which maps output strings from the channel encoder into waveforms that are appropriate for the channel. (Traditionally, modulation has evolved as an art disjoint from coding, but some recent research has indicated the merits of combined coding and modulation. We will touch upon this aspect toward the end of this article.) Following the channel, the demodulator and the decoders have the corresponding inverse functions which finally render the desired information to the receiver. In brief, this article concentrates on data transmission codes.

Binary Symmetric Channels

Each binary digit or (group of digits) at the input of the modulator is transmitted as a waveform signal over the transmission channel. Physical transmission channels may distort the signal, the net result of which is to occasionally reproduce at the output of the demodulator a binary string that is different from what was actually sent. In many practical cases, the error events in successive binary digit positions are mutually statistically independent. And in many such binary memoryless channels the probability of error, ϵ, is the same for a transmitted 0 as well as for a transmitted 1 (Fig. 1). Such a *binary symmetric channel* (BSC) is an important abstraction in data transmission coding.

If a binary n-vector is transmitted sequentially (i.e., bit by bit) over a binary symmetric channel with bit error probability ϵ, the number of errors is a random variable with a Bernoulli distribution:

$$P(i) = \binom{n}{i} \epsilon^i (1-\epsilon)^{n-1}, \qquad 0 \le i \le n$$

If $\epsilon < \frac{1}{2}$, as is the case for most practically useful channels, $P(i)$ is seen to diminish exponentially in i as $(\epsilon/1 - \epsilon)^i$. This implies that $P(0) > P(1) > P(2) > \cdots > P(n)$. More specifically, $P(0)$ is typically very large, $P(1)$ is $O(\epsilon)$ (i.e., on the order of ϵ), $P(2)$ is $O(\epsilon^2)$, and so forth. Thus, even minimal levels of error correction can bring about a significant performance improvement on the BSC.

Hamming Distance and Hamming Weight

The BSC can be modeled as a mod-2 additive noise channel characterized by the relation $Y = X \oplus N$, where X is the

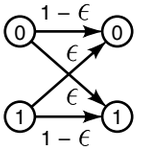


Figure 1. The binary symmetric channel with error probability ϵ.

transmitted binary digit, N is a noise bit, Y is the corresponding output bit, and "$\oplus$" denotes mod-2 addition. The *Hamming weight* of a binary n-vector is defined as the number of 1's in it, and the *Hamming distance* [in honor of R. W. Hamming, a coding theory pioneer (10)] between two binary vectors is defined as the number of bit positions where the elements of the two vectors are different. It is easy to see that the mod-2 sum of two binary n-vectors has a Hamming weight equal to the Hamming distance between the two vectors. If a binary input n-vector $\boldsymbol{X}^n$ to a BSC produces the output n-vector $\boldsymbol{Y}^n$, then the noise pattern $\boldsymbol{N}^n = \boldsymbol{X}^n \oplus \boldsymbol{Y}^n$ is a binary n-vector whose Hamming weight is equal to the Hamming distance between $\boldsymbol{X}^n$ and $\boldsymbol{Y}^n$. (If $a \oplus b = c$, then $a = b \oplus c$).

Consider the n-space $\boldsymbol{S}^n$ of all binary n-vectors. Out of the total 2^n n-vectors in $\boldsymbol{S}^n$, if we choose only a few vectors well separated from each other, we can hope that noise-corrupted versions of one codeword will not be confused with another valid codeword. To illustrate, suppose we choose a code with two binary n-vector codewords $\boldsymbol{X}_1^n$ and $\boldsymbol{X}_2^n$ which are mutually at a Hamming distance d. In Fig. 2 we have shown the example of $\boldsymbol{S}^4$ with $\boldsymbol{X}_1^4 = (0000)$ and $\boldsymbol{X}_2^4 = (1111)$ at Hamming distance $d = 4$. ($\boldsymbol{S}^4$ is a four-dimensional hypercube, with each node having four neighbors at unit distance along the four orthogonal axes.) It can be seen that if codeword 0000 has at most one bit altered by the channel, the resulting 4-tuple (e.g., 0001) is still closer to 0000 than to 1111 so that a nearest-codeword decoding rule decodes correctly. But if 0000 encounters two bit errors (e.g., 0011), the resulting word is at equal distance from either codeword; and if there are three bit errors (e.g., 1101), the nearest codeword now is 1111 and a decoding error results. In general, two codewords at a Hamming distance d can be correctly decoded if the number of errors incurred in the BSC is at most $\lfloor (d-1)/2 \rfloor$, where $\lfloor x \rfloor$ is the integer part of the number x. If in fact there are more than two codewords in the code, it should be obvious that the pair of codewords with the minimum Hamming distance determine the maximum number of bit errors tolerated. Thus, a code with minimum distance $d_{\min}$ can correct all error patterns of Hamming weight not exceeding $t = \lfloor (d_{\min} - 1)/2 \rfloor$.

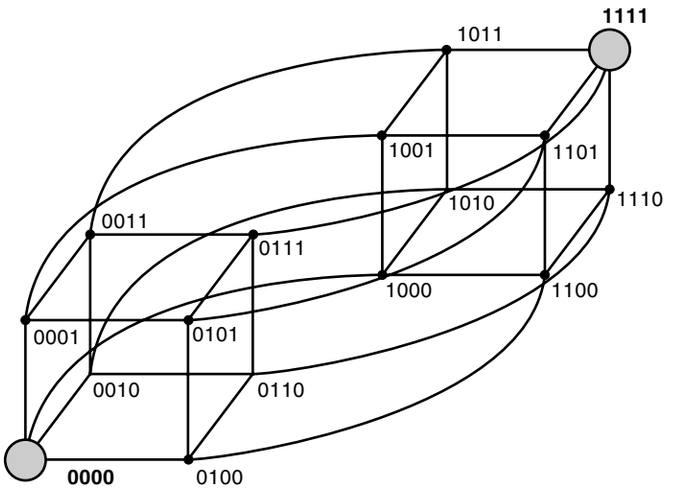


Figure 2. Minimum distance decoding. The two codewords 0000 and 1111 are at Hamming distance 4 in the space of binary 4-tuples.

The essence of code design is the selection of a sufficient number of n-vectors sufficiently spaced apart in binary n-space. Decoding can in principle be done by table lookup but is not feasible in practice as the code size grows. Thus we are motivated to look for easily implemented decoders. Such practical coding schemes fall generally into two broad categories: block coding and convolutional coding.

BLOCK CODING

Linear Block Codes

An (n, k) *block code* maps every k-bit data sequence into a corresponding n-bit codeword, $k < n$. There are 2^k distinct n-vector codewords in a linear block code. The *code rate* $R = k/n$ is a measure of the data efficiency of the code. A linear block code has the property that for any two codewords $\boldsymbol{X}_i^n$ and $\boldsymbol{X}_j^n$, their bitwise mod-2 sum $\boldsymbol{X}_i^n \oplus \boldsymbol{X}_j^n$ is also a codeword. Using a geometric perspective, we can view the code as a k-dimensional linear subspace of the n-dimensional vector space $\boldsymbol{S}^n$, spanned by k basis vectors. Using matrix notation, we can then represent the linear encoding operation as $\boldsymbol{Y}^n = \boldsymbol{X}^k\boldsymbol{G}$, where the k-vector $\boldsymbol{X}^k$ is the data vector, $\boldsymbol{Y}^n$ is the corresponding n-vector codeword, and $\boldsymbol{G}$ is the $k \times n$ binary-valued *generator matrix.* The rows of $\boldsymbol{G}$ are a set of basis vectors for the k-space and thus are mutually linearly independent. Linear codes have the important feature that the minimum distance of the code is equal to the smallest among the nonzero Hamming weights of the codewords. (The all-zero n-vector is necessarily a codeword in each linear n-vector code.) If the codewords are of the specific concatenated form $\boldsymbol{Y}^n = (\boldsymbol{X}^k\boldsymbol{P}^{n-k})$, where $\boldsymbol{P}^{n-k}$ is a *parity vector* comprising $n - k$ *parity bits* which are solely functions of $\boldsymbol{X}^k$ (i.e., if the codeword $\boldsymbol{Y}^n$ contains the data word $\boldsymbol{X}^k$ explicitly), then the code is termed *systematic.* Systematic linear block codes have generator matrices with the special structural form $\boldsymbol{G} = [\boldsymbol{I}_k\boldsymbol{P}]$, where $\boldsymbol{I}_k$ is the $k \times k$ identity matrix and $\boldsymbol{P}$ is a $k \times n - k$ parity generator matrix. Any linear block code can be put into an equivalent code that is also systematic. A (7,4) Hamming code (discussed below) is one example of a linear block code with the following generator matrix in its systematic form:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

For every linear (n, k) block code, there is a *parity check matrix* $\boldsymbol{H}$ which is an $(n - k \times n)$ binary valued matrix with the property that $\boldsymbol{G}\boldsymbol{H}^{\mathrm{T}} = \boldsymbol{0}$. Given $\boldsymbol{G} = [\boldsymbol{I}_k\boldsymbol{P}]$, the corresponding parity check matrix has the structure $\boldsymbol{H} = [\boldsymbol{P}^{\mathrm{T}}\boldsymbol{I}_{n-k}]$. The parity check matrix for the (7,4) systematic Hamming code is as follows:

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

The condition $\boldsymbol{G}\boldsymbol{H}^{\mathrm{T}} = \boldsymbol{0}$ implies that every row in $\boldsymbol{G}$, and consequently every codeword, is orthogonal to every row in $\boldsymbol{H}$. Every codeword $\boldsymbol{X}^n$ satisfies the parity check condition $\boldsymbol{X}^n\boldsymbol{H}^{\mathrm{T}}$

$= \mathbf{0}$. For an arbitrary $\boldsymbol{Y}^n$ appearing at the output of a BSC, the $n - k$ vector $S(\boldsymbol{Y}^n) = \boldsymbol{Y}^n\boldsymbol{H}^{\mathrm{T}}$ is called the *syndrome* of $\boldsymbol{Y}^n$.

The 2^{n-k} syndromes have a one-to-one correspondence with a set of 2^{n-k} n-vector error patterns that the (n, k) linear code is capable of correcting. If n is small, a table lookup will suffice to find the error pattern from the syndrome. A *standard array* (11) as shown below helps to mechanize this procedure:

$$\begin{bmatrix} 0 & X_1 & X_2 & \cdots & X_i & \cdots & X_{2^k-1} \\ N_1 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ N_2 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ N_j & \cdots & \cdots & \cdots & Y^n & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ N_{2^{n-k}-1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

The top row of the standard array consists of the 2^k codewords. The first element, $\boldsymbol{N}_1$ in the next row is chosen to be an n-vector error pattern that the code is expected to correct. It must not be one of the elements in the preceding row(s). The succeeding elements of this row are obtained by adding this error pattern to the corresponding codeword in the top row. Additional rows are formed by repeating this procedure, each time choosing the first element of the row to be a pattern that has not appeared already in the rows above. Each row of the resulting $2^{n-k} \times 2^k$ standard array is called a *coset,* and the first element in each row a *coset leader.* For a BSC with error probability $\epsilon < \frac{1}{2}$, it is natural to choose the coset leaders N to have the least Hamming weight possible. Given the standard array, the output of a BSC, $\boldsymbol{Y}^n$, is located in the standard array. The codeword $\boldsymbol{X}_i$ at the top of the column that $\boldsymbol{Y}^n$ belongs to is declared as the transmitted codeword, with the error pattern produced by the BSC being the coset leader $\boldsymbol{N}_j$ for the coset that $\boldsymbol{Y}^n$ belongs to. If the BSC produces an error pattern which is not one of the coset leaders, the decoder will clearly make a decoding error. In the standard array for the (7,4) Hamming code, the coset leaders can be chosen to be the set of all 7-bit patterns with Hamming weight 1. Hence this code corrects all single error patterns and none else.

The matrix $\boldsymbol{H}^{\mathrm{T}}$ generates an $(n, n - k)$ code (comprising all linear combinations of its $n - k$ linearly independent rows). The codes generated by $\boldsymbol{G}$ and $\boldsymbol{H}^{\mathrm{T}}$ are referred to as dual codes of each other. The weight spectrum of a block code of length n is defined as the $(n + 1)$-vector $(A_0, \ldots, A_n)$, where A_i is the number of codewords with Hamming weight i. The MacWilliams identities (12) link the weight spectra of dual codes. In particular, if $k \gg n - k$, the weight spectrum of the (n, k) code with 2^k codewords may be obtained more easily from the weight spectrum of the dual code with only 2^{n-k} codewords, by means of the MacWilliams identities. The weight spectrum of a linear block code determines the probability of undetected error when the code is used over a BSC. Whenever the n-vector error pattern generated by the BSC coincides with one of the codewords, the error becomes undetectable by the code. This undetected-error probability is

$$P_{\mathrm{UDE}} = \sum_{i=0}^{n} A_i \epsilon^i (1 - \epsilon)^{n-i}$$

and is clearly an important performance parameter, especially when codes are used for error detection only.

For the general class of linear block codes, the encoder implements the multiplication of the data vector by the generator matrix. Decoding consists of computing the syndrome (by matrix multiplication) and looking up the corresponding coset leader in the standard array. These lookup procedures become difficult for codes with moderate to large values of block length n. This motivates the study of a subclass of linear block codes, namely, cyclic codes, with features that facilitate more easily implemented decoders.

Cyclic Codes

A cyclic code is a linear block code with the special property that every cyclic shift of a codeword is also a codeword. Cyclic codes were first proposed by Prange (13). Polynomial algebra, where binary vectors are represented by polynomials with binary coefficients, is a useful framework for characterization of cyclic codes. A binary n-vector $\boldsymbol{X}^n = (x_1, x_2, \ldots, x_n)$ has the polynomial representation $X(D) = x_1D^{n-1} + x_2D^{n-2} + \cdots + x_n$, with degree not exceeding $n - 1$. For instance, (0101) corresponds to $X(D) = D^2 + 1$. Analogous to the generator matrix of a linear block code, a cyclic code can be characterized in terms of a *generator polynomial* $G(D)$ such that every codeword has a polynomial representation of the form $X(D) = G(D)R(D)$. Here $G(D)$ is a polynomial of degree $n - k$ and $G(D)$ is a factor of $D^n - 1$. The polynomial $R(D)$ has degree $k - 1$ or less, representing the k-bit data vector $(r_1, \ldots, r_k)$ being encoded. A code polynomial is generated by multiplying the data polynomial $R(D)$ by the generator polynomial $G(D)$. It can be verified that multiplication of polynomials corresponds to convolution of the corresponding vectors. This observation leads to simple implementation of encoders using shift-register based digital circuits.

Denoting $G(D) = g_0 + g_1D + g_2D^2 + \cdots + g_{n-k}D^{n-k}$, the generator *matrix* $\boldsymbol{G}$ for the linear block code generated by $G(D)$ has the following form:

$$G = \begin{bmatrix} g_0 & g_1 & g_2 & \ldots & 0 & 0 \\ 0 & g_0 & g_1 & \ldots & . & . \\ . & . & . & \ldots & . & . \\ . & . & . & \ldots & g_{n-k} & 0 \\ 0 & 0 & 0 & \ldots & g_{n-k-1} & g_{n-k} \end{bmatrix}$$

As an example, binary cyclic codes of length $n = 7$ are generated by the factors of $D^7 - 1 = (D + 1)(D^3 + D + 1)(D^3 + D^2 + 1)$. The first degree polynomial $G_1(D) = D + 1$ generates the (7, 6) code with a single overall parity bit, while $G_2(D) = D^3 + D + 1$ results in the (7, 4) Hamming code.

The polynomial $H(D)$ such that $G(D)H(D) = D^n - 1$ is known as the *parity check polynomial* for the code generated by $G(D)$. [Since $H(D)$ is also a factor of $D^n - 1$, it also can generate a cyclic code, which is the dual of the code generated by $G(D)$.] The polynomial $H(D) = h_0 + h_1D + h_2D^2 + \cdots + h_kD^k$ specifies the form of the parity check matrix $\boldsymbol{H}$ of the code as follows:

$$H = \begin{bmatrix} h_k & h_{k-1} & \ldots & 0 \\ 0 & h_k & \ldots & h_0 \\ 0 & 0 & \ldots & h_1 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & h_k \end{bmatrix}$$

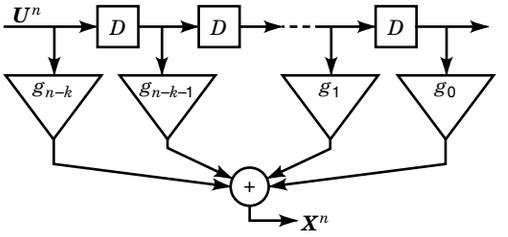


Figure 3. A generic cyclic encoder.

The special structure of $\boldsymbol{G}$ and $\boldsymbol{H}$ for cyclic codes greatly simplifies the implementation of the encoder and the syndrome computer. A generic encoder for a cyclic code is shown in Fig. 3. The k-bit data vector $\boldsymbol{U}^k$ is pipelined through the shift register for n clock times, thus generating the codeword $\boldsymbol{X}^n$ at the output. The encoder utilizes $n - k$ single-bit delay units D, binary multipliers, and binary adders. The circuit complexity is seen to grow only linearly in block length n. For decoding, we can develop a circuit for syndrome calculation for a received n-vector, structured very similarly to Fig. 3. Logic circuits are used to expand the $n - k$ bit syndrome into an n-bit error pattern which is then added to the received codeword to effect error correction.

BCH Codes

Bose and Ray-Chaudhuri (14) and independently Hoquenghem (15) discovered a remarkably powerful subclass of cyclic codes, referred to as BCH codes. The BCH code family is practically the most powerful class of linear codes, especially for small to moderate block lengths. BCH codes can be designed for a guaranteed *design distance* δ (which of course cannot exceed the true minimum distance $d_{\min}$ of the resulting code). Specifically, given δ and hence $t = \lfloor(\delta - 1)/2\rfloor$, and for any integer m, there is a t-error correcting binary BCH code with block length $n = 2^m - 1$ for which the number of parity check bits is no more than mt. Powerful algorithms exist for decoding BCH codes. The polynomial algebraic approach to BCH decoding was pioneered by Peterson (16) for binary BCH codes and extended to nonbinary BCH codes by Gorenstein and Zierler (17). Major contributions came later from Chien (18), Berlekamp (19), and Massey (20). An alternative approach to BCH decoding based on finite field Fourier transforms has gained attention recently, from the work of Blahut (21).

Reed–Solomon Codes

Reed–Solomon codes (22,23) are a very powerful generalization of BCH codes. Binary Reed–Solomon codes can be defined, for any integer m, as follows. Serial data is organized into m-bit symbols. Each symbol can take one of $n = 2^m$ values. The Reed–Solomon code block length is $N = 2^m - 1$ symbols, or Nm bits. Out of these, K symbols are data symbols (i.e., $k = mK$), and $N - K$ symbols are parity symbols computed according to the algebraic description of the code. Reed–Solomon decoding can recover from up to $t = \lfloor(N - K)/2\rfloor$ symbol errors. If symbol erasures are marked as such (i.e., if additional side information is available as to whether a symbol is in error or not, though it is not known what the errors are), then the Reed–Solomon erasure correction limit is $t = N - K$ symbol errors. Since the code can in particular correct t consecutive symbol errors or erasures, it is especially effective against burst errors. The Reed–Solomon codes are *maximum-distance separable;* that is, for the admissible choices of n and k, the Reed–Solomon codewords are spaced apart at the maximum possible Hamming distance.

Perfect Codes

An (n, k) linear code can correct 2^{n-k} error patterns. For some integer t, if the set of error patterns consists of exactly all error patterns of Hamming weight t or less and no other error patterns at all, such a code is termed as a perfect t-error correcting code. This would require, for binary codes, that n, k, and t satisfy the following equality:

$$\sum_{i=0}^{t} \binom{n}{i} = 2^{n-k}$$

The only known perfect codes are the Hamming codes, the double-error-correcting ternary Golay code, and the triple-error-correcting binary Golay code, described below. Tietvainen (24) proved that no other perfect codes exist.

Hamming Codes

Hamming codes are single error correcting codes. For $t = 1$, the condition for a perfect code becomes $1 + n = 2^m$ where $m = n - k$. For integers m, Hamming single-error-correcting codes exist with m parity check bits and block length $n = 2^m - 1$. The rate of the $(2^m - 1, 2^m - m - 1)$ code is $R = (2^m - m - 1)/(2^m - 1)$, which approaches 1 as m increases. The generator matrix G that was displayed earlier while describing linear codes is indeed a generator matrix for a Hamming (7, 4) code. It is possible to rearrange the generator matrix of the code so that the decimal value of the m-bit syndrome word indicates the position of the (single) errored bit in the codeword. Adding an overall parity bit to a basic Hamming code results in a $(2^m, 2^m - m - 1)$ code capable of detecting double errors in addition to correcting single errors. Such codes are particularly effective in data transmission with automatic repeat request (ARQ). If the bit error rate is $\epsilon < \frac{1}{2}$, then the single error patterns which appear with probability $O(\epsilon)$ are the most common and are corrected by the code, thus avoiding the need for retransmission. The double error patterns have a lower probability $O(\epsilon^2)$ and are flagged for retransmission requests. Other error patterns occur with negligibly small probabilities $O(\epsilon^3)$ or less.

Hamming codes are cyclic codes. For block lengths $n = 2^m - 1$, their generator polynomials are the primitive polynomials of degree m over the binary field. (An mth degree primitive polynomial with binary coefficients has the property that its m roots can be characterized as the m primitive elements in a finite field of 2^m elements.) In fact, Hamming codes are BCH codes as well. However, they are decodable by far simpler methods than the general BCH decoding algorithms. Most notably, Hamming codes are perfect codes.

Golay Codes

Two codes discovered by Golay (25) are the only other perfect codes, apart from the Hamming codes mentioned above. For $n = 23$ and $t = 3$, the total number of 0-, 1-, 2-, and 3-error

binary patters of length 23 add up to $2048 = 2^{11}$. Choosing $m = 11$, we can form the (23, 12) triple-error-correcting Golay code. It is also possible to form an (11, 6) double-error-correcting perfect code over ternary alphabet. The Golay codes also are BCH codes, and hence cyclic codes.

Extended and Shortened Codes

Earlier we indicated that a single-error-correcting Hamming code can be made double-error-detecting as well by adding an extra overall parity bit. This results in increasing the block length by one. Such modified codes are known as *extended codes*. Adding an overall parity bit to a code of length $n = 2^m - 1$ results in a codeword whose length is a power of two. This may be advantageous in byte-oriented data handling, or in matching a prespecified field length in a data packet. Extended Reed–Solomon codes are another practical example. As already seen, the natural block length of Reed–Solomon codes is $2^m - 1$ m-bit symbols, and frequently there are reasons to have a block length which is a power of two. Adding an overall parity symbol accomplishes this task. Extended codes may have smaller minimum distance than their original counterparts, but in many instances the minimum distance remains unchanged. An example is the (7, 4) Hamming code and its (8, 4) extended version, both of which have minimum distance 4.

Shortened codes result from the reverse process where we seek to reduce the block length of a basic code. For example, the Reed–Solomon code with 8-bit symbols has a natural block length of 255 symbols. If the encoded data is to be transported in fixed length packets of 240 symbols each, we can set 15 information symbols to zeroes and then delete them before transmission. Shortening can increase minimum distance in general. The shortened code has fewer information bits and the same number of parity bits, so that the error correction capability normalized with respect to block length increases upon shortening.

Product Codes

Elias (26) showed how to combine two block codes into a *product code*. Cyclic product codes were studied by Burton and Weldon (27). Suppose we have an (n_1, k_1) code and an (n_2, k_2) code. Arrange $k = k_1k_2$ data bits in an array of k_1 rows and k_2 columns. Extend each row of k_2 bits into an n_2 bit codeword using the (n_2, k_2) code. Next, extend each of the resulting n_2 columns to n_1 bits using (n_1, k_1) code. The resulting array of $n = n_1n_2$ bits is the product encoding for the original k bits. The rate of the product code is the product of the rates of the constituent codes. If the constituent codes respectively have minimum distances d_1 and d_2, the product code has a minimum distance $d_{\min} = d_1d_2$. Product codes are frequently capable of correcting not only all error patterns of weight $\lfloor (d_1d_2 - 1)/2 \rfloor$ but also many higher weight patterns. However, the simplistic approach of row-wise decoding first, followed by column-wise decoding, may not achieve the full error correction capability of product codes.

Interleaved Coding

The binary symmetric channel models the random error patterns which are bit-to-bit independent. That the bit error probability ϵ is less than 0.5 is the basis for the minimum weight choice of coset leaders (correctable error patterns) in the standard array. In a burst noise channel, errored bit positions tend to cluster together to form error bursts. Codes designed to correct minimum weight error patterns are not directly useful in presence of burst errors. *Interleaved coding* is a technique that allows random-error-correcting codes to effectively combat burst errors. Interleaving renders the burst noise patterns of the channel as apparent random errors to the decoder.

Figure 4 depicts an elementary block interleaver to illustrate the idea. Suppose a burst noise channel is known to generate error bursts spanning at most six consecutive bits. Further, suppose that a (7, 4) Hamming single-error-correcting code is to be used. First we read 24 consecutive data bits row-wise into a 6×4 array, as in Fig. 4. The column size is chosen to be six so as to at least equal the maximum burst length. The row size is chosen to be exactly equal to the number of information digits in the code. Next we extend each row by three more positions by appending the three parity bits appropriate for a Hamming (7, 4) codeword. The extended transmission array is therefore 6×7. The transmission sequence is column-wise—that is, in the sequence 1, 5, 9, 13, 17, 21, 2, 6, 10, 14, . . . (see Fig. 4). The array transmits 42 bits for each 24-bit input. A six-bit error burst may affect the successively transmitted bits 10, 14, 18, 22, 3 and 7, as shown in Fig. 4. Notice that each bit in this error burst belongs to a different Hamming codeword. Thus all such errors are corrected if the error burst does not exceed six bits and there are no other burst error within this span of 42 bits. Suitably sized block interleavers are often effective in burst noise environments.

Concatenated Codes

Consider the following example. Let $n = 8$ and $N = 2^n - 1 = 255$. An (N, K) Reed–Solomon code has $N = 255$ code symbols which can be represented as 8-bit binary words. Transmitted through a binary symmetric channel with bit error probability ϵ, each 8-bit symbol can be in error with probability $\Delta = 1 - (1 - \epsilon)^8$. The code can recover from up to $t_0 = \lfloor (N - K)/2 \rfloor$ symbol errors. The probability of successful decoding is, therefore,

$$P_{s0} = \sum_{i=0}^{t_0} \binom{N}{i} \Delta^i (1 - \Delta)^{255-i}$$

We can now *concatenate* the Reed–Solomon code with a Hamming (8, 4) single-error correcting/double-error detecting

1	2	3	4	P1	P2	P3
5	6	7	8	P4	P5	P6
9	10	11	12	P7	P8	P9
13	14	15	16	…	…	…
17	18	19	20	…	…	…
21	22	23	24	…	…	…

Figure 4. An illustration of interleaved coding.

code, as follows. Each 8-bit symbol of the Reed–Solomon code is chosen as a Hamming (8, 4) codeword. To do this, we organize raw data into consecutive blocks of four bits and encode each such into a Hamming (8, 4) codeword. Then each set of K consecutive 8-bit Hamming codewords is encoded into a 255-symbol Reed–Solomon codeword. The probability of a symbol error now becomes smaller, $\delta = 1 - (1 - \epsilon)^8 - 8 \in (1 - \epsilon)^7$, assuming triple and higher errors are negligibly infrequent. Besides, the double-error detection feature identifies the errored symbols in the Reed–Solomon code. With this side information, the Reed–Solomon code can now recover from a greater number of symbol errors, $t_1 = 255 - K$. The probability of successful decoding is now found as

$$P_{s1} = \sum_{i=1}^{t_1} \binom{N}{i} \delta^i (1-\delta)^{255-i}$$

The power of this concatenated coding approach is evident from comparing the above two expressions for the probability of successful decoding. The Reed–Solomon code is the outer code and the Hamming code is the inner code. The inner code cleans up the milder error events and reserves the outer code for the more severe error events. In particular, in a burst noise environment, a long codeword may have some parts completely obliterated by a noise burst while other parts may be affected by occasional random errors. The inner code typically corrects most of the random errors, and the outer Reed–Solomon code combats the burst noise. In most applications the outer code is a suitably sized Reed–Solomon code. The inner code is often a convolutional code (discussed below), though block codes can be used as well, as shown above.

The invention of concatenated coding by Forney (28) was a major landmark in coding theory. Later, Justesen (29) used the concatenation concept to obtain the first constructive codes with rates that do not vanish asymptotically for large block lengths.

Performance Limits of Block Codes

The key performance parameters of a block code are the code rate and the minimum distance. In this section we highlight some of the known bounds on these parameters. The *Hamming bound,* also known as the *sphere packing bound,* is a direct consequence of the following geometrical view of the code space. Let an (n, k) code have minimum distance d. There are 2^k codewords in this code. Around each codeword we can visualize a "sphere" comprising all n-vectors that are within Hamming distance $\lfloor (d-1)/2 \rfloor$ from that codeword. Each such sphere consists of all the n-tuples that result from perturbations of the codeword at the center of the sphere by Hamming weight at most $\lfloor (d-1)/2 \rfloor$. Any two such spheres around two distinct codewords must be mutually exclusive if unambiguous minimum-distance decoding is to be feasible. Thus the total "volume" of all 2^k such mutually exclusive spheres must not exceed the total number of possible n-tuples, 2^n. Thus,

$$2^n \geq 2^k \sum_{i=0}^{t} \binom{n}{i}$$

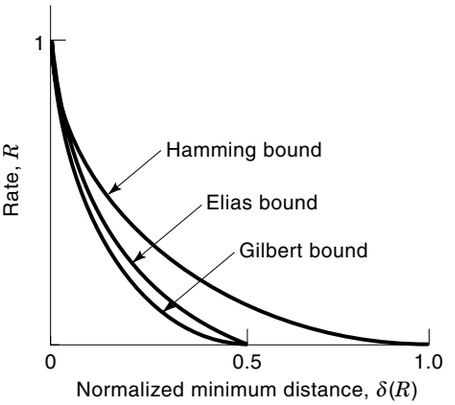


Figure 5. Bounds on the minimum distance of block codes.

This can be expressed in the following form of the Hamming upper bound on code rate R:

$$R \leq 1 - \frac{1}{n} \log \sum_{i=0}^{t} \binom{n}{i}$$

Asymptotically for large n, this reduces to the form

$$R \leq 1 - H_2\left(\frac{\delta(R)}{2}\right)$$

where $H_2(x) = -x \log x - (1-x) \log(1-x)$ is the binary entropy function; $\delta(R)$ is the largest value of the normalized minimum distance of a rate-R code as the block length n goes to vary large values (i.e., $\lim \sup_{N\to\infty} d_{\min}/n$); and the logarithm is to the base 2 so that R is in bits of information per binary digit.

The Hamming upper bound asserts that for a given $\delta(R)$, no code can exceed the rate given by the bound above. The *Gilbert bound,* on the other hand, is a constructive bound which states that it is possible, for a given $\delta(R)$, to construct codes with rates at least as large as the value R specified by the bound. The asymptotic Gilbert bound states that

$$R \geq 1 - H_2(\delta(R))$$

The *Elias bound* is a tighter upper bound on the feasible code rates compared to the Hamming bound. In its asymptotic form the Elias bound is stated as follows:

$$\delta(R) \leq 2\lambda_R(1 - \lambda_R)$$

where

$$R = 1 - H_2(\lambda_R)$$

These bounds are shown in Fig. 5. The feasibility region of "good" block codes lies between the Gilbert and Elias bounds. Hamming bound originally appeared in Ref. 10, and the Gilbert bound in Ref. 30. The Elias bound was first developed by Elias *circa* 1959 but appeared in print only in 1967 paper by Shannon, Gallager, and Berlekamp (see Ref. 5, p. 3). Proofs for these bounds are found in many coding theory books (e.g., Ref. 3). It had been conjectured for some time that the Gilbert

bound was asymptotically tight—that is, that it was an upper bound as well as a lower bound and that all long, good codes would asymptotically meet the Gilbert bound exactly. This perception was disproved for nonbinary codes by the work of Tsfasman et al. (31). Also McEliece et al. (32) obtained some improvements on the Elias bound. See also Ref. 33 for tabulations of the best-known minimum distances of block codes.

CONVOLUTIONAL CODES

Convolutional Encoders

Convolutional codes were originally proposed by Elias (34). Probabilistic search algorithms were developed by Fano (35) and Wozencraft and Reiffan (36) as practical decoding algorithms. Massey (37) proposed the use of threshold decoding for convolutional codes as a simpler though less efficient alternative. The Viterbi algorithm (38) was developed later as an efficient decoder for short convolutional codes. We will briefly outline Wozencraft's sequential decoding and the Viterbi algorithm, after examining the basic structure of convolutional codes.

Convolutional coding is based on the notion of passing an arbitrarily long sequence of input data bits through a linear sequential machine whose output sequence has memory properties and consequent redundancies that allow error correction. A linear sequential machine produces each output symbol as a linear function of the current input and a given number of the immediate past inputs, so that the output symbols have "memory" or temporal correlation. Certain symbol patterns are more likely than others, and this allows error correction based on maximum likelihood principles. The output of the linear sequential machine is the convolution of its impulse reponse with the input bit stream, hence the name. Block codes and convolutional codes are traditionally viewed as the two major classes error correction codes, although we will recognize shortly that it is possible to characterize finite length convolutional codes in a formalism similar to that used to describe block codes.

A simple convolutional encoder is shown in Fig. 6. For every input bit, the encoder produces two output bits. The code rate is hence $\frac{1}{2}$. (More generally, a convolutional encoder may accept k input bits at a time and produce n output bits, implementing a rate k/n code.) The output of the encoder in Fig. 6 is a function of the current input and the two previous inputs. One input bit is seen to affect three successive pairs of output bits. We say that the *constraint length* of the code is therefore $K = 6$. There are other definitions of the constraint length, as the number of consecutive input bits that affect a given output ($K = 3$) or the minimum number of delay elements needed to implement the encoder ($K = 2$).

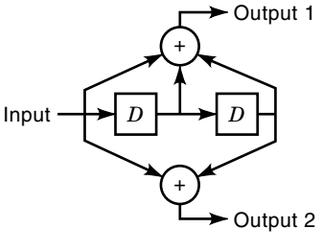


Figure 6. A convolutional encoder.

It must be noted that Fig. 6 could have been redrawn with only two memory elements to store the two previous bits; the current input bit could be residing on the input line. The memory order of the encoder in Fig. 6 is thus only two, and the encoder output is determined by the state of the encoder (which is the content of the two memory registers) and by the new input bit. Whether we use an extra memory register to hold the incoming new bit or not is similar in spirit to the distinction between the Moore and the Mealy machines in the theory of finite-state sequential machines (39).

The impulse response of the encoder at the upper output of the convolutional encoder in Fig. 6 is '1 1 1' and that at the lower output line is '1 0 1'. The output sequences at these lines are therefore the discrete convolutions of the input stream with these impulse responses. The following infinite-dimensional generator matrix represents the mapping of the infinite input sequence $(x_0, x_1, x_2, \ldots)$ into the infinite output sequence $(y_0, y_1, y_2, \ldots)$ where y_{2n} and y_{2n+1} are the two output bits corresponding to input bit x_n:

$$x_0x_1x_2\ldots]\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \ldots \\ 0 & 0 & 0 & 0 & 0 & 0 & \ldots \\ 1 & 1 & 0 & 0 & 0 & 0 & \ldots \\ 1 & 0 & 1 & 1 & 0 & 0 & \ldots \\ 1 & 1 & 1 & 0 & 1 & 1 & \ldots \\ 0 & 0 & 1 & 1 & 1 & 0 & \ldots \\ 0 & 0 & 0 & 0 & 1 & 1 & \ldots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} = [y_0y_1y_2\ldots$$

Also, in terms of the impulse response polynomials $G_1(D) = 1 + D + D^2$ and $G_2(D) = 1 + D^2$, respectively, for the upper and lower output lines in Fig. 6, we can relate the input polynomial $X(D)$ to the respective output polynomials as

$$Y_i(D) = X(D)G_i(D), \qquad i = 1, 2$$

However, these matrix and polynomial algebraic approaches are not as productive here as they were for the block codes. More intuitive insight into the nature of convolutional codes can be furnished in terms of its tree and trellis diagrams.

Trees, State Diagrams, and Trellises

The most illuminating representation of a convolutional code is in terms of the associated tree diagram. The encoding process starts at the root node of a binary tree, as shown in Fig.

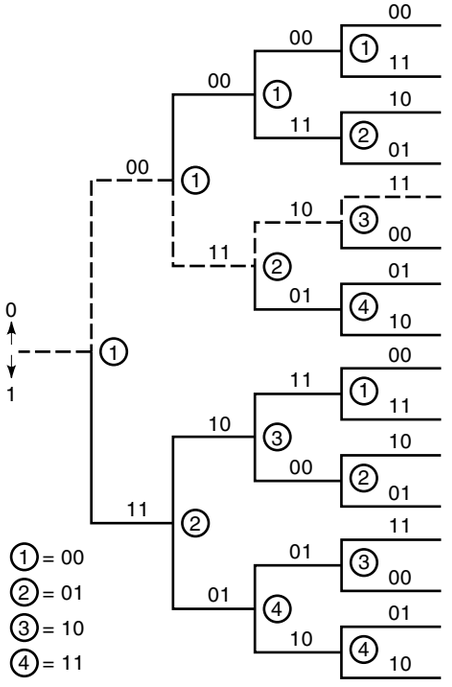


Figure 7. A tree diagram for the convolutional encoder in Fig. 6.

7 for the encoder in Fig. 6. Each node spawns two branches. Each successive input bit causes the process to move to one of the next higher level nodes. If the input bit is a zero, the upper branch is taken, otherwise the lower one. The labeling on each branch shows the bit pair produced at the output for each branch. Tracing an input sequence through the tree, the concatenation of the branch labels for that path produces the corresponding codeword.

Careful inspection of the tree diagram in Fig. 7 reveals a certain repetitive structure depending on the "state" of the encoder at each tree node. The branching patterns from any two nodes with identical states are seen to be identical. This allows us to represent the encoder behavior most succinctly in terms of a state transition diagram in Fig. 8. The state of the encoder is defined as the contents of the memory elements at any time. The encoder in Fig. 7 has four states, 1 = 00, 2 = 01, 3 = 10 and 4 = 11. The solid lines in Fig. 8 indicate state transitions caused by a zero input, and the dotted lines indicate input one. The labels on the branches are the output bit pairs, as in the tree diagram in Fig. 7.

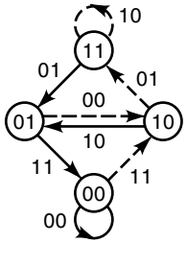


Figure 8. The state transition diagram for the convolutional encoder in Fig. 6.

Data sequences drive the encoder through various sequences of state transitions. The pattern of all such possible state transition trajectories in time is known as a *trellis diagram.* In Fig. 9 we have the trellis diagram for an encoder that starts in state 00 and encodes a 7-bit input sequence whose last two bits are constrained to be zeroes. This constraint, useful in Viterbi decoding to be described below, terminates all paths in state 00. The trellis diagram in Fig. 9 thus contains 2^5 distinct paths of length 7 beginning and ending in state 00.

Weight Distribution for Convolutional Codes

An elegant method for finding the weight distribution of convolutional codes is to redraw the state transition diagram such as in Fig. 8, in the form shown in Fig. 10 with the all-zero state (00 in our example) split into two, a starting node and an ending node. To each directed path between two states, we assign a "gain" W^i, where W is a dummy variable and the exponent i is the Hamming weight of the binary sequence emitted by the encoder upon making the indicated state transition. For example, in Fig. 10, the transition from 1 to 2 causes the bit pair 11 to be emitted, with Hamming weight $i = 2$, so that the gain is W^2. In transitions that emit 01 or 10, the gain is W and in the case where 00 is emitted, the gain is $W^0 = 1$. We can now use a "signal flow graph" technique due to Mason (40) to obtain a certain "transfer function" of the encoder. In the signal flow graph method, we postulate an input signal S_{in} at the starting state and compute the output signal S_{out} at the ending state, using the following relations among signal flow intensities at the various nodes:

$$\begin{aligned} S_{\text{out}} &= S_2 W^2 \\ S_2 &= (S_3 + S_4)W \\ S_4 &= (S_3 + S_4)W \\ S_3 &= S_{\text{in}} W^2 \end{aligned}$$

The transfer function $T(W) = S_{\text{out}}/S_{\text{in}}$ can be readily found to be

$$T(W) = \frac{W^5}{(1 - 2W)} = \sum_{i=0}^{\infty} 2^i W^{5+i} = W^5 + 2W^6 + 4W^7 + \cdots$$

Each term in the above series corresponds to a set of paths of a given weight. The coefficient 2^i gives the number of paths of weight $5 + i$. There is exactly one path of weight 5, two paths of weight 6, four of weight 7, and so on. There are no paths of weight less than 5. The path with weight 5 is seen to be the closest in Hamming distance to the all-zero codeword. This distance is called the *free distance,* d_{free}, of the code. In the present example, $d_{\text{free}} = 5$. The free distance of a convolutional code is a key parameter in defining its error correction, as will be seen in the next section.

Maximum Likelihood (Viterbi) Decoding for Convolutional Codes

Each path in a trellis diagram corresponds to a valid code sequence in a convolutional code. A received sequence with bit errors in it will not necessarily correspond exactly to any one particular trellis path. The Viterbi algorithm (38) is a

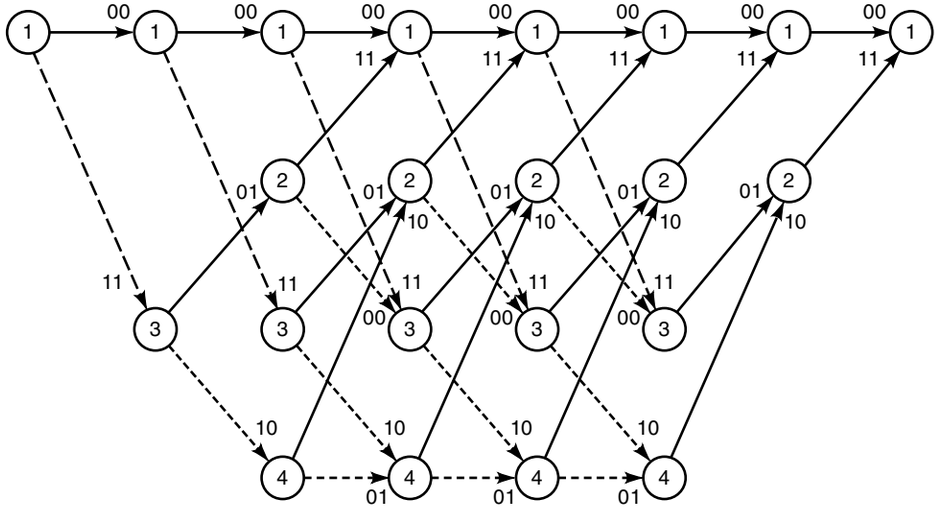


Figure 9. The trellis diagram for the convolutional encoder in Fig. 6.

computationally efficient way for discovering the most likely transmitted sequence for any given received sequence of bits. With reference to the trellis diagram in Fig. 9, suppose that we have received the sequence 11 01 10 01 01 10 11. Starting in state 1 at time $t = 0$, the trellis branches out to states 1 or 2 in time $t = 1$, and from there to all four states 1, 2, 3, 4, in time $t = 2$. At this point there is exactly one unique path to each of the four current possible states staring from state 1. In order to reach state 1 at time $t = 2$, the trellis path indicates the transmitted prefix sequence 00 00 which is at a Hamming distance three from the actual received prefix 11 01. The path reaching state 2 in time $t = 2$ in the trellis diagram similarly corresponds to the transmitted prefix sequence 11 01 which is seen to be at Hamming distance zero from the corresponding prefix of the received sequence. Similarly we can associate Hamming distances 3 and 2 respectively to the paths reaching states 3 and 4 in time $t = 2$ in the trellis diagram.

Now we extend the trellis paths to time $t = 3$. Each state can be reached at time $t = 3$ along two distinct paths. For instance, in order to reach state 1 in time $t = 3$, the encoder could have made a 1 to 1 transition, adding an incremental Hamming distance of one to the previous cumulative value of three; or it could have made the 2 to 1 transition, adding one unit of Hamming weight to the previous value of zero. Thus at time $t = 3$, there are two distinct paths merging at state 1: the state sequence 1–1–1–1 with a cumulative Hamming distance of four from the given received sequence, and the sequence 1–3–2–1 with a cumulative Hamming distance of one. Since the Hamming weights of the paths are incremental, for any path emanating from state 1 at time $t = 3$, the prefix with the lower cumulative distance is clearly the better choice. Thus at this point we discard the path 1–1–1–1 from further consideration and retain the unique *survivor path* 1–3–2–1 in association with state 1 at the current time. Similarly we explore the two contending paths converging at the other three states at this time ($t = 3$) and identify the minimum distance (or maximum likelihood, for the BSC) survivors for each of those states.

The procedure now iterates. At each successive stage, we identify the survivor paths for each state. If the code sequence were infinite, we would have four infinitely long parallel path traces through the trellis in our example. In order to choose one of the four as the final decoded sequence, we require the encoder to "flush out" the data with a sequence of zeroes, two in our example. The last two zeroes in the seven-bit input data to the encoder cause the trellis paths to converge to state 1 or 2 at time $t = 6$ and to state 1 at $t = 7$. By choosing the survivors at these states, we finally have a complete trellis path starting from state 1 at time $t = 0$ and ending in state 1 at time $t = 7$. The output labels of the successive branches along this path gives the decoder's maximum likelihood estimate of the transmitted bits corresponding to the received sequence.

The average bit error rate of the Viterbi decoder, P_b, can be shown to be bounded by an exponential function of the free distance of the code, as below:

$$P_b \approx N_{d_{free}}[2\sqrt{\epsilon(1-\epsilon)}]^{d_{free}} \approx N_{d_{free}}[2\sqrt{\epsilon}]^{d_{free}}$$

This applies to codes that accept one input bit at a time, as in Fig. 6. $N_{d_{free}}$ is the total number of nonzero information bits on all trellis paths of weight d_{free}, and it can in general be found via an extension of the signal flow transfer function method outlined above. The parameter ϵ is the BSC error probability and is assumed to be very small in the above approximation.

The Viterbi algorithm needs to keep track of only one survivor path per state. The number of states, however, is an exponential function of the memory order. For short convolutional codes of modestly sized state space, the Viterbi algo-

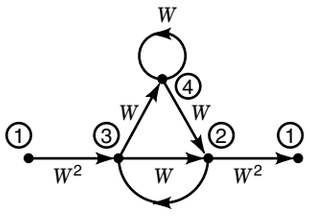


Figure 10. The signal flow graph for the convolutional encoder in Fig. 6.

rithm is an excellent choice for decoder implementation. A memory order of 7 or 8 is typically the maximum feasible. This, in turn, limits the free distance and hence the bit error probability. For long convolutional codes, the survivor path information storage required per state becomes large. In practice, we may choose to retain only some most recent segment of the history of each survivor path. The resulting "truncated" Viterbi algorithm is no longer the theoretically ideal maximum likelihood decoder, though its performance is usually close to the ideal decoder. All these considerations restrict the application of the Viterbi algorithm to short convolutional codes with small constraint lengths. Within these limitations, however, the Viterbi algorithm affords excellent performance.

Sequential Decoding for Convolutional Codes

Tree diagrams lead to one viable strategy for decoding convolutional codes. Given a received sequence of bits (possibly containing errors), the decoder attempts to map it to one path along the tree, proceeding node by node and keeping track of the cumulative Hamming distance of the path from the received sequence. Along a wrong path, the cumulative Hamming distance exceeds a preset threshold after a few nodes, whereupon the decoder backtracks to the previous node and explores another path. The time to decode any given sequence in this scheme is a random variable, but its expected value remains bounded for code rates below a number $R_{\text{comp}} < C$, where R_{comp} is the *computational cutoff rate* and C is the channel capacity. This technique, known as sequential decoding, is an appropriate technique for decoding very long convolutional codes.

A sequential decoder executes a random number of computations to decode a received sequence—unlike the Viterbi decoder, which executes a fixed number of computations per code sequence. This can be a strength or a weakness, depending on the average noise intensity. If the noise level is high, the sequential decoder typically has to explore many false paths before it discovers the correct path. But the Viterbi algorithm produces an output after a fixed number of computations, possibly faster than the sequential decoder. On the other hand, if the noise level is low, the Viterbi algorithm still needs to execute all of its fixed set of computations whereas the sequential decoder will typically land on the right tree path after only a few trials. Also, sequential decoding is preferred in applications where long codes are needed to drive the postdecoding error probability to extremely low values. In such cases, complexity considerations eliminate Viterbi algorithm as a viable choice.

An efficient approach to implementing sequential decoding is the *stack algorithm*. The key idea here is that the previously explored paths and their likelihood metrics can be stored in a stack ordered according to the likelihood metric value, with the most likely path at the top. The topmost path is then extended to the set of branches extending from that node, metrics are recomputed, and the stack is updated with the new information.

ADDITIONAL TOPICS

Burst Noise Channels

In the foregoing discussion, we had mostly assumed the binary symmetric channel model which was the basis for minimum distance decoding. Burst error channels are another important class of transmission channels encountered in practice, both in wireline and wireless links. Errored bit positions tend to cluster together in such channels, making direct application of much of the foregoing error correction codes futile in such cases. We have already mentioned interleaved coding as a practical method for breaking the error clusters into random patterns and then using random-error correcting codes. Also we noted that Reed–Solomon codes have an intrinsic burst error correction capability. In addition, there have been error correction codes specifically developed for burst noise channels. For a detailed treatment of this subject, see, for example, Ref. 8, Chap. 9.

Intersymbol Interference Channels and Precoding

Binary data are transmitted by mapping the 0's and 1's into baseband or radio-frequency (RF) pulses. In a bandwidth-limited channel, the channel response waveform corresponding to one input pulse tends to overlap those of succeeding pulses, if the input pulse rate is high. This *intersymbol interference* (ISI) can be controlled by appropriately shaping the input spectrum by *precoding* the input pulse waveform. By suitably constraining the 0/1 transition patterns, it becomes possible to receive the input bit stream despite the overlap of the pulse response waveforms. This technique has been important in high-speed modem designs for the wireline channel. Because of the recent interest in digital subscriber lines, there has been much activity in this area. We cite Ref. 41 as an example of recent work and for pointers to earlier work in this important area.

Synchronization Codes

Coding techniques described so far implicitly assume synchronization; that is, the decoder knows the exact times when one codeword ends and the next begins, in a stream of binary data. In real life this of course cannot be assumed. Codes that can self-synchronize are therefore important. Key results in this direction is summarized in standard coding theory sources such as Ref. 4. However, the practical use of these synchronization codes does appear to be limited, compared to more advanced timing and synchronization techniques used in modern digital networks.

Soft Decision Decoding

The actual inputs and outputs of the physical channel are analog waveforms. The demodulator processes the noisy waveform output of the physical channel and furnishes a noisy estimate of the currently transmitted bit or symbol. A *hard decision* is made at the demodulator output when a threshold device maps the noisy analog data into a 0 or a 1 (in the binary case). Instead, we can retain the analog value (or a finely quantized version of it) and then make an overall decision about the identity of an entire codeword from these *soft decision* data. Clearly, the soft decision data retain more information, and hence the overall decision made on an entire codeword can be expected to be more reliable than the concatenation of bit-by-bit hard decisions. Analysis and practical implementations have borne out this expectation, and soft decision decoding enables achievement of the same bit error rate with a lower signaling power requirement than that for hard

decision decoding. Many recent text books on digital communications (e.g., Ref. 42) contain details of this approach.

Combined Coding and Modulation

As mentioned earlier, coding and modulation have traditionally developed in mutual isolation. Ungerboeck (43) proposed the idea that redundancy for error correction coding may be embedded into the design of modulation signal constellations, and combined decoding decisions may be based on the Euclidean distance between encoded signal points rather than on Hamming distance. The approach has been found to be capable of significant improvements in the performance of coded communications. For more details on this topic, see Ref. 44 or one of the more recent textbooks in digital communications such as Ref. 42.

Turbo Codes

The discovery of "turbo codes" (45) is perhaps the most spectacular event in coding research in recent times. Turbo codes have made it possible to approach the ultimate Shannon limits for communication much more closely than was previously possible. Turbo codes are essentially a battery of parallel concatenated encoders. The outputs of these component encoders are merged by interleaving, and they are punctured as needed to get the desired code rate. Relatively simple, iterative soft decision decoding methods provide surprisingly superior performance.

APPLICATIONS

Data transmission coding is intrinsically an applications-oriented discipline, its deep mathematical foundations notwithstanding. Early work by Hamming (10) and others were quickly applied to computer data storage and transmission. The minimum-distance decoding approach is ideally suited ro random, independent-error environments such as found in space communications, where coding applications registered some early success (e.g., see Chapter 3 of Ref. 23). The markedly clustered or bursty nature of error patterns in terrestrial wireline and radio channels initially limited coding applications in this arena to only error detection and retransmission (8, Chapter 15), and not forward error correction. Cyclic redundancy check (CRC) codes are cyclic codes used for error detection only, and they are ubiquitous in modern data transmission protocols. Also, more recently, the needs of the high-speed modems created more opportunities for error correction applications. Many specific codes have lately been adopted into international standards for data transmission. Chapters 16 and 17 in Ref. 8 furnish an excellent summary of applications of block and convolutional codes, respectively. Applications of Reed–Solomon codes to various situations are well documented in Ref. 23, including its use in compact disc players, deep space communications, and frequency hop spread spectrum packet communications.

The advent of the broadband integrated service digital network (B-ISDN) has created ample instances of coding applications (46). The asynchronous transfer mode (ATM) adaptation layer 1 (AAL-1) has a standardized option for the use of Reed–Solomon coding for recovery from ATM cell losses (47, p. 75). The recently adopted high-definition television (HDTV) standards use Reed–Solomon coding for delivery of compressed, high-rate digital video (48). Almost all of the recent digital wireless technologies, such as GSM, IS-54 TDMA, IS-95 CDMA, cellular digital packet data (CDPD), and others (49), have found it advantageous to make use of error correction coding to mitigate the excessive noisiness of the wireless channel.

In summary, over the past 50 years following the inception of information theory (1), not only has the art of data transmission codes matured into a variety of applications technologies, but also we are remarkably close to the ultimate theoretical limits of coding performance predicted in Ref. 1.

BIBLIOGRAPHY

1. C. E. Shannon, A mathematical theory of communications, *Bell Syst. Tech. J.,* **27**: 379–423, 623–656, 1948.
2. R. G. Gallager, *Information Theory and Reliable Communication,* New York: Wiley, 1968.
3. E. R. Berlekamp, *Algebraic Coding Theory,* New York: McGraw-Hill, 1989.
4. W. W. Peterson and E. J. Weldon, Jr., *Error Correcting Codes,* 2nd ed., Cambridge, MA: MIT Press, 1972.
5. E. R. Berlekamp, *Key Papers in the Development of Coding Theory,* New York: IEEE Press, 1974.
6. F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes,* Amsterdam, The Netherlands: North-Holland, 1977.
7. R. E. Blahut, *Theory and Practice of Error Control Codes,* Reading, MA: Addison-Wesley, 1983.
8. S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications,* Englewood Cliffs, NJ: Prentice-Hall, 1983.
9. A. M. Michelson and A. H. Levesque, *Error Control Techniques for Digital Communications,* New York: Wiley, 1985.
10. R. W. Hamming, Error detecting and error correcting codes, *Bell Syst. Tech. J.,* **29**: 147–160, 1950.
11. D. E. Slepian, A class of binary signaling alphabets, *Bell Syst. Tech. J.,* **35**: 203–234, 1956.
12. F. J. MacWilliams, A theorem on the distribution of weights in a systematic code, *Bell Syst. Tech. J.,* **42**: 79–94, 1963.
13. E. Prange, Cyclic error-correcting codes in two symbols, *AFCRC-TN-57-103,* Air Force Cambridge Research Center, Cambridge, MA, 1957.
14. R. C. Bose and D. K. Ray-Chaudhuri, On a class of error correcting binary group codes, *Inf. Control,* **3**: 68–79, 1960.
15. A. Hocquenghem, Codes correcteurs d'erreurs, *Chiffres,* **2**: 147–156, 1959, in French.
16. W. W. Peterson, Encoding and decoding procedures for the Bose–Chaudhuri codes, *IRE Trans. Inf. Theory,* **6**: 459–470, 1960.
17. D. C. Gorenstein and N. Zierler, A class of error-correcting codes in p^m symbols, *J. Soc. Ind. Appl. Math (SIAM),* **9**: 207–214, 1961.
18. R. T. Chien, Cyclic decoding procedure for the BCH codes, *IEEE Trans. Inf. Theory,* **10**: 357–363, 1964.
19. E. R. Berlekamp, On decoding binary BCH codes, *IEEE Trans. Inf. Theory,* **11**: 577–580, 1965.
20. J. L. Massey, Shift register synthesis and BCH decoding, *IEEE Trans. Inf. Theory,* **15**: 122–127, 1969.
21. R. E. Blahut, Transform techniques for error control codes, *IBM J. Res. Develop.,* **23**: 299–315, 1979.
22. I. S. Reed and G. Solomon, Polynomial codes over certain finite fields, *J. Soc. Ind. Appl. Math. (SIAM),* **8**: 300–304, 1960.

23. S. B. Wicker and V. K. Bhargava, *Reed–Solomon Codes and Their Applications,* Piscataway, NJ: IEEE Press, 1994.
24. A. Tietvainen, A short proof for the nonexistence of unknown perfect codes over GF(q), $q > 2$, *Ann. Acad. Sci. Fenn. A,* **580**: 1–6, 1974.
25. M. J. E. Golay, Binary coding, *IRE Trans. Inf. Theory,* **4**: 23–28, 1954.
26. P. Elias, Error-free coding, *IRE Trans. Inf. Theory,* **4**: 29–37, 1954.
27. H. O. Burton and E. J. Weldon, Jr., Cyclic product codes, *IRE Trans. Inf. Theory,* **11**: 433–440, 1965.
28. G. D. Forney, Jr., *Concatenated Codes,* Cambridge, MA: MIT Press, 1966.
29. J. Justesen, A class of constructive asymptotically algebraic codes, *IEEE Trans. Inf. Theory,* **18**: 652–656, 1972.
30. E. N. Gilbert, A comparison of signaling alphabets, *Bell Syst. Tech. J.,* **31**: 504–522, 1952.
31. M. A. Tsfasman, S. G. Vladut, and T. Zink, Modular curves, Shimura curves and Goppa codes which are better than the Varsharmov–Gilbert bound, *Math. Nachr.,* **109**: 21–28, 1982.
32. R. J. McEliece et al., New upper bounds on the rate of a code via the Delsarte–MacWilliams inequalities, *IEEE Trans. Inf. Theory,* **23**: 157–166, 1977.
33. T. Verhoeff, An updated table of minimum-distance bounds for binary linear codes, *IEEE Trans. Inf. Theory,* **33**: 665–680, 1987.
34. P. Elias, Coding for noisy channels, *IRE Conv. Rec.,* **4**: 37–47, 1955.
35. R. M. Fano, A heuristic discussion of probabilistic decoding, *IEEE Trans. Inf. Theory,* **9**: 64–74, 1963.
36. J. M. Wozencraft and B. Reiffan, *Sequential Decoding,* Cambridge, MA: MIT Press, 1961.
37. J. L. Massey, *Threshold Decoding,* Cambridge, MA: MIT Press, 1963.
38. A. J. Viterbi, Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, *IEEE Trans. Inf. Theory,* **13**: 260–269, 1967.
39. G. D. Forney, Jr., Convolutional codes I: Algebraic structure, *IEEE Trans. Inf. Theory,* **16**: 720–738, 1970.
40. S. J. Mason, Feedback theory—Further properties of signal flow graphs, *Proc. IRE,* **44**: 920–926, 1956.
41. R. F. H. Fischer and J. B. Huber, Comparison of precoding schemes for digital subscriber lines, *IEEE Trans. Commun.,* **45**: 334–343, 1997.
42. S. G. Wilson, *Digital Modulation and Coding,* Upper Saddle River, NJ: Prentice-Hall, 1996.
43. G. Ungerboeck, Channel coding with amplitude/phase modulation, *IEEE Trans. Inf. Theory,* **28**: 55–67, 1982.
44. G. Ungerboeck, *IEEE Commun. Mag.,* **25** (2): 12–21, 1987.
45. C. Berrou and A. Glavieux, Near-optimum error correcting coding and decoding: Turbo codes, *IEEE Trans. Commun.,* **44**: 1261–1271, 1996.
46. E. Ayanoglu, R. D. Gitlin, and N. C. Oguz, Performance improvement in broadband networks using a forward error correction for lost packet recovery, *J. High Speed Netw.,* **2**: 287–304, 1993.
47. C. Partridge, *Gigabit Networks,* Reading, MA: Addison-Wesley, 1994.
48. T. S. Rzeszewski, *Digital Video: Concepts and Applications across Industries,* Piscataway, NJ: IEEE Press, 1995.
49. T. S. Rappaport, *Wireless Communications: Principles and Practice,* Upper Saddle River, NJ: Prentice-Hall, 1996.

GEORGE THOMAS
University of Southwestern
Louisiana

} { { } }

Wiley Encyclopedia of Electrical and Electronics Engineering

Information Theory of Modulation Codes and Waveforms
Standard Article

Costas N. Georghiades[1]
[1]Texas A&M University

DOI: 10.1002/047134608X.W4207.pub2
Article Online Posting Date: August 17, 2007

Abstract | Full Text: HTML PDF (794K)

Abstract

The fundamental problem of communication is the conveying of information (which may take several different forms) from a generating source through a communication medium to a desired destination. This conveyance of information, invariably, is achieved by transmitting signals that contain the desired information in some form and that efficiently carry the information through the communication medium. We refer to the process of superimposing an information signal onto another for efficient transmission as modulation.

Keywords: analog modulation; digitally modulated signal; binary modulation; baseband pulse-amplitude modulation; quadrature amplitude modulation; continuous-phase modulation

INFORMATION THEORY OF MODULATION CODES AND WAVEFORMS

INTRODUCTION

The fundamental problem of communication is the conveying of information (which may take several different forms) from a generating source through a communication medium to a desired destination. This conveyance of information, invariably, is achieved by transmitting signals that contain the desired information in some form and that efficiently carry the information through the communication medium. We refer to the process of superimposing an information signal onto another for efficient transmission as *modulation*.

Several factors dictate modulating the desired information signal into another signal more suitable for transmission. The following factors affect the choice of modulation signals:

1. The need to use signals that efficiently propagate through the communication medium at hand. For example, if the communication medium is the atmosphere (or free space), one might use a radio frequency (RF) signal at some appropriate frequency, whereas for underwater communications, one might use an acoustical signal.
2. Communication media invariably distort stochastically signals transmitted through them, which makes information extraction at the receiver nonperfect and most often nonperfect. Thus, a need exists to design modulation signals that are robust to the stochastic (and other) effects of the channel, to minimize its deleterious effects on information extraction.
3. It is highly desirable that communication systems convey large amounts of information per unit time. The price we pay in increasing the in formation rate is often an increase in the required transmitted signal bandwidth. We are interested in modulation signals that can accommodate large information rates at as small a required bandwidth as possible.
4. The power requirements (i.e., average power and peak power) of the transmitted signals to achieve a certain level of performance in the presence of noise introduced during transmission are of paramount importance, especially in power-limited scenarios, such as portable radio and deepspace communications. Our preference is for signals that require as little power as possible for a desired performance level.

The problem of designing modulation signals that possibly optimize some aspect of performance, or satisfy some constraints imposed by the communication medium or the hardware, is known generally as signal design. Signal design problems are important and widely prevalent in communications.

Currently, a proliferation of products make use of modulation to transmit information efficiently. Perhaps the most prevalent and oldest examples are commercial broadcast stations that use frequency modulation (FM) or amplitude modulation (AM) to transmit audio signals through the atmosphere. Another example are data modems that are used to transmit and receive data through telephone lines. These two examples have obvious similarities but also some very important differences. In the broadcast station example, the information to be communicated (an audio signal) is analog and is used to directly modulate a radio-frequency (RF) carrier, which is an example of *analog modulation.* On the other hand, the data communicated through a modem come from the serial port of a computer and are discrete (in fact they are binary; that is, they take two possible values, "0" or "1"), which results in a *digitally modulated* signal. Clearly, the difference between analog and digital modulation is not in the nature of the transmitted signals, because the modulation signals are analog in both cases. Rather, the difference is in the nature of the set of possible modulation signals, which is discrete (and in fact finite) for digitally modulated signals and infinitely uncountable for analog modulation.

The simplest possible digital modulation system consists of two modulation signals. One signal corresponds to the transmission of a "0" and the other of a "1," which is called *binary modulation.* Binary digits (bits) are communicated using binary modulation by assigning a signal in a one-to-one correspondence to each of the two possible logical values of a bit. This mapping between bits and signals is done at a rate equal to the bit rate (i.e., the number of bits/second arriving at the input of the modulator). In response to each transmitted modulation signal, the channel produces a received signal at its output, which is a randomly distorted replica of the transmitted signal. To extract the information superimposed on the modulation signals, a processor, called a receiver or a detector, processes the noisy signal received. The function of the detector is to decide which of the two (in this case) possible signals was transmitted, and in doing so correctly, it recovers the correct value for the transmitted bit. Because of the presence of stochastic noise in the received signal, the receiver may make an incorrect decision for some transmitted bits. The probability of making a decision error in extracting the transmitted bits is known as the bit-error probability or the bit-error rate (BER). The performance of communication systems using digital modulation is invariably measured by their achieved BER, as a function of the transmitted energy per information bit. Receivers that achieve the smallest possible BER for a given channel and modulation signal set are called optimal.

Binary modulation systems are the simplest to implement and detect, but they are not necessarily the most efficient in communicating information. Modulators with larger signal sets use a smaller bandwidth to transmit a given information bit rate. For example, one can envision having a modulation signal set containing four (instead of two) signals: $s_1(t)$, $s_2(t)$, $s_3(t)$, $s_4(t)$. With four signals, we can assign to each a two-bit sequence in a one-to-one cor-

respondence, for example, as follows:

$$s_1(t) \Leftrightarrow 00$$
$$s_2(t) \Leftrightarrow 01$$
$$s_3(t) \Leftrightarrow 10$$
$$s_4(t) \Leftrightarrow 11$$

In this case, each time a transmitted signal is detected correctly, the receiver extracts two (correct) bits. The bit rate has also doubled compared with a binary modulator for the same signaling rate (transmitted signals per second). Because bandwidth is proportional to the signaling rate, we have effectively doubled our transmission efficiency using a modulator with four signals instead of two. Of course, the job of the receiver is now harder because it has to make a four-way decision, instead of just a binary decision, and everything else being the same, the probability of making an erroneous decision increases. We refer to the above modulator as a 4-ary modulator (or a quaternary modulator).

Clearly, the idea can be extended to modulation signal sets that contain $M = 2^k$ signals, for some integer $k = 1, 2, 3 \cdots$. In this case, each transmitted signal carries k bits. We refer to modulators that use M signals as M-ary modulators. As in the 4-ary modulator example above, the advantage of a larger number of modulation signals is that the number of signals per second that needs to be transmitted to accommodate a certain number of bits per second decreases as M increases. Because the number of signals per second determines to a large extent the bandwidth required, more signals means a smaller required bandwidth for a given number of transmitted bits per second, which is a desirable result. The price paid for large signal sets is in complexity and, as previously pointed out, in possibly reduced performance for the same expended average energy per bit.

Although many analog modulation (communication) systems are still in use, the trend is for systems to become digital. Currently, two prominent examples of analog systems becoming digital are cellular phones and digital TV broadcasts. Digital modulation techniques are by far the more attractive.

ANALOG MODULATION

The most prevalent medium for everyday communication is through RF (sinusoidal) carriers. Three quantities exist whose knowledge determines exactly the shape of an RF signal: (1)its amplitude; (2)its phase; and (3)its frequency, as indicated in equation 1:

$$s(t) = A(t)\cos[2\pi\ f_c t + \phi(t)] \tag{1}$$

where f_c is the frequency of the sinusoidal signal in Hertz. Information can be conveyed by modulating the amplitude, the instantaneous frequency, or the phase of the carrier (or combinations of the three quantities).

Amplitude Modulation

Let the information signal *m(t)* be baseband and bandlimited to some bandwidth W Hz. A baseband signal bandlimited to W Hz has a frequency spectrum centered at the origin and contains substantially no energy above W HZ. We assume, which is a good assumption in practice, that $W << f_c$. In amplitude modulation (AM), the information signal modulates the amplitude of the carrier according to:

$$u(t) = Am(t)\cos(2\pi\ f_c t + \phi) \tag{2}$$

where ϕ is some fixed carrier phase. Insight into the process of modulation is obtained by looking at the Fourier transform of the modulated signal, given by (see, for example, Reference (1))

$$U(\ f) = \frac{A}{2}[M(\ f - \ f_c)e^{j\phi} + M(\ f + \ f_c)e^{-j\phi}] \tag{3}$$

Figure 1plots the magnitude of the Fourier transform of the modulated signal for a simple choice (for presentation purposes) of the Fourier transform of the information signal. It is easy to see that, whereas the information signal has bandwidth W, the modulated signal has a bandwidth of $2W$. Also, as can be observed from equation 3, no unmodulated carrier component exists, which would be manifested as delta functions at the carrier frequency f_c. We refer to this scheme as double-sideband, suppressed-carrier (DSB-SC), amplitude modulation.

Demodulation of DSB-SC amplitude modulated signals can be achieved by multiplying the received signal by a locally generated replica of the carrier, which is generated by a local oscillator (LO). For best performance, the locally generated carrier must match as closely as possible the frequency and phase of the received carrier. It is usually reasonable to assume the receiver generated carrier frequency matches the carrier frequency in the received signal well.[1] Neglecting noise, for simplicity, and assuming perfect frequency synchronization, the demodulator is described mathematically by

$$\begin{aligned} z(t) &= u(t)\cos(2\pi\ f_c t + \hat{\phi}) \\ &= \frac{A}{2}\, m(t)\cos(\phi - \hat{\phi}) + \frac{A}{2}\, m(t)\cos(4\pi\ f_c t + \phi + \hat{\phi}) \end{aligned} \tag{4}$$

where $\hat{\phi}$ is the phase of the locally generated carrier. Now the component in equation 4at twice the carrier frequency is easily filtered out by low-pass filtering to yield

$$\hat{m}(t) = \frac{A}{2} m(t)\cos(\phi - \hat{\phi}) \tag{5}$$

which is a scaled version of the modulation signal. In the presence of noise, to maximize the signal-to-noise ratio (SNR), it is important that the phase error $(\phi - \hat{\phi})$ be small. The problem of phase synchronization is an important one and is often practically achieved using a phase-locked loop (PLL) (see, for example, References (2)–(5).) When the locally generated carrier is perfectly phase and frequency locked to the phase and frequency of the received signal, detection of the information is referred to as coherent. This is in contrast to noncoherent detection, when the phase of the locally generated carrier does not match that of the received signal. Clearly, coherent detection achieves the ultimate limit in performance. It can be approached in practice by using sophisticated algorithms, at the cost of increased complexity.

A simpler, noncoherent, detector can be used if the transmitted carrier contains an unmodulated component (or a "pilot tone") resulting in what is referred to as DSB mod-

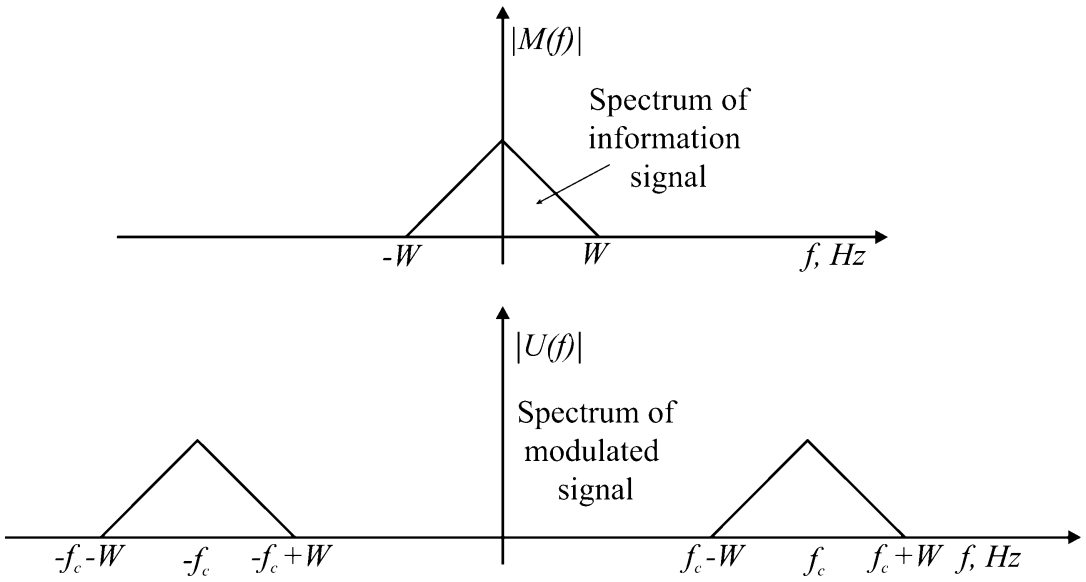


Figure 1. The magnitude of the Fourier transform of a DSB-SC amplitude-modulated signal. (Figure is not to scale).

ulation. In conventional AM (such as in broadcasting), the modulated signal takes the form

$$u(t) = A[1 + am(t)]\cos(2\pi\ f_c t + \phi)$$

with the constraint that $|m(t)| \leq 1$; $a,\ 0 \leq a \leq 1$, is the modulation index. Figure 2shows an example of a conventionally modulated AM signal. Clearly, the modulated signal for conventional AM has a strong unmodulated component at the carrier frequency that carries no information but uses power, and thus, a significant power penalty exists in using it. The benefit resulting from the reduced power efficiency is that simple receivers can now be used to detect the signal. The loss in power efficiency can be justified in broadcasting, where conventional AM is used, because in this case only one high-power transmitter draws power from the power grid, with millions of (now simpler and therefore less costly) receivers.

Demodulation of AM Signals

The most popular detection method for conventional AM is envelope detection. This method consists of passing the received modulated signal [usually after RF amplification and down conversion to some intermediate frequency (IF)] through a rectifier followed by a simple low-pass filter (in the form of a simple, passive, RC circuit). This simple detector is shown in Fig. 3.

Double-sideband amplitude modulation is wasteful in bandwidth, requiring a bandwidth that is twice the baseband signal bandwidth. It can be shown that the two sidebands are redundant, and that the information signal can be obtained if only one sideband was transmitted, which reduces the required bandwidth by a factor of two compared with DSB-AM. At the same time, an improvement in power efficiency, occurs because transmitting the redundant sideband requires not only extra bandwidth but also extra power. When only one sideband is transmitted, the resulting signal is referred to as single sideband (SSB). The general form of a single-sideband signal is

$$u(t) = A[m(t)\cos(2\pi\ f_c t) \pm \hat{m}(t)\sin(2\pi\ f_c t)] \tag{6}$$

where $\hat{m}(t)$ is the Hilbert transform of $m(t)$ given by

$$\hat{m}t = m(t) * \frac{1}{\pi t} \Leftrightarrow \hat{M}(\ f) = M(\ f)H(\ f)$$

where *H(f)* is the Fourier transform of $h(t) = 1/\pi t$ and is given by

$$H(\ f) = \begin{cases} -j, & f > 0 \\ j, & f < 0 \\ 0, & f = 0, \end{cases}$$

In equation 6 , the plus or minus sign determines whether the upper or the lower sideband is chosen. Figure 4shows the spectrum of an upper sideband SSB signal. For a more complete exposition to SSB, including modulation and demodulation methods, consult References (1) and (6–9).

Another amplitude modulation scheme, widely used in TV broadcasting, is vestigial sideband (VSB). The reader is referred to References (1) and (6–9) for more information.

Angle Modulation

Angle modulation of a sinusoidal carrier includes phase modulation (PM) and frequency modulation (FM). In phase modulation, the information signal modulates the instantaneous phase of a high-frequency sinusoidal carrier, whereas in frequency modulation, the information signal directly modulates the instantaneous frequency of the carrier. As the instantaneous frequency and phase of a signal are simply related (the instantaneous frequency is the scaled derivative of the instantaneous phase), clearly PM and FM are also closely related and have similar properties. For angle modulation, the modulated signal is given by

$$u(t) = A\cos[2\pi\ f_c t + \phi(t)]$$

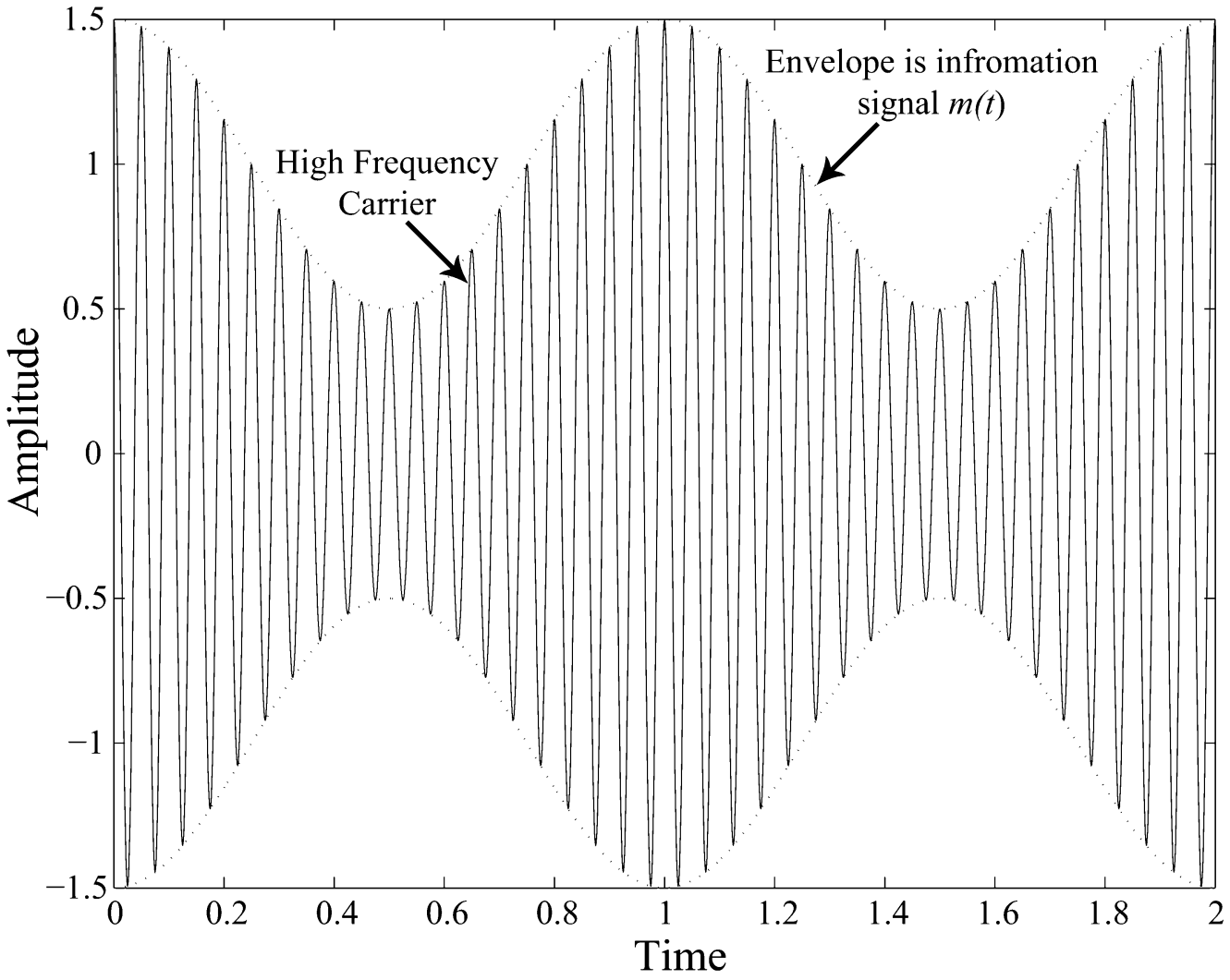


Figure 2. Illustration of a conventionally amplitude-modulated signal.

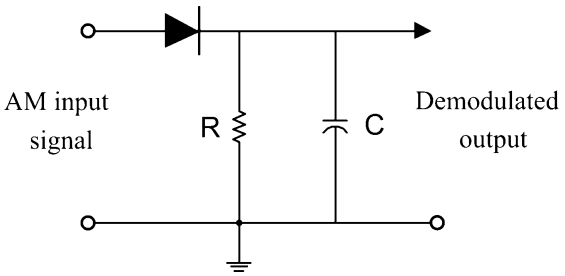


Figure 3. A simple demodulator for conventional AM signals.

where

$$\phi(t) = \begin{cases} d_p m(t) & \text{PM} \\ 2\pi d_f \displaystyle\int_{-\infty}^{t} m(\tau)d\tau & \text{FM} \end{cases}$$

The constants d_p and d_f are the *phase* and *frequency deviation* constants, respectively. These constants, along with the peak amplitude of the information signal, define the peak phase deviation and peak frequency deviation constants, given by

$$\Delta\phi = d_p \cdot |m(t)|$$

and

$$\Delta f = d_f \cdot |m(t)|$$

In turn, the peak deviation constants define the phase and frequency modulation indices according to

$$\beta_p = \Delta\phi$$

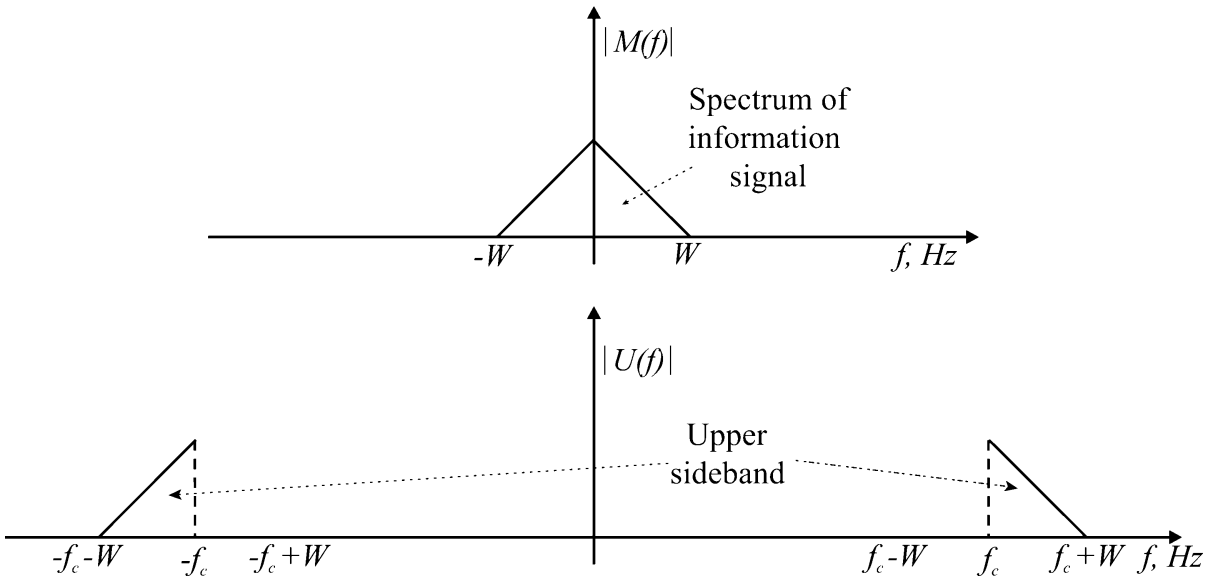


Figure 4. The spectrum of an upper sideband SSB signal.

and

$$\beta_f = \frac{\Delta_f}{W}$$

where W is the bandwidth of the information signal signal $m(t)$. As an example, the peak frequency deviation for FM broadcasts is 75 KHz, and the signal bandwidth is limited to 15 KHz, which yields a modulation index of 5. For illustration, Fig. 5shows typical waveforms for frequency and phase modulation.

The spectrum of an angle-modulated signal is much more difficult to obtain mathematically than in the AM case because angle modulation is nonlinear. Moreover, strictly speaking, angle-modulated signals have an infinite bandwidth. However, an approximation for the effective bandwidth (i.e., the frequency band containing most of the signal energy) of angle-modulated signals is given by Carson's rule:

$$B = 2(\beta + 1)W$$

where β is the phase- or frequency-modulation index and W is the bandwidth of the information signal. The bandwidth of the modulated signal increases linearly as the modulation index increases. FM systems with a small modulation index are called narrowband FM, whereas systems with a large modulation index are called wideband FM. One popular and practical way to generate wideband FM is to first generate a narrowband FM signal (which is easily generated) and then, through frequency multiplication, to convert it into a wideband FM signal at an appropriate carrier frequency. Wideband FM is used in broadcasting, and narrowband FM is used in point-to-point FM radios.

Detection of FM or PM signals takes several different forms, including (PLLs) and discriminators, which convert FM into AM that is then detected as such. For more information on ways to modulate and demodulate angle modulated signals, consult References (1,3,5), and (9).

DIGITAL MODULATION

A wide variety of digital modulation methods exists, depending on the communication medium and the mode of communication, both of which impose constraints on the nature of transmitted signals. For example, for optical systems that use an optical carrier [generated by a light-emitting diode (LED) or a laser], various modulation schemes are particularly suitable, which may not be suitable for RF communications systems. Similarly, modulation schemes used in magnetic recording systems may not be suitable for other systems. Generally, as indicated in the Introduction, the modulation must be matched to the channel under consideration.

Signal Space

In designing and describing digital modulation schemes, it is often desirable to consider modulation signals as points in some appropriate signal space, spanned by a set of orthonormal-basis signals. The dimensionality of the signal space equals the number of orthonormal-basis signals that span it.

A set of signals $\{\phi_1(t), \phi_2(t), \cdots, \phi_N(t)\}$, for $0 \le t \le T$ is orthonormal if the following condition holds:

$$\int_0^T \phi_i(t)\phi_j(t)dt = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

If $s(t)$ is any signal in the N-dimensional space spanned by these signals, then it can be expressed as

$$s(t) = \sum_{i=1}^{N} s_i \phi_i(t)$$

for some set or real numbers $s_1, s_2, \cdot, s_N$. The N coefficients uniquely describing $s(t)$ are obtained using

$$s_k = \int_0^T s(t)\phi_k(t)dt, \;\; k = 1, 2, \cdots, N$$

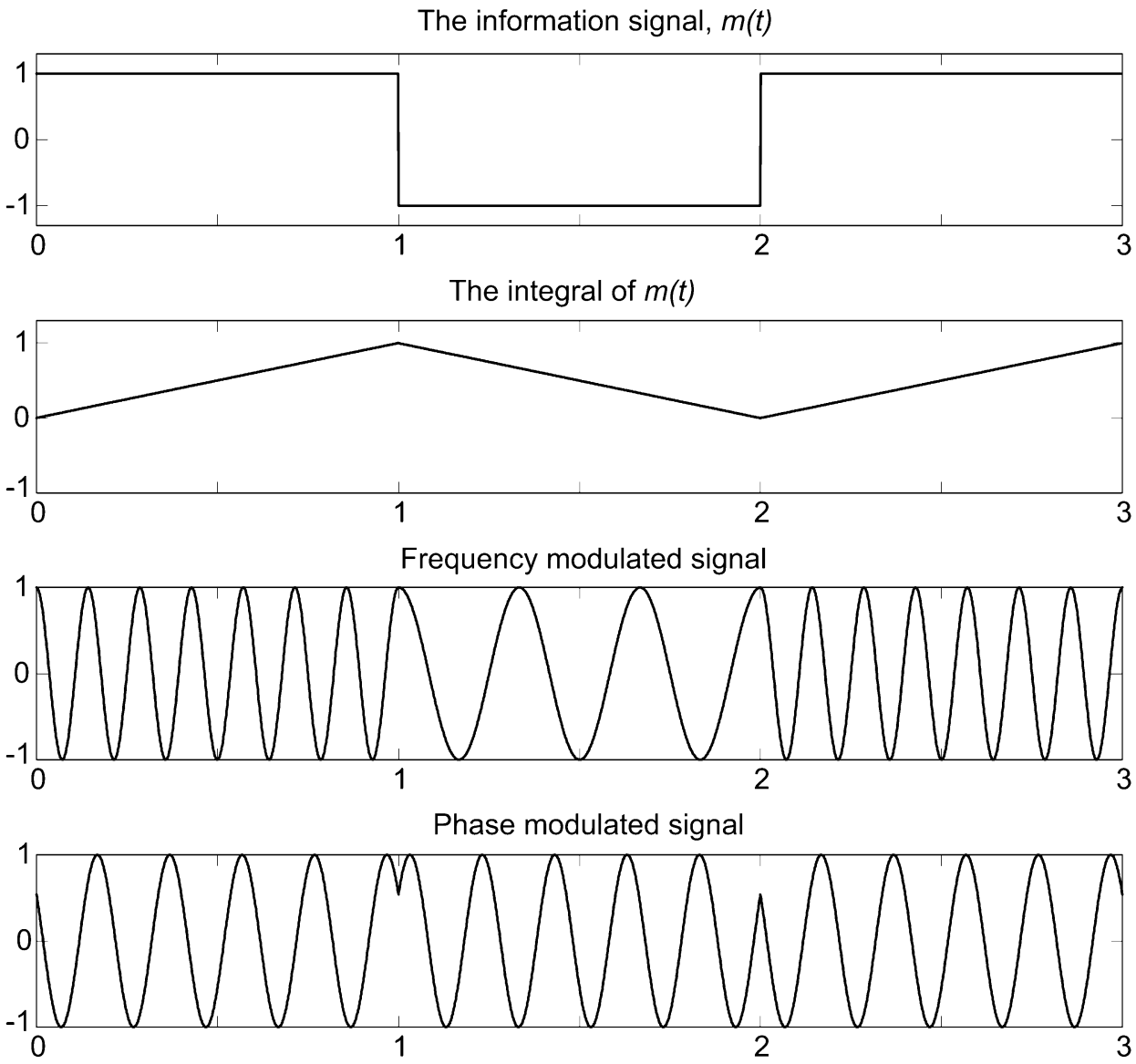


Figure 5. Illustration of frequency- and phase-modulated signals.

Figure 6illustrates the concept of signal space for the special case of two dimensions. In the figure, four distinct signals are represented as points in the signal space.

Perhaps the most widely known and used modulation schemes are those pertaining to RF communication, some of which are examined next.

Phase-Shift Keying

Under phase-shift keying (PSK), the information bits determine the phase of a carrier, which takes values from a discrete set in accordance with the information bits. The general form of M-ary PSK signals (i.e., a PSK signal set containing signals) is given by

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos(2\pi\ f_c t + \theta_i), \quad i = 1, 2, \cdots, M, \quad 0 \le t \le T \tag{7}$$

where

$$\theta_i = \frac{2\pi(i-1)}{M}$$

and

$$E = \int_0^T s_i^2(t)dt$$

is the *signal energy*. Equation (7)is rewritten in a slightly different form as

$$\begin{aligned} s_i(t) &= \sqrt{E}[\cos(\theta_i)\sqrt{\frac{2}{T}}\cos(2\pi\ f_c t) - \sin(\theta_i)\sqrt{\frac{2}{T}}\sin(2\pi\ f_c t)] \\ &= \sqrt{E}[\cos(\theta_i)\phi_1(t) - \sin(\theta_i)\phi_2(t)] \end{aligned}$$

where $\phi_1(t)$ and $\phi_2(t)$ are easily observed to be orthonormal. Thus, PSK signals are points in a two-dimensional space spanned by $\phi_1(t)$ and $\phi_2(t)$. Figure 7illustrates various PSK signal constellations, including binary PSK (BPSK) and 4-

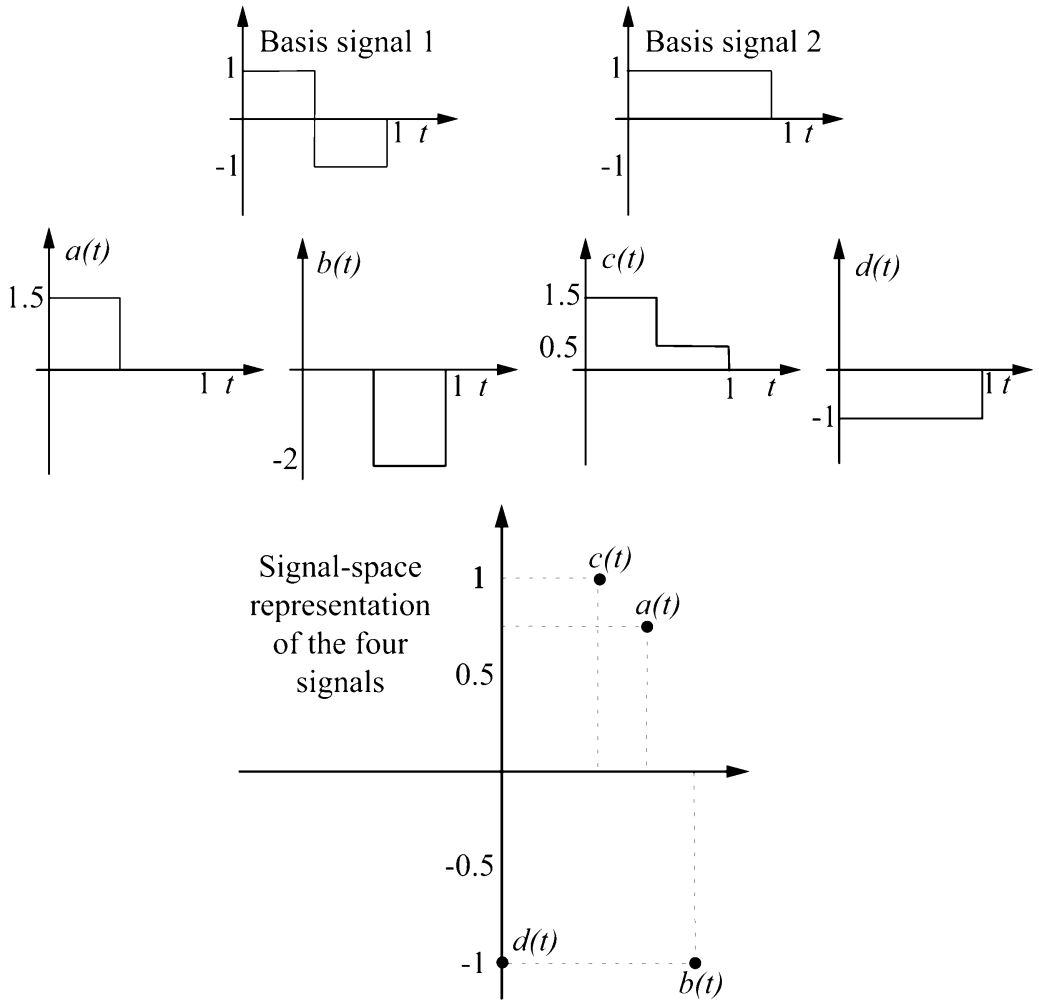


Figure 6. Illustration of the concept of signal space. The two signals on top are the basis signals. Signals *a(t)*, *b(t)*, *c(t)*, and *d(t)* are represented in signal space as points in the two-dimensional space spanned by the two basis signals.

ary PSK, also known as quadrature PSK (QPSK). The figure also illustrates the mapping of information bits to each signal in the constellation. The illustrated mapping, known as Gray coding, has the property that adjacent signals are assigned binary sequences that differ in only one bit. This property is desirable in practice, because, when a detection error is made, it is more likely to be to a signal adjacent to the transmitted signal. Then Gray coding results in a single bit error for the most likely signal errors.

Performance in Additive Gaussian Noise. The simplest channel for data transmission is the additive, white, Gaussian noise (AWGN) channel. For this channel, the transmitted signal is corrupted by and additive Gaussian process, resulting in a received signal given by

$$r(t) = s_i(t) + n(t), \quad 0 \le t \le T \tag{8}$$

where $n(t)$ is zero-mean, white Gaussian noise of spectral density $N_0/2$.

For PSK signals, the optimum receiver (detector), also known as a maximum-likelihood (ML) receiver, decides which of the M possible PSK signals was transmitted by finding the modulation signal that maximizes

$$l_1 = \int_0^T r(t)s_i(t)dt$$

This signal is the well-known correlation receiver, where the most likely signal transmitted is chosen as the one most correlated with the received signal. The correlation receiver involves a multiplication operation, followed by integration. Because processing is linear, it is possible to obtain the same result by passing the received signal through a linear filter with an appropriate impulse response and sampling it at an appropriate instant. The impulse response $h_i(t)$ of the linear filter is easily derived as

$$h_i(t) = s_i(T - t)$$

This linear filter implementation of the optimum receiver is called a matched-filter receiver.

For binary PSK, the probability that the optimal receiver makes a decision error is given by

$$P_{\text{BPSK}}(e) = \frac{1}{2}\text{erfc}(\sqrt{\frac{E}{N_0}}) \tag{9}$$

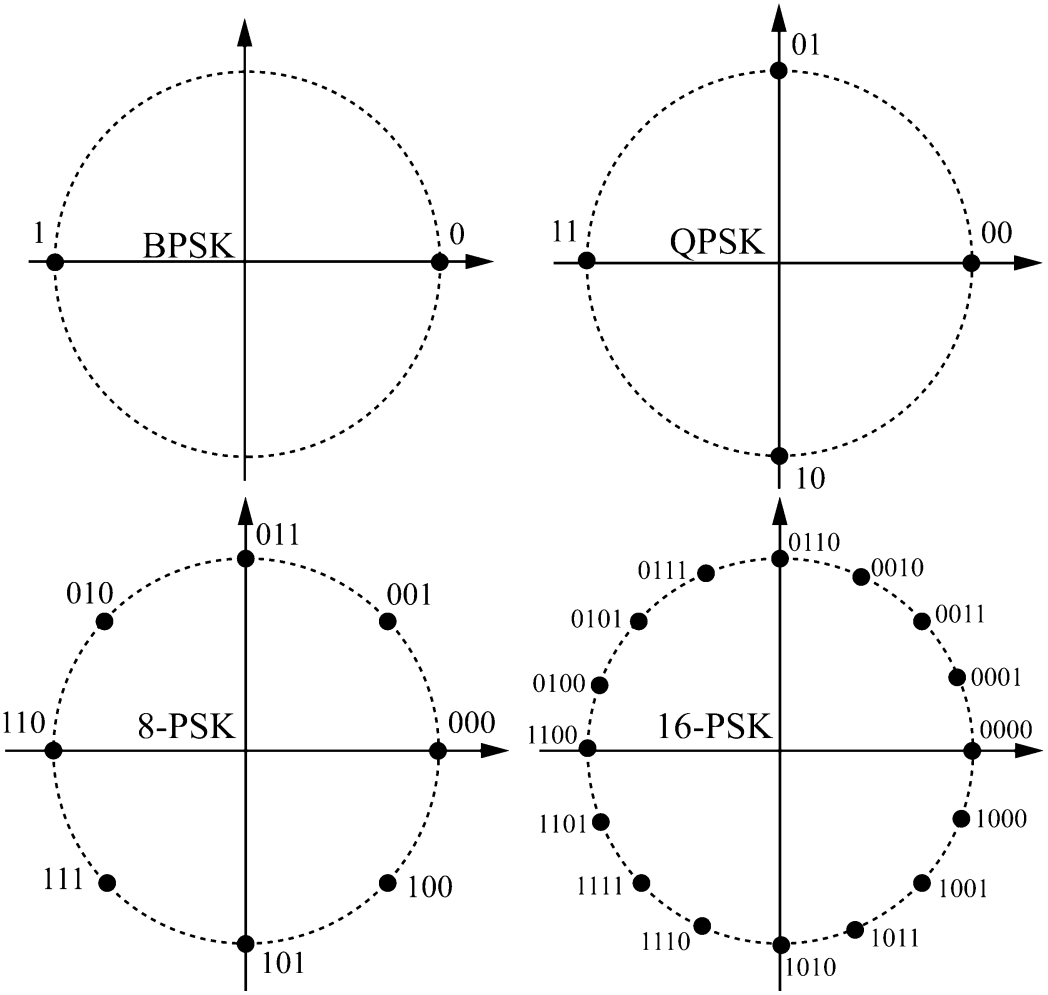


Figure 7. Signal space representation of various PSK constellations. The bit assignments correspond to Gray coding.

where

$$\mathrm{erfc}(x) = 1 - \frac{2}{\sqrt{\pi}} \int_0^x e^{-y^2} dy$$

is the complimentary error-function. In equation 9 , the ratio E/N_0 is the SNR, which determines performance. The performance of QPSK is also derived easily and is given by

$$P_{\mathrm{QPSK}}(e) = P_{\mathrm{BPSK}}(e)[2 - P_{\mathrm{BPSK}}(e)]$$

where $\mathrm{P_{BPSK}(e)}$ is as given in equation 9 . An exact expression for the error probability of larger PSK constellations also exists and is found, for example, in Chapter 9 of Reference (1). Figure 8shows the error probability of various PSK constellations as a function of the SNR per information bit.

Baseband Pulse-Amplitude Modulation

Pulse-amplitude modulation (PAM) is the digital equivalent of AM. The difference is that now only discrete amplitudes are allowed for transmission. M-ary PAM is a one-dimensional signaling scheme described mathematically by

$$s_i(t) = (2i - 1 - M)\sqrt{E}\, p(t), \quad i = 1, 2, \cdots, M, \quad 0 \le t \le T$$

where *p(t)* is a unit-energy baseband pulse. Figure 9shows the signal-space representation of PAM signals assuming $E = 1$. In contrast to PSK signals, clearly not every signal has the same energy; in which case, the constellation is described by its average energy:

$$E_{\mathrm{av}} = \frac{E}{M} \sum_{i=1}^{M} (2i - 1 - M)^2 = (\frac{M^2 - 1}{3})E$$

Performance in Additive Gaussian Noise. Based on the data *r(t)* received (as given in equation 8), the maximum-likelihood receiver for PAM signaling chooses as the most likely signal transmitted the signal that maximizes

$$l_i = (2i - 1 - M) \cdot r - \frac{\sqrt{E}}{2}(2i - 1 - M)^2$$

where

$$r = \int_0^T r(t)\, p(t) dt$$

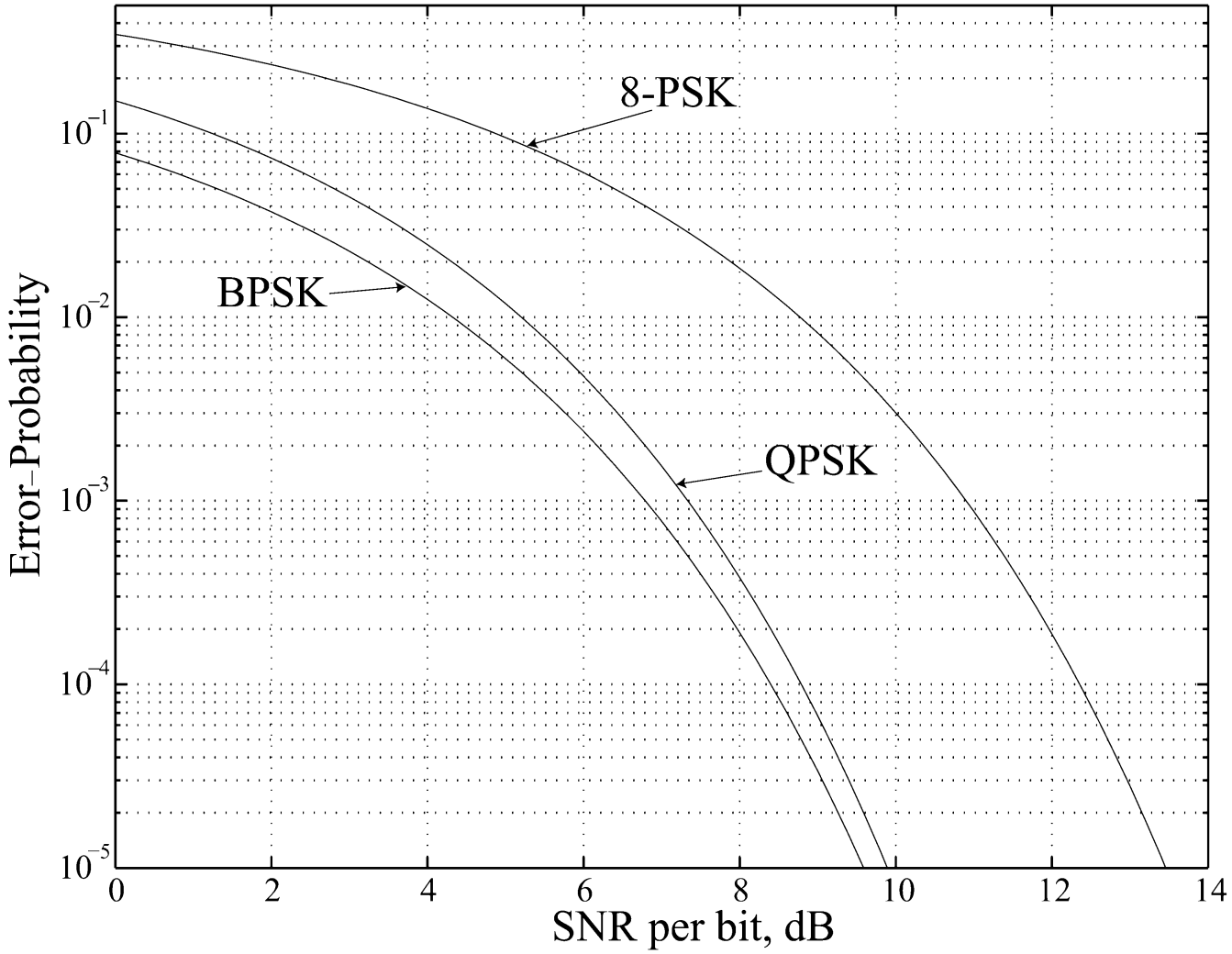


Figure 8. Symbol error probability for BPSK, QPSK, and 8-PSK as a function of the SNR per bit.

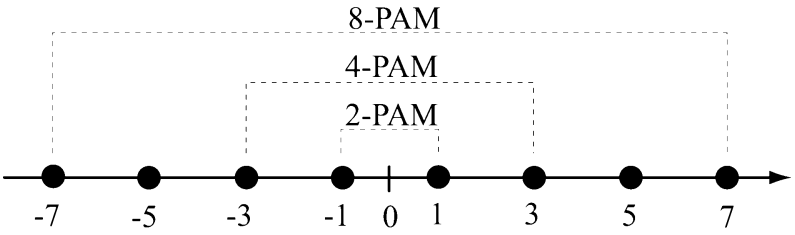


Figure 9. The signal space representation of various PAM constellations.

In signal space, the decision boundaries for this receiver are midway between constellation points, and a decision is made accordingly, based on where r falls on the real line. The error probability for M-ary PAM signals is given by

$$\mathrm{P_{PAM}(e)} = \frac{(M-1)}{M}\mathrm{erfc}\left(\sqrt{\frac{3}{M^2-1}\frac{E_{\mathrm{av}}}{N_0}}\right)$$

The error probability for various PAM constellations is shown in Fig. 10as a function of SNR per bit.

Quadrature Amplitude-Modulation

Quadrature amplitude modulation (QAM) is a popular scheme for high-rate, high-bandwidth efficiency systems. QAM is a combination of both amplitude and phase modulation. Mathematically, M-ary QAM is described by

$$s_i(t) = \sqrt{E}\, p(t)[A_i \cos(2\pi\ f_c t) + B_i \sin(2\pi\ f_c t)], \quad 0 \le t \le T,$$
$$i = 1, 2, \cdots, M$$

where A_i and B_i take values from the set $\{\pm 1, \pm 3, \pm 5, \cdots\}$ and E and $p(t)$ are as defined earlier. The signal space rep-

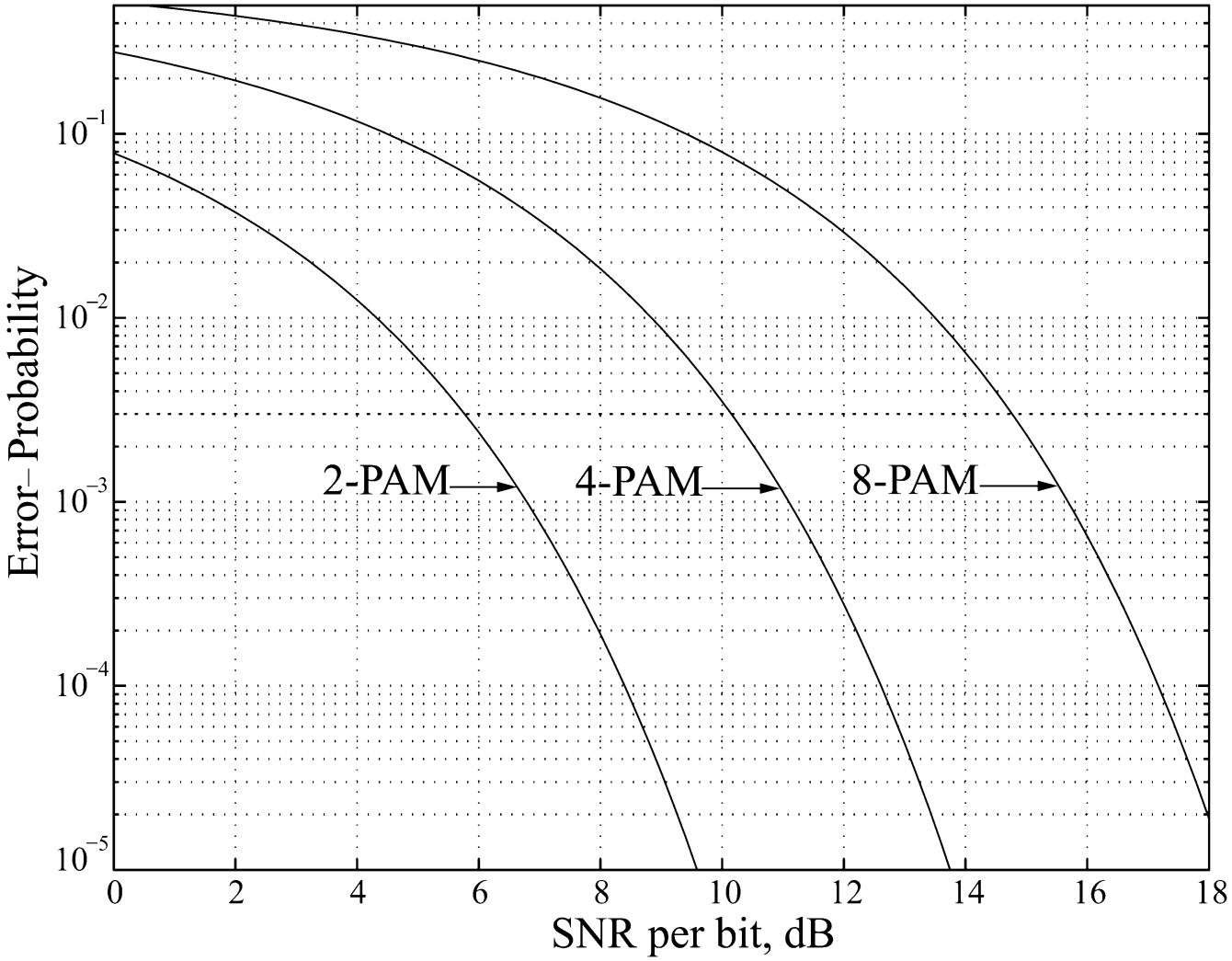


Figure 10. Symbol error probability for 2-, 4-, and 8-PAM as a function of SNR per bit.

resentation of QAM signals is shown in Fig. 11for various values of M, which are powers of 2; that is, $M = 2^k,\ k = 2, 3, \cdots$. For even values of k, the constellations are *square*, whereas for odd values, the constellations have a cross shape and are thus called *cross* constellations. For square constellations, QAM corresponds to the independent amplitude modulation of an in-phase carrier (i.e., the cosine carrier) and a quadrature carrier (i.e., the sine carrier).

Performance in Additive Gaussian Noise. The optimum receiver for QAM signals chooses the signal that maximizes

$$l_i = A_i r_c + B_i r_s - \frac{\sqrt{E}}{4}(A_i^2 + B_i^2)$$

where

$$r_c = \int_0^T r(t)\, p(t)\cos(2\pi\ f_c t)dt$$

and

$$r_s = \int_0^T r(t)\, p(t)\sin(2\pi\ f_c t)dt$$

For square constellations that correspond to independent PAM of each carrier, an exact error probability is derived easily and is given by

$$P_{\mathrm{QAM}}(\mathrm{e}) = 1 - [1 - (1 - \frac{1}{\sqrt{M}})\mathrm{erfc}(\sqrt{\frac{3}{2(M-1)} \cdot \frac{E_{\mathrm{av}}}{N_0}})]^2$$

For cross constellations, tight upper bounds and good approximations are available. Figure 12plots the symbol error probability of various square QAM constellations as a function of SNR per bit.

Frequency-Shift Keying

As the name implies, frequency-shift keying (FSK) modulates the frequency of a carrier to convey information. FSK is one of the oldest digital modulation techniques and was the modulation of choice for the first, low-rate modems. Its main attribute, which makes it of interest in some appli-

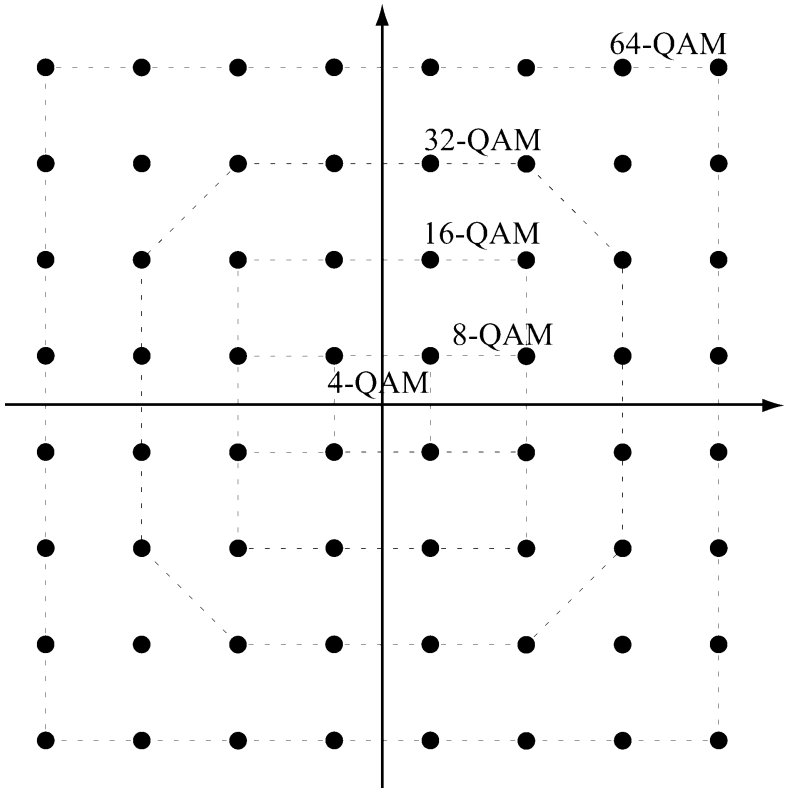


Figure 11. Signal space representation of various QAM constellations.

cations, is that it can be detected noncoherently (as well as coherently), which reduces the cost of the receiver. Mathematically, the modulated M-ary FSK signal is described by

$$s_i(t) = \sqrt{\frac{2E}{T}}\cos[2\pi(f_c + f_i)t], \quad 0 \le t \le T, \quad i = 1, 2, \cdots, M$$

where

$$f_i = (\frac{2i - 1 - M}{2})\Delta \mathrm{r}\ f$$

Δf is the minimum frequency separation between modulation tones. For orthogonal signaling (i.e., when the correlation between all pairs of distinct signals is zero), the minimum tone spacing is $1/2T$. This a condition is often imposed in practice. Orthogonal signaling performs well as a function of energy per bit, but it is also bandwidth-inefficient, which makes it impractical for high-speed, band limited applications.

Performance in Additive Gaussian Noise. FSK is detected coherently or incoherently. Coherent detection requires a carrier phase synchronization subsystem at the receiver that generates locally a carrier phase-locked to the received carrier. The optimum receiver for coherent detection makes decisions by maximizing the following (implementation assumes phase-coherence):

$$l_i = \int_0^T r(t)s_i(t)dt$$

For binary (orthogonal) signaling, the error probability is given simply by

$$\mathrm{P_{FSK}(e)} = \frac{1}{2}\mathrm{erfc}\,(\sqrt{\frac{E}{2N_0}}), \quad \text{(coherent FSK)}$$

which is 3 dB worse than BPSK. For M-ary signaling, an exact expression exists in integral form and is found, for example, in Reference (10). Noncoherent detection does not assume phase coherence and does not attempt to phase-lock the locally generated carrier to the received signal. In this case, it is easy to argue that the phase difference between the LO carrier and the received carrier is completely randomized. An optimum receiver is also derived in this case, and it is one that maximizes over the set of frequency tones

$$l_i = r_{ci}^2 + r_{si}^2$$

where

$$r_{ci}^2 = \int_0^T r(t)\cos[2\pi(f_c + f_i)t]dt$$

and

$$r_{si}^2 = \int_0^T r(t)\sin[2\pi(f_c + f_i)t]dt$$

The exact error-probability performance of this noncoherent receiver is available in analytical form, but it is complicated to compute for the general M-ary case (see, for example, Reference (10)). For the binary case, the error prob-

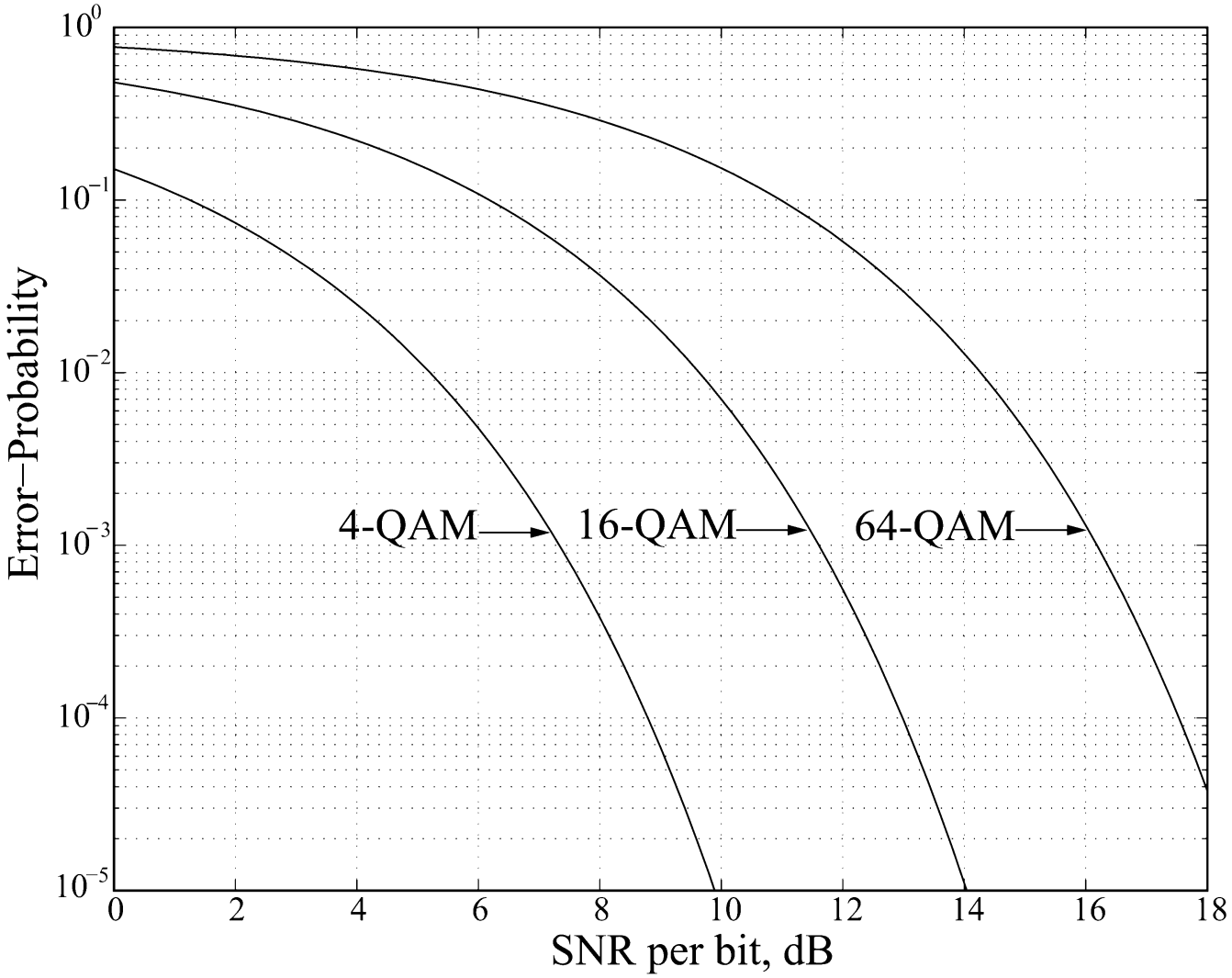


Figure 12. Symbol error probability as a function of SNR per bit for 4-, 16-, and 64-QAM.

ability has a simple form given by

$$\mathrm{P_{FSK}(e)} = \frac{1}{2} e^{-\frac{E}{2N_0}} \quad \text{(noncoherent FSK)}$$

Figure 13compares the performance of coherent and incoherent binary FSK. At an error probability of about 10^{-6}, noncoherent detection is inferior only slightly more than half a decibel compared with coherent detection. However, this small loss is well compensated for by the fact that no carrier phase synchronization is needed for the former.

Continuous-Phase Modulation

All modulation schemes described so far are memoryless, in the sense that the signal transmitted in a certain symbol interval does not depend on any past (or future) symbols. In many cases, for example, when a need exists to shape the transmitted signal spectrum to match that of the channel, it is necessary to constrain the transmitted signals in some form. Invariably, the imposed constraints introduce memory into the transmitted signals. One important class of modulation signals with memory are continuous-phase modulation (CPM) signals. These signals constrain the phase of the transmitted carrier to be continuous, thereby reducing the spectral sidelobes of the transmitted signals. Mathematically, the modulation signals for CPM are described by the expression

$$u(t) = A\cos[2\pi\ f_c t + \phi(t; \mathbf{d})]$$

where

$$\phi(t; \mathbf{d}) = 2\pi \sum_{k=-\infty}^{n} d_k h_k q(t - kT), \quad nT \le t \le (n+1)T$$

The d_k are the modulation symbols and h_k are the modulation indices, which may vary from symbol to symbol. For binary modulation, the modulation symbols are either 1 or -1. Finally, $q(t)$ is the integral of some baseband pulse $p(t)$ containing no impulses (thus guaranteeing that $q(t)$ is continuous)

$$q(t) = \int_{-\infty}^{t} p(\tau) d\tau$$

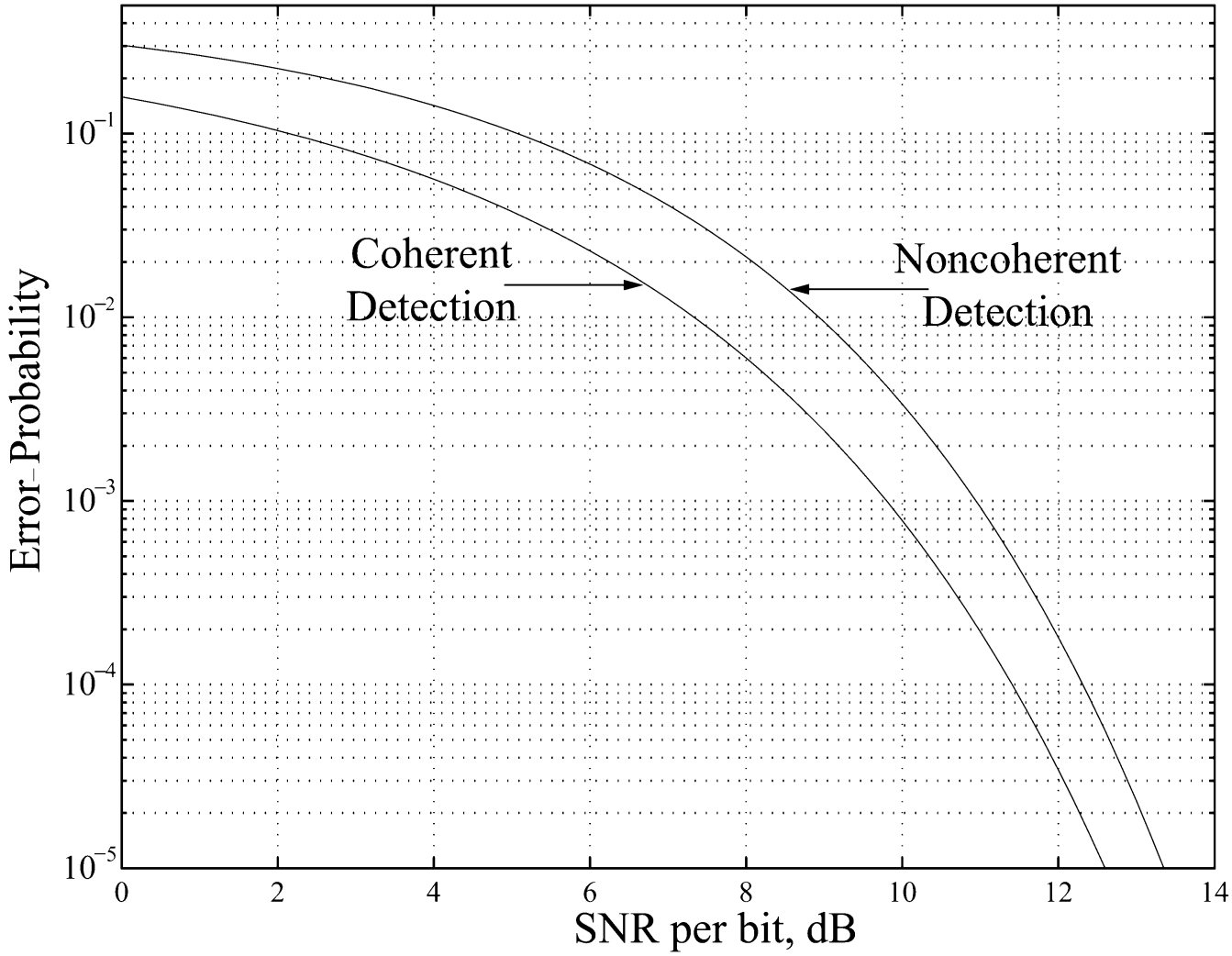


Figure 13. Error probability comparison between coherent and noncoherent FSK.

When $p(t)$ is zero for $t \geq T$, we have what is called *full-response* CPM, otherwise, we have *partial-response* CPM. In general, partial-response CPM achieves better spectral sidelobe reduction than does full-response CPM. A special case of CPM in which the modulation indices are all equal and $p(t)$ is a rectangular pulse of duration T seconds is called continuous-phase FSK (CPFSK). If, $h = 1/2$, we have what is called minimum-shift keying (MSK). A variation of MSK, in which the rectangular baseband pulse is first passed through a filter with a Gaussian-shape impulse response for further reduction in the spectral sidelobes, is called Gaussian MSK (GMSK). Various simple ways for detecting GMSK are available, which combined with its spectral efficiency, has made it a popular modulation scheme. In particular, it is the modulation scheme originally used for the European digital cellular radio standard, known as GSM. For more information on CPM signaling, including spectral characteristics and performance in noise, refer to Reference (10).

Modulation Codes

Another technique for shaping the spectrum of transmitted modulation signals is putting constraints on the sequence of bits sent to the modulator. This coding of bits to shape the spectrum of the transmitted modulation signals is called modulation coding or line coding. Important examples of the use of such codes are in magnetic and optical recording channels. Simple examples of modulation codes are found in the baseband transmission of binary data where a pulse is sent for a binary "1" and its negative for a "0" (called antipodal signaling). If the pulse amplitude does not return to zero in response to consecutive similar bits, then we have nonreturn-to zero (NRZ) signaling. If the pulse returns to zero, then we have return-to-zero (RZ) signaling. The encoding of bits using NRZ and RZ signaling is illustrated in Fig. 14.

It is often desirable to have a transmitted pulse sequence, in response to random input bits, with no spectral component at zero frequency (i.e., in dc). This condition is desirable, for example, when the modulation signals are sent through a channel with a null at dc. If the bits arriving

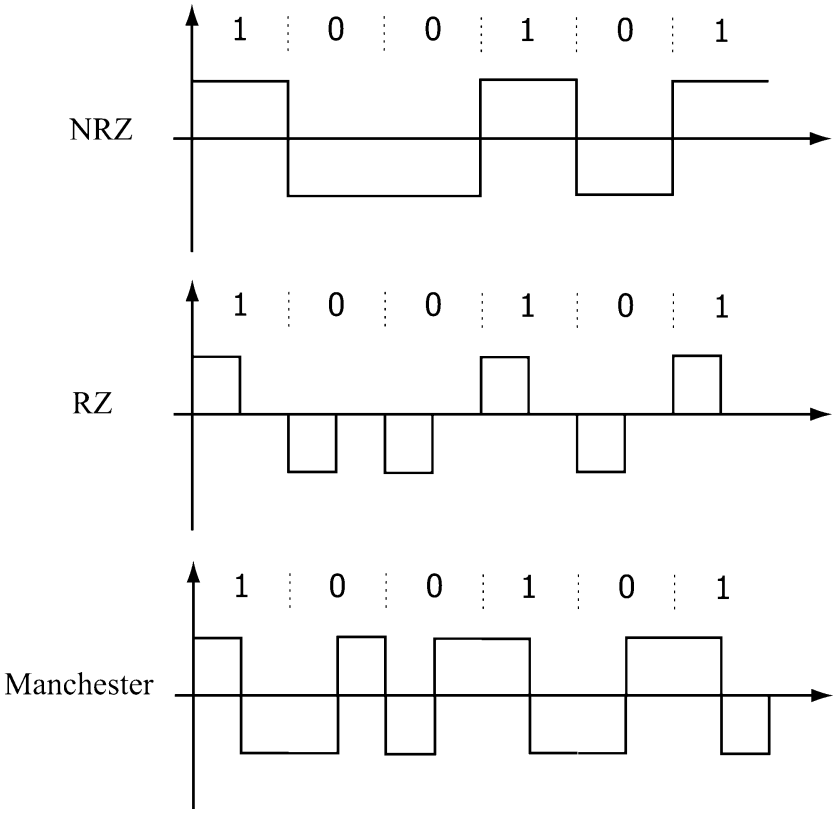


Figure 14. Illustration of NRZ, RZ, and Manchester coding.

at the input of the modulator are truly random (each with probability 1/2 of being zero or one) and independent, then the expected value of the dc component of the transmitted signal is zero. However, at any given time (even though the average is zero), a significant dc component may be caused by the transmission of a long sequence of zeros or ones. Besides the creation of a dc component, these long sequences of zeros or ones also negatively affect the performance of the timing recovery system at the receiver, whose function is to establish time synchronization (essential before data detection).

Biphase or Manchester pulses have the property of zero dc over each bit interval. These pulses and their encoding are illustrated in Fig. 14, along with NRZ and RZ signaling. An important property of a line code that describes the dc variations of a baseband signal is the running digital sum (RDS) (11). The RDS is the running sum of the baseband amplitude levels. It has been shown that, if the RDS for a modulation code is bounded, then the code has a null at dc (12). This process facilitates transmission of the modulated data through channels with a null at dc and avoids a form of intersymbol-interference (ISI) known as baseline wander. A converse result also shows that modulation codes, generated by finite-state machines, which have a spectral null at dc, have a bounded RDS (13).

Run-Length Limited Codes. Run-length limited (RLL) codes are an important class of modulation codes, which are often used in magnetic recording systems. RLL codes impose constraints on the minimum and maximum number of consecutive zeros between ones and are also called *(d, k)* codes, where d is the minimum number of zeros and k is the maximum number of zeros between ones. The minimum number of zeros between ones ensures that ISI is kept small, and the maximum number of zeros between ones ensures that the transmitted signal has enough transitions in it to aid in timing recovery. RLL codes (and in fact a much larger class of codes) are conveniently described by finite-state machines (FSMss). An FSM consists of a set of interconnected states that describe the allowable bit transitions (paths). The interconnections between all possible pairs of states are often described by a two-dimensional state transition matrix, which is known also as the *adjacency* matrix. A one at the i,j position in the matrix means that there is a path from state i to state j. A zero means that no path exists between the two states. Figure 15shows the FSM for the (1,3) *(d, k)* code. It consists of four states, and its adjacency matrix is given by

$$\mathrm{A} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Clearly, the constraints imposed on the binary sequences (in the form of d and k) limit the number of possible sequences of a given length n, which satisfy the constraint to a subset of the total number of 2^n possible sequences. If the number of sequences of length n satisfying the *(d, k)* constraints is $M(n)$, then the *capacity* of the code is defined

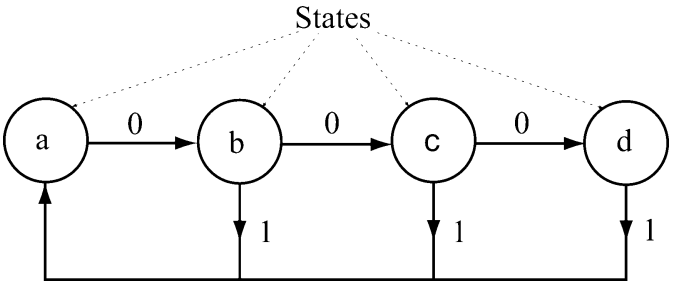


Figure 15. The finite-state machine for the ((1, 3)) RLL code.

by

$$C(d, k) = n \to \infty \lim \frac{1}{n} \log_2[M(n)] \tag{10}$$

For a fixed n, the ratio on the right-hand side of equation 10is called the rate of the code (which is the fraction of information bits per transmitted bit). It can be shown that the rate of the code is monotonically nondecreasing in n. Thus, the capacity of the code is the largest achievable rate. Shannon (14,15) has shown that the capacity of a FSM (the *(d, k)* code is just an example) is given by

$$C(d, k) = \log_2(\lambda_{\max})$$

where $\lambda_{\max}$ is the largest real eigenvalue of the adjacency matrix of the FSM. As an example, the eigenvalues of the adjacency matrix for the (1,3) code are 1.4656, -1.0000, $-0.2328 + 0.7926$i, and $-0.2328 - 0.7926$i The largest real eigenvalue is 1.4656, and thus, the capacity of the code is $\log_2(1.4656) = 0.5515$. For an excellent overview of information theory, including Shannon's result above, consult Reference (16).

The fact that an FSM is found that produces sequences satisfying the necessary constraints does not automatically imply that a code has been constructed. The problem of assigning information bits to encoded bits still exists. The problem of constructing such codes from their FSM representation has been studied by Adler et al. (17). An excellent tutorial paper on the topic can be found in Reference (18). Practical examples of applying the results of Reference (17) are, for example, in References (19) and (20). Another important class of codes that shapes the spectrum of the transmitted data and achieves a coding gain in the process is the class of matched spectral null (MSN) codes. The interested reader is referred to the paper by Karabed and Siegel (21)for more details.

Yet, another, very important class of modulation signals includes those signals that combine coding and modulation for improved performance. These combined modulation and coding techniques and, in particular, trellis-coded modulation (TCM) became better known from the breakthrough paper of Unger-boeck (22). In contrast to previous classic coding techniques that separate the coding and modulation problems, TCM achieves a coding gain (i.e., improved performance) without expanding bandwidth. It is thus very appealing in band limited applications, such as telephone modems, where it has been widely employed.

[1] This assumption is not as easy to justify when the receiver moves relative to the transmitter, because of the frequency offset caused by the Doppler effect.

BIBLIOGRAPHY

1. Proakis J.; Salehi M. *Communication Systems Engineering*; Prentice-Hall: Englewood Cliffs, NJ, 1994.
2. Gardner F. M. *Phaselock Techniques*; Wiley: New York, 1966.
3. Viterbi A. J. *Principles of Coherent Communications*; McGraw-Hill: New York, 1966.
4. Lindsey W. C. *Synchronization Systems in Communications*; Prentice-Hall: Englewood Cliffs, NJ, 1972.
5. Blanchard A. *Phase-Locked Loops: Application to Coherent Receiver Design*; Wiley: New York, 1976.
6. Stremler F. G. *Introduction to Communication Systems*; 3rd ed.; Addison-Wesley: Reading, MA, 1990.
7. Haykin S. *Communication Systems*, 3rd ed.; Wiley: New York, 1994.
8. Roden M. S. *Analog and Digital Communication Systems*; Prentice-Hall: Englewood Cliffs, NJ, 1991.
9. Couch L. W. *Modern Communication Systems*; Prentice-Hall: Englewood Cliffs, NJ, 1995.
10. Proakis J. *Digital Communications*, 3rd ed.; McGraw-Hill: New York, 1995.
11. Franaszek P. A. Sequence-State Coding for Digital Transmission. *Bell Syst. Tech. J.*, 1968, **47**, 143.
12. Calderbank A. R.; Mazo J. Spectral Nulls and Coding with Large Alphabets. *IEEE Commun. Mag.*December 1991.
13. Yoshida S.; Yajima Y. On the Relationship Between Encoding Automaton and the Power Spectrum of its Output Sequence. *Trans. IECE* 1976, **E59**,p. 97.
14. Shannon C. E. A Mathematical Theory of Communication. *Bell Syst. Tech. J.* 1948, **27**,pp 379–423.
15. Shannon C. E. A Mathematical Theory of Communication. *Bell Syst. Tech. J.* 1948, **27**,pp. 623–656.
16. Cover T. M.; Thomas J. A. Elements of Information Theory; Wiley Interscience: New York, 1991.
17. Adler R. L.; Coppersmith D.; Hassner M. Algorithms for Sliding Block Codes. *IEEE Trans. Inform. Theory* 1983, **IT-29**,pp. 5–22.
18. Marcus B. H.; Siegel P. H.; Wolf J. K. Finite-State Modulation Codes for Data Storage. *IEEE J. Select. Areas Commun.* 1992, **10**,pp. 5–37.
19. Calderbank A. R.; Georghiades C. N. Synchronizable Codes for the Optical OPPM Channel. *IEEE Trans. Inform. Theory* 1994, **40**,pp. 1097–1107.
20. Soljanin E.; Georghiades C. N. Coding for Two-Head Recording Systems. *IEEE Trans. Inform. Theory* 1995, **41**,pp. 747–755.

21. Karabed R.; Siegel P. Matched-Spectral Null Codes for Partial Response Channels. *IEEE Trans. Inform. Theory* 1991, **IT-37**,pp. 818–855.
22. Ungerboeck G. Channel Coding with Multilevel/Phase Signals. *IEEE Trans. Inform. Theory* 1982, **IT-28**,pp. 55–67.

COSTAS N. GEORGHIADES
Texas A&M University

} { { } }

Information Theory of Multiaccess Communications
Standard Article

Tien M. Nguyen[1], Hung Nguyen[2], Boi N. Tran[3]
[1]The Aerospace Corporation, El Segundo, CA
[2]Mountain Technology Inc., Milpitas, CA
[3]The Boeing Company, Long Beach, CA

DOI: 10.1002/047134608X.W4211
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (204K)

Abstract

The sections in this article are

Fixed-Assignment Multiple Access

Random-Assignment Multiple Access

Performance Comparison of Multiple Access Techniques

Applications of Random-Access Techniques in Cellular Telephony

Keywords: communications resources; multiple access; multiplexing; frequency-division multiple access; time-division multiple access; code-division multiple access; space-division multiple access; polarization-division multiple access; fixed-assignment multiple access; random access; controlled random access; guard band; direct-sequence spread spectrum; frequency-hopping spread spectrum; processing gain; near–far problem; crosscorrelation; random-assignment multiple access; pure ALOHA; slotted ALOHA; reservation ALOHA; carrier-sense multiple access; data-sense multiple access; 1-persistent carrier-sense multiple access; nonpersistent carrier-sense multiple access; p-persistent carrier-sense multiple access; polling technique; token passing; cellular digital packet data; controlled random-assignment multiple access; carrier-sense multiple access with busy-tone signaling; cellular telephone system; mobile telephone switching office; public switched telephone network; mobile switching service center; common air interface; roaming service; mobile identification number; electronic serial number; advanced mobile phone system; extended European total access cellular system IS-54; US digital cellular; global mobile system, IS-95

INFORMATION THEORY OF MULTIACCESS COMMUNICATIONS

The basic communications resources available to users are frequency and time. The efficient allocation of these communications resources lies in the domain of communications multiple access. The term "multiple access" means the remote sharing of a communications resource (e.g., satellite). The term *multiple access* is often confused with the term *multiplexing*. Multiplexing indicates the local sharing of a communications resource (e.g., a circuit board). Normally, for multiplexing, the resource allocation is normally assigned a priori. This article focuses on the theory of multiple access. High level description of various multiple access techniques and a comparison among them will be given.

For multiple access, there are three basic techniques for distributing the communications resources: frequency-division multiple access (FDMA), time-division multiple access (TDMA), and code-division multiple access (CDMA). For FDMA, one specifies the subbands of frequency to be allocated to users. For TDMA, periodically recurring time slots are identified and then allocated to users. This technique allows users to access the resource at fixed or random times, depending on the systems. For CDMA, full channel bandwidth is utilized simultaneously with the time resource. In addition, two other techniques for multiple access are also available, namely, space-division multiple access (SDMA) and polarization-division multiple access (PDMA). SDMA, also referred to as multibeam frequency reuse multiple access, uses spot beam antennas to separate radio signals by pointing them in different directions, which allows for reuse of the same frequency band. PDMA, or dual polarization frequency reuse, employs orthogonal polarization to separate signals, which also allows for reuse of the same frequency band.

The three basic multiple access schemes are implemented with various multiuser access algorithms to form fixed-assignment or random-access schemes. In a fixed-assignment access scheme, a fixed allocation of communication resources, frequency or time, or both, is made on a predetermined basis to a single user. The random-access scheme allows the users to access communications resources randomly. When the random-access algorithm exercises some control over the access method to improve the efficiency of the uncontrolled random access methods, the result is referred to as the controlled random access technique.

This article describes the underlying theory behind the multiple access techniques and their applications in satellite and cellular systems. Both fixed- and random-access techniques will be described with their associated applications. Since the article is intended for readers who are unfamiliar

with this field, only high level descriptions with minimum technical details are presented.

FIXED-ASSIGNMENT MULTIPLE ACCESS

As mentioned earlier, multiple access techniques are required for multiple users to efficiently share remote communications resources. There are two major categories of multiple access methods: fixed assignment and random access. This section describes the three basic approaches for fixed-assignment multiple access: FDMA, TDMA, and CDMA. For completeness, brief descriptions of SDMA and PDMA are also presented.

Frequency-Division Multiple Access

The frequency-division multiple access (FDMA) technique is derived based on the frequency-division multiplexing (FDM) method. The FDM method involves mixing (or heterodyning) the signals at the same frequency band with fixed frequencies from local oscillators to different frequency bands and then combining the resulting multiple signals (at different frequency bands) for transmission as a single signal with a wider bandwidth (1). Figure 1 shows the FDM scheme (1, Fig. 9.3, page 480). Note that "guard bands" between the frequency assignments are provided as buffer zones to mitigate the adjacent channel interference. For a fixed-assignment FDMA system, a user is assigned to a fixed subchannel for transmission, and the subchannel is retained until released by the assigned user. The receiver terminal has precise knowledge of the transmission subchannel, and a filter is used to extract the designated signal out of the received composite signal.

The advantages of the fixed-assignment FDMA are (2):

- Channel capacity increases as information bit rate decreases. To reduce the information bit rate one can use an efficient modulation method such as M-ary phase-shift keying (PSK) (3) or the continuous phase modulation (CPM) technique (4).
- Implementation is simple due to technological advances.

The disadvantages associated with a fixed-assignment FDMA are:

- It needs to back-off the transmitter high power amplifier (HPA) from saturation point to avoid intermodulation caused by AM–AM and AM–PM distortions (1,5). This means it is power inefficient.
- FDMA is involved with narrowband technology, which also involves narrowband filters that may not be realizable in very large scale integrated (VLSI) digital circuits. This means higher cost for terminals even under volume production conditions.
- It is inflexible due to limited fixed bit rate per channel.

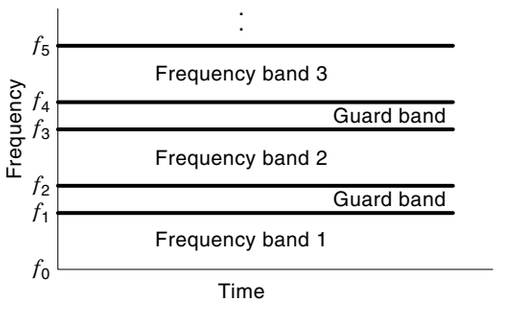


Figure 1. Illustration of frequency-division multiple access technique.

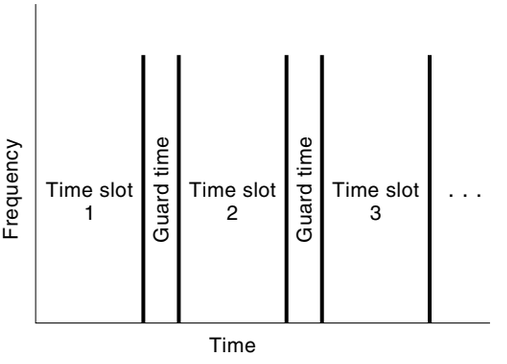


Figure 2. Illustration of time-division multiple access technique.

Time-Division Multiple Access

Time-division multiple access (TDMA) uses the full spectrum occupancy that is allocated to the system for a short duration of time called the time slot, as shown in Fig. 2 (1, Fig. 9.7, p. 484). Note that the guard band is provided here for crosstalk avoidance. TDMA employs the time-division multiplexing method in which the system time is divided into multiple time slots used by different users. Several time slots make up a frame. Each slot is made up of a preamble plus information bits addressed to various terminal users as shown in Fig. 3 (1, Fig. 9.9, p. 485). In a fixed-assignment TDMA system, a transmit controller assigns different users to different time slots, and the assigned time slot is retained by that user until the user releases it. At the receiving end, a user terminal synchronizes to the TDMA signal frame and extracts the time

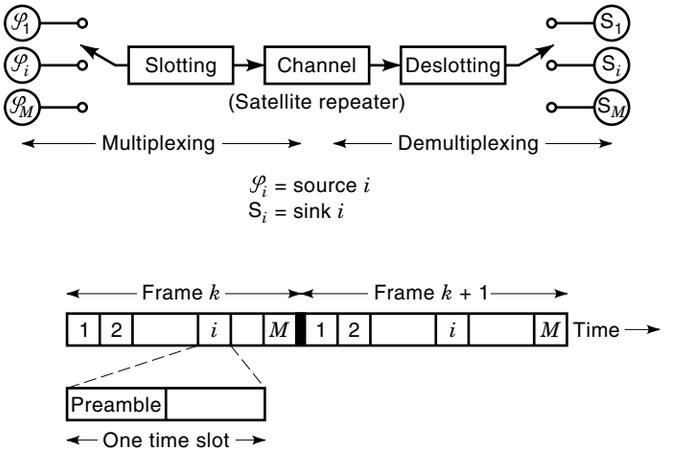


Figure 3. Illustration of fixed-assignment time-division multiple access technique.

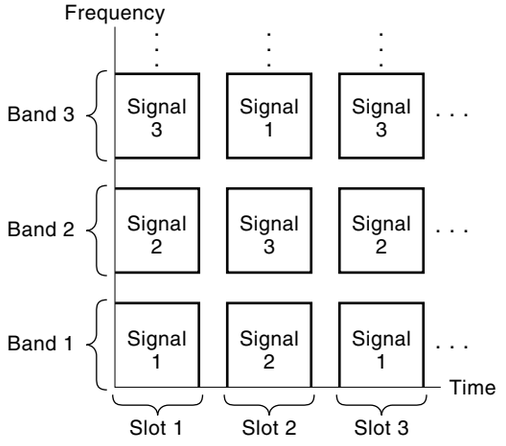


Figure 4. Illustration of code-division multiplexing.

slot assigned to that user. Figure 3 illustrates the demultiplexing procedure for a fixed-assignment TDMA system.

The advantages of a fixed-assignment TDMA include:

- When used with a constant modulation scheme, the transmitter HPA can operate at saturation. This means it is power efficient.
- It is flexible due to variable bit rates allowed for users.
- VLSI technology can be used for low cost in volume production.
- TDMA utilizes bandwidth more efficiently because no frequency guard band is required between the channels.

The disadvantages associated with fixed-assignment TDMA are (2):

- TDMA requires higher peak power than FDMA. This may cause significant drawback for mobile applications due to the shortening of battery life.
- Complicated signal processing is used in the detection and synchronization with a time slot.

Code-Division Multiple Access

Code-division multiple access (CDMA) is a hybrid combination of FDMA and TDMA (1,6). Figure 4 illustrates this concept (1, Fig. 9.14, p. 491). For CDMA, the system operates simultaneously over the entire system frequency bandwidth and system time. In CDMA systems the users are kept separate by assigning each of them a distinct user-signal code. The design of these codes is usually based on spread-spectrum (SS) signaling to provide sufficient degrees of freedom to separate different users in both time and frequency domains (although the use of SS does not imply CDMA). SS technique can be classified into two categories, namely, direct-sequence SS (DS-SS) and frequency-hopping SS (FH-SS) (7). Hence, CDMA can also categorize into DS-CDMA and FH-CDMA. In CDMA systems a hybrid combination of DS and FH for CDMA is also allowed. In the following, brief descriptions of the DS-CDMA and FH-CDMA are given.

Direct-Sequence CDMA. In DS-CDMA systems each of N users is preassigned its own code, $PN_i(t)$, where $i = 1, 2, 3, \ldots, N$. The user codes are selected such that they are approximately orthogonal to each other. This means that the cross-correlation of two different codes is approximately zero; that is

$$\int_0^{T_c} PN_i(t)PN_j(t)\,dt \approx \begin{cases} 1, & \text{for } i = j \\ 0, & \text{for } i \neq j \end{cases} \tag{1}$$

where T_C denotes the time duration of the code and usually is referred to as the chip duration. Since the assigned codes are orthogonal to each other, they can be spread over the entire spectrum of the communication resource simultaneously.

The modulated signal for user 1 is denoted as

$$S_1(t) = A_1(t)\cos[\omega_0 t + \phi_1(t)] \tag{2}$$

where $A_1(t)$, ω_o, and $\phi_1(t)$ are the amplitude, angular frequency, and phase, respectively, of the signal specified for user 1. Note that the modulated waveform presented in Eq. (2) is expressed in general form, without any restriction placed on modulation type. Then the spread signal is obtained by multiplying signal $S_1(t)$ with the code $PN_1(t)$, and the resultant signal, $S_1(t)PN_1(t)$, is then transmitted over the channel. Figure 5 shows a simplified block diagram for a typical CDMA system (1, Fig. 10.25, p. 572). Here the bandwidth of the code $PN_1(t)$ is much larger than the unspread signal $S_1(t)$. If one denotes the code rate for $PN_1(t)$ as R_c and the signal data rate

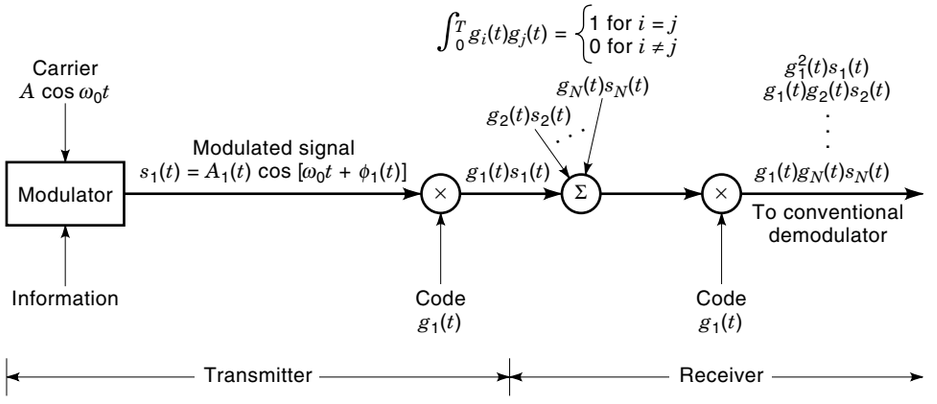


Figure 5. Illustration of code-division multiple access technique.

as R_s, then the processing gain G_p of the system is given by (1,7)

$$G_p\,(\text{dB}) = 10\log\left(\frac{R_c}{R_s}\right) \tag{3}$$

The processing gain provides an indication of how well the signal $S_1(t)$ is being protected from interfering signals (intentional or unintentional). The larger the value of G_p, the better the protection the code can provide.

The spread signal $S_1(t)PN_1(t)$ is received in the presence of other spread signals, $S_2(t)PN_2(t)$, $S_3(t)PN_3(t)$, . . ., $S_N(t)PN_N(t)$. Assuming that the noise at the receiver is zero and the signal delays are negligible, we can write the received signal $R(t)$ as

$$R(t) = S_1(t)PN_1(t) + \sum_{i=2}^{N} S_i(t)PN_i(t) \tag{4}$$

Here we will also assume that the receiver is configured to receive messages from user 1 so that the second term shown in Eq. (4) is an interference signal. To recover the signal $S_1(t)$, the received signal $R(t)$ is despread by multiplying $R(t)$ with the code $PN_1(t)$ stored at the receiver,

$$R(t)PN_1(t) = S_1(t) + \sum_{i=2}^{N} S_i(t)PN_i(t)PN_1(t) \tag{5}$$

Here we have used the property $PN_1^2(t) = 1$. If we chose the code to have the orthogonal property, that is, the codes are chosen to satisfy the condition expressed in Eq. (1), then it can be shown that the undesired signal expressed in the second term of Eq. (5) is negligible (7,8). Since the codes are not perfectly orthogonal, the second term of Eq. (5) is negligible for a limited number of users. The performance degradation caused by the crosscorrelation in the second term sets the maximum number of simultaneous users. A rule of thumb for determining the maximum number of users N appears to be that (7)

$$N \approx \frac{10^{G_p(\text{dB})/10}}{10} \tag{6}$$

While the code design is of paramount importance, of potential greater importance in DS-CDMA is the so-called near–far problem (7,9,10). Since the N users are usually geographically separated, a receiver is trying to detect the ith user, which is much farther than the jth user. Therefore, if each user transmits with equal power, the power received by the jth user would be much stronger than that received by the ith user. This particular problem is often so severe that DS-CDMA systems will not work without appropriate power control algorithms.

Advantages associated with DS-CDMA include:

- Multiple users can share the communication resources, both frequency and time, simultaneously.
- Communication privacy is possible due to assigned codes being known only to the users.
- There is an inherent robustness against mobile channel degradations such as fading and multipath (7–10).
- There is greater resistance to interference effects in a frequency reuse situation.
- More flexibility is possible because there is no requirement on time and frequency coordination among the various transmitters.

The disadvantages of DS-CDMA are:

- It requires power control algorithms due to the near–far problem.
- Timing alignments must be within a fraction of a coded sequence chip.
- Performance degradation increases as the number of users increases.

Frequency-Hopping CDMA. An alternative to DS-CDMA is FH-CDMA (1,7). In FH-CDMA systems each user is assigned a specific hopping pattern, and if all hopping patterns assigned are orthogonal, the near–far problem will be solved (except for possible spectral spillover from a specified frequency slot into adjacent slots). In practice, the codes assigned for these hopping patterns are not truly orthogonal; thus, interference will result when more than one signal uses the same frequency at a given instant of time. A simplified block diagram for a typical FH-CDMA modulator is shown in Fig. 6 (1, Fig, 9.15, p. 492).

FH-CDMA can be classified as fast FH-CDMA or slow FH-CDMA. Fast FH systems use a single frequency hop for each transmitted symbol. This means that, for fast FH systems, the hopping rate equals or exceeds the information symbol rate. On the other hand, slow FH systems transmit two or more symbols in the time interval between frequency hops.

The advantages associated with FH-CDMA include:

- Multiple users can share the communication resources, both frequency and time, simultaneously.
- Communication privacy is possible due to assigned codes being known only to the users.
- There is an inherent robustness against mobile channel degradations such as fading and multipath (7–10).

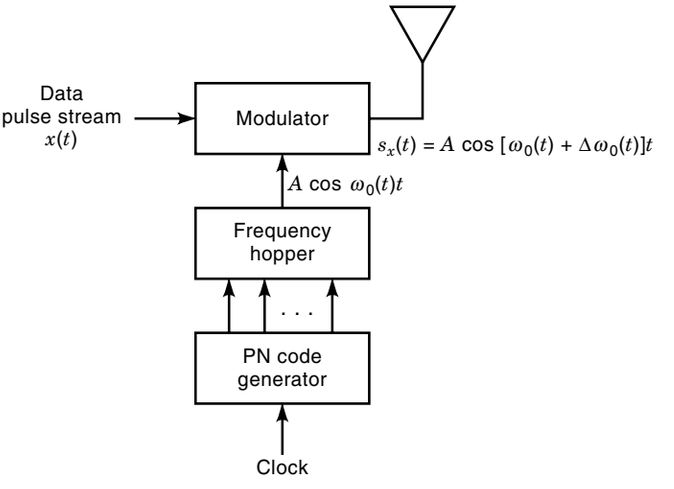


Figure 6. Illustration of code-division multiple access frequency hopping.

- There is an inherent robustness against interference.
- The near–far problem does not exist.
- Network implementation for FH-CDMA is simpler than DS-CDMA systems because the required timing alignments must be within a fraction of a hop duration as compared to a fraction of a chip duration.
- It performs best when a limited number of signals are sent in the presence of nonhopped signals.

The disadvantages are:

- Performance degradation is possible due to spectral spillover from a specified frequency slot into adjacent slots.
- Frequency synthesizer can be very costly.
- As the hopping rate increases the reliability decreases and synchronization becomes more difficult.

Space-Division Multiple Access

For wireless applications, space-division multiple access (SDMA) can be classified into cell-based and beamforming-based SDMA. The difference between the two approaches can best be illustrated in Fig. 7 (11, Fig. 1.1, p. 4) for cell-based SDMA and Fig. 8 (11, Fig. 1.2, p. 5) for beamforming-based SDMA.

A primitive form of SDMA exists when frequency carriers are reused in different cells separated by a special distance to reduce the level of co-channel interference. The larger the number of cells the higher the level of frequency reuse and thus the higher capacity that can be attained. This has resulted in cell-based SDMA, which has been predominant for quite a long time.

In a frequency reuse system, the term *radio capacity* is used to measure the traffic capacity, and is defined as

$$C_r = \frac{M}{K \cdot S} \tag{7}$$

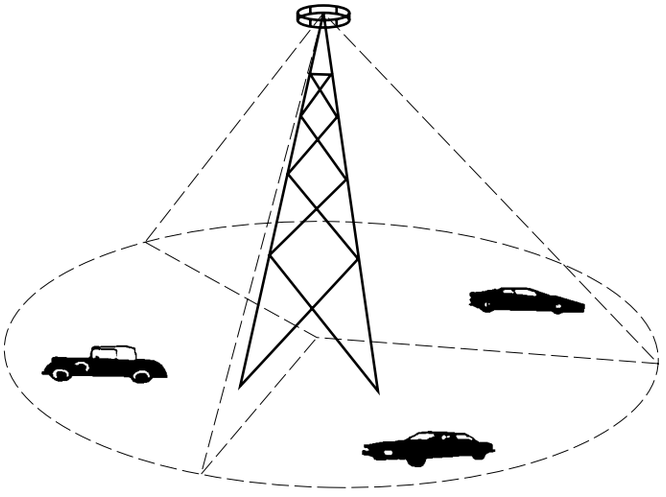


Figure 7. Illustration of the cell-based space-division multiple access. A different set of carrier frequencies is used in each of the sectors. These frequencies are used in other sectors of other cell sites. The frequency reuse pattern is selected to minimize the interference.

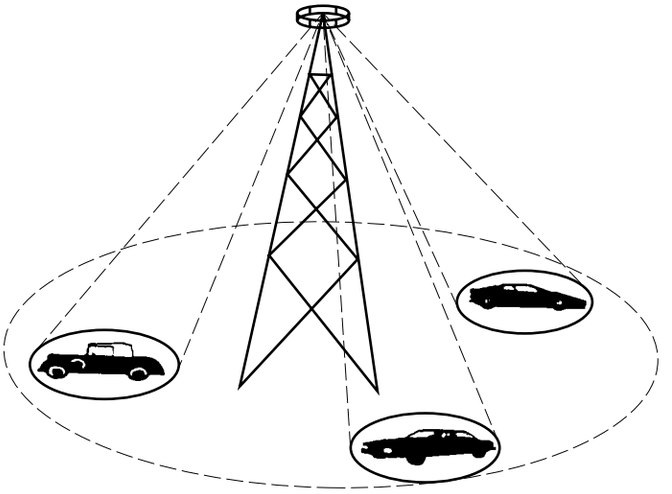


Figure 8. Illustration of beamforming-based space-division multiple access.

where M is the total number of frequency channels, K is the cell reuse factor and S is the number of sectors in a cell. K can be expressed as

$$K = \frac{1}{3}\left(\frac{D}{R}\right)^2 \tag{8}$$

where D is the distance between two co-channel cells and R is the cell radius. The corresponding average signal-to-interferer ratio (SIR) can be calculated for different types of sectoring systems, including adaptive beamforming with several beams in beamforming-based SDMA.

The system benefits of beamforming-based SDMA include:

- Improvement of multipath fading problems since narrower beams are used and the implicit optimal diversity combining performed by the beamformer
- More flexible coverage of each base station to match the local propagation conditions

Table 1 lists the capacity and SIR for several SDMA configurations (12).

Adaptive beamforming algorithms require a certain reference signal in the optimization process. If the reference signal is not explicit in the received data, blind adaptive beamforming (BAF) can be used instead. For digital communication signals, one can vary certain signal properties such as constant modulus applicable to FSK or PSK signals to result in the

Table 1. Radio Capacity and Signal-to-Noise Ratio for Different Cells

	K	S	Capacity (Channels/Cell)	SIR (dB)
Omnicells	7	1	$M/7$	18
120° sectorial cells	7	3	$M/21$	24.5
60° sectorial cells	4	6	$M/24$	26
60° sectorial beams	7	6	$3M/7$	20
N adaptive beams	7	1	$MN/7$	18

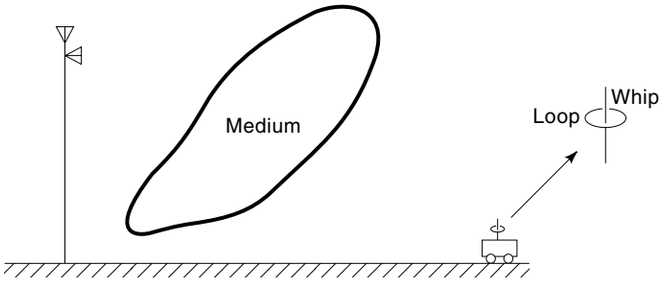


Figure 9. Illustration of horizontal and vertical polarization diversity signals.

constant modulus adaptive beamforming algorithm (13), or the cyclostationary properties of bauded signals to suggest the spectral self-coherence restoral (SCORE) algorithm (14).

Another method (15) that can be considered blind adaptive beamforming is based on decision-directed equalization to combat intersymbol interference (ISI) in digital communications. Using this concept, a BAF demodulates the beamformer output and uses it to make a decision in favor of a particular value in the known alphabet of the transmit sequence. A reference signal is then generated based on the modulated output of this decided demodulated beamformer output.

Polarization-Division Multiple Access

Signals transmitted in either horizontal or vertical electric field are uncorrelated at both the mobile and base station's receiver. Suppose that a received vertically polarized signal is

$$\Gamma_{11} = \sum_{i-1}^{N} a_i e^{j\psi_i} e^{-j\beta V t \cos\phi_i} \tag{9}$$

and the received horizontally polarized signal is

$$\Gamma_{22} = \sum_{i=1}^{N} a_i' e^{j\psi_i'} e^{-j\beta V t \cos\phi_i} \tag{10}$$

where a_i and ψ_i are the amplitude and phase, respectively, for each wave path and a_i' and ψ_i' are their counterparts in Eq. (9), V is the vehicle velocity, and ϕ_i is the angle of arrival of the ith wave. Although these two polarized waves arrived at the receiver from the same number of incoming waves, it is not difficult to see that Γ_{11} and Γ_{22} are uncorrelated because of their different amplitudes and phases. Thus, a PDMA system can be illustrated as in Fig. 9 (16, Fig. 9-6, p. 281). In this system, the base station can be two vertical and horizontal dipoles and the antenna at the mobile can be a pair of whip and loop antennas.

RANDOM-ASSIGNMENT MULTIPLE ACCESS

Fixed-assignment multiple access is most efficient when each user has a steady flow of information for transmission. However, this method becomes very inefficient when the information to be transmitted is intermittent or bursty in nature. As an example, for mobile cellular systems, where the subscribers pay for service as a function of channel connection time, fixed-assignment access can be very expensive for transmitting short messages. In this case, the random-access methods are more flexible and efficient than the fixed-access methods. This section discusses the three basic random-access schemes, namely, pure ALOHA, modified ALOHA (slotted and reservation), and carrier-sense multiple access with collision detection.

Pure ALOHA

Pure ALOHA (P-ALOHA), or basic ALOHA, was developed at the University of Hawaii in 1971 with the goal of connecting several university computers by the use of random-access protocol (17). The system concept is very simple and has been summarized by Sklar (1). The algorithm is listed below for future comparison with the enhanced version, the so-called slotted ALOHA.

- *Mode 1: Transmission Mode.* Users transmit at any time they desire, encoding their transmissions with an error detection code.
- *Mode 2: Listening Mode.* After a message transmission, a user listens for an acknowledgment (ACK) from the receiver. Transmissions from other users will sometimes overlap in time, causing reception errors in the data in each of the contending messages. We say the messages have collided. In such cases, the errors are detected, and the users receive a negative acknowledgment (NACK).
- *Mode 3: Retransmission Mode.* When a NACK is received, the messages are simply retransmitted. Of course, if the colliding users were retransmitted immediately, they would collide again. Therefore, the users retransmit after a random delay.
- *Mode 4: Timeout Mode.* If after a transmission, the user does not receive either an ACK or NACK within a specified time, the user retransmits the message.

Figure 10 shows the concept of the pure ALOHA algorithm (6, Fig. 11.15, p. 465).

Modified ALOHA

In order to improve the pure ALOHA algorithm, the slotted (18) and reservation ALOHA algorithms (19) have been proposed. Based on the summary described in Sklar (1), a brief description of these algorithms will be given here.

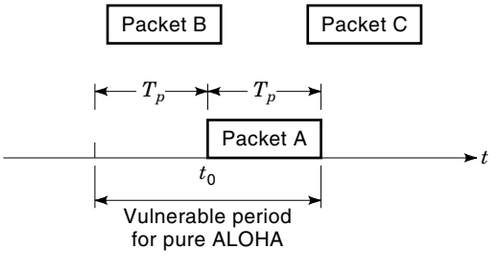


Figure 10. Illustration of collision mechanism in pure ALOHA.

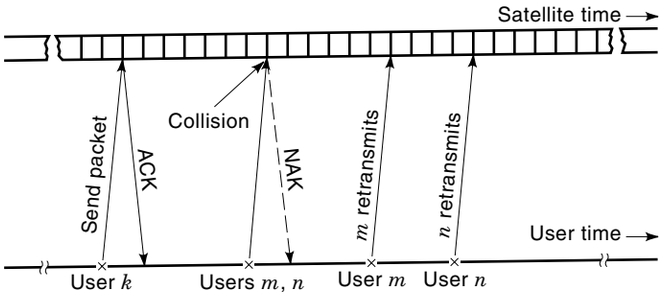


Figure 11. Illustration of slotted ALOHA.

Slotted ALOHA. The operation of the slotted ALOHA (S-ALOHA) is illustrated in Fig. 11 (1, Fig. 9.21, p. 501). A sequence of synchronization pulses is broadcast to all users for coordination among the users. Messages are sent through data packets with constant length between the synchronization pulses and can be started only at the beginning of a time slot. This modification reduces the rate of collisions by half, since only a packet transmitted in the same slot can interfere with one another (1). In S-ALOHA systems the users retransmit after a random delay of an integer number of slot times when a NACK occurs.

Reservation ALOHA. Significant improvement can be achieved with the reservation ALOHA (R-ALOHA) scheme. This scheme has two modes, namely, an unreserved mode and reserved mode.

The unreserved or quiescent mode, mode has three stages:

- A time frame is formed and divided into several reservation subslots.
- Users employ these small subslots to reserve message slots.
- After requesting a reservation, the users listen for an ACK and slot assignment.

The reserved mode has four stages:

- The time frame is divided into $M + 1$ slots whenever a reservation is made.
- The first M slots are used for message transmissions.
- The last slot is subdivided into subslots to be used for reservation/requests.
- Users send message packets only in their assigned portion of the M slots.

Figure 12 shows an example of the R-ALOHA system (1, Fig. 9.22, p. 503). In this example, the users seek to reserve 3 slots with $M = 5$ slots and $V = 6$ subslots. Compared with S-ALOHA, R-ALOHA is very efficient for high traffic intensity.

Carrier-Sense Multiple Access

To improve the previous algorithms and to make efficient use of the communications resources, the user terminal listens to the channel before attempting to transmit a packet. This protocol is called listen-before-talk and usually is referred to as carrier-sense multiple access (CSMA) protocol (12). This algorithm is widely used in both wired and wireless local area networks (LANs), where the transmission delays are low. There are several modified versions of CSMA, namely, CSMA with busy-tone signaling, CSMA with collision detection, and CSMA with collision avoidance. In addition, there is another modified version of CSMA, called data-sense multiple access (DSMA), which has been developed and adopted for use in wireless packet data networks such as cellular digital packet data (CDPD).

This section describes the three basic CSMA schemes, namely, 1-persistent CSMA, nonpersistent CSMA, and p-persistent CSMA. Modified versions of CSMA will also be described briefly.

1-Persistent Carrier-Sense Multiple Access. 1-Persistent carrier-sense multiple access (1-P CSMA) is the simplest form of CSMA. In the basic form, 1-P CSMA is unslotted. The "1-persistent" signifies the strategy in which the message is sent with probability 1 as soon as the channel is available. After sending the packet, the user station waits for an ACK, and if none is received in a specified amount of time, the user will wait for a random amount of time and then resume listening to the channel. When the channel is sensed idle, the packet is retransmitted immediately. In unslotted form, the system does not require synchronization between the user stations and all transmissions are synchronized to the time slots. In contrast with the unslotted form, the slotted 1-P CSMA requires synchronization among all user stations and all transmissions, whether initial transmissions or retransmissions, are synchronized to the time slots (1,6).

Nonpersistent Carrier-Sense Multiple Access. The main difference between the 1-P CSMA and nonpersistent carrier-

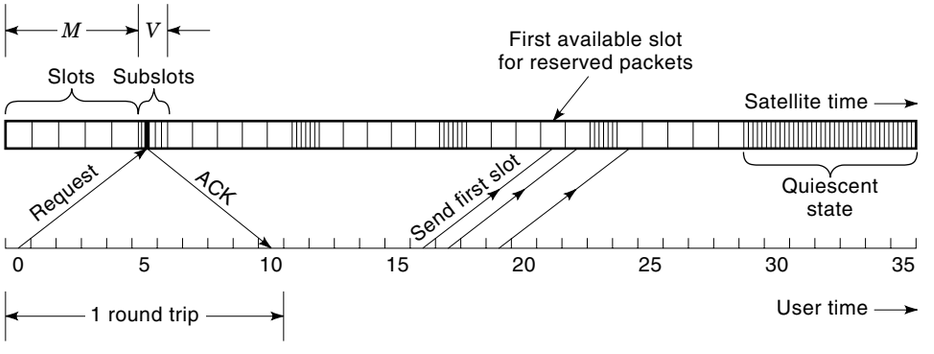


Figure 12. Illustration of reservation ALOHA. Station seeks to reserve 3 slots ($M = 5$ slots, $V = 6$ subslots).

sense multiple access (NP CSMA) is that a user station does not sense the channel continuously while it is busy. Instead, after sensing the busy condition, the NP CSMA system waits a randomly selected interval of time before sensing again. This random waiting time associated with NP CSMA could eliminate most of the collisions that would result from multiple users transmitting simultaneously upon sensing the transition from the busy to idle condition.

p-Persistent Carrier-Sense Multiple Access. The p-persistent carrier-sense multiple access (pP CSMA) is a generalization of the 1-P CSMA scheme, which is applicable to slotted channels. In this scheme, the slotted length is chosen to be the maximum propagation delay. In this system, a message is sent from a station with probability p when the channel is sensed to be idle. With probability $q = 1 - p$ the station defers action to the next slot, where the station senses the channel again. If the next slot is idle, the station transmits with probability p or defers with probability q. This procedure is repeated until either the whole frame has been transmitted or the channel is sensed to be busy. If the channel is busy, the station monitors the channel continuously until it becomes free; then it starts the above procedure again (6).

Carrier-Sense Multiple Access with Busy-Tone Signaling. In wireless networks, the user terminals are not always within the range and line-of-sight of each other, and when this situation occurs, it is referred to as "hidden terminal problem." This problem can be solved by using the carrier-sense multiple access with busy-tone signal (CSMA/BTS) technique (6). This technique divides the system bandwidth into two channels: a message channel and a busy-tone channel. The scheme works as follows. Whenever the central station senses signal energy on the message channel, it transmits a simple busy-tone signal on the busy-tone channel, and this tone is detectable by all the user stations. With this technique, a user station first senses the channel by detecting the busy-tone signal to determine if the network is busy. The procedure the user station then follows in transmission of the message depends on the particular version of CSMA being used in the network, and any of the CSMA techniques described earlier can be chosen.

Carrier-Sense Multiple Access with Collision Detection. The carrier-sense multiple access with collision detection (CSMA/CD) technique, also referred to as the "listen-while-talk" (LWT) technique, can be used with 1-P CSMA, NP CSMA, or pP CSMA, each with a slotted or unslotted version (6). In the operation of CSMA/CD, if the channel is detected to be idle or busy, a user station first sends the message (in the form of data packets) using the procedure dictated by the selected protocol in use. While sending the packets, the user station keeps monitoring the transmission; it stops transmission, aborting the collided packets and sending out a jamming signal, as soon as it detects a collision. The retransmission back-off procedure is initiated immediately after detecting a collision. The purpose of the jamming signal is to force consensus among users as to the state of the network, in that it ensures that all other stations know of the collision and go into back-off condition. Design of a proper back-off algorithm to ensure stable operation of the network is an important topic for communications design engineers.

Carrier-Sense Multiple Access with Collision Avoidance. The carrier-sense multiple access with collision avoidance (CSMA/CA) technique is widely used in many WLANs. The specific collision avoidance strategy for CSMA/CA is different from one manufacturer to another. In one system, CSMA/CA is referred to as CSMA with an exponential back-off strategy and an acknowledgment scheme. Note that the exponential back-off strategy is referred to as a collision avoidance mechanism. Other systems can employ R-ALOHA as the collision avoidance strategy.

Data-Sense Multiple Access. Digital or data sense multiplex access (DSMA) is commonly used in full-duplex wireless data communication networks such as CDPD and trans-European trunked radio (TETRA) (6). In these systems, communications from the mobile to base (also referred to as reverse channel or uplink) and from base to mobile (also referred to as forward channel or downlink) are performed on different frequency channels using different access techniques. The downlink uses TDMA, while the uplink uses DSMA. Interleaved among other signals broadcast on the downlink, the base station transmits a busy–idle bit in each time frame to report the status of the uplink channel. A mobile terminal will check this flag bit before transmission. If this bit indicates idle channel, the terminal proceeds to send its packet in the following time slot. As soon as the transmission starts, the base station switches the flag bit to busy state until the transmission from the mobile terminal is completed.

Polling Technique

The polling technique is a form of "control" random-assignment multiple access. In systems using this technique one station is used as a controller that periodically polls all the other stations to determine if they have data to transmit (17). Note that in R-ALOHA the control is distributed among all user terminals, while the polling technique utilizes centralized control.

Based on Refs. 6 and 20, a brief description of this technique is given here. Usually the controller station in the system is given a polling, instructing the order in which the terminals are poled. If the polled station has something to transmit, it starts transmission. If not, a "negative reply" or "no reply" is detected by the controller, which then polls the next terminal in the sequence. This technique is efficient only if (1) the round-trip propagation delay is small (due to constant exchange of control messages between the controller and terminals), (2) the overhead due to the polling message is low, and (3) the user population is not large and bursty.

Token Passing

This technique is another form of controlled random-assignment multiple access and it has been used widely in wired local area networks (LANs) for connecting computers. However, this scheme is not very popular in wireless networks. In this system a ring or loop topology is used. Figure 13 illustrates a typical token-ring network (1, Fig. 9.42, p. 529). As shown in Fig. 13, in the token-ring network, messages are passed from station to station along unidirectional links, until they return to the original station. This scheme passes the access privilege sequentially from station to station around the ring. Any station with data to send may, upon receiving

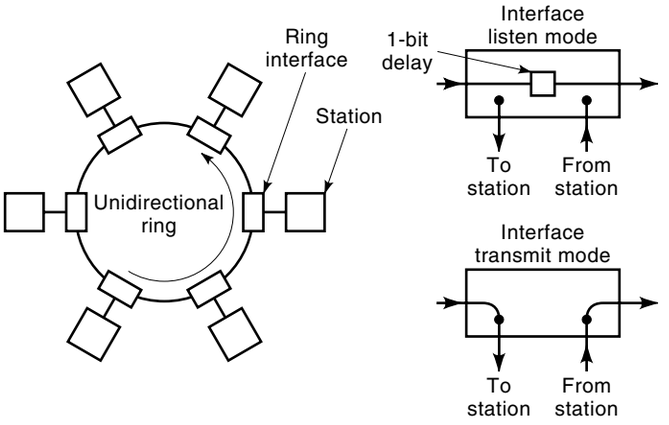


Figure 13. Illustration of the token-ring network.

the token, remove the token from the ring, send its message, and then pass on the control token.

PERFORMANCE COMPARISON OF MULTIPLE ACCESS TECHNIQUES

This section compares various standard mutiple access techniques such as FDMA, TDMA, pure ALOHA, slotted ALOHA, unslotted/slotted 1-P CSMA, and unslotted/slotted NP CSMA.

FDMA versus TDMA

Comparison between FDMA and TDMA schemes is not straightforward. It involves several issues such as bit error rate (BER) performance, throughput performance, system delay, and implementation. Some of these issues have greater or lesser performance depending on the type of system in which the access method is to be employed. In this section we briefly describe some of the major issues of comparison between FDMA and CDMA.

Bit Rate Capability. If one neglects all overhead elements such as guard bands in FDMA and guard time in TDMA, then the data rate capability for both systems is identical. The effective data rate is given by

$$R = \frac{Mb}{T} \tag{11}$$

where M is the number of disjoint channels and b is the number of data bits transmitted over T seconds.

Message Packet Delay. The message packet delay is defined as the packet waiting time before transmission plus packet transmission time. If we let M be the number of users generating data at a constant uniform rate of R/M bits/s, and use FDMA and TDMA systems that each transmit a packet of N bits every T seconds, then one can show that the average packet delays for FDMA and TDMA, respectively, are given by (6)

$$\text{Delay}_{\text{FDMA}} = T \tag{12}$$

$$\text{Delay}_{\text{TDMA}} = \text{Delay}_{\text{FDMA}} - \frac{T}{2}\left(1 - \frac{1}{M}\right) \tag{13}$$

Therefore, based on Eq. (13), TDMA is superior to FDMA with respect to the average delay packet when there are two or more users. It is interesting to note that for larger numbers of users the average delay packet of TDMA is half that of FDMA.

Spurious Narrowband Interference. An FDMA system outperforms a TDMA system in the presence of spurious narrowband interference. In an FDMA system, where the format is a single user per channel per carrier, the narrowband interference can impair the performance of only one user channel. On the other hand, in a TDMA system, a narrowband interference signal can cause performance degradation to all user channels in the TDMA data stream.

Pure ALOHA versus Slotted ALOHA

If one defines the throughput S as the number of successfully delivered packets per packet transmission time T_p, and G is the offered traffic load in packets per packet time, then the throughputs for P-ALOHA and S-ALOHA, respectively, are given by (1,6)

$$S_{\text{P-ALOHA}} = Ge^{-2G} \tag{14}$$

$$S_{\text{S-ALOHA}} = Ge^{-G} \tag{15}$$

The maximum throughput S occurs at

$$S_{\text{P-ALOHA}}(\max) = \frac{1}{2e} = 0.18 \tag{16}$$

$$S_{\text{S-ALOHA}}(\max) = \frac{1}{e} = 0.37 \tag{17}$$

at the values of $G = 0.5$ and 1, for P-ALOHA and S-ALOHA, respectively. This means that for a P-ALOHA channel, only 18% of the communication resource can be utilized. Comparing Eqs. (16) and (17), we see that, for S-ALOHA, there is an improvement of two times the P-ALOHA. A plot of P-ALOHA (or ALOHA) and S-ALOHA is shown in Fig. 14 (6, Fig. 11.19, p. 473).

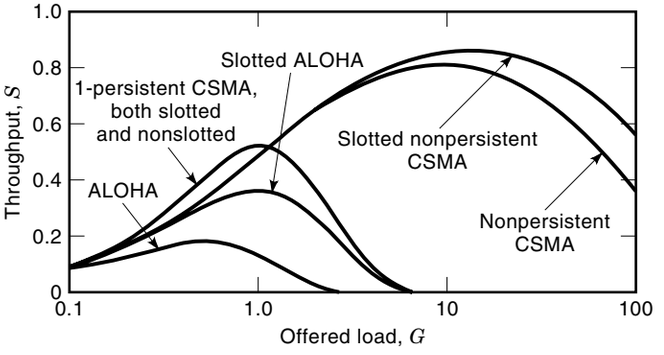


Figure 14. Throughput performance comparison of multiple access techniques.

1-P CSMA versus NP CSMA

Again, using the same definition for the throughput, and letting a be the normalized propagation delay, we have

$$a = \frac{\tau}{T_{\mathrm{p}}} \tag{18}$$

The parameter a described here corresponds to the time interval, which is normalized to the packet duration, during which a transmitted packet can suffer a collision in the CSMA schemes. Note that practical values of a on the order of 0.01 are usually of interest. The throughput for unslotted 1-P CSMA is found to be (6)

$$S_{\mathrm{Unslot-1P}} = \frac{G\left[1 + G + aG\left(1 + G + \dfrac{aG}{2}\right)\right] e^{-G(1+2a)}}{G(1+2a) - (1 - e^{-aG}) + (1 + aG)e^{-G(1+a)}} \tag{19}$$

For slotted 1-P CSMA,

$$S_{\mathrm{Slot-1P}} = \frac{G[1 + a - e^{aG}]e^{-G(1+a)}}{(1+a)(1 - e^{-aG}) + ae^{-G(1+a)}} \tag{20}$$

For unslotted NP-CSMA,

$$S_{\mathrm{Unslot-NP}} = \frac{Ge^{-aG}}{G(1+2a) + e^{-aG}} \tag{21}$$

For slotted NP-CSMA,

$$S_{\mathrm{Slot-NP}} = \frac{aGe^{-aG}}{1 + a - e^{-aG}} \tag{22}$$

The plots of Eqs. (19), (20), (21), and (22), for $a = 0.01$, are shown in Fig. 14 (6, Fig. 11.19, p. 473). This figure shows that, for low levels of offered traffic, the persistent protocols provide the best throughput, but for higher load levels, the nonpersistent protocols are by far the best. The figure also shows that the slotted NP-CSMA protocol has a peak throughput almost twice that of 1-P CSMA schemes.

APPLICATIONS OF RANDOM-ACCESS TECHNIQUES IN CELLULAR TELEPHONY

The objective for earlier mobile radio systems was to achieve a large coverage area by using a high-powered transmitter with antenna on a tall tower to extend the receiving area. The extensive coverage from this approach has also resulted in limited user capacity capability, since increasing frequency reuse would certainly increase interference for the users of the system. At the same time, government regulatory agencies are not able to allocate frequency bands in a timely manner to keep up with demand for wireless services. It is therefore necessary to construct a mobile radio system to achieve both high capacity and large coverage area with the constraint of a crowded radio spectrum.

Cellular Communications Concept

The cellular communications concept was developed to provide the solution for spectral congestion and user capacity by replacing the single high power transmitter representing a large cell with several small low powered transmitters as in small cells. Figure 15 (21, Fig. 2.1, p. 27) illustrates the arrangement of the smaller cells to achieve frequency reuse in the allocated frequency band where cells labeled with the same letter use the same group of channels. The hexagonal shape of each cell serves to model the conceptual and idealistic boundary of each cell in terms of coverage and would be much more irregular in a real environment due to differing propagation effects and practical consideration in base station placement.

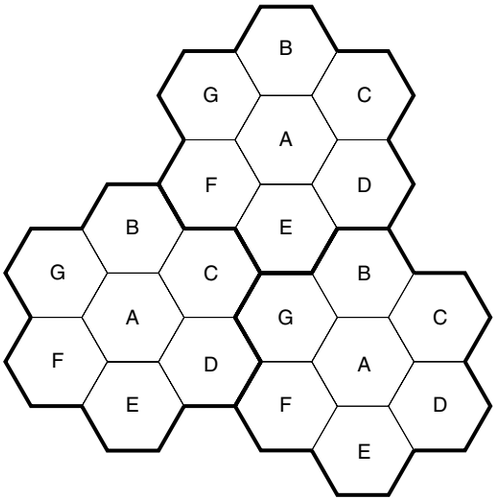


Figure 15. Illustration of the cellular frequency reuse concept. Cells with the same letter use the same set of frequencies. A cell cluster is outlined in bold and replicated over the coverage area.

Cellular Telephone System Terminology. Figure 16 (22, Fig. 1.5, p. 15) shows a basic cellular telephone system consisting of *mobile stations, base stations,* and a *mobile switching service center* (MSC), sometimes called a *mobile telephone switching office* (MTSO). The function of the MSC is to provide connectivity to all *mobile units* to the *public switched telephone network* (PSTN) in a cellular system. Each mobile unit communi-

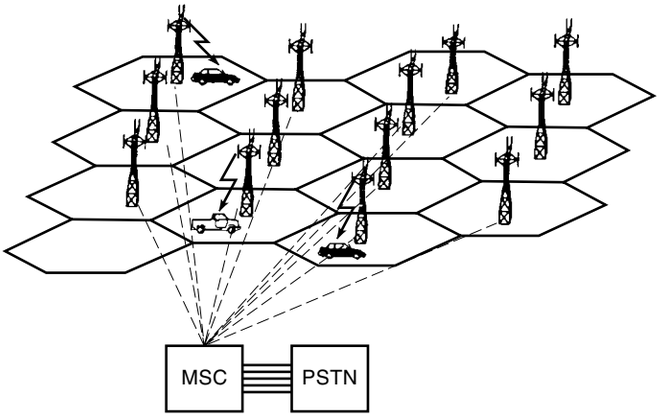


Figure 16. Illustration of a cellular system. The towers represent base stations, which provide radio access between mobile users and the mobile switching center.

cates with the base and may be handed off to any other base stations during the call.

The mobile unit handset contains a *transceiver,* antenna, and control unit, whereas a base station consists of several transmitters and receivers to handle simultaneous full duplex calls. The base station typically consists of a tower, with multiple antennas for receiving transmitting RF signals, and associated electronics at the base. The communication lines between the base station and the MSC can be regular telephone and point-to-point microwave links. The typical MSC handles the routing, billing, and system maintenance functions of calls to and from the mobile units, and multiple MSCs can be used together by a wireless operator.

The communication between the base station and mobile units is defined by a standard common air interface (CAI). The CAI typically specifies the communication parameters, such as multiple access methods and modulation type, and the use of four different channels for data transmission. From the base station to the mobile unit, the forward voice channel (FVC) is used for voice transmission and the forward control channel (FCC) is used for initiating and controlling mobile calls. From the mobile unit to the base station, the reverse voice channel (RVC) and the reverse control channel (RCC) accomplish the same functionality as the forward channel, only in the other direction to ensure full duplex communications.

All cellular systems provide *roaming* service for a cellular subscriber who uses the mobile unit in a service area other than the one area subscribed to by the mobile user. The registration of a roamer is accomplished by the MSC using the FCC to ask for all mobile units, which are not registered to report their MIN, and ESN reported over the RCC. This information is then used for validation as well as billing purposes.

The Process of a Cellular Call. When a mobile unit is first powered up, it scans for a group of forward control channels to find the strongest available one to lock on and changes to another channel when the signal level drops below a specified level. The control channels are standardized over a geographic area. The standard ensures that the mobile unit will be using the same control channel when ready to make a phone call.

Upon initiating a phone call on the reverse control channel using the subscriber's telephone number (*mobile identification number* or MIN), *electronic serial number* (ESN), called telephone number, and other control information, the base station relays this information to the MSC, which validates the request and makes the connection to the called party through the PTSN or through another MSC in the case of a called mobile unit. Once the appropriate full duplex voice channels are allocated, the connection is established as a phone call.

For a call to a mobile from a PSTN phone, the MSC dispatches the request to all base stations in the cellular system. Then the base stations, using a paging message, broadcast the called telephone number (or MIN) over the forward control channel. When the mobile unit receives the paging message, it responds to the base station by identifying itself over the reverse control channel. The base station relays this information to the MSC, which then sets up the appropriate voice channels and connection for the call.

Overview of Cellular Systems

Since the world's first cellular system was implemented by Nippon Telephone and Telegraph (NTT) and deployed in Japan in 1979, many other systems have been developed in other countries. Tables 2–4 list the cellular systems in three major geographical areas of the world—North America, Europe, and Japan.

In the United States, the Advance Mobile Phone System (AMPS) was introduced in 1983 by AT&T as the first major analog cellular system based in FDMA technology. By 1991, the TIA (Telecommunication Industry Standard) IS-54B digital standard was developed to allow US cellular operators to ease the transition of analog cellular phone to an all digital system using TDMA. To increase capacity in large AMPS markets, Motorola developed the narrowband AMPS (N-AMPS) that essentially provides three users in the 30 kHz bandwidth AMPS standard and thus reduces voice quality. By 1993, a cellular system based on CDMA was developed by Qualcomm Inc. and standardized as TIA IS-95. At the same time as IS-95, cellular digital packet data (CDPD) was introduced as the first data packet switching service that uses a full 30 kHz AMPS channel on a shared basis and utilizes slotted CSMA/CD as the channel access method. The auction of the 1900 MHz PCS band by the US government in 1995 opens the market for other competing cellular standards, such as the popular European GSM standard, which is implemented in the DCS-1900 standard.

In the United Kingdom, the E-TACS (Extended European Total Access Cellular System) was developed in 1985 and is virtually identical to the US AMPS system except for the smaller voice channel bandwidth. The Nordic Mobile Telephone (NMT) system in the 450 MHz and 900 MHz bands was developed in 1981 and 1986 using FDMA technology and was deployed in the Scandinavian countries. In Germany, a cellular standard called C-450 was introduced in 1985. Because of the need to standardize over these different cellular systems in Europe, the GSM (Global System for Mobile) was first deployed in 1991 in a new 900 MHz band dedicated as the cellular frequency band throughout Europe.

In Japan, JTACS and NTACS (Narrowband and Japanese Total Access Communications System) are analog cellular systems similar to AMPS and NAMPS. The Pacific Digital Cellular (PDC) standard provides digital cellular coverage using a system similar to North America's IS-54.

Major Cellular Systems

Currently, only a few of the cellular standards have survived or been developed into major systems around the world in terms of the number of users. These major systems are briefly described in this section.

AMPS and ETACS. In AMPS and ETACS, the FCC (forward control channel) continuously transmits control messages data at 10 kbit/s (8 kbit/s for ETACS) using binary FSK with a spectral efficiency of 0.33 bit/s/Hz. When a voice call is in progress, three in-band SATs (supervisory signal tones) at 5970 Hz, 6000 Hz, or 6030 Hz serve to provide a handshake between the mobile unit and base station. Other control signals are bursty signaling tone (ST) on the RVC to indicate end of call, and blank-and-burst transmission in the voice

Table 2. Cellular Standards in North America

Standard	Year of Introduction	Multiple Access Technique	Frequency Band (MHz), Reverse/Forward	Data/Control Parameters	Channel Bandwidth (kHz)
AMPS	1983	FDMA	824–849/869–894	FM/10 kbps FSK	30
IS-54	1991	TDMA	824–849/869–894	48.6 kbps π/4DQPSK/ 10 kbps FSK	30
NAMPS	1992	FDMA	824–849/869–894	FM/10 kbps FSK	10
CDPD	1993	FH/Packet	824–894	GMSK (BT = 0.5) 19.2 kbps	30
IS-95	1993	CDMA	824–894, 1.8–2.0 GHz	QPSK/BPSK	1.25
DCS-1900 (GSM)	1994	TDMA	1.85–1.99 GHz	GMSK	200

band having a duration less than 100 ms so as not to affect voice quality.

Prior to frequency modulation, voice signals are processed using a compander, a pre-emphasis filter, a deviation limiter, and a postdeviation limiter filter. These steps are taken to accommodate a large speech dynamic range, to prevent spurious emission, and to minimize interference with the in-band SAT signal. The channel coding on the forward and reverse control channels is BCH(40, 28) on FCC and BCH(48,36) on RCC. The line code used is Manchester.

IS-54. The analog AMPS system was not designed to support the demand for large capacity in large cities. Cellular systems using digital modulation techniques potentially offer large improvements in capacity and system performance. The IS-54 standard, also known as the USDC (US Digital Cellular) was set up to share the same frequencies, the frequency reuse plan, and base stations as AMPS so that both base stations and subscriber units can be provided with both AMPS and USDC channels within the same equipment. This way, US cellular carriers would be able to gradually replace analog phone and base stations with digital ones.

To maintain compatibility with AMPS phones, USDC forward and reverse control channels use exactly the same signaling techniques as AMPS while USDC voice channels use $\pi/4$ DQPSK at a rate of 48.6 kbit/s and spectral efficiency of 1.62 bit/s/Hz.

The numbers of USDC control channels are doubled from AMPS to provide flexibility to service offerings such as paging. There are three types of supervisory channels: the coded digital verification color code (CDVCC), whose function is similar to the SAT in AMPS, and the slow associated control channel (SACCH) and fast associated control channel (FACCH), which carry various control messages to effect power control and call processing.

The USDC voice channel occupies the 30 kHz bandwidth in each of the forward and reverse links and uses a TDMA scheme with six time slots to support a maximum of three users. For full-rate speech, each user is assigned two time slots in an equally spaced fashion as compared to one slot per user for half-rate speech.

The speech coder used in IS-54 is called the vector sum excited linear predictive (VSELP) code and is based on a code book that determines how to quantize the residual excitation signal. The VSELP coder has an output bit rate of 7950 bps and can produce a speech frame every 20 ms. The 159 bits of speech within a speech frame are divided into two classes according to their perceptual importance. Class 1 of 77 bits, being more important, are error protected using a rate $\frac{1}{2}$ convolutional code of constraint length $K = 6$, in addition to using a 7 bit CRC error detection code on the 12 most significant bits. Before transmission, the encoded speech data are interleaved over two time slots with the speech data from adjacent frames. For demodulation, differential detection may be performed at IF or base band, and equalization is needed based on training pattern imbedded in the data.

The IS-136 standard (formerly known as IS-54 Rev. C), recently introduced, is an improved version of IS-54. This standard comes with the addition of a DQPSK digital control channel to the existing FSK control channel, a greatly improved digital speech coder, new cellular features, and protocol additions to allow greater mobility management and better cellular service. The IS-136 protocol can be used in both the 800 MHz cellular band and the 1900 MHz PCS.

Global Mobile System. Global Mobile System or GSM utilizes two bands of 25 MHz set aside for system use in all

Table 3. Cellular Standards in Europe

Standard	Year of Introduction	Multiple Access Technique	Frequency Band (MHz), Reverse/Forward	Data/Control Parameters	Channel Bandwidth (kHz)
NMT-450	1981	FDMA	453–457.5/463–467.5	FM/10 kbps FSK	25
E-TACS (UK)	1985	FDMA	872–905/917–950	FM/10 kbps FSK	25
C-450 (Germany, Portugal)	1985	FDMA	450–455.74/460–465.74	FM	20/10
NMT-900	1986	FDMA	890–915/935–960	FM/10 kbps FSK	12.5
GSM	1990	TDMA and slow FH	890–915/935–960	GMSK (BT = 0.3)	200
DCS-1800	1993	TDMA	1710–1785/1805–1880	GMSK	200

Table 4. Cellular Standards in Japan

Standard	Year of Introduction	Multiple Access Technique	Frequency Band (MHz), Reverse/Forward	Data/Control Parameters	Channel Bandwidth (kHz)
NTT	1979	FDMA	400/800	FM	25
JTACS	1988	FDMA	860–925	FM/10 kbps FSK	25
PDC	1993	TDMA	810–830 1429–1453/940–960 1477–1501	$\pi/4$ DQPSK/10 kbps FSK	25
NTACS	1993	FDMA	843–925	FM	12.5

member countries. The multiaccess method is a combination of TDMA and slow FH. The use of FH combined with interleaving is for mitigation of fading caused by multipath transmission or interference effects. Frequency hopping is carried out on a frame-by-frame basis, and as many as 64 different channels may be used before the hopping sequence is repeated.

The available forward and reverse frequency bands are divided into 200 kHz wide channels. There are two types of GSM channels—traffic channels (TCH), carrying digitally encoded user speech or data, and control channels (CCH), carrying signaling and synchronizing commands between the base stations and subscriber units. There are three main control channels in the GSM system—*broadcast channel* (BCH), *common control channel* (CCCH), and *dedicated control channel* (DCCH). Each control channel consists of several logical channels distributed in time to provide the necessary GSM control function (22).

Each TDMA frame has 8 time slots for up to eight users with an aggregate bit rate of up to 24.7 kbit/s per user. The modulation used is 0.3 GMSK. The following full-rate speech and data channels are supported:

- Full-rate speech channel (TCH/FS) carries the user speech digitized at the raw rate of 13 kbit/s. With GSM channel coding applied, the full-rate speech channel is sent at 22.8 kbit/s.
- Full-rate data channel for 9600 bit/s (TCH/F9.6) carries raw user data sent at 9600 bit/s. With GSM forward error correction coding, this 9600 bit/s data is sent at 22.8 kbit/s.
- Full-rate data channel for 4800 bit/s (TCH/F4.8) carries raw user data sent at 4800 bit/s. With GSM forward error correction coding, this 4800 bit/s data is sent at 22.8 kbit/s.
- Full-rate data channel for 2400 bit/s (TCH/F2.4) carries raw user data sent at 2400 bit/s. With GSM forward error correction coding, this 2400 bit/s data is sent at 22.8 kbit/s.
- Half-rate speech channel (TCH/HS) carries the user speech digitized at half the rate of full-rate speech. With GSM channel coding applied, the full-rate speech channel is sent at 11.4 kbit/s.
- Half-rate data channel for 4800 bit/s (TCH/H4.8) carries raw user data sent at 4800 bit/s. With GSM forward error correction coding, this 4800 bit/s data is sent at 11.4 kbit/s.
- Half-rate data channel for 2400 bit/s (TCH/H2.4) carries raw user data sent at 2400 bit/s. With GSM forward error correction coding, this 2400 bit/s data is sent at 1.4 kbit/s.

The GSM speech code is based on the residually excited linear predictive (RELP) coding, which is enhanced by including a long-term predictor. The GSM coder takes advantage of the fact that, in a normal conversation, a person speaks less than 40% of the time on average. By incorporating a voice activity detector (VAD), the GSM system operates in a discontinuous transmission mode, thus providing longer subscriber battery life and reduced radio interference when the transmitter is not active during the speech silent period. Channel coding for speech and control channels is based on a rate $\frac{1}{2}$ convolutional encoder with constraint length $K = 5$, whereas channel coding for data channels is based on a modified CCITT V.110 modem standard.

Security is built into GSM by ciphering the contents of the data block with encryption keys known only to the base station and the subscriber unit and is further enhanced by changing encryption algorithm from call to call.

IS-95. Similar to IS-54, TIA IS-95 is designed to be compatible with the existing US analog system, where base stations and mobile units can work in the dual mode operation. Since this is a direct-sequence CDMA system, the need for frequency planning within a region is virtually eliminated.

Specification for IS-95 reverse link operation is in the 824–849 MHz band and forward link operation is in the 869–894 MHz band. The maximum user data rate is 9600 bps and is spread to a channel chip rate of 1.2288 Mchips/s using a combination of techniques. Each mobile subscriber is assigned a different spreading sequence to provide perfect signal separation from other users.

Unlike other cellular standards, the user data rate but not the channel chip rate changes in real-time depending on the voice activity and network requirement. On the forward link, the base station simultaneously transmits user data for (or broadcast to) all mobile subscribers by using a different spreading code (Walsh functions) for each subscriber. A pilot code is also broadcast at a high power level to allow all mobiles to perform coherent carrier detection while estimating the channel condition. On the reverse link, all mobiles would respond asynchronously and have a constant signal level due to power control exercised by the base station to avoid the "near–far problem" arising from different received power levels.

The user data stream on the reverse link is first convolutionally coded with a $\frac{1}{3}$ rate code. After interleaving, each block of six encoded symbols is mapped to one of the 64 orthogonal Walsh functions to provide 64-ary orthogonal signal-

ing. A final fourfold spreading, giving a data rate of 1.2288 Mchips/s, is achieved by user specific codes having periods of $2^{42} - 1$ chips and base station specific codes having period of 2^{15}. For the forward traffic channels, Table 5 summarizes the modulation parameters for different data rates.

Note that Walsh functions are used for different purposes on the forward and reverse channels. On the forward channels, Walsh functions are used for spreading to indicate a particular user channel, whereas on the reverse channel, Walsh functions are used for data modulation.

The speech encoder exploits gaps and pauses to reduce the output from 9600 bps to 1200 bps during the silent period. Rates lower than 9600 bps are repeated to achieve a constant coded rate of 19,200 symbols per second for all possible information data rates.

At both the base station and subscriber unit, RAKE receivers (1) are used to combine the delayed replica of the transmitted signal and therefore reduce the degree of fading. In IS-95, a three finger RAKE receiver is used at the base station.

Cellular Digital Packet Data. There are a number of wide-area packet-switched data services being offered over a dedicated network using the specialized mobile radio (SMR) frequency band near 800/900 MHz, for example, ARDIS (Advanced Radio Data Information Service) and RAM Mobile Data System. However, CDPD is the packet-switched network that uses the existing analog cellular network such as AMPS. CDPD occupies the voice channel on a secondary, noninterfering basis by utilizing the unused airtime between channel assignment by the MSC, which is estimated to be 30% of the time. CDPD supports broadcast, dispatch, electronic mail, and field monitoring applications. Figure 17 (2, Fig. 14.2, p. 361) illustrates a typical CDPD network.

The CDPD network has three interfaces: the *air link interface* (A-Interface), the *external interface* (E-Interface) *for external network interface,* and the *inter-service provider interface* (I-Interface) for cooperating CDPD service providers. The mobile subscribers (M-ES) are able to connect through the *mobile data base stations* (MDBS) to the Internet via the *intermediate systems* (MD-IS and IS), which act as servers and routers for the subscribers. Through the I-Interface, CDPD can carry either Internet protocol (IP) or OSI connectionless protocol traffic.

In CDPD, the forward channel serves as a beacon and transmits data from the PSTN side of the network while the reverse channel serves as the access channel and links all the mobile subscribers to the CDPD network. Collisions result when many mobile subscribers attempt to access the channel simultaneously and are resolved by slotted DSMA/CD.

Table 5. Summary of Forward Traffic Channel Modulation Parameters

Parameter	Data Rate (bps)			
User data rate	9600	4800	2400	1200
Coding data rate	1/2	1/2	1/2	1/2
Data repetition period	1	2	4	8
Baseband coded data rate	19,200	19,200	19,200	19,200
PN chips/coded data bit	64	64	64	64
PN chip rate (Mcps)	1.2288	1.2288	1.2288	1.2288
PN chips/bit	128	256	512	1024

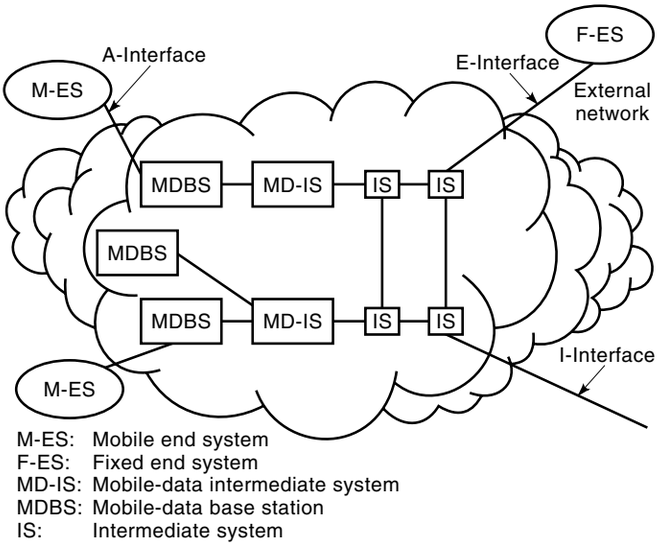


Figure 17. Cellular digital packet data network.

At the physical layer, CDPD transmissions are carried out using fixed-length blocks. The channel coding used is Reed–Salomon (63,47) block code with 6 bit symbols. For each packet, 282 bits are encoded into 378 bit blocks and provide correction for up to eight symbols. At the OSI layer 2, the mobile data link protocol (MDLP) is used to convey information between the data link layer across the common air interface. The MDLP also provides logical data link connection, sequence control, error detection, and flow control. The radio resource management protocol (RRMP) is a layer 3 function used for the management of radio resources, base station identification and configuration, channel hopping, and handoffs.

BIBLIOGRAPHY

1. B. Sklar, *Digital Communications—Fundamentals and Applications,* Englewood Cliffs, NJ: Prentice-Hall, 1988.
2. V. K. Garg and J. E. Wilkes, *Wireless and Personal Communications Systems,* Upper Saddle River, NJ: Prentice-Hall, 1996.
3. K. Feher, *Digital Communications—Satellite / Earth Station Engineering,* Englewood Cliffs, NJ: Prentice-Hall, 1981.
4. J. B. Anderson, T. Aulin, and C.-E. Sunberg, *Digital Phase Modulation,* New York: Plenum, 1986.
5. A. A. M. Saleh, Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers. *IEEE Trans. Commun.,* **COM-29**: 1715–1720, 1981.
6. K. Pahlavan and A. H. Levesque, *Wireless Information Networks,* New York: Wiley, 1995.
7. C. E. Cook et al. (eds.), *Spread-Spectrum Communications,* Piscataway, NJ: IEEE Press, 1983.
8. T. M. Nguyen, Optimize PSK systems through direct-sequence techniques. *Microwaves RF,* **24** (1): 118–126, 1985.
9. G. C. Hess, *HandBook of Land-Mobile Radio System Coverage,* Norwood, MA: Artech House, 1998.
10. W. C. Lee, *Mobile Communications Design Fundamentals,* New York: Wiley, 1993.
11. J. Litva and T. Lo, *Digital Beamforming in Wireless Communications,* Norwood, MA: Artech House, 1996.

12. L. Kleinrock and F. A. Tobagi, Carrier sense multiple access for packet switched radio channels, *Proc. IEEE ICC'74,* 1974.
13. B. Agee, Blind separation and capture of communication signals using a multitarget constant modulus beamformer, *Proc. 1989 IEEE Military Commun. Conf.,* 1989, pp. 19.2.1–19.2.7.
14. B. Agee, S. V. Schell, and W. A. Gardner, Spectral self coherence restoral: A new approach to blind adaptive signal extraction using antenna arrays, *Proc. IEEE,* **78**: 753–767, 1990.
15. R. Gooch and B. Sublett, Joint spatial temporal in a decision directed adaptive antenna system, *Proc. 23rd Asilomar Conf.: Signals, Systems, and Computers,* Noorwijk, The Netherlands, 1989.
16. W. C. Y. Lee, *Mobile Communications Engineering,* New York: McGraw-Hill, 1982.
17. N. Abramson, The ALOHA system—another alternative for computer communications, *Proc. Fall Joint Comput. Conf. AFIPF,* **37**: 1970.
18. N. Abramson, Packet switching with satellites, *Proc. Fall Joint Comput. Conf. AFIPF,* **42**: 1973.
19. W. Crowther et al., A system for broadcast communication: reservation ALOHA, *Proc. 6th Hawaii Int. Conf. Syst. Sci.,* 1973.
20. F. A. Tobagi, Multiaccess protocols in packet communication systems, *IEEE Trans. Commun.,* **COM-28**: 468–488, 1980.
21. T. S. Rappaport, *Wireless Communications Principles and Practice,* Upper Saddle River, NJ: Prentice-Hall, 1996.
22. L. Hanzo, The Pan-European Cellular System, in J. D. Gibson (ed.), *The Mobile Communications Handbook,* Boca Raton, FL: CRC Press, 1996.

TIEN M. NGUYEN
The Aerospace Corporation

HUNG NGUYEN
Mountain Technology Inc.

BOI N. TRAN
The Boeing Company

} { { } }

Wiley Encyclopedia of Electrical and Electronics Engineering

Information Theory of Radar and Sonar Waveforms
Standard Article

Mark R. Bell[1]
[1]Purdue University, West Lafayette, IN

DOI: 10.1002/047134608X.W4218
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (410K)

Abstract

The sections in this article are

Matched Filter Processing

The Ambiguity Function

Radar Waveform Design

Current and Future Directions

Keywords: ambiguity function; synthetic aperture; uncertainty ellipse; resolution; range-doppler; coherent; noncoherent

INFORMATION THEORY OF RADAR AND SONAR WAVEFORMS

Radar and active sonar systems extract information about an environment by illuminating it with electromagnetic or acoustic radiation. The illuminating field is scattered by objects in the environment, and the scattered field is collected by a receiver, which processes it to determine the presence, positions, velocities, and scattering characteristics of these objects. These active pulse-echo systems provide us with tools for observing environments not easily perceived using our senses alone. The key idea in any pulse-echo measurement system is to transmit a pulse or waveform and listen for the echo. Information about the scattering objects is extracted by comparing the transmitted pulse or waveform with the received waveform scattered by the object. Many characteristics, including the delay between transmission and reception, the amplitude of the echo, and changes in the shape of the transmitted waveform, are useful in providing information about the scattering objects.

Two primary attributes characterizing the echo return in a pulse-echo system are the round-trip propagation delay and the change in the received waveform resulting from the Doppler effect. The Doppler effect induces a compression or dilation in time for the scattered signal as a result of radial target motion toward or away from the pulse-echo sensor. For narrowband signals normally encountered in radar and narrowband active sonar systems, this is well approximated by a shift in the scattered waveform's center or carrier frequency proportional to the carrier frequency and the closing radial velocity between the target and scatterer (1). For wideband signals encountered in impulse radar and wideband sonar systems, this approximation is not accurate, and the Doppler effect must be modeled explicitly as a contraction or dilation of the time axis of the received signal.

One of the chief functions of a radar or sonar system is to distinguish, resolve, or separate the scattered returns from targets in the illuminated environment. This can be done by resolving the scatterers in delay, Doppler, or both delay and Doppler. In many problems of practical importance, resolution in delay or Doppler alone is not sufficient to achieve the desired resolution requirements for the pulse-echo measurement system. In these cases, joint delay-Doppler resolution is essential. The resolution capabilities of any pulse-echo system are a strong function of the shape of the transmitted waveforms employed by the system.

In the course of early radar development, radar systems were designed to measure the delay—and hence range—to the target, or they were designed to measure the Doppler frequency shift—and hence radial velocity—of the target with respect to the radar. The waveforms used for range measurement systems consisted of very narrow pulses for which the time delay between transmission and reception could easily be measured; these systems are referred to as *pulsed radar systems*. The waveforms used in the pulsed delay measurement radars were narrow pulses, with the ability to resolve closely spaced targets determined by the narrowness of the the pulses. If the returns from two pulses overlapped because two targets were too close to each other in range, the targets could not be resolved. So from a range resolution point of view, narrow pulses were considered very desirable. However, because the ability to detect small targets at a distance depends on the total energy in a pulse, it is not generally possible to make the pulses arbitrarily narrow and still achieve the necessary pulse energy without requiring unrealistic instantaneous power from the transmitter.

As radar systems theory and development progressed, it became clear that it was not pulse width per se that determined the delay resolution characteristics of a radar waveform, but rather the bandwidth of the transmitted radar signal. As a result, waveforms of longer duration—but appropriately modulated to achieve the necessary bandwidth to meet the desired delay resolution requirements—could be employed, which would allow for both sufficient energy to meet detection requirements and sufficient bandwidth to meet delay resolution requirements. The first detailed studies of waveforms with these properties were conducted by Woodward and Davies (2).

MATCHED FILTER PROCESSING

Radar systems typically process scattered target returns for detection by filtering of the received signal with a bank of matched filters matched to various time delayed and Doppler shifted versions of the transmitted signal. It is well known that a matched filter—or the corresponding correlation receiver—provides the maximum signal-to-noise ratio of all lin-

ear time-invariant receiver filters when the signal is being detected in additive white noise. Of course, if the filter is mismatched in delay or Doppler, the response, and hence signal-to-noise ratio, of the output will no longer be maximum. While this suboptimality of mismatched filters can in some cases be detrimental (e.g., where processing constraints only allow for a small number of Doppler filters), it provides the basis for target resolution in matched filter radar. We will now see how this gives rise to the notion of the ambiguity function—a key tool in radar resolution and accuracy assessment.

Let $s(t)$ be the baseband analytic signal transmitted by the radar system. After being demodulated down to baseband, the received signal due to a scatterer with round-trip delay τ_0 and Doppler frequency shift ν_0 is

$$r(t) = s(t - \tau_0)e^{j2\pi\nu_0 t}\, e^{j\phi}$$

where $e^{j\phi}$ is the phase shift in the received carrier due to the propagation delay τ_0; hence, $\phi = 2\pi f_0\tau_0$. If we process this signal with a matched filter

$$h_{\tau,\nu}(t) = s^*(T - t + \tau)e^{-j2\pi\nu(T-t)}$$

matched to the signal

$$q(t) = s(t - \tau)e^{j2\pi\nu t}$$

and designed to maximize the signal output at time T, the matched filter output at time T is given by

$$\begin{aligned}\mathcal{C}_T(\tau, \nu) &= \int_{-\infty}^{\infty} r(t)h_{\tau,\nu}(T - t)\, dt \\ &= \int_{-\infty}^{\infty} s(t - \tau_0)e^{j2\pi\nu_0 t}\, e^{j\phi}s^*(t - \tau)e^{-j2\pi\nu t}\, dt \\ &= e^{j\phi}\int_{-\infty}^{\infty} s(u)e^{j2\pi\nu_0(u+\tau_0)}s^*(u - (\tau - \tau_0))e^{-j2\pi\nu(u+\tau_0)}\, du \\ &= e^{j\phi}e^{-j2\pi(\nu-\nu_0)\tau_0}\int_{-\infty}^{\infty} s(u)s^*(u - (\tau - \tau_0))e^{-j2\pi(\nu-\nu_0)u}du \\ &= e^{j\phi}e^{-j2\pi(\nu-\nu_0)\tau_0}\chi_s(\tau - \tau_0, \nu - \nu_0)\end{aligned}$$

where $\chi_s(\tau, \nu)$ is the *ambiguity function* of $s(t)$, defined as

$$\chi_s(\tau, \nu) = \int_{-\infty}^{\infty} s(t)s^*(t - \tau)e^{j2\pi\nu t}\, dt$$

For narrowband signals, $\nu\tau_0 \ll 1$ and $\nu_0\tau_0 \ll 1$ for all ν, ν_0, and τ_0 of interest, however, $f_0\tau_0 \gg 1$. Hence, we can write

$$\mathcal{C}_T(\tau, \nu) = e^{-j\phi}\chi_s(\tau - \tau_0, \nu - \nu_0) \tag{1}$$

Because $h_{\tau,\nu}(t)$ is a linear time-invariant filter, if we have N scatterers with scattering strengths $\mu_1, \ldots, \mu_N$, delays $\tau_1, \ldots, \tau_N$, and Doppler shifts $\nu_1, \ldots, \nu_N$, the response of $h_{\tau,\nu}(t)$ to the collection of scatterers is

$$\mathcal{C}_T(\tau, \nu) = \sum_{i=1}^{N} \mu_i e^{-j\phi_i}\chi_s(\tau - \tau_i, \nu - \nu_i)$$

where ϕ_i is the carrier phase shift in the return from the ith scatterer resulting from the propagation delay τ_i. Furthermore, if $\mu(\tau, \nu)$ describes a continuous scattering density, the response of the matched filter $h_{\tau,\nu}(t)$ to this scattering density is

$$\mathcal{C}_T(\tau, \nu) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mu(t, v)e^{-j\phi(t)}\chi_s(\tau - t, \nu - v)\, dt\, dv$$

Here, $\phi(\tau) = e^{j2\pi f_0\tau}$ is the carrier phase shift caused by the propagation delay τ. If we define $\gamma(\tau, \nu) = \mu(\tau, \nu)e^{-j2\pi f_0\tau}$, this becomes

$$\mathcal{C}_T(\tau, \phi) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \gamma(t, v)\chi_s(\tau - t, \nu - v)\, dt\, dv$$

which is the two-dimensional convolution of $\gamma(\tau, \nu)$ with $\chi_s(\tau, \nu)$, and can be thought of as the image of $\gamma(\tau, \nu)$ obtained using an imaging aperture with point-spread function $\chi(\tau, \nu)$ (3, Chap. 4), as shown in Fig 1.

THE AMBIGUITY FUNCTION

As we have seen, the ambiguity function plays a significant role in determining the delay-Doppler resolution of a radar system. The ambiguity function was originally introduced by Woodward (2), and several related but functionally equivalent forms have been used since that time. Two common forms currently used are the *asymmetric ambiguity function* and the *symmetric ambiguity function,* and they are defined as follows. The *asymmetric ambiguity function* of a signal $s(t)$ is defined as

$$\chi_s(\tau, \nu) = \int_{-\infty}^{\infty} s(t)s^*(t - \tau)e^{j2\pi\nu t}\, dt \tag{2}$$

and the *symmetric ambiguity function* of $s(t)$ is defined as

$$\Gamma_s(\tau, \nu) = \int_{-\infty}^{\infty} s(t + \tau/2)s^*(t - \tau/2)e^{-j2\pi\nu t}\, dt \tag{3}$$

The notation "$*$" denotes complex conjugation. The asymmetric ambiguity function is the form typically used by radar engineers and most closely related to the form introduced by Woodward (2). The symmetric ambiguity function is more widely used in signal theory because its symmetry is mathematically convenient and it is consistent with the general theory of time–frequency distributions (4).

The asymmetric ambiguity function $\chi_s(\tau, \nu)$ and the symmetric ambiguity function $\Gamma_s(\tau, \nu)$ are related by

$$\Gamma_s(\tau, \nu) = e^{j\pi\nu t}\chi_s(\tau, -\nu)$$

and

$$\chi_s(\tau, \nu) = e^{j\pi\nu t}\Gamma_s(\tau, -\nu)$$

so knowledge of one form implies knowledge of the other. In practice, the *ambiguity surface* $A_s(\tau, \nu)$, given by the modulus of the symmetric ambiguity function,

$$A_s(\tau, \nu) = |\Gamma_s(\tau, \nu)| = |\chi_s(\tau, -\nu)|$$

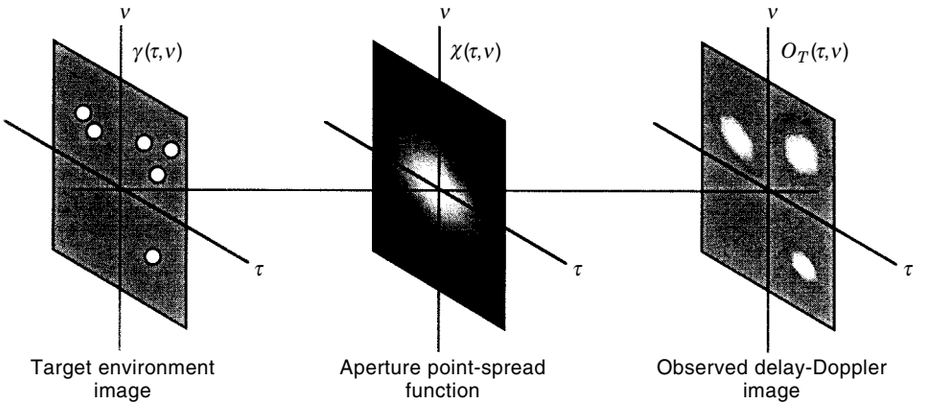


Figure 1. Imaging interpretation of a delay-Doppler pulse-echo system. A waveform $s(t)$ with ambiguity function $\chi(\tau, \nu)$ gives rise to a delay-Doppler image $\mathcal{O}_T(\tau, \nu)$ that is the convolution of the ideal image $\gamma(\tau, \nu)$ with the point-spread function $\chi(\tau, \nu)$.

is usually sufficient to characterize a waveform's delay-Doppler resolution characteristics, as it gives the magnitude of the matched filter response for a delay-Doppler mismatch of (τ, ν).

Figures 2 and 3 show ambiguity surfaces of a simple pulse

$$s_1(t) = \begin{cases} 1, & \text{for } |t| < 1/2 \\ 0, & \text{elsewhere} \end{cases}$$

and a linear FM "chirp"

$$s_2(t) = \begin{cases} e^{j\pi\alpha t^2}, & \text{for } |t| < 1/2 \\ 0, & \text{elsewhere} \end{cases}$$

(with $\alpha = 8$), respectively. The ambiguity function of $s_1(t)$ is

$$\Gamma_{s_1}(\tau, \nu) = \begin{cases} (1 - |\tau|)\text{sinc}[\nu(1 - |\tau|)], & \text{for } |\tau| \le 1 \\ 0, & \text{elsewhere} \end{cases}$$

The ambiguity function of $s_2(t)$ is

$$\Gamma_{s_2}(\tau, \nu) = \begin{cases} (1 - |\tau|)\text{sinc}[(\nu - \alpha\tau)(1 - |\tau|)], & \text{for } |\tau| \le 1 \\ 0, & \text{elsewhere} \end{cases}$$

These figures illustrate the very different delay-Doppler resolution characteristics provided by these signals when they are processed using a matched filter.

The shape or properties of the main lobe of the ambiguity surface $|\Gamma_s(\tau, \nu)|$ centered about the origin determine the ability of the corresponding waveform to resolve two scatterers close together in both delay and Doppler. The ambiguity surface squared $|\Gamma_s(\tau, \nu)|^2$ close to the origin can be expanded as a two-dimensional Taylor series about $(\tau, \nu) = (0, 0)$. From this it follows that the ambiguity surface itself may be approximated by (8, pp. 21–22)

$$|\Gamma_s(\tau, \nu)| \approx \Gamma(0, 0)[1 - 2\pi^2 T_G^2 \nu^2 - 4\pi\rho T_G B_G \tau\nu - -2\pi^2 B_G^2 \tau^2] \tag{4}$$

where

$$B_G = \sqrt{\overline{f^2} - \overline{f}^2}$$

is the *Gabor bandwidth* of the signal,

$$T_G = \sqrt{\overline{t^2} - \overline{t}^2}$$

is the *Gabor timewidth* of the signal, the frequency and time

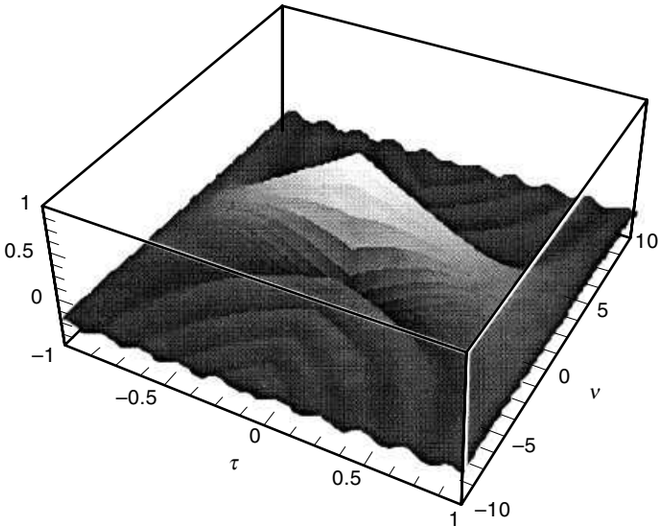


Figure 2. Symmetric ambiguity function $\Gamma_1(\tau, \nu)$ of a rectangular pulse of duration 1.

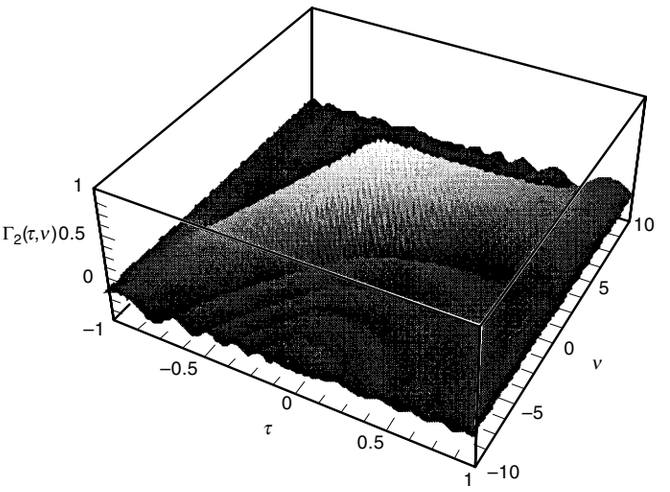


Figure 3. Symmetric ambiguity function $\Gamma_2(\tau, \nu)$ of a linear FM chirp of duration 1.

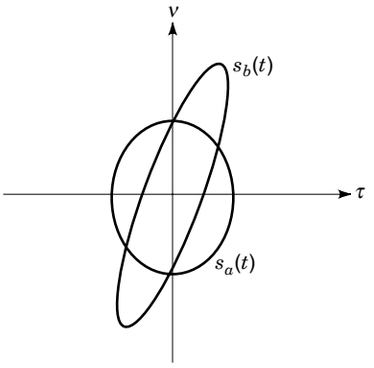


Figure 4. Uncertainty ellipses corresponding to $s_a(t) = e^{-\beta t^2}$ and $s_b(t) = e^{-\beta t^2} e^{j\pi\alpha t^2}$.

moments of the signal $s(t)$ are

$$\overline{f^n} = \frac{1}{E_s} \int_{-\infty}^{\infty} f^n |S(f)|^2 \, df$$

and

$$\overline{t^n} = \frac{1}{E_s} \int_{-\infty}^{\infty} t^n |s(t)|^2 \, dt$$

respectively, and the *skew parameter* ρ is

$$\rho = \frac{1}{TB} \operatorname{Re} \left\{ \frac{j}{2\pi E_s} \int_{-\infty}^{\infty} t\dot{s}(t) s^*(t) \, dt - \overline{t}\,\overline{f} \right\}$$

where $\dot{s}(t)$ is the derivative of $s(t)$.

The shape of the main lobe about the origin of the ambiguity function can be determined by intersecting a plane parallel to the (τ, ν) plane with the main lobe near the peak value. Using the approximation of Eq. (4) and setting it equal to the constant ambiguity surface height γ_0 specified by the intersecting plane, we have

$$\Gamma(0, 0)[1 - 2\pi^2 T_G^2 \nu^2 - 4\pi\rho T_G B_G \tau\nu - 2\pi^2 B_G^2 \tau^2] = \gamma_0$$

which we can rewrite as

$$B_G^2 \tau^2 + 2\rho B_G T_G \tau\nu + T_G^2 \nu^2 = C \tag{5}$$

where C is a positive constant. This is the equation of an ellipse in τ and ν, and this ellipse is known as the *uncertainty ellipse* of the waveform $s(t)$. The uncertainty ellipse describes the shape of the main lobe of $|\Gamma_s(\tau, \nu)|$ in the region around its peak and hence provides a concise description of the capability of $s(t)$ to resolve closely spaced targets concentrated in the main lobe region. The value of C itself is not critical, since the shape of the uncertainty ellipse is what is of primary interest. Figure 4 shows the uncertainty ellipses of a Gaussian pulse

$$s_a(t) = e^{-\beta t^2}$$

and a linear FM chirp modulated Gaussian pulse

$$s_b(t) = e^{-\beta t^2} e^{j\pi\alpha t^2}$$

While the uncertainty ellipse provides a rough means of determining the resolution performance of a waveform for resolving closely spaced targets in isolation from other interfering scatterers, it is not sufficient to completely characterize a waveform's measurement characteristics. Target returns with delay-Doppler coordinates falling in the sidelobes of the ambiguity function can have a significant effect on a radar's measurement and resolution capabilities. For this reason, in order to effectively design radar waveforms for specific measurement tasks, it is important to have a thorough understanding of the properties of ambiguity functions.

Properties of Ambiguity Functions

In order to gain a thorough understanding of the delay-Doppler resolution characteristics of various signals under matched filter processing, it is necessary to understand the general properties of ambiguity functions. With this in mind, we now consider the properties of ambiguity functions. Proofs of these properties may be found in Refs. 5 (Chap. 9), 6 (Chaps. 5–7), 7 (Chap. 4), 8, and 9 (Chap. 10).

Property 1. The energy in the signal $s(t)$ is given by

$$E_s = \Gamma_s(0, 0) = \int_{-\infty}^{\infty} |s(t)|^2 \, dt$$

Property 2 (Volume).

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |\Gamma_s(\tau, \nu)|^2 \, d\tau \, d\nu = |\Gamma_s(0, 0)|^2 = E_s^2$$

Property 3. The *time autocorrelation function* $\phi_s(\tau)$ of the signal $s(t)$ is given by

$$\phi_s(\tau) = \Gamma_s(\tau, 0) = \int_{-\infty}^{\infty} s(t + \tau/2) s^*(t - \tau/2) \, dt$$

Property 4. The energy spectrum of the signal $s(t)$ is given by

$$\Gamma_s(0, \nu) = \int_{-\infty}^{\infty} |s(t)|^2 e^{-j2\pi\nu t} \, dt$$

Property 5. The symmetric ambiguity function of the signal $s(t)$ can be written as

$$\Gamma_s(\tau, \nu) = \int_{-\infty}^{\infty} S(f + \nu/2) S^*(f - \nu/2) e^{j2\pi f\tau} \, df$$

where

$$S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi f t} \, dt$$

is the Fourier transform of $s(t)$.

Property 6. If $s(0) \neq 0$, $s(t)$ can be recovered from $\Gamma_s(\tau, \nu)$ using the relationship

$$s(t) = \frac{1}{s^*(0)} \int_{-\infty}^{\infty} \Gamma_s(t, \nu) j\pi\nu t \, dt$$

where

$$|s(0)|^2 = \int_{-\infty}^{\infty} \Gamma_s(0,\ \nu)\, d\nu$$

Property 7 (Time Shift). Let $s'(t) = s(t - \Delta)$. Then

$$\Gamma_{s'}(\tau,\ \nu) = e^{-j2\pi\nu\Delta}\Gamma_s(\tau,\ \nu)$$

Property 8 (Frequency Shift). Let $s'(t) = s(t)e^{j2\pi ft}$. Then

$$\Gamma_{s'}(\tau,\ \nu) = e^{j2\pi f\tau}\Gamma_s(\tau,\ \nu)$$

Property 9 (Symmetry). $\Gamma_s(\tau,\ \nu) = \Gamma_s^*(-\tau,\ -\nu)$.

Property 10 (Maximum). The largest magnitude of the ambiguity function is always at the origin:

$$|\Gamma_s(\tau,\ \nu)| \le \Gamma_s(0,\ 0) = E_s$$

This follows directly from the Schwarz inequality.

Property 11 (Time Scaling). Let $s'(t) = s(at)$, where $a \neq 0$. Then

$$\Gamma_{s'}(\tau,\ \nu) = \frac{1}{|a|}\Gamma_s(a\tau,\ \nu/a)$$

Property 12 (Quadratic Phase Shift). Let $s'(t) = s(t)e^{j\pi\alpha t^2}$. Then

$$\Gamma_{s'}(\tau,\ \nu) = \Gamma_s(\tau,\ \nu - \alpha\tau)$$

Property 13 (Self-transform). $|\Gamma_s(\tau,\ \nu)|^2$ is its own Fourier transform in the sense that

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |\Gamma_s(\tau,\ \nu)|^2 e^{-j2\pi f\tau}\, e^{j2\pi t\nu}\, d\tau\, d\nu = |\Gamma_s(t,\ f)|^2$$

Property 14 (Wigner Distribution). The two-dimensional inverse Fourier transform of the ambiguity function $\Gamma_s(\tau, \nu)$ of a signal $s(t)$ is its Wigner distribution $W_s(t, f)$:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Gamma_s(\tau,\ \nu)e^{j2\pi f\tau}\, e^{j2\pi t\nu}\, d\tau\, d\nu = W_s(t,\ f)$$

where the Wigner distribution of $s(t)$ is defined as (4,8)

$$W_s(t,\ f) = \int_{-\infty}^{\infty} s(t + \tau/2)s^*(t - \tau/2)e^{j2pif\tau}\, d\tau$$

These properties of the ambiguity function have immediate implications for the design of radar waveforms. From the imaging analogy of delay-Doppler measurement, where the ambiguity function plays the role of the imaging aperture, it is clear that an ideal ambiguity function would behave much like a pinhole aperture—a two-dimensional Dirac delta function centered at the origin of the delay-Doppler plane. Such an ambiguity function would yield a radar system giving a response of unity if the return had the assumed delay and Doppler, but a response of zero if it did not. Such a system would in fact have perfect delay-Doppler resolution properties. Unfortunately, such an ambiguity function does not exist. This can be seen by considering Property 1 and Property 2 of the ambiguity function. Property 1 states that the height of $|\Gamma_s(\tau,\ \nu)|^2$ at the origin is $|\Gamma_s(0,\ 0)|^2 = E_s^2$. Property 2 states that the total volume under $|\Gamma_s(\tau,\ \nu)|^2$ is E_s^2. So if we try to construct a thumbtack-like $|\Gamma_s(\tau,\ \nu)|^2$ approximating an ideal delta function, we run into the problem that as the height $|\Gamma_s(0,\ 0)|$ increases, so does the volume under $|\Gamma_s(\tau,\ \nu)|^2$. This means that for a signal with a given energy, if we try to push the volume of the ambiguity function down in one region of the delay-Doppler plane, it must pop up somewhere else. So there are limitations on just how well any waveform can do in terms of overall delay-Doppler ambiguity performance. In fact, the radar waveform design problem corresponds to designing waveforms that distribute the ambiguity volume in the $(\tau,\ \nu)$ plane in a way appropriate for the delay-Doppler measurement problem at hand. We now investigate some of these techniques.

The Wideband Ambiguity Function

In the situation that the waveforms being considered are not narrowband or the target velocity is not small compared with the velocity of wave propagation, the Doppler effect cannot be modeled accurately as a frequency shift. In this case, it must be modeled as a contraction or dilation of the time axis. When this is the case, the ambiguity functions $\chi_s(\tau,\ \nu)$ and $\Gamma_s(\tau,\ \nu)$ defined in Eqs. (2) and (3) can no longer be used to model the output response of the delay and Doppler (velocity) mismatched matched filter. In this case, the *wideband ambiguity function* must be used (10–13). Several slightly different but mathematically equivalent forms of the wideband ambiguity function have been introduced. One commonly used form (13) is

$$\Psi_s(\tau,\ \gamma) = \sqrt{|\gamma|}\int_{-\infty}^{\infty} s(t)s^*(\gamma(t - \tau))\, dt \tag{6}$$

where γ is the *scale factor* arising from the contraction or dilation of the time axis as a result of the Doppler effect. Specifically,

$$\gamma = \frac{1 - v/c}{1 + v/c}$$

where v is the radial velocity of the target with respect to the sensor (motion away from the sensor positive), and c is the velocity of wave propagation in the medium. While the theory of wideband ambiguity functions is not as well developed as for the case of narrowband ambiguity functions, a significant amount of work has been done in this area. See Ref. 13 for a readable survey of current results. We will focus primarily on the narrowband ambiguity function throughout the rest of this article.

RADAR WAVEFORM DESIGN

The problem of designing radar waveforms with good delay-Doppler resolution has received considerable attention (14–24). Waveforms developed for this purpose have generally fallen into three broad categories:

1. Phase and frequency modulation of individual radar pulses

2. Pulse train waveforms
3. Coded waveforms

We will now investigate these techniques and consider how each can be used to improve radar delay-Doppler resolution characteristics and shape the ambiguity functions of radar waveforms in desirable ways.

Phase and Frequency Modulation of Radar Pulses

The fundamental observation that led to the development of phase and frequency modulation of radar pulses was that it is not the duration of a pulse, but rather its bandwidth, that determines its range resolution characteristics. Early range measurement systems used short duration pulses to make range measurements, and narrow pulses were used to obtain good range resolution, but this put a severe limitation on the detection range of these systems, because detection performance is a function of the total energy in the transmitted pulse, and with the peak power limitations present in most real radar systems, the only way to increase total energy is to increase the pulse duration. However, if the pulse used is simply gating a constant frequency sinusoidal carrier, increasing the duration decreases the bandwidth of the transmitted signal. This observation led to the conjecture that perhaps it is large bandwidth instead of short pulse duration that leads to good range resolution. This conjecture was in fact shown to be true (14). We now investigate this using ambiguity functions.

The ambiguity function of the simple rectangular pulse

$$s_1(t) = \begin{cases} 1, & \text{for } |t| \le T \\ 0, & \text{elswhere} \end{cases}$$

of duration T is

$$\Gamma_1(\tau, \nu) = \begin{cases} (T - |\tau|)\text{sinc}[\nu(T - |\tau|)], & \text{for } |t| \le T \\ 0, & \text{elswhere} \end{cases}$$

and the ambiguity function of the linear FM "chirp" pulse

$$s_2(t) = \begin{cases} e^{j\pi\alpha t^2}, & \text{for } |t| \le T \\ 0, & \text{elswhere} \end{cases}$$

of the same duration is

$$\Gamma_2(\tau, \nu) = \begin{cases} (T - |\tau|)\text{sinc}[(\nu - \alpha\tau)(T - |\tau|)], & \text{for } |t| \le T \\ 0, & \text{elswhere} \end{cases}$$

[Note that $\Gamma_2(\tau, \nu)$ is easily obtained from $\Gamma_1(\tau, \nu)$ using Property 12 of the ambiguity function.] If we compare the time autocorrelation functions $\phi_1(\tau) = \Gamma_1(\tau, 0)$ and $\phi_2(\tau) = \Gamma_2(\tau, 0)$ for various values of the linear FM modulation index α as shown in Fig. 5, we see that, although pulse durations are equivalent (in this case we take $T = 1$), there is a significant difference in range resolution. With increasing α, we also have increasing bandwidth. Looking at the ambiguity function of the linear FM chirp shown in Fig. 3, and comparing the ambiguity function of the simple rectangular pulse in Fig. 2, it is clear that the broadening of the pulse bandwidth has brought about increased delay resolution–however, not without cost.

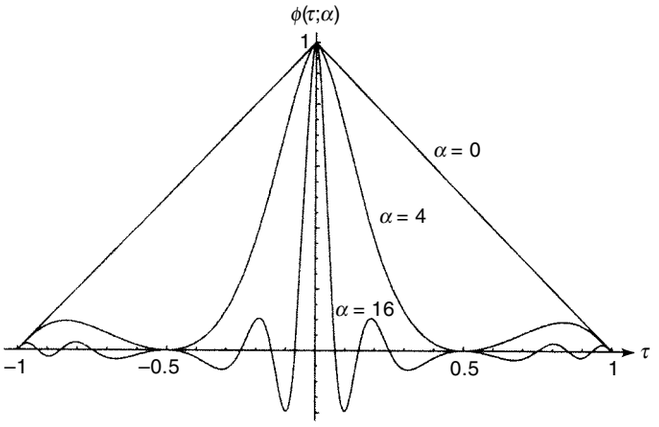


Figure 5. Time correlation $\phi(\tau, \alpha) = \Gamma_s(\tau, 0)$ for linear FM chirp pulses of duration 1 and modulation indices α of 0, 4, and 16.

From Property 12, the quadratic phase shift property, we see that the matched filter will not only have a large response to the signal with the desired delay τ and Doppler ν, but also to any signal with delay $\tau + \Delta\tau$ and Doppler $\nu + \Delta\nu$, where $\Delta\nu - \alpha\,\Delta\tau = 0$. This locus of peak response for the chirp is oriented along the line of slope α in the (τ, ν) plane. So when matched filtering for a chirp with some desired delay and Doppler shift imposed on it, we are never certain if a large response is the result of a scatterer at the desired delay and Doppler, or a scatterer with a delay-Doppler offset lying near the locus of maximal delay-Doppler response. While for a single scatterer the actual delay and Doppler can be determined by processing with a sufficiently dense band of matched filters in delay and Doppler, scatterers lying along this maximal response locus are hard to resolve if they are too close in delay and Doppler. From the point of view of detection, however, there is a benefit to this "Doppler tolerance" of the chirp waveform. It is not necessary to have a bank of Doppler filters as densely located in Doppler frequency in order to detect the presence of targets (25, Chap. 9).

Coherent Pulse Train Waveforms

Another way to increase the delay-Doppler resolution and ambiguity characteristics of radar waveforms is through the use of pulse trains—waveforms synthesized by repeating a simple pulse shape over and over. An extension of this basic idea involves constructing the pulse train as a sequence of shorter waveforms—not all the same—from a prescribed set of waveforms (26). Most modern radar systems employ pulse trains instead of single pulses for a number of reasons. Regardless of whether the pulse train returns are processed coherently (keeping track of the phase reference from pulse-to-pulse and using it to construct a matched filter) or nonco-

herently (simply summing the pulse-to-pulse amplitude of the matched filter output without reference to phase), a pulse train increases receiver output signal-to-noise ratio, and hence increases detection range [e.g., see Ref. 25 (Chaps. 6 and 8)]. Furthermore, when processed coherently in a pulse-Doppler processor, flexible, high-resolution delay-Doppler processing is possible. In discussing pulse trains, we will focus on coherent pulse-Doppler waveforms, as pulse-Doppler radar systems have become the dominant form of radar for both surveillance and synthetic aperture radar (SAR) applications.

A pulse train is constructed by repeating a single pulse $p(t)$ regularly at uniform intervals T_r; T_r is called the *pulse repetition interval* (PRI). The frequency $f_r = 1/T_r$ is called the *pulse repetition frequency* (PRF) of the pulse train. Typically, the duration τ_p of the pulse $v(t)$ is much less than T_r. A uniform pulse train $s(t)$ made up of N repeated pulses and having PRI T_r can be written as

$$x(t) = \sum_{n=0}^{N-1} p(t - nT_r)$$

A typical example of such a pulse train in which the pulse $p(t)$ repeated is a simple rectangular pulse is shown in Fig. 6. Centering this pulse train about the origin of the time axis, we can write it as

$$s(t) = \sum_{n=0}^{N-1} p(t - nT_r + (N-1)T_r/2) \tag{7}$$

The symmetric ambiguity function of this pulse train is (6,8)

$$\Gamma_s(\tau,\ \nu) = \sum_{n=-(N-1)}^{N-1} \left[\frac{\sin \pi \nu T_r(N - |n|)}{\sin \pi \nu T_r}\right] \cdot \Gamma_p(\tau - nT_r,\ \nu) \tag{8}$$

where $\Gamma_p(\tau,\ \nu)$ is the ambiguity function of the elementary pulse $p(t)$ used to construct the pulse train.

In order to gain an understanding for the behavior of the ambiguity function of the pulse train, consider the special case of a uniform pulse train of $N = 5$ rectangular pulses, each of length $\tau_p = 1$ with a PRI of $T_r = 5$. The plot of this ambiguity function is shown in Fig. 7. A similar plot in which $p(t)$ is a linear FM chirp of the form

$$p(t) = \begin{cases} e^{j\pi\alpha t^2}, & \text{for } |t| \le 1 \\ 0, & \text{elsewhere} \end{cases}$$

and $\alpha = 8$ is shown in Fig. 8. From the form of Eq. (8), we see that the ambiguity function of the pulse train has "grating lobes" centered at $(\tau,\ \nu)$ pairs given by

$$(\tau,\ \nu) = (nT_r,\ k/T_r)$$

where n is any integer with $|n| \le N - 1$, and k is any integer. From the behavior of the Dirichlet function

$$\left[\frac{\sin \pi \nu T_r(N - |n|)}{\sin \pi \nu T_r}\right]$$

weighting the delayed copies of $\Gamma_p(\tau,\ \nu)$ in Eq. (8), it is clear that the peak amplitudes of these grating lobes fall off as we move farther away from the main lobe ($n = 0$ and $k = 0$).

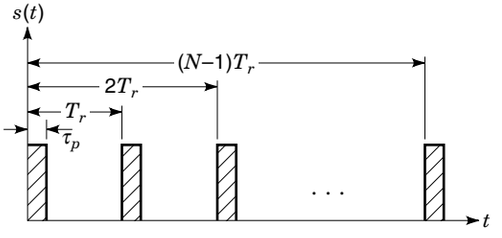


Figure 6. Uniform pulse train waveform $s(t)$ constructed by repeating a basic pulse shape $p(t)$ N times with a pulse repetition interval of T_r.

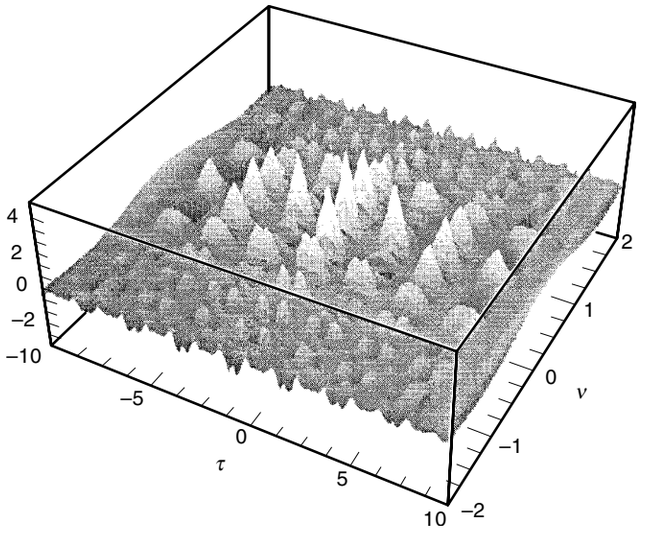


Figure 7. The ambiguity function for a uniform pulse train of rectangular pulses.

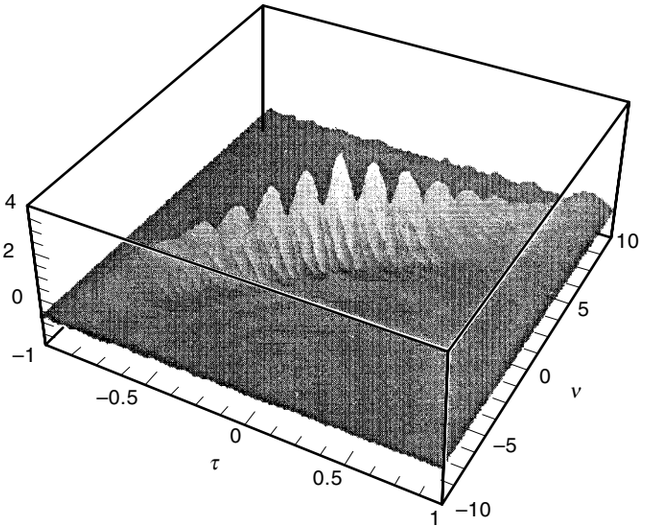


Figure 8. The ambiguity function for a uniform pulse train of linear FM chirp pulses.

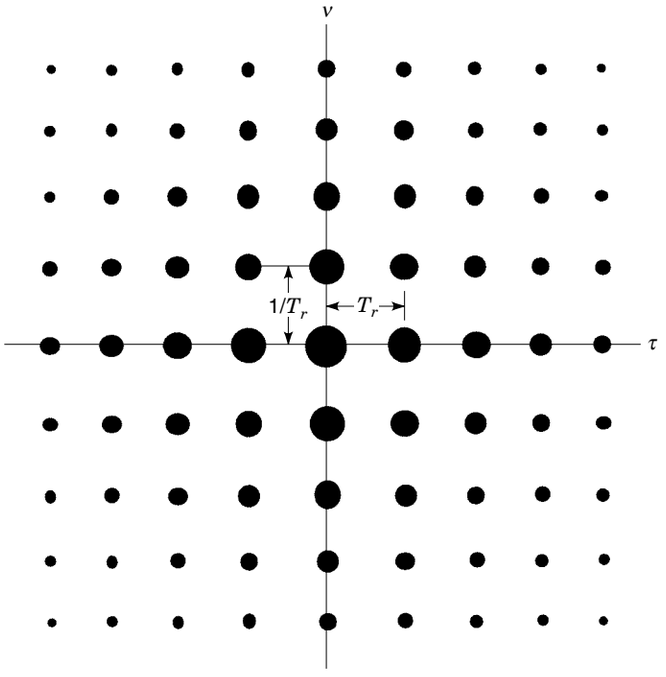


Figure 9. Locations of grating sidelobes in the ambiguity function of a uniform pulse train.

Figure 9 shows this grating lobe behavior for a uniform pulse train.

By observing the main lobe of the uniform pulse train, we see that its delay resolution is approximately τ_p—the range resolution of the elementary pulse $p(t)$—while the Doppler resolution is approximately $1/NT_r$, a value that can be made arbitrarily small by making N sufficiently large, limited only by practical considerations in coherently processing the received signal. However, the ambiguities introduced through the grating lobes at $(nT_r, k/T_r)$ can result in uncertainty in the actual delay and Doppler of the target. As a result, both the range and Doppler determined radial velocity of the target can be ambiguous. While in principle this ambiguity can be resolved in the case of a small number of targets using the fact that the sidelobes have successively smaller amplitude as we move away from the main lobe, this approach is not practical because of the way in which the bank of matched filters is actually implemented in a pulse-Doppler processor. Hence, another approach to resolving $(nT_r, k/T_r)$ ambiguity is needed. We will briefly discuss approaches that can be taken.

One way to reduce the effects of the range ambiguity is to make T_r large. This makes the delay ambiguity large, and often the delay ambiguity (and hence unambiguous measurement range) can be made sufficiently large so that range ambiguity is no longer a problem for ranges of interest. Of course, this complicates the Doppler ambiguity problem, because the pulse repetition frequency (PRF) $1/T_r$ is the effective sampling rate of the pulse-Doppler processor. A large value of T_p results in a low PRF and hence low sampling rate, and there is significant aliasing of the Doppler signal. Some systems do use this approach to deal with the ambiguity problem, using range differences (often called range rate measurements) from pulse to pulse to resolve the Doppler ambiguity; however, this approach is only successful in sparse target environments. When there are many targets in proximity in both delay and Doppler, sorting out the ambiguity becomes unwieldy. Another disadvantage of these *low PRF pulse trains* is that they have lower duty cycles for a given pulse width, resulting in a significant decrease in average transmitted power (and hence detection range) for a given elemental pulse width τ_p and peak power constraint.

At the other extreme, if one makes T_r very small, the effects of Doppler ambiguity can be minimized. In fact, if $1/T_r$ is greater than the maximum Doppler frequency shift we expect to encounter, there is no Doppler ambiguity. However, there will most likely be severe range ambiguities if such *high PRF pulse trains* are used.

For most radar surveillance problems involving the detection of aircraft and missiles, the size of the surveillance volume and the target velocities involved dictate that there will be ambiguities in both delay and Doppler, and most often a *medium PRF pulse train* is employed. In this case the PRF is usually selected to meet the energy efficiency (duty-cycle) constraints to ensure reliable detection and to make the nature of the delay-Doppler ambiguities such that they are not extreme in either the delay or Doppler dimension. In this case, delay-Doppler ambiguities can be resolved by changing the PRF from one coherent N-pulse train to the next by changing T_r from pulse train to pulse train. This technique is sometimes called *PRF staggering,* and is effective in sparse environments. As can be seen from Eq. (8), proper selection of the T_r from pulse train to pulse train makes this feasible, because in general, with proper selection of the PRIs T_r used, only the true delay-Doppler (τ, ν) will be a feasible solution for all T_r. An additional benefit of changing T_r from pulse train to pulse train is that it alleviates the "blind range" problem in monostatic radars. These radars cannot transmit and receive simultaneously. When they transmit a pulse train, the receiver is turned off during pulse transmission and is turned on to listen for target returns in the periods between pulses. Hence target returns having delays corresponding to the time intervals of successive pulse transmissions are not seen by the radar. Changing T_r from pulse train to pulse train moves the blind ranges around, ensuring nearly uniform surveillance coverage at all ranges.

Phase and Frequency Coded Waveforms

Another highly successful approach to designing waveforms with desirable ambiguity functions has been to use phase and/or frequency coding. The general form of a coded waveform (with coding in both phase and frequency) is

$$s(t) = \sum_{n=0}^{N-1} p_T(t - nT) \exp\{j2\pi d_n t/T\} \exp\{j\phi_n\} \qquad (9)$$

The coded waveform $s(t)$ consists of a sequence of N identical baseband pulses $p_T(t)$ of length T; these pulses $p_T(t - nT)$ are usually referred to as the *chips* making up the waveform $s(t)$. Usually, the chip pulse $p_T(t)$ has the form

$$p_T(t) = \begin{cases} 1, & \text{for } 0 \le t < T \\ 0, & \text{elsewhere} \end{cases}$$

Note that each chip pulse $p_T(t - nT)$ is of duration T and each successive pulse is delayed by T, so there are no empty spaces in the resulting coded waveform $s(t)$ of duration NT. In fact, for the rectangular $p_T(t)$ specified above, $|s(t)| = 1$ for all $t \in [0, NT)$. However, each pulse in the sequence is modulated by an integral frequency modulating index d_n and a phase ϕ_n that can take on any real number value. To specify the modulating frequency and phase patterns of a coded waveform, we must specify a length N sequence of frequency indices $\{d_0, \ldots, d_{N-1}\}$ and a length N sequence of phases $\{\phi_0, \ldots, \phi_{N-1}\}$. If $d_n = 0$ for $n = 0, \ldots, N - 1$, then the coded waveform is strictly phase modulated. If $\phi_n = 0$ for $n = 0, \ldots, N - 1$, then the coded waveform is strictly frequency modulated. The asymmetric ambiguity function of $s(t)$ as given in Eq. (9) is given by (26)

$$\chi_s(\tau, \nu) = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} e^{j(\phi_n - \phi_m)} e^{j2\pi(d_m/T)\tau} e^{-j2\pi\nu nT} \chi_{p_T}\left(\tau - (n-m)T, \nu - \frac{(d_n - d_m)}{T}\right) \quad (10)$$

There are many families of coded phase and frequency modulated waveforms. We will consider a few of the most interesting of these. For a more thorough treatment of coded waveforms, see Refs. 5 (Chap. 6), 6 (Chap. 8), and 7 (Chap. 8).

Frequency Coded Waveforms. Consider an N-chip frequency coded waveform with the rectangular $p_T(t)$ defined above (here we assume $\phi_0 = \cdots = \phi_{N-1} = 0$):

$$s(t) = \sum_{n=0}^{N-1} p_T(t - nT) \exp\{j2\pi d_n t/T\} \quad (11)$$

Waveforms of this kind are sometimes referred to as *frequency hopping waveforms,* because the frequency of the waveform "hops" to a new frequency when transitioning from chip to chip. Now suppose we take the sequence of frequency modulation indices to be

$$\phi_n = n, \quad n = 0, \ldots, N - 1$$

Then the resulting $s(t)$ is a stepped frequency approximation to a linear FM chirp. Here we have used each of the frequency modulation indices in the set $\{0, \ldots, N - 1\}$ once and only once. In general, we can describe the order in which the indices are used to construct the waveform using a *frequency index sequence* of the form $(d_0, \ldots, d_{N-1})$. So, for example, the stepped linear FM sequence has frequency index sequence (0, 1, 2, . . ., $N - 1$). There are of course $N!$ possible frequency coded waveforms that use each of these indices once and only once, since there are $N!$ permutations of the N elements or, equivalently, $N!$ distinct frequency index sequences. Some of these permutations give rise to waveforms with ambiguity functions that are very different from that of the stepped frequency approximation to the linear FM chirp. For the purpose of comparison, we consider two such waveforms, the 16-chip stepped linear FM waveform, and the 16-chip Costas waveform (20). Before we do this, we introduce the notion of the *sidelobe matrix.*

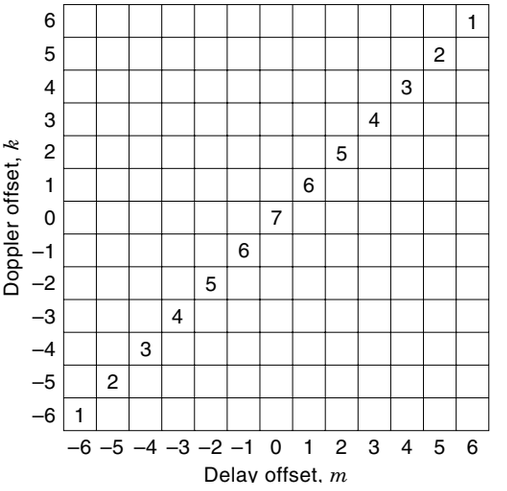


Figure 10. Ambiguity matrix of the 7-chip stepped frequency chirp having frequency index sequence (0, 1, 2, 3, 4, 5, 6).

The sidelobe matrix gives the heights of the major sidelobes of a frequency coded waveform. These can be shown to occur at locations $(\tau, \nu) = (mT, k/T)$, where m and k are integers. The sidelobe matrix is a table of the relative heights of $|\Gamma_s(mT, k/T)| = |\chi_s(mT, k/T)|$ for integer values of m and k in the range of interest. So, for example, the sidelobe matrix of a 7-chip stepped linear FM chirp having frequency index sequence (0, 1, 2, 3, 4, 5, 6) is shown in Fig. 10, whereas that for a 7-chip Costas waveform with frequency index sequence (3, 6, 0, 5, 4, 1, 2) is shown in Fig. 11. Blank entries in the sidelobe matrix correspond to zero. Clearly, there is a significant difference between the ambiguity matrices (and hence ambiguity functions) of these two frequency coded waveforms, despite the fact that they have the same duration, same number of chips, and same set of modulating frequencies. It is only the order in which the modulating frequencies are used that determines their ambiguity behavior.

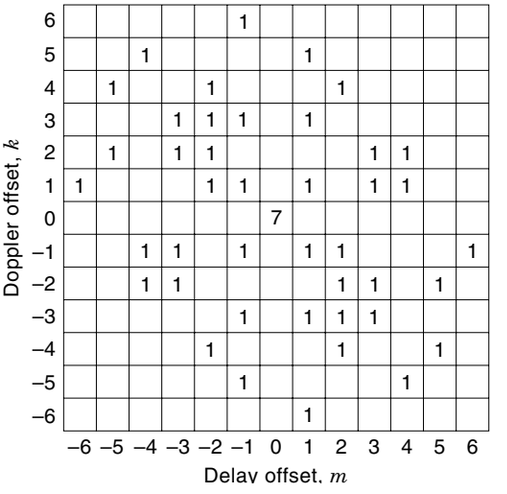


Figure 11. Ambiguity matrix of the 7-chip stepped frequency coded Costas waveform having frequency index sequence.

In looking at the ambiguity matrix of the Costas waveform in Fig. 11, it is apparent that from the point of view of both mainlobe delay-Doppler resolution and sidelobe delay-Doppler ambiguity, the Costas waveform is nearly ideal. All of the main sidelobes have a height of 1, while the mainlobe has a height of 7. In fact, by definition, an N-chip Costas waveform is a frequency coded waveform with a frequency index sequence that is a permutation of the numbers 0, 1, 2, . . ., $N - 1$ such that the mainlobe entry of the ambiguity matrix is N, while the maximum sidelobe entry is 1 (20). Sequences $(d_0, \ldots, d_{N-1})$ yielding Costas waveforms can be found for arbitrary N by exhaustive search; however, this becomes a computationally intense task, because the number of N-chip Costas sequences grows much more slowly in N, than $N!$, the number of N-chip frequency coded waveforms. For large N, this approach becomes impractical. More efficient techniques for constructing Costas waveforms are discussed in Refs. 21 and 22. One very efficient technique for constructing Costas waveforms of length $N = p - 1$, where p is a prime number, is the *Welch algorithm,* which involves a simple iteration having computational complexity proportional to N.

Phase Coded Waveforms. Consider an N-chip phase coded waveform with the rectangular $p_T(t)$ [here we assume $d_0 = \cdots = d_{N-1} = 0$ in Eq. (9)]:

$$s(t) = \sum_{n=0}^{N-1} p_T(t - nT) \exp\{j\phi_n\} \tag{12}$$

The sequence of phases $(\phi_0, \ldots, \phi_{N-1})$ specifies the phase angle to be applied to each of the N chips making up the waveform $s(t)$.

These waveforms are very similar to the types of waveforms used in direct-sequence spread-spectrum communications and hence are often referred to as *direct-sequence waveforms.* Most often, the set of phases considered is a finite set, such as $\{0, \pi\}$, $\{0, \pi/2, \pi, 3\pi/2\}$, or more generally $\{0, \pi/L, 2\pi/L, \ldots, (L - 1)\pi/L\}$, where the phases ϕ_n take on values from these sets, often repeating values unlike the frequency coded waveforms we considered in the last section.

One family of phase coded waveforms that have been applied to radar problems are the pseudonoise (PN) sequences or m-sequences commonly used in spread-spectrum communications (27–29). These waveforms take on values of either $+1$ or -1 on each chip, and hence the phases are taken from the set $\{0, \pi\}$. These waveforms are useful for generating very wide bandwidth signals by taking N large and T small. These sequences have excellent correlation properties and are easily generated using linear and nonlinear feedback shift register circuits. Their correlation properties give rise to sharp thumbtack-like responses when evaluated on the zero-Doppler ($\nu = 0$) axis. As a result, high resolution and low range ambiguity measurements can be made using these waveforms. These waveforms have the appearance of wideband noise when observed with a spectral analyzer and hence are hard to detect without detailed knowledge of the phase sequence $(\phi_0, \phi_1, \phi_2, \ldots, \phi_{N-1})$ and have thus been used for *low probability of intercept* (LPI) "quiet radar" systems, where it is not desired to give away the fact that the radar is in operation. It is rumored that the US B-2 Stealth bomber employs a high-resolution radar system using PN sequences of this kind (30).

There are many specialized families of phase coded waveforms, most of which have the property that they have excellent delay (range) resolution and ambiguity properties along the $\nu = 0$ axis. Many of these waveforms also have fairly good ambiguity and resolution properties off the zero-Doppler axis as well. Examples of these waveforms include those generated by Barker codes, Frank codes, and Gold Codes (see Ref. 27 for details on these and other related families of waveforms).

One final family of phase codes worth mentioning are the *complementary codes* originally introduced by Marcel Golay (31) for use in optical spectroscopy, but later adapted to radar measurement problems as well. Complementary codes are actually families of phase coded codewords. Golay originally introduced complementary codes having two codewords of equal length, with each chip taking on a value of either $+1$ or -1. The two codewords had the property that their delay sidelobes along the zero-Doppler axis exactly negatives each other, while their main lobes are identical. As a result, if there is no Doppler offset and two measurements of the same target scenario can be made independently, the properly delayed matched filter outputs can be added, and the result is a response in which the delay sidelobes are completely canceled. This results in excellent ambiguity function sidelobe cancellation along the zero-Doppler axis (32). Golay's basic idea has been extended to nonbinary waveforms, complementary waveform sets with more than two waveforms, and nonzero-Doppler offsets (18,19,26).

CURRENT AND FUTURE DIRECTIONS

While the classical theory of radar and sonar signals is in many ways mature, there are a number of interesting efforts to extend the theory and practice of radar and sonar signal design. We briefly outline a few of these.

One area that has received significant attention is the design of sets of multiple radar waveforms for use together. The simplest examples of these waveform sets are Golay's complementary sequence waveforms (31), which we have already considered, as well as their extensions (18,19,26), which we discussed in the last section. The basic idea is to make complementary diverse measurements that allow for extraction of greater information about the target environment than can be obtained with a single waveform. Another reason for designing sets of waveforms for use together is for use in multistatic radar and sonar systems, where there may be several transmitters and receivers in different locations. By allowing each receiver to listen to the returns from all transmitters, it is possible to extract much more information about the environment than is possible with a single—or even multiple—monostatic systems. For these systems to be feasible, it is important that the waveforms in the set have low cross-correlation, as well as envelope and spectral characteristics that allow for efficient amplification and transmission in real systems. In Refs. 33 and 34, designs for a family of waveforms of this type for sonar applications are considered. Another novel approach to multiple waveform imaging is Bernfeld's chirp-Doppler radar (35,36), which uses a mathematical analogy between measurement using a chirp and transmission to-

mography to obtain "projections" of a delay-Doppler scattering profile. These projections are then used to form a reconstruction of the delay-Doppler profile using the inverse Radon transform techniques typically employed in projection tomography.

When making measurements using sets of waveforms, the question of which waveforms from the set to transmit and in what order they should be transmitted naturally arises. This gives rise to the notion of adaptive waveform radar (37). In Ref. 38, the problem of designing and adaptively selecting waveforms for transmission to effect target recognition is considered. The approach used selects waveforms from a fixed set (designed for a particular ensemble of targets to be classified) in such a way that the Kullback–Leibler information measure is maximized by each selection.

The idea of designing radar waveforms matched to specific target tasks has also been considered. In Ref. 39, the problems of wideband radar waveform design for detection and information extraction for targets with resonant scattering are considered. It is noted that waveforms for target detection versus information extraction have very different characteristics. It is shown that waveforms for target detection should have as much energy as possible in the target's largest scattering modes, under the energy and time–bandwidth constraints imposed on the system, while waveforms for information extraction (e.g., target recognition) should have their energy distributed among the target's scattering modes in such a way that the information about the target is maximized.

BIBLIOGRAPHY

1. T. P. Gill, *The Doppler Effect,* New York: Academic Press, 1965.
2. P. M. Woodward, *Probability and Information Theory, with Applications to Radar,* London: Pergamon, 1953.
3. J. W. Goodman, *Fourier Optics,* New York: McGraw-Hill, 1968.
4. L. Cohen, *Time-Frequency Analysis,* Upper Saddle River, NJ: Prentice-Hall, 1995.
5. A. W. Rihaczek, *Principles of High Resolution Radar,* New York: McGraw-Hill, 1969; Santa Monica, CA: Mark Resources, 1977.
6. N. Levanon, *Radar Principles,* New York: Wiley-Interscience, 1988.
7. C. E. Cook and M. Bernfeld, *Radar Signals,* New York: Academic Press, 1967.
8. R. E. Blahut, Theory of Remote Surveillance Algorithms, in R. E. Blahut, W. Miller, C. H. Wilcox (eds.), *Radar and Sonar,* part I, New York: Springer-Verlag, 1991.
9. C. W. Helstrom, *Elements of Signal Detection and Estimation,* Upper Saddle River, NJ: Prentice-Hall, 1995.
10. R. A. Altes, Target position estimation in radar and sonar, generalized ambiguity analysis for maximum likelihood parameter estimation, *Proc. IEEE,* **67**: 920–930, 1979.
11. L. H. Sibul and E. L. Titlebaum, Volume properties for the wideband ambiguity function, *IEEE Trans. Aerosp. Electron. Syst.,* **17**: 83–86, 1981.
12. H. Naparst, Dense target signal processing, *IEEE Trans. Inf. Theory,* **37**: 317–327, 1991.
13. L. G. Weiss, Wavelets and wideband correlation processing, *IEEE Sig. Proc. Mag.,* **11** (4): 13–32, 1994.
14. J. R. Klauder, The design of radar signals having both high range resolution and high velocity resolution, *Bell Syst. Tech. J.,* 808–819, July 1960.
15. C. H. Wilcox, *The synthesis problem for radar ambiguity functions,* MRC Tech. Summary Rep. 157, Mathematics Research Center, US Army, Univ. Wisconsin, Madison, WI, Apr. 1960.
16. S. M. Sussman, Least-squares synthesis of radar ambiguity functions, *IRE Trans. Inf. Theory,* Apr. 1962.
17. W. L. Root, Radar resolution of closly spaced targets, *IRE Trans. Mil. Electron.,* **MIL-6** (2): 197–204, 1962.
18. C. C. Tseng and C. L. Liu, Complementary sets of sequences, *IEEE Trans. Inf. Theory,* **IT-18**: 644–652, 1972.
19. R. Sivaswami, Multiphase complementary codes, *IEEE Trans. Inf. Theory,* **24**: 546–552, 1978.
20. J. P. Costas, A study of a class of detection waveforms having nearly ideal range-Doppler ambiguity properties, *Proc. IEEE,* **72**: 996–1009, 1984.
21. S. W. Golomb and H. Taylor, Constructions and properties of Costas arrays, *Proc. IEEE,* **72**: 1143–1163, 1984.
22. S. W. Golomb, Algebraic constructions for Costas arrays, *J. Combinatorial Theory Ser. A,* **37**: 13–21, 1984.
23. O. Moreno, R. A. Games, and H. Taylor, Sonar sequences from Costas arrays and the best known sonar sequences with up to 100 symbols, *IEEE Trans. Inf. Theory,* **39**: 1985–1987, 1993.
24. S. W. Golomb and O. Moreno, On Periodicity Properties of Costas Arrays and a Conjecture on Permutation Polynomials, *Proc. IEEE Int. Symp. Inf. Theory,* Trondheim, Norway, 1994, p. 361.
25. J. Minkoff, *Signals, Noise, and Active Sensors: Radar, Sonar, Laser Radar,* New York: Wiley, 1992.
26. J. C. Guey and M. R. Bell, Diversity waveform sets for delay-Doppler imaging, *IEEE Trans. Inf. Theory,* **44**: 1504–1522, 1998.
27. D. V. Sarwate and M. B. Pusley, Crosscorrelation properties of pseudorandom and related sequences, *Proc. IEEE,* **68**: 593–619, 1980.
28. S. W. Golomb, *Shift Register Sequences,* San Francisco: Holden-Day, 1967.
29. R. J. McEliece, *Finite Fields for Computer Scientists and Engineers,* Norwell, MA: Kluwer, 1987.
30. R. Vartabedian, Unmasking the Stealth Radar, *The Los Angeles Times,* Sec. D, 1–2, Sun., July 28, 1991.
31. M. J. E. Golay, Complementary series, *IRE Trans. Inf. Theory,* **6**: 400–408, 1960.
32. R. Turyn, Ambiguity functions of complementary sequences, *IEEE Trans. Inf. Theory,* **9**: 46–47, 1963.
33. G. Chandran and J. S. Jaffe, Signal set design with constrained amplitude spectrum and specified time-bandwidth product, *IEEE Trans. Commun.,* **44**: 725–732, 1996.
34. J. S. Jaffe and J. M. Richardson, Code-Division Multiple Beam Imaging, *IEEE Oceans '89, part 4: Acoustics,* Seattle: Arctic Studies, 1989, pp. 1015–1020.
35. M. Bernfeld, Chirp Doppler radar, *Proc. IEEE,* **72**: 540–541, 1984.
36. M. Bernfeld, Tomography in Radar, in F. A. Grünbaum, J. W. Helton, and P. Khargonekar (eds.), *Signal Processing, part II: Control Theory and Applications,* New York: Springer-Verlag, 1990.
37. D. T. Gjessing, *Target Adaptive Matched Illumination Radar,* London: Peregrinus, 1986.
38. S. Sowelam and A. H. Tewfik, Adaptive Waveform Selection for Target Classification, *Proc. of EUSIP-96, VIII Eur. Signal Process. Conf.,* Trieste, Italy, 1996.
39. M. R. Bell, Information theory and radar waveform design, *IEEE Trans. Inf. Theory,* **39**: 1578–1597, 1993.

MARK R. BELL
Purdue University

} { { } }

Information Theory of Spread-Spectrum Communication
Standard Article

Michael J. Medley[1]
[1]Air Force Research Laboratory, Rome, NY

DOI: 10.1002/047134608X.W4217
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (141K)

Abstract

The sections in this article are

Spread Spectrum Systems

Spreading the Spectrum

Applications

INFORMATION THEORY OF SPREAD-SPECTRUM COMMUNICATION

SPREAD SPECTRUM SYSTEMS

Since its inception in the mid-1950s, the term *spread spectrum* (SS) has been used to characterize a class of digital modulation techniques which satisfy the following criteria (1):

1. The transmitted signal is characterized by a bandwidth that is much greater than the minimum bandwidth necessary to send the information.
2. Spreading is accomplished prior to transmission using a spreading sequence, or code, that is independent of the information being sent.
3. Detection/demodulation at the receiver is performed by correlating the received signal with a synchronously generated replica of the spreading code used at the transmitter.

Despite what might seem to be an inefficient utilization of resources, that is, increasing bandwidth without gain over noise, the combined process of "spreading" and "despreading" the information-bearing signal offers potential improvement in communications capability that more than offsets the cost incurred in using additional bandwidth. Indeed, SS offers such benefits as

- Interference suppression
- Power spectral density reduction
- Selective addressing capability
- Resistance to multipath fading

Interference suppression refers to the SS system's ability to operate reliably in an environment corrupted or congested by a level of interference that would compromise the utility of conventional digital modulation techniques. In general, SS signaling is considered robust with respect to interference in the sense that the received signal-to-interference power ratio is independent of the time-frequency distribution of the interference energy (2). Accordingly, SS systems have found application in military communications in which hostile sources intentionally jam the channel as well as in civilian settings wherein other users or wireless services inadvertently hinder data transmission through the channel. Due to its effectiveness against a variety of interference sources, including narrowband, wideband, multiple-access and multipath interference, interference suppression has long been considered the primary advantage of SS communications.

The combined advantages of interference suppression and power spectral density reduction go a long way to explain the military's historical involvement in and application of SS research since World War II [although this historical marker contradicts the mid-1950s date previously espoused, the exact origins of SS communications are rather nebulous and defy precise attribution regarding date and source of origin (1)]. While interference suppression facilitates reliable operation in hostile environments, power spectral density reduction is often exploited to produce low probability of intercept (LPI) or low probability of detect (LPD) waveforms. Low power spectral density is a direct result of spreading the power-limited, nominal-bandwidth information signal over a much greater bandwidth. LPI and LPD, combined with appropriate encryption/decryption techniques, effectively establish the basis of military and civilian covert communications.

The transition of interest in SS communications from primarily defense-oriented markets to commercial products and enterprises has been due in part to two commercially recognized deficiencies: (1) a way in which to support multiple users while simultaneously using bandwidth efficiently, and (2) a means of combating multipath fading in mobile communications. As discussed in the following sections, the autocorrelation of the SS waveform closely approximates that of an impulse function. This noiselike quality of the spread signal facilitates the design and implementation of multiuser/multiple-access communications systems in which several users are assigned unique *signature* codes and are allowed to transmit simultaneously. At the receiver, each user's desired signal is extracted from the composite sum of all user signals plus noise through correlation with the appropriate signature sequence—this description delineates the basis of code-division multiple-access (CDMA) systems in use today. In light of the fact that bandwidth is a physically limited commodity, CDMA essentially allows the number of users supported by existing channels to increase independently of bandwidth at the cost of lower performance, that is, higher error rate. Accordingly, bandwidth is conserved and, thus, utilized more efficiently. Robustness to multipath is also realized as a result of the SS waveform's similarity to white noise. Due to the similarity between the autocorrelation response of the SS waveform and an impulse function, multiple time-delayed replicas of the original signal plus noise can be resolved and coherently combined at the receiver to effectively raise the signal-to-noise ratio (SNR).

Spreading Codes

Based on the previous definition of SS systems, it is apparent that some type of code, or sequence, capable of spreading the information bandwidth must be identified. Here, such codes are discussed—the actual mechanisms by which they effect bandwidth spreading are the focus of subsequent sections.

In practice, data-independent pseudorandom, or pseudonoise (PN), sequences govern the spreading and despreading processes. As their name implies, pseudonoise spreading codes have statistical properties closely approximating those of sampled white noise; in fact, to the unintended listener, such sequences appear as random binary sequences. Although spreading codes can be generally categorized into two classes, periodic and aperiodic, the most often used spreading codes in contemporary communications systems are periodic in nature. This is in part due to the limited number of aperiodic, or Barker, sequences with sufficient length for practical applications as well as the availability of simple shift register structures capable of producing pseudorandom periodic sequences (3).

In many applications, maximal length sequences, or *m-sequences*, are often used because of their ease of generation and good randomness properties. These binary-valued, *shift register* sequences are generated as the output of an n-stage maximum-length shift register (MLSR) with a feedback network consisting of modulo-2 adders. Due to its cyclic nature, an n-stage MLSR produces a periodic sequence with period

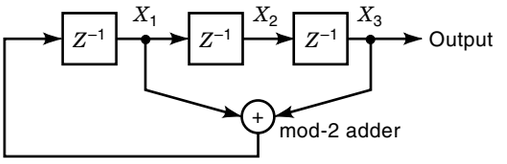


Figure 1. Maximum-length shift register with $n = 3$ and period, $L = 2^n - 1 = 7$.

$L = 2^n - 1$; L is also the length of the corresponding m-sequence. Each sample in the sequence is called a *chip,* meaning that a given m-sequence is $2^n - 1$ chips long. Specific properties of m-sequences include (3):

Balance Property. In each period of a maximal length sequence, the number of 1s is always one more than the number of 0s.

Run Property. Among the runs of 1s and 0s in each period of an m-sequence, one-half of the runs of each kind are of length one, one-fourth are of length two, one-eighth are of length three, and so on, provided these fractions represent meaningful numbers of runs.

Correlation Property. The autocorrelation function of an m-sequence is periodic and binary-valued, that is,

$$R[k] = \begin{cases} +L & k = iL \\ -1 & k \neq iL \end{cases} \tag{1}$$

where i is any integer and L is the length of the code. Note that this expression is valid for all m-sequences regardless of the value of L.

Figure 1 illustrates an $n = 3$-stage MLSR as an example. Assuming that the MLSR initial state is set to $[X_1, X_2, X_3] = [1, 0, 0]$, the resulting binary output sequence, determined by cycling through successive register states $[X_1, X_2, X_3] = [1, 0, 0]$, $[1, 1, 0]$, $[1, 1, 1]$, $[0, 1, 1]$, $[1, 0, 1]$, $[0, 1, 0]$ and $[0, 0, 1]$, is (0, 0, 1, 1, 1, 0, 1). Successive iterations return the MLSR state to its initial value, [1, 0, 0], wherein the process as well as the resulting output sequence begin to repeat. The MLSR output sequence is thus periodic in the sense that the $L = 7$-chip m-sequence, (0, 0, 1, 1, 1, 0, 1), is repeated every seven iterations as long as the MLSR is allowed to run. Clearly, the spreading sequence (0, 0, 1, 1, 1, 0, 1) contains four ones and three zeros as is consistent with the balance property. Likewise, the total presence of four runs—two of length one, one of length two, and one of length three essentially meets the specifications of the run property.

As an illustration of the correlation property, Fig. 2 depicts a 7-chip m-sequence and its corresponding autocorrelation function. In this case, the spreading sequence, (0, 0, 1, 1, 1, 0, 1), which is converted to $(-1, -1, 1, 1, 1, -1, 1)$ for transmission, produces the cyclic autocorrelation response given in Eq. (1) with $L = 7$. Further details regarding the origin and implementation of m-sequences as well as other potential spreading codes, including Barker, Gold and Kasami sequences, are found in the literature (2–7).

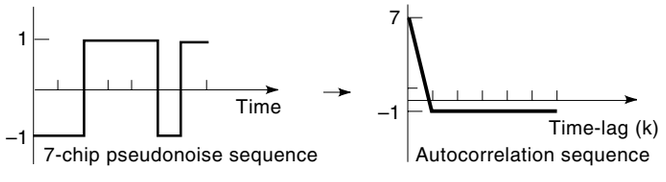


Figure 2. Length $L = 7$ m-sequence and corresponding cyclic autocorrelation response.

SPREADING THE SPECTRUM

As stated in the definition, spread spectrum is accomplished using a spreading code that is independent of the information being sent. The nature and properties of a common class of spreading waveforms has been addressed in the preceding section. Here, the physical mechanisms by which spectrum spreading is accomplished are discussed. Although there are various approaches to generating spread spectrum waveforms, such as direct-sequence (DS) modulation, frequency-hop (FH) and time-hop (TH) as well as hybrid variants incorporating aspects of each of these, each approach is fundamentally based on the underlying spreading code and endeavors to create a wideband signal from the given information data. Of these techniques, DS and FH spread spectrum are most commonly employed. Information regarding other spread spectrum formats is presented in (5,7).

Direct-Sequence Spread Spectrum

In direct-sequence spread spectrum (DS-SS), the wideband transmitted signal is obtained by multiplying the binary baseband information signal, $b(t)$, by the spreading code as shown in Fig. 3. Note that this figure inherently incorporates the use of binary phase-shift keying (BPSK) modulation and is thus representative of a DS/BPSK spread spectrum system; more generally, the combination of DS-SS with M-ary PSK modulation is referred to as DS/MPSK-SS. Although practical DS/MPSK-SS systems often modulate individual data bits using only a portion of the total m-sequence, it is assumed here, for convenience, that each bit is modulated by a single, full-length m-sequence with the number of chips in the spreading code equal to its length, L. With the bit period defined as T_b, the number of chips per bit is given by the ratio $T_b/T_c = L$ where T_c is the chip duration. The rate of the DS-SS waveform, also called the *chip rate* of the system, is $R_c = 1/T_c$ chips/s.

In practice, the bit duration, T_b, is typically much greater than T_c. Consequently, the chip rate is often orders of magnitude larger than the original bit rate $R_b = 1/T_b$, thus necessitating the increase, or spread, in transmission bandwidth. As shown in Fig. 4, the frequency response of the spread waveform has a $\text{sinc}(x) = \sin(x)/x$ shape with main lobe bandwidth equal to $2R_c$. Pulse shaping can be used to minimize the sidelobes and effectively reduce the DS-SS bandwidth if necessary.

Given the baseband information signal, $b(t)$, and the spreading code, $c(t)$, the DS-SS waveform is given by

$$m(t) = c(t)b(t) \tag{2}$$

Subsequent transmission over a noisy channel corrupted by interference produces the receiver input

$$r(t) = m(t) + i(t) + n(t) \tag{3}$$

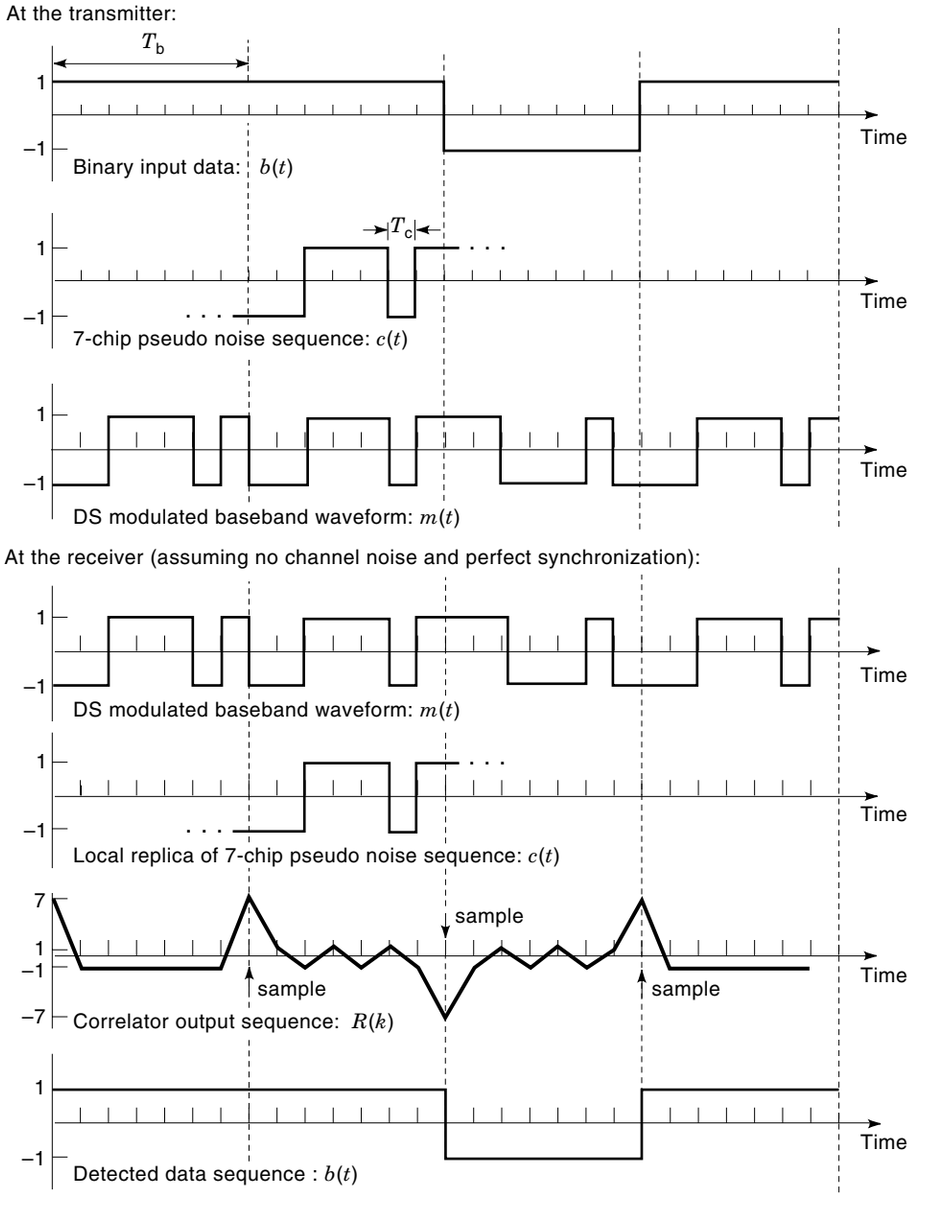


Figure 3. Direct-sequence spread spectrum modulation and demodulation.

where $i(t)$ and $n(t)$ denote interference and white noise, respectively. Often when using SS signaling, the interference power is assumed to be much greater than that of the noise and this expression is simplified as

$$r(t) = m(t) + i(t) \tag{4}$$

$$= c(t)b(t) + i(t) \tag{5}$$

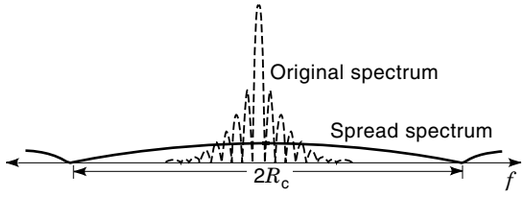


Figure 4. Magnitude-squared frequency response of a DS-SS waveform.

At the receiver, demodulation, or despreading, as depicted in Fig. 3 in the absence of noise and interference, is accomplished by multiplying $r(t)$ with a synchronized replica of the spreading code, that is,

$$u(t) = c(t)r(t) \tag{6}$$

$$= c^2(t)b(t) + c(t)i(t) \tag{7}$$

$$= b(t) + c(t)i(t) \tag{8}$$

with the final equality a result of the relationship, $c^2(t) = 1$ for all t. Subsequent integration of $u(t)$ over each symbol produces the correlator output which, when sampled at the appropriate instances, yields the detected data sequence. The preceding steps demonstrate that multiplying a signal once by the spreading code spreads its energy across a much larger bandwidth while a second multiplication reverses this process by despreading the energy and restoring the spread waveform

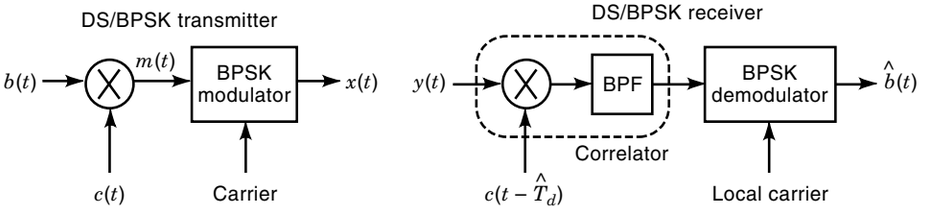


Figure 5. Synchronized DS/BPSK transmitter/receiver structures.

to its original, prespread condition. Equation (8) verifies that the information signal, $b(t)$, which is multiplied twice by the spreading code, is recovered, and returned to its initial state, while the interference, which is multiplied only once, undergoes spreading due to $c(t)$.

Whereas the previous discussion has focused on baseband signals, practical implementations typically modulate the baseband SS waveform onto a sinusoidal carrier as diagrammed in Fig. 5. Here, sinusoidal modulation produces the DS/BPSK SS signal,

$$x(t) = \sqrt{2P}m(t)\cos 2\pi f_c t \tag{9}$$

where P denotes the average power and f_c is the carrier frequency. The receiver input is thus the bandpass waveform

$$y(t) = x(t) + i(t) + n(t) \tag{10}$$

Figure 5 illustrates correlation as performed at the receiver by multiplying the received signal with a synchronized copy of the spreading code, $c(t - \hat{T}_d)$, where $\hat{T}_d$ represents the estimated propagation delay of the transmitted signal, and bandpass filtering to remove out-of-band components. Subsequent BPSK demodulation produces the estimate of the transmitted data sequence, $\hat{b}(t)$.

Synchronization between the received signal and the spreading sequence is typically performed in two stages: (1) an *acquisition* stage, which provides coarse alignment between the two waveforms, typically to within a fraction of a chip, and (2) a *tracking* stage, which maintains fine synchronization and, essentially, the best possible alignment between $y(t)$ and $c(t)$. Rudimentary discussions of synchronization techniques for SS systems are presented in (4,5), while more in-depth expositions are found in (5,8).

As demonstrated in Eq. (8), multiplication of the received signal with a locally generated, synchronized copy of the spreading code simultaneously collapses the spread data signal back to its original bandwidth while spreading any additive noise or interference to the full SS bandwidth or greater. As shown in Fig. 5, a bandpass filter with bandwidth matched to that of the original data is subsequently used to recover the data and reject a large fraction of the spread interference energy. The ratio of the signal-to-noise ratio (SNR) after despreading, $\mathrm{SNR_o}$, to the input signal-to-noise ratio, $\mathrm{SNR_i}$, is defined as the *processing gain,* G_p, that is,

$$G_p \triangleq \frac{\mathrm{SNR_o}}{\mathrm{SNR_i}} \tag{11}$$

Note that in both $\mathrm{SNR_i}$ and $\mathrm{SNR_o}$ the noise term implicitly denotes the sum of additive white Gaussian noise (AWGN) plus any additional interference. Given an input data rate of R_b bits/s, Gp can be approximated in DS-SS systems by the ratio of the chip rate to the data rate,

$$G_p \approx \frac{R_c}{R_b} = \frac{T_b}{T_c} = N \tag{12}$$

where N corresponds to the number of chips per spread data bit; $N = L$ when individual data bits are modulated by the entire spreading sequence. Note that, in practice, the entire spreading code may not be used to modulate each data bit (depending on the application, a subset of $K < L$ chips may be used). In essence, G_p roughly gauges the antijam capability and LPI/D quality of the SS system.

System performance is ultimately a function of $\mathrm{SNR_o}$, which determines the bit-error-rate (BER) experienced by the communication link. For a given data rate, spreading the transmitted signal energy over a larger bandwidth allows the receiver to operate at a lower value of $\mathrm{SNR_i}$. The range of $\mathrm{SNR_i}$ for which the receiver can provide acceptable performance is determined by the *jamming margin,* M_J, which is expressed in decibels (dB) as

$$M_J = G_p - [\mathrm{SNR_{o_{min}}} + L_{\mathrm{sys}}] \tag{13}$$

where $\mathrm{SNR_{o_{min}}}$ is the minimum $\mathrm{SNR_o}$ required to support the maximum allowable BER, and L_{sys} accounts for any losses due to receiver implementation. Hence, in addition to G_p, M_J represents another metric available to system designers indicating how much interference can be tolerated while still maintaining a prescribed level of reliability.

Frequency-Hop Spread Spectrum

In contrast to DS-SS, which directly employs the spreading sequence to modulate a phase-shift-keyed version of the information bearing waveform, frequency-hop spread spectrum (FH-SS) utilizes the spreading code to determine the carrier frequency, or frequency slot, used to transmit data over a specific period of time. In this manner, a broadband signal is generated by sequentially moving, or *hopping,* a *fixed-frequency* data-modulated carrier throughout the frequency spectrum as directed by a pseudorandom pattern known (ideally) only to the transmitter and its intended receivers. Figure 6 shows the idealized frequency spectrum of a FH-SS sig-

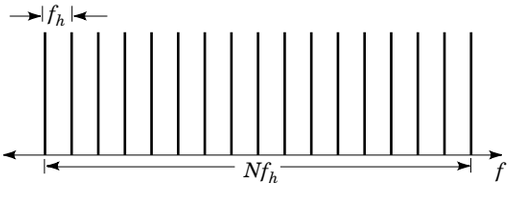


Figure 6. Idealized frequency spectrum of a FH-SS waveform.

nal in which the N hop frequencies are equally spaced at intervals of f_h Hz—the spread bandwidth of this signal is thus Nf_h Hz; in practice, each of the illustrated tones is replaced by the actual narrowband signal spectrum associated with the underlying narrowband modulation scheme employed.

The modulation format most often used in conjunction with FH-SS is M-ary FSK (MFSK); this combination is simply referred to as FH/MFSK. Figure 7 depicts typical FH/MFSK transmitter and receiver block diagrams. In the FH/MFSK transmitter, $k = \log_2 M$ information bits determine which of the M frequencies of the MFSK modulator is to be transmitted. The function of the frequency synthesizer is to produce a sinusoidal waveform, or tone, which when *mixed* with the MFSK modulator output effectively shifts its position in frequency. Note that the mixing operation as well as the required bandpass filtering is performed by the up-converter. As might be surmised, the frequency of the synthesizer output is pseudorandomly determined by the PN generator driving it. Typically, $j = \log_2 N$ chips of the spreading code are fed into the frequency synthesizer to select one of N possible tones. The FH/MFSK receiver shown in Fig. 7 simply reverses the processes performed in the transmitter by down-converting the received signal with a locally generated copy of the tone used at the transmitter and subsequently performing conventional MFSK demodulation to produce the estimated information signal, $\hat{b}(t)$.

As discussed above, at any instant in time, the actual amount of bandwidth used in FH/MFSK signaling is identical to that of conventional MFSK. This bandwidth is much less than the effective FH-SS bandwidth realized by averaging over many hops. Recognizing that the total number of possible tones is $N = 2^j$, the FH/MFSK-SS bandwidth is roughly Nf_h and, in practice, is limited primarily by the operational range of the frequency synthesizer. Frequency hopping over very large bandwidths typically precludes the use of coherent demodulation techniques due to the inability of most frequency synthesizers to maintain phase coherence over successive hops. Consequently, noncoherent demodulation is usually performed at the receiver (5).

Whereas the term *chip* in DS-SS corresponds to the samples of the spreading sequence, in FH-SS, it refers to the FH/MFSK tone with the shortest duration. The amount of time spent at each hop determines whether the FH/MFSK system is classified as *slow frequency-hopping* (SFH/MFSK) or *fast frequency-hopping* (FFH/MFSK). In SFH/MFSK systems, several MFSK symbols are transmitted per hop with the chip rate, R_c, equal to the MFSK symbol rate, R_s. The converse is true in FFH/MFSK-SS, wherein several hops are performed per MFSK symbol, with the resulting chip rate equal to the hopping rate, R_h.

In the example of SFH/MFSK-SS shown in Fig. 8, the information signal $b(t)$, whose bit rate, R_b, is related to the bit duration, T_b, via $R_b = 1/T_b$, is segmented into two-bit pairs which select the frequency (one out of four possible frequencies assuming $M = 4$-FSK modulation) to be transmitted. Since two bits are required per MFSK output, the duration of each symbol is $T_s = 2T_b$ yielding the symbol rate, $R_s = 1/T_s = \frac{1}{2}T_b$ (note that R_s is equivalent to f_h of Fig. 6). Using the periodically repeated m-sequence generated by the MLSR in Fig. 1, that is, the sequence (**0**, **0**, **1**, **1**, **1**, **0**, **1**, 0, . . .) with boldface type denoting the first period, the output of the MFSK modulator is hopped through $N = 8$ different frequency slots. To determine the hopping pattern, the PN sequence is divided into successive (not necessarily disjoint) $k = 3$ bit segments, each indicating the particular frequency slot to be used; in this case, frequency assignment is unique since $N = 2^k$. Below the resulting SFH/MFSK waveform diagram in Fig. 8 are the corresponding 3-bit segments, **001**, **110**, **100**, **111**, **010**, . . . governing the hopping pattern. Note that in this example the 000 sequence never appears and thus N is effectively only seven; such an aberration is seldom encountered in practice and, even if it were, the general principle illustrated here would still be valid. In this example, two symbols are transmitted per hop. Thus, the hop duration, $T_h = 2T_s$ with the corresponding hop rate given by $R_h = 1/T_h = R_s/2$. The effective FH-SS bandwidth is $B_{ss} = NR_s$.

Figure 9 illustrates FFH/MFSK-SS signaling. As in the SFH/MFSK example, two-bit pairs from $b(t)$ drive the MFSK modulator thus again yielding the symbol duration, $T_s = 2T_b$. In contrast to the previous example, however, multiple hops in frequency occur per MFSK symbol. Although frequency hop assignment is again governed by the periodic PN sequence segmented into the 3-bit patterns, **001**, **110**, **100**, **111**, **010**, **011**, **101**, 001, . . . (boldface denotes initial register states associated with the 7-chip m-sequence), in this example, *two* 3-bit patterns are used per symbol; the actual 3-bit pairs used per symbol are listed below the FFH/MFSK waveform diagram. Accordingly, the hop duration, $T_h = T_s/2$, with $R_h = R_b$. The overall FH-SS bandwidth, which is independent of the hop rate, is again $B_{ss} = NR_s$.

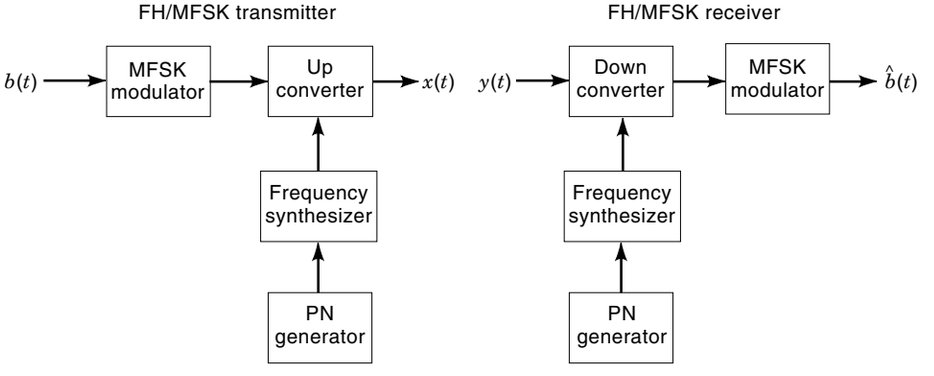


Figure 7. Synchronized FH/MFSK transmitter/receiver structures.

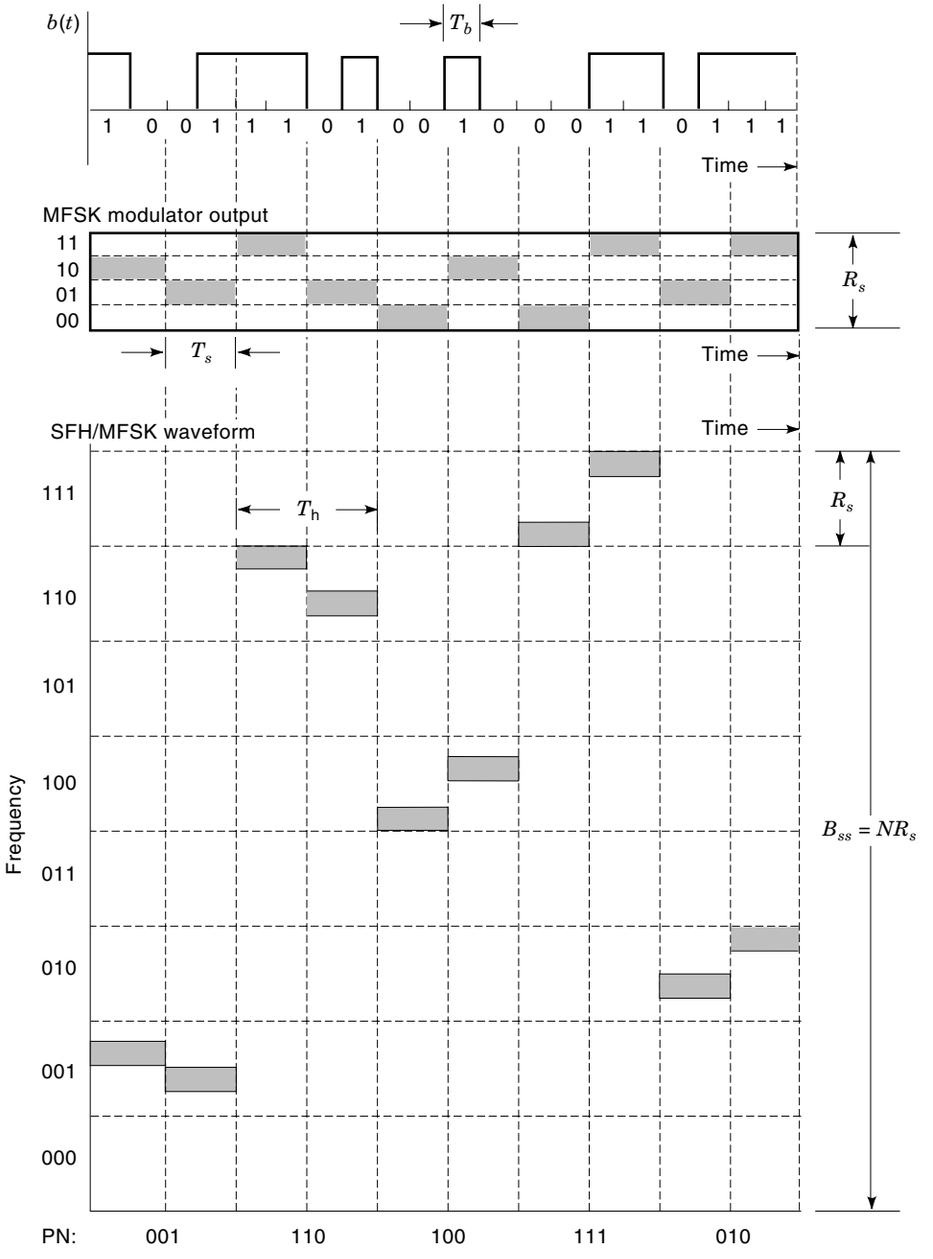


Figure 8. SFH/MFSK modulation with $M = 4$, $N = 8$, and a dwell time of $2T_b$ s.

In general, FFH/MFSK-SS is an effective means of combating certain types of jammers called *follower* and *repeat-back* jammers which attempt to intercept the frequency of the transmitted waveform and retransmit it along with additional frequencies so as to degrade receiver performance (4). When using FFH/MFSK-SS, the jammers do not typically have sufficient time to intercept and jam the spread waveform before it hops to another frequency. The price paid for such evasion, however, is the need for fast-frequency synthesizers capable of changing frequency at the required hopping rates.

As in DS-SS, the processing gain, G_p, serves as a metric indicating the signaling scheme's robustness with respect to interference. For either fast-FH or slow-FH, the effective processing gain can be approximated as the ratio of the spread spectrum bandwidth, B_{ss}, to the original data rate, R_b, that is,

$$G_p \approx \frac{B_{ss}}{R_b} \tag{14}$$

Assuming that the interference energy is spread over the entire FH bandwidth and that the original data rate is approximately equal to the symbol rate, $R_b = R_s$, the processing gain for either FH-SS system shown in Fig. 8 and Fig. 9 is approximately equal to N, the number of different frequencies over which the MFSK modulator output is hopped. The expression for the jamming margin, M_J, as given in Eq. (13), holds.

APPLICATIONS

Primary applications of spread spectrum in contemporary communications include antijam (AJ) communications, code-

division multiple-access (CDMA), and multipath interference rejection. Not surprisingly, each of these applications has been directly foreshadowed by the list of attributes associated with SS signaling presented at the beginning of this topic.

AntiJam Communications

As previously discussed, the AJ capability of a SS system is directly related to its overall processing gain, G_p. Although in theory the processing gain associated with a DS-SS waveform can be arbitrarily increased by using longer spreading codes, stringent synchronization requirements and practical bandwidth considerations limit its availability. FH-SS, on the other hand, is limited in spread bandwidth and, thus, processing gain, only by the operational limits of the frequency synthesizer. In practice, physical implementations of FH-SS are typically capable of sustaining wider bandwidth signals than practical DS-SS systems.

Even though SS systems possess a fundamental level of inherent interference immunity, different types of interference pose threats to DS-SS and FH-SS systems. In particular, pulsed-noise jammers are especially effective against DS/MPSK systems, while partial-band and multitone interference, perhaps due to CDMA *overlay* and/or narrowband services present within the SS bandwidth, represent significant threats to reliable FH/MFSK systems. Additional sources of interference, as well as their effects on SS communications, are found throughout the literature (4–7).

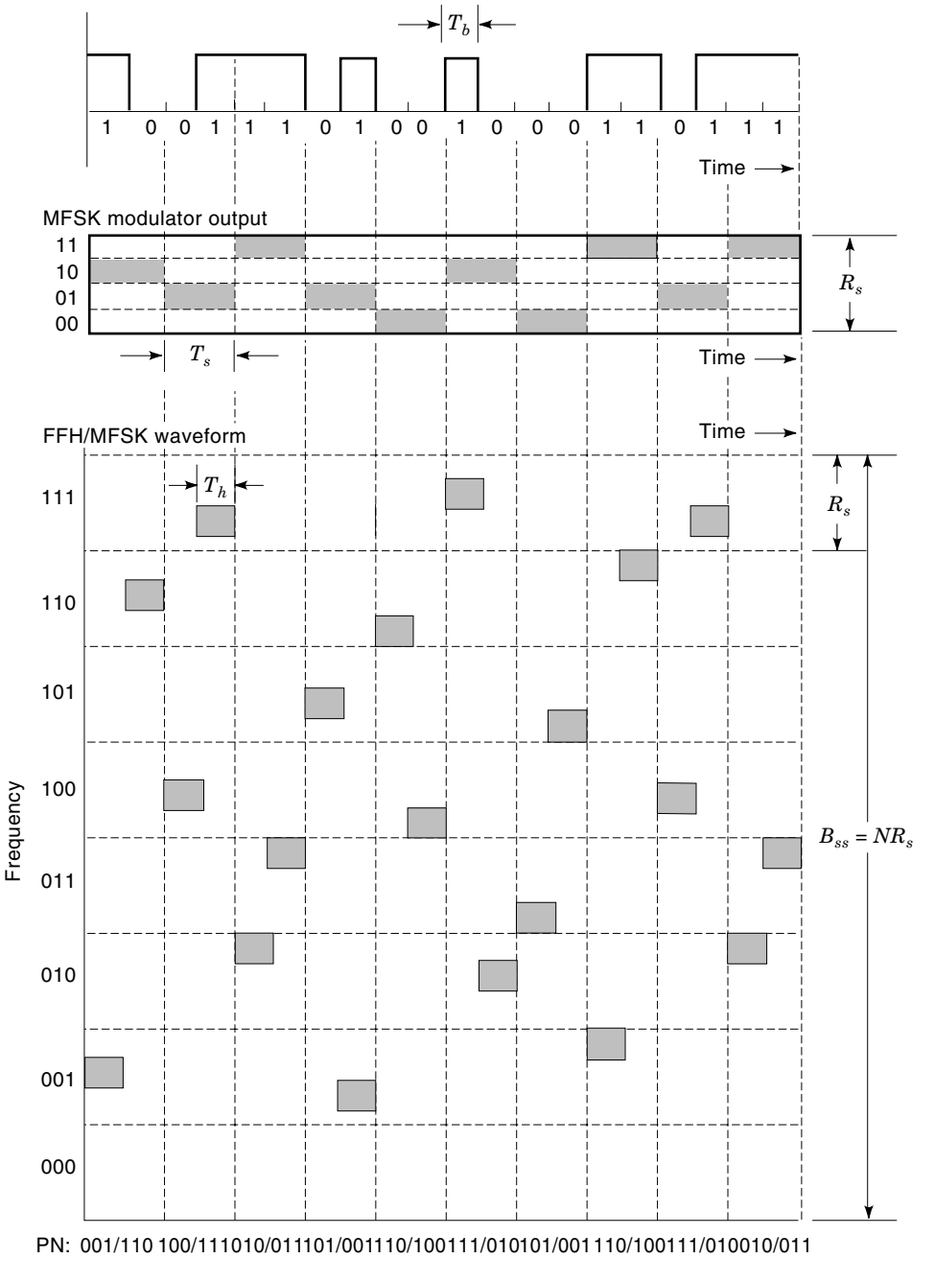


Figure 9. FFH/MFSK modulation with $M = 4$, $N = 8$, and a dwell time of T_b s.

Code-Division Multiple-Access

Prior to the introduction of code-division multiple-access (CDMA), conventional multiple-access techniques focused on dividing the available time or frequency space into disjoint partitions and assigning them to individual users. In time-division multiple-access (TDMA), users are multiplexed in time and allowed to transmit sequentially over a given channel. In contrast, in frequency-division multiple-access (FDMA), each user is assigned a portion of the channel bandwidth, separated from other users by a guard band, and allowed to use the channel simultaneously without interfering with one another. As opposed to partitioning either the time or frequency plane, CDMA provides both time and frequency diversity to its users through the use of spread spectrum modulation techniques.

In CDMA, each user is assigned a pseudorandom signature code, or sequence, similar in structure to the m-sequences discussed earlier. Gold codes and Kasami sequences, like m-sequences, have impulselike autocorrelation responses and are frequently used in such applications. Unlike m-sequences, however, these codes are generated as a set of spreading codes whose members possess minimal cross-correlation properties (4). Low cross-correlation among multiple users allows them to communicate simultaneously without significantly degrading each other's performance. In contrast to TDMA, CDMA does not require an external synchronization network and it offers graceful degradation as more users are added to the channel (due to the fact that since the spreading codes approximate wideband noise, each additional CDMA user appears as an additional noise source which incrementally raises the noise floor of the channel). In addition, CDMA also offers the benefits of SS communications including resistance to multipath as well as jamming.

Multipath Suppression

In many communications systems, actual data transmission occurs along direct, line-of-sight paths, as well as from a number of physical paths which are the result of reflections of the transmitted signal off of various scatterers such as buildings, trees, and mobile vehicles. Multipath interference is a result of the combination of these direct and indirect signal transmissions arriving at the receiver at a slight delay relative to each other. When the direct path signal is substantially stronger than the reflected components, multipath does not represent much of a challenge, if any, to reliable communications. When the direct path signal is either nonexistent or, more likely, comparable in strength to the indirect, delayed components, however, multipath interference results in variations in the received signal's amplitude, which is called *fading*.

Under slow fading conditions, multipath can be combatted directly through the use of DS-SS. Due to the noiselike property of the DS-SS waveform, multipath signal components, when correlated with the local reference code, can be resolved in time (provided the multipath spread is greater than a chip duration) and combined coherently to improve data detection. Under these conditions, the degradation in receiver performance due to multipath is directly related to the chip rate associated with DS modulation—the greater the chip rate, the less effect multipath will have on performance.

FH-SS can also be used to combat multipath interference provided that the transmitted signal hops fast enough relative to the differential time delay between the direct path and multipath signal components. In this case, much of the multipath energy falls into frequency slots vacated by the FH-SS waveform and, thus, its effect on the demodulated signal is minimized (3).

BIBLIOGRAPHY

1. R. A. Scholtz, The origins of spread-spectrum communications, *IEEE Trans Commun.,* **COM-30**: 822–854, 1982.
2. R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, Theory of spread-spectrum communications—A tutorial, *IEEE Trans. Commun.,* **COM-30**: 855–884, 1982.
3. S. Haykin, *Digital Communications,* New York: Wiley, 1988.
4. J. G. Proakis, *Digital Communications,* 3rd ed., New York: McGraw-Hill, 1995.
5. M. K. Simon et al., *Spread Spectrum Communications Handbook,* New York: McGraw-Hill, 1994.
6. B. Sklar, *Digital Communications Fundamentals and Applications,* Englewood Cliffs, NJ: Prentice-Hall, 1988.
7. R. E. Ziemer and R. L. Peterson, *Digital Communications and Spread Spectrum Systems,* New York: Macmillan, 1985.
8. R. C. Dixon, *Spread Spectrum Systems with Commercial Applications,* 3rd ed., New York: Wiley-Interscience, 1994.

Reading List

C. E. Cook and H. S. Marsh, An introduction to spread-spectrum, *IEEE Comm. Mag.,* **21** (2): 8–16, 1983.

J. K. Holmes, *Coherent Spread Spectrum Systems,* New York: Wiley, 1982.

A. J. Viterbi, Spread spectrum communications—Myths and realities, *IEEE Commun. Mag.,* **17** (3): 11–18, 1979.

MICHAEL J. MEDLEY
Air Force Research Laboratory

INFORMATION VISUALIZATION. See DATA VISUALIZATION.

INFRARED. See PHOTODETECTORS QUANTUM WELL.

} { { } }

Wiley Encyclopedia of Electrical and Electronics Engineering

Information Theory of Stochastic Processes
Standard Article
John C. Kieffer[1]
[1]University of Minnesota

DOI: 10.1002/047134608X.W4215
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (279K)

Abstract

The sections in this article are

Asymptotic Equipartition Property

Information Stability Property

Application to Source Coding Theory

Application to Channel Coding Theory

Final Remarks

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*

INFORMATION THEORY OF STOCHASTIC PROCESSES

This article starts by acquainting the reader with the basic features in the design of a data communication system and discusses, in general terms, how the information theory of stochastic processes can aid in this design process. At the start of the data communication system design process, the communication engineer is given a source, which generates information, and a noisy channel through which this information must be transmitted to the end user. The communication engineer must then design a data communication system so that the information generated by the given source can be reliably transmitted to the user via the given channel. System design consists in finding an encoder and decoder through which the source, channel, and end user can be linked as illustrated in Fig. 1.

To achieve the goal of reliable transmission, the communication engineer can use discrete-time stochastic processes to model the sequence of source outputs, the sequence of channel inputs, and the sequence of channel outputs in response to the channel inputs. The probabilistic behavior of these processes can then be studied over time. These behaviors will indicate what level of system performance can be achieved by proper encoder/decoder design. Denoting the source in Fig. 1 by (S and denoting the channel in Fig. 1 by C, one would like to know the rate $R(S)$ at which the source generates information, and one would like to know the maximum rate $R(C)$ at which the channel can reliably transmit information. If $R(S) \leq R(C)$, the design goal of reliable transmission of the source information through the given channel can be achieved.

Information theory enables one to determine the rates $R(S)$ and R (C). Information theory consists of two subareas—*source coding theory* and *channel coding theory*. Source coding theory concerns itself with the computation of R (S) for a given source model S, and channel coding theory concerns itself with the computation of $R(C)$ for a given channel model C.

Suppose that the source generates an output U_i at each discrete instant of time $i = 1, 2, 3, \ldots$. The discrete-time stochastic process $\{U_\mathrm{i}: i \geq 1\}$ formed by these outputs may obey an information-theoretic property called the *asymptotic equipartition property*, which will be discussed in the section entitled "Asymptotic Equipartition Property." The asymptotic equipartition property will be applied to source coding theory in the section entitled "Application to Source Coding Theory." If the asymptotic equipartition property is satisfied, there is a nice way to characterize the rate $R(S)$ at which the source S generates information over time.

Suppose that the channel generates a random output Y_i at time i in response to a random input X_i at time i, where $i = 1, 2, 3, \ldots$. The discrete-time stochastic process $\{(X_\mathrm{i}, Y_\mathrm{i}): i \geq 1\}$ consisting of the channel input–output pairs (called a *channel pair process*) may obey an information-theoretic property called the *information stability property*, which shall be discussed in the section entitled "Information Stability Property." The information stability property will be applied to channel coding theory in the section entitled "Application to Channel Coding Theory." If sufficiently many channel pair processes obey the information stability property, there will be a nice way to characterize the rate R (C) at which the channel C can reliably transmit information.

In conclusion, the information theory of stochastic processes consists of the development of the asymptotic equipartition property and the information stability property. In this article we discuss these properties, along with their applications to source coding theory and channel coding theory.

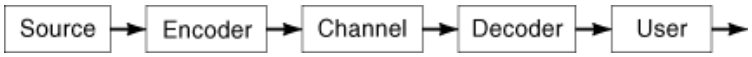


Fig. 1. Block diagram of data communication system.

Asymptotic Equipartition Property

If the asymptotic equipartition property holds for a random sequence $\{Ui: i \geq 1\}$, then, for large n, the random vector $(U_1, U_2, \ldots, U_n)$ will be approximately uniformly distributed. In order to make this idea precise, we must first discuss the concept of entropy.

Entropy. Let U be a discrete random variable. We define a nonnegative random variable $h(U)$, which is a function of U, so that

$$h(U) = -\log \Pr[U = u]$$

whenever $U = u$. The logarithm is taken to base two (as are all logarithms in this article). Also, we adopt the convention that $h(U)$ is defined to be zero, whenever $\Pr[\ U = u] = 0$. The random variable $h(U)$ is called the *self-information* of U.

The expected value of $h(U)$ is called the *entropy* of U and is denoted $H(U)$. In other words,

$$H(U) = E[h(U)] = \sum_u -\Pr[U = u] \log \Pr[U = u]$$

where E (here and elsewhere) denotes the expected value operator. Certainly, $H(U)$ satisfies

$$0 \leq H(U) \leq \infty$$

We shall only be interested in the finite entropy case in which $H(U) < \infty$. One can deduce that U has finite entropy if U takes only finitely many values. Moreover, the bound

$$H(U) \leq \log N \qquad (1)$$

holds in this case, where N is the number of values of U. To see why Eq. (1) is true, we exploit *Shannon's inequality*, which says

$$-\sum_u p(u) \log\ p(u) \leq -\sum_u p(u) \log q(u) \qquad (2)$$

whenever $\{p(u)\}$ and $\{q(u)\}$ are probability distributions on the space in which U takes its values. In Shannon's inequality, take

$$\begin{aligned} p(u) &= \Pr[U = u] \\ q(u) &= 1/N \end{aligned}$$

for each value u of U, thereby obtaining Eq. (1). If the discrete random variable U takes on a countably infinite number of values, then $H(U)$ may or may not be finite, as the following examples show.

Example 1. Let the set of values of U be 2, 3, 4, ...}, and let

$$\Pr[U = u] = \frac{C}{u(\log u)^2}$$

for every value u of U, where C is the normalization constant that makes these probabilities sum to one. It can be verified that $H(U) = \infty$.

Example 2. Let U follow a geometric distribution

$$\Pr[U = u] = p^{u-1}(1-p), \quad u = 1, 2, 3, \ldots$$

where p is a parameter satisfying $0 < p < 1$. It can be verified that

$$H(U) = \frac{-p \log p - (1-p)\log(1-p)}{1-p} < \infty$$

We are now ready to discuss the asymptotic equipartition property. Let $\{U_i: i \geq 1\}$ be a discrete-time stochastic process, in which each random variable U_i is discrete. For each positive integer n, let U^n denote the random vector $(U_1, U_2, \ldots, U_n)$. (This notational convention shall be in effect throughout this article.) We assume that the process $\{U_i: i \geq 1\}$ obeys the following two properties:

(1) $H(U^n) < \infty$, $n \geq 1$.
(2) The sequence $\{H(U^n)/n: n \geq 1\}$ has a finite limit.

Under this assumption, we can define a nonnegative real number $\bar{H}$ by

$$\bar{H} \triangleq \lim_{n\to\infty} n^{-1} H(U^n)$$

The number $\bar{H}$ is called the *entropy rate* of the process $\{U_i: i \geq 1\}$. Going further, we say that the process $\{U_i: i \geq 1\}$ obeys the *asymptotic equipartition property* (AEP) if

$$\lim_{n\to\infty} Pr[|n^{-1}h(U^n) - \bar{H}| > \epsilon] = 0, \quad \forall \epsilon > 0 \qquad (3)$$

What does the AEP tell us? Let ε be a fixed, but arbitrary, positive real number. The AEP implies that we may find, for each positive integer n, a set E_n consisting of certain n-tuples in the range of the random vector U^n, such that the sets $\{E_n\}$ obey the following properties:

$$2^{-n(\bar{H}+\epsilon)} \leq Pr[U^n = u^n] \leq 2^{-n(\bar{H}-\epsilon)}$$

(2.3) $\lim n\to\infty \Pr[U^N \in E_N] = 1$. For each n, if u^n is an n-tuple in E_n, then (2.5) For sufficiently large n, if $|E_n|$ is the number of n-tuples in E_n, then

$$2^{n(\bar{H}-\epsilon)} \leq |E_n| \leq 2^{n(\bar{H}+\epsilon)}$$

In loose terms, the AEP says that for large n, U^n can be modeled approximately as a random vector taking roughly 2^{nH} equally probable values. We will apply the AEP to source coding theory in the section entitled "Application to Source Coding Theory."

Example 3. Let $\{U_i : i \geq 1\}$ consist of independent and identically distributed (IID) discrete random variables. Letting $H(U_1) < \infty$, assumptions (2.1) and (2.2) hold, and the entropy rate is $\bar{H} = H(U_1)$. By the law of large numbers, the AEP holds.

Example 4. Let $\{U_i: i \geq 1\}$ be a stationary, ergodic homogeneous Markov chain with finite state space. Assumptions (2.1) and (2.2) hold, and the entropy rate is given by $\bar{H} = H(U^2) - H(U_1)$. Shannon (1) proved that the AEP holds in this case.

Extensions. McMillan (2) established the AEP for a stationary ergodic process $\{U_i: i \geq 1\}$ with finite alphabet. He established L^1 convergence, namely, he proved that

$$\lim_{n\to\infty} E[|n^{-1}h(U^n) - \bar{H}|] = 0$$

which is a stronger notion of convergence than the notion of convergence in Eq. (3). In the literature, McMillan's result is often referred to as the Shannon–McMillan Theorem. Breiman (3) proved almost sure convergence of the sequence $\{n^{-1}h(U^n): n \geq 1\}$ to the entropy rate $\bar{H}$:, for a stationary ergodic finite alphabet process $\{U_i : i \geq 1\}$. This is also a notion of convergence that is stronger than Eq. (3). Breiman's result is often referred to as the Shannon–McMillan–Breiman Theorem. Gray and Kieffer (4) proved that a type of nonstationary process called an asymptotically mean stationary process obeys the AEP. Verdú and Han (5) extended the AEP to a class of information sources called flat-top sources. Many other extensions of the AEP are known. Most of these results fall into one of the three categories described below.

(1) *AEP for Random Fields.* A random field $\{U_g: g \in G\}$ is given in which G is a countable group, and there is a finite set A such that each random variable U_g takes its values in A. A sequence $\{F_n: n \geq 1\}$ of growing finite subsets of G is given in which, for each n, the number of elements of F_n is denoted by $|F_n|$. For each n, let U_n denote the random vector

$$U^{F_n} \triangleq (U_g : g \in F_n)$$

One tries to determine conditions on $\{U_g\}$ and $\{F_n\}$ under which the sequence of random variables $\{|F_n|^{-1}h(U^{\mathrm{Fn}}): n \geq 1\}$ converges to a constant. Results of this type are contained in Refs. (6) (L^1 convergence) and (7) (almost sure convergence).

(2) *Entropy Stability for Stochastic Processes.* Let $\{U_i: i \geq 1\}$ be a stochastic process in which each random variable U_i is real-valued. For each $n = 1, 2, \ldots$, suppose that the distribution of the random vector U_{n} is absolutely continuous, and let F_n be its probability density function. For each n, let g_{n} be an n-dimensional probability density function different from F_n. One tries to determine conditions on $\{U_i\}$ and $\{g_{\mathrm{n}}\}$ under which the sequence of random variables

$$\left\{n^{-1}\log\frac{f_n(U^n)}{g_n(U^n)} : n \geq 1\right\}$$

converges to a constant. A process $\{U_i: i \geq 1\}$ for which such convergence holds is said to exhibit the *entropy stability property* (with respect to the sequence of densities $\{g_{\mathrm{n}}\}$). Perez (8) and Pinsker [(9), Sections 7.6, 8.4, 9.7, 10.5, 11.3] were the first to prove theorems showing that certain types of processes $\{U_i: i \geq 1\}$

exhibit the entropy stability property. Entropy stability has been studied further (10 11 12 13 14,15. In the textbook (16), Chapters 7 and 8 are chiefly devoted to entropy stability.

(3) *Entropy Stability for Random Fields.* Here, we describe a type of result that combines types (i) and (ii). As in (i), a random field $\{U_g: g \in G\}$ and subsets $\{F_n: n \geq 1\}$ are given, except that it is now assumed that each random variable U_g is real-valued. It is desired to find conditions under which the sequence of random variables

$$\left\{ |F_n|^{-1} \log \frac{f_n(U^{F_n})}{g_n(U^{F_n})} : n \geq 1 \right\}$$

converges to a constant, where, for each n, F_n is the probability density function of the $|F_n|$-dimensional random vector U^Fn and g_n is some other $|F_n|$-dimensional probability density function. Tempelman (17) gave a result of this type.

Further Reading. In this article, we have focused on the application of the AEP to communication engineering. It should be mentioned that the AEP and its extensions have been exploited in many other areas as well. Some of these areas are ergodic theory (18,19), differentiable dynamics (20), quantum systems (21), statistical thermodynamics (22), statistics (23), and investment theory (24).

Information Stability Property

The information stability property is concerned with the asymptotic information-theoretic behavior of a pair process, that is, a stochastic process $\{(X_i, Y_i): i \geq 1\}$ consisting of pairs of random variables. In order to discuss the information stability property, we must first define the concepts of mutual information and information density.

Mutual Information. Let X, Y be discrete random variables. The mutual information between X and Y, written $I(X; Y)$, is defined by

$$I(X;Y) \triangleq \sum_{x,y} \Pr[X = x, Y = y] \log \frac{\Pr[X = x, Y = y]}{\Pr[X = x] \Pr[Y = y]}$$

where we adopt the convention that all terms of the summation in which $\Pr[X = x, Y = y] = 0$ are taken to be zero. Suppose that X, Y are random variables that are not necessarily discrete. In this case, the mutual information $I(X; Y)$ is defined as

$$I(X;Y) = \sup_{(X_d, Y_d)} I(X_d; Y_d)$$

where the supremum is taken over all pairs of random variables (X_d, Y_d) in which X_d, Y_d are discrete functions of X, Y, respectively. From Shannon's inequality, Eq. (2), $I(X; Y)$ is either a nonnegative real number or is $+\infty$. We shall only be interested in mutual information when it is finite.

Example 5. Suppose X and Y are independent random variables. Then $I(X;Y) = 0$. The converse is also true.

Example 6. Suppose X is a discrete random variable. The inequality

$$I(X;Y) \leq \min[H(X), H(Y)]$$

always holds. From this inequality, we see that if $H(X)$ or $H(Y)$ is finite, then $I(X;Y)$ is finite. In particular, we see that $I(X;Y)$ is finite if either X or Y take finitely many values.

Example 7. Suppose X, Y are real-valued random variables, with variances $\sigma^2 x > 0$, $\sigma^2_y > 0$, respectively. Let (X, Y) have a bivariate Gaussian distribution, and let ρ_{xy} be the correlation coefficient, defined by

$$\rho_{xy} \triangleq \frac{E[XY] - E[X]E[Y]}{\sigma_x \sigma_y}$$

It is known (9, p. 123) that

$$I(X;Y) = -(1/2)\log(1 - \rho_{xy}^2)$$

In this case, we conclude that $I(X;Y) < \infty$ if and only if $-1 < \rho xy < 1$.

Example 8. Suppose X and Y are real-valued random variables, and that (X, Y) has an absolutely continuous distribution. Let $f(X, Y)$ be the density function of (X, Y), and let $f(X)$ and $g(Y)$ be the marginal densities of X, Y, respectively. It is known (9, p. 10) that

$$I(X;Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \log \frac{f(x,y)}{f(x)g(y)}\, dx\, dy \qquad (4)$$

Information Density. We assume in this discussion that X, Y are random variables for which $I(X;Y) < \infty$. The *information density* $i(X;Y)$ of the pair (X, Y) shall be defined to be a random variable, which is a function of (X, Y) and for which

$$I(X;Y) = E[i(X;Y)] \qquad (5)$$

In other words, the expected value of the information density is the mutual information. Let us first define the information density for the case in which X and Y are both discrete random variables. If $X = X$ and $Y = Y$, we define

$$i(X;Y) \triangleq \begin{cases} \log \dfrac{\Pr[X = x, Y = y]}{\Pr[X = x]\Pr[Y = y]}, & \Pr[X = x, Y = y] > 0 \\ 0, & \text{otherwise} \end{cases}$$

Now suppose that X, Y are not necessarily discrete random variables. The information density of the pair (X, Y) can be defined (16, Chap. 5) as the unique random variable $I(X;Y)$ such that, for any $\varepsilon > 0$, there exist discrete random variables X^ε, Y^ε, functions of X, Y, respectively, such that

$$E[|i(X', Y') - i(X;Y)|] < \epsilon$$

whenever X', Y' are discrete random variables such that

- X^ε is a function of X' and X' is a function of X.
- Y^ε is a function of Y' and Y' is a function of Y.

Example 9. In Example 8, if $I(X; Y) < \infty$, then

$$i(X;Y) = \log \frac{f(X,Y)}{f(X)g(Y)}$$

Example 10. If X is a discrete random variable with finite entropy, then

$$I(X;X) = H(X)$$

$$i(X;X) = h(X)$$

We are now ready to discuss the information stability property. Let $\{(X_i, Y_i)\colon i \geq 1\}$ be a pair process satisfying the following two properties:

(1) (10.1)$I(X^n; Y^n) < \infty$, $n \geq 1$.
(2) (10.2)The sequence $\{n^{-1} I(X^n; Y^n)\colon n \geq 1\}$ has a finite limit.

We define the *information rate* of the pair process $[(X_i, Y_i)\colon I \geq 1\ ]$ to be the nonnegative real number

$$\bar{I} \triangleq \lim_{n\to\infty} n^{-1} I(X^n;Y^n)$$

A pair process $[(X_i, Y_i)\colon I \geq 1\ ]$ satisfying (10.1) and (10.2) is said to obey the *information stability property* (ISP) if

$$\lim_{n\to\infty} \Pr[|n^{-1} i(X^n;Y^n) - \bar{I}| > \epsilon] = 0, \quad \forall \epsilon > 0$$

We give some examples of pair processes obeying the ISP.

Example 11. Let the stochastic process $[X_i : I \geq 1\ ]$ and the stochastic process $[Y_i : I \geq 1\ ]$ be statistically independent. For every positive integer n, we have $I(X^n; Y^n) = 0$. It follows that the pair process $[(X_i, Y_i)\colon I \geq 1\ ]$ obeys the ISP and that the information rate is zero.

Example 12. Let us be given a semicontinuous stationary ergodic channel through which we must transmit information. "Semicontinuous channel" refers to the fact that the channel generates an infinite sequence of random outputs $[Y_i\}$ from a continuous alphabet in response to an infinite sequence of random inputs $\{X_i\}$ from a discrete alphabet. "Stationary ergodic channel" refers to the fact that the channel pair process $\{(X_i, Y_i)\}$ will be stationary and ergodic whenever the sequence of channel inputs $\{X_i\}$ is stationary and ergodic. Suppose that $\{X_i\}$ is a stationary ergodic discrete-alphabet process, which we apply as input to our given channel. Let $[Y_i\ ]$ be the resulting channel output process. In proving a channel coding theorem (see the section entitled "Application to Channel Coding Theory"), it could be useful to know whether the stationary and ergodic pair process $\{(X_i, Y_i)\colon I \geq 1\}$ obeys the information stability property. We quote a result that allows us to conclude that the ISP holds in this type of situation. Appealing to Theorems 7.4.2 and 8.2.1 of (9), it is known that a stationary and ergodic pair process $[(X_i, Y_i)\colon I \geq 1\ ]$ will obey the ISP provided that X_1 is discrete with $H(X_1) < \infty$. The proof of this fact in (9) is too complicated to discuss here. Instead, let us deal with the special case in which we assume that Y_1 is also discrete with $H(Y_1) < \infty$. We easily deduce that $[(X_i,Y_i)\colon I \geq 1\ ]$ obeys the ISP. For we can write

$$n^{-1} i(X^n;Y^n) = n^{-1} h(X^n) + n^{-1} h(Y^n) - n^{-1} h(X^n, Y^n) \quad (6)$$

for each positive integer n. Due to the fact that each of the processes $\{X_i\}$, $\{Y_i\}$, $\{(X_i,Y_i)\}$ obeys the AEP, we conclude that each of the three terms on the right hand side of Eq. (6) converges to a constant as $n \to \infty$. The left side of Eq. (6) therefore must also converge to a constant as $n \to \infty$.

Example 13. An IID pair process $[(X_i,Y_i)\colon I \geq 1\,]$ obeys the ISP provided that $I(X_1; Y_1) < \infty$. In this case, the information rate is given by $\tilde{I} = I\,(X_1; Y_1)$. This result is evident from an application of the law of large numbers to the equation

$$n^{-1} i(X^n; Y^n) = n^{-1} \sum_{i=1}^{n} i(X_i; Y_i)$$

This result is important because this is the type of channel pair process that results when an IID process is applied as input to a memoryless channel. (The memoryless channel model is the simplest type of channel model—it is discussed in Example 21.)

Example 14. Let $[(X_i,Y_i)\colon I \geq 1\,]$ be a Gaussian process satisfying (10.1) and (10.2). Suppose that the information rate of this pair process satisfies $\bar{I} > 0$. It is known that the pair process obeys the ISP (9, Theorem 9.6.1).

Example 15. We assume that $[(X_i,Y_i)\colon I \geq 1\,]$ is a stationary Gaussian process in which, for each I, the random variables X_i and Y_i are real-valued and have expected value equal to zero. For each integer $k \geq 0$, define the trix

$$R(k) = \begin{bmatrix} R_{1,1}(k) & R_{1,2}(k) \\ R_{2,1}(k) & R_{2,2}(k) \end{bmatrix} = \begin{bmatrix} E[X_1 X_{k+1}] & E[X_1 Y_{k+1}] \\ E[Y_1 X_{k+1}] & E[Y_1 Y_{k+1}] \end{bmatrix}$$

Assume that

$$\sum_{k=0}^{\infty} |R_{i,j}(k)| < \infty, \quad i,j = 1,2$$

Following (25, p. 85), we define the spectral densities

$$\begin{bmatrix} S_{1,1}(\omega) & S_{1,2}(\omega) \\ S_{2,1}(\omega) & S_{2,2}(\omega) \end{bmatrix} = \sum_{k=-\infty}^{\infty} R(k) \exp(-j\omega k), \quad -\pi \leq \omega \leq \pi \tag{7}$$

where in Eq. (7), for $k < 0$, we take $R(k) = R(-k)^{\mathrm{T}}$. Suppose that

$$\int_{-\pi}^{\pi} \log\left(1 - \frac{|S_{1,2}(\omega)|^2}{S_{1,1}(\omega) S_{2,2}(\omega)}\right) d\omega < \infty$$

where the ratio $|S_{1,2}(\omega)|^2 / S_{1,1}(\omega) S_{2,2}(\omega)$ is taken to be zero whenever $S_{1,2}(\omega) = 0$. It is known (9, Theorem 10.2.1) that the pair process $[(X_i,Y_i)\colon I \geq 1\,]$ satisfies (10.1) and (10.2), and that the information rate $\tilde{I}$ is expressible as

$$\tilde{I} = -\frac{1}{4\pi} \int_{-\pi}^{\pi} \log\left(1 - \frac{|S_{1,2}(\omega)|^2}{S_{1,1}(\omega) S_{2,2}(\omega)}\right) d\omega \tag{8}$$

Furthermore, we can deduce that $[(X_i, Y_i)\colon I \geq 1\,]$ obeys the ISP. For, if $\tilde{I} > 0$, we can appeal to Example 14. On the other hand, if $\tilde{I} = 0$, Eq. (8) tells us that the processes $\{X_i\}$ and $\{Y_i\}$ are statistically independent, upon which we can appeal to Example 11.

Example 16. Let $\{(X_i,Y_i)\}\colon I \geq 1\,]$ be a stationary ergodic process such that, for each positive integer n,

$$\Pr[Y_1 \in A_1, Y_2 \in A_2, \ldots, Y_n \in A_n | X_1, X_2, \ldots, X_n] = \prod_{i=1}^{n} \Pr[Y_i \in A_i | X_i] \tag{9}$$

holds almost surely for every choice of measurable events $A_1, A_2, \ldots, A_n$. [The reader not familiar with the types of conditional probability functions on the two sides of Eq. (9) can consult (26, Chap. 6).] In the context of communication engineering, the stochastic process $[\,Y_i : i \geq 1\,]$ may be interpreted to be the process that is obtained by passing the process $[\,X_i : i \geq 1\,]$ through a memoryless channel (see Example 21). Suppose that $I(X_1; Y_1) < \infty$. Then, properties (10.1) and (10.2) hold and the information stability property holds for the pair process $[(X_i,Y_i)\colon i \geq 1\,]$ (14, 27).

Example 17. Let $[(X_i,Y_i)\colon i \geq 1\,]$ be a stationary ergodic process in which each random variable X_i is real-valued and each random variable Y_i is real-valued. We suppose that (10.1) and (10.2) hold and we let $\tilde{I}$ denote the information rate of the process $[(X_i,Y_i)\colon i \geq 1\,]$. A *quantizer* is a mapping Q from the real line into a finite subset of the real line, such that for each value q of Q, the set $[\,r\colon Q(r) = q\,]$ is a subinterval of the real line. Suppose that Q is any quantizer. By Example 12, the pair process $[(Q(X_i),Q(Y_i))\colon i \geq 1\,]$ obeys the ISP; we will denote the information rate of this process by $\tilde{I}_Q$. It is known that $[(X_i,Y_i)\colon I \geq 1\,]$ satisfies the information stability property if

$$\bar{I} = \sup_Q \bar{I}_Q \tag{10}$$

where the supremum is taken over all quantizers Q. This result was first proved in (9, Theorem 8.2.1). Another proof of the result may be found in (28), where the result is used to prove a source coding theorem. Theorem 7.4.2 of (9) gives numerous conditions under which Eq. (10) will hold.

Example 18. This example points out a way in which the AEP and the ISP are related. Let $[\,X_i : I \geq 1\,]$ be any process satisfying (2.1) and (2.2). Then the pair process $\{(X_i,X_i)\colon i \geq 1\}$ satisfies (10.1) and (10.2). The entropy rate of the process $[\,X_i : i \geq 1\,]$ coincides with the information rate of the process $(X_i,X_i)\colon i \geq 1\,]$. The AEP holds for the process $[\,X_i : i \geq 1\,]$ if and only if the ISP holds for the pair process $[(X_i,X_i)\colon i \geq 1\,]$. To see that these statements are true, the reader is referred to Example 10.

Further Reading. The exhaustive text by Pinsker (9) contains many more results on information stability than were discussed in this article. The text by Gray (16) makes the information stability results for stationary pair processes in (9) more accessible and also extends these results to the bigger class of asymptotically mean stationary pair processes. The text (9) still remains unparalleled for its coverage of the information stability of Gaussian pair processes. The paper by Barron (14) contains some interesting results on information stability, presented in a self-contained manner.

Application to Source Coding Theory

As explained at the start of this article, source coding theory is one of two principal subareas of information theory (channel coding theory being the other). In this section, explanations are given of the operational significance of the AEP and the ISP to source coding theory.

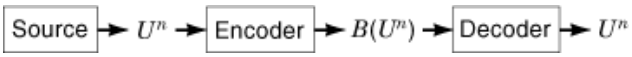


Fig. 2. Lossless source coding system.

An information source generates data samples sequentially in time. A fixed *abstract information source* is considered, in which the sequence of data samples generated by the source over time is modeled abstractly as a stochastic process $[\, U_i : i \geq 1\,]$. Two coding problems regarding the given abstract information source shall be considered. In the problem of *lossless source coding*, one wishes to assign a binary codeword to each block of source data, so that the source block can be perfectly reconstructed from its codeword. In the problem of *lossy source coding*, one wishes to assign a binary codeword to each block of source data, so that the source block can be approximately reconstructed from its codeword.

Lossless Source Coding. The problem of lossless source coding for the given abstract information source is considered first. In lossless source coding, it is assumed that there is a finite set A (called the *source alphabet*) such that each random data sample U_i generated by the given abstract information source takes its values in A. The diagram in Fig. 2 depicts a *lossless source coding system* for the block $U^n = (U_1, U_2, \ldots, U_n)$, consisting of the first n data samples generated by the given abstract information source.

As depicted in Fig. 2, the lossless source coding system consists of *encoder* and *decoder*. The encoder accepts as input the random source block U^n and generates as output a random binary codeword $B(U^n)$. The decoder perfectly reconstructs the source block U^n from the codeword $B(U^n)$. A nonnegative real number R is called an *admissible lossless compression rate* for the given information source if, for each $\delta > 0$, a Fig. 2 te system can be designed for fficiently large n so that

$$\lim_{n\to\infty} \Pr[n^{-1}|B(U^n)| \leq R + \delta] = 1 \qquad (11)$$

where $|\, B(U^n)|$ denotes the length of the codeword $B(U^n)$.

Let us now refer back to the start of this article, where we talked about the rate $R(S)$ at which the information source S in a data communication system generates information over time (assuming that the information must be losslessly transmitted). We were not precise in the beginning concerning how $R(S)$ should be defined. We now define $R(S)$ to be the minimum of all admissible lossless compression rates for the given information source S.

As discussed earlier, if the communication engineer must incorporate a given information source S into the design of a data communication system, it would be advantageous for the engineer to be able to determine the rate $R(S)$. Let us assume that the process $\{U_i : i \geq 1\}$ modeling our source S obeys the AEP. In this case, it can be shown that

$$R(S) = \bar{H} \qquad (12)$$

where $\bar{H}$ is the entropy rate of the process $\{U_i\}$. We give here a simple argument that $\bar{H}$ is an admissible lossless compression rate for the given source, using the AEP. [This will prove that $R(S) \leq \bar{H}$. Using the AEP, a proof can also be given that $R(S) \geq \bar{H}$, thereby completing the demonstration of Eq. (12), but we omit this proof.] Let A^n be the set of all n-tuples from the source alphabet A. For each $n \geq 1$, we may pick a subset E_n of A^n so that properties (2.3) to (2.5) hold. [The ε in (2.4) and (2.5) is a fixed, but arbitrary, positive real number.] Let F_n be the set of all n-tuples in A^n, which are not contained in E_n. Because of property (2.5), for sufficiently large n, we may assign each n-tuple in E_n a unique binary codeword of length $1 + \lceil n(\bar{H} + \varepsilon)\rceil$, so that each codeword begins with 0. Letting $|A|$ denote the number of symbols in A, we may assign each n-tuple in F_n a unique binary codeword of length $1 + \lceil \mathrm{CR}n \log |A|\rceil$, so that each codeword begins with 1. In this way, we have a lossless

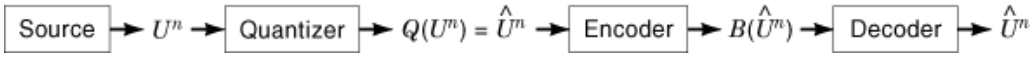


Fig. 3. Lossy source coding system.

codeword assignment for all of A^n, which gives us an encoder and decoder for a Fig. 2 lossless source coding system. Because of property (2.3), Eq. (11) holds with $R = \bar{H}$ and $\delta = 2\varepsilon$. Since ε (and therefore δ) is arbitrary, we can conclude that $\bar{H}$ is an admissible lossless compression rate for our given information source.

In view of Eq. (12), we see that for an abstract information source modeled by a process $\{U_i : i \geq 1\}$ satisfying the AEP, the entropy rate $\bar{H}$ has the following operational significance:

- No $R < \bar{H}$ is an admissible lossless compression rate for the given source.
- Every $R \geq \bar{H}$ is an admissible lossless compression rate for the given source.

If the process $\{U_i : i \geq 1\}$ does not obey the AEP, then Eq. (12) can fail, even when properties (2.1) and (2.2) are true and thereby ensure the existence of the entropy rate $\bar{H}$. Here is an example illustrating this phenomenon.

Example 19. Let the process $\{U_i : i \geq 1\}$ modeling the source S have alphabet $A = \{0, 1\}$ and satisfy, for each positive integer n, the following properties:

$$\Pr[U^n = u^n] = \begin{cases} (1/2)(1 + 2^{-n}), & u^n \text{ all zeros} \\ (1/2)2^{-n}, & \text{otherwise} \end{cases}$$

Properties (2.1) and (2.2) are satisfied and the entropy rate is $\bar{H} = \frac{1}{2}$. Reference 29 shows that $R(S) = 1$.

Extensions. The determination of the minimum admissible lossless compression rate $R(S)$, when the AEP does not hold for the process [$U_i : I \geq 1$] modeling the abstract source S, is a problem that is beyond the scope of this article. This problem was solved by Parthasarathy (29) for the case in which [$U_i : I \geq 1$] is a stationary process. For the case in which [$U_i : I \geq 1$] is nonstationary, the problem has been solved by Han and Verdú (30, Theorem 3).

Lossy Source Coding. The problem of lossy coding of a given abstract information source is now considered. The stochastic process $[U_i: I \geq 1]$ is again used to model the sequence of data samples generated by the given information source, except that the source alphabet A is now allowed to be infinite. Figure 3 depicts a *lossy source coding system* for the source block $U^n = (U_1, U_2, \ldots, U_n)$.

Comparing Fig. 3 to Fig. 2, we see that what distinguishes the lossy system from the lossless system is the presence of the quantizer in the lossy system. The quantizer in Fig. 3 is a mapping Q from the set of n-tuples An into a finite subset $Q(A$n) of A_n. The quantizer Q assigns to the random source block U^n a block

$$Q(U^n) = \hat{U}^n = (\hat{U}_1, \hat{U}_2, \ldots, \hat{U}_n) \in Q(A^n) \subset A^n$$

The encoder in Fig. 3 assigns to the quantized source block $\hat{U}^n$ a binary codeword $B^{(\hat{U}^n)}$ from which the decoder can perfectly reconstruct $\hat{U}^n$. Thus the system in Fig. 3 reconstructs not the original source block U_n, but $\hat{U}^n$, a quantized version of U^n.

In order to evaluate how well lossy source coding can be done, one must specify for each positive integer n a nonnegative real-valued function ρ_n on the product space $A^n \times A^n$ (called a *distortion measure*). The quantity ρ_n (U_n,$\hat{U}^n$) measures how closely the reconstructed block $\hat{U}^n$ in Fig. 3 resembles the source block U^n. Assuming that ρ_n is a jointly continuous function of its two arguments, which vanishes whenever the arguments are equal, one goal in the design of the lossy source coding system in Fig. 3 would be:

- *Goal 1.* Ensure that ρ_n (U_n,$\hat{U}^n$) is sufficiently close to zero.

However, another goal would be:

- *Goal 2.* Ensure that the length $|B(\hat{U}^n)|$ of the codeword $B(\hat{U}^n)$ is sufficiently small.

These are conflicting goals. The more closely one wishes $\hat{U}^n$ to resemble U_n [corresponding to a sufficiently small value of ρ_n (U_n,$\hat{U}^n$)], the more finely one must quantize U^n, meaning an increase in the size of the set $Q(A^n)$, and therefore an increase in the length of the codewords used to encode the blocks in $Q(A^n)$. There must be a trade-off in the accomplishment of Goals 1 and 2. To reflect this trade-off, two figures of merit are used in lossy source coding. Accordingly, we define a pair (R, D) of nonnegative real numbers to be an *admissible rate-distortion pair* for lossy coding of the given abstract information source, if, for any $\varepsilon > 0$, the Fig. 3 system can be designed for sufficiently large n so that

$$\lim_{n\to\infty} \Pr[\rho_n(U^n, \hat{U}^n) \leq D + \epsilon] = 1 \tag{13}$$

$$\lim_{n\to\infty} \Pr[n^{-1}|B(\hat{U}^n)| \leq R + \epsilon] = 1 \tag{14}$$

We now describe how the information stability property can allow one to determine admissible rate-distortion pairs for lossy coding of the given source. For simplicity, we assume that the process $[\, U_i : I \geq 1\,]$ modeling the source outputs is stationary and ergodic. Suppose we can find another process $\{V_i : I \geq 1\,]$ such that

- The pair process $[(U_i,V_i) : I \geq 1\,]$ is stationary and ergodic.
- There is a finite set $\hat{A} \subset A$ such that each V_i takes its values in $\hat{A}$.

Appealing to Example 12, the pair process $[(U_i,V_i): I \geq 1\,]$ satisfies the information stability property. Let $\tilde{I}$ be the information rate of this process. Assume that the distortion measures $[\, \rho_n\,]$ satisfy

$$\rho_n((u_1, u_2, \ldots, u_n), (\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_n)) = n^{-1} \sum_{i=1}^{n} \rho_1(u_i, \hat{u}_i)$$

for any pair of n-tuples $(u_1, \ldots, U_n)$, $(\hat{u}_1, \ldots, \hat{u}_n)$ from A_n. (In this case, the sequence of distortion measures $[\, \rho_n\,]$ is called a *single letter fidelity criterion.*) Let $D = E[\rho_1(U_1, V_1)\,]$. Via a standard argument (omitted here) called a random coding argument [see proof of Theorem 7.2.2 of (31)], information stability can be exploited to show that the pair $(\tilde{I}, D)$ is an admissible rate-distortion pair for our given abstract information source. [It should be pointed out that the random coding argument not only exploits the information stability property but also exploits the property that

$$\lim_{n\to\infty} \Pr[\rho_n(U^n, V^n) \leq D + \epsilon] = 1, \quad \forall \epsilon > 0$$

which is a consequence of the *ergodic theorem* [(32), Chap. 3]].

Example 20. Consider an abstract information source whose outputs are modeled as an IID sequence of real-valued random variables [U_i: $I \geq 1$]. This is called the *memoryless* source model. The squared-error single letter fidelity criterion [ρ_n] is employed, in which

$$\rho_n((u_1, u_2, \ldots, u_n), (\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_n)) = n^{-1} \sum_{i=1}^{n} (u_i - \hat{u}_i)^2$$

It is assumed that $E[U^2_1] < \infty$. For each $D > 0$, let $R(D)$ be the class of all pairs of random variables (U, V) in which

- U has the same distribution as U_1.
- V is real-valued.
- $E[(U - V)^2] \leq D$.

The *rate distortion function* of the given memoryless source is defined by

$$r(D) \triangleq \min\{I(U;V): (U,V) \in \mathsf{P}(D)\}, \quad D \geq 0$$

Shannon (33) showed that any (R, D) satisfying $R \geq r(D)$ is an admissible rate-distortion pair for lossy coding of our memoryless source model. A proof of this can go in the following way. Given the pair (R, D) satisfying $R \geq r(D)$, one argues that there is a process [V_i : $I \geq 1$] for which the pair process [U_i,V_i): $I \geq 1$] is independent and identically distributed, with information rate no bigger than R and with $E[(U_1 - V_1)^2] \leq D$. A random coding argument exploiting the fact that [(U_i, V_i): $I \geq 1$] obeys the ISP (see Example 13) can then be given to conclude that (R, D) is indeed an admissible rate-distortion pair. Shannon (33) also proved the converse statement, namely, that *any* admissible rate-distortion pair (R, D) for the given memoryless source model must satisfy $R \geq r(D)$. Therefore the set of admissible rate-distortion pairs for the memoryless source model is the set

$$\{(R, D): R \geq r(D)\} \tag{15}$$

Extensions. The argument in Example 20 exploiting the ISP can be extended [(31), Theorem 7.2.2] to show that for any abstract source whose outputs are modeled by a stationary ergodic process, the set in Eq. (15) coincides with the set of all admissible rate-distortion pairs, provided that a single letter fidelity criterion is used, and provided that the rate-distortion function $r(D)$ satisfies $r(D) < \infty$ for each $D > 0$. [The rate-distortion function for this type of source must be defined a little differently than for the memoryless source in Example 20; see (31) for the details.] Source coding theory for an abstract source whose outputs are modeled by a stationary nonergodic process has also been developed. For this type of source model, it is customary to replace the condition in Eq. (13) in the definition of an admissible rate-distortion pair with the condition

$$\limsup_{n \to \infty} E[\rho_n(U^n, \hat{U}^n)] \leq D + \epsilon$$

A source coding theorem for the stationary nonergodic source model can be proved by exploiting the information stability property, provided that the definition of the ISP is weakened to include pair processes [(U_i, V_i): $I \geq 1$] for which the sequence [$n^{-1} I(U^n; V^n)$: $n \geq 1$] converges to a nonconstant random variable. However, for this source model, it is difficult to characterize the set of admissible rate-distortion pairs by use of the ISP. Instead, Gray and Davisson (34) used the ergodic decomposition theorem (35) to characterize this

set. Subsequently, source coding theorems were obtained for abstract sources whose outputs are modeled by asymptotically mean stationary processes; an account of this work can be found in Gray (16).

Further Reading. The theory of lossy source coding is called *rate-distortion theory*. Reference (31) provides excellent coverage of rate-distortion theory up to 1970. For an account of developments in rate-distortion theory since 1970, the reader can consult (36,37).

Application to Channel Coding Theory

In this section, explanations are given of the operational significance of the ISP to channel coding theory. To accomplish this goal, the notion of an abstract channel needs to be defined. The description of a completely general abstract channel model would be unnecessarily complicated for the purposes of this article. Instead, an abstract channel model is chosen that will be simple to understand, while of sufficient generality to give the reader an appreciation for the concepts that shall be discussed.

We shall deal with a semicontinuous channel model (see Example 12) in which the channel input phabet is finite and the channel output alphabet is the real line. We proceed to give a precise formulation of this channel model. We fix a finite set A, from which inputs to our abstract channel are to be drawn. For each positive integer n, let A^n denote the set of all n-tuples $X^n = (X_1, X_2, \ldots, X_n)$ in which each $X_i \in A$, and let R^n denote the set of all n-tuples $Y^n = (Y_1, Y_2, \ldots, Y_n)$ in which each $Y_i \in R$, the set of real numbers. For each $n \geq 1$, a function F_n is given that maps each 2 n-tuple $(X^n, Y^n) \in A^n \times R^n$ into a nonnegative real number $F_n(Y^n|X^n)$ so that the following rules are satisfied:

- For each $X^n \in A^n$, the mapping $Y^n \to F_n(Y^n|X^n)$ is a jointly measurable function of n variables.
- For each $X^n \in A^n$,

$$\int \int \cdots \int_{R^n} f_n(y^n|x^n)dy^n = 1$$

For each $n \geq 2$, each $(x_1, x_2, \ldots, x_n) \in A^n$, and each $(y_1, \ldots, y_n - 1) \in R^n - 1$,

$$\int_{-\infty}^{\infty} f_n(y_1, y_2, \ldots, y_n|x_1, x_2, \ldots, x_n)\, dy_n = f_{n-1}(y_1, \ldots, y_{n-1}|x_1, \ldots, x_{n-1}) \quad (16)$$

We are now able to describe how our abstract channel operates. Fix a positive integer n. Let $X^n \in A^n$ be any n-tuple of channel inputs. In response to X^n, our abstract channel will generate a random n-tuple of outputs from R^n. For each measurable subset En of R^n, let Pr[En$|x^n$] denote the conditional probability that the channel output n-tuple will lie in En, given that the channel input is X^n. This conditional probability is computable via the formula

$$\Pr[E_n|x^n] = \int \int \cdots \int_{E_n} f_n(y^n|x^n)\, dy^n$$

We now need to define the notion of a channel code for our abstract channel model. A *channel code* for our given channel is a collection of pairs $[(x(i), E(i)): i = 1, 2, \ldots, 2^k]$ in which

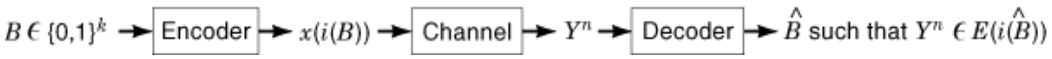


Fig. 4. Implementation of a (k, n) channel code.

(1) k is a positive integer.
(2) For some positive integer n,

- $x(1), x(2), \ldots, X(2^k)$ are n-tuples from A^n.
- $E(1), E(2), \ldots, E(2^k)$ are subsets of R^n, which form a partition of R^n.

The positive integer n given by (ii) is called the *number of channel uses* of the channel code, and the positive integer k given by (i) is called the *number of information bits* of the channel code. We shall use the notation $\boldsymbol{c}_n$ as a generic notation to denote a channel code with n channel uses. Also, a channel code shall be referred to as a (k,n) channel code if the number of channel uses is n and the number of information bits is k. In a channel code $\{(x(i), E(i))\}$, the sequences $\{x(i)\}$ are called the *channel codewords*, and the sets $\{E(i)\}$ are called the *decoding sets*.

A (k,n) channel code $\{(x(i), E(i)): i = 1, 2, \ldots, 2^k\}$ is used in the following way to transmit data over our given channel. Let $\{0, 1\}^k$ denote the set of all binary k-tuples. Suppose that the data that one wants to transmit over the channel consists of the k-tuples in $\{0, 1\}^k$. One can assign each k-tuple $B \in [0, 1]^k$ an integer index $I = I(B)$ satisfying $1 \leq I \leq 2^k$, which uniquely identifies that k-tuple. If the k-tuple B is to be transmitted over the channel, then the *channel encoder* encodes B into the channel codeword $X(I)$ in which $I = I(B)$, and then $x(i)$ is applied as input to the channel. At the receiving end of the channel, the *channel decoder* examines the resulting random channel output n-tuple Y^n that was received in response to the channel codeword $x(i)$. The decoder determines the unique random integer J such that $Y^n \in E(J)$ and decodes Y^n into the random k-tuple $\hat{B} \in \{0,1\}^k$ whose index is J. The transmission process is depicted in Fig. 4.

There are two figures of merit that tell us the performance of the (k, n) channel code $\boldsymbol{c}_n$ depicted in Fig. 4, namely, the *transmission rate* $R(\boldsymbol{c}_n)$ and the *error probability* $e(\boldsymbol{c}_n)$. The transmission rate measures how many information bits are transmitted per channel use and is defined by

$$R(\mathbf{c}_n) \stackrel{\Delta}{=} \frac{k}{n}$$

The error probability gives the worst case probability that $\hat{B}$ in Fig. 4 will not be equal to B, over all possible $B \in \{0, 1\}^k$. It is defined by

$$e(\mathbf{c}_n) \stackrel{\Delta}{=} \max_{B \in \{0,1\}^k} \{1 - \Pr[E(i(B)) | x(i(B))]\}$$

It is desirable to find channel codes that simultaneously achieve a large transmission rate and a small error probability. Unfortunately, these are conflicting goals. It is customary to see how large a transmission rate can be achieved for sequences of channel codes whose error probabilities $\to 0$. Accordingly, an *admissible transmission rate* for the given channel model is defined to be a nonnegative number R for which there exists

a sequence of channel codes [$\boldsymbol{c}_{\mathrm{n}} : n = 1, 2, \ldots$] satisfying both of the following:

$$\liminf_{n\to\infty} R(\mathbf{c}_n) \geq R$$

$$\lim_{n\to\infty} e(\mathbf{c}_n) = 0$$

We now describe how the notion of information stability can tell us about admissible transmission rates for our channel model. Let [$X_i : i \geq 1$] be a sequence of random variables taking their values in the set A, which we apply as inputs to our abstract channel. Because of the consistency crite- rion, Eq. (16), the abstract channel generates, in response to [X_i: $i \geq 1$], a sequence of real-valued random outputs [Y_i: $i \geq 1$] for which the distribution of the pair process [(X_i,Y_i): $i \geq 1$] is uniquely specified by

$$\begin{aligned}\Pr[(X_1, X_2, \ldots, X_n) = x^n,\ (Y_1, Y_2, \ldots, Y_n) \in E_n] \\ = \Pr[(X_1, X_2, \ldots, X_n) = x^n]\Pr[E_n|x^n]\end{aligned}$$

for every positive integer n, every n-tuple $X^n \in A^n$, and every measurable set En $\subset R^n$. Suppose the pair process [(X_i,Y_i): $i \geq 1$] obeys the ISP with information rate $\tilde{I}$. Then a standard argument [see (38), proof of Lemma 3.5.2] can be given to show that $\tilde{I}$ is an admissible transmission rate for the given channel model.

Using the notation introduced earlier, the *capacity*$R(C)$ of an abstract channel C is defined to be the maximum of all admissible transmission rates. For a given channel C, it is useful to determine the capacity $R(C)$. (For example, as discussed at the start of this article, if a data communication system is to be designed using a given channel, then the channel capacity must be at least as large as the rate at which the information source in the system generates information.) Suppose that an abstract channel C possesses at least one input process [$X_i : i \geq 1$] for which the corresponding channel pair process [(X_i,Y_i): $i \geq 1$] obeys the ISP. Define R_{ISP} (C) to be the supremum of all information rates of such processes [(X_i,Y_i): $I \geq 1$]. By our discussion in the preceding paragraph, we have

$$R(C) \geq R_{\mathrm{ISP}}(C)$$

For some channels C, one has $\mathscr{R}C$ ($\mathscr{R}$) = $\mathscr{R}$ $_{\mathrm{ISP}}(C)$. For such a channel, an examination of channel pair processes satisfying the ISP will allow one to determine the capacity.

Examples of channels for which this is true are the memoryless channel (see Example 21 below), the finite-memory channel (39), and the finite-state indecomposable channel (40). On the other hand, if $\mathscr{R}$ (C) > $\mathscr{R}_{\mathrm{ISP}}(C)$ for a channel C, the concept of information stability cannot be helpful in determining the channel capacity—some other concept must be used. Examples of channels for which $\mathscr{R}(C) > \mathscr{R}_{\mathrm{ISP}}(C)$ holds, and for which the capacity $\mathscr{R}(C)$ has been determined, are the $\tilde{I}$ continuous channels (41), the weakly continuous channels (42), and the historyless channels (43). The authors of these papers could not use information stability to determine capacity. They used instead the concept of "information quantiles," a concept beyond the scope of this article. The reader is referred to Refs. 41–43 to see what the information quantile concept is and how it is used.

Example 21. Suppose that the conditional density functions [$F_n : n = 1, 2, \ldots$] describing our channel satisfy

$$f_n(y^n|x^n) = \prod_{i=1}^{n} f_1(y_i|x_i)$$

for every positive integer n, every n-tuple $x^n = (x_1, \ldots, x_n)$ from A^n, and every n-tuple $Y^n = (Y_1, \ldots, Y_n)$ from R^n. The channel is then said to be *memoryless*. Let R^* be the nonnegative real number defined by

$$R^* \triangleq \sup_{(X,Y)} I(X;Y) \tag{17}$$

where the supremum is over all pairs (X, Y) in which X is a random variable taking values in A, and Y is a real-valued random variable whose conditional distribution given X is governed by the function f_1. (In other words, we may think of Y as the channel output in response to the single channel input X.) We can argue that R^* is an admissible transmission rate for the memoryless channel as follows. Pick a sequence of IID channel inputs $[\, X_i : I \geq 1\,]$ such that if $[\, Y_i : I \geq 1\,]$ is the corresponding sequence of random channel outputs, then $I(X_1; Y_1) = R^*$. The pairs $[(X_i,Y_i): i \geq 1\,]$ are IID, and the process $[(X_i,Y_i): i \geq 1\,]$ obeys the ISP with information rate $\tilde{I} = R^*$ (see Example 13). Therefore R^* is an admissible transmission rate. By a separate argument, it is well known that the converse is also true; namely, every admissible transmission rate for the memoryless channel is less than or equal to R^* (1). Thus the number R^* given by Eq. (17) is the capacity of the memoryless channel.

Final Remarks

It is appropriate to conclude this article with some remarks concerning the manner in which the separate theories of source coding and channel coding tie together in the design of data communication systems. In the section entitled "Lossless Source Coding," it was explained how the AEP can sometimes be helpful in determining the minimum rate $R(S)$ at which an information source S can be losslessly compressed. In the section entitled "Application to Channel Coding Theory," it was indicated how the ISP can sometimes be used in determining the capacity $R(C)$ of a channel C, with the capacity giving the maximum rate at which data can reliably be transmitted over the channel. If the inequality $R(S) \leq R(C)$ holds, it is clear from this article that reliable transmission of data generated by the given source S is possible over the given channel C. Indeed, the reader can see that reliable txsransmission will take place for the data communication system in Fig. 1 by taking the encoder to be a two-stage encoder, in which a good source encoder achieving a compression rate close to $R(S)$ is followed by a good channel encoder achieving a transmission rate close to $R(C)$. On the other hand, if $R(S) > R(C)$, there is no encoder that can be found in Fig. 1 via which data from the source S can reliably be transmitted over the channel C [see any basic text on information theory, such as (44), for a proof of this result]. One concludes from these statements that in designing a reliable encoder for the data communication system in Fig. 1, one need onlyonsider the two-stage encoders consisting of a good source encoder followed by a good channel encoder. This principle, which allows one to break down the problem of encoder design in communication systems into the two separate simpler problems of source encoder design and channel encoder design, has come to be called "Shannon's separation principle," after its originator, Claude Shannon.

Shannon's separation principle also extends to lossy transmission of source data over a channel in a data communication system. In Fig. 1, suppose that the data communication system is to be designed so that the data delivered to the user through the channel C must be within a certain distance D of the original data generated by the source S. The system can be designed if and only if there is a positive real number R such that (1) (R, D) is an admissible rate-distortion pair for lossy coding of the source S in the sense of the "Lossy Source Coding" section, and (2) $R \leq R(C)$. If R is a positive real number satisfying (1) and (2), Shannon's separation principle tells us that the encoder in Fig. 1 can be designed as a two-stage encoder consisting of source encoder followed by channel encoder in which:

- The source encoder is designed to achieve the compression rate R and to generate blocks of encoded data that are within distance D of the original source blocks.
- The channel encoder is designed to achieve a transmission rate close to $R(C)$.

It should be pointed out that Shannon's separation principle holds only if one is willing to consider arbitrarily complex encoders in communication systems. [In defining the quantities $R(S)$ and $R(C)$ in this article, recall that no constraints were placed on how complex the source encoder and channel encoder could be.] It would be more realistic to impose a complexity constraint specifying how complex an encoder one is willing to use in the design of a communication system. With a complexity constraint, there could be an advantage in designing a "combined source–channel encoder" which combines data compression and channel error correction capability in its operation. Such an encoder for the communication system could have the same complexity as two-stage encoders designed according to the separation principle but could afford one a better data transmission capability than the two-stage encoders. There has been much work in recent years on "combined source–channel coding," but a general theory of combined source–channel coding has not yet been put forth.

BIBLIOGRAPHY

1. C. Shannon, A mathematical theory of communication, *Bell Syst. Tech. J.*, **27**: 379–423, 623–656, 1948.
2. B. McMillan, The basic theorems of information theory, *Ann. Math. Stat.*, **24**: 196–219, 1953.
3. L. Breiman, The individual ergodic theorem of information theory, *Ann. Math. Stat.*, **28**: 809–811, 1957.
4. R. Gray J. Kieffer, Asymptotically mean stationary measures, *Ann. Probability*, **8**: 962–973, 1980.
5. S. Verdú T. Han, The role of the asymptotic equipartition property in noiseless source coding, *IEEE Trans. Inf. Theory*, **43**: 847–857, 1997.
6. J. Kieffer, A generalized Shannon-McMillan theorem for the action of an amenable group on a probability space, *Ann. Probability*, **3**: 1031–1037, 1975.
7. D. Ornstein B. Weiss, The Shannon-McMillan-Breiman theorem for a class of amenable groups, *Isr. J., Math.*, **44**: 53–60, 1983.
8. A. Perez, Notions généralisées d′incertitude, d′entropie et d′information du point de vue de la théorie de martingales, *Trans. 1st Prague Conf. Inf. Theory, Stat. Decision Funct., Random Process.*, pp. 183–208, 1957.
9. M. Pinsker, *Information and Information Stability of Random Variables and Processes*, San Francisco: Holden-Day, 1964.
10. A. Ionescu Tulcea Contributions to information theory for abstract alphabets, *Ark. Math.*, **4**: 235–247, 1960.
11. A. Perez, Extensions of Shannon-McMillan's limit theorem to more general stochastic processes, *Trans. 3rd Prague Conf. Inf. Theory*, pp. 545–574, 1964.
12. S. Moy, Generalizations of Shannon-McMillan theorem, *Pac. J. Math.*, **11**: 705–714, 1961.
13. S. Orey, On the Shannon-Perez-Moy theorem, *Contemp. Math.*, **41**: 319–327, 1985.
14. A. Barron, The strong ergodic theorem for densities: Generalized Shannon-McMillan-Breiman theorem, *Ann. Probability*, **13**: 1292–1303, 1985.
15. P. Algoet T. Cover, A sandwich proof of the Shannon- McMillan-Breiman theorem, *Ann. Probability*, **16**: 899–909, 1988.
16. R. Gray, *Entropy and Information Theory*, New York: Springer-Verlag, 1990.
17. A. Tempelman, Specific characteristics and variational principle for homogeneous random fields, *Z. Wahrschein. Verw. Geb.*, **65**: 341–365, 1984.
18. D. Ornstein, *Ergodic Theory, Randomness, and Dynamical Systems*, Yale Math. Monogr. 5, New Haven, CT: Yale University Press, 1974.
19. D. Ornstein B. Weiss, Entropy and isomorphism theorems for actions of amenable groups, *J. Anal. Math.*, **48**: 1–141, 1987.
20. R. Mañé *Ergodic Theory and Differentiable Dynamics*, Berlin and New York: Springer-Verlag, 1987.

21. M. Ohya, Entropy operators and McMillan type convergence theorems in a noncommutative dynamical system, *Lect. Notes Math.*, **1299**, 384–390, 1988.
22. J. Fritz, Generalization of McMillan's theorem to random set functions, *Stud. Sci. Math. Hung.*, **5**: 369–394, 1970.
23. A. Perez, Generalization of Chernoff's result on the asymptotic discernability of two random processes, *Colloq. Math. Soc. J. Bolyai*, No. 9, pp. 619–632, 1974.
24. P. Algoet T. Cover, Asymptotic optimality and asymptotic equipartition properties of log-optimum investme, *Ann. Probability*, **16**: 876–898, 1988.
25. A. Balakrishnan, *Introduction to Random Processes in Engineering*, New York: Wiley, 1995.
26. R. Ash, *Real Analysis and Probability*, New York: Academic Press, 1972.
27. M. Pinsker, Sources of messages, *Probl. Peredachi Inf.*, **14**, 5–20, 1963.
28. R. Gray J. Kieffer, Mutual information rate, distortion, and quantization in metric spaces, *IEEE Trans. Inf. Theory*, **26**: 412–422, 1980.
29. K. Parthasarathy, Effective entropy rate and transmission of information through channels with additive random noise, *Sankhyā, Ser. A*, **25**: 75–84, 1963.
30. T. Han S. Verdú, Approximation theory of output statistics, *IEEE Trans. Inf. Theory*, **39**: 752–772, 1993.
31. T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression*, Englewood Cliffs, NJ: Prentice–Hall, 1971.
32. W. Stout, *Almost Sure Convergence*, New York: Academic Press, 1974.
33. C. Shannon, Coding theorems for a discrete source with a fidelity criterion, *IRE Natl. Conv. Rec.*, Part 4, pp. 142–163, 1959.
34. R. Gray L. Davisson, Source coding theorems without the ergodic assumption, *IEEE Trans. Inf. Theory*, **20**: 502–516, 1974.
35. R. Gray L. Davisson, The ergodic decomposition of stationary discrete random processes, *IEEE Trans. Inf. Theory*, **20**: 625–636, 1974.
36. J. Kieffer, A survey of the theory of source coding with a fidelity criterion, *IEEE Trans. Inf. Theory*, **39**: 1473–1490, 1993.
37. T. Berger J. Gibson, Lossy source coding, *IEEE Trans. Inf. Theory*, **44**: 2693–2723, 1998.
38. R. Ash, *Information Theory*, New York: Interscience, 1965.
39. A. Feinstein, On the coding theorem and its converse for finite-memory channels, *Inf. Control*, **2**: 25–44, 1959.
40. D. Blackwell, L. Breiman, A. Thomasian, Proof of Shannon's transmission theorem for finite-state indecomposable channels, *Ann. Math. Stat.*, **29**: 1209–1220, 1958.
41. R. Gray D. Ornstein, Block coding for discrete stationary d? continuous noisy channels, *IEEE Trans. Inf. Theory*, **25**: 292-306, 1979.
42. J. Kieffer, Block coding for weakly continuous channels, *IEEE Trans. Inf. Theory*, **27**, 721–727, 1981.
43. S. Verdú T. Han, A general formula for channel capacity, *IEEE Trans. Inf. Theory*, **40**: 1147–1157, 1994.
44. T. Cover J. Thomas, *Elements of Information Theory*, New York: Wiley, 1991.

READING LIST

R. Gray L. Davisson, *Ergodic and Information Theory*, Benchmark Pap. Elect. Eng. Comput. Sci. Vol. 19, Stroudsburg, PA: Dowden, Hutchinson, & Ross, 1977.

IEEE Transactions of Information Theory, Vol. 44, No. 6, October, 1998. (Special issue commemorating fifty years of information theory.)

JOHN C. KIEFFER
University of Minnesota

} { { } }

Wiley Encyclopedia of Electrical and Electronics Engineering

Maximum Likelihood Imaging
Standard Article
Timothy J. Schulz[1]
[1]Michigan Technological University, Houghton, MI

DOI: 10.1002/047134608X.W4212
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (290K)

Abstract

The sections in this article are

MAXIMUM LIKELIHOOD IMAGING

Imaging science is a rich and vital branch of engineering in which electromagnetic or acoustic signals are measured, processed, analyzed, and interpreted in the form of multidimensional images. Because these images often contain information about the physical, biological, or operational properties of remote objects, scenes, or materials, imaging science is justly considered to be a fundamental component of that branch of engineering and science known as remote sensing. Many subjects benefit directly from advances in imaging science—these range from astronomy and the study of very large and distance objects to microscopy and the study of very small and nearby objects.

The photographic camera is probably the most widely known imaging system in use today. The familiar imagery recorded by this device usually encodes the spectral reflectance properties of an object or scene onto a two-dimensional plane. The familiarity of this form of imagery has led to a common definition of an image as "an optical reproduction of an object by a mirror or lens." There are, however, many other imaging systems in use and the object or scene properties encoded in their imagery can be very different from

those recorded by a photographic camera. Temperature variations can, for instance, be "imaged" with infrared sensing, velocity variations can be "imaged" with radar, geological formations can be "imaged" with sonar, and the physiological function of the human brain can be "imaged" with positron emission tomography (PET).

A photographic camera forms images in a manner very similar to the human eye, and, because of this, photographic images are easily interpreted by humans. The imagery recorded by an infrared camera might contain many of the features common to visible imagery; however, the phenomena being sensed are different and some practice is required before most people can faithfully interpret raw infrared imagery. For both of these modalities, though, the sensor data is often displayed as an image without the need for significant signal processing. The data acquired by an X-ray tomograph or synthetic aperture radio telescope, however, are not easily interpreted, and substantial signal processing is required to form an "image." In these situations, the processing of raw sensor data to form imagery is often referred to as image reconstruction or image synthesis (1), and the importance of signal processing in these applications is great. To confirm this importance, the 1979 Nobel prize in physiology and medicine was awarded to Alan M. Cormack and Sir Godfrey N. Hounsfield for the development and application of the signal processing methods used for X-ray computed tomography, and the 1974 Nobel prize in physics was awarded to Sir Martin Ryle for the development of aperture synthesis techniques used to form imagery with radio telescope arrays. For both of these modalities the resulting images are usually very different from the visible images formed by photographic cameras, and significant training is required for their interpretation.

Imagery formed by photographic cameras, and similar instruments such as telescopes and microscopes, can also be difficult to interpret in their raw form. Focusing errors, for example, can make imagery appear blurred and distorted, as can significant flaws in the optical instrumentation. In these situations, a type of signal processing known as image restoration (2,3) can be used to remove the distortions and restore fidelity to the imagery. Processing such as this received national attention after the discovery of the Hubble Space Telescope aberrated primary mirror in 1990, and one of the most successful and widely used algorithms for restoring resolution to Hubble imagery was based on the maximum-likelihood estimation method (4). The motivation for and derivation of this image-restoration algorithm will be discussed in great detail later in this article.

When signal processing is required for the formation or improvement of imagery, the imaging problem can usually be posed as one of statistical inference. A large number of estimation-theoretic methods are available for solving statistical-inference problems (5–9), and the method to be used for a particular application depends largely on three factors: (1) the structure imposed on the processing; (2) the quantitative criteria used to define image quality; and (3) the physical and statistical information available about the data collection process.

Structure can be imposed on processing schemes for a variety of reasons, but the most common is the need for fast and inexpensive processing. The most common structure imposed for this reason is linear processing, whereby imagery is formed or improved through linear combinations of the measured data. In some situations structural restrictions such as these are acceptable, but in many others they are not and the advent of faster and more sophisticated computing resources has served to greatly lessen the need for and use of structural constraints in imaging problems.

Many criteria can be used to quantify image quality and induce optimal signal-processing algorithms. One might ask, for example, that the processed imagery produce the "correct" image on average. This leads to an unbiased estimator, but such an estimator may not exist, may not be unique, or may result in imagery whose quality is far from adequate. By requiring that the estimated image also have, in some sense, the smallest deviations from the correct image this criterion could be modified to induce the minimum variance, unbiased estimator (MVUE), whose imagery may have desirable qualities, but whose processing structure can be difficult or impossible to derive and implement. The maximum-likelihood method for estimation leads to an alternative criterion whereby an image is selected to optimize a mathematical cost function that is induced by the physical and statistical model for the acquired data. The relative simplicity of the maximum-likelihood estimation method, along with the fact that maximum-likelihood estimates are often asymptotically unbiased with minimum variance, makes this a popular and widely studied method for statistical inference. It is largely for this reason that the development and utilization of maximum-likelihood estimation methods for imaging are the focus of this article.

One of the most important steps in the utilization of the maximum-likelihood method for imaging is the development of a practical and faithful model that represents the relationship between the object or scene being sensed and the data recorded by the sensor. This modeling step usually requires a solid understanding of the physical and statistical characteristics of electromagnetic- or acoustic-wave propagation, along with an appreciation for the statistical characteristics of the data acquired by real-world sensors. For these reasons, a strong background in the fields of Fourier optics (10,11), statistical optics (12–14), basic probability and random-process theory (15,16), and estimation theory (5–9) is essential for one wishing to apply maximum-likelihood methods to the field of imaging science.

Statistical inference problems such as those encountered in imaging applications are frequently classified as ill-posed problems (17). An image-recovery or -restoration problem is ill posed if it is not well posed, and a problem is well posed in the classical sense of Hadamard if the problem has a unique solution and the solution varies continuously with the data. Abstract formulations of image recovery and restoration problems on infinite-dimensional measurement and parameter spaces are almost always ill posed, and their ill-posed nature is usually due to the discontinuity of the solution. Problems that are formulated on finite-dimensional spaces are frequently well-posed in the classical sense—they have a unique solution and the solution is continuous in the data. These problems, however, are often ill conditioned or badly behaved and are frequently classified as ill posed even though they are technically well posed.

For problems that are ill posed or practically ill posed, the original problem's solution is often replaced by the solution to a well-posed (or well-behaved) problem. This process is referred to as regularization and the basic idea is to change the

problem in a manner such that the solution is still meaningful but no longer badly behaved (18). The consequence for imaging problems is that we do not seek to form a "perfect" image, but instead settle for a more stable—but inherently biased—image. Many methods are available for regularizing maximum-likelihood estimation problems, and these include: penalty methods, whereby the mathematical optimization problem is modified to include a term that penalizes unwanted behavior in the object parameters (19); sieve methods, whereby the allowable class of object parameters is reduced in some manner to exclude those with unwanted characteristics (20); and stopping methods, whereby the numerical algorithms used to solve a particular optimization problem are prematurely terminated before convergence and before the object estimate has obtained the unwanted features that are characteristic of the unconstrained solution obtained at convergence (21). Penalty methods can be mathematically, but not always philosophically, equivalent to the maximum a posteriori (MAP) method, whereby an a priori statistical model for the object is incorporated into the estimation procedure. The MAP method is appealing and sound provided that a physically justified model is available for the object parameters. Each of these regularization methods is effective at times, and the method used for a particular problem is often a matter of personal taste.

SCALAR FIELDS AND COHERENCE

Because most imaging problems involve the processing of electromagnetic or acoustic fields that have been measured after propagation from a remote object or scene, a good place to begin our technical discussion is with a review of scalar waves and the concept of coherence. The scalar-wave theory is widely used for two reasons: (1) acoustic wave propagation is well-modeled as a scalar phenomenon; and (2) although electromagnetic wave propagation is a vector phenomenon, the scalar theory is often appropriate, particularly when the dimensions of interest in a particular problem are large in comparison to the electromagnetic field wavelength.

A scalar field is in general described by a function in four dimensions $s(x, y, z; t)$, where x, y, and z are coordinates in three-dimensional space, and t is a coordinate in time. In many situations, the field fluctuations in time are concentrated about some center frequency f_0, so that the field can be conveniently expressed as

$$s(x, y, z; t) = a(x, y, z; t) \cos [2\pi f_0 t + \theta(x, y, z; t)] \tag{1}$$

or, in complex notation, as

$$s(x, y, z; t) = \mathrm{Re}\{u(x, y, z; t) e^{j2\pi f_0 t}\} \tag{2}$$

where

$$u(x, y, z; t) = a(x, y, z; t) e^{j\theta(x, y, z; t)} \tag{3}$$

is the complex envelope for the field. Properties of the field amplitude a, phase θ, or both are often linked to physical or operational characteristics of a remote object or scene, and the processing of remotely sensed data to determine these properties is the main goal in most imaging applications.

Coherence is an important concept in imaging that is used to describe properties of waveforms, sensors, and processing algorithms. Roughly speaking, coherence of a waveform refers to the degree to which a deterministic relationship exists between the complex envelope phase $\theta(x, y, z; t)$ at different time instances or spatial locations. Temporal coherence at time delay τ quantifies the relationship between $\theta(x, y, z; t)$ and $\theta(x, y, z; t + \tau)$, whereas the spatial coherence at spatial shift $(\Delta_x, \Delta_y, \Delta_z)$ quantifies the relationship between $\theta(x, y, z; t)$ and $\theta(x + \Delta_x, y + \Delta_y, z + \Delta_z; t)$. A coherent sensor is one that records information about the complex-envelope phase of a waveform, and a coherent signal-processing algorithm is one that processes this information. Waveforms that are coherent only over vanishingly small time delays are called temporally incoherent; waveforms that are coherent only over vanishingly small spatial shifts are called spatially incoherent. Sensors and algorithms that neither record nor process phase information are called incoherent.

Many phenomena in nature are difficult, if not impossible within our current understanding, to model in a deterministic manner, and the statistical properties of acoustic and electromagnetic fields play a fundamental role in modeling the outcome of most remote sensing and imaging experiments. For most applications an adequate description of the fields involved is captured through second-order averages known as coherence functions. The most general of these is the mutual coherence function, which is defined mathematically in terms of the complex envelope for a field as

$$\Gamma_{12}(\tau) = E[u(x_1, y_1, z_1, t + \tau) u^*(x_2, y_2, z_2, t)] \tag{4}$$

The proper interpretation for the expectation in this definition depends largely on the application, and much care must be taken in forming this interpretation. For some applications a definition involving time averages will be adequate, whereas other applications will call for a definition involving ensemble averages.

The mutual coherence function is often normalized to form the complex degree of coherence as

$$\gamma_{12}(\tau) = \frac{\Gamma_{12}(\tau)}{[\Gamma_{11}(0)\Gamma_{22}(0)]^{1/2}} \tag{5}$$

and it is tempting to define a coherent field as one for which $|\gamma_{12}(\tau)| = 1$ for all pairs of spatial locations, (x_1, y_1, z_1) and (x_2, y_2, z_2), and for all time delays, τ. Such a definition is overly restrictive and a less restrictive condition, as discussed by Mandel and Wolf (22), is that

$$\max_{\tau} |\gamma_{12}(\tau)| = 1 \tag{6}$$

for all pairs of spatial locations, (x_1, y_1, z_1) and (x_2, y_2, z_2). Although partial degrees of coherence are possible, fields that are not coherent are usually called incoherent. In some situations a field is referred to as being fully incoherent over a particular region and its mutual coherence function is modeled over this region as

$$\Gamma_{12}(\tau) \simeq \kappa I(x_1, y_1, z_1) \delta_3(x_1 - x_2, y_1 - y_2, z_1 - z_2) \delta_1(t - \tau) \tag{7}$$

where $I(\cdot)$ is the incoherent intensity for the field, $\delta_3(\cdot, \cdot, \cdot)$ is the three-dimensional Dirac impulse, $\delta_1(\cdot)$ is the one-di-

mensional Dirac impulse, and κ is a constant with appropriate units. Most visible light used by the human eye to form images is fully incoherent and fits this model. Goodman (13) and Mandel and Wolf (22) provide detailed discussions of the coherence properties of electromagnetic fields.

INCOHERENT IMAGING

Astronomical telescopes, computer assisted tomography (CAT) scanners, PET scanners, and many forms of light microscopes are all examples of incoherent imaging systems; the waveforms, sensors, and algorithms used in these situations are all incoherent. The desired image for these systems is typically related to the intensity distribution of a field that is transmitted through, reflected by, or emitted from an object or scene of interest. For many of these modalities it is common to acquire data over a variety of observing scenarios, and the mathematical model for the signal acquired by these systems is of the form

$$I_k(y) = \int h_k(y,x) I(x)\, dx, \quad k = 1, 2, \ldots, K \tag{8}$$

where $I(\cdot)$ is the object incoherent intensity function—usually related directly to the emissive, reflective, or transmissive properties of the object, $h_k(\cdot\,,\,\cdot)$ is the measurement kernel or system point-spread function for the kth observation, $I_k(\cdot)$ is the incoherent measurement signal for the kth observation, x is a spatial variable in two- or three-dimensions, and y is usually a spatial variable in one-, two-, or three-dimensions. The mathematical forms for the system point-spread functions $\{h_k(\cdot\,,\,\cdot)\}$ are induced by the physical properties of the measurement system, and much care should be taken in their determination. In telescope and microscope imaging, for example, the instrument point-spread functions model the effects of diffraction, optical aberrations, and inhomogeneities in the propagation medium; whereas for transmission or emission tomographs, geometrical optics approximations are often used and the point-spread functions model the system geometry and detector uncertainties.

For situations such as astronomical imaging with ground-based telescopes, each measurement is in the form of a two-dimensional image, whereas for tomographic systems each measurement may be in the form of a one-dimensional projection of a two-dimensional transmittance or emittance function. In either situation, the imaging task is to reconstruct the intensity function $I(\cdot)$ from noisy measurements of $I_k(\cdot)$, $k = 1, 2, \ldots, K$.

Quantum Noise in Incoherent Imagery

Light and other forms of electromagnetic radiation interact with matter in a fundamentally random manner, and, because of this, statistical models are often used to describe the detection of optical waves. Quantum electrodynamics (QED) is the most sophisticated theory available for describing this phenomenon; however, a semiclassical theory for the detection of electromagnetic radiation is often sufficient for the development of sound and practical models for imaging applications. When using the semiclassical theory, electromagnetic energy is transported according to the classical theory of wave propagation—it is only during the detection process that the field energy is quantized.

When optical fields interact with a photodetector, the absorption of a quantum of energy—a photon—results in the release of an excited electron. This interaction is referred to as a photoevent, and the number of photoevents occurring over a particular spatial region and time interval are referred to as photocounts. Most detectors of light record photocounts, and although the recorded data depend directly on the image intensity, the actual number of photocounts recorded is a fundamentally random quantity. The images shown in Fig. 1 help to illustrate this effect. Here, an image of Simeon Poisson (for whom the Poisson random variable is named) is shown as it might be acquired by a detector when 1 million, 10 million, and 100 million total photocounts are recorded.

Statistical Model

For many applications involving charge coupled devices (CCD) and other detectors of optical radiation, the semiclassical theory leads to models for which the photocounts recorded by each detector element are modeled as Poisson random variables whose means are determined by the measurement intensity $I_k(\cdot)$. That is, the expected number of photocounts acquired by the nth photodetector during the kth observation interval is

$$I_k[n] = \gamma \int_{\mathscr{Y}_n} I_k(y)\, dy \tag{9}$$

where n is a two-dimensional discrete index to the elements of the detector array, $\mathscr{Y}_n$ is the spatial region over which the nth detector element integrates the image intensity, and γ is a nonnegative scale factor that accounts for overall detector efficiency and integration time. Furthermore, the number of photocounts acquired by different detector elements are usually statistically independent, and the detector regions are often small in size relative to the fluctuations in the image intensity so that the integrating operation can be well-modeled by the sampling operation

$$I_k[n] \simeq \gamma |\mathscr{Y}_n| I_k(y_n) \tag{10}$$

where y_n is the location of the nth detector element and $|\mathscr{Y}_n|$ is its integration area.

Other Detector Effects

In addition to the quantum noise, imaging detectors introduce other nonideal effects into the imagery that they record. The efficiency with which detectors convert electromagnetic energy into photoevents can vary across elements within a detector array, and this nonuniform efficiency can be captured by attaching a gain function to the photocount mean

$$I_k[n] = a[n]\gamma |\mathscr{Y}_n| I_k(y_n) \tag{11}$$

Seriously flawed detector elements that fail to record data are also accommodated with this model by simply setting the gain to zero at the appropriate location. If different detectors are used for each observation the gain function may need to vary with each frame and, therefore, be indexed by k.

Because of internal shot noise, many detectors record photoevents even when the external light intensity is zero. The resulting photocounts are usually modeled as independent

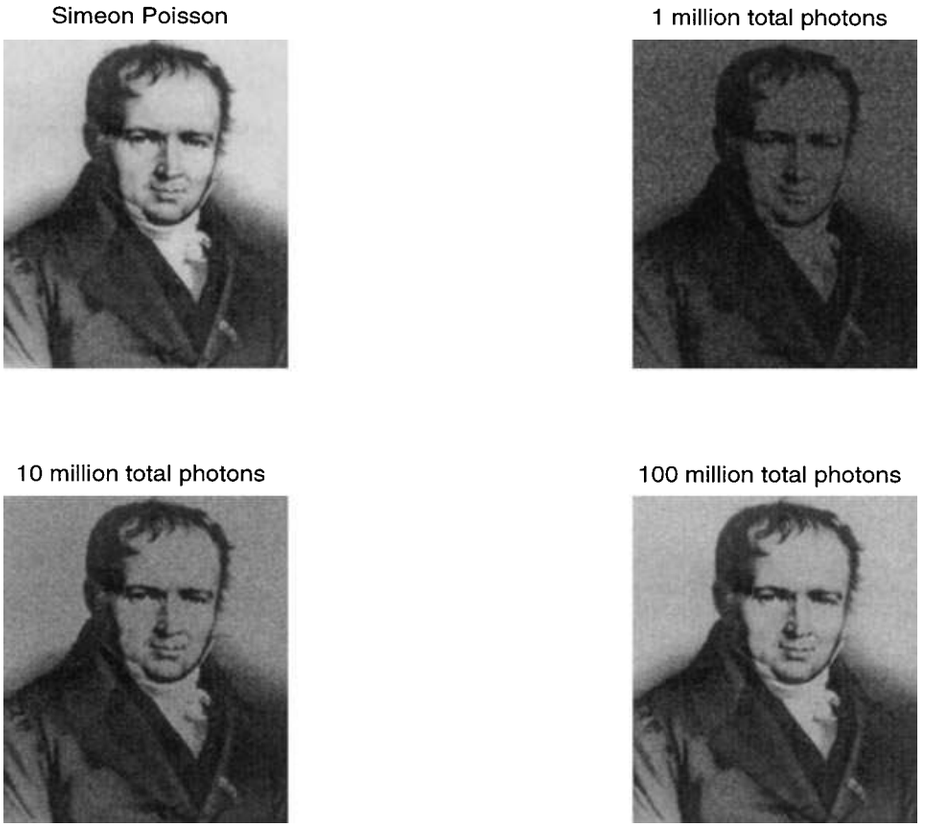


Figure 1. Image of Simeon Poisson as it might be acquired by a detector when 1 million, 10 million, and 100 million total photocounts are recorded.

Poisson random variables, and this phenomenon is accommodated by inserting a background term into the imaging equation

$$I_k[n] \simeq a[n]\gamma|\mathscr{Y}_n|I_k(y_n) + I_b[n] \tag{12}$$

As with the gain function, if different detectors are used for each observation this background term may need to vary with each frame and, therefore, be indexed by k. With the inclusion of these background counts, the number of photocounts acquired by detector element n is a Poisson random variable with mean $I_k[n]$ and is denoted by $N_k[n]$.

The data recorded by many detectors are also corrupted by another form of noise that is induced by the electronics used for the data acquisition. For CCD detectors, this is *read-out* noise and is often approximated as additive, zero-mean Gaussian random variables so that the recorded data are modeled as

$$d_k[n] = N_k[n] + g_k[n] \tag{13}$$

where $g_k[n]$ models the read-out noise at the nth detector for the kth observation. The variance of the read-out noise $\sigma^2[\cdot]$ may vary with each detector element, and the read-out noise for different detectors is usually modeled as statistically independent.

The appropriate values for the gain function $a[\cdot]$, background function $I_b[\cdot]$, and read noise variance $\sigma^2[\cdot]$ are usually selected through a controlled study of the data acquisition system. A detailed discussion of these and other camera effects for optical imaging is given in Ref. 23.

Maximum-Likelihood Image Restoration

Consistent with the noise models developed in the previous sections, the data recorded by each detector element in a photon-counting camera are a mixture of Poisson and Gaussian random variables. Accordingly, the probability of receiving N photocounts in the nth detector element is

$$\Pr\{N_k[n] = N; I\} = \exp(-I_k[n])(I_k[n])^N/N! \tag{14}$$

where

$$\begin{aligned} I_k[n] &= a[n]\gamma|\mathscr{Y}_n|I_k(y_n) + I_b[n] \\ &= a[n]\gamma|\mathscr{Y}_n|\int h_k(y_n, x)I(x)\,dx + I_b[n] \end{aligned} \tag{15}$$

contains the dependence on the unknown intensity function $I(\cdot)$. Furthermore, the probability density for the read-out noise is

$$p_{g_k[n]}(g) = (2\pi\sigma^2[n])^{-1/2}\exp[-g^2/(2\sigma[n])] \tag{16}$$

so that the density for the measured data is

$$\begin{aligned} p_{d_k[n]}(d; I) &= \sum_{N=0}^{\infty} p_{g_k[n]}(d - N)\Pr\{N_k[n] = N; I\} \\ &= \frac{(2\pi\sigma^2[n])^{-1/2}}{N}\sum_{N=0}^{\infty}\exp[-(d-N)^2/(2\sigma[n])] \\ &\quad \exp(-I_k[n])(I_k[n])^N \end{aligned} \tag{17}$$

For a given data set $\{d_k[\cdot]\}$, the maximum-likelihood estimate of $I(\cdot)$ is the intensity function that maximizes the likelihood

$$l(I) = \prod_{k=1}^{K} \prod_{n} p_{d_k[n]}(d_k[n]; I) \tag{18}$$

or, as is commonly done, its logarithm (the log-likelihood)

$$\begin{aligned} \mathscr{L}(I) &= \ln l(I) \\ &= \sum_{k=1}^{K} \sum_{n} \ln p_{d_k[n]}(d_k[n]; I) \end{aligned} \tag{19}$$

The complicated form for the measurement density $p_{d_k[n]}(\cdot\,; I)$ makes this an overly complicated optimization. When the read-out noise variance is large (greater than 50 or so), however, $\sigma^2[n]$ can be added to the measured data to form the modified data

$$\begin{aligned} \tilde{d}_k[n] &= d_k[n] + \sigma^2[n] \\ &= N_k[n] + g_k[n] + \sigma^2[n] \\ &\simeq N_k[n] + M_k[n] \end{aligned} \tag{20}$$

where $M_k[n]$ is a Poisson-distributed random variable whose mean value is $\sigma^2[n]$. The modified data at each detector element are then similar (in distribution) to the sum of two Poisson-distributed random variables $N_k[n]$ and $M_k[n]$ and, as such, are also Poisson-distributed with the mean value $I_k[n] + \sigma^2[n]$. This approximation is discussed by Snyder et al. in Refs. 23 and 24. The probability mass function for the modified data is then modeled as

$$\Pr[\tilde{d}_k[n] = D; I] = \exp\{-(I_k[n] + \sigma^2[n])\}(I_k[n] + \sigma^2[n])^D / D! \tag{21}$$

so that the log-likelihood is

$$\begin{aligned} \mathscr{L}(I) = \sum_{k=1}^{K} \sum_{n} \{ &-(I_k[n] + \sigma^2[n]) \\ &+ \tilde{d}_k[n] \ln(I_k[n] + \sigma^2[n]) - \ln d_k[n]! \} \end{aligned} \tag{22}$$

Two difficulties are encountered when attempting to find the intensity function $I(\cdot)$ that maximizes the log-likelihood $\mathscr{L}(I)$: (1) the recovery of an infinite-dimensional function $I(\cdot)$ from finite data is a terribly ill-conditioned problem; and (2) the functional form of the log-likelihood does not admit a closed form, analytic solution for the maximizer even after the dimension of the parameter function has been reduced.

To address the dimensionality problem, it is common to approximate the parameter function in terms of a finite-dimensional basis set

$$I(x) \simeq \sum_{m} I[m] \psi_m(x) \tag{23}$$

where the basis functions $\{\psi_m(\cdot)\}$ are chosen in an appropriate manner. When expressing the object function with a predetermined grid of image pixels, for example, $\psi_m(\cdot)$ might be an indicator function that denotes the location of the mth pixel. For the same situation, the basis functions might alternatively be chosen as two-dimensional impulses co-located with the center of each pixel. Many other basis sets are possible and a clever choice here can greatly affect estimator performance, but the grid of two-dimensional impulses is probably the most common. Using this basis, the data mean is expressed as

$$\begin{aligned} I_k[n] &= a[n]\gamma|\mathscr{Y}_n| I_k(y_n) + I_b[n] \\ &= a[n]\gamma|\mathscr{Y}_n| \int h_k(y_n, x) \sum_{m} I[m] \delta_2(x - x_m)\, dx + I_b[n] \\ &= a[n]\gamma|\mathscr{Y}_n| \sum_{m} h_k(y_n, x_m) I[m] + I_b[n] \end{aligned} \tag{24}$$

where y_n denotes the location of the nth measurement, x_m denotes the location of the mth object pixel, and $\delta_2(\cdot)$ is the two-dimensional Dirac impulse. The estimation problem, then, is one of estimating the discrete samples $I[\cdot]$ of the intensity function from the noisy data $\{d_k[\cdot]\}$. Because $I[\cdot]$ represents samples of an intensity function, this function is physically constrained to be nonnegative.

Ignoring terms in the log-likelihood that do not depend upon the unknown object intensity, the optimization problem required to solve for the maximum-likelihood object estimate is

$$\begin{aligned} \hat{I}[n] = \arg\max_{I \geq 0} \Bigg\{ &-\sum_{k=1}^{K} \sum_{n} (I_k[n] + \sigma^2[n]) \\ &+ \sum_{k=1}^{K} \sum_{n} \tilde{d}_k[n] \ln(I_k[n] + \sigma^2[n]) \Bigg\} \end{aligned} \tag{25}$$

where $\tilde{d}_k[n] = d_k[n] + \sigma^2[n]$ is the modified data and

$$I_k[n] = a[n]\gamma|\mathscr{Y}_n| \sum_{m} h_k(y_n, x_m) I[m] + I_b[n]$$

is the photocount mean. The solution to this problem generally requires the use of a numerical method, and a great number of techniques are available for this purpose. General-purpose techniques such as those described in popular texts on optimization theory (25,26) can be applied. In addition, specialized numerical methods devised specifically for the solution of maximum-likelihood and related problems can be applied (27,28)—a specific example is discussed in the following section.

The Expectation-Maximization Method. The expectation-maximization (EM) method is a numerical technique devised specifically for maximum-likelihood estimation problems. As described in Ref. 27, the classical formulation of the EM procedure requires one to augment the measured data—commonly referred to as the *incomplete data*—with a set of *complete data* which, if measured, would facilitate direct estimation of the unknown parameters. The application of this procedure then requires one to alternately apply an *E-step,* wherein the conditional expectation of the complete-data log-likelihood is determined, and an *M-step,* wherein all parameters are simultaneously updated by maximizing the expectation of the complete-data log-likelihood with respect to all of the unknown parameters. In general, the application of the EM procedure results in an iterative algorithm that produces a sequence of parameter estimates that monotonically increases the measured data likelihood.

The application of the EM procedure to the incoherent imaging problems has been proposed and described for numerous applications (29–32). The general application of this method is outlined as follows. First, recall that the measured (or incomplete) data $\tilde{d}_k[n]$ for each observation k and detector element n are independent Poisson variables with the expected value

$$E\{\tilde{d}_k[n]\} = a[n]\gamma|\mathscr{Y}_n|\sum_m h_k(y_n, x_m)I[m] + I_b[n] + \sigma^2[n] \quad (26)$$

Because the sum of Poisson random variables is still a Poisson random variable (and the expected value is the sum of the individual expected values), the incomplete data can be statistically modeled as

$$\tilde{d}_k[n] = \sum_m N_k^c[n, m] + M_k^c[n] \quad (27)$$

where for all frames k, detector locations n, and object pixels m, the data $N_k^c[n, m]$ are Poisson random variables, each with the expected value

$$E\{N_k^c[n, m]\} = a[n]\gamma|\mathscr{Y}_n|h_k(y_n, x_m)I[m] \quad (28)$$

and for all frames k and detector locations n, the data $M_k^c[n]$ are Poisson random variables, each with the expected value

$$E\{M_k^c[n]\} = I_b[n] + \sigma^2[n] \quad (29)$$

In the terminology of the EM method, these data $\{N_k^c[\cdot, \cdot], M_k^c[\cdot]\}$ are the complete data, and although they cannot be observed directly, their measurement, if possible, would greatly facilitate direct estimation of the underlying object intensity.

Because the complete data are independent, Poisson random variables, the complete-data log-likelihood is

$$\begin{aligned}\mathscr{L}^c(I) = &-\sum_k\sum_n\sum_m a[n]\gamma|\mathscr{Y}_n|h_k(y_n, x_m)I[m] \\ &+\sum_k\sum_n\sum_m N_k^c[n, m]\ln(a[n]\gamma|\mathscr{Y}_n|h_k(y_n, x_m)I[m])\end{aligned} \quad (30)$$

where terms not dependent upon the unknown object intensity $I[\cdot]$ have been omitted. Given an estimate for the object intensity $I^{\text{old}}[\cdot]$, the EM procedure makes use of the complete data and their corresponding log-likelihood to update the object intensity estimate in such a way that $I^{\text{new}}[\cdot]$ increases the measured data log-likelihood. The E-step of the EM procedure requires the expectation of the complete-data log-likelihood, conditional on the measured (or incomplete) data and using the old object intensity estimate $I^{\text{old}}[\cdot]$

$$\begin{aligned}Q(I; I^{\text{old}}) &= E[\mathscr{L}^c(I)|\{\tilde{d}_k[n]\}; I^{\text{old}}] \\ &= -\sum_k\sum_n\sum_m a[n]\gamma|\mathscr{Y}_n|h_k(y_n, x_m)I[m] \\ &\quad +\sum_k\sum_n\sum_m E[N_k^c[n, m]|\{\tilde{d}_k[n]\}; I^{\text{old}}] \\ &\quad \ln(a[n]\gamma|\mathscr{Y}_n|h_k(y_n, x_m)I[m])\end{aligned} \quad (31)$$

The intensity estimate is then updated in the M-step by maximizing this conditional expectation over I

$$I^{\text{new}} = \arg\max_{I\geq 0} Q(I; I^{\text{old}}) \quad (32)$$

It is straightforward to show that the object estimate is then updated according to

$$I^{\text{new}}[m] = \frac{\sum_k\sum_n E[N_k^c[n, m]|\{\tilde{d}_k[n]\}; I^{\text{old}}]}{\sum_k\sum_n a[n]\gamma|\mathscr{Y}_n|h_k(y_n, x_m)} \quad (33)$$

As described in Ref. 29, the conditional expectation is evaluated as

$$\begin{aligned}&E[N_k^c[n, m]|\{\tilde{d}_k[n]\}; I^{\text{old}}] \\ &\quad = \frac{a[n]\gamma|\mathscr{Y}_n h_k(y_n, x_m)I^{\text{old}}[m]}{\sum_{m'} a[n]\gamma|\mathscr{Y}_n h_k(y_n, x_{m'})I^{\text{old}}[m'] + I_b[n] + \sigma^2[n]}\tilde{d}_k[n]\end{aligned} \quad (34)$$

so that the iterative formula for updating the object estimate is

$$\begin{aligned}I^{\text{new}}[m] = I^{\text{old}}[m] &\\ \frac{\sum_k\sum_n h_k(y_n, x_m)\left[\frac{a[n]\gamma|\mathscr{Y}_n|\tilde{d}_k[n]}{\sum_{m'} a[n]\gamma|\mathscr{Y}_n h_k(y_n, x_{m'})I^{\text{old}}[m'] + I_b[n] + \sigma^2[n]}\right]}{\sum_k\sum_n a[n]\gamma|\mathscr{Y}_n|h_k(y_n, x_m)} &\end{aligned} \quad (35)$$

For the special case of uniform gain with no background or detector noise, the iterative algorithm proposed by Richardson (33) and Lucy (34) has the same form as these iterations. An excellent historical perspective of the application of the EM method to imaging problems is presented in Ref. 35, and detailed discussions of the convergence properties of this algorithm along with the pioneering derivations for applications in emission tomography can be found in Ref. 36.

Figures 2 and 3 illustrate the use of this technique on imagery acquired by the Hubble Space Telescope (HST). Shortly after the launch of the HST with its aberrated primary mirror in 1990, the imagery acquired by this satellite became a focus of national attention. Whereas microscopic flaws in the telescope's mirror resulted in the severely distorted imagery, image restoration methods were successful in restoring much of the lost resolution (4). Figure 2, for example, shows imagery of the star cluster R136 in a star formation called 30 Doradus as acquired by the telescope and as restored using the methods described in this article. Also shown in this figure are imagery acquired by the telescope after its aberrated mirror was corrected, along with a processed image showing the potential advantage of applying image restoration methods to imagery acquired after the correction. Figure 3 contains an image of Saturn along with restorations formed by simple inverse filtering, Wiener filtering, and by the maximum-likelihood method. According to scientific staff at the Space Telescope Science Institute, the maximum-likelihood restoration

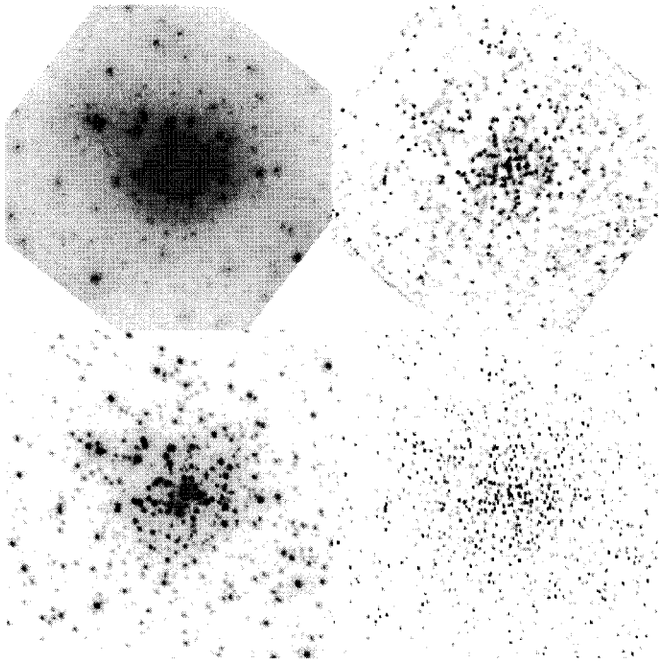


Figure 2. Imagery of the star cluster R136 in the star formation 30 Doradus as acquired by the Hubble Space Telescope both before and after its aberrated primary mirror was corrected. Upper left: raw data acquired with the aberrated primary mirror; upper right: restored image obtained from imagery acquired with the aberrated primary mirror; lower left: raw data acquired after correction; lower right: restored image obtained from imagery acquired after the correction. (Courtesy of R. J. Hanisch and R. L. White, Space Telescope Science Institute and NASA.)

provides the best trade-off between resolution and noise amplification.

Regularization. Under reasonably unrestrictive conditions, the EM method described in the previous section produces a sequences of images that converges to a maximum-likelihood solution (36). Imaging problems for which this method is applicable are often ill-conditioned or practically ill-posed, however, and because of this the maximum-likelihood image estimates frequently exhibit severe noise artifacts. Common methods for addressing this problem are discussed briefly in this section.

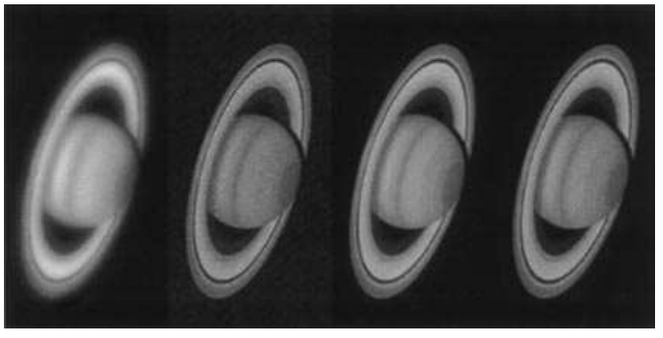


Figure 3. Raw imagery and restorations of Saturn as acquired by the Hubble Space Telescope. From left to right: telescope imagery; restoration produced by simple inverse filtering; restoration produced by Wiener filtering; restoration produced by the maximum-likelihood method. (Courtesy of R. J. Hanisch and R. L. White, Space Telescope Science Institute and NASA.

Stopping Rules. Probably the simplest method to implement for overcoming the noise artifacts seen in maximum-likelihood image estimates obtained by numerical procedures is to terminate the iterative process before convergence. Implementation of such a procedure is straightforward; however, the construction of optimal "stopping rules" can be challenging. Criteria for developing these rules for problems in coherent imaging are discussed in Refs. 21, 37, 38.

Sieve Methods. The basic idea behind the method of sieves is to constrain the set of allowable image estimates to be in a smooth subset called a sieve. The sieve is selected in a manner that depends upon the degree to which the problem is ill-conditioned and upon the noise level. Badly ill-conditioned problems and noisy data require a "small" sieve set containing only very smooth functions. Problems that are better conditioned with little noise can accommodate "large" sieve sets, and the sieve is ideally selected so that its "size" grows with decreasing noise levels in such a manner that the constrained image estimate converges to the true image as the noise level shrinks to zero. Establishing this consistency property for a sieve can, however, be a difficult task.

The general method of sieves as a statistical inference tool was introduced by Grenander (20). The application of this method to problems in incoherent imaging was proposed and investigated by Snyder et al. (39,40). The method is based on a kernel sieve defined according to

$$\mathscr{S} = \left\{ I : I[m] = \sum_p s[m, p]\alpha[p] \right\} \tag{36}$$

where intensity functions within the sieve set $\mathscr{S}$ are determined by the nonnegative parameters $\{\alpha[p]\}$. The sieve-constrained optimization problem then becomes one of maximizing the likelihood subject to the additional constraint $I \in \mathscr{S}$. The smoothness properties of the sieve are induced by the sieve kernel $s[\cdot, \cdot]$. With a Gaussian kernel, for instance, the smoothness of the sieve set is determined by the variance parameter σ

$$s[m, p] = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(m-p)^2}{2\sigma^2} \right) \tag{37}$$

This Gaussian kernel was investigated in Refs. 39, 40, but kernels with other mathematical forms can be used. The EM method can, with straightforward modifications, be applied to problems in which kernel sieves are used for regularization.

Penalty and MAP Methods. Another method for regularizing maximum-likelihood estimation problems is to augment the likelihood with a penalty function

$$\mathscr{C}(I) = \mathscr{L}(I) - \gamma\Phi(I) \tag{38}$$

where Φ is a function that penalizes undesirable qualities (or rewards desirable ones) of the image estimate, and γ is a nonnegative scale factor that determines the relative contribution of the penalty to the optimization problem. The penalized image estimate is then selected to maximize the cost function $\mathscr{C}$, which involves a trade between maximizing the likelihood $\mathscr{L}$

and minimizing the penalty Φ. The choice of the penalty can greatly influence the resulting image estimate, as can the selection of the scale factor γ. A commonly used penalty is the quadratic smoothness penalty

$$\Phi(I) = \sum_n \sum_{m \in \mathcal{N}_n} w_{nm}(I[n] - I[m])^2 \tag{39}$$

where $\mathcal{N}_n$ denotes a neighborhood of pixel locations about the nth object pixel, and the coefficients w_{nm} control the link between pixel n and m. This penalty can also be induced by using a MAP formulation with Gaussian Markov random field (GMRF) prior model for the object. However, because the use of this penalty often results in excessive smoothing of the object edges, alternative penalties have been developed and investigated (41–43). A particularly interesting penalty is induced by using a MAP formulation with the generalized Gaussian Markov random field (GGMRF) model (43). The use of this prior results in a penalty function of the form

$$\Phi(I) = \gamma^q \sum_n \sum_{m \in \mathcal{N}_n} w_{nm}|I[n] - I[m]|^q \tag{40}$$

where $q \in [1, 2]$ is a parameter that controls the smoothness of the reconstruction. For $q = 2$ this is the common quadratic smoothing penalty, whereas smaller values of q will, in general, allow for sharper edges in the object estimates.

Although the EM method is directly applicable to problems in which stopping rules or kernel sieves are used, the EM approach is less simple to use when penalty or MAP methods are employed. The major difficulty arises because the maximization step usually has no closed-form solution; however, approximations and modifications can be used (41,44) to address this problem.

Alternative Numerical Approaches

A major difficulty encountered when using the EM method for incoherent-imaging problems is its slow convergence (45). Many methods have been proposed to overcome this problem, and a few of these are summarized briefly here. Because of the similarities of the EM method to gradient ascent, line-search methods can be used to accelerate convergence (45), as can other gradient-based optimization methods (46,47). Substantial improvements in convergence can also be obtained by using a generalization of the EM method—the space-alternating generalized expectation-maximization (SAGE) method (28,48)—whereby convergence is accelerated through a novel choice for the complete data at each iteration. In addition, a coordinate descent (or ascent) optimization method has been shown to provide for greatly reduced computational time (49).

COHERENT IMAGING

For synthetic aperture radar (SAR), ultrasound, and other forms of coherent array imaging, an object or scene is illuminated by a highly coherent source (such as a radar transmitter, laser, or acoustic transducer), and heterodyne, homodyne, or holographic methods are used to record amplitude and phase information about the reflected field. The basic signal model for these problems is of the form:

$$u_p = \int h_p(x)u(x)\,dx + w_p, \quad p = 1, 2, \ldots, P \tag{41}$$

where p is an index to sensor locations (either real or synthetic), u_p is the complex-amplitude measured by the pth sensor, $u(x)$ is the complex-amplitude of the field that is reflected from an object or scene of interest, $h_p(x)$ is a sensor response function for the pth sensor measurement, and w_p accounts for additive sensor noise. The response function accounts for both the sensor characteristics and for wave propagation from the object or scene to the sensor; in the Fraunhofer approximation for wave propagation, these functions take on the form of a Fourier-transform kernel (10).

When the object or scene gives rise to diffuse reflections, the Gaussian speckle model (50) is often used as a statistical model for the reflected field $u(\cdot)$. That is, $u(\cdot)$ is modeled as a complex Gaussian random process (13,51,52) with zero-mean and the covariance

$$E[u(x)u^*(x')] \simeq s(x)\delta_2(x - x') \tag{42}$$

where $s(\cdot)$ is the object incoherent scattering function. The sensor noise is often modeled as zero-mean, independent complex Gaussian variables with variance σ^2 so that the recorded data are complex Gaussian random variables with zero-mean and the covariance

$$E[u_p u^*_{p'}] = \int h_p(x)h^*_{p'}(x)s(x)\,dx + \sigma^2\delta[p - p'] \tag{43}$$

where $\delta[\cdot]$ is the Kronecker delta function. The maximum-likelihood estimation of the object scattering function $s(\cdot)$ then becomes a problem of covariance estimation subject to the linear structure constraint of Eq. (43).

Using vector-matrix notation the data covariance is, as a function of the unknown object scattering function

$$\begin{aligned} \boldsymbol{R}(s) &= E[\boldsymbol{u}\boldsymbol{u}^\dagger] \\ &= \int \boldsymbol{h}(x)\boldsymbol{h}^\dagger(x)s(x)\,dx + \sigma^2\boldsymbol{I} \end{aligned} \tag{44}$$

where $\boldsymbol{u} = [u_1 u_2 \cdots u_P]^T$ is the data vector, $\boldsymbol{h}(x) = [h_1(x)h_2(x) \cdots h_P(x)]^T$ is the system response vector, $[\cdot]^T$ denotes matrix transposition, $[\cdot]\dagger$ denotes Hermitian matrix transposition, and $\boldsymbol{I}$ is the $P \times P$ identity matrix. Accordingly, the data log-likelihood is

$$L(s) = -\ln\det[\boldsymbol{R}(s)] - \mathrm{tr}[\boldsymbol{R}^{-1}(s)\boldsymbol{S}] \tag{45}$$

where $\boldsymbol{S} = \boldsymbol{u}\boldsymbol{u}\dagger$ is the data sample-covariance. Parameterization of the parameter function as in Eq. (23) is a natural step before attempting to solve this problem, but direct maximization of the likelihood is still a difficult problem. Because of this, the EM method has been proposed and discussed in Refs. 53–55 for addressing this problem, and the resulting algorithm has been shown to produce parameter estimates with lower bias and variance than alternative methods (56). A major problem with this method, though, is the high computational cost; however, the application of the SAGE method (28) to this problem has shown great promise for reducing the computational burden (57). The development and application of regularization methods for problems in coherent imaging is an area of active research.

SUMMARY

Imaging science is a rich and vital area of science and technology in which information-theoretic methods can be and

have been applied with great benefit. Maximum-likelihood methods can be applied to a variety of problems in image restoration and synthesis, and their application to the restoration problem for incoherent imaging has been discussed in great detail in this article. To conclude, the future of this field is best summarized by the following quote from Bracewell (58):

> The study of imaging now embraces many major areas of modern technology, especially the several disciplines within electrical engineering, and will be both the stimulus for, and recipient of, new advances in information science, computer science, environmental science, device and materials science, and just plain high-speed computing. It can be confidently recommended as a fertile subject area for students entering upon a career in engineering.

BIBLIOGRAPHY

1. H. Stark (ed.), *Image Recovery: Theory and Application,* Orlando, FL: Academic Press, 1987.
2. A. K. Katsaggelos (ed.), *Digital Image Restoration,* Heidelberg: Springer-Verlag, 1991.
3. R. L. Lagendijk and J. Biemond, *Iterative Identification and Restoration of Images,* Boston: Kluwer, 1991.
4. R. J. Hanisch and R. L. White (eds.), *The Restoration of HST Images and Spectra—II,* Baltimore, MD: Space Telescope Science Institute, 1993.
5. H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I,* New York: Wiley, 1968.
6. H. V. Poor, *An Introduction to Signal Detection and Estimation,* 2nd ed., New York: Springer-Verlag, 1994.
7. B. Porat, *Digital Processing of Random Signals: Theory and Methods,* Englewood Cliffs, NJ: Prentice-Hall, 1993.
8. L. L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis,* Reading, MA: Addison-Wesley, 1991.
9. S. M. Kay, *Modern Spectral Estimation: Theory and Applications,* Englewood Cliffs, NJ: Prentice-Hall, 1988.
10. J. W. Goodman, Introduction to Fourier Optics, 2nd edition, New York: McGraw-Hill, 1996.
11. J. D. Gaskill, *Linear Systems, Fourier Transforms, and Optics,* New York: Wiley, 1978.
12. M. Born and E. Wolf, *Principles of Optics,* 6th edition, Elmsford, NY: Pergamon, 1980.
13. J. W. Goodman, *Statistical Optics,* New York: Wiley, 1985.
14. B. E. A. Saleh and M. C. Teich, *Fundamentals of Photonics,* New York: Wiley, 1991.
15. A. Papoulis, *Probability, Random Variables, and Stochastic Processes,* 3rd edition, New York: McGraw-Hill, 1991.
16. R. M. Gray and L. D. Davisson, *Random Processes: An Introduction for Engineers,* Englewood Cliffs: Prentice-Hall, 1986.
17. A. Tikhonov and V. Arsenin, *Solutions of Ill-Posed Problems,* Washington, DC: Winston, 1977.
18. W. L. Root, Ill-posedness and precision in object-field reconstruction problems, *J. Opt. Soc. Am., A,* **4** (1): 171–179, 1987.
19. J. R. Thompson and R. A. Tapia, *Nonparametric Function Estimation, Modeling, and Simulation,* Philadelphia: SIAM, 1990.
20. U. Grenander, *Abstract Inference,* New York: Wiley, 1981.
21. E. Veklerov and J. Llacer, Stopping rule for the MLE algorithm based on statistical hypothesis testing, *IEEE Trans. Med. Imaging,* **6**: 313–319, 1987.
22. L. Mandel and E. Wolf, *Optical Coherence and Quantum Optics,* New York: Cambridge University Press, 1995.
23. D. L. Snyder, A. M. Hammoud, and R. L. White, Image recovery from data acquired with a charge-coupled-device camera, *J. Opt. Soc. Am., A,* **10** (5): 1014–1023, 1993.
24. D. L. Snyder et al., Compensation for readout noise in CCD images, *J. Opt. Soc. Am., A,* **12** (2): 272–283, 1995.
25. D. G. Luenberger, *Linear and Nonlinear Programming,* Reading, MA: Addison-Wesley, 1984.
26. R. Fletcher, *Practical Methods of Optimization,* New York: Wiley, 1987.
27. A. P. Dempster, N. M. Laird, and D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *J. R. Stat. Soc., B,* **39**: 1–37, 1977.
28. J. A. Fessler and A. O. Hero, Space-alternating generalized expectation-maximization algorithm, *IEEE Trans. Signal Process.* **42**: 2664–2677, 1994.
29. L. A. Shepp and Y. Vardi, Maximum-likelihood reconstruction for emission tomography, *IEEE Trans. Med. Imaging,* **MI-1**: 113–121, 1982.
30. D. L. Snyder and D. G. Politte, Image reconstruction from list-mode data in an emission tomography system having time-of-flight measurements, *IEEE Trans. Nucl. Sci.,* **NS-30**: 1843–1849, 1983.
31. K. Lange and R. Carson, EM reconstruction algorithms for emission and transmission tomography, *J. Comput. Assisted Tomography,* **8**: 306–316, 1984.
32. T. J. Holmes, Maximum-likelihood image restoration adapted for noncoherent optical imaging, *J. Opt. Soc. Am., A,* **6**: 666–673, 1989.
33. W. H. Richardson, Bayesian-based iterative method of image restoration, *J. Opt. Soc. Am.,* **62** (1): 55–59, 1972.
34. L. B. Lucy, An iterative technique for the rectification of observed distributions, *Astronom. J.,* **79** (6): 745–754, 1974.
35. Y. Vardi and D. Lee, From image deblurring to optimal investments: Maximum likelihood solutions for positive linear inverse problems, *J. R. Stat. Soc. B,* **55** (3): 569–612, 1993.
36. Y. Vardi, L. A. Shepp, and L. Kaufman, A statistical model for positron emission tomography, *J. Amer. Stat. Assoc.,* **80**: 8–37, 1985.
37. T. Hebert, R. Leahy, and M. Singh, Fast MLE for SPECT using an intermediate polar representation and a stopping criterion, *IEEE Trans. Nucl. Sci.,* **NS-34**: 615–619, 1988.
38. J. Llacer and E. Veklerov, Feasible images and practicle stopping rules for iterative algorithms in emission tomography, *IEEE Trans. Med. Imaging,* **MI-8**: 186–193, 1989.
39. D. L. Snyder and M. I. Miller, The use of sieves to stabilize images produced with the EM algorithm for emission tomography, *IEEE Trans. Nucl. Sci.,* **NS-32**: 3864–3871, 1985.
40. D. L. Snyder et al., Noise and edge artifacts in maximum-likelihood reconstructions for emission tomography, *IEEE Trans. Med. Imaging,* **MI-6**: 228–238, 1987.
41. P. J. Green, Bayesian reconstructions from emission tomography data using a modified EM algorithm, *IEEE Trans. Med. Imaging,* **9**: 84–93, 1990.
42. K. Lange, Convergence of EM image reconstruction algorithms with Gibbs priors, *IEEE Trans. Med. Imaging,* **MI-9**: 439–446, 1990.
43. C. A. Bouman and K. Sauer, A generalized Gaussian image model for edge-preserving MAP estimation, *IEEE Trans. Image Process.,* **2**: 296–310, 1993.
44. T. Hebert and R. Leahy, A generalized EM algorithm for 3-D Bayesian reconstruction from Poisson data using Gibbs priors, *IEEE Trans. Med. Imaging,* **MI-8**: 194–202, 1989.

45. L. Kaufman, Implementing and accelerating the EM algorithm for positron emission tomography, *IEEE Trans. Med. Imaging,* **MI-6**: 37–51, 1987.
46. E. U. Mumcuoglu et al., Fast gradient-based methods for Bayesian reconstruction of transmission and emission PET images, *IEEE Trans. Med. Imag.,* **MI-13**: 687–701, 1994.
47. K. Lange and J. A. Fessler, Globally convergent algorithm for maximum a posteriori transmission tomography, *IEEE Trans. Image Process.,* **4**: 1430–1438, 1995.
48. J. A. Fessler and A. O. Hero, Penalized maximum-likelihood image reconstruction using space-alternating generalized EM algorithms, *IEEE Trans. Image Process.,* **4**: 1417–1429, 1995.
49. C. A. Bouman and K. Sauer, A unified approach to statistical tomography using coordinate descent optimization, *IEEE Trans. Image Process.,* **5**: 480–492, 1996.
50. J. C. Dainty (ed.), Laser speckle and related phenomena. *Topics in Applied Physics,* vol. 9, 2nd ed., Berlin: Springer-Verlag, 1984.
51. F. D. Neeser and J. L. Massey, Proper complex random processes with applications to information theory, *IEEE Trans. Inf. Theory,* **39**: 1293–1302, 1993.
52. K. S. Miller, *Complex Stochastic Processes: An Introduction to Theory and Applications,* Reading, MA: Addison-Wesley, 1974.
53. D. B. Rubin and T. H. Szatrowski, Finding maximum likelihood estimates of patterned covariance matrices by the EM algorithm, *Biometrika,* **69** (3): 657–660, 1982.
54. M. I. Miller and D. L. Snyder, The role of likelihood and entropy in incomplete-data problems: Applications to estimating point-process intensities and Toeplitz constrained covariances, *Proc. IEEE,* **75**: 892–907, 1987.
55. D. L. Snyder, J. A. O'Sullivan, and M. I. Miller, The use of maximum-likelihood estimation for forming images of diffuse radar-targets from delay-doppler data, *IEEE Trans. Inf. Theory,* **35**: 536–548, 1989.
56. M. J. Turmon and M. I. Miller, Maximum-likelihood estimation of complex sinusoids and Toeplitz covariances, *IEEE Trans. Signal Process.,* **42**: 1074–1086, 1994.
57. T. J. Schulz, Penalized maximum-likelihood estimation of structured covariance matrices with linear structure, *IEEE Trans. Signal Process.,* **45**: 3027–3038, 1997.
58. R. N. Bracewell, *Two-Dimensional Imaging,* Upper Saddle River, NJ: Prentice-Hall, 1995.

TIMOTHY J. SCHULZ
Michigan Technological University

MEASUREMENT. See ACCELERATION MEASUREMENT; DENSITY MEASUREMENT; DISPLACEMENT MEASUREMENT; MAGNETIC FIELD MEASUREMENT; MILLIMETER WAVE MEASUREMENT; Q-FACTOR MEASUREMENT.

MEASUREMENT, ATTENUATION. See ATTENUATION MEASUREMENT.

MEASUREMENT, C-V. See C-V PROFILES.

} { { } }

Queueing Theory
Standard Article
Nader Mehravari[1]
[1]Lockheed Martin, Owego, NY

DOI: 10.1002/047134608X.W4208
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (456K)

Abstract

The sections in this article are

History of the Development of Queueing Theory

Applications of Queueing Theory

Specification and Characterization of Queueing Systems

Notions of Probability Theory of Importance to the Study of Queues

Modeling and Analysis of Elementary Queueing Systems

References To More Advanced Topics

QUEUEING THEORY

TELETRAFFIC THEORY

NETWORK OF QUEUES

All of us, either directly or through the use of various machines that we have become dependent upon, wait for service in a variety of lines on a regular basis. Customers wait in lines at banks to be served by a bank teller; drivers wait in their cars in traffic jams or at toll booths; patients wait in doctors' waiting rooms; electronic messages wait in personal computers to be delivered over communication networks; telephone calls are put on hold to be answered by operators; computer programs are stored in computer memory to be executed by a time-sharing computer system; and so on. In many situations, scarce resources are to be shared among a collection of users who require the use of these resources at unspecified times. They also require the use of these resources for random periods of time. This probabilistic nature of requests causes these requests to arrive while the resources are in use by other members of the user community. A mechanism must be put in place to provide an orderly access to the resources requested. The most common mechanism is to put the user requests in a waiting line or "queue." "Queueing theory" deals with the study of the behavior and the control of waiting lines. It provides us with the necessary mathematical structure and probability tools to model, analyze, study, evaluate, and simulate systems involving waiting lines and queues. It is a branch of applied mathematics, applied probability theory, and operations research. It is known under various names such as: queueing theory, theory of stochastic server systems, theory of systems of flows, traffic or teletraffic theory, congestion theory, and theory of mass service. Standard texts on queueing theory include Refs. 1–31. For a summary of many of the most important results in queueing theory, the reader is referred to a survey paper by Cooper (7). For a bibliography of books and survey papers on queueing theory see Refs. 8, 29. For nontechnical articles explaining queueing theory for the layman the reader is referred to Refs. 9, 26.

A typical queueing system can be described as one where customers arrive for service, wait for service, and, leave the system after being served. The service requests occur according to some stochastic process, and the time required for the server(s) to service a request is also probabilistically distributed. In general, arrivals and departures (i.e., service completions) cannot be synchronized, so waiting time may result. It is, therefore, critical to be able to characterize waiting time and many other important performance measures of a queueing system. For a typical queueing system, one is interested in answering questions such as: How long does a typical customer have to wait? What is the number of customers in the system at any given point in time? How large should the waiting room be to accommodate certain percentage of potential customers? How many servers are needed to keep the waiting time below a certain limit? What are subjective and economical advantages and disadvantages of modifying various parameters of the systems such as the number of servers or the size of the waiting room? How often is the server busy? Queueing theory attempts to answer these and other related questions through detailed mathematical analysis and provides us with the necessary tools to evaluate related performance measures.

The purpose of this article is to provide an introductory overview of the fundamental notions of queueing theory. The remaining sections of this article will discuss the following topics: a brief history of the development of queueing theory; applications of queueing theory; specification and characterization of queueing systems; notions of probability theory of importance to queueing theory; modeling and analysis of elementary queueing systems; references to more advanced topics; and a list of references.

HISTORY OF THE DEVELOPMENT OF QUEUEING THEORY

The English word "queue" is borrowed from the French word "queue" which itself is taken from the Latin word "cauda" meaning "tail." Most researchers and scientists in the field prefer the spelling "queueing" over "queuing." However, many American dictionaries and software spell checkers prefer the spelling "queuing." For further discussion of "queueing" vs. "queuing" spelling, see Refs. 27, 28. Queueing theory has been under development since the early years of this century. It has since progressed considerably, and today it is based upon a vast collection of results, methods, techniques, and voluminous literature. A good summary of the early history of queueing theory can be found in Ref. 6, pp. 20–25.

Historically, queueing theory originated as a very practical subject. It was developed to provide models to predict the behavior of systems that attempt to provide service for randomly arising demands. Much of the early work was developed in relation with problems in telephone traffic engineering. The pioneering work of Agner Krarup Erlang, from 1909 to 1929, laid the foundations of modern teletraffic and queueing theory. Erlang, a Danish mathematician and engineer who worked for the Copenhagen Telephone Exchange, published his first article in 1909 on the application of probability theory to telephone traffic problems (10). Erlang's work soon drew the attention of other probability theorists such as T. C. Fry and E. C. Molina in the 1920s, who expanded much of Erlang's work on the application of the theory to telephone systems. Telephony remained one of the principal applications until about 1950.

In the years immediately following World War II, activity in the fields of probability theory and operations research (11, 12) grew rapidly, causing a new surge of interest in the subject of queueing theory. In the late 1950s, queueing theory became one of the most popular subjects within the domains of applied mathematics and applied probability theory. This popularity, however, was fueled by its mathematical challenges and not by its applications. Clever and elegant mathematical techniques has enabled researchers (such as Pollaczek, Kolmogorov, Khin-

chine, Crommelin, and Palm) to derive exact solutions for a large number of mathematical problems associated with models of queueing systems. Regrettably, in the period of 1950–1970, queueing theory, which was originated as a very practical subject, had become of little direct practical value.

Since the 1970s there has seen a rebirth and explosion of queueing theory activities with an emphasis on practical applications. The performance modeling and analysis of computer systems and data transmission networks opened the way to investigate queues characterized by complex service disciplines and interconnected systems. Most of the theoretical advances since the 1970s are directly attributable to developments in the area of computer systems performance evaluation as represented in Refs. 13–16.

APPLICATIONS OF QUEUEING THEORY

Interest in queueing theory has often been stimulated by practical problems and real world situations. Queueing theory concepts have applications in many disciplines such as telephone systems traffic engineering, migration and population models in biology, electrical and fluid flow models, clustering models in chemistry, manufacturing systems, computer systems, digital data transmission systems, flow through communication networks, inventory control, time sharing and processor sharing computer systems, telecommunications, machine repair, taxi stands, aircraft landing, loading and unloading ships, scheduling patients in hospitals, factory production flow, intelligent transportation systems, call centers, and so on. There are many other important applications of the queueing theory as presented in Refs. 1–6 and 13–16. We elaborate further on only two of these applications in this section.

Queueing theory has played a major role in the study of both packet switching and circuit switching communication networks. Queueing arises naturally in packet switching networks where user messages are broken into small units of transmission called packets. Packets arriving at various intermediate network nodes, on the way to their final destination, are buffered in memory, processed to determine the appropriate outgoing route, and then are transmitted on the chosen outgoing link when their time for transmission comes up. If, for example, the chosen outgoing link is in use when it is time for a given packet to be transmitted, then that packet must be kept in the memory (i.e., queued) until the link becomes available. The time spent in the buffer waiting for transmission is an important measure of system performance. This waiting time depends on various parameters such as nodal processing power, transmission link speed, packet lengths, traffic rates in terms of packets per second, and so on. Queueing theory provides the necessary mathematical tools to model and analyze such queueing configurations.

For another example of application of queueing theory consider a typical bank and the mechanism that bank management has put in place to direct incoming customers to the available bank tellers. In some banks, each teller has his or her own queue and incoming customers are free to join the waiting line of any of the tellers based on some personal preferences. Some customers often join the shortest queue, and some join the queue of a particular teller that they personally know, whereas others may join the queue of the teller that is perceived to be the fastest. On the other extreme, some banks (via the use of various directional signs and/or ropes) direct all the incoming customers into a single waiting line that feeds all the tellers. The customer at the head of this queue is then served by the next available bank teller. The question now becomes which one of these two modes of operation is more appropriate. The answer strongly depends on such parameters as the performance measures that the bank management is interested in optimizing, the number and the speed of the tellers, the type of banking transactions, and the number of incoming customers visiting the bank in a typical day. Similar issues arise in other cases such as supermarket checkout counters, fast-food restaurants, airport landing and departure schedules, and multiprocessor computer systems. Queueing theory methods enable us to model, analyze, and decide on the best strategy for such applications.

SPECIFICATION AND CHARACTERIZATION OF QUEUEING SYSTEMS

Figure 1 represents the basic elements of a queueing system. As shown in Fig. 1, a basic queueing system is one where members of a population (i.e., customers or entities of some kind) arrive at a service station to receive service of some type. After receiving service, the units depart the service facility. A "queue" or waiting line is developed whenever the service facility cannot service all the units requiring service. Although many queueing systems may be represented by similar diagrams, an accurate representation of such a system requires a detailed characterization of the underlying parameters and processes.

Key Parameters and Varieties of Queueing Systems

To fully describe a queueing system analytically, various aspects and parameters of the system must be known. The most important of them are presented here.

The Arrival Pattern. Let the successive customers arrive to the system at times $t_1, t_2, t_3, \ldots$, where $0 \leq t_1 < t_2 < t_3 < \cdots < t_n < \cdots$. Then we define $y_i = t_{i+1} - t_i$, where $i = 1, 2, 3, \ldots$, as the interarrival times of the customers. We normally assume that arrival times form a stochastic process and that the interarrival times, y_i, are independent and identically distributed (iid) according to probability distribution function $A(\cdot)$, where $A(\tau) = P(y_i \leq \tau)$. Function $A(\cdot)$ is then referred to as the interarrival time distribution or simply the arrival distribution. Additional information such as whether each arrival event contains one or a group of customers of fixed or random size (i.e., "bulk arrivals") can also be specified if applicable.

Customer Population and Behavior. The customer population, or the source of the customers, can either be finite or infinite. Infinite customer populations are normally easier to describe mathematically and analyze their performance analytically. This is because in a finite population source

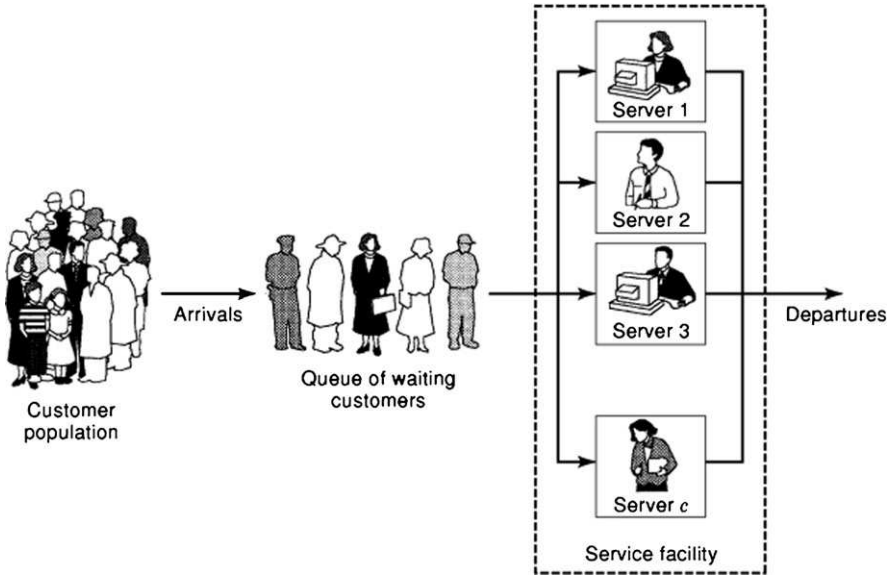


Figure 1. Basic elements of a typical queueing system.

model, the number of customers in the system affects the arrival rate which in turn makes the analysis more difficult. In addition to the properties of the entire customer population, behavior of individual customers could also be of importance and, therefore, must be formally specified. For example, if a customer decides not to join the system after seeing the size of the queue, it is said that the customer has "balked." Or, for example, a customer is said to have "reneged" if, after having waited in the queue for some time, he or she becomes impatient and leaves the system before his service begins. Customers, if allowed, may "jockey" from one queueing system to another (with a perceived shorter waiting time, for example).

The Service Mechanism. The queue's service mechanism is described by specifying the number of servers, c, and the stochastic characterization of the service times. It is normally assumed that the service times of successive customers, $x_1, x_2, x_3, \ldots$, are iid with probability distribution $B(\cdot)$, where $B(\tau) = P(x_i \leq \tau)$, and are also independent of the interarrival times $y_1, y_2, y_3, \ldots$. Additional information such as whether the customers are served individually or in groups (of fixed or random size) can also be specified if applicable.

The Queueing Discipline. The queueing discipline is the description of the mechanism for determining which of the waiting customers gets served next, along with the associated rules regarding formation of the queue. The most basic queueing disciplines are listed and described below:

1. First-Come First Served (*FCFS*) or First-In First-Out (*FIFO*) The waiting customers are served in the order of their arrival times.
2. Last-Come First-Served (*LCFS*) or Last-In First-Out (*LIFO*) The customer who has arrived last is chosen as the one who gets served when a server becomes available.
3. Service in Random Order (*SIRO*) or Random Selection for Service (*RSS*) The customer to be served next is chosen stochastically from the waiting customers according to a uniform probability distribution. In general, the probability distribution used to choose the next customer could be any discrete probability distribution.
4. Priority (*PR* or *PRI*) There could also be some notion of priority in the queueing system where the customer population is divided in two or more priority classes. Any waiting member of a higher priority class is chosen to be served before any customer from a lower priority class. Queueing systems with priority classes are divided into two types. Under a "preemptive priority" discipline, whenever a higher priority customer arrives while a lower priority customer is in service, the lower priority customer is preempted and is taken out of service without having his service completed. In this case, the preempted customer is placed back in the queue ahead of all customers of the same class. Under the "non-preemptive priority" discipline, once the service of any customer is started, it is allowed to be completed regardless of arrivals from higher priority classes. Moreover, the preemptive priority queueing systems can further be divided into two types. Under the discipline of "preemptive resume," whenever a preempted customer reenters service he simply continues his service where he left off. Under "preemptive repeat," a preempted customer draws a new value of service time from the service time distribution each time it reenters service.

Maximum Number of Customers Allowed. In many systems the capacity of queueing system is assumed to be infinite, which implies that every arriving customer is allowed to join the queue and wait until served. However, in many real-life situations, the queueing systems have either no or only a finite amount of capacity for customers to wait.

In a queueing system with no room for customers to wait, whenever all the servers are busy, any additional arriving customer is turned away; this type of system is referred to as "loss systems." Loss systems have been used to model the behavior of many dial-up telephone systems and telephone switching equipment. Queueing systems with a positive but finite waiting room have been deployed to characterize the performance of various computing and telecommunications systems where the finite waiting room models the finite amount of memory or buffer present in such real-world systems.

Number of Servers. In general a queueing system can have either one, or finitely many, or an infinite number of servers. "Single-server systems" are the simplest ones where a maximum of one user can be served at any given point in time. A "multiserver system" contains c servers, where $0 < c < \infty$, and can serve up to c simultaneous customers at any given point in time. An "infinite-server system" is one in which each arriving customer is immediately provided with an idle server.

Performance Measures

In any queueing system there are many performance tradeoffs to be considered. For example, if the number of servers in the system is so large that queues rarely form, then the servers are likely to be idle a large fraction of time, resulting in wasting of resources and extra expense. On the other hand, if almost all customers must join long queues, and servers are rarely idle, there might be customer dissatisfaction and possibly lost customers which again has negative economical consequences. Queueing theory provides the designer the necessary tools to analyze the system and ensure that the proper level of resources are provided in the system while avoiding excessive cost. The designer can accomplish this, for example, by considering several alternative system architectures and by evaluating each by queueing theory methods. In addition, the future performance of an existing system can also be predicted so that upgrading of the system can be achieved in a timely and economical fashion. For example, an analytical queueing model of a computer communication network might indicate that, in its present configuration, it cannot adequately support the expected traffic load two years in the future. The model may make it possible to evaluate different alternatives for increased capacity such as increasing the number of nodes in the network, increasing the computing power of existing nodes, providing more memory and buffer space in the network nodes, increasing the transmission speeds of the communication links, or increasing the number of communication links. Determining the most appropriate solution can be done through careful evaluation of various performance measures of the queueing systems.

The following performance measures represent some of the most common and important aspects of queueing systems which are normally investigated:

1. The Queue Length This performance measure is related to the number of customers waiting in the system. Some authors use this term to represent only the number of customers in the queue proper (i.e., not including the one or more customers who are being served), and others use it to represent the total number of customers in the system. In the former case it is often referred to as the "queue length," and in the latter case it is often referred to as the "number in the system."
2. The Waiting Time This performance measure is related to the amount of time spent by a customer in the system. This term is used in two different ways. Some authors use the term to refer to the total time spent by a customer in the queueing system, which is the sum of the time spent by the customer in the waiting line before service and the service time itself. Others define it as only the time spent in the queue before the service. In the former case it is often referred to as the "system time," and in the latter case it is often referred to as the "queueing time."
3. The Busy Period This is the length of time during which the server is continuously busy. Any busy period begins when a customer arrives at an empty system, and it ends when the number of customers in the system reaches zero. The time period between two successive busy periods is referred to as the "idle period" for obvious reasons.

Kendall's Notation for Queueing Systems

It is a common practice to use a short-hand notation of the form $A/B/c/K/m/Z$ to denote various aspects of a queueing system. This notation is referred to as *Kendall's notation*. This type of short-hand was first developed by Kendall (17) and later extended by Lee (18). It defines some of the basic parameters which must be known about a queue in order to study its behavior and analyze its performance. In Kendall's notation $A/B/c/K/m/Z$, A describes the interarrival time distribution, B describes the service time distribution, c is the number of (parallel) servers, K is the maximum number of customers allowed in the system (waiting plus in service), m is the size of the customer population, and Z describes the queue discipline. The traditional symbols used in the first and second positions of Kendall's notation, and their meanings, are:

M	Exponentially distributed interarrival time or service time distribution
D	Deterministic (i.e., constant) interarrival time or service time distribution
E_k	k-stage Erlangian (Erlang-k) interarrival time or service time distribution
H_k	k-stage Hyperexponential interarrival time or service time distribution
G	General interarrival or service time distribution

The third, fourth, and fifth positions in Kendall's notation could be any positive integer. The traditional symbols used in the last position of Kendall's notation are: FCFS, FIFO, LCFS, LIFO, SIRO, RSS, PR, and PRI, as described earlier in this section; and also GD, which refers to a general queue discipline.

As an example of Kendall notation, an M/D/2/50/∞/SIRO queueing system is one with exponential interarrival time, constant service time, 2 parallel servers, a system capacity of 50 (i.e., a maximum of 48 in the queue and 2 in service), a customer population that is infinitely large, and the waiting customers are served in a random order.

Whenever the last three elements of Kendall's notation are omitted, it is meant that $K = \infty$, $m = \infty$, and Z = FCFS (i.e., there is no limit to the queue size, the customer source is infinite, and the queue discipline is FCFS). As an example of the shorter version of Kendall's notation, an $M/M/1$ queue has Poisson arrivals, exponential service time, and 1 server, there is no limit to the queue size, the customer source is infinite, and the queue discipline is FCFS.

It should be noted that although Kendall's notation is quite useful and very popular, it is not meant to characterize all possible models and configurations of queueing systems. For example, Kendall's notation is normally not used to indicate bulk arrivals, or queues in series, and so on.

NOTIONS OF PROBABILITY THEORY OF IMPORTANCE TO THE STUDY OF QUEUES

Probability theory has a major and fundamental role in the study and analysis of queueing models. As mentioned earlier, queueing theory is considered a branch of applied probability theory. It is assumed here that the reader is familiar with the basic notions of elementary probability theory such as notions of events, probability, statistical independence, distribution and density functions, and expectations or averages. The reader is referred to Ref. 19 for a complete treatment of probability theory. Here we discuss a few aspects of probability notions which are of great importance to the study of queues.

Probability Distributions of Importance to Queueing Theory

As is indicative of Kendall's notation, queueing theory deals with a large number of different types of probability distributions to mathematically model the behavior of customer interarrival times and the customer service times. In the rest of this section, we briefly describe some of the most important probability distributions that are used often in various queueing theory analysis.

Exponential Probability Distribution. The probability distribution most commonly assumed for customer interarrival time and for customer service times in queueing models is the exponential distribution. This popularity is due to its pleasant mathematical properties which often result in much simplification of the analytical work. A continuous random variable X has an exponential distribution with parameter $\lambda > 0$ if its density function $f(\cdot)$ is defined by

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{for } x \geqslant 0 \\ 0 & \text{for } x < 0 \end{cases} \tag{1}$$

It distribution function is given by

$$F(x) = \begin{cases} 1 - e^{-\lambda x} & \text{for } x \geqslant 0 \\ 0 & \text{for } x < 0 \end{cases} \tag{2}$$

Both its mean and its standard deviation are equal to $1/\lambda$.

The exponential distribution is unique among the continuous distributions because it has the so-called "memoryless property" or "Markov property." The memoryless property is that if we know that a random variable has an exponential distribution, and we know that the value of the random variable is at least some value, say t, then the distribution for the remaining value of the variable (i.e., the difference between the total value and t) has the same exponential distribution as the total value. That is,

$$P(X > t + h | X > t) = P(X > h) \quad \text{for } t > 0, h > 0 \tag{3}$$

Another interpretation of Eq. (3) is that, if X is the waiting time until a particular event occurs and t units of time have produced no event, then the distribution of further waiting time is the same as it would be if no waiting time had passed; that is, the system does not "remember" that t time units have produced no "arrival."

Poisson Probability Distribution and Poisson Random Process. Poisson random variable is used in many applications where we are interested in counting the number of occurrences of an event (such as arrivals to a queueing system) in a certain time period or in a region of space. Poisson random variables also appear naturally in many physical situations. For example, the Poisson probability mass function gives an accurate prediction for the relative frequencies of the number of particles emitted by a radioactive mass during a fixed time period. A discrete random variable X is said to have a Poisson distribution with parameter $\lambda > 0$ if X has a probability mass function of the form

$$P(k; \lambda) = P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda} \quad \text{for } k = 0, 1, 2, 3, \ldots \tag{4}$$

Both the mean and the standard deviation of the Poisson random variable are equal to λ.

Now consider a situation in which events occur at random instants of time at an average rate of λ events per second. For example, an event could represent the arrival of a customer to a service station or the breakdown of a component in some system. Let $N(t)$ be the number of event occurrences in the time interval $[0, t]$. $N(t)$ is then a nondecreasing, integer-valued, continuous-time random process. Such a random process is said to be a Poisson process if the number of event occurrences in the time interval $[0, t]$ has a Poisson distribution with mean λt. That is,

$$P(N(t) = k) = \frac{(\lambda t)^k}{k!} e^{-\lambda t} \quad \text{for } k = 0, 1, 2, 3, \ldots \tag{5}$$

Like the exponential distribution, Poisson process also has a number of unique properties which has made it very at-

tractive for analytical studies of queueing systems. In particular, Poisson process has a "memoryless property"; occurrence of events during a current interval of time is independent of occurrences of events in previous intervals. In other words, events occurring in nonoverlapping intervals of time are independent of each other. Furthermore, the interevent times (i.e., interarrival times in case of queueing system) in a Poisson process from an iid sequence of exponential random variables with mean $1/\lambda$.

Erlang-k Probability Distribution. A. K. Erlang (10) used a special class of gamma random variables (19), now often called "Erlang-k" or "k-stage Erlangian," in his study of delays in telephone traffic. A random variable, T, is said to be an Erlang-k random variable with parameter λ or to have an Erlang distribution with parameters k and λ, if T is gamma random variable with the density function f given by

$$f(x) = \begin{cases} \dfrac{\lambda k(\lambda k x)^{k-1}}{(k-1)!} e^{-\lambda k x} & \text{for } x \geqslant 0 \\ 0 & \text{for } x < 0 \end{cases} \tag{6}$$

The mean and variance of Erlang-k random variable are $1/\lambda$ and $1/(k\lambda^2)$, respectively. An Erlang-k random variable can be obtained by adding k independent exponentially distributed random variables each with parameter λk. The physical model that Erlang had in mind was a service facility consisting of k identical independent service substations connected in series one after another, each with an exponential distribution of service time. He wanted this special facility to have the same average service time as a single service facility whose service time was exponential with parameter λ. Thus the service time, T, for the facility with k stages could be written as the sum of k exponential random variables, each with parameter λk.

Hyperexponential Probability Distribution. If the service time of a queueing system has a large standard deviation relative to the mean value, it can often be approximated by a hyperexponential distribution. The model representing the simplest hyperexponential distribution is one with two parallel stages in the facility; the top one having exponential service with parameter μ_1, and the bottom stage having exponential service with parameter μ_2. A customer entering the service facility chooses the top stage with probability α_1 or the bottom stage with probability α_2, where $\alpha_1 + \alpha_2 = 1$. After receiving service at the chosen stage, with the service time being exponentially distributed with average service rate μ_i, the customer leaves the service facility. A new customer is not allowed to enter the facility until the original customer has completed service. The probability density function for the service time, the probability distribution function, mean, and variance are given by

$$f_x(t) = \alpha_1\mu_1 e^{-\mu_1 t} + \alpha_2\mu_2 e^{-\mu_2 t} \qquad \text{for } t \geq 0 \tag{7}$$

$$F_x(t) = P(x \leq t) = 1 - \alpha_1 e^{-\mu_1 t} - \alpha_2 e^{-\mu_2 t} \quad \text{for } t \geq 0 \tag{8}$$

$$E[x] = \frac{\alpha_1}{\mu_1} + \frac{\alpha_2}{\mu_2} \tag{9}$$

$$\text{Var}[x] = \frac{2\alpha_1}{\mu_1^2} + \frac{2\alpha_2}{\mu_2^2} - \left(\frac{\alpha_1}{\mu_1} + \frac{\alpha_2}{\mu_2}\right)^2 \tag{10}$$

The two-stage hyperexponential distribution described above can be generalized to k stages for any positive integer greater than 2.

Notions of Transient and Steady State

Analysis of a queueing system often involves the study of the system's characteristics over time. A system is defined to be in "transient state" if its behavior and associated performance measures are dependent on time. This usually occurs at the early stages of the operation of the system where its behavior is heavily dependent on the initial state of the system. A system is said to be in "steady state" or "equilibrium" when the behavior of the system becomes independent of time. This usually occurs after the system has been in operation for a long time, and the influence of initial conditions and of the time since start-up have diminished. In steady state, the number of customers in the system and in the queue are independent of time.

A necessary condition for a queueing system to reach steady state is that the elapsed time since the start of the operation is mathematically long enough (i.e., the limit as time tends to infinity). However, this condition is not sufficient to guarantee that a queueing system is in steady state. In addition to elapsed time, particular parameters of the queueing system itself will have an effect on whether and when the system reaches steady state. For example, if the average arrival rate of customers is higher than the overall average service rate of the system, then the queue length will continue to grow forever and steady state will never be reached. Although many authors have studied the transient behavior of queueing systems, the majority of the key results and existing literature deal with steady-state behavior of queueing systems.

Random Variables of Interest

In this section we define and list the key random variables and associated notations used in queueing theory and in the rest of this article. Some of the primary random variables and notations are graphically illustrated in Fig. 2 and many more are listed in Table 1. Clearly, there are some obvious relationships between some of the random variables listed in Fig. 2 and/or Table 1. For example, with respect to the number of customers in the system, we must have

$$N(t) = N_q(t) + N_s(t) \tag{11}$$

and

$$N = N_q + N_s \tag{12}$$

In Eq. (12), it is assumed that the queueing system has reached the steady state. It should, however, be noted that although the system is in steady state, quantities N, N_q, and N_s are random variables; that is, they are not constant and have probability distributions associated with them. In other words, "steady state" means that the probabilities are independent of time but not that the system becomes deterministic.

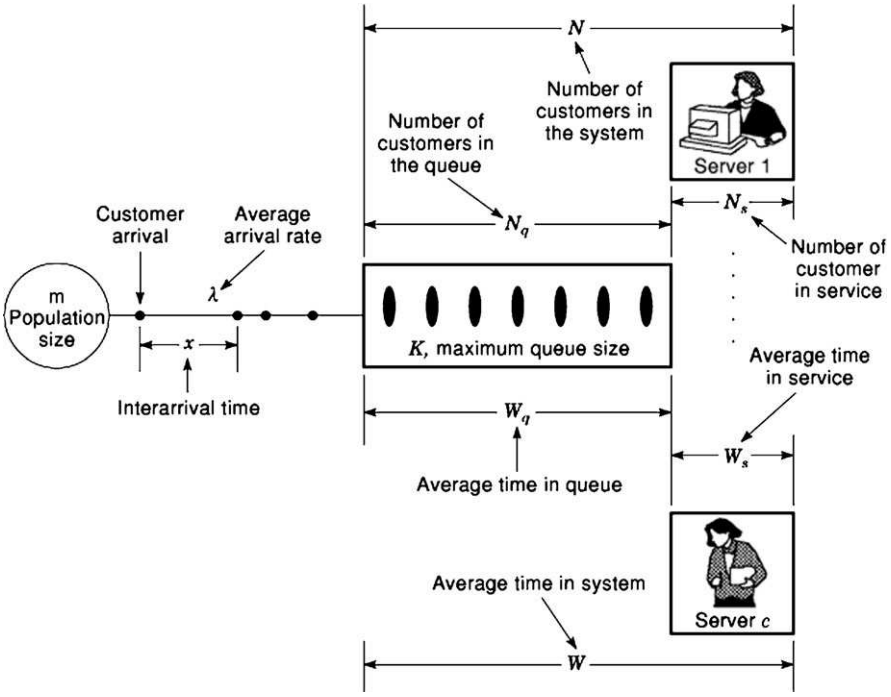


Figure 2. Graphical representation of some variables of importance to queueing theory.

Table 1. Definition of Some of the Key Variables of Importance to Queueing Theory

y	Random variable representing customer interarrival times
λ	Average arrival rate of customers to the system
$1/\lambda$	Average interarrival time of customers to the system
x	Random variable representing customer service times
μ	Average service time per customer
$1/\mu$	Average service rate per customer
c	Number of servers
ρ	Traffic intensity or offered load
$N(t) \to N$	Random variables representing number of customers in the system at time t and in steady state
$N_q(t) \to N_q$	Random variables representing number of customers in the queue at time t and in steady state
$N_s(t) \to N_s$	Random variables representing number of customers in service at time t and in steady state
$P_n(t) \to P_n$	Transient and steady-state probability of having exactly n customers in the system
L	Average steady-state number of customers in the system
L_q	Average steady-state number of customers in the queue
L_s	Average steady-state number of customers in service
$w(x)$	Random variable representing steady state probability distribution of the waiting time in the system
W	Average steady-state time spent by a customer in the system
W_q	Average steady-state time spent by a customer in the queue
W_s	Average steady-state time spent by a customer in service

Applying expectations operation to both sides of Eq. (12), we get

$$L = L_q + L_s \tag{13}$$

There are similar obvious relationships between some of the random variables related to waiting times. For example, the total time in the queueing system for any customer is the sum of his waiting time in the queue and his service time, that is,

$$W = W_q + W_s \tag{14}$$

We are clearly interested in studying relationships between other random variables and parameters of the interest which might not be as obvious as those given in Eqs. (11)–(14). Development of such relationships are a major byproduct of modeling and analysis of queueing systems, as will be discussed in the next section.

MODELING AND ANALYSIS OF ELEMENTARY QUEUEING SYSTEMS

In this section we present, in some detail, some of the key techniques used by queueing theory community to model and analyze some of the elementary queueing models. In particular, we will illustrate the application of birth-and-death stochastic processes to the analysis of these models.

Little's Formula

Little's formula (which is also known as "Little's result" and "Little's theorem") is one of the most fundamental and often used results in queueing theory. It provides a simple, but very general, relationship between the average waiting time and the average number of customers in a queueing system. Its first rigorous proof in its general form was given by J. D. C. Little (20). Its validity and proofs of some special cases, however, were known to researchers prior to Little's proof. Consider an arbitrary queueing system in

steady state. Let L, W, and λ be the average number of customers in the system, average time spent by customers in the system, and average number of customer arrivals per unit time, respectively. Little's theorem states that

$$L = \lambda W \tag{15}$$

regardless of the interarrival and service time distributions, the service discipline, and any dependencies within the system.

Rigorous proof of Little's theorem is given in every standard queueing theory text (1–6). What follows is an intuitive justification of Little's result given in Ref. 12. Suppose that the system receives a reward (or penalty) of 1 for every unit of time that a customer spends in it. Then the total expected reward per unit time is equal to the average number of customers in the system, L. On the other hand, the average number of customers coming into the system per unit time is λ; the expected reward contributed by each customer is equal to his average residence time, W. Since it does not matter whether the reward is collected on arrival or continuously, we must have $L = \lambda W$. A different interpretation of Little's result is obtained by rewriting it as $\lambda = L/W$. Since a customer in the system remains there for an average time of W, his average rate of departure is $1/W$. The total average departure rate is, therefore, L/W. Thus, the relation holds if the average arrival rate is equal to the average departure rate. But the latter is clearly the case since the system is in equilibrium.

It is important to note that we have not even specified what constitutes "the system," nor what customers do there. It is just a place where customers (entities) arrive, remain for some time, and then depart after having received service. The only requirement is that the processes involved should be stationary (i.e., system should be in steady state). Therefore, we can apply Little's theorem not only to the entire queueing system [as represented by Eq. (15)], but also to particular subsections of it. For example, applying Little's theorem to only the waiting line portion of a $G/G/c$ queueing system, where $1 \leq c \leq \infty$, results in

$$L_q = \lambda W_q \tag{16}$$

where L_q and W_q are as defined in Table 1. Now consider another situation, where the "system" is defined as the "set of c servers" in a $G/G/c$ queueing system, where $1 \leq c \leq \infty$. Since every incoming customer enters a server eventually, the rate of arrivals into the "set of c servers" is also λ. The average time a customer spends in the system here is simply $1/\mu$. According to Little's theorem, the average number of customers in the system is therefore λ/μ. Thus in any $G/G/c$ or $G/G/\infty$ system in steady state, the average number of busy servers is equal to the traffic intensity, ρ. When $c = 1$, the average number of busy servers is equal to the probability that the server is busy. Therefore, in any single-server system in the steady state we have

$$P(\text{there are customers in the system}) = \rho \tag{17}$$

$$P(\text{system is idle}) = 1 - \rho \tag{18}$$

Birth-and-Death Process

Most elementary queueing models assume that the inputs (i.e., arriving customers) and outputs (i.e., departing customers) of the queueing system occur according to the so-called "birth-and-death process." This important process in probability theory has application in other areas also. However, in the context of queueing theory, the term "birth" refers to the arrival of a new customer and the term "death" refers to the departure of a served customer. The state of the system at time t, for $t \geq 0$, is given by random variable $N(t)$ defined as the number of customers in the system at time t. Thus the birth-and-death process describes probabilistically how $N(t)$ changes at t increases.

Formally speaking, a stochastic process is a birth-and-death process if it satisfies the following three assumptions: (1) Given $N(t) = n$, the current probability distribution of the remaining time until the next birth is exponentially distributed with parameter λ_n for $n = 0, 1, 2, \ldots$; (2) given $N(t) = n$, the current probability distribution of the remaining time until the next death is exponentially distributed with parameter μ_n for $n = 0, 1, 2, \ldots$; and (3) only one birth or death can occur at a time. Figure 3, which shows the state transition diagram of a birth-and-death process, graphically summarizes the three assumptions just described. The arrows in this diagram show the only possible transitions in the state of the system, and the label for each arrow gives the mean rate for the transition when the system is in the state at the base of the arrow.

Except for a few special cases, analysis of the birth-and-death process is very difficult when the system is in a transient condition. On the other hand, it is relatively easy to derive the probability distribution of the number of customers in the system in steady state. In steady state, the probability of finding the system in a given state does not change with time. In particular, the probability of there being more than k customers in the system is constant. The transition from state k to state $k + 1$ increases this probability, and the transition from state $k + 1$ to state k decreases it. Therefore, these two transitions must occur at the same rate. If this were not so, the system would not be in steady state. This yields to the following key principle: In equilibrium, the average rate into any state is equal to the average rate out of that state. This basic principle can be used to generate a set of equations called the "balance equations." After constructing the balance equations for all the states in terms of the unknown probabilities P_n, this system of equations can then be solved to find these probabilities. As shown in Fig. 3, there are only two transitions associated with state zero which result in the following balance equation for that state:

$$\mu_1 P_1 = \lambda_0 P_0 \tag{19}$$

There are four transitions associated with state 1 resulting in the following balance equation for that state:

$$\lambda_0 P_0 + \mu_2 P_2 = (\lambda_1 + \mu_1) P_1 \tag{20}$$

Balance equations for states $n \geq 2$ are similar to that of state 1 and can be easily be generated by inspecting the associated transitions in Fig. 3. This collection of balance

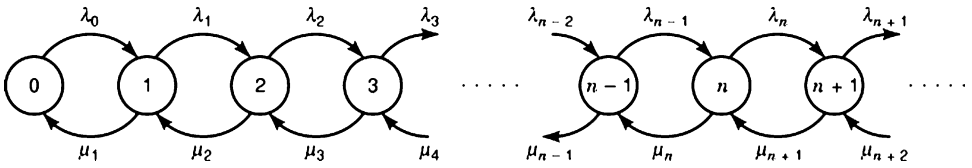


Figure 3. State transition diagram for a birth-and-death process.

equations along with the auxiliary equation

$$\sum_{n=0}^{\infty} P_n = 1 \tag{21}$$

can be solved for P_n, $n = 0, 1, 2, 3, \ldots$, resulting in the following set of steady-state probabilities for the number of customers in the system:

$$P_0 = \frac{1}{1 + \sum_{n=1}^{\infty} C_n} \tag{22}$$

$$P_n = C_n P_0 \quad \text{for } n = 1, 2, 3, \ldots \tag{23}$$

where

$$C_n = \frac{\lambda_{n-1}\lambda_{n-2}\cdots\lambda_0}{\mu_n \mu_{n-1}\cdots\mu_1} \quad \text{for } n = 1, 2, 3, \ldots \tag{24}$$

Given these expressions for the steady-state probability of number of customers in the system, we can derive the average number of customers in the system by

$$L = \sum_{n=0}^{\infty} n P_n \tag{25}$$

These steady-state results have been derived under the assumption that the λ_n and μ_n parameters are such that the process actually can reach a steady-state condition. This assumption always holds if $\lambda_n = 0$ for some value of n, so that only a finite number of states (those less than n) are possible. It also always holds when $\lambda_n = \lambda$ and $\mu_n = \mu$ for all n and when $\rho = \lambda/\mu < 1$.

M/*M*/1 Queue

Consider the simplest model of a nontrivial queueing model. This model assumes a Poisson arrival process (i.e., exponentially distributed interarrival times), an exponentially distributed service time, a single server, infinite queue capacity, infinite population of customers, and FCFS discipline. If the state of the system at time t, for $t \geq 0$, is given by the random variable $N(t)$, defined as the number of customers in the system at time t, it represents a birth-and-death process with rates

$$\lambda_n = \lambda \quad \text{and} \quad \mu_n = \mu \quad \text{for } n = 0, 1, 2, 3, \ldots \tag{26}$$

Therefore, by using Eqs. (22)–(24), we get

$$P_n = (1 - \rho)\rho^n \quad \text{for } n \geq 0, \rho < 1 \tag{27}$$

where $\rho = \lambda/\mu$. The mean number of customers in the system can now be computed as

$$L = E[N] = \sum_{k=0}^{\infty} k p_k = \frac{\rho}{1 - \rho} \tag{28}$$

Having found the mean number of customers in the system, we can now use Little's formula to determine the average total waiting time, W, as follows:

$$W = \frac{L}{\lambda} = \frac{\rho/\lambda}{1 - \rho} = \frac{1/\mu}{1 - \rho} = \frac{1}{\mu - \lambda} \tag{29}$$

Behavior of the average number of customers in the system (i.e., L) and the normalized average waiting time (i.e., $W\mu$) for the *M*/*M*/1 queue as a function of traffic intensity, ρ, has been graphically shown in Fig. 4. Note that the average waiting time and the queue length explode as traffic intensity approaches 1. Therefore, the *M*/*M*/1 queue is stable only if $0 \leq \rho < 1$.

Other Elementary Queueing Systems

There are a number of other single-queue models whose steady-state behavior can be determined via birth-and-death process techniques. We briefly mention the most important ones and refer the reader to standard texts on queueing theory (1–6) for detailed analysis and the associated mathematical expressions. Lack of space prevents us from listing all the associated results and formulas in these areas. The reader is referred to Ref. 3 (pp. 400–409) for a tabular listing of all the key formulas related to important queueing models.

***M*/*M*/1/*K*.** The *M*/*M*/1 model is somewhat unrealistic in the sense that, for example, no communication link can have an unlimited number of buffers. The *M*/*M*/1/*K* system is a more accurate model of this type of system in which a limit of K customers is allowed in the system. When the system contains K customers, arriving customers are turned away. This model can easily be analyzed by truncating the birth-and-death state diagram of the *M*/*M*/1 queue to only K states. This results in a birth-and-death process with coefficients

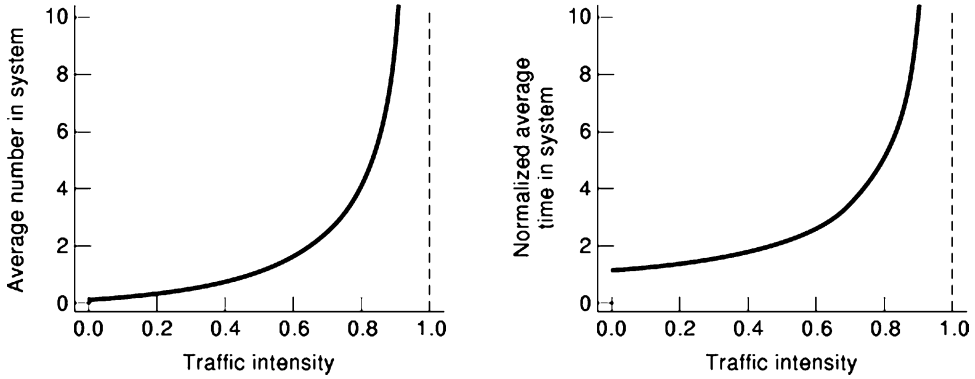


Figure 4. Performance characteristics of $M/M/1$ queue.

$$\lambda_n = \begin{cases} \lambda & \text{for } n = 0, 1, 2, 3, \ldots, K-1 \\ 0 & \text{for } n \geq K \end{cases} \tag{30}$$

and

$$\mu_n = \begin{cases} \mu & \text{for } n = 1, 2, 3, \ldots, K \\ 0 & \text{for } n \geq K \end{cases} \tag{31}$$

M/M/c. For this model we assume exponential interarrival times, exponential service times, and c identical servers. This system can be modeled as a birth-and-death process with the coefficients

$$\lambda_n = \lambda \quad \text{for } n = 0, 1, 2, \ldots \tag{32}$$

and

$$\mu_n = \begin{cases} n\mu & \text{for } n = 1, 2, 3, \ldots, c \\ c\mu & \text{for } n \geq c \end{cases} \tag{33}$$

Note that Eq. (33) agrees with Eq. (26) when $c = 1$; that is, for the $M/M/1$ queueing system, as it should. Historically, the expression of the probability that an arriving customer must wait is known as "Erlang's C Formula" or "Erlang's Delay Formula" (3, p. 404). Tables of values of Erlang's C Formula are often given in standard queueing texts; see, for example, Ref. 1 (pp. 320–323).

M/M/c/c. This system is sometimes called the "$M/M/c$ loss system" because customers who arrive when all the servers are busy are not allowed to wait for service and are lost. Each newly arriving customer is given his private server; however, if a customer arrives when all servers are occupied, that customer is lost; when modeling telephone calls, it is said that this is a system where blocked calls are cleared. The birth-and-death coefficients for this model are

$$\lambda_n = \begin{cases} \lambda & \text{for } n < c \\ 0 & \text{for } n \geq c \end{cases}$$

and

$$\mu_n = n\mu \quad \text{for } n = 1, 2, 3, \ldots, c$$

Historically, the expression for the probability that all servers are busy in an $M/M/c/c$ queueing system is referred to as "Erlang's B Formula" or "Erlang's Loss Formula" (3, p. 404). Tables of values of Erlang's B Formula are often given in standard queueing texts; see, for example, Ref. 1 (pp. 316–319).

***M/M/∞* Queueing System.** Mathematically speaking, an $M/M/\infty$ queueing system has an infinite number of servers which cannot be physically realized. $M/M/\infty$ queueing systems are used to model situations where a server is always immediately provided for each arriving customer. The coefficients of the associated birth-and-death process are given by

$$\lambda_n = \lambda \quad \text{for } n = 0, 1, 2, 3, \ldots \tag{34}$$

and

$$\mu_n = n\mu \quad \text{for } n = 1, 2, 3, \ldots \tag{35}$$

Solving the birth-and-death equations for the steady-state probability of number of customers in the queue results in

$$P_n = \frac{(\lambda/\mu)^n}{n!} e^{-\lambda/\mu} \quad \text{for } n = 0, 1, 2, 3, \ldots$$

Therefore, the number of customers in an $M/M/\infty$ queue is distributed according to a Poisson distribution with parameter λ/μ. The average number of customers in the system is simply $L = \lambda/\mu$ and the average waiting time is $W = 1/\mu$. This answer is obvious since if we provide each arriving customer his own server, then his time in the system is equal to his service time. $M/M/\infty$ models can be used to estimate the number of lines in use in large communications networks or as an estimate of values of $M/M/c$ or $M/M/c/c$ systems for large values of c.

***M/M/1/K/K* and *M/M/c/K/K* Queueing Systems.** These queueing systems, with a limited source model in which there are only K customers, is usually referred to as the

"machine repair model" or "machine interference model." One way to interpret these models is to assume that there is a collection of K machines, each of which has an up time which is exponentially distributed. The operating machines are outside of the system and enter the system only when they break down and thus need repair. The one repairman (or c repairmen) repairs the machines at an exponential rate. The coefficients of the associated birth-and-death process are

$$\lambda_n = \lambda(K - n) \quad \text{for } n = 0, 1, 2, \ldots, K - 1 \tag{36}$$

and

$$\mu_n = \mu \quad \text{for } n = 1, 2, \ldots, K \tag{37}$$

REFERENCES TO MORE ADVANCED TOPICS

The discussion of previous sections has been limited to some of the more elementary, but important, queueing models. However, the queueing theory literature currently contains a vast amount of results dealing with much more advanced and sophisticated queueing systems whose discussions are outside of the scope of this introductory article. The purpose of this section is to inform the reader of the existence of such advanced and complex models and to refer the interested reader to appropriate sources for further investigation.

Imbedded Markov Chain Queueing Models

Our discussion of queueing models in the previous section was limited to those whose probabilistic characterization could be captured by birth-and-death processes. When one ventures beyond the birth-and-death models into the more general Markov processes, then the type of solution methods used previously no longer apply. In the preceding sections we dealt mainly with queues with Poisson arrivals and exponential service times. These assumptions imply that the future evolution of the system will depend only on the present state of the system and not on the past history. In these systems, the state of the system was always defined as the number of customers in the system.

Consider the situation in which we like to study a queueing system for which the knowledge of the number of customers in the system is not sufficient to fully characterize its behavior. For example, consider a $D/M/1$ queue in which the service times are exponentially distributed, but the customer interarrival times are a constant. Then the future evolution of the system from some time t would depend not only on the number of customers in the system at time t, but also on the elapsed time since the last customer arrival. This is so because the arrival epoch of the next customer in a $D/M/1$ queue is fully determined by the arrival time of the last customer. A different and powerful method for the analysis of certain queueing models, such as the one mentioned above, is referred to as the "imbedded Markov chain" which was introduced by Kendall (17). The reader is referred to Refs. 1–6 for detailed discussion of imbedded Markov chain techniques and its application for analyzing such queueing systems as $M/G/1$, $GI/M/c$, $M/D/c$, $E_k/M/c$.

Queueing Systems with Priority

Queueing models with priority are those where the queue discipline is based on a priority mechanism where the order in which the waiting customers are selected for service is dependent on their assigned priorities. Many real queueing systems fit these priority-discipline models. Rush jobs are taken ahead of other jobs, important customers may be given precedence over others, and data units containing voice and video signals may be given higher priority over data units containing no real-time information in a packet switched computer communication network. Therefore, the use of queueing models with priority often provides much needed insight into such situations. The inclusion of priority makes the mathematical analysis of models much more complicated. There are many ways in which notions of priority can be integrated into queueing models. The most popular ones were defined earlier in this article under queue disciplines. They include such priority disciplines as non-preemptive priority, preemptive resume priority, and preemptive repeat priority (21).

Networks of Queues

Many queueing systems encountered in practice are queueing networks consisting of a collection of service facilities where customers are routed from one service center to another, and they receive service at some or all of these service facilities. In such systems, it is necessary to study the entire network in order to obtain information about the performance of a particular queue in the network. Such models have become very important because of their applicability to modeling computer communication networks. This is a current area of great research and application interest with many difficult problems. Networks of queues can be described as a group of nodes (say n of them) where each node represents a service center each with c_i servers, where $i = 1, 2, \ldots, n$. In the most general case, customers may arrive from outside the system to any node and may depart the system from any node. The customers entering the system traverse the network by being routed from node to node after being served at each node they visit. Not all customers enter and leave from the same nodes, or take the same path through the network. Customers may return to nodes previously visited, skip some nodes, or choose to remain in the system forever. Analytical results on queueing networks have been limited because of the difficulty of the problem. Most of the work has been confined to cases with a Poisson input and exponential service times and probabilistic routing between the nodes. The reader is referred to Ref. 22 for a complete treatment of network of queues.

Simulation of Queueing Systems

Very often, analytical solutions to many practical queueing models are not possible. This is often due to many factors such as the complexity of the system architecture, the nature of the queue discipline, and the stochastic characteristics of the input arrival streams and service times. For example, it would be impractical to develop analytical solutions to a multinode multiserver system where the customers are allowed to recycle through the system, the ser-

vice times are distributed according to truncated Gaussian distribution, and each node has its own complex queueing discipline. For analytically intractable models, it may be necessary to resort to analysis by simulation. Another area that simulation could be used for is those models in which analytical results are only available for steady state and one needs to study the transient behavior of the system.

Generally speaking, simulation refers to the process of using computers to imitate the operation of various kinds of real-world systems or processes. While simulation may offer a mechanism for studying the performance of many analytically intractable models, it is not without its disadvantages. For example, since simulation can be considered analysis by experimentation, one has all the usual problems associated with running experiments in order to make inferences concerning the real world, and one must be concerned with such things as run length, number of replications, and statistical significance. Although simulation can be a powerful tool, it is neither cheap nor easy to apply correctly and efficiently. In practice, there seems to be a strong tendency to resort to simulation from the outset. The basic concept is easy to understand, it is relatively easy to justify to management, and many powerful simulation tools are readily available. However, an inexperienced analyst will usually seriously underestimate the cost of many resources required for an accurate and efficient simulation study.

Viewing it from a high level, a simulation model program consists of three phases. The data generation phase involves the production of representative interarrival times and service times where needed throughout the queueing system. This is normally achieved by employing one of the many random number generation schemes. The so-called bookkeeping phase of a simulation program deals with (a) keeping track of and updating the state of the system whenever a new event (such as arrival or departure) occurs and (b) monitoring and recording quantities of interest such as various performance measures. The final phase of a simulation study is normally the analysis of the output of the simulation run via appropriate statistical methods. The reader is referred to Refs. 23 and 24 for a comprehensive look at simulation techniques.

Cyclic Queueing Models

This area deals with situations where a collection of queues is served by a single server. The server visits each queue according to some predetermined (or random) order and serves each queue visited for a certain amount of time (or certain number of customers) before traversing to the next queue. Other terms used to refer to this area of queueing theory are "round-robin queueing" or "queueing with vacations." As an example, a time-shared computer system where the users access the central processing unit through terminals can be modeled as a cyclic queue. The reader is referred to Ref. 25 and Section 5.13 of Ref. 1 for detailed discussion of cyclic queues.

Control of Queues

This area of queueing theory deals with optimization techniques used to control the stochastic behavior and to optimize certain performance measures of a queueing systems. Examples of practical questions that deal with this area of queueing theory include the following (22, Chap. 8): When confronted with the choice of joining one waiting line among many (such a supermarket checkout counter or highway tool booths), how does one choose the "best" queue? Should a bank direct the customers to form a single waiting line, or should each bank teller have his or her own queue? Should a congested link in a communication network be replaced with another link twice as fast, or should it be augmented with a second identical link working in parallel with the first one?

BIBLIOGRAPHY

1. R. B. Cooper *Introduction to Queueing Theory*, 2nd ed., New York: Elsevier/North-Holland, 1981.
2. D. Gross C. H. Harris *Fundamentals of Queueing Theory*, New York: Wiley, 1985.
3. L. Kleinrock *Queueing Systems, Volume I: Theory*, New York: Wiley-Interscience, 1975.
4. L. Kleinrock *Queueing Systems, Volume II: Computer Applications*, New York: Wiley-Interscience, 1976.
5. E. Gelenbe G. Pujolle *Introduction to Queueing Networks*, Paris: Wiley, 1987.
6. T. L. Saaty *Elements of Queueing Theory*, New York: McGraw-Hill, 1961.
7. R. B. Cooper Queueing theory. In D. P. Heyman and M. J. Sobel (eds.), *Stochastic Models*, Handbooks of Operations Research and Management Science, Vol. 2, New York: North-Holland, 1990.
8. N. U. Prabhu A bibliography of books and survey papers on queueing systems: theory and applications, *Queueing Systems*, **1**: 1–4, 1987.
9. M. A. Leibowitz Queues, *Sci. Am.*, **219** (2): 96–103, 1968.
10. A. K. Erlang The theory of probabilities and telephone conversations, *Nyt Tidsskrift Matematik*, Series B, **20**: 33–39, 1909.
11. F. S. Hillier G. J. Lieberman *Introduction to Operations Research*, 4th ed., Oakland, CA: Holden-Day, 1986.
12. H. A. Taha *Operations Research: An Introduction*, New York: Macmillan, 1971.
13. E. Gelenbe I. Mitrani *Analysis and Synthesis of Computer Systems*, New York: Academic Press, 1980.
14. J. F. Hayes *Modeling and Analysis of Computer Communications Networks*, New York: Plenum Press, 1984.
15. C. H. Sauer K. M. Chandy *Computer Systems Performance Modeling*, Englewood Cliffs, NJ: Prentice-Hall, 1981.
16. J. N. Daigle *Queueing Theory for Telecommunications*, Reading, MA: Addison-Wesley, 1992.
17. D. G. Kendall Stochastic processes occurring in the theory of queues and their analysis by the method of imbedded markov chains, *Ann. Math. Stat.*, **24**: 338–354, 1953.
18. A. M. Lee *Applied Queueing Theory*, London: Macmillan, 1966.
19. A. Leon-Garcia *Probability and Random Processes for Electrical Engineering*, Reading, MA: Addison-Wesley, 1989.
20. J. D. C. Little A proof for the queueing formula $L = \lambda$ W, *Oper. Res.*, **9**: 383–387, 1961.
21. N. K. Jaiswell *Priority Queues*, New York: Academic Press, 1968.

22. J. Walrand *An Introduction to Queueing Networks*, Englewood Cliffs, NJ: Prentice-Hall, 1988.
23. P. Bratley B. L. Fox L. E. Schrage *Guide to Simulation*, New York: Springer-Verlag, 1983.
24. A. M. Law W. D. Kelton *Simulation Modeling and Analysis*, 2nd ed., New York: McGraw-Hill, 1991.
25. H. Takagi *Analysis of Polling Systems*, Cambridge, MA: MIT Press, 1986.
26. "Queueing Theory," http://en.wikipedia.org/wiki/Queuing_theory.
27. Myron Hlynka,"What is the proper spelling — queueing or queuing?," http://www2.uwindsor.ca/ hlynka/qfaq.html.
28. Jeff Miller,"A Collection of Word Oddities and Trivia," http://members.aol.com/gulfhigh2/words6.html.
29. Myron Klynka,"Myron Hlynka's Queueing Theory Page," http://www2.uwindsor.ca/ hlynka/queue.html.
30. John N. Daigle," Queueing Theory with Application to Packet Telecommunications," Springer, 2004.
31. N. U. Prabhu," Foundations of Queueing Theory," Springer, 1997.

NADER MEHRAVARI
Lockheed Martin, Owego, NY

} { { } }

Signal Detection Theory
Standard Article
Brian L. Hughes[1]
[1]North Carolina State University, Raleigh, NC

DOI: 10.1002/047134608X.W4214
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (152K)

Abstract

The sections in this article are

Basic Principles

Detection of Known Signals in Noise

Detection of Signals with Unknown Parameters

Detection of Random Signals

Advanced Topics

Further Reading

SIGNAL DETECTION THEORY

In remote sensing and communications, we are often required to decide whether a particular signal is present—or to distinguish among several possible signals—on the basis of noisy observations. For example, a radar transmits a known electromagnetic signal pulse into space and detects the presence of targets (e.g., airplanes or missiles) by the echoes which they reflect. In digital communications, a transmitter sends data symbols to a distant receiver by representing each symbol by a distinct signal. In automatic speech recognition, an electrical microphone signal is processed in order to extract a

sequence of phonemes, the elementary sounds that make up spoken language. Similar problems arise in sonar, image processing, medical signal processing, automatic target recognition, radio astronomy, seismology, and many other applications.

In each of these applications, the signal received is typically distorted and corrupted by spurious interference, which complicates the task of deciding whether the desired signal is present. In particular, the received signal in radar and communication systems inevitably contains random fluctuations, or *noise,* due to the thermal motion of electrons in the receiver and ions in the atmosphere, atmospheric disturbances, electromagnetic clutter, and other sources. This noise component is inherently unpredictable; the best we can do is to describe it statistically in terms of probability distributions. In some situations, the desired signal may also be distorted in unpredictable ways—by unknown time delays, constructive and destructive interference of multiple signal reflections, and other channel impairments—and may be best modeled as a random process.

The task of the receiver is to decide whether the desired signal is present in an observation corrupted by random noise and other distortions. The mathematical framework for dealing with such problems comes from the field of *hypothesis testing,* a branch of the theory of statistical inference. In engineering circles, this is also called *detection theory* because of its early application to radar problems. Detection theory provides the basis for the design of receivers in communication and radar applications, algorithms for identifying edges and other features in images, algorithms for parsing an electrical speech signal into words, and many other applications.

In addition to deciding whether a signal is present, we often want to estimate real-valued parameters associated with the signal, such as amplitude, frequency, phase, or relative time delay. For example, once a target has been detected, a radar will typically attempt to determine its range by estimating the round-trip propagation delay of the pulse echo. Problems of this type are the province of *estimation theory,* a field of statistical inference closely related to detection theory. In essence, detection theory deals with the problem of deciding among a finite number of alternatives, whereas estimation theory seeks to approximate real-valued signal parameters.

This article provides an overview of the basic principles and selected results of signal detection theory. The subject of estimation theory is treated elsewhere in this volume (see ESTIMATION THEORY). A more complete and detailed treatment of both topics can be found in Refs. 1 and 2. In the next section, we introduce some fundamental concepts that underlie the design of optimal detection procedures. In succeeding sections, we apply these concepts to the problem of detecting signals in additive Gaussian noise. Finally, we close with a discussion of selected advanced topics in detection theory.

BASIC PRINCIPLES

Simple Hypotheses

The design of optimal receivers in radar and communications is based on principles from the theory of statistical hypothesis testing. The fundamental problem of hypothesis testing is to decide which of several possible statistical models best describes an observation Y. For simplicity, consider the problem of deciding between two models, "target present" or "target absent." Suppose the probability density function (pdf) of Y is given by $p_1(y)$ when the target is present and by $p_0(y)$ when the target is absent. The problem of deciding whether Y is best modeled by $p_0(y)$ or $p_1(y)$ can be expressed as a choice between two hypotheses:

$$\begin{aligned} H_0 &: \ Y \text{ has pdf } p_0(y) \qquad \text{(target absent)} \\ H_1 &: \ Y \text{ has pdf } p_1(y) \qquad \text{(target present)} \end{aligned} \tag{1}$$

where H_0 is often called the *null hypothesis* and H_1 is the *alternative hypothesis.* A *detector* (or *decision rule* or *hypothesis test*) is a procedure for deciding which hypothesis is true on the basis of the observation Y. More precisely, a detector is a function that assigns to each possible observation $Y = y$ a decision $d(y) = H_0$ or H_1. There are two possible ways for the detector to make an error: It may conclude that a target is present when there is none (a *false alarm*), or it may decide that no target is present when in fact there is one (a *miss*). The performance of a detector d can therefore be measured by two quantities, the *probability of false alarm* $P_F(d)$ and the *probability of a miss* $P_M(d)$. Ideally, we would like to make both error measures as small as possible; however, these are usually conflicting objectives in the sense that reducing one often increases the other. In order to determine which detector is best for a particular application, we must strike a balance between $P_F(d)$ and $P_M(d)$ which reflects the relative importance of these two types of errors.

Several methods can be used to weigh the relative importance of $P_F(d)$ and $P_M(d)$. If the prior probabilities of the hypotheses are known, say $\pi = \Pr\{H_0\} = 1 - \Pr\{H_1\}$, it is natural to seek a *minimum-probability-of-error detector*—that is, one that minimizes the average error probability:

$$\pi P_F(d) + (1 - \pi) P_M(d)$$

Such detectors are appropriate in digital communication receivers where the hypotheses represent the possible transmitted data symbols and where the goal is to minimize the average number of errors that occur in a series of transmissions. More generally, when the two kinds of errors are not equally serious, we can assign a cost C_{ij} to choosing hypothesis H_i when H_j is actually true ($i, j = 0, 1$). A detector that minimizes the *average cost* (or *risk*) is called a *Bayes detector.* It sometimes happens that the prior probabilities of H_0 and H_1 are not known, in which case the Bayes and minimum-probability-of-error detectors cannot be applied. In this case, it often makes sense to choose a detector that minimizes the average cost for the worst prior probability—for example, one that minimizes

$$\max_{0 \le \pi \le 1} \pi P_F(d) + (1 - \pi) P_M(d) = \max\{P_F(d), P_M(d)\}$$

The resulting detector is called a *minimax detector.* Finally, in other circumstances it may be difficult to assign costs or prior probabilities. In radar, for example, what is the prior probability of an incoming missile, and what numerical cost is incurred by failing to detect it? In situations like this, it seems inappropriate to weigh the relative importance of false alarms and misses in terms of numerical costs. An alternative approach is to seek a detector that makes the probability of

a miss as small as possible for a given probability of false alarm:

$$\text{minimize } P_{\text{M}}(d) \quad \text{subject to} \quad P_{\text{F}}(d) \leq \alpha$$

A detector of this type is called a *Neyman–Pearson detector.*

Remarkably, all of the optimal detectors mentioned above take the same general form. Each involves computing a *likelihood ratio* from the received observation

$$\Lambda(y) = \frac{p_1(y)}{p_0(y)} \tag{2}$$

and comparing it with a threshold τ. When Y is observed, the detectors choose H_1 if $\Lambda(Y) > \tau$ and choose H_0 if $\Lambda(Y) < \tau$. This detector can be expressed concisely as

$$\Lambda(Y) \underset{H_0}{\overset{H_1}{\gtrless}} \tau \tag{3}$$

When $\Lambda(Y) = \tau$, minimax and Neyman–Pearson detectors may involve a random decision, such as choosing H_0 or H_1 based on the toss of a biased coin. The minimum-probability-of-error, Bayes, minimax, and Neyman–Pearson detectors mentioned earlier differ only in their choice of threshold τ and behavior on the boundary $\Lambda(Y) = \tau$.

Composite Hypotheses

Thus far we have assumed that the probability distribution of Y is known perfectly under both hypotheses. It is very common, however, for a signal to depend on parameters that are not known precisely at the detector. In radar, for example, since the distance to the target is not known at the outset, the radar pulse will experience an unknown propagation delay as it travels to the target and back. In digital communications, the phase of the carrier signal is often unknown to the receiver. In such situations, the hypothesis "target present" corresponds to a collection of possible probability distributions, rather than one. A hypothesis of this type is called a *composite hypothesis,* in contrast to a *simple hypothesis* in which Y is described by a single pdf.

Let θ denote an unknown parameter associated with the observation, and let $p_0(y|\theta)$ and $p_1(y|\theta)$ denote the conditional probability densities of Y given θ under H_0 and H_1, respectively. In some cases, it may be appropriate to model θ as a random variable with known probability densities $q_0(\theta)$ and $q_1(\theta)$ under hypothesis H_0 and H_1, respectively. In such cases, the composite hypothesis testing problem is equivalent to a simple hypothesis testing problem with probability densities

$$p_0(y) = \int_\theta p_0(y|\theta) q_0(\theta)\, d\theta, \qquad p_1(y) = \int_\theta p_1(y|\theta) q_1(\theta)\, d\theta \tag{4}$$

and the optimal detectors are again of the form shown in Eq. (3). If θ is a random variable with unknown probability densities under H_0 and H_1, we can follow a minimax-type approach and look for the detector that minimizes the worst-case average cost over all probability densities $q_0(\theta)$ and $q_1(\theta)$.

When θ cannot be modeled as a random variable, the situation is more complex. Occasionally, there exists a detector that is simultaneously optimal for all θ, in the sense that it minimizes $P_{\text{M}}(d|\theta)$ for each θ over all detectors with a given false-alarm probability, $\max_\theta P_{\text{F}}(d|\theta) \leq \alpha$. A detector with this property is said to be *uniformly most powerful.* When a uniformly most powerful detector does not exist, it is natural to use an estimate $\hat{\theta}$ of the unknown parameter derived from the observation $Y = y$. The most commonly used estimates are *maximum likelihood estimates,* which are defined as the value of θ that maximizes the conditional probability density of the observation:

$$p_i(y|\hat{\theta}_i) = \max_\theta p_i(y|\theta), \qquad i = 0,\ 1$$

Substituting the maximum likelihood estimates $\hat{\theta}_0$ and $\hat{\theta}_1$ into the likelihood ratio, we obtain the *generalized likelihood ratio* (GLR):

$$\Lambda_{\text{G}}(y) = \frac{\max_\theta p_1(y|\theta)}{\max_\theta p_0(y|\theta)}$$

Detectors based on the GLR take the same form as the likelihood ratio detector [Eq. (3)], with $\Lambda_{\text{G}}(y)$ substituted for $\Lambda(y)$.

Multiple Hypotheses and Observations

Each of the detectors described above extends in a straightforward way to a sequence of observations $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$. In this case, the hypotheses become

$$H_0: \ \boldsymbol{Y} \text{ has pdf } p_0(y_1, \ldots, y_n) \qquad \text{(target absent)}$$
$$H_1: \ \boldsymbol{Y} \text{ has pdf } p_1(y_1, \ldots, y_n) \qquad \text{(target present)}$$

Again, the minimum-probability-of-error, Bayes, minimax, and Neyman–Pearson detectors are of the form shown in Eq. (3), where the likelihood ratio is

$$\Lambda(y_1, \ldots, y_n) = \frac{p_1(y_1, \ldots, y_n)}{p_0(y_1, \ldots, y_n)}$$

The generalized likelihood ratio detector also extends in an analogous way.

We have so far considered only detection problems involving two hypotheses. In some situations there may be more than two possible models for the observed data. For example, digital communication systems often use nonbinary signaling techniques in which one of M possible symbols is transmitted to the receiver in each unit of time. The receiver then has M hypotheses from which to choose, one corresponding to each possible transmitted symbol. The hypothesis-testing problem can then be expressed as

$$H_i: Y \text{ has pdf } p_i(y), \qquad i = 0, \ldots, M-1$$

In such situations, we are usually interested in finding a minimum-probability-of-error detector for some given prior probabilities $\pi_i = \Pr\{H_i\}$, $i = 0, \ldots, M-1$. The average probability of error for a detector d is given by

$$\sum_{i=0}^{M-1} \Pr\{d(Y) \neq H_i | H_i \text{ is true}\} \pi_i \tag{5}$$

This error probability is minimized by the *maximum a posteriori probability* (MAP) detector, which chooses the hypothesis

that is most probable given the observation $Y = y$. Mathematically, the MAP detector takes the form

$$d(y) = H_i \text{ that maximizes } q(H_i|y) \tag{6}$$

where

$$q(H_i|y) = \frac{p_i(y)\pi_i}{p(y)}, \qquad p(y) = \sum_i p_i(y)\pi_i$$

is the conditional probability of hypothesis H_i given the observation $Y = y$. In digital communications, the possible transmitted symbols are often equally likely ($\pi_i = 1/M$, $i = 1, \ldots, M - 1$), in which case the MAP detector reduces to the *maximum likelihood* (ML) detector

$$d(y) = H_i, \qquad \text{where } i \text{ maximizes } p_i(y) \tag{7}$$

It is easy to check that the MAP and ML detectors reduce to likelihood ratio detectors when $M = 2$.

The results presented in this section form the basis for the design of optimal receivers for a wide variety of communications and remote sensing problems. In the following sections, we apply these results to the problem of detecting signals in noise. In the process, we obtain several of the most important and widely used receivers in communications and radar as particular instances of the likelihood ratio detector [Eq. (3)].

DETECTION OF KNOWN SIGNALS IN NOISE

We now consider the problem of detecting the presence or absence of a discrete-time signal observed in noise. A detector for this purpose is also called a *receiver* in the terminology of radar and communications. We assume for now that both the signal and the noise statistics are known precisely at the receiver, in which case the detection problem can be expressed as a choice between the simple hypotheses:

$$\begin{aligned} H_0: &\quad Y_i = N_i, && i = 1, \ldots, n \\ H_1: &\quad Y_i = s_i + N_i, && i = 1, \ldots, n \end{aligned}$$

where s_i, $i = 1, \ldots, n$ is a deterministic signal and N_i, $i = 1, \ldots, n$ is a *white Gaussian noise* sequence—that is, a sequence of independent and identically distributed (i.i.d.) Gaussian random variables with mean zero and variance $\sigma^2 > 0$. Thus, the probability densities of $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ under both hypotheses are multivariate Gaussian where

$$p_1(\boldsymbol{y}) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - s_i)^2\right\} \tag{8}$$

and $p_0(\boldsymbol{y})$ is given by the same formula, with the s_i's set to zero.

From the previous section, we know that each of the optimal detectors (minimum probability of error, Bayes, minimax, Neyman–Pearson) reduces to a likelihood ratio detector [Eq. (3)]. From Eq. (8), the likelihood ratio for this problem takes the form

$$\Lambda(\boldsymbol{y}) = \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}[(y_i - s_i)^2 - y_i^2]\right\}$$

It is easy to verify that $\Lambda(\boldsymbol{y})$ is a monotonically increasing function of the test statistic

$$\sum_{i=1}^{n} y_i s_i - \frac{1}{2}\sum_{i=1}^{n} s_i^2 \tag{9}$$

Thus, the likelihood ratio detector [Eq. (3)] can be expressed in the equivalent form

$$\sum_{i=1}^{n} Y_i S_i \underset{H_0}{\overset{H_1}{\gtrless}} \tau' \tag{10}$$

where the quadratic term in Eq. (9) has been merged with the threshold τ'. The optimal receiver thus consists of correlating the received sequence against the desired signal and comparing the result to a threshold. A receiver of this type is called a *correlation receiver*.

This receiver extends in a natural way to continuous-time detection. The correlation receiver extends in a natural way to continuous-time detection problems. A proof of this extension is nontrivial and requires generalizing the likelihood ratio to continuous-time observations (see Chapter 6 of Ref. 1 for details). Consider the signal detection problem

$$\begin{aligned} H_0: &\quad Y(t) = N(t), && 0 \le t < T \\ H_1: &\quad Y(t) = s(t) + N(t), && 0 \le t < T \end{aligned}$$

where $s(t)$ is a known deterministic signal and $N(t)$ is a *continuous-time white Gaussian noise* process with two-sided power spectral density $N_0/2$ (see KALMAN FILTERS AND OBSERVERS). The likelihood ratio is again a monotonically increasing function of a correlation statistic

$$\int_0^T y(t)s(t)\,dt - \frac{1}{2}\int_0^T s^2(t)\,dt \tag{11}$$

Merging the second term with the threshold, we again find that the likelihood ratio detector is a correlation receiver, which is illustrated in Fig. 1.

The correlation in Fig. 1 can also be expressed as a filtering operation:

$$\int_0^T y(t)s(t)\,dt = \int_{-\infty}^{\infty} h(T - t)y(t)\,dt$$

where $h(t) = s(T - t)$, $0 \le t \le T$. Here $h(t)$ can be regarded as the impulse response of a linear time-invariant filter. The frequency response of this filter is given by the Fourier transform of $h(t)$:

$$H(f) = S^*(f)e^{-2\pi jfT}$$

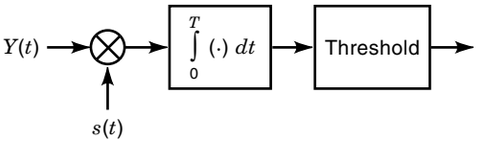


Figure 1. Correlation receiver.

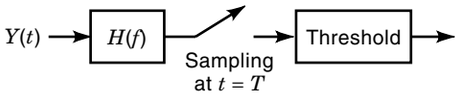


Figure 2. Matched filter receiver.

where $S^*(f)$ is the complex conjugate of the Fourier transform of $s(t)$. The correlation receiver can therefore be implemented in the form of a filter sampled at time $t = T$, as illustrated in Fig. 2.

Since the amplitude of the filter $H(f)$ matches the signal spectrum $S(f)$, this form of the detector is called a *matched-filter receiver*. The matched filter has the property that it maximizes the *signal-to-noise ratio* (the ratio of signal power to noise power) at the input to the threshold operation (see Ref. 2).

The receiver in Fig. 1 is optimal for deciding whether a known signal is present or absent in white Gaussian noise. Very similar correlation structures appear in receivers for deciding among several possible signals. For a detection problem involving M possible signals, the minimum-probability-of-error detector will compare the outputs of a bank of M correlation branches, one for each possible signal. For example, consider the problem of deciding between two equally likely ($\pi_0 = \pi_1 = \frac{1}{2}$) signals in white Gaussian noise:

$$\begin{aligned} H_0: &\quad Y(t) = s_0(t) + N(t), \qquad 0 \le t < T \\ H_1: &\quad Y(t) = s_1(t) + N(t), \qquad 0 \le t < T \end{aligned}$$

The receiver that minimizes the average probability of error [Eq. (5)] in this case is the maximum likelihood detector. When $Y(t) = y(t)$ is received, the ML detector chooses the hypothesis H_i such that $s(t) = s_i(t)$ maximizes the correlation statistic [Eq. (11)]. Thus, the optimal receiver consists of a correlation receiver with a branch for each possible transmitted signal, as illustrated in Fig. 3 [where E_i is the energy of $s_i(t)$].

As in the case of one signal, the correlation receiver in Fig. 3 can be implemented in the alternative form of a bank of matched filters, each sampled at $t = T$.

DETECTION OF SIGNALS WITH UNKNOWN PARAMETERS

In the preceding section, we assumed that the desired signal is known precisely at the receiver. However, this assumption is often unrealistic in practice. Unknown path losses, Doppler shifts, and propagation delays can lead to uncertainty about the amplitude, phase, frequency, and delay of the signal. When signal parameters are unknown, the detection problem involves composite hypotheses. As discussed earlier, detection procedures for composite-hypothesis testing depend on whether the unknown parameter is modeled as random or deterministic. In this section, we consider only the example of an unknown random parameter.

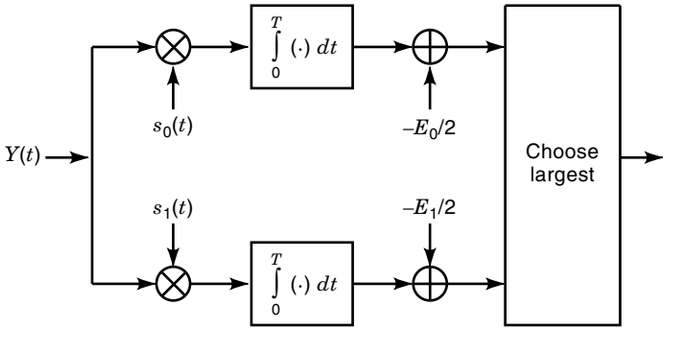


Figure 3. Correlation receiver for binary signals.

Many radar and communication problems involve detection of a sinusoidal signal with an unknown phase. The phase is typically modeled as a random variable Θ, uniformly distributed on $[0, 2\pi)$. For example, consider the discrete-time binary detection problem:

$$\begin{aligned} H_0: &\quad Y_i = N_i, \qquad i = 1, \cdots, n \\ H_1: &\quad Y_i = A\cos(\omega i T_s + \Theta) + N_i, \qquad i = 1, \ldots, n \end{aligned}$$

where A is a known constant, T_s is the sampling interval, ω is a frequency such that $n\omega T_s$ is an integer multiple of 2π, and N_i is a discrete-time white Gaussian noise sequence which is independent of Θ. The likelihood ratio for this detection problem is given by Eqs. (3) and (4). Given $\Theta = \theta$, the conditional probability density of $\boldsymbol{Y}$ under H_1 is

$$p_1(\boldsymbol{y}|\theta) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}[y_i - A\cos(\omega i T_s + \theta)]^2\right\}$$

and the unconditional pdf is given by

$$p_1(\boldsymbol{y}) = \frac{1}{2\pi}\int_0^{2\pi} p_1(\boldsymbol{y}|\theta)\,d\theta$$

After some manipulation, the likelihood ratio reduces to

$$\Lambda(\boldsymbol{y}) = \frac{p_1(\boldsymbol{y})}{p_0(\boldsymbol{y})} = \exp\left\{-\frac{A^2 n}{4\sigma^2}\right\} I_0\left(\frac{Aq}{\sigma^2}\right)$$

where

$$q^2 = \left[\sum_{i=1}^{n} y_i \cos(\omega i T_s)\right]^2 + \left[\sum_{i=1}^{n} y_i \sin(\omega i T_s)\right]^2 \qquad (12)$$

and

$$I_0(x) = \frac{1}{2\pi}\int_0^{2\pi} \exp\{x\cos\theta\}\,d\theta$$

is a modified Bessel function of the first kind. Since $I_0(x)$ is symmetric in x and monotonically increasing for $x \ge 0$, the likelihood ratio is an increasing function of the quadrature statistic [Eq. (12)], and the likelihood ratio detector [Eq. (3)] can be expressed in the alternate form:

$$q^2 \underset{H_0}{\overset{H_1}{\gtrless}} \tau'$$

This detector is called a *quadrature receiver*. It consists of correlating the received signal with two phase-shifted versions of the desired signal, $\cos(\omega i T_s)$ and $\sin(\omega i T_s)$. The two correla-

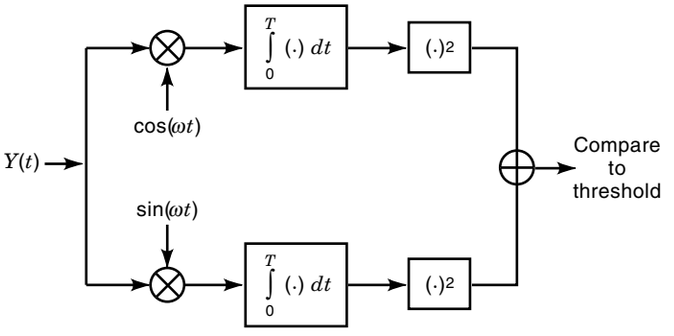


Figure 4. Quadrature receiver.

tions are then squared, summed, and compared to a threshold.

This detector extends in a straightforward way to the detection of continuous-time sinusoidal signals with random phase. Consider the detection problem

$$\begin{aligned} H_0: &\quad Y(t) = N(t), && 0 \le t < T \\ H_1: &\quad Y(t) = A\cos(\omega t + \Theta) + N(t), && 0 \le t < T \end{aligned}$$

where Θ is a random phase uniformly distributed on $[0, 2\pi)$, A is a constant, ω is an integer multiple of $2\pi/T$, and $N(t)$ is white Gaussian noise. The likelihood ratio detector reduces to a threshold test involving the quadrature statistic:

$$\left[\int_0^T y(t)\cos(\omega t)\,dt\right]^2 + \left[\int_0^T y(t)\sin(\omega t)\,dt\right]^2$$

The resulting continuous-time quadrature receiver is illustrated in Fig. 4.

DETECTION OF RANDOM SIGNALS

So far we have assumed the receiver knows the desired signal exactly, with the possible exception of specific parameters such as amplitude, phase, or frequency. However, sometimes the received signal may be so distorted by the channel that it must be modeled by a more complex type of random process. In certain situations, for example, the transmitted signal propagates to the receiver by many different paths due to signal reflection and scattering. In such cases, the received signal consists of many weak replicas of the original signal, called *multipath signals,* with different amplitudes and relative time delays. The superposition of these multipath signals can resemble a Gaussian random process statistically. Typical examples include channels that use ionospheric reflection or tropospheric scattering as a primary mode of propagation, and land mobile radio, where scattering and reflection by nearby ground structures can produce a similar effect.

In this section, we consider the problem of detecting signals that are described by random processes. We again begin by considering a discrete-time detection problem:

$$\begin{aligned} H_0: &\quad Y_i = N_i, && i = 1, \ldots, n \\ H_1: &\quad Y_i = S_i + N_i, && i = 1, \ldots, n \end{aligned}$$

where $\boldsymbol{S} = (S_1, \ldots, S_n)$ is a zero-mean Gaussian random sequence with known covariance $\boldsymbol{E}\{SS^{\mathrm{T}}\} = \Sigma$, N_i is discrete-time white Gaussian noise, $\boldsymbol{E}\{\cdot\}$ denotes the expectation, and T denotes transpose. Note that $\boldsymbol{Y}$ is a zero-mean Gaussian vector under hypotheses H_0 and H_1, with respective covariances $\sigma^2\boldsymbol{I}$ and $\sigma^2\boldsymbol{I} + \Sigma$. The likelihood ratio is then

$$\Lambda(\boldsymbol{y}) = |\boldsymbol{I} + \sigma^{-2}\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}\boldsymbol{y}^T(\sigma^2\boldsymbol{I} + \Sigma)^{-1}\boldsymbol{y} + \frac{1}{2}\boldsymbol{y}^T(\sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}\right\}$$

Since this is a monotonically increasing function of the test statistic $\boldsymbol{y}^T\boldsymbol{Q}\boldsymbol{y}$, where

$$\boldsymbol{Q} = \boldsymbol{I} - (\boldsymbol{I} + \sigma^{-2}\Sigma)^{-1} = \Sigma(\sigma^2\boldsymbol{I} + \Sigma)^{-1} \tag{13}$$

the likelihood ratio detector [Eq. (3)] can be expressed as

$$\boldsymbol{Y}^T\boldsymbol{Q}\boldsymbol{Y} \underset{H_0}{\overset{H_1}{\gtrless}} \tau'$$

This detector is called a *quadratic receiver.* In the particular case when the desired signal is also a white noise process (i.e., $\Sigma = \alpha^2\boldsymbol{I}$), the quadratic receiver statistic $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{y}$ is proportional to $\|\boldsymbol{y}\|^2$ and the likelihood ratio detector reduces to

$$\|\boldsymbol{Y}\|^2 \underset{H_0}{\overset{H_1}{\gtrless}} \tau''$$

Since $\|\boldsymbol{y}\|^2$ is proportional to the average energy in the sequence $\boldsymbol{y}$, this detector is called an *energy detector* or *radiometer.*

In continuous time, the likelihood ratio detector takes a more complex but analogous form. Consider the problem of deciding among the hypotheses

$$\begin{aligned} H_0: &\quad Y(t) = N(t), && 0 \le t < T \\ H_1: &\quad Y(t) = S(t) + N(t), && 0 \le t < T \end{aligned}$$

where $S(t)$ is a zero-mean Gaussian noise process with autocovariance function

$$C(t, u) = E\{S(t)S(u)\}$$

and $N(t)$ is a white Gaussian noise process with one-sided power spectral density $N_0/2$. The likelihood ratio detector for this problem can also be expressed in terms of a quadratic statistic

$$\int_0^T \int_0^T Q(t, u)Y(t)Y(u)\,dt\,du$$

where $Q(t, u)$ is the solution to the integral equation

$$C(t, u) = \int_0^T Q(t, \xi)C(\xi, u)\,d\xi + \frac{N_0}{2}Q(t, u), \qquad 0 \le t, u < T$$

This equation is a continuous-time analog of Eq. (13), as can be seen by writing Eq. (13) in the alternative form $\Sigma = \boldsymbol{Q}\Sigma + \sigma^2\boldsymbol{Q}$.

ADVANCED TOPICS

Detection in Colored Gaussian Noise

In the preceding sections, we assumed white Gaussian noise models for both the discrete and continuous-time detection problems. When the noise is Gaussian but not white, we can transform the detection problem into an equivalent problem involving white noise. For example, suppose we are interested in detecting a known signal $\boldsymbol{s} = (s_1, \ldots, s_n)$,

$$H_0: \quad \boldsymbol{Y} = \boldsymbol{N}$$
$$H_1: \quad \boldsymbol{Y} = \boldsymbol{s} + \boldsymbol{N}$$

where $\boldsymbol{N}$ is a zero-mean Gaussian noise vector with positive-definite covariance matrix Σ_N. Using the Cholesky decomposition (see p. 84 of Ref. 1), the noise covariance can be written in the form

$$\Sigma_N = CC^{\mathrm{T}}$$

where C is an $n \times n$ nonsingular lower-triangular matrix. Since C is invertible, it is intuitive that no information is lost by taking the observation to be $\boldsymbol{Y}' = C^{-1}\boldsymbol{Y}$ instead of $\boldsymbol{Y}$. The detection problem can then be expressed as

$$H_0: \quad \boldsymbol{Y}' = \boldsymbol{N}'$$
$$H_1: \quad \boldsymbol{Y}' = \boldsymbol{s}' + \boldsymbol{N}'$$

where $\boldsymbol{s}' = C^{-1}\boldsymbol{s}$ and $\boldsymbol{N}' = C^{-1}\boldsymbol{N}$. It is easy to verify that $\boldsymbol{N}'$ is a white Gaussian noise vector with covariance $\Sigma_{N'} = \boldsymbol{I}$; thus, the likelihood ratio detector is the correlation receiver [Eq. (10)]. Here, the overall approach is to *prewhiten* the original detection problem, by transforming it to an equivalent problem involving white noise. After prewhitening, the detection problem can be solved by the methods described in the previous sections.

A similar prewhitening procedure can be performed for continuous-time detection problems. Let $N(t)$ be a zero-mean colored Gaussian noise process with known autocovariance function $R(t, u) = E\{N(t)N(u)\}$. Under mild conditions on $R(t, u)$, there is a *whitening filter* $h(t, u)$ with the property that

$$N'(t) = \int_0^T h(t, u)N(u)\,du, \qquad 0 \le t < T$$

is a white Gaussian noise process with unit power spectral density. This filter can be used to transform a detection problem involving $N(t)$ into an equivalent problem involving white Gaussian noise.

Detection in Non-Gaussian Noise

We have thus far focused exclusively on Gaussian noise models. Gaussian processes can accurately model many important noise sources encountered in practice, such as electrical noise due to thermal agitation of electrons in the receiver electronics, radio emissions from the motion of ions in the atmosphere, and cosmic background radiation. However, other sources of noise are not well described by Gaussian distributions, such as impulsive noise due to atmospheric disturbances or radar clutter.

While the receivers presented in this article can be used in the presence of non-Gaussian noise, they are not optimal for this purpose and may perform poorly in comparison to the likelihood ratio detector [Eq. (3)] based on the actual non-Gaussian noise statistics. In contrast to the simple linear correlation operations that arise in Gaussian detection problems, optimal detectors for non-Gaussian noise often involve more complicated nonlinear operations. A thorough treatment of detection methods for i.i.d. non-Gaussian noise can be found Kassam (3). A recent survey of detection techniques for dependent non-Gaussian noise sequences is given in Poor and Thomas (4).

Nonparametric Detection

Throughout this article, we have assumed the receiver knows the probability density of the observation under each hypothesis, with the possible exception of a real-valued parameter θ. Under this assumption, the detection problem is a choice between composite hypotheses that each represents a collection of possible densities, say

$$\Omega_0 = \{p_0(y|\theta) : \theta \in \Theta_0\}, \qquad \Omega_1 = \{p_1(y|\theta) : \theta \in \Theta_1\}$$

This is called a *parametric model,* because the set of possible probability distributions under both hypotheses can be indexed by a finite number of real parameters.

In practice, however, precise models for the signal and the underlying noise statistics are frequently not available. In such cases, it is desirable to find detectors that perform well for a large class of possible probability densities. When the probability classes Ω_0 and Ω_1 are so broad that a parametric model cannot describe them, the model is said to be *nonparametric.* In general, nonparametric detection methods may be classified as robust or simply nonparametric depending on the breadth of the underlying probability classes Ω_0 and Ω_1.

In *robust detection,* the probability densities of the observation are known approximately under each hypothesis and the aim is to design detectors that perform well for small deviations from these densities. Usually, the probability classes Ω_0 and Ω_1 consist of small nonparametric neighborhoods of the nominal probability densities. One widely studied model for these neighborhoods is the *ϵ-contamination class*

$$\Omega_i = \{p(y) : p(y) = (1-\epsilon)p_i(y) + \epsilon h(y)\}, \qquad i = 0, 1$$

where $p_i(y)$ is the nominal probability density under hypothesis H_i, $0 \le \epsilon < 1$ is small enough so that Ω_0 and Ω_1 do not overlap, and $h(y)$ is an arbitrary probability density. In robust detection, the performance of a detector d is typically measured by worst-case performance over all probability densities in Ω_0 and Ω_1. Optimal detectors are those that yield the best worst-case performance. For ϵ-contamination models, the optimal robust detector consists of a likelihood ratio detector for the nominal probability densities that includes some type of soft-limiting operation. For example, a robust form of the correlation receiver (appropriate for small deviations from the Gaussian noise model) is obtained by replacing $Y_i s_i$ with $g(Y_i s_i)$ in Eq. (10), where g is a soft-limiter of the form

$$g(x) = \begin{cases} b & \text{if } x > b \\ x & \text{if } a < x < b \\ a & \text{if } x < a \end{cases}$$

An extensive survey of the robust detection literature prior to 1985 can be found in Kassam and Poor (5).

The term *nonparametric detection* is usually reserved for situations in which very little is known about the probability distribution of the underlying noise, except perhaps that it is symmetric and possesses a probability density. In such situations, the aim is to develop detectors that provide a guaranteed false-alarm probability over very wide classes of noise distributions. The simplest nonparametric detector is the *sign detector,* which counts the number of positive observations in a sequence and compares it to a threshold. It can be shown that this detector provides a constant false-alarm probability for detecting the presence or absence of a constant positive signal in any i.i.d. zero-median additive noise sequence. A discussion of further results in nonparametric detection may be found in Gibson and Melsa (6).

Sequential Detection

All of the discrete-time detection problems considered above involve a fixed number of observations. There are some situations, however, in which it may be advantageous to vary the number of observations. In radar systems, for example, a series of observations might correspond to repeated measurements of a weak target. Naturally, we want to detect the target as soon as possible—that is, using the fewest observations. Detection methods that permit a variable number of observations are the subject of *sequential detection.* Such methods are applicable whenever each observation carries a cost, and we want to minimize the overall average cost of making a reliable decision.

One of the most important techniques in sequential detection is a Neyman–Pearson-type test called the *sequential probability ratio test* (SPRT) (1). Suppose we want to decide between the two hypotheses

$$H_0: \quad Y_i \text{ is i.i.d. with pdf } p_0(y), \qquad i = 1, 2, \ldots$$
$$H_1: \quad Y_i \text{ is i.i.d. with pdf } p_1(y), \qquad i = 1, 2, \ldots$$

using the smallest average number of observations necessary to achieve a probability of false alarm P_{F} and probability of miss P_{M}. The SPRT involves testing the accumulated data after each observation time $j = 1, 2, \ldots$. The test statistic at time j consists of the likelihood ratio of all observations up to that time, that is,

$$\Lambda_j(y_1, \ldots, y_j) = \frac{\prod_{i=1}^{j} p_1(y_i)}{\prod_{i=1}^{j} p_0(y_i)}$$

At time j, we calculate $\Lambda_j(Y_1, \ldots, Y_j)$ and compare it to two thresholds, τ_0 and τ_1. If $\Lambda_j \geq \tau_1$ we decide in favor of H_1, if $\Lambda_j \leq \tau_0$ we decide in favor of H_1, otherwise we take another observation and repeat the test. The thresholds τ_0 and τ_1 are chosen to provide the desired false-alarm and miss probabilities. The SPRT minimizes the average number of observations under both H_0 and H_1, subject to constraints on P_F and P_M.

FURTHER READING

A more complete and detailed treatment of most of the topics covered in this article can be found in the books by Poor (1) and by Srinath, Rajasekaran, and Viswanathan (2). Further information on the applications of detection theory in communications and radar is contained in the books by Proakis (7) and by Nathanson (8).

Current research in signal detection and its applications is published in a wide variety of journals. Perhaps chief among these are the *IEEE Transactions on Information Theory, IEEE Transactions on Signal Processing,* and the *IEEE Transactions on Communications.*

BIBLIOGRAPHY

1. H. V. Poor, *An Introduction to Signal Detection and Estimation,* New York: Springer-Verlag, 1988.
2. S. Srinath, P. K. Rajasekaran, and R. Viswanathan, *Introduction to Statistical Signal Processing with Applications,* Englewood Cliffs, NJ: Prentice-Hall, 1996.
3. S. A. Kassam, *Signal Detection in Non-Gaussian Noise,* New York: Springer-Verlag, 1987.
4. H. V. Poor and J. B. Thomas, Signal Detection in Dependent Non-Gaussian Noise, in H. V. Poor and J. B. Thomas, (eds.), *Advances in Statistical Signal Processing,* Vol. 2, *Signal Detection,* Greenwich, CT: JAI Press, 1993.
5. S. A. Kassam and H. V. Poor, Robust techniques for signal processing: A survey, *Proc. IEEE,* **73**: 433–481, 1985.
6. J. D. Gibson and J. L. Melsa, *Introduction to Nonparametric Detection with Applications,* New York: Academic Press, 1975.
7. J. G. Proakis, *Digital Communications,* 2nd ed., New York: McGraw-Hill, 1989.
8. F. E. Nathanson, *Radar Design Principles,* New York: McGraw-Hill, 1991.

BRIAN L. HUGHES
North Carolina State University

} { { } }

Wiley Encyclopedia of Electrical and Electronics Engineering

Trellis-Coded Modulation
Standard Article

Hamid Jafarkhani[1] and Vahid Tarokh[2]
[1]AT&T Labs—Research, Red Bank, NJ
[2]AT&T Labs—Research, Red Bank, NJ

DOI: 10.1002/047134608X.W4216
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (143K)

Abstract

The sections in this article are

Historical Remarks

Overview

Trellises as Finite-State Machines

Mapping by Set-Partitioning

Decoding Trellis Codes: the Dynamic Programming Algorithm

Multidimensional Trellis Codes

Research Activities

TRELLIS-CODED MODULATION

Any communication in nature suffers from impairments such as noise, which corrupts the data transmitted from the transmitter to the receiver. In this article, we consider the principles behind trellis-coded modulation (TCM), which is an established method to combat the aforementioned impairments. TCM is one of the main components of the modern modulator-demodulator (modem) systems for data transmission over telephone lines.

HISTORICAL REMARKS

Trellis diagrams (or state transition diagrams) were originally introduced in communications by Forney (1) to describe maximum likelihood sequence detection of convolutional codes. They were employed to soft decode convolutional codes using a dynamic programming algorithm (also known as the Viterbi algorithm).

The concept of trellis was later extended by Bahl et al. (2) to linear block codes where they were used as a natural framework to implement the maximum a posteriori probability (MAP) algorithm. Later, Forney unveiled the trellis structure of Euclidean Codes and Lattices.

Trellis-coded modulation is perhaps the most frequently applied branch of trellis theory. Such an implementation combines channel coding and modulation for transmission over band-limited channels. Specifically, trellis-coded modulation integrates the trellis of convolutional codes with M-ary linear modulation schemes such as, for example, M-phase-shift keying. Generally, modulation schemes containing larger Euclidean distances between their signal sets provide more robustness against noise over Gaussian channels. On the other hand, traditionally channel codes were designed so that distinct codewords have large Hamming distances (3). These two criteria are not equivalent unless 2-amplitude modulation or 4-phase-shift keying (4-PSK) modulation is used. Combining channel coding and modulation makes it possible to use a distance measure in coding which is equivalent to Euclidean distance in modulation. When the noise is additive white Gaussian, trellis-coded modulation provides 3–6 dB improvements over uncoded modulation schemes for the same bandwidth efficiency. Although Massey had proposed the idea of combining channel coding and modulation in 1974 (4), the first trellis-coded modulation scheme was introduced by Ungerboeck and Csjaka in 1976 (5,6).

OVERVIEW

Figure 1 shows a block diagram of a communication system in which binary data are transmitted over a noisy channel. Since the signal transmitted over the physical channel is a continuous electrical waveform, the modulation scheme converts its binary (discrete) input to continuous signals which are suitable for transmission over band-limited channels. If the effects of noise on the transmitted signal can be modeled by adding uncorrelated Gaussian noise samples, the channel is called an *additive Gaussian noise* channel. The ratio of the transmitted power to the noise power, signal-to-noise ratio (SNR), is an important parameter which affects the performance of the modulation scheme. For a given SNR and band-

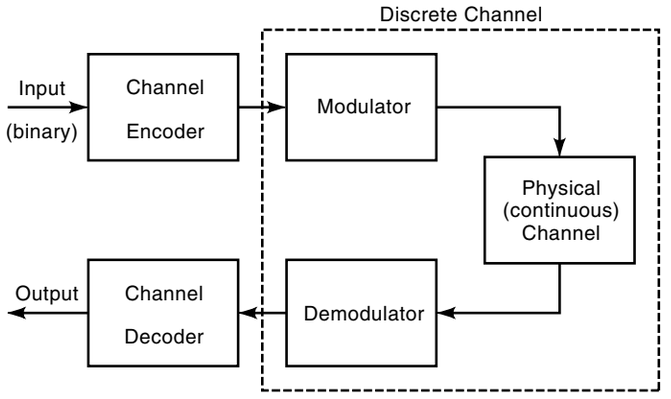


Figure 1. Block diagram of a communication system.

width, there is a theoretical limit for the maximum bit rate which can be reliably transferred over a continuous channel (Shannon capacity) (7). If the bit rate is less than the Shannon capacity, the objective of a modulation scheme is to minimize the bit error rate for a given SNR and a given bandwidth.

The combination of modulation, continuous channel, and demodulation can be considered as a discrete channel. Because of the hard-decision at the demodulator, the input and output of the discrete channel are binary. The effects of noise in the physical channel translates into bit errors in the discrete channel. The job of channel coding is to correct errors by adding some redundancy to the bit stream. In other words, error correcting codes systematically add new bits to the bit stream such that the decoder can correct some of the bit errors by using the structure of the redundancy. Of course, the adding redundancy reduces the effective bit rate per transmission bandwidth.

Before the seminal work of Ungerboeck and Csjaka, channel codes and modulation schemes were designed separately. Error correcting codes were designed to have codewords with large Hamming distance from each other. Modulation schemes utilize signal sets with maximum Euclidean distance. Since Hamming distance and Euclidean distance are not equivalent for most modulation schemes, designing modulation and coding scheme separately results in about 2 dB loss in SNR. In contrast, trellis-coded modulation is designed to maximize Euclidean distance between the channel signal sets by combining channel codes and modulation (Fig. 2). For a given rate and bandwidth, trellis-coded modulation uses a redundant signal set at the modulator and a maximum likelihood soft decoder at the demodulator. In trellis-coded modulation, the necessary redundancy of coding comes from expanding the signal sets not bandwidth expansion, as will be discussed in the next section. Designing a good coded modulation scheme is possible by maximizing the free Euclidean distance for the code. In fact, Ungerboeck and Csjaka's point of departure from traditional coding is that the free distance of a trellis-coded modulation can be significantly more than that of the corresponding uncoded modulation scheme.

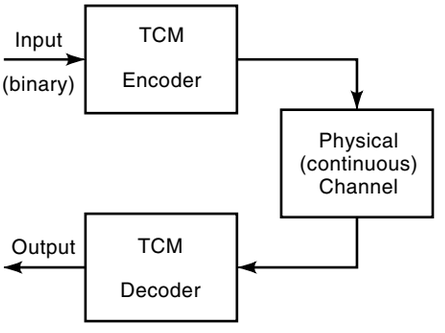


Figure 2. Using trellis-coded modulation to combine channel coding and modulation.

A trellis (state-transition diagram) can be used to describe trellis-coded modulation. This trellis is similar to that of convolutional codes. However, the trellis branches in trellis-coded modulation consist of modulation signals instead of binary codes. Since the invention of trellis-coded modulation, it has been used in many practical applications. The use of trellis-coded modulation in modulator–demodulators (modems) for data transmission over telephone lines has resulted in tremendous increased in the bit rate. International Telegraph and Telephone Consultative Committee (CCITT) and its successor International Telecommunication Union (ITU) have widely utilized trellis-coded modulation in high-speed modems for data transmission over telephone lines (8–10).

TRELLISES AS FINITE-STATE MACHINES

Much of the existing literature (11–13) uses *set partitioning* and trellis structure of convolutional codes to describe trellis-coded modulation. This may be attributed to the fact that this approach was taken by Ungerboeck and Csjaka in their seminal paper where the foundation of coded modulation was laid. In this exposition, the goal is to present the results with the required background kept as small as possible. In this light, we pursue a different line of thought and approach the topic using finite-state machines.

Finite-State Machines

A *finite-state machine* can be thought of as a three-tuple $\mathscr{M} = (\mathscr{S}, \mathscr{T}, \mathscr{L})$, where $\mathscr{S}$, $\mathscr{T}$, and $\mathscr{L}$, respectively are referred to as the *set of states,* the *set of transitions,* and the *defining alphabet* of $\mathscr{M}$. Each element of the set $\mathscr{T}$ is a *transition* (s_i, s_e, l) with $s_i, s_e \in \mathscr{S}$ and $l \in \mathscr{L}$. Such a transition is said to start in s_i, end in s_e, and is labelled with l. All transitions starting from the same state, s_i, and ending at the same state, s_e, are called *parallel transitions.* For each state s, the number of transitions starting (respectively ending) in s is called the out-degree (respectively the in-degree) of s.

The finite-state machine $\mathscr{M}$ is said to be regular if the in-degrees and out-degrees of all the states of $\mathscr{S}$ are the same. The machine $\mathscr{M}$ is binary if it is regular and if the out-degrees and in-degrees of elements of $\mathscr{S}$ as well as the number of states of $\mathscr{S}$ are powers of 2. In this article, we are only interested in binary machines.

The Trellis of a Binary Finite-State Machine

Every finite-state machine $\mathscr{M}$ has a trellis diagram $T(\mathscr{M})$ which is a graphical way to represent the evolution path of $\mathscr{M}$. Let $\mathscr{M}$ denote a binary finite-state machine having 2^n states. A trellis diagram $T(\mathscr{M})$ of $\mathscr{M}$ is defined as a labelled directed graph having levels 0, 1, 2, 3, ... Each level of $\mathscr{M}$

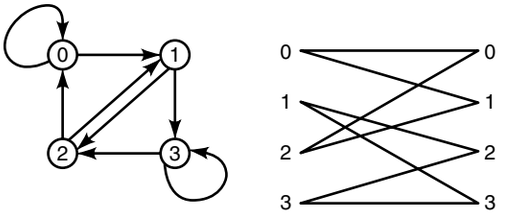


Figure 3. A four-state finite-state machine and the corresponding trellis. Graphical equivalence between trellises and finite state machines is clearly visible.

has 2^n states labelled 0, 1, . . ., $2^n - 1$ corresponding to, respectively, s_0, s_1, . . ., s_{2^n-1} elements of $\mathcal{S}$. There is an edge labelled with l between state i of level k and j of level $k + 1$ if and only if $(s_i, s_j, l) \in \mathcal{T}$ where $i, j = 1, 2, \ldots, 2^n$, $k = 1, 2, \ldots$ and $l \in \mathcal{L}$.

Figure 3 shows an example of a finite-state machine $\mathcal{M}$ containing four states and the corresponding trellis diagram $T = T(\mathcal{M})$. In Fig. 3, we only show the transitions between different states (not the labels). One can use different labels on the transitions to construct different codes. This is the subject of the next section. It is clear that given a trellis diagram T as defined, one can construct a finite-state machine $\mathcal{M}$ such that $T = T(\mathcal{M})$ and vice versa.

Trellis Codes

A *trellis code* is the trellis of a binary finite-state machine where the alphabet $\mathcal{L}$ comes from a signal constellation having unit average energy (we use unit average energy for all signal constellations in this article). Practical signal modulation includes but is not restricted to the 4-PSK, 8-PSK, and 16-quadrature amplitude (16-QAM) constellations. In this light, we only consider these signal constellations here.

Let $\mathcal{M}$ denote a trellis code with 2^n states such that the in-degree and out-degree of each state is 2^R. Let $T(\mathcal{M})$ denote the trellis of $\mathcal{M}$ and assume that at time zero the machine is at state zero. The trellis code $\mathcal{M}$ can be used to encode R bits of information at each time instance. At each time $t = 0, 1, 2, \ldots$ a block of R bits of data denoted by $B(t)$ arrives at the encoder. Depending on the 2^R possible values of this block of data and the state $s_i(t)$ of the machine at time t, a transition beginning in that state such as $(s_i(t), s_e(t + 1), l(t))$ is chosen. The trellis code then moves to the state $s_e(t + 1)$ and outputs $l(t)$ the label of the transition. Thus, $B(0)B(1)B(2) \ldots$ is mapped to the codeword $l(0)l(1)l(2) \ldots$. We let $C(\mathcal{M})$ denote the set of all possible output sequences and also refer to it as the *code* of $\mathcal{M}$ when there is no ambiguity.

The alert reader notices that such an encoder may be completely useless. Indeed, if all the transitions are labelled with the same signal constellation symbol, all bit sequences will be mapped to the same codeword. Thus, it is important to design the trellis code so that such a scenario is avoided.

The assignment of labels to transition in particular is what determines the performance of a code over a transmission media. Thus, a performance criterion is needed before designing a trellis code for real applications. In most of the situations, an exact performance criterion is intractable for design and a tractable approximate criterion is used instead. Tractable approximate design criteria are known for the Gaussian channel, rapidly fading channel, slowly fading channel, and numerous other cases. A good general reference for trellis codes is (14).

Trellis Codes for the Gaussian Channel

The design criterion (albeit an approximate one) for the Gaussian channel is well established in the literature. In general a code $\mathcal{C}$ is expected to perform well over a Gaussian channel if the codewords are as far from each other (in terms of Euclidean distance) as possible. The computation of Euclidean distance of two codewords of a code is not that difficult and hence this criterion is tractable for design. To remove any ambiguity, we mathematically define the distance between two paths of $T(\mathcal{M})$ with the same starting and ending states. Without loss of generality, let us assume that the two paths emerge at time $t = 0$ and remerge at time $t = t'$. Suppose that the branches are labelled c_t^1 and c_t^2, $t = 0, 1, \ldots, t'$, for the first and second path, respectively. Then, the distance between the two paths is defined by $\sum_{t=0}^{t'} |c_t^1 - c_t^2|^2$.

For the design of a trellis code $\mathcal{M}$, the minimum of distances between any two paths of $T(\mathcal{M})$ that emerge from some state at some time and remerge at another state of the trellis at a later time dominates the performance of the code. This quantity is called the *free distance* of the trellis code. Thus trellis codes that are useful for Gaussian channel must have large free distances.

However, in pursuing such a design, we should take the bandwidth requirements into account. Fixing the symbol duration (time to transmit a constellation symbols), the dimensionality of the signal constellation directly relates to the bandwidth requirement for the channel. This is a fundamental result known as the Landau–Pollak–Slepian Theorem (15,16). The consequence of this result is that a comparison between the free distances of two trellis codes is justified only if they use signal constellations of same dimensionality.

An Ungerboek–Csjaka Idea

Suppose that we would like to design a trellis code for the transmission of R bits per channel use. One way of transmission is using a trellis code $\mathcal{M}$ that has one state and use a signal constellation $\mathcal{SC}$ having 2^R elements. The 2^R edges between the state of level t with that of $t + 1$ in $T(\mathcal{M})$ are labelled with the different signal constellation symbols. This trellis code is called the *uncoded* signal constellation $\mathcal{SC}$. The uncoded binary phase-shift keying (BPSK) constellation is given in Fig. 4. Clearly the free distance of the uncoded signal constellation $\mathcal{SC}$ is the minimum distance between the points of $\mathcal{SC}$.

One way of obtaining larger free distances is to use a signal constellation having more than 2^R elements for transmission of R bits per channel use. In practice, it is good to double the constellation size while designing over the Gaussian channels. As the dimensionality of the signal constellation is fixed and the number of signals in the constellation is doubled, we can expect a reduction in minimum distance of the new constellation.

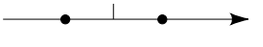


Figure 4. An uncoded BPSK constellation. Each point represents a signal to be transmitted over the channel.

As an example to transmit 1 bit per channel use we will use a 4-PSK (Fig. 5) instead of BPSK constellation. The minimum distance of the 4-PSK constellation is $\sqrt{2}$ while the minimum distance of the BPSK constellation is 2 (both have unit average energy). Thus, there is a loss in minimum distance by doubling the size of constellation. A Trellis code on 4-PSK alphabet can only be useful as compared to the uncoded case if it can compensate this loss by having a larger free distance than 2.

Ungerboeck and Csjaka demonstrated that there exist trellis codes that can outperform the uncoded signal constellations. They also proposed mapping by set partitioning as the machinery to construct these trellis codes.

MAPPING BY SET-PARTITIONING

Let $\mathcal{SC}$ be a signal set. Let $\mathcal{SC}_1 \subseteq \mathcal{SC}$ such that $|\mathcal{SC}|$ the number of elements of $\mathcal{SC}$ be a multiple of $|\mathcal{SC}_1|$. A *partitioning* of $\mathcal{SC}$ based on $\mathcal{SC}_1$ is a collection Σ_1 of disjoint subsets of $\mathcal{SC}$ such that Σ_1 contains $\mathcal{SC}_1$ and $\cup_{X \in \Sigma_1} X = \mathcal{SC}$. Elements of Σ_1 are called the *cosets* of $\mathcal{SC}_1$ in $\mathcal{SC}$. The concept of partitioning can be extended to the nested chains of subsets of $\mathcal{SC}$.

Specifically, consider a decreasing chain of subsets of a signal constellation $\mathcal{SC}$

$$\mathcal{SC} = \mathcal{SC}_0 \supseteq \mathcal{SC}_1 \supseteq \mathcal{SC}_2 \supseteq \ldots \supseteq \mathcal{SC}_J$$

such that $|\mathcal{SC}_i|$ is a multiple of $|\mathcal{SC}_{i+1}|$ for $i = 0, 1, \ldots, J-1$. Such a decreasing chain induces partitioning in each level. First, $\mathcal{SC}$ is partitioned into a set Σ_1 of cosets of $\mathcal{SC}_1$ in $\mathcal{SC}$ which in particular contains $\mathcal{SC}_1$. Each element of Σ_1 contains $|\mathcal{SC}_1|$ elements of $\mathcal{SC}$. In a similar way, $\mathcal{SC}_1$ can be partitioned into cosets of $\mathcal{SC}_2$ in $\mathcal{SC}_1$ and the other elements of Σ_1 can be partitioned into sets of cardinality $|\mathcal{SC}_2|$. The result is Σ_2, the collection of all the cosets of $\mathcal{SC}_2$ in $\mathcal{SC}$ which in particular includes $\mathcal{SC}_2$. The process is then repeated for J times and all the cosets of $\mathcal{SC}_i$ in $\mathcal{SC}_j$ for $1 \le j \le i \le J$ are derived. In this article, we are only interested in partitions based on *binary* chains corresponding to the case when $|\mathcal{SC}_i|$, $i = 1, 2, \ldots, J$, are powers of two.

The central theme of the Ungerboeck–Csjaka paper (5) is that given a binary set partitioning based on a decreasing chain of subsets of $\mathcal{SC}$ as described, the minimum distance of cosets of $\mathcal{SC}_i$ in $\mathcal{SC}$ is a nondecreasing function of i. Indeed, if the partitioning is done in a clever way, the distances can substantially increase. Examples of such a set partitioning for the 4-PSK, 8-PSK, and 16-QAM are given in Figs. 6, 7, and 8, respectively. The notations

$$\begin{aligned} A_k &= \cos(2\pi k/4) + \sin(2\pi k/4)\mathbf{j}, k = 0, 1, 2, 3 \\ B_k &= \cos(2\pi k/8) + \sin(2\pi k/8)\mathbf{j}, k = 0, 1, 2, \ldots, 7 \\ Q_{k_1,k_2} &= ((2k_1 - 3) + (2k_2 - 3)\mathbf{j})/\sqrt{10}, \\ & k_1 = 0, 1, 2, 3, k_2 = 0, 1, 2, 3 \end{aligned}$$

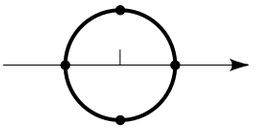


Figure 5. An uncoded 4-PSK constellation. Each point represents a signal to be transmitted over the channel.

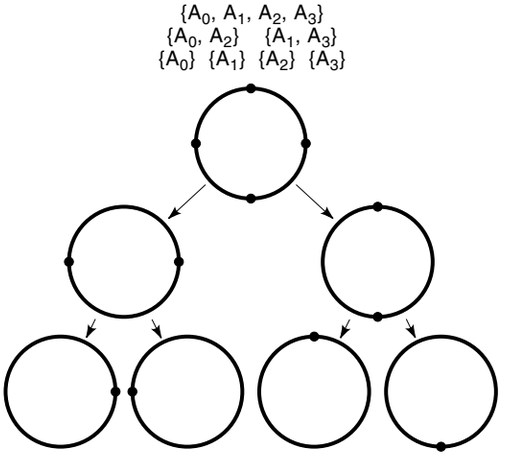


Figure 6. Set partitioning for 4-PSK constellation. The partitioning increases the minimum distance in each level.

(where $\mathbf{j} = \sqrt{-1}$) are used to represent the 4-PSK, 8-PSK, and 16-QAM constellations throughout this article.

As can be seen from Fig. 8, the minimum distances of the partitions in the 16-QAM case increase by a factor of $\sqrt{2}$ for each level. By choosing appropriate signals from each partition level as the labels of transitions of a finite-state machine, we could achieve very high free distances. This is the heart of Ungerboeck–Csjaka design and is called *mapping by set partitioning*.

The general heuristic rules established for design by Ungerboeck–Csjaka are

- Parallel transitions (those starting from and ending in the same states) are assigned to signal points with maximum Euclidean distance.
- The signal points should occur with the same frequency.
- Transitions originating from and merging into any state are assigned from elements of different cosets.

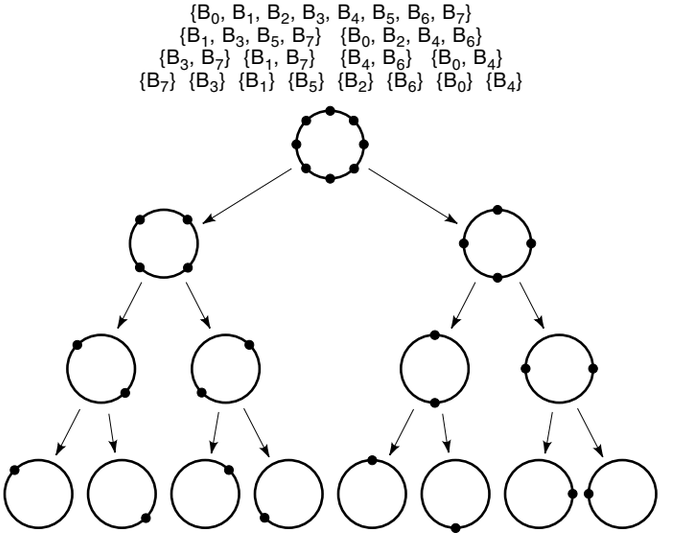


Figure 7. Set partitioning for 8-PSK constellation. The partitioning increases the minimum distance in each level.

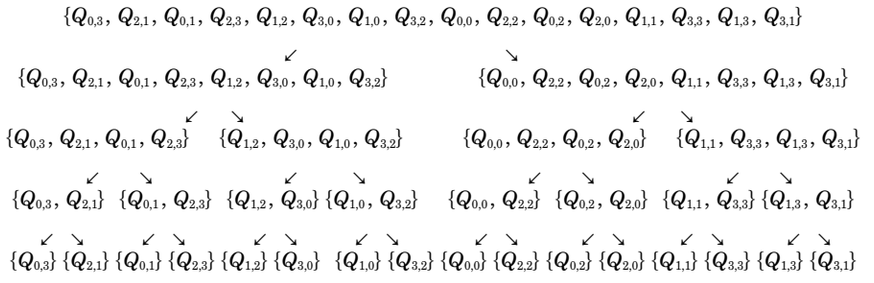


Figure 8. Set partitioning for 16-QAM constellation. The partitioning increases the minimum distance in each level.

These rules follow the intuition that good codes should have symmetry and large free distances. Examples of 4-PSK, 8-PSK, and 16-QAM codes are given in Tables 1–5.

From these tables, it is clear that by increasing the number of states in the trellis, the free distance (and hence the performance) can be improved. However, we will see that this has a penalty in terms of decoding complexity.

Let us now consider an example. Consider the set partitioning of the 8-PSK and the four-state trellis code given in Table 3 based on the previous partitioning. As can be seen from the table, the labels of the transitions originating from each state of the trellis belong to the same coset while those of distinct states belong to different cosets. The design has a lot of symmetries as it is expected that good codes should demonstrate a lot of symmetries. It can be easily shown that free distance of the previous trellis code is $\sqrt{2}$ times the minimum distance of a 4-PSK constellation. This translates into 3-dB asymptotic gain (in SNR). In general the asymptotic gain of a trellis code with rate R bits per channel use (2^{R+1} elements in the constellation) over an uncoded constellation with the same rate is defined by 10 log $d^2_{\text{free}}/d^2_{\min}$ where d_{free} is the minimum free distance of the code and $d_{\min}$ is the minimum distance between the uncoded constellation elements.

Figures 9 and 10 give information about the coding gain versus the number of states of best 8-PSK and 16-QAM trellis codes known for transmission of 2 and 3 bits/channel use, respectively.

DECODING TRELLIS CODES: THE DYNAMIC PROGRAMMING ALGORITHM

Decoding trellis codes is usually done through the dynamic programming algorithm also known as the Viterbi algorithm. The Viterbi algorithm is in some sense an infinite algorithm that decides on the path taken by the encoder. This was proved to be optimum for sequence estimation by Forney. However, in practice one has to implement a finite version of the algorithm. Naturally, only practice is of interest here.

Table 1. A 4-State 4-PSK Trellis Code

	$s_e = 0$	$s_e = 1$	$s_e = 2$	$s_e = 3$
$s_i = 0$	A_0	A_2		
$s_i = 1$			A_1	A_3
$s_i = 2$	A_2	A_0		
$s_i = 3$			A_3	A_1

Note: The states s_i and s_e are, respectively, the beginning and ending states. The corresponding transition label is given in the table. Blank entries represent transitions that are not allowed.

To understand the implementation of the decoder, we first define the *constraint length* $\nu(C)$ of a trellis code $C(\mathcal{M})$ to be the minimum t such that there exists two paths of time length t starting at the same state and remerging at another state. Practically, we choose a multiple of $\nu(C)$ depending on the decoding delay allowed in the application and refer to it as the *decoding depth* $\theta(C)$. We then proceed to execute the finite decoding depth Viterbi algorithm. At each stage of the algorithm, for every possible state s of the encoder, a *survivor path* $P_t(s)$ of length $\theta(C)$ and an accumulated metric $m_t(s)$ is preserved. We denote the possible states of the encoder by s_i, $i = 0, 1, \ldots, 2^n - 1$, and the received signal at time t by r_t. We always follow the convention that the encoder is in the zero state at time zero.

The decoder starts by setting $m_0(s_0) = 0$ and $m_0(s_i) = \infty$ for all $i = 1, 2, \ldots, 2^n - 1$. In practice, one can choose a large number instead of ∞. Further, at the beginning of the decoding process, the decoder sets the survivor paths $P_t(s_i)$, $i = 0, 1, 2, \ldots, 2^n - 1$, to be the void string. In other words, at the beginning of the decoding nothing is saved as the survivor paths of each state.

The decoder then starts decoding by computing the branch metrics of each branch at time $t = 0, 1, 2, 3, \ldots$. Suppose that a branch at time t is labelled with c_t, then the metric of this branch is $|r_t - c_t|^2$. The decoder computes for each state s_i, the sum of the accumulated metric $m_t(s_j)$ and the branch metric of any state s_j with any branch starting at state s_j at time t and ending in state s_i at time $t + 1$. The decoder then computes the minimum of all these possible sums and sets $m_{t+1}(s_j)$ to be this minimum. If this minimum is given by the state i at time t and some branch b_t, the survivor path $P_{t+1}(s_j)$ is given by the path $P_t(s_i)$ continued by the branch b_t. This process is then repeated at each time.

The decoder starts outputting decision bits after time $t \geq \theta(C)$, where $\theta(C)$ denotes the decoding depth. At each time $t \geq \theta(C)$, the decoder looks at the survivor path of the state with the lowest accumulated metric. The decoder outputs the sequence of bits corresponding to the branch of path at time $t - \theta(C)$. In this way, a decoding delay of $\theta(C)$ must be tolerated.

MULTIDIMENSIONAL TRELLIS CODES

The trellis codes constructed in the previous section use an element of a two-dimensional constellation for labels. It is neither necessary to have a two-dimensional constellation nor only one symbol of the constellation per label of transitions. This gives rise to *multidimensional trellis codes* or M-TCM codes.

Table 2. An 8-State 4-PSK Trellis Code

	$s_e = 0$	$s_e = 1$	$s_e = 2$	$s_e = 3$	$s_e = 4$	$s_e = 5$	$s_e = 6$	$s_e = 7$
$s_i = 0$	A_0	A_2						
$s_i = 1$			A_1	A_3				
$s_i = 2$					A_2	A_0		
$s_i = 3$							A_3	A_1
$s_i = 4$	A_0	A_2						
$s_i = 5$			A_1	A_3				
$s_i = 6$					A_2	A_0		
$s_i = 7$							A_3	A_1

Note: The states s_i and s_e are, respectively, the beginning and ending states. The corresponding transition label is given in the table. Blank entries represent transitions that are not allowed.

Table 3. A 4-State 8-PSK Trellis Code

	$s_e = 0$	$s_e = 1$	$s_e = 2$	$s_e = 3$
$s_i = 0$	B_0, B_4	B_2, B_6		
$s_i = 1$			B_1, B_5	B_3, B_7
$s_i = 2$	B_2, B_6	B_0, B_4		
$s_i = 3$			B_3, B_7	B_1, B_5

Note: The states s_i and s_e are, respectively, the beginning and ending states. The corresponding possible transition labels are given in the table. Blank entries represent transitions that are not allowed.

Table 4. An 8-State 8-PSK Trellis Code

	$s_e = 0$	$s_e = 1$	$s_e = 2$	$s_e = 3$	$s_e = 4$	$s_e = 5$	$s_e = 6$	$s_e = 7$
$s_i = 0$	B_0	B_4	B_2	B_6				
$s_i = 1$					B_1	B_5	B_3	B_7
$s_i = 2$	B_4	B_0	B_6	B_2				
$s_i = 3$					B_5	B_1	B_7	B_3
$s_i = 4$	B_2	B_6	B_0	B_4				
$s_i = 5$					B_3	B_7	B_1	B_5
$s_i = 6$	B_6	B_2	B_4	B_0				
$s_i = 7$					B_7	B_3	B_5	B_1

Note: The states s_i and s_e are, respectively, the beginning and ending states. The corresponding possible transition labels are given in the table. Blank entries represent transitions that are not allowed.

Table 5. A 4-State 16-QAM Trellis Code

	$s_e = 0$	$s_e = 1$	$s_e = 2$	$s_e = 3$
$s_i = 0$	$Q_{1,3}, Q_{3,3}, Q_{1,1}, Q_{3,1}$	$Q_{0,0}, Q_{0,2}, Q_{2,0}, Q_{2,2}$		
$s_i = 1$			$Q_{0,1}, Q_{0,3}, Q_{2,1}, Q_{2,3}$	$Q_{1,0}, Q_{1,2}, Q_{3,0}, Q_{3,2}$
$s_i = 2$	$Q_{0,0}, Q_{0,2}, Q_{2,0}, Q_{2,2}$	$Q_{1,3}, Q_{3,3}, Q_{1,1}, Q_{3,1}$		
$s_i = 3$			$Q_{1,0}, Q_{1,2}, Q_{3,0}, Q_{3,2}$	$Q_{0,1}, Q_{0,3}, Q_{2,1}, Q_{2,3}$

Note: The states s_i and s_e are, respectively, the beginning and ending states. The corresponding possible transition labels are given in the table. Blank entries represent transitions that are not allowed.

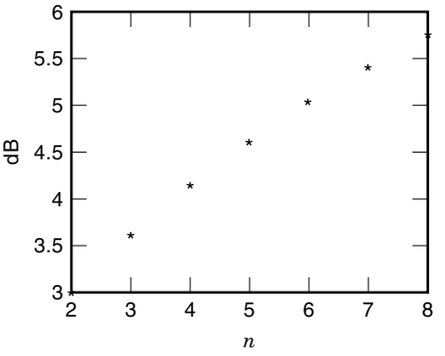


Figure 9. Asymptotic coding gain of coded 8-PSK over uncoded 4-PSK (number of states = 2^n). Coding gain represents the improvement in the performance of the coded system over that of the uncoded system.

An example of an M-TCM code is given in Ref. 16, which is a four-dimensional trellis code known as the Wei code (17).

RESEARCH ACTIVITIES

An active area of theoretical research is studying the tradeoff between the complexity and coding gain of trellis codes. In essence, we would like to see trellis codes with lower complexity of decoding and higher coding gain. Much effort has been put into finding solutions to this problem, but only meager improvements have been observed over the codes constructed in the original paper of Ungerboeck–Csjaka.

A second active area is to find suboptimal algorithms for decoding trellis codes which give performance close to that of the optimum Viterbi algorithm. Numerous papers have been written on this topic proposing reduced complexity algorithms including the sequential decoding algorithm and the *M*-algorithm (18). These decoding algorithms perform close to optimal but do not seem promising due to other implementation problems including the problem with buffer overflow.

Another research area is to combine mathematical objects called *lattices* with trellis codes (14). These theoretically achieve higher coding gains but have other implementation problems including the design of slicer and increased decoding complexity.

Trellis ideas were also applied to quantization giving rise to *trellis-coded quantization* which can be used to quantize various sources (19,20).

In general, we believe that a fruitful area of research may be the study of implementation issues of trellis codes over channels with ISI and non-Gaussian channels in the presence of various impairments due to practical situations. There is a well-established body of literature on this topic (21,22) but we believe that there is a lot more to be done.

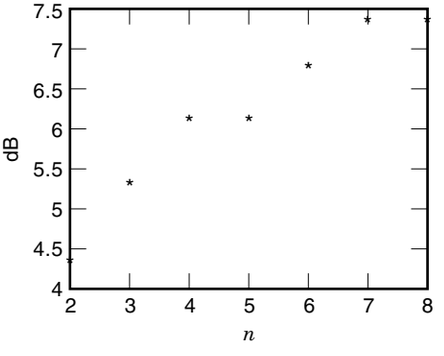


Figure 10. Asymptotic coding gain of coded 16-QAM over uncoded 8-PSK (number of states = 2^n). Coding gain represents the improvement in the performance of the coded system over that of the uncoded system.

BIBLIOGRAPHY

1. G. D. Forney, Trellises old and new, in *Communications and Cryptography: Two Sides of One Tapestry,* R. E. Blahut et al. (eds.). Dordrecht, The Netherlands: Kluwer, 1994.
2. L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, Optimum decoding of linear codes for minimizing symbol error rates, *IEEE Trans. Inf. Theory,* **IT-20**: 284–287, 1974.
3. F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes,* New York: Elsevier, 1996.
4. J. L. Massey, Coding and modulation in digital communications, *Proc. 1974 Int. Zürich Seminar on Digital Commun.,* E2(1)–(4), Mar. 1974.
5. G. Ungerboeck and I. Csajka, On improving data-link performance by increasing the channel alphabet and introducing sequence coding, *Int. Symp. Inform. Theory,* Ronneby, Sweden, June 1976.
6. G. Ungerboeck, Channel coding with multilevel/phase signals, *IEEE Trans. Inf. Theory,* **IT-28**: 55–67, 1982.
7. C. E. Shannon, A mathematical theory of communication, *Bell Syst. Tech. J.,* **27**: 379–423, 1948.
8. CCITT, A family of 2-wire, duplex modems operating on the general switched telephone network and on leased telephone-type circuits, Recommendation V.32, 1984.
9. CCITT, 14400 bits per second modem standardized for use on point-to-point 4-wire leased telephone-type circuits, Recommendation V.33, 1988.
10. ITU-T, A modem operating at data signaling rates of up to 33600 bit/s for use on the general switched telephone network and on leased point-to-point 2-wire telephone-type circuits, Recommendation V.34, 1996.
11. G. Ungerboeck, Trellis-coded modulation with redundant signal sets part II: State of the art, *IEEE Commun. Magazine,* **25**: 12–21, 1987.
12. E. A. Lee and D. G. Messerschmitt, *Digital Communication,* Boston: Kluwer, 1988.
13. J. G. Proakis, *Digital Communications,* New York: McGraw-Hill, Inc. 1989.
14. E. Biglieri et al., *Introduction to Trellis-Coded Modulation with Applications,* New York: Macmillan, 1991.
15. H. O. Pollak and H. J. Landau, Prolate spheroidal wave functions, Fourier Analysis and uncertainty II, *Bell Syst. Tech. J.,* **40**: 65–84, 1961.
16. H. J. Landau and H. O. Pollak, Prolate spheroidal wave functions, fourier analysis and uncertainity III: The dimension of the space of essentially time and band-limited signals, *Bell Syst. Tech. J.,* **41**: 1295–1366, 1962.
17. L. F. Wei, Trellis-coded modulation with multi-dimensional constellations, *IEEE Trans. Inf. Theory,* **IT-33**: 483–501, 1987.

18. J. M. Wozencraft and B. Reiffen, *Sequential Decoding,* Cambridge, MA: MIT Press, 1961.
19. M. W. Marcellin and T. R. Fischer, Trellis-coded quantization of memoryless and Gauss-Markov sources, *IEEE Trans. Commun.,* **38**: 82–93, 1990.
20. M. Wang and T. R. Fischer, Trellis-coded quantization designed for noisy channels, *IEEE Trans. Inf. Theory,* **40**: 1792–1802, 1994.
21. D. Divsalar and M. K. Simon, The design of trellis-coded MPSK for fading channel: Performance criteria, *IEEE Trans. Comm.,* **36**: 1004–1012, 1988.
22. D. Divsalar and M. K. Simon, The design of trellis-coded MPSK for fading channel: Set partitioning for optimum code design, *IEEE Trans. Commun.* **36**: 1013–1021, 1988.

HAMID JAFARKHANI
VAHID TAROKH
AT&T Labs

TRELLIS CODES. See TRELLIS-CODED MODULATION.

TRENDS IN SYSTEMS ENGINEERING. See SYSTEMS ENGINEERING TRENDS.

TRENDS, SYSTEMS ENGINEERING. See SYSTEMS ENGINEERING TRENDS.

TRIANGLE WAVE GENERATION. See RAMP GENERATOR.