(2)

APPLICATIONS OF SIGNAL PROCESSING IN DIGITAL COMMUNICATIONS

Appendix

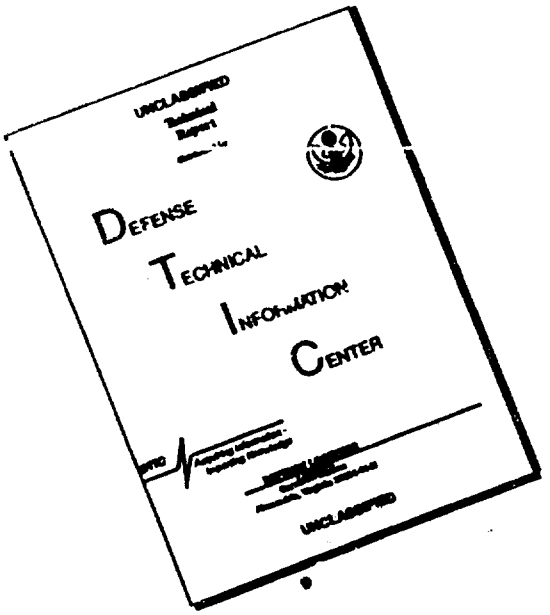
Final Technical Report
by
Michele Elia

April 1991

United State Army
EUROPEAN RESEARCH OFFICE OF THE U.S.ARMY
London England

Contract Number DAJA45-86-C-0044
R&D No. 5205-EE-01
Dipartimento di Elettronica - Politecnico di Torino
Corso Duca degli Abruzzi 24 - I-10129 Torino, Italy

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

Research Papers

Appendice A: E.Biglieri, M.Elia, Multidimensional Modulation and Coding for Bandlimited Digital Channels, *IEEE Transactions on Information Theory*, vol.IT-34, n.4, July 1988, pp.803-809.

Appendice B: E.Biglieri, S.Barberis, M.Catena, Analysis and Compensation of Nonlinearities in Digital Transmission Systems, *IEEE Journal on Selected Areas in Communications*, January 1988.

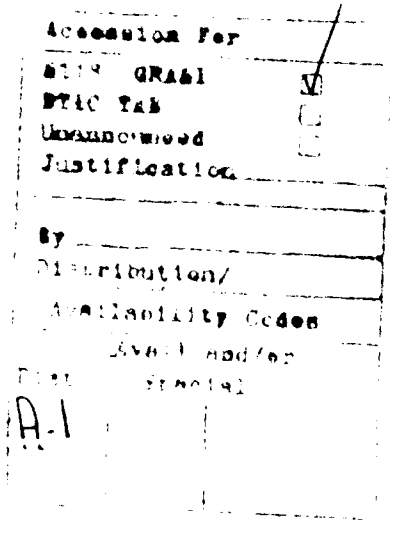
Appendice C: M.Elia, Group Codes and Signal Design for data Transmission, Campinas, Brasile ISICT'87, July 1987.

Appendice D: M.Elia, F.Neri, A Note on Addition Chains and Some Related Conjectures, Advanced Inter. Workshop on SEQUENCES, Combinatorics, Compression, Security and Transmission, Salerno, Italy, June 1988, pp.166-181.

Appendice E: M.Elia, C.Losana, F.Neri, A Note on the Complete Decoding of Kerdock Codes, *IEEE International Symposium on Information Theory*, Kobe, Japan, June 1988.

Appendice F: M.Elia, D.Vellata, Multiplication in Galois Field $GF(2^m)$, *Internal Report*, June 1988.

Appendice G: M.Elia, F.Neri, On the Concatenation of Binary Linear Codes, submitted to *IEEE Transactions on Communications*, 1989.



Appendix A

Multidimensional Modulation and Coding for Band-Limited Digital Channels

EZIO BIGLIERI, SENIOR MEMBER, IEEE, AND MICHELE ELIA, SENIOR MEMBER, IEEE

Abstract—A class of multidimensional signals, based on what we call generalized group alphabets, is introduced, and its basic properties are derived. The combination of generalized group alphabets and coding is also examined: two coding schemes are considered—Ungerboeck's scheme for combination with convolutional codes, and Ginzburg's scheme for combination with block codes. The performance of these schemes makes them attractive for transmission over band-limited digital channels.

I. Introduction

IN DIGITAL RADIO communications both the available spectrum and the transmitter power are limited. Thus to cope with the ever-increasing demand for more efficient transmission, new modulation techniques are needed. One way to increase the transmission efficiency, suggested by Shannon's fundamental theorem itself, is to increase the dimensionality of the signal space [1], [12]. For this solution to have practical applications, however, the system complexity should not increase prohibitively. Conventional systems, like quadrature amplitude modulation (QAM) and phase-shift keying (PSK), use two-dimensional signals obtained through the inphase and quadrature components of a sinusoidal carrier. Four-dimensional signal spaces can be realized in a similar way by simultaneously using two channels, each with separately modulated inphase and quadrature components. The two bandpass channels can be two orthogonally polarized electromagnetic waves, or time-division or frequency-division multiplexed signals transmitted on a common medium. Results on specific designs of four- and eight-dimensional signal sets can be found in [2]–[6].

We consider a structured class of multidimensional alphabets which we call "generalized group alphabets" that are based on a peak-energy constraint. They generalize the "group codes" of Slepian [7] that are based on an equal-energy constraint. The most striking feature of these alphabets is that they exhibit a considerable degree of symmetry.

Generalized group alphabets form a large class of codes; to date, most of the good alphabets that have been proposed for multidimensional signaling belong to this family. After a description of the main features of these alphabets, we show how they can be used in conjunction with error-control codes. For this purpose the alphabets must be partitioned into a chain of subsets, where the minimum distance between subsets increases with depth. The concept of a fair partition is introduced, and it is shown how it can be obtained through the action of a group of orthogonal matrices on a set of vectors. The method of dividing a signal alphabet into subsets via the action of an orthogonal group is due to Ginzburg [8]. Finally, we provide some examples of actual designs that show how our techniques can be applied to generate codes. However, no attempt has been made to discover optimum codes.

II. Generalized Group Alphabets

Consider a set of K n-dimensional vectors $X = \{X_1, \cdots, X_K\}$, called the *initial set*, and L orthogonal $n \times n$ matrices $S_1, \cdots, S_L$ that form a finite group G under multiplication.

Definition 1: The set of vectors $GX_1, GX_2, \cdots, GX_K$ obtained from the action of G on the vectors of the initial set is called a *generalized group alphabet* (GGA). G is called its *generating group*.

Definition 2: A GGA is called *separable* if the vectors of the initial set are transformed by G into either disjoint or coincident vector sets, i.e.,

$$GX_j \cap GX_k = \begin{cases} 0, & j \neq k \\ GX_j, & j = k. \end{cases}$$

If $\|X\|$ denotes the Euclidean length of a vector X, the quantity $\|X\|^2$ is proportional to the energy of the signal associated with X for transmission over a continuous channel. Since an orthogonal matrix transforms a vector into one with the same length, the signals associated with a GGA have as many energy levels as there are in the initial set. The special case of a GGA with $K = 1$, and hence only one energy level, was extensively studied in [7].

Definition 3: A GGA is called *regular* if the number of vectors in each subalphabet GX_j, $j = 1, \cdots, K$, does not depend on j, i.e., each vector of the initial set is trans-

Manuscript received February 16, 1987, revised July 3, 1987. This work was supported in part by the United States Army through its European Research Office, and in part by the Italian Department of Education under a "[illegible] Grant." This paper was presented in part at the IEEE International Symposium on Information Theory, Brighton, England, June 1985.

E. Biglieri was with the Dipartimento di Elettronica, Politecnico di Torino, Torino, Italy. He is now with the Department of Electrical Engineering, 6731 Boelter Hall, University of California, Los Angeles, CA 90024-1600.

M. Elia is with the Dipartimento di Elettronica, Politecnico di Torino, Corso Duca degli Abruzzi 24, I-10129 Torino, Italy.

IEEE Log Number 8822479.

0018-9448/88/0700-0803$01.00

formed by G into the same number of distinct vectors. A regular GGA is called *strongly regular* if each set GX_j contains exactly L distinct vectors.

The following result follows directly from the definitions.

Proposition 1: The number M of vectors in a regular GGA is a multiple of K. If GGA is strongly regular, then $M = KL$.

Next we exhibit four examples of these alphabets. Notice that for $K = 1$ every GGA is regular, but not necessarily strongly regular [7], [16].

Alphabet 1 (Asymmetric M-PSK: Two Dimensions, One Energy Level): Choose an initial vector $X = (\cos\vartheta, \sin\vartheta)$, ϑ a given constant, an integer $M = 2^\mu$, and consider the group of 2×2 orthogonal matrices of the form $R^i T^j$, $i = 0, 1, \cdots, M-1$, $j = 1, 2$, where

$$R = \begin{bmatrix} \cos(2\pi/M) & \sin(2\pi/M) \\ -\sin(2\pi/M) & \cos(2\pi/M) \end{bmatrix}$$

and

$$T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

It is seen that the effect of R on a two-dimensional vector is to rotate it by an angle $2\pi/M$, and the effect of T is to exchange its components. This group has $2M$ elements and gives rise to a separable alphabet of M or $2M$ vectors, according to the choice of the initial vector. Notice that the alphabet is strongly regular only when it has $2M$ elements (asymmetric M-PSK [13], [14]).

Alphabet 2 (Four Dimensions, One Energy Level): Consider the group of matrices which act on a four-dimensional initial vector by permuting its components and replacing them with their negatives. This group has $4!2^4$ elements. If the initial vector is $X_1 = (a, a, a, 0)$, $a = 1/\sqrt{3}$, the resulting (separable) alphabet has $M = 32$ distinct unit-energy vectors (see Fig. 1).

A				B				C				D			
a	a	a	0	a	a	0	a	a	0	a	a	0	a	a	a
0	-a	a	a	-a	a	-a	0	a	a	0	-a	-a	0	-a	a
a	0	-a	a	0	-a	-a	a	-a	a	a	0	-a	a	0	-a
a	-a	0	-a	-a	0	a	a	0	-a	a	-a	a	a	-a	0
-a	-a	-a	0	-a	-a	0	-a	-a	0	-a	-a	0	-a	-a	-a
0	a	-a	-a	a	-a	a	0	-a	-a	0	a	a	0	a	-a
-a	0	a	-a	0	a	a	-a	a	-a	-a	0	a	-a	0	a
-a	a	0	a	a	0	-a	-a	0	a	-a	a	-a	-a	a	0

Fig. 1. Alphabet 2 and its fair partition.

Alphabet 3 (Two Dimensions, Three Energy Levels): Our third example is shown in Fig. 2. Points 1, 2, 3, and 4 denote the four vectors in the initial set. The matrices generating the code are those associated to plane rotations by multiples of $\pi/2$. The resulting (strongly regular, separable) alphabet is the conventional 16-QAM.

Alphabet 4 (Four Dimensions, Two Energy Levels): This alphabet which has two energy levels, $K = 4$, and $M = 128$,

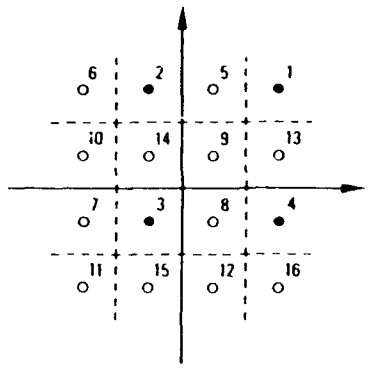


Fig. 2. Alphabet 3 and its fair partition.

is obtained from the initial set of vectors

$$\begin{matrix} c & c & c & 0 \\ -b & c & c & 0 \\ c & -b & c & 0 \\ c & c & -b & 0 \end{matrix}$$

with $c = 0.389$ and $b = 0.939$. If we apply to this initial set the same matrix group which generates Alphabet 2, we get a separable alphabet with 128 vectors (see Fig. 3). Among them, 32 have energy $3c^2$, and 96 have energy $b^2 + 2c^2$. The average energy is 1.

We consider now some distance properties of the elements of a GGA. Choose a partition of it into m subsets $Z_1, Z_2, \cdots, Z_m$. For each subset Z_i, we can define the *intradistance set* as the set of all the Euclidean distances among pairs of vectors in Z_i. For any pair of distinct subsets Z_i, Z_j, we define their *interdistance set* as the set of all the Euclidean distances between a vector in Z_i and a vector in Z_j.

Definition 4: The partition of a separable GGA into m subsets $Z_1, \cdots, Z_m$ is called *fair* if all the subsets are distinct, include the same number of vectors, and their intradistance sets are equal.

We shall now exhibit a constructive method to generate fair partitions of a GGA. Consider the generating group G of the GGA, one of its subgroups, say H, and the partition of G into left cosets of H. We have the following result.

Theorem 1: If the left cosets of the subgroup H are applied to the initial set of a strongly regular GGA, this procedure results into a fair partition of the GGA. Under the same hypotheses, if H is a normal subgroup, then left and right cosets give rise to the same fair partition.

Proof: Let S denote an element of G_g not belonging to H, and SH the corresponding left coset. If X_i, X_j are two (not necessarily distinct) vectors of the initial set, and S_h, S_k are two elements of H, the intradistance set associated with the coset SH includes the quantities

$$d_{ij}^2(S, S_h, S_k) \triangleq \|SS_hX_j - SS_kX_i\|^2$$

as S_h, S_k run through H, and X_i, X_j run through the initial

A	B	C	D	E	F	G	H
c c c 0	c c 0 c	c 0 c c	0 c c c	-c c c 0	c c 0-c	c 0-c c	0-c c c
-b c c 0	c c 0-b	c 0-b c	0-b c c	b c c 0	c 0 b	c 0 b c	0 b c c
c-b c 0	-b c 0 c	c 0 c-b	0 c-b c	-c-b c 0	-b c 0-c	c 0-c-b	0-c-b c
c c-b 0	c-b 0 c	-b 0 c c	0 c c b	-c c-b 0	c-b 0-c	-b 0-c c	0-c c-b
-c-c-c 0	-c-c 0-c	-c 0-c-c	0-c-c-c	c-c-c 0	-c-c 0 c	-c 0 c-c	0 c-c-c
b-c-c 0	-c-c 0 b	-c 0 b-c	0 b-c-c	-b-c-c 0	-c-c 0-b	-c 0-b-c	0-b-c-c
-c b-c 0	b-c 0-c	-c 0-c b	0-c b-c	c b c 0	b-c 0 c	-c 0 c b	0 c b-c
-c-c b 0	-c b 0-c	b 0-c-c	0-c-c b	c-c b 0	-c b 0 c	b 0 c-c	0 c-c b
I	J	K	L	M	N	O	P
c-c c 0	-c c 0 c	c 0 c-c	0 c-c c	c c-c 0	c-c 0 c	-c 0 c c	0 c c-c
-b-c c 0	-c c 0-b	c 0-b-c	0-b-c c	-b c-c 0	c-c 0-b	-c 0-b c	0-b c-c
c b c 0	b c 0 c	c 0 c b	0 c b c	c-b c 0	-b-c 0 c	-c 0 c-b	0 c-b-c
c-c-b 0	-c-b 0 c	-b 0 c-c	0 c-c-b	c c b 0	c b 0 c	b 0 c c	0 c c b
-c c-c 0	c-c 0-c	-c 0-c c	0-c c-c	-c-c c 0	-c c 0-c	c 0-c-c	0-c-c c
b c-c 0	c-c 0 b	-c 0 b c	0 b c-c	b-c c 0	-c c 0 b	c 0 b-c	0 b-c c
-c-b-c 0	-b-c 0-c	-c 0-c-b	0-c-b-c	-c b c 0	b c 0-c	c 0-c b	0-c b c
-c c b 0	c b 0-c	b 0-c c	0-c c b	-c-b-c 0	-c-b 0-c	-b 0-c-c	0-c-c-b

Fig. 3. Alphabet 4 and its fair partition.

vector set. We have

$$d_{ij}^2(S, S_h, S_k) = \|X_j\|^2 + \|X_i\|^2 - 2X_j^T S_h^T S^T S S_k X_i$$
$$= \|X_j\|^2 + \|X_i\|^2 - 2X_j^T S_h^T S_k X_i$$

where the superscript T denotes transpose.

As the right side of the last equation does not depend on S, we have shown that the intradistance set associated with the left cosets of H are independent of the coset. Moreover, if H is normal, then right cosets and left cosets give rise to the same fair partition: in fact, normality implies that for every S

$$SH = HS.$$

What makes Theorem 1 work is the fact that the orthogonal matrices form a group of isometries. Hence a more abstract formulation is possible, extending to nonfinite groups. As pointed out by the editor, lattices and sublattices equipped with isometric transformations (translations) fit this more general approach. However, for our presentation we choose the framework that was fruitfully used for the description of "group codes for the Gaussian channel," and that was based on finite groups of orthogonal matrices [7] (see also [8]).

The condition of strong regularity of the GGA can be removed, but in this case it may happen that different cosets generate the same element of the partition. Hence some of the cosets must be removed from consideration. Moreover, notice that if H is a normal subgroup of G, then we do not need to distinguish between left or right coset partitions. On the contrary, if H is not normal, the partitions obtained from right cosets may not be fair, as shown by the following counterexample.

Example 1: Let us consider the four-dimensional alphabet generated by the action of the natural matrix representation of the permutation group S_4 on the initial vector $(-3d/2, -d/2, d/2, 3d/2)$, d a constant. Let us consider the partition induced by the subgroup H of the matrices leaving invariant the fourth component of the initial vector. This subgroup is isomorphic to S_3. The left and right coset partitions associated with H are shown in Table I. It can be seen that the partition associated with right cosets is not fair because its intradistance sets are not equal.

In some cases, we are interested in further partitioning every element Z_i into the same number of subsets. We are led to the concept of a *chain partition*. This concept is also found in the work of Ungerboeck [10] and Ginzburg [8].

Definition 5: The chain partition of a separable GGA is called *fair* if any two elements of the partition at the same level of the chain include the same number of vectors and have equal intradistance sets.

For fair chain partitions we have the following theorem, whose proof is straightforward and will be omitted.

Theorem 2: Consider a strongly regular GGA and a chain of subgroups of its generating group G, that is,

$$H_1 \subset H_2 \subset H_3 \subset \cdots \subset H_s = G.$$

Use H_{s-1} and its left cosets to generate a partition of GGA. Then use H_{s-1} and its left cosets in H_s to further partition all the sets of the previous partition. Repeat the procedure with H_{s-2}, and so on, until H_1 and its left

TABLE I
LEFT AND RIGHT COSET PARTITIONS OF A GGA

Left Coset Partition	Right Coset Partition
$(-3d/2, -d/2, d/2, 3d/2)$	$(-3d/2, -d/2, d/2, 3d/2)$
$(-d/2, -3d/2, d/2, 3d/2)$	$(-d/2, -3d/2, d/2, 3d/2)$
$(d/2, -d/2, -3d/2, 3d/2)$	$(d/2, -d/2, -3d/2, 3d/2)$
$(-3d/2, d/2, -d/2, 3d/2)$	$(-3d/2, d/2, -d/2, 3d/2)$
$(-d/2, d/2, -3d/2, 3d/2)$	$(-d/2, d/2, -3d/2, 3d/2)$
$(d/2, -3d/2, -d/2, 3d/2)$	$(d/2, -3d/2, -d/2, 3d/2)$
$(3d/2, -3d/2, -d/2, d/2)$	$(3d/2, -d/2, d/2, -3d/2)$
$(3d/2, -d/2, -3d/2, d/2)$	$(-d/2, 3d/2, d/2, -3d/2)$
$(3d/2, d/2, -d/2, -3d/2)$	$(d/2, -d/2, 3d/2, -3d/2)$
$(3d/2, -3d/2, d/2, -d/2)$	$(3d/2, d/2, -d/2, -3d/2)$
$(3d/2, -d/2, d/2, -3d/2)$	$(-d/2, d/2, 3d/2, -3d/2)$
$(3d/2, d/2, 3d/2, -d/2)$	$(d/2, 3d/2, -d/2, -3d/2)$
$(-3d/2, 3d/2, -d/2, d/2)$	$(-3d/2, 3d/2, d/2, -d/2)$
$(-d/2, 3d/2, -3d/2, d/2)$	$(-d/2, -3d/2, d/2, -d/2)$
$(d/2, 3d/2, -d/2, -3d/2)$	$(d/2, 3d/2, -3d/2, -d/2)$
$(-3d/2, 3d/2, d/2, -d/2)$	$(-3d/2, d/2, 3d/2, -d/2)$
$(-d/2, 3d/2, d/2, -3d/2)$	$(3d/2, d/2, -3d/2, -d/2)$
$(d/2, 3d/2, -3d/2, -d/2)$	$(d/2, -3d/2, 3d/2, -d/2)$
$(-3d/2, -d/2, 3d/2, d/2)$	$(-3d/2, -d/2, 3d/2, d/2)$
$(-d/2, -3d/2, 3d/2, d/2)$	$(-d/2, -3d/2, 3d/2, d/2)$
$(d/2, -d/2, 3d/2, -3d/2)$	$(3d/2, -d/2, -3d/2, d/2)$
$(-3d/2, d/2, 3d/2, -d/2)$	$(-3d/2, 3d/2, -d/2, d/2)$
$(-d/2, d/2, 3d/2, -3d/2)$	$(-d/2, 3d/2, -3d/2, d/2)$
$(d/2, -3d/2, 3d/2, -d/2)$	$(3d/2, -3d/2, -d/2, d/2)$

cosets in $\boldsymbol{H}_2$ are used. The resulting chain partition of GGA is fair.

A theorem concerning the interdistance sets sheds some further light on the symmetry properties of GGA's.

Theorem 3: Let $\boldsymbol{H}$ be a normal subgroup of $\boldsymbol{G}$. The partition of a strongly regular GGA obtained by applying the left cosets of $\boldsymbol{H}$ to the initial set $\boldsymbol{X}$ has the following property. The interdistance set associated with any two cosets, say $S_1\boldsymbol{H}$ and $S_2\boldsymbol{H}$, is a function only of the coset $S_3\boldsymbol{H}$, where $S_3 = S_1^T S_2$, and not of S_1, S_2 separately.

Proof: Let S_1 and S_2 denote two coset leaders. If X_i, X_j are two (not necessarily distinct) vectors of the initial set $\boldsymbol{X}$, and S_h, S_k are two elements of $\boldsymbol{H}$, the distances among elements of the cosets $S_1\boldsymbol{H}$ and $S_2\boldsymbol{H}$ includes the quantities

$$d_{ij}(S_1, S_2, S_h, S_k) \triangleq \|S_1 S_h X_j - S_2 S_k X_i\|$$

as S_h, S_k run through $\boldsymbol{H}$ and X_i, X_j run through $\boldsymbol{X}$. We have

$$\begin{aligned} d_{ij}^2(S_1, S_2, S_h, S_k) &= \|X_j\|^2 + \|X_i\|^2 - 2X_j^T S_h^T S_1^T S_2 S_k X_i \\ &= \|X_j\|^2 + \|X_i\|^2 - 2X_j^T S_h^T S_3 S_k X_i. \end{aligned}$$

Finally, as $\boldsymbol{H}$ is a normal subgroup, we have

$$S_1\boldsymbol{H}S_2\boldsymbol{H} = S_1S_2\boldsymbol{H} = S_3\boldsymbol{H};$$

i.e., $S_3\boldsymbol{H}$ is another coset.

We now provide some examples of fair partitions of a GGA. Consider first the rotation group which generates Alphabet 3 (see Fig. 2) and its partition into the two cosets associated with the rotations $0, \pi$, and $\pi/2, -\pi/2$, respectively. The GGA is fairly partitioned into the two subalphabets $\{1,2,3,4,9,10,11,12\}$ and $\{5,6,7,8,13,14,15,16\}$.

Fig. 1 shows a fair partition of the Alphabet 2 in four subsets of eight vectors each. This partition is obtained as follows: denote by α the orthogonal matrix whose effect on a vector is to cyclically shift its components to the right by one position and to change sign to the second component. Then the set

$$\boldsymbol{H} = \{\alpha^0, \alpha^1, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7\}$$

is a cyclic normal subgroup of the group $\boldsymbol{G}$ generating the alphabet, and its cosets generate the fair partition.

A fair partition of Alphabet 4 into 16 subsets of eight vectors each stems from the subgroup $\{I, -I\}$, where I is the 4×4 identity matrix (see Fig. 3). A fair partition of Alphabet 1 is obtained by considering the two cosets of the subgroup $\{R_i\}_{i=0}^{M-1}$.

Definition 6: Let $\boldsymbol{R}$ be a left coset of $\boldsymbol{G}$ in the fair partition of a GGA and S_g an element of $\boldsymbol{G}$. We define the *distance profile* [15] associated with $\boldsymbol{R}$ and S_g as the polynomial in the indeterminate w:

$$F(w, S_g, R) \triangleq \sum_{d^2} a(d^2) w^{d^2}$$

where $a(d^2)$ is the number of elements of $\boldsymbol{RX}$ that have squared distance d^2 with respect to an element of the set $S_g\boldsymbol{RX}$. Note that a given element of $\boldsymbol{RX}$ may be accounted for more than once as it contributes with different squared distances with respect to different elements of the set $S_g\boldsymbol{RX}$. The sum of $a(d^2)$ equals the square of the cardinality of $\boldsymbol{RX}$.

Example 2: Consider $K = 1$, $X_1 = (1,0)^T$, and the group of plane rotations

$$S_i = \begin{bmatrix} \cos(i\pi/2) & \sin(i\pi/2) \\ -\sin(i\pi/2) & \cos(i\pi/2) \end{bmatrix}, \qquad i = 0,1,2,3.$$

The subgroup $\{S_0, S_2\}$ is normal. The distance profiles are summarized in Table II.

TABLE II
DISTANCE PROFILES FOR EXAMPLE 2

$\boldsymbol{R}$	S_g	$F(w, S_g, \boldsymbol{R})$
$\{S_0, S_2\}$	S_0	$2w^0 + 2w^4$
$\{S_0, S_2\}$	S_1	$4w^2$
$\{S_0, S_2\}$	S_2	$2w^0 + 2w^4$
$\{S_0, S_2\}$	S_3	$4w^2$
$\{S_1, S_3\}$	S_0	$2w^0 + 2w^4$
$\{S_1, S_3\}$	S_1	$4w^2$
$\{S_1, S_3\}$	S_2	$2w^0 + 2w^4$
$\{S_1, S_3\}$	S_3	$4w^2$

Definition 7: A fair partition of a GGA is called *homogeneous* if the set $\{F(w, S, \boldsymbol{R})\}_{S \in G}$ does not depend on $\boldsymbol{R}$. It is called *strongly homogeneous* if $F(w, S, \boldsymbol{R})$ does not depend on $\boldsymbol{R}$ for any S.

Theorem 4: If $\boldsymbol{G}$ is a commutative group, all the partitions generated by its subgroups are strongly homogeneous.

Proof: Let H be a subgroup of G: this is obviously normal so that the partition induced by H is fair. Let X_i, X_j be two elements of the initial set X, S an element of G, S_H an element of H. Then for any $S_g \in G$ the computation of $F(w, S_g, SH)$ involves enumerating the squared distances

$$\|SS_H X_i - S_g S S_{1H} H X_j\|^2 = \|SS_H X_i - SS_g S_{1H} X_j\|^2$$
$$= \|S_H X_i - S_g S_{1H} X_j\|^2$$

which do not depend on S and hence on the element of the fair partition.

Theorem 5: If H is a subgroup of G in a strongly regular GGA, the partition generated by the left cosets of H is homogeneous.

Proof: Let H be a subgroup of G. Then the partition induced by the left cosets of H is fair. Let X_i, X_j be two elements of the initial set, S an element of G, S_H and S_{1H} two elements of H. Then for any $S_g \in G$ the computation of $F(w, S_g, SH)$ involves enumerating the squared distances

$$\|SS_H X_i - S_g S S_{1H} X_j\|^2 = \|S_H X_i - S^T S_g S S_{1H} X_j\|^2$$
$$= \|S_H X_i - S_g' S_{1H} X_j\|^2$$

so that $F(w, S_g, SH) = F(w, S_g', SH)$, and as S_g runs through G so does $S_g' = S^T S_g S$. Thus the assertion is proved.

III. Multidimensional Coded Signals: Block Codes

We shall now see how the multidimensional alphabets described in the previous section can be used in conjunction with codes to further enhance their performance. In this section, we shall focus our attention on block codes, while the next section will be devoted to convolutional (trellis) codes.

Himai and Hirakawa [18] and Ginzburg [8] have described constructions which make it possible to design alphabets with an arbitrary signal distance and with a regular structure by employing algebraic properties of block codes. Fig. 4 shows Ginzburg's construction. The L block encoders $C_1, C_2, \cdots, C_L$ accept source symbols, and output L blocks $(q_{i1}, q_{i2}, \cdots, q_{iN})$, $i = 1, \cdots, L$, of N symbols each. The modulator f maps each L-tuple $(q_{j1}, \cdots, q_{jL})$, $j = 1, \cdots, N$, into the vector

$$x_j = f(q_{j1}, \cdots, q_{jL}), \qquad j = 1, \cdots, N$$

chosen from a set A of $M = M_1, \cdots, M_L$ elements. This mapping is obtained as follows. In the set A we define a system of L partitions such that each class of the lth partition includes M_l classes of the $(l-1)$th partition so that it will consist of $M(l) = M_1 M_2 \cdots M_l$ signals. By numbering the classes of the $(l-1)$th level occurring in a class of the lth level we can obtain a one-to-one mapping of the set of classes of the $(l-1)$th partition onto the set of integers $\{0, \cdots, M_l - 1\}$. Therefore, if q_{ij} are chosen in the set $\{0, \cdots, M_l - 1\}$, $l = 1, \cdots, L$, any L-tuple $(q_{j1}, \cdots, q_{jL})$ defines a unique value of the jth elementary signal $x_j = f(q_{j1}, \cdots, q_{jL})$ (see Fig. 5).

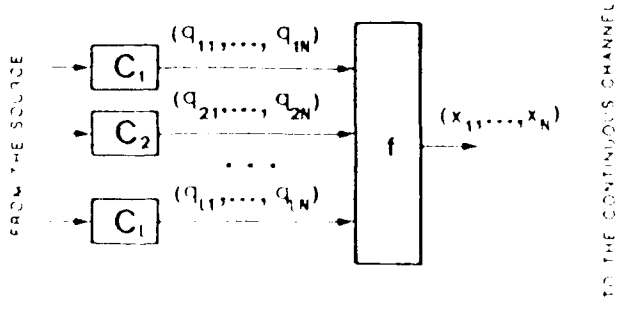


Fig. 4. Ginzburg construction.

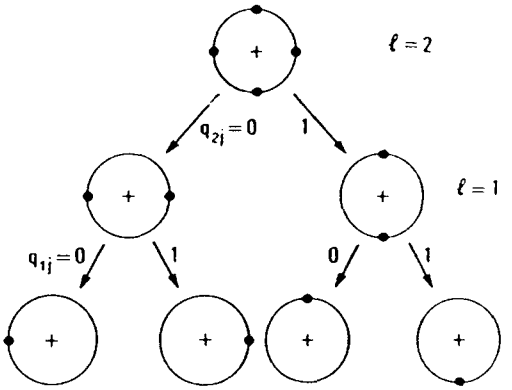


Fig. 5. Example of Ginzburg construction.

Ginzburg proved that the alphabet obtained in this way has a minimum squared Euclidean distance D^2 that satisfies

$$D^2 \geq \min_{1 \leq l \leq L} \left(\delta_l^2 d_l\right)$$

where $d_1, \cdots, d_L$ are the minimum Hamming distances of the L block codes $C_1, \cdots, C_L$, and δ_i^2 is the minimum squared Euclidean distance between the symbols in each subalphabet of the ith partition.

Consider now Ginzburg's constructions based on generalized group alphabets. By associating with each level the elements of a fair partition (the concept of a fair chain can be used here), all the subalphabets at a given level have the same minimum distance. From the fair partition of Alphabet 2 described before, we have $\delta_1^2 = 2/3$, and $\delta_2^2 = 2$. Thus using the $(N, k, 3)$ Hamming code on GF(4) [9, p. 193, 194] and the trivial $(N, N, 1)$ code on GF(8), with $N = (4^m - 1)/3$, $k = N - m$, $m \geq 1$, we have $D^2 \geq 2$. The resulting alphabet has a rate

$$R = [5(4^m - 1) - 6m] / [4(4^m - 1)]$$

and

$$D^2 \log_2 M = 10 - 12m/(4^m - 1).$$

For example, choosing $m = 2$ we get a rate $R = 1.05$ and $D^2 \log_2 M \geq 8.4$; with $m = 3$ we get $R = 1.18$ and $D^2 \log_2 M > 9.4$.

Using Alphabet 4 and the partition described, we have $\delta_1^2 = 2c^2$, and $\delta_2^2 = 8c^2$. The (18,15,4) extended Hamming code [19, p. 36] on GF(16) and the trivial (18,18,1) code on GF(8) can be employed, providing a squared minimum distance $D^2 \geq 1.211$. This alphabet yields $R = 1.583$ and $D^2 \log_2 M \geq 7.67$.

IV. Multidimensional Coded Signals: Trellis (Ungerboeck) Codes

We shall now see how an Ungerboeck code [10] can be designed using a multidimensional alphabet generated as described in Section II. Such codes can be specified as in [17]. Each coded symbol depends on $k + \nu$ source bits, namely, the block $\tau = (a_1, \cdots, a_k)$ of k bits generated by the source, plus ν bits preceding this block. The ν bits determine one of the $N = 2^\nu$ states of the encoder, say $\sigma = (a_{k+1}, \cdots, a_{k+\nu})$, $a_n = 0,1$. The encoder state for the next coded symbol is obtained by shifting the a_n k places to the right, dropping the right-most k bits and inserting on the left the most recent source bits. The encoded symbol x depends on τ and σ; we write

$$x = f(\tau, \sigma) \tag{4.1}$$

where x is an element of a GGA. This encoding procedure can be described using a trellis (Fig. 6 shows a section of such a trellis, obtained for $\nu = 2$).

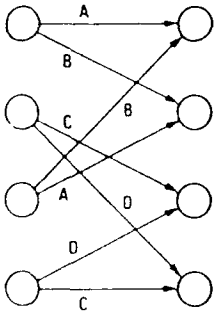


Fig. 6. Four-state trellis code for Alphabet 2.

We conjecture that a good code should show a good deal of symmetry to be reflected by the structure of the function f in (4.1), or, equivalently, by the assignment of symbols to the branches connecting any pair of nodes in the code trellis (for further details see, e.g., [10], [11]). This can be obtained in our framework by assigning to the branches associated with each node the set of symbols obtained from a fair partition of a GGA. This is equivalent to the procedure suggested in [10] and called "mapping by set partitioning"; thus our procedure can be viewed as a systematic way to achieve set partitioning.

The most widely used single parameter that specifies the performance of these codes on the additive white Gaussian noise channel is the *free Euclidean distance*. This can be computed using either a generating function approach or a modified bidirectional search algorithm [20], [21], or procedures based on Viterbi algorithm and described in [23], [24] (see also [25, pp. 561-564]). The generating function technique consists of enumerating all possible distances between sequences of symbols associated with paths in the trellis. In general [11], the generating function can be obtained as the transfer function of a state diagram regarded as a signal flow graph. The state diagram is defined over an expanded set of $N^2 = 2^{2\nu}$ states. For the special case of a trellis based upon a linear binary convolutional code and a strongly homogeneous partition of a GGA, the minimum distance can be computed from a generating function obtained as the transfer function of a state diagram including only $N = 2^\nu$ states. (See [15, theorem 3]).

We shall describe two examples of four-dimensional Ungerboeck codes. The first example originates from Alphabet 2. It has minimum distance $2a^2 = 0.66$. The fair partition described before gives four subsets of eight vectors each, with minimum intradistance $6a^2 = 2$. By choosing a four-state trellis code with the structure described in Fig. 6, we get a squared free distance $6a^2 = 2$. If this figure is compared to the minimum distance achieved by using two independent 4-PSK signals, which transmit the same amount of information over the same number of dimensions, we see that an energy saving of 3 dB is obtained.

Consider now Alphabet 4. It has a minimum square distance 0.3. The fair partition described gives 16 subalphabets of eight vectors each, with minimum intradistance 1.2. By using the four-state Ungerboeck code described in Fig. 7, the squared free distance obtained is $d_{\text{free}}^2 = 1.2$. By comparing this to the minimum distance obtained by using two independent 8/4-PSK signals, we see that an energy saving of about 4.3 dB is obtained.

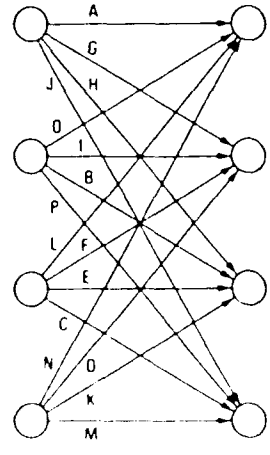


Fig. 7. Four-state trellis code for Alphabet 4.

V. Connections with Related Work

Recently, Calderbank and Sloane [22] have described a method of constructing multidimensional trellis codes where the alphabet is a finite subset of a lattice L with an equal number of points from each coset of a sublattice M of L. As pointed out by the editor, the symmetry and homogeneity properties of these alphabets are almost identical to those of GGA's. Subalphabet edge effects are the

reason why the correspondence is not quite exact. To see how the two methods are related, consider the partition of the integer lattice $L = Z^2$ generated by (1,0) and (0,1) into eight cosets of the sublattice M generated by (2,2) and (2, −2). The sublattice M consists of all vectors with norm divisible by 8. Let $T_{a,b}$ be the translation given by $T_{a,b}$: $(x, y) \to (x + a, y + b)$. Let $X = \{(0,1)\}$, and let $G = \langle T_{0,1}, T_{1,0} \rangle$ be the group of all translations. Define a chain of subgroups H_3, H_2, H_1, G, by $H_1 = \langle T_{1,1}, T_{1,-1} \rangle$, $H_2 = \langle T_{2,0}, T_{0,2} \rangle$, $H_3 = \langle T_{2,2}, T_{2,-2} \rangle$. This chain of subgroups corresponds to the eight-way partition of L.

VI. Conclusion

Ginzburg [8] described a method of dividing a signal alphabet into a chain of subsets via the action of a group of orthogonal matrices. We generalize this approach by introducing generalized group alphabets, and we consider the combination of these alphabets with block or trellis codes. Some actual designs show that consideration of GGA's may lead to transmission systems providing good performance with band-limited channels at the price of a relatively modest complexity.

References

[1] D. Slepian, "Bounds on communication," *Bell Syst. Tech. J.*, vol. 42, pp. 681–707, May 1963.

[2] S. G. Wilson, H. A. Sleeper, and N. K. Srinath, "Four-dimensional modulation and coding: An alternate to frequency reuse," in *Proc. ICC'84*, Amsterdam, The Netherlands, May 1984, pp. 919–923.

[3] S. Wilson and H. A. Sleeper, "Four-dimensional modulation and coding: An alternate to frequency reuse," Communications Systems Laboratory, Dept. of Electrical Engineering, University of Virginia, Charlottesville, Tech. Rep. UVA/528200/EE83/107, Sept. 1983.

[4] A. Gersho and V. Lawrence, "Multidimensional signal design for digital transmission over bandlimited channels," in *Proc. ICC'84*, Amsterdam, The Netherlands, May 1984, pp. 377–380.

[5] G. R. Welti and J. S. Lee, "Digital transmission with coherent four-dimensional modulation," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 497–502, July 1974.

[6] L. H. Zetterberg and H. Brandstrom, "Codes for combined phase and amplitude modulated signals in a four-dimensional space," *IEEE Trans. Commun.*, vol. COM-25, pp. 943–950, Sept. 1977.

[7] D. Slepian, "Group codes for the Gaussian channel," *Bell Syst. Tech. J.*, vol. 47, pp. 575–602, Apr. 1968.

[8] V. V. Ginzburg, "Mnogomernıye signaly dlya neprerynvnogo kanala," *Probl. Peredach. Inform.*, no. 1, pp. 28–46, Jan.–Mar. 1984 (in Russian). English translation: "Multidimensional signals for a continuous channel," *Probl. Inform. Transmission*, pp. 20–34, 1984.

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

[10] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55–67, Jan. 1982.

[11] E. Biglieri, "High-level modulation and coding for nonlinear satellite channels," *IEEE Trans. Commun.*, vol. COM-32, pp. 616–626, May 1984.

[12] I. Jacobs, "Comparison of M-ary modulation system," *Bell Syst. Tech. J.*, vol. 46, pp. 843–864, May–June 1967.

[13] D. Divsalar and J. H. Yuen, "Asymmetric MPSK for trellis codes," in *Proc. GLOBECOM '84*, Atlanta, GA, November 26–29, 1984, pp. 20.6.1–20.6.8.

[14] D. Divsalar and M. K. Simon, "Combined trellis coding with asymmetric modulations," in *Proc. GLOBECOM '86*, New Orleans, LA, Dec. 2–5, 1985, pp. 21.2.1–21.2.7.

[15] E. Zehavi and J. K. Wolf, "On the performance evaluation of trellis codes," *IEEE Trans. Inform. Theory*, vol. IT-33, no. 2, pp. 196–202, Mar. 1987.

[16] E. Biglieri and M. Elia, "On the existence of group codes for the Gaussian channel," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 399–402, May 1972.

[17] R. Calderbank and J. E. Mazo, "A new description of trellis codes," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 784–791, Nov. 1984.

[18] H. Himai and S. Hirakawa, "A new multilevel coding method using error-correcting codes," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 371–377, 1977.

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

[20] L. R. Bahl, C. D. Cullum, W. D. Frazer, and F. Jelinek, "An efficient algorithm for computing the free distance," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 437–439, May 1972.

[21] K. J. Larsen, "Comments on L. R. Bahl, C. D. Cullum, W. D. Frazer, and F. Jelinek: An efficient algorithm for computing the free distance," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 577–579, July 1973.

[22] A. R. Calderbank and N. J. A. Sloane, "New trellis codes based on lattices and cosets," *IEEE Trans. Inform. Theory*, vol. IT-33, no. 2, pp. 177–195, Mar. 1987.

[23] M. M. Mulligan and S. G. Wilson, "An improved algorithm for evaluating trellis phase code," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 6, pp. 846–851, Nov. 1984.

[24] R. C. P. Saxena, "Optimum encoding in finite state code modulation," Dept. of Electrical, Computer, and System Engineering, Rensselaer Polytechnic Institute, Troy, NY, Rep. TR83-2, 1983.

[25] S. Benedetto, E. Biglieri, and V. Castellani, *Digital Transmission Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

Appendix B

Analysis and Compensation of Nonlinearities in Digital Transmission Systems

ENZO BIGLIERI, SENIOR MEMBER, IEEE, SERGIO BARBERIS, AND MAURIZIO CATENA

Abstract—We consider the compensation of channel nonlinearities in digital radio communication systems. A discrete system with memory, *inserted between* the source and the modulator, is designed with the aim of providing an equivalent channel with a distortionless linear part and no nonlinearities up to a given order. This design is based on a Volterra series model of the channel, and on the theory of *p*th-order inverse systems.

Since the compensator design is based on a mathematical model of the channel, the problem of model identification is considered. A modeling technique is described, based on computer simulation and application of orthogonal Volterra series. Several examples show the performance of this class of compensators.

I. Introduction and Motivation of the Work

As THE demand for RF spectrum increases, high-speed data transmission over radio channels is likely to benefit from consideration of high-capacity modulation formats, such as multilevel quadrature amplitude modulation (QAM). Their application has been slowed down by the presence of amplitude (AM/AM) and phase (AM/PM) nonlinearities present in radio-frequency (RF) power amplifiers driven at or near saturation for better efficiency. Actually, the nonlinear distortions introduced by these amplifiers make the standard channel model, i.e., the additive white Gaussian noise channel, far from realistic, and hence system designs based on it far from optimum.

On the additive white Gaussian noise channel, it is well known that QAM signals with a rectangular constellation provide a better bit-error rate (BER) performance than phase-shift keying (PSK) with an equal number of points. However, this situation seems to be reversed when nonlinear distortions are present in the channel. To cope with this problem, the simplest approach is to back off from the saturation point of the amplifier characteristics, in order to have the signal amplitude fluctuations to involve a region in which AM/AM characteristics are close to linear, and AM/PM characteristics close to a constant. However, this procedure causes a loss of power efficiency that might be unacceptable. In fact, in several instances it was verified (see, for example, [20]) that, as the number of energy levels in the signal constellation increases, the TWT working point should be backed off more to compensate for the nonlinear behavior of the amplifier. In this situation, it may occur that the beneficial effect of the increase in linearity is offset by the corresponding decrease of amplifier's output power. As a result, PSK (which has only one energy level) may perform better than QAM (which has more) [20]. This is why more sophisticated solutions are called for.

From the above discussion, it appears rather natural to investigate two-dimensional signal constellations that outperform PSK, and yet do not suffer excessive degradations due to channel nonlinearities. This paper is devoted to this problem, through an approach that combines the choice of the modulation format and the compensation of channel nonlinearities.

The channel model on which our analysis will be based is time-discrete. We assume for simplicity that the modulated signal is sent through a nonlinear system with memory before being affected by additive white Gaussian noise at its output. In other words, the discrete channel consists of two separate parts: a noiseless deterministic part, and a noise adder. Traditionally, there are two philosophies intended to cope with the problem of channel nonlinearities. One consists of accepting the channel as is, without trying to do anything to modify its behavior, and to design the receiver so as to minimize the joint effects of intersymbol interference, nonlinearities, and noise. The most effective nonlinear signal processing technique based on this approach is maximum likelihood sequence estimation (MLSE), to be performed by the Viterbi algorithm [12], [14], [17, ch. 10]. Unfortunately, this technique requires a processing complexity which may make it unsuitable for implementation at very high data rates. For this reason, suboptimum receiver schemes are attractive: among them, we can recall nonlinear equalization schemes [3], [4], nonlinear cancellers [1], [2], optimum linear equalizers [15], and optimum linear receiving filters [16]. It must be kept in mind, however, that a fundamental limit to the performance of any of these receivers (and, more generally, of any conceivable receiver, either linear or nonlinear) depends on the minimum Euclidean distance between the signals observed at the output of the noiseless (i.e., deterministic) part of the channel [18]. Stated in words, this limitation is due to the fact that the signal to be processed by the receiver is af-

0733-8716/88/0100-0031$01.00 © 1988 IEEE

fected by noise; hence, any attempt to compensate for the channel distortion by introducing a sort of "inverse distortion" will enhance the noise. For this reason, it appears logical to investigate solutions based on the compensation of the nonlinearity *before* noise addition. This procedure should make the channel look as similar as possible to a Gaussian channel.

If this approach is chosen, there are several factors and constraints that should be kept in mind. One of them is, of course, the ultimate performance that the nonlinearity compensation system can achieve. The second one is the ease of design, the implementation complexity, and the cost. The third is that the compensator itself may expand the signal bandwidth, in spite of the fact that out-of-band emission must be kept under control [7]. In fact, while a predistorter reduces out-of-band emission after the amplifier, it may increase it before the amplifier. This can be a problem, for example, in a satellite system with the predistorter located in the ground station to compensate for the on-board nonlinearity. Finally, in certain cases provision must be made for *adaptive* compensation: in fact, when a constellation with a large number of points is used by the modulator, even variations in amplifier characteristics caused by temperature changes, dc power variations, and component aging can degrade the system performance [9]. Both analog and digital predistorters can in principle be considered, however, besides being more complex and expensive, and less flexible, the analog predistorters seem to perform worse than their digital counterparts [5], [6]. Hence, consistent with our assumption of a discrete channel model, we shall consider digital predistortion.

In this paper we consider digital predistortion of a channel, i.e., the design of a device to be inserted at the transmitter front-end of the transmission system and whose aim is to compensate for the unwanted effects of the nonlinear channel. This design will be based on the concept of *p*th-order inverse of a nonlinear system. The theory of *p*th-order inverses was developed by Schetzen (see [23, ch. 7]) and is applied here to discrete, bandpass systems. Since the first step in predistorter design is modeling the nonlinearity, we argue that the best way of doing this is to base it on computer identification of the simulated system. This interaction between computer simulation and analysis will provide us with an orthogonal Volterra series model, whose parameters are used to design the compensator.

Section II will present the concepts of a predistorter with memory and of a *p*th-order inverse. Design of compensators based on *p*th-order inverses is introduced in Section III. Channel modeling via the orthogonal Volterra series is the subject of Section IV, while Section V covers some examples of application.

II. Predistortion for the Nonlinear Compensation

A. Memoryless Predistortion

In this section we shall consider the action of a predistorter, i.e., a device placed in front of the channel whose aim is to compensate for the unwanted effects of channel nonlinearities. Assume first that the channel has no memory (i.e., no bandwidth-limiting components exist) and consider the effect of a predistorter placed just before the nonlinear channel. In this situation, which we shall refer to as *memoryless predistortion*, the compensator acts by skewing the signal constellation in such a way that, when passed through the nonlinear device, it will resume the original shape (e.g., a rectangular 16 QAM structure). In other words, the compensator task is to invert the discrete transmission channel. This operation does not modify the spectrum, and hence the bandwidth occupancy, of the transmitted signal, but of course its effectiveness is critically dependent on the assumption that the channel has no memory.

B. Predistortion with Memory. The *p*th-Order Inverse

Consider instead, the more realistic assumption of a channel with memory. In this situation, the compensator is faced with a far more difficult task, the inversion of a nonlinear system with memory. Now, not all nonlinear systems possess an inverse. Also, many systems can be inverted only for a restricted range of input amplitudes [23, p. 123 ff.]. However, it is always possible to define a *pth-order inverse*, for which the input amplitude range is not restricted [23, ch. 7].

Our use of the *p*th-order inverse theory will be based on a Volterra-series model of the discrete nonlinear channel (see [17] and the references therein). This model provides an exceedingly general characterization of nonlinear systems with memory based on the so-called Volterra kernels, a set of parameters which can be thought of as the extension of the nonlinear case of the concept of impulse response of a linear channel. Given a nonlinear system H, its *p*th-order inverse is one that, when cascaded to H, results in a system in which the first-order Volterra kernel is a unit impulse, and the second through the *p*th-order Volterra kernels are zero [23]. In other words, if the *p*th-order nonlinear inverse channel is synthesized at the transmitter's front end, the compensated transmission channel will exhibit no linear distortion, and no nonlinear distortion up to order p. Obviously, the performance of the *p*th-order compensated channel will depend on the effect of the residual distortions.

C. Memoryless Predistortion Versus Predistortion with Memory

Before proceeding further with an analytical description of the compensation based on *p*th order channel inversion, it is convenient to stop the discussion for a while, and provide an interpretation of the two types of predistorters described in the previous subsection. Memoryless predistortion is the operation of changing the (source) symbols a_n into the (channel) symbols $b_n = g(a_n)$, where $g(\cdot)$ is a suitable complex function. If these modified symbols are viewed as a new signal set entering the channel (and matched to it), we may think of the compensator as being incorporated in the modulator. In conclusion, the design of a predistorter for a channel without memory is equivalent to the design of a new modulation scheme.

Consider then a predistorter with memory. Its operation consists of transforming the source symbols a_n into channel symbols b_n whose values depend not only on a_n but also on L previous symbols. Thus,

$$b_n = \Gamma(a_n, a_{n-1}, \cdots, a_{n-L}). \tag{2.1}$$

If we define the *state of the compensator* at time n, and we denote it by σ_n

$$\sigma_n = (a_{n-1}, \cdots, a_{n-L}) \tag{2.2}$$

we can also write

$$b_n = \Gamma(a_n, \sigma_n) \tag{2.3}$$

which shows explicitly the dependence of the channel symbols b_n on the state of the compensator. This "sliding block" representation of the compensator operation shows that the compensator itself is equivalent to a trellis encoder. (This equivalence was first proved by Calderbank and Mazo [24].) In conclusion, we can think of the design of a predistorter with memory as of the choice of a trellis code, made in order to compensate for the channel nonlinearity.

III. Compensation Based on pth-Order Channel Inversion

We start our discussion by considering the cascade of two nonlinear systems (for motivation's sake, the reader can view one of the two systems as the compensator, and the other one as the channel to be compensated). We shall base our treatment of the subject on Volterra series representations of bandpass systems [17, ch. 10], and we shall use, for notational simplicity, tensor notations, as suggested in [22]. These notations imply that any subscript occurring twice in the same term is to be summed over the appropriate range of discrete time. Thus, for example, we write $x_i y_i$ instead of $x_1 y_1 + x_2 y_2 + \cdots$.

A. Cascading Bandpass Nonlinear Systems

Subject to certain regularity conditions, a bandpass nonlinear system can be described by the input–output relationship

$$\begin{aligned} y_n = {} & \sum_{a=-\infty}^{\infty} h^{(1)}_{n;a} x_a + \sum_{a=-\infty}^{\infty}\sum_{b=-\infty}^{\infty}\sum_{c=-\infty}^{\infty} \\ & \cdot h^{(3)}_{n;a,b,c} x_a x_b x_c^* \\ & + \sum_{a=-\infty}^{\infty}\sum_{b=-\infty}^{\infty}\sum_{c=-\infty}^{\infty}\sum_{d=-\infty}^{\infty}\sum_{e=-\infty}^{\infty} \\ & h^{(5)}_{n;a,b,c,d,e} x_a x_b x_c x_d^* x_e^* + \cdots \\ & h^{(1)}_{n;a} x_a + h^{(3)}_{n;a,b,c} x_a x_b x_c^* \\ & + h^{(5)}_{n;a,b,c,d,e} x_a x_b x_c x_d^* x_e^* + \cdots \end{aligned} \tag{3.4}$$

From (3.4), it is seen that the system is characterized by the *Volterra kernels* $h^{(1)}_{n;a}, h^{(3)}_{n;a,b,c}, \cdots$. Notice that only odd order polynomials appear: this is due to the bandpass nature of the nonlinearity.

Consider now two bandpass nonlinear systems. Let the first (the compensator) be characterized by Volterra kernels f, the second (the channel) by Volterra kernels g, and denote by h the kernels of the system resulting from the cascade of the two. The first-, third-, and fifth-order h-kernels are explicitly given by

$$h^{(1)}_{n;a} = g^{(1)}_{n;v} f^{(1)}_{v;a} \tag{3.5}$$

$$\begin{aligned} h^{(3)}_{n;a,b,c} = {} & g^{(1)}_{n;v} f^{(3)}_{v;a,b,c} \\ & + g^{(3)}_{n;v,w,z} f^{(1)}_{v;a} f^{(1)}_{w;b} f^{(1)*}_{z;c} \end{aligned} \tag{3.6}$$

and

$$\begin{aligned} & h^{(5)}_{n;a,b,c,d,e} \\ & = g^{(1)}_{n;v} f^{(5)}_{v;a,b,c,d,e} \\ & \quad + g^{(3)}_{n;v,w,z} f^{(1)}_{v;a} f^{(1)}_{w;b} f^{(3)*}_{z;c,d,e} \\ & \quad + g^{(3)}_{n;v,w,z} f^{(1)}_{v;a} f^{(3)}_{w;b,c,d} f^{(1)*}_{z;e} \\ & \quad + g^{(3)}_{n;v,w,z} f^{(3)}_{v;a,b,c} f^{(1)}_{w;d} f^{(1)*}_{z;e} \\ & \quad + g^{(5)}_{n;v,w,z,y,u} f^{(1)}_{v;a} f^{(1)}_{w;b} f^{(1)}_{z;c} f^{(1)*}_{y;d} f^{(1)*}_{u;e}. \end{aligned} \tag{3.7}$$

It can be observed that (3.5) expresses a relationship between first-order kernels which is nothing but the discrete convolution of impulse responses of linear systems.

B. Volterra Coefficients of pth-Order Compensator

Consider now pth-order compensation. Under the assumption that the linear part of system f, i.e., the linear functional determined by the first-order kernel of f, is invertible, it is possible to find a system g such that its cascade with f gives a system with no linear distortion, i.e.,

$$\begin{aligned} g^{(1)}_{n;v} f^{(1)}_{v;a} &= f^{(1)}_{n;v} g^{(1)}_{v;a} \\ &= \delta_{n,a} \begin{cases} = 1 & n = a \\ = 0 & \text{elsewhere.} \end{cases} \end{aligned} \tag{3.8}$$

This choice provides the first-order compensator. Equation (3.8) expresses nothing but the Nyquist criterion for the absence of intersymbol interference in the overall channel. In appearance, this sounds like a rather pleasant result, as it shows that even when dealing with a nonlinear channel the linear part must be designed (at least, if the "pth-order criterion" is accepted) to be Nyquist's. In the following, we shall see how the concept of "linear part of a channel" must be correctly interpreted.

The third-order compensator is obtained by choosing $g^{(3)}$ so as to have $h^{(3)} = 0$; by taking the discrete convolution of both sides of (3.6) with $g^{(3)} g^{(1)} g^{(1)*}$, and recalling (3.8), we get

$$g^{(3)}_{n;a,b,c} = g^{(1)}_{n;v} f^{(3)}_{v;a,b,c} - g^{(1)}_{n;a} g^{(1)}_{b;b} g^{(1)*}_{z;c}. \tag{3.9}$$

The fifth-order compensator is obtained by choosing $h^{(5)}$ so as to have $h^{(5)} = 0$, by taking the discrete convolution of both sides of (3.7) with $g^{(1)} g^{(1)} g^{(1)} g^{(1)*} g^{(1)*}$, and recalling (3.8), we get the required $g^{(5)}$.

Before going further, let us consider a special situation (which is admittedly rather simplistic, but gives rise to considerations that might be interesting). Assume that the channel nonlinearity is the cascade of a linear system L and a memoryless device D. The pth-order compensator for this channel can be easily computed, providing a result which matches intuition. In fact, it is the cascade of a linear filter, the inverse of L (say, L^{-1}), preceded by a nonlinear memoryless device, the pth-order inverse of D. Notice that the cascade L^{-1} and L gives rise to a Nyquist filter. This result shows that one way to compensate for the channel nonlinearity in this case consists of removing the channel memory and compensating for the resulting memoryless nonlinearity by memoryless predistortion.

A more realistic model, suitable as an approximation to a number of single-channel digital satellite communication systems, assumes that the linear part of the channel has already been compensated by a suitable combination of channel filtering and linear equalization at the receiver's front end. In this situation some simplifications arise. In particular we get, for the first- and third-order compensators

$$g^{(1)}_{n,a} = \delta_{n,a}$$
$$g^{(3)}_{n,a,b,c} = -f^{(3)}_{n,a,b,c}. \tag{3.10}$$

C. The Effect of Compensation on Power Spectrum

We consider now the effect of a pth-order compensator on the signal power density spectrum. The continuous-time signal at the modulator output can be given the form

$$x(t) = \sum_{n=-\infty}^{\infty} b_n s(t - nT) \tag{3.11}$$

where (b_n) is the sequence of channel symbols, T is the symbol period (equivalently, T^{-1} is the baud rate), and $s(t)$ is the basic waveform used by the modulator. The power density spectrum of signal (3.11) is given by (see [17, p. 33])

$$G(f) = \frac{1}{T}|S(f)|^2 \left\{ \sum_{n=-\infty}^{\infty} \beta_n \exp(-j2\pi fnT) \right\} \tag{3.12}$$

where β_n is the autocorrelation of the symbol sequence at the compensator output and $S(f)$ is the Fourier transform of $s(t)$. It is easily recognized that the brackets on the RHS of (3.12) contain the discrete Fourier transform of the sequence (β_n), i.e., the power density spectrum of the sequence at the compensator output. This is a periodic function of f with period $1/T$.

From (3.12), we see that the spectrum shaping effect of the compensator can be analyzed by evaluating the autocorrelation sequence (β_n). For example, a linear compensator responding to the source symbol sequence (a_n), $E\{|a_n|^2\} = 1$, with the sequence $(a_n + Aa_{n-1})$, A a real constant, will cause a spectral shaping $(1 + A^2) + 2A\cos(2\pi fT)$. A fact that might be unexpected *a priori* is that the *nonlinear* terms of the compensator may be irrelevant in shaping the spectrum. Consider, as an example, the compensator output $a_n + Aa_{n-1} + Ba_n a_{n-1} a^*_{n-2}$. By direct calculation, it can be seen that $\beta_0 = (1 + A^2 + B^2)$ and $\beta_{-1} = \beta_1 = A$, while $\beta_i = 0$ for $|i| \geq 2$. Hence, the third-order nonlinearity has, for $A^2 >> B^2$ (as is the case when relatively mild nonlinearities must be compensated), very little effect.

D. Computing the Linear Part of the Compensator

The computation of the linear part of the compensator, i.e., of the kernels $g^{(1)}$ that solve (3.8), deserves some further attention. By rewriting explicitly (3.8), we have

$$\sum_n f^{(1)}_{k-n} g^{(1)}_n = \delta_{k,0} \tag{3.13}$$

where δ denotes the Kronecker symbol. Since we are interested in a finite-complexity compensator, we consider a (perhaps approximate) solution of (3.13) which includes just a finite number of terms in the summation. Thus, our problem is equivalent to the design of a zero-forcing equalizer of finite length. Two technical assumptions are necessary here, namely, that there exists only a finite number of nonzero $f^{(1)}$-kernels, and that the polynomial whose coefficients are these kernels has no root with unit magnitude. Under these conditions, a solution exists for the kernels $g^{(1)}$ with values that decrease in magnitude away from a "center kernel." The procedure for computing these kernels, which requires finding the roots of a polynomial and the solution of a set of linear equations, can be found in [27].

IV. The Role of Channel Modeling—Orthogonal Volterra Series

From our preceding discussion it is seen that the compensator design is based on a Volterra-series model for the nonlinear transmission channel. Thus, the availability of such a model is crucial. As, apart from some very simple cases, analytical evaluation of Volterra coefficients is not feasible, computational techniques should be used. Basically, two approaches are available, which we shall refer to as "block modeling" and "global identification."

Consider first block modeling. It is based on a model of the channel as a cascade of linear, time-invariant filters and bandpass nonlinear devices whose input–output relationships are given in the form of a Taylor series. In this case, the Volterra kernels are evaluated by combining the input–output relationships of the building blocks that form the channel (see, for example, [19]). Although this approach is apparently simple and straightforward, particularly when the channel itself is composed of a reduced number of blocks, in its application some care must be exercised by taking two important points into consideration. First of all, in many cases the number of nonzero Volterra coefficients is so large that the number of computations involved in evaluating higher order coefficients

may be impractical. The second one is more subtle. Perhaps the most important fact to be kept in mind when considering the identification of the channel is that nonlinear systems behave differently for different input signals. To understand the consequences of this statement, consider a simple example. Assume we are dealing with a channel responding to the input sequence x_n with the sequence $y_n = \alpha x_n + \beta x_n^3$, and assume that x_n can take only the values $+1$ or -1. Under these conditions, as $x_n = x_n^3$, the system behaves as *linear*, with input–output relationship $y_n = (\alpha + \beta) x_n$. On the other hand, if the input sequence can take values -3, -1, 1, and 3, the system really behaves as nonlinear. Hence, we realize that in a Volterra-series model each one of the nonlinear terms affects the transmitted sequence differently if different modulation formats are used. As an example, the third-order Volterra kernel $h_{011}^{(3)}$ has a different behavior on PSK and QAM signals. In fact, this kernel multiplies a term

$$b_n b_{n-1} b_{n-1}^*.$$

For PSK $b_{n-1} b_{n-1}^* = |b_{n-1}|^2 =$ constant, and hence the kernel contributes to *linear* distortion only. As a conclusion, the Volterra kernels should be rearranged, after computation, to account for effects like this. Besides operating by inspection, a general way to reduce the Volterra coefficients in order to account for the modulation format at hand is based on an orthogonal Volterra series. We shall consider this point further on.

Consider then global identification. This is entirely based on computer simulation, and consists of identifying the Volterra kernels of the transmission system (already in their reduced version) through a gradient algorithm (see, [17, ch. 10] for further details about the identification algorithm). Using global identification, the reduction problem mentioned above can be solved at once by using what we call an orthogonal Volterra series, a type of expansion that depends on the channel input characteristics and does not need any further reduction.

A. Underlying Theory

The Volterra expansion (3.4) has the structure of a Taylor series, and as such shares with the Taylor series some negative features. For example, it might be inadequate to represent highly nonlinear systems, or, equivalently, nonlinear systems with large outputs. Moreover, the Volterra model of a given channel may not be improved by adding more terms to the series. Finally, even when the channel input sequence are independent random variables, the terms in (3.4) are not even uncorrelated. Now, many of the drawbacks of the Volterra series can be fixed up by using orthogonal polynomial expansions. These consist of using an input–output relationship of the type

$$y_n = h_0^{(1)} Q^{(1)}(x_n) + h_{n_1 n_2}^{(2)} Q^{(2)}(x_{n_1}, x_{n_2}) + \cdots \tag{4.1}$$

where $Q^{(i)}$ denotes a polynomial of degree i that is orthogonal with respect to the sequence of random variables (x_n). More precisely, the expectation $E[Q^{(i)} Q^*(j)]$ is equal to zero if $i \neq j$, or if $i = j$ but the arguments of $Q^{(i)}$ and $Q^{(j)}$ are not a permutation of each other. If it is assumed that the sequence (x_n) is a stationary sequence of independent, identically distributed random variables, the construction of these orthogonal polynomials is a relatively straightforward task. In fact, the resulting polynomials turn out to be a generalization of multidimensional Hermite polynomials, as defined by Grad [25]. They can be constructed, by using an observation of Zadeh [26], according to the following rule:

$$Q^{(i)}(x_{k1}, x_{k2}, \cdots, x_{ki}) = P_{n1}(x_{k1}) \cdots P_{ni}(x_{ki}) \tag{4.2}$$

where $n_1, \cdots, n_i$ denote the number of indexes of the arguments of $Q^{(i)}$ equal to $k_1, \cdots, k_i$, respectively, and $P_j(\cdot)$ are polynomials in a single indeterminate orthogonal with respect to the random variable x_n, i.e.,

$$E[P_i(x_n)\, P_j(x_n)] = 0 \qquad \text{for} \quad i \neq j$$

where $E[\cdot]$ denotes expectation with respect to x_n. For example,

$$Q^{(3)}(x_1, x_1, x_3) = P_2(x_1)\, P_1(x_3).$$

Consider then the problem of constructing the polynomials $P(\cdot)$. They can be found using a procedure based on the selection of a sequence of linearly independent monomials in the variable x_n, say $f_0, f_1, \cdots$. Explicit formulas are (see also [22, p. 608ff])

$$P_k(x_n) = \det \begin{bmatrix} f_k & f_{k-1} & \cdots & f_0 \\ E[f_{k-1}^* f_k] & E[|f_{k-1}|^2] & \cdots & E[f_{k-1}^* f_0] \\ & & \cdots & \\ E[f_0^* f_k] & E[f_0^* f_{k-1}] & \cdots & E[|f_0|^2] \end{bmatrix}.$$

B. Application to Digital Radio Modulation Systems

In our situation, we can start from the sequence of monomials

$$1, x_n, x_n^*, |x_n|^2, x_n^2, x_n^{*2}, x_n^3,$$
$$|x_n|^2 x_n, |x_n|^2 x_n^*, x_n^{3*}, \cdots.$$

This sequence must be reduced by taking into account the particular type of modulation scheme involved, which may render some of the monomials linearly dependent. For example, with unit-energy PSK we have $|x_n|^2 = 1$, and consequently the fourth, the eighth, and the ninth monomials above must be deleted from the list. Furthermore, for four-phase PSK we have $x_n^3 = \pm x_n^*$, which causes the seventh and the tenth monomial to be deleted, too.

Finally, the polynomials $Q^{(i)}$ associated with the particular modulation scheme can be constructed as follows. Use first rule (4.2), then delete the polynomials which correspond to the components of the channel output falling outside of the bandwidth of interest (see [17, pp. 542 ff] for further details). In practice, this corresponds to keeping only the terms of the type x, $x x x^*$, $x x x^* x^*$,

etc. For example, the Q-polynomials for unit-energy PSK are, up to order three

$$x_i,\ x_i x_j x_k^*,\ x_i^2 x_j^*.$$

Similarly, for unit-energy 16 QAM we have

$$x_i,\ x_i x_j x_k^*,\ x_i^2 x_j^*,\ \left[\,|x_i|^2 - 1\right] x_j,$$
$$|x_i|^2 x_i - 1.32\, x_i.$$

V. Some Examples of Application

We shall now consider some examples of applications of the concepts outlined in previous sections. Examination of a few simple situations will allow us to show the applicability of this theory, and will hopefully enhance its comprehension.

We deal with a nonlinear channel modeled using a bandpass orthogonal Volterra series whose coefficients for PSK signaling are given in [17, p. 566]. This channel results from the cascade of a rectangular shaping filter, a fourth-order Butterworth filter with 3 dB bandwidth $1.7/T$ (T is the signaling period), a typical TWT amplifier exhibiting both AM/AM and AM/PM conversion, and a second-order Butterworth filter with 3 dB bandwidth $1.1/T$. The amplifier is driven at saturation when the sequence at the input of the discrete channel has magnitude 1. (See [17, ch. 10], for more details about this channel.) We proceed to compensate for this channel by inserting in front of it a nonlinear device with memory obtained as an approximation of the channel inverse. In particular, we denote by $(r_1, r_3, \cdots, r_p)$ the compensator obtained by retaining in it only r_1 first-order Volterra coefficients, r_3 third-order coefficients, etc. Thus, for example, (3, 1) indicates a third-order compensator with three first-order and one third-order coefficients. The coefficients are chosen whose indexes are the same as the Volterra coefficients of the channel having the largest magnitudes. Our computational experience has shown this choice to be the most effective, although no formal proof of its optimality has been obtained yet.

Consider first transmitting an 8 PSK symbol sequence driving the amplifier at saturation. The reduced Volterra kernels are listed in Fig. 1. The symbols are $\exp(j0)$, $\exp(j\pi/4)$, $\cdots$, $\exp(j7\pi/4)$. Without any compensation, the samples of the received signal form the constellation shown in Fig. 2, where only the first quadrant is shown for sake of clarity. If a (1, 1) compensator is used, the corresponding constellation looks like Fig. 3. The reduction in the constellation spread is apparent. Notice also the phase rotation introduced, which compensates for the rotation caused by the amplifier's AM/PM. A (4, 1) compensator gives the result shown in Fig. 4, while the effect of a (4, 5) compensator is depicted in Fig. 5.

For 16 QAM signals with the highest energy level driving the amplifier at saturation, the channel quality without compensation is even less satisfactory. Fig. 6 shows the received constellation in the first quadrant: it is seen that

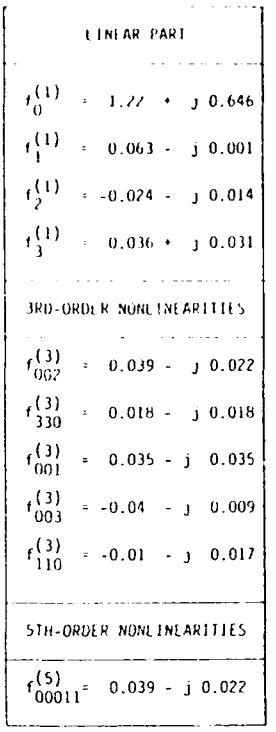


Fig. 1. A set of Volterra kernels for a PSK channel.

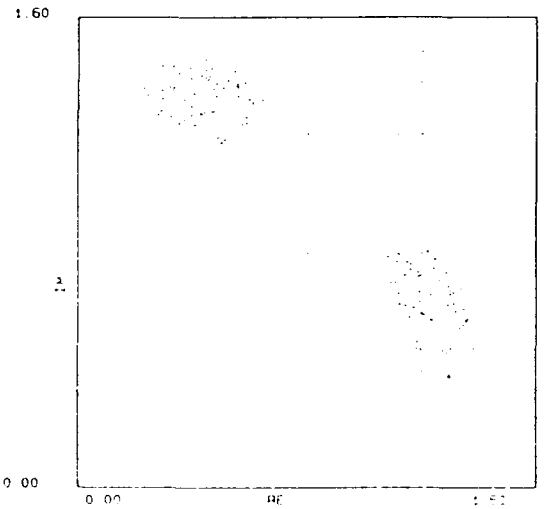


Fig. 2. Signal constellation at the output of the channel of Fig. 1 when 8 PSK is used.

two clusters overlap. The effect of a (1, 1) compensator is shown in Fig. 7, while Fig. 8 depicts the effect of a (4, 1) compensator. Similar results have been obtained for 16 PSK: see Fig. 9 (uncompensated channel), Fig. 10 [(1, 1) compensator], Fig. 11 [(4, 1) compensator], and Fig. 12 [(4, 5) compensator].

For all these situations, the effect of the compensator on the power density spectrum was evaluated, and found to be practically irrelevant: actually, the difference between the power spectra of uncompensated and compensated signals never exceeded a fraction of a decibel.

Consider then the case of a channel whose linear part

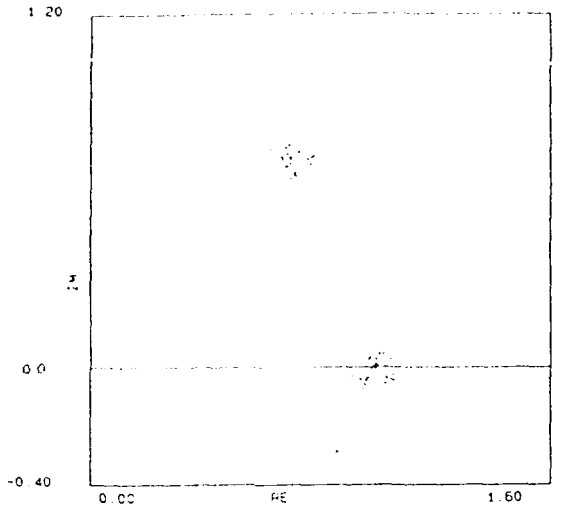


Fig. 3. Same as in Fig. 2, with a (1, 1) compensator.

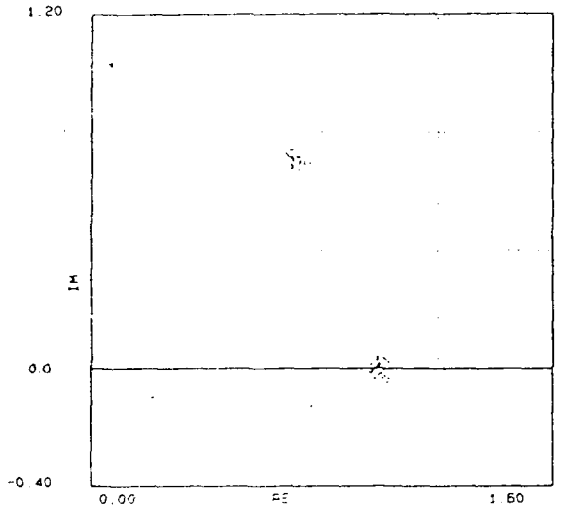


Fig. 4. Same as in Fig. 2 with a (4, 1) compensator.

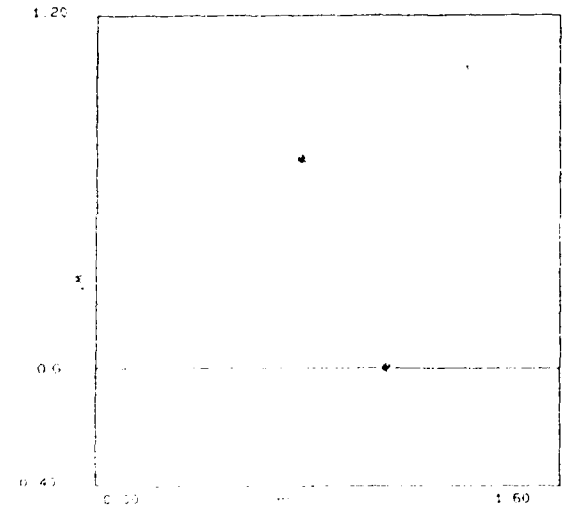


Fig. 5. Same as in Fig. 2 with a (4, 5) compensator.

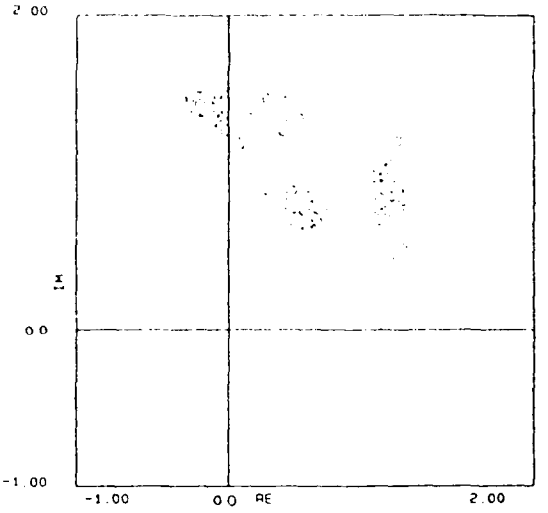


Fig. 6. Signal constellation at the output of the channel of Fig. 1 when 16 QAM is used.

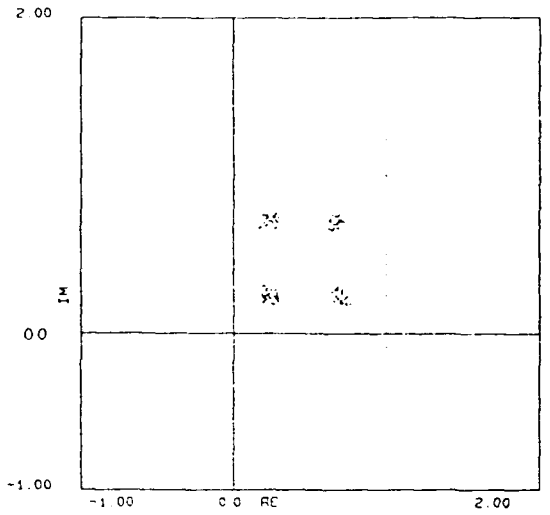


Fig. 7. Same as in Fig. 6, with a (1, 1) compensator.

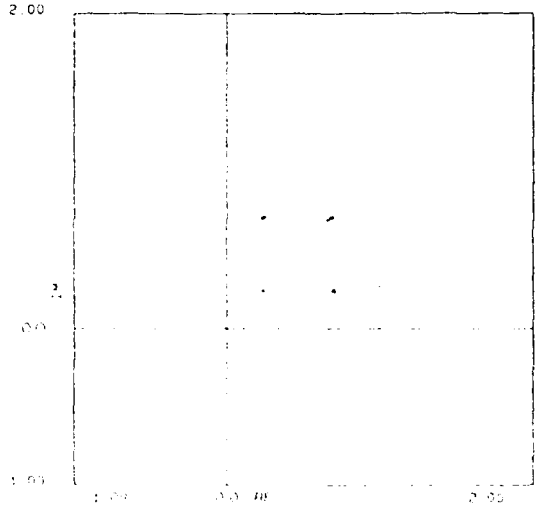


Fig. 8. Same as in Fig. 6, with a (4, 1) compensator.

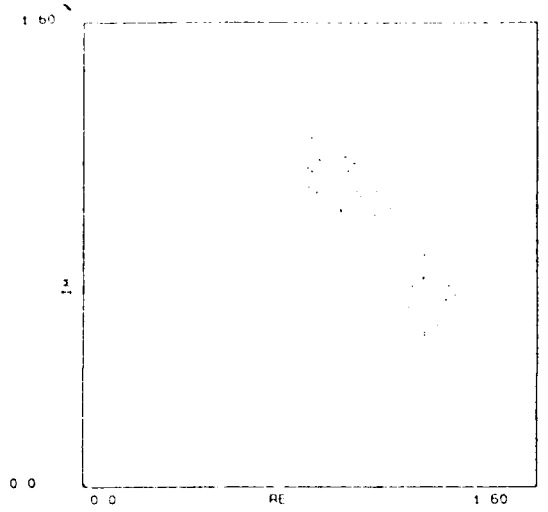


Fig. 9. Signal constellation at the output of the channel of Fig. 1 when 16 PSK is used.

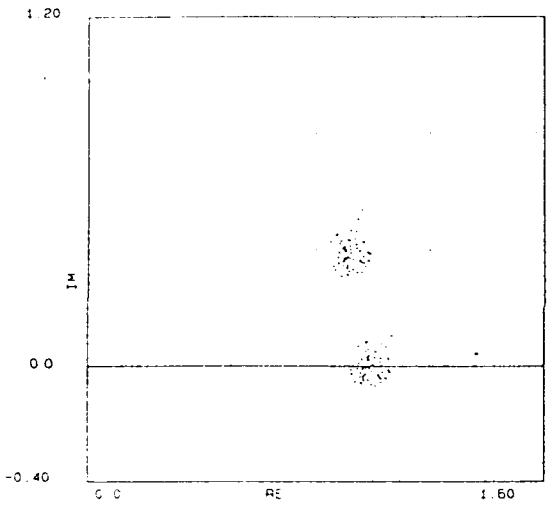


Fig. 10. Same as in Fig. 9, with a (1, 1) compensator.

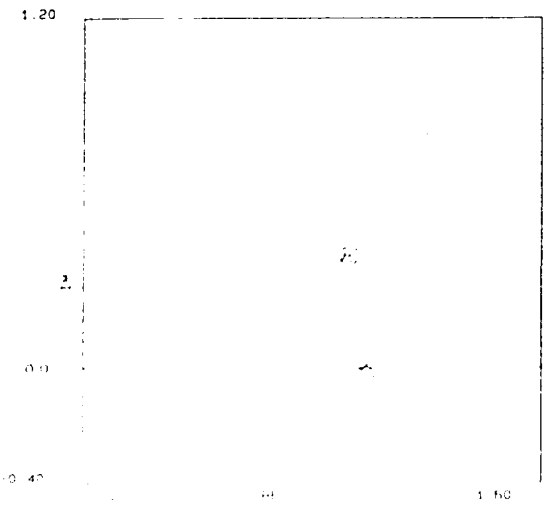


Fig. 11. Same as in Fig. 9, with a (4, 1) compensator.

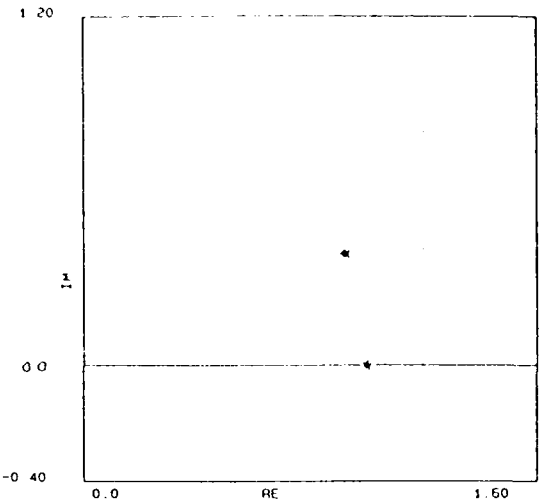


Fig. 12. Same as in Fig. 9, with a (4, 5) compensator.

has been designed to satisfy the Nyquist criterion for no intersymbol interference. Specifically, assume the transmitter and receiver filters to have the common shape of a square-root raised cosine, with a rolloff factor 0.5. The channel between them is modeled through a nonlinear amplifier exhibiting AM/AM and AM/PM conversion effects, driven at saturation, and whose input–output characteristics are described using a model due to Saleh (see [28, eqs. (1)–(5)]) with parameters

$$\alpha_a = 1.9638 \qquad \alpha_\varphi = 2.5293$$

$$\beta_a = 0.9945 \qquad \beta_\varphi = 2.8168.$$

Block identification of this channel turns out to provide rather disappointing results. For example, we get a center linear kernel whose value is $h_0^{(1)} = 1.97 + j0.08$, which fails to account for the rotation (about 40°) introduced by the amplifier at its saturation point. We need to *reduce* the Volterra expansion obtained by block identification, or, even better, to use global identification and orthogonal polynomials. This operation provides the coefficients for the orthogonal Volterra series. The largest among them are listed, up to order three, in Fig. 13. It can be seen that the central linear coefficient reflects the phase rotation caused by the nonlinear amplifier. Figs. 14 and 15 provide a comparison among the scattering diagrams of 8 PSK and 16 PSK, respectively, at the output of a channel with (1, 0) compensation (i.e., compensated only for the rotation and the amplitude scaling) and (3, 6) compensation. Inspection of these scattering diagrams shows that the effect of the third-order compensator, although evident, is less dramatic than for the cases considered previously.

VI. Conclusions

We have considered the design of digital compensators for nonlinear channels. Our design is based on the theory of pth-order inverse of nonlinearities, and on a computer-

LINEAR PART	
$f^{(1)}_{-2}$	= -0.01
$f^{(1)}_{-1}$	= 0.02 + j 0.01
$f^{(1)}_{0}$	= 0.73 + j 0.57
$f^{(1)}_{1}$	= 0.02 + j 0.01
$f^{(1)}_{2}$	= -0.01
3RD-ORDER NONLINEARITIES	
$f^{(3)}_{-1-10}$	=-0.02 - j 0.01
$f^{(3)}_{110}$	=-0.02 - j 0.01

Fig. 13. A set of orthogonal Volterra-series coefficients.

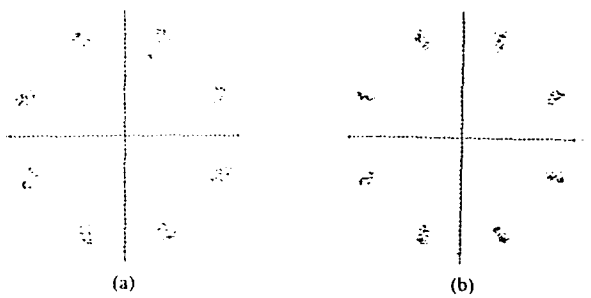


Fig. 14. Signal constellations at the output of the channel modeled by the coefficients of Fig. 13 for 8 PSK. (a) With (1, 0) compensation. (b) With (3, 6) compensation.

Fig. 15. Signal constellations at the output of the channel modeled by the coefficients of Fig. 13 for 16 PSK. (a) With (1, 0) compensation. (b) With (3, 6) compensation.

aided analysis of the system to be compensated. A number of examples was worked out to show the applicability of this approach. In principle, it is possible to compensate a given channel to any desired degree of accuracy. However, obvious complexity limitations make the approach presented here more useful when the uncompensated channel is strongly nonlinear. In fact, if only a third-order compensator is allowed, our results show that it will work better with a channel with a few strong low-order nonlinearities than with a channel which has small nonlinear Volterra coefficients, but many of them are of a higher order. In the latter case, a certain amount of power backoff may prove more beneficial than a nonlinear compensation. (Notice that the backoff can be included in our model by simply multiplying the right-hand side of (3.13) by a factor smaller than one.)

References

[1] A. Gersho and E. Biglieri, "Adaptive equalization of channel nonlinearities for data transmission," presented at the 1984 Int. Conf. Commun. (ICC'84), Amsterdam, The Netherlands, May 1984.

[2] E. Biglieri, A. Gersho, R. D. Gitlin, and T. L. Lim, "Adaptive cancellation of nonlinear intersymbol interference for voiceband data transmission," *IEEE J. Select. Areas Commun.*, vol. SAC-2, pp. 765-777, Sept. 1984.

[3] D. D. Falconer, "Adaptive equalization of channel nonlinearities in QAM data transmission," *Bell. Syst. Tech. J.*, vol. 57, pp. 2589-2611, Sept. 1978.

[4] S. Benedetto and E. Biglieri, "Nonlinear equalization of digital satellite channels," *IEEE J. Select. Areas Commun.*, vol. SAC-1, pp. 57-62, Jan. 1983.

[5] S. Pupolin and L. J. Greenstein, "Digital radio performance when the transmitter spectral shaping follows the power amplifier," *IEEE Trans. Commun.*, vol. COM-35, pp. 261-266, Mar. 1987.

[6] S. Pupolin and L. G. Greenstein, "Performance analysis of digital radio links with nonlinear transmit amplifiers," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 534-546, Apr. 1987.

[7] J. Namiki, "An automatically controlled predistorter for multilevel quadrature amplitude modulation," *IEEE Trans. Commun.*, vol. COM-31, pp. 707-712, May 1983.

[8] W. J. Weber, III, "The use of TWT amplifiers in M-ary amplitude and phase shift keying systems," in *Proc. 1975 Int. Conf. Commun.* (ICC'75), San Francisco, CA, June 16-18, 1975, pp. 36.17-36.21.

[9] A. A. M. Saleh and J. Salz, "Adaptive linearization of power amplifiers in digital radio systems," *Bell Syst. Tech. J.*, vol. 62, no. 4, pp. 1019-1033, Apr. 1983.

[10] P. Hetrakul and D. P. Taylor, "Compensation of bandpass nonlinearities for satellite communications," in *Proc. 1975 Int. Conf. Commun.* (ICC'75), San Francisco, CA, June 16-18, 1975, pp. 36.22-36.26.

[11] K. Sam Shanmugan and M. J. Ruggles, "An adaptive linearizer for 16-QAM transmission over nonlinear satellite channels," in *Proc. Globecom '86*, Dec. 1986, paper #4.3.

[12] M. F. Mesiya, P. J. McLane, and L. L. Campbell, "Maximum likelihood receiver for carrier-modulated data transmission systems," *IEEE Trans. Commun.*, vol. COM-22, pp. 624-636, May 1974.

[13] V. K. Dubey and D. P. Taylor, "Maximum likelihood sequence detection for QPSK on non-linear, bandlimited channels," *IEEE Trans. Commun.*, to be published.

[14] W. van Etten and F. van Vugt, "Maximum likelihood receivers for data sequences transmitted over nonlinear channels," *A.E.Ü.*, vol. 34, pp. 216-223, 1980.

[15] A. F. Elrefaye and L. Kurz, "A minimum mean square error equalizer for nonlinear satellite channels," *IEEE Trans. Commun.*, vol. COM-35, pp. 556-560, May 1987.

[16] E. Biglieri, M. Elia, and L. LoPresti, "The optimal linear receiving filter for digital transmission over nonlinear channels," *IEEE Trans. Inform. Theory*, to be published.

[17] S. Benedetto, E. Biglieri, and V. Castellani, *Digital Transmission Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[18] G. D. Forney, Jr., "Lower bounds on error probability in the presence of large intersymbol interference," *IEEE Trans. Commun.*, vol. COM-20, pp. 76-77, Feb. 1972.

[19] S. Benedetto, E. Biglieri, and R. Daffara, "Modeling and performance evaluation of nonlinear satellite links—A Volterra series approach," *IEEE Trans. Aerospace Electron. Syst.*, vol. AES-15, pp. 494-507, July 1979.

[20] E. Biglieri, "High-level modulation and coding for nonlinear satellite channels," *IEEE Trans. Commun.*, vol. COM-32, pp. 616-626, May 1984.

[21] E. Biglieri, "Theory of Volterra processors and some applications," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Paris, France, May 1982, pp. 294-297.

[22] J. F. Barrett, "The use of functionals in the analysis of nonlinear physical systems," *J. Electron. Contr.*, vol. 15, pp. 567-615, Dec.

1963, reprinted with corrections in A. H. Haddad, ed., *Nonlinear Systems* (Benchmark Papers in Electrical Engineering and Computer Science, Vol. 10). Stroudsburg, PA: Dowden, Hutchinson and Ross, 1975.
[23] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. New York: Wiley, 1980.
[24] R. Calderbank and J. E. Mazo, "A new description of trellis codes," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 784-791, Nov. 1984.
[25] H. Grad, "Note on N-dimensional Hermite polynomials," *Commun. Pure Appl. Math.*, vol. 2, pp. 325-330, Dec. 1949.
[26] L. A. Zadeh, "On the representation of nonlinear operators," *IRE Western Conv. Rec.*, pt. 2, pp. 105-113, 1957.
[27] D. W. Lytle, "Convergence criteria for transversal equalizers," *Bell Syst. Tech. J.*, pp. 1775-1800, Oct. 1968.
[28] A. A. M. Saleh, "Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers," *IEEE Trans. Commun.*, vol. COM-29, pp. 1715-1720, Nov. 1981.

Ezio Biglieri (M'73-SM'82), for a photograph and biography, see this issue, p. 4.

Sergio Barberis was born in [redacted], on [redacted]. He received the degree in electronic engineering from the Politecnico di Torino, Torino, Italy, in 1986.

Since 1987 he has been with the Data Communication System Division of AET S.P.A., Torino. His current interests are in the area of digital communication systems.

Maurizio Catena was born in [redacted], Italy, in 1961. He received the degree in electronic engineering from the Politecnico di Torino, Torino, Italy, in 1986.

His present interests include negative effects of nonlinearities in digital transmission systems.

Appendix C

GROUP CODES AND SIGNAL DESIGN FOR DIGITAL TRANSMISSION

by

Michele Elia

Dipartimento di Elettronica - POLITECNICO DI TORINO - ITALY

I - INTRODUCTION

Symmetry seems to be a feature intrinsic to every life process. It should be a very stimulating undertaking to discuss the fundamental role played by symmetry in art, music, chemistry, biology, physics, computer science and more generally in every mathematical science. A fascinating sample of this subject was provided by H. Weyl [53] in his last book dedicated to a synthetic view of symmetry. Nevertheless in this paper we limit our considerations to the key role of symmetry in communication theory. In this field symmetry plays an indispensable part in reducing the complexity of every data transmission scheme.

The algebraic notion of group underlies both the geometrical description of digital signals proposed by Shannon, [43], and the geometrical methods of error control codes developed shortly after Shannon's work. However the introduction and systematic use of methodology, machinery and language of group theory in both coding theory and signal design must be ascribed to Slepian [2,3].

In some way Slepian's approach parallels Klein's Erlagen program on the foundation of geometry: all geometric objects and concepts can be formulated starting from the abstract notion of group which provides

---

This work has been sponsored in part by the United States Army through its European Research Office grant N. DAJA45-86-C-0044, and in part by Consiglio Nazionale delle Ricerche through grant N. 86.02428.07.

the appropriate tool for every useful and applied mathematical theory. In Klein's words "a geometry is defined by a group of transformations, and investigates everything that is invariant under the transformations of the given group". In our context the main object left invariant by the group is a code, as will be defined later.

The Shannon theory of any communication process shows that the information is inherently discrete and also that the quantity of information that can be processed by every practical system is finite.

Signals for sending information over physical channels are essentially time- and frequency-limited; as a consequence the dimension of the signal space is finite. The signal energy, defined as the integral of the signal square over its finite time interval, induces an euclidean metric in this signal space. Therefore, by using an orthonormal basis, we associate to each signal a point (or vector) in an euclidean finite dimensional space. In this way a finite set of signals corresponds to a finite constellation of points that we call a code.

Early in the fifties Slepian introduced the concept of group code in the design of signal sets for the Gaussian channel. A group code is a set of M unit vectors spanning an n-dimensional real space, on which the matrices of a finite group representation operate transitively.

A straightforward generalization of Slepian's group codes is obtained by considering a set of initial vectors instead of just one vector. The resulting set of vectors is called generalized group alphabet.

The present awakening of interest in group codes is due to their increasing use in transmission schemes of combined modulation with either convolutional or block codes, an approach initiated by Ungerboeck.

A fundamental problem for Slepian's group codes is the choice of the initial vector that maximizes the minimum distance. A second basic problem concerns the existence of group codes for every pair of integers with M greater than n. The classification of all configurations of given dimension is constructively important. As far as we know, only the classification in dimension three is complete. The same problems, formulated for generalized group alphabets, seem even more difficult.

However the field is wide and deserves investigations either from a purely theoretical point of view or for practical applications.
We are aware of the fact that the theory of group codes is still incomplete, but the open problems really challenge the human thinking and stimulate the research work of engineers and mathematicians alike.

II - SIGNAL SETS: THE GEOMETRICAL MODEL

Signals for sending information are essentially limited both in time and frequency. According to a point of view accepted in the past, the simultaneous concentration attainable in both domains is limited by an uncertainty principle, so named after the analogous relations in quantum mechanics. Moreover energy constraints are imposed for practical purposes.
Finite bandwidth W and finite time duration T together imply that the dimension of the Hilbert space of the signals is essentially finite.
If we require strictly finite duration and simultaneously maximum concentration of signal energy in a given bandwidth, we have a problem whose natural mathematical setting is the calculus of variations. This problem has been thoroughly discussed, [30,5,40,41], even if its consequences have not received much attention from the signal designers yet.
Let V be a Hilbert space with support the interval [0,T], and let the scalar product be defined as

$$(\varphi,\psi) = \int_0^T \varphi(t)\,\overline{\psi(t)}\,dt \qquad \varphi(t),\ \psi(t)\in V$$

where overbar denotes complex conjugation.
The norm square $\|.\|^2$, defined as $\|\varphi\|^2 = (\varphi,\varphi)$ represents the energy of the signal $\varphi(t)\in V$. In the set of linear operators acting in V and having a discrete spectrum, the operators associated to linear filters

are of particular interest. Let $H(f)$ denote the filter transfer function. Therefore the Fourier transforms $\Phi(f)$ and $\Psi(f)$, respectively of filter input and output signal, are related by

$$\Psi(f) = H(f)\ \Phi(f) \quad .$$

The problem now is to seek the input function $\varphi(t)$, of unit energy, for which the energy of the corresponding output functions $\psi(t)$, in the bandwidth $[-W,W]$, is as large as possible. That is, we want to maximize the following integral

$$I_1 = \int_{-W}^{W} \Psi(f)\ \overline{\Psi(f)}\ df = \int_{-W}^{W} H(f)\ \overline{H(f)}\ \Phi(f)\ \overline{\Phi(f)}\ df$$

under the constraint

$$I_2 = \int_{-\infty}^{\infty} \Phi(f)\ \overline{\Phi(f)}\ df = 1 \quad .$$

By means of Lagrange's multipliers the solution is found to be the eigenfunction associated to the largest eigenvalue of the integral equation

$$(1) \qquad \int_{0}^{T} K(t-s)\ \varphi(s)\ ds = \lambda\ \varphi(t) \qquad t\in[0,T]$$

where the positive definite kernel is defined by the Fourier transform

$$K(t-s) = \int_{-W}^{W} H(f)\overline{H(f)}\ \exp[2\pi j(t-s)f]\ df.$$

The positive eigenvalues λ, ordered in decreasing order, exhibit the typical trend shown in Fig.1, which demonstrates that the dimension of the signal space of functions limited both in time and frequency is essentially finite and can be taken to be approximately $2WT$, [5]. (If $2TW>10$, this statement is true within an energy dispersion of some few per cent and irrespective of $H(f)$).

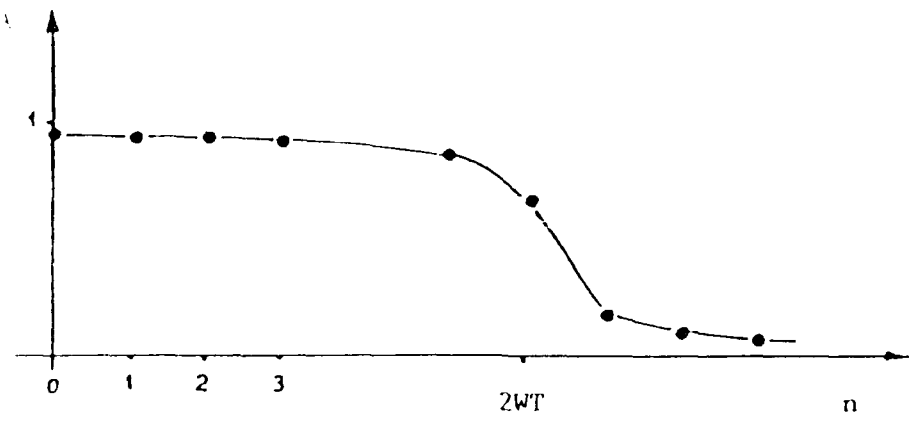


Fig.1 - Typical behavior of the eigenvalues of equation (1)

A natural orthogonal basis $B = \{\psi_i(t)\}_{i=1}^{n}$, $n \leq 2WT$, for the space of the signals limited both in time and frequency is provided by the set of normalized eigenfunctions associated to the set of eigenvalues of greatest value. By means of the basis B, we can uniquely associate to a given set A of M signals

$$m_i(t) = \sum_{j=1}^{n} x_{ij}\, \psi_j(t) \qquad i=1, \ldots, M$$

a set C of M vectors

$$X_i=(x_{i1}, \ldots, x_{in}) \qquad i=1, \ldots, M \ .$$

that we call code. The square of the Euclidean length of a vector X is equal to the energy of the signal m(t).

We can now describe the operation of a quite general model of transmission scheme at the level of signal manipulation.

A transmitter associates to every source symbol, in a one-to-one way, a signal chosen in the set A and sends this signal through the channel.

The channel operates by adding to the transmitted waveform m(t) a sample of a zero-mean random process v(t) with known spectral density.

The received signal is thus

$$r(t) = m_\xi(t) + v(t) \qquad t\in[0,T]$$

where ξ is a random variable taking values in the set $\{1,\ldots,M\}$.

If we confine ourselves to coherent detection, from the observation of r(t) over the interval [0,T], the receiver makes an estimate of the value taken by ξ, that is, an estimation of the symbol emitted by the

source. Let us suppose that all the information relevant to every detection criterion lies in the signal space, therefore any decision can be taken by referring to the vector

$$\underline{r}=(r_1, \ldots, r_n)$$

where

$$r_i = \int_0^T r(t)\, \overline{\psi_i(t)}\, dt \quad .$$

This is equivalent to considering a discrete-time continuous-amplitude additive channel that produces

$$\underline{r} = \underline{X}_\xi + \underline{N}$$

where: $\underline{N}$ is a random vector with known probability density f(.);

$\underline{X}_\xi$ is a transmitted code vector from the code $\mathbf{C}$.

At the receiver end, the decision taker may be described by an exhaustive partition of the n-dimensional space into M' disjoint regions R_i, i=1,...,M', if the received vector $\underline{r}$ falls in region R_j then the detected symbol will correspond to the integer j. We say that the demodulator takes a "hard" decision or a "soft" decision depending on whether M'=M or M'>M respectively. In conclusion the channel is modelled by a discrete memoryless channel with M input symbols and M' output symbols.

III - MEASURES OF PERFORMANCE

The performance evaluations of group codes on communication channels rule the development of the entire theory of group codes. Hereafter we briefly review some important performance indices used in digital communication systems. In order to avoid discussions depending on transmission protocols, here and in the following we will deal only with transmission schemes based on hard decisions. In this context the most typical index is error probability, i.e. the probability that the receiver takes a wrong decision about the symbol emitted by the infor-

mation source. Assuming in particular equienergetic codes, white Gaussian noise channel and maximum likelihood decision criterion at the receiver's end, then the regions R_i, i=1,...,M, will be connected hypercones bounded by hyperplanes with the vertices in the origin. Therefore the error probability is given by a sum of n-dimensional integrals; letting $\hat{R}_i$ denote the complementary region of R_i in R^n and let $p\{X_i\}$ be the probability of sending message i, we have

$$p(e) = \sum_{i=1}^{M} \int_{\hat{R}_i} f(X-X_i)\, dX\, p\{X_i\} \quad .$$

A second important index is the configuration matrix $C=(c_{ij})$ defined as the Gram matrix of the set of vectors, i.e.

$$c_{ij} = X_i^T X_j \quad .$$

This matrix C occupies a central position in the theory of group codes. It conveys all the information relevant to evaluate code performances on the white Gaussian channel and is also useful to compute other performance indices.

A third relevant index is the minimum distance defined as the minimum distance between any pair of distinct vectors of the code, that is

$$d^2_{min} = \min_{i \neq j} \| X_i - X_j \|^2 \quad .$$

The evaluation of each performance index is usually very hard, so that frequently the knowledge of upper and/or lower bounds is of sufficient interest. As an example we derive an upper bound for the error probability, that applies to symmetric point configurations.

Let us assume that the code has a symmetry such that the error probabilities conditioned on a given code vector do not depend on this vector, i.e. $p\{e\} = p\{e|X_i\}$ i=1,...,M

Let the region R_i, i=1,...,M, be bounded by the set of s hyperplanes of equations $\|X-X_i\|^2 = \|X-X_j\|^2$

where j belongs to a convenient subset of $\{1,\ldots,M\}$; the explicit equation of each hyperplane turns out to be $X^T(X_i-X_j) = 0$.
Applying the union bound, we get a general upper bound for the error probability

$$p(e) = p\{e|X_i\} = \int_{\hat{R}_i} f(X-X_i)\,dX < \sum_{j=1}^{s} \int_{\Omega_j} f(X-X_i)\,dX$$

$$\leq s \int_{\Omega_o} f(X-X_1)\,dX$$

where Ω_j is the halfspace defined by the inequality $X^T(X_i-X_j) \leq 0$.

Ω_o is the halfspace defined by the inequality $X^T(X_1-X_o) \leq 0$.

and X_o is a code vector at the minimum distance from X_1.

More detailed comments on performance indices will be provided after the description of the main features of group codes.

IV - GROUP CODES

Symmetry seems to be an unavoidable occurrence for reducing the complexity of every high-dimensional set of signals as required by Shannon's channel theorem to guarantee high coding performance. For instance, we can take advantage of symmetry in designing good decoding algorithms for error control codes. Symmetry makes feasible the new digital modulation schemes which combine error control codes and modulations.
As we observed in the introduction, symmetry cannot be separated from the notion of group which discloses symmetry's real nature and constitutes its formal counterpart. It was early in the fifties that Slepian introduced the group codes for Gaussian channels; his ideas found a definitive formulation in a stimulating paper [3], in 1965.

Now let us formally define the main object of this paper.

Definition 1.

Consider a finite set $S(G) = \{D(g): g\in G\}$ of real orthogonal matrices that form a faithful representation of a finite group G and consider an n-dimensional unit vector X_1. The set $S(G)X_1 = \{X_g=D(g)X_1 : g\in G\}$ of M vectors generated by the action of S(G) on X_1 is called group code and denoted by [M,n], if it spans the n-dimensional space; otherwise it is called planar group code.

In the present theory, group representations by matrices having real entries are a fundamental mathematical tool.

The theory of group representations originated in the middle of the nineteenth century from the works of many mathematicians. Equipped with the theory of group characters, (the character of $g\in G$ is the trace of the matrix D(g)), the theory of matrix groups assumed a central role in the development of modern algebra. We do not try to survey this subject. To coding theorists we recommend the book by Blake and Mullin [12], while for a thorough development of the topic we refer to the books by Curtis and Reiner [24], Burrow [17] and van der Waerden [48]. Old fashioned but very rich and suggestive is the book by Burnside, [16].

For easy reference and later use we recall some results concerning group representations.

1 - A group representation is either irreducible or completely reducible, i.e. it can be written as direct sum of irreducible components.

2 - A representation with real entries may be either real reducible, or real irreducible. In this second case it may still be complex reducible or not.

3 - The number of distinct irreducible components is equal to the number of group classes.

4 - Given two representations of groups G and G_1 we obtain a representation of their direct product by means of the direct matrix sum

$D(g\ g') = D(g) \oplus D(g')$ $\qquad g \in G$ and $g' \in G_1$

The concept of direct matrix sum is very important in describing the structure of group codes. The general observation fits a paradigmatic principle: in many instances to split a problem means to solve it.

Let $|G|$ denote the cardinality of the group G. The cardinality M of the code may be less than or equal to $|G|$. In case it is less there exists a subgroup H of G such that the initial vector is left invariant, i.e.

$$HX_1 = X_1$$

where with HX_1 we denote the set $\{X: X=D(h)X_1\ ,\ h \in H\}$.

The proof of the following theorem is straightforward and follows from definition 1 and elementary properties of the groups.

Theorem 1.

i) $|G| \geq M$ and $|G|\ |\ M!$

ii) if $|G| > M$ then $M\ |\ |G|$

where $d|b$ means that d is a divisor of b.

The following theorem concerning the subgroup H, has an important consequence on the existence conditions for group codes. It is also useful to clarify the relations between the group and the code.

Theorem 2.

The subgroup H cannot be normal.

See [7, 12, 35] for a proof.

Theorem 3.

If G is abelian then $|G| = M$.

Besides the abstract properties of the group G, also conditions concerning the skeleton of its representations are important for distinguishing between planar and non planar codes.
In order that an initial vector exists such that the generated set of vectors spans the n-dimensional space, the representations of the group G must satisfy the condition expressed in the following theorem.

Theorem 4.

Given an n-dimensional representation D(g) of a group G, a vector $X_1 \varepsilon E^n$ exists such that the set $\{D(g)X_1, \ g\varepsilon G\}$ of vectors spans E^n if and only if every irreducible representation contained in D(g) appears with a multiplicity less than or equal to its dimension.
For a proof see Blake and Mullin [12].

Definition 2.

A representation is said full homogeneous if every irreducible component has a multiplicity equal to its dimension.

The symmetry of a group code is exploited by the configuration matrix. According to the previous definition, it is an M by M matrix of rank n the entries of which are the scalar products $c_{ij} = X_i^T X_j \quad i,j=1,\ldots,M$. It is also of interest to define an extended configuration matrix C^e whenever $|G|>M$. Let $X_g=D(g)X_1$ be the vector produced by the action of the element $g\varepsilon G$. We define the extended configuration matrix as the $|G|$ by $|G|$ Gram matrix whose entries are

$$c_{gg'} = X_g^T X_{g'} \qquad g,g' \varepsilon G$$

Since $H\neq\{e\}$, the vectors of the set $S(G)X_1$ are not all distinct; in fact the same vector appears with multiplicity $|H|$.
The following theorem illustrates the shape and structure of configuration matrices which rely in depth on the associated group.

Theorem 5.

The rows of any configuration matrix of a group code are permutations of the first one.
This applies to both extended and not extended configuration matrices.
For a proof see [3] and [10].

It is not hard to verify that the extended C^e configuration matrix is the Kronecker product of C by a matrix J, (possibly with a re-ordering of rows and columns):

$$C^e = C \otimes J$$

where J is a convenient matrix of which all entries are 1s.
The importance of the configuration matrix C of a group codes, was enhanced by Slepian's proof, [3], that it is possible to recover the vectors of the code from C. Let $P_H(g)$, $g \in G$, denote the permutation matrices of the right permutation representation of G induced by its subgroup H; let AG(H) be the group algebra of G generated by these permutation matrices, and let AZ(H) be the centralizing algebra of AG(H). We have the following theorems.

Theorem 6.

The extended configuration matrix of a group code can be written as the sum

$$C^e = \Sigma_g c(g)\ L(g)$$

where L(g), $g \in G$, are the permutation matrices of the left regular permutation representation of G.

Theorem 7.(Slepian)

The extended configuration matrix commutes with all the permutation matrices of the right regular permutation representation of G, i.e. C^e belongs to the centraling algebra of the group algebra of the right regular permutation matrices.

The configuration matrices of different group codes generated by diffe-

rent irreducible representations of the same group G may originate an orthogonal basis in the regular group algebra AG({e}), as stated in the following theorem due to Blake.

Theorem 8.

Let D(g) and D'(g) be real irreducible representations of the finite group G of dimensions n_i and n_j, respectively, and C_i and C_j the configuration matrices of the group codes $\{D(g)X_i,\ g\in G\}$ and $\{D'(g)X_j,\ g\in G\}$, respectively. Then

i) if D(g) and D'(g) are not equivalent, then $C_i\ C_j = 0$ for any X_i and X_j;

ii) if $D(g) \simeq D'(g)$ and $X_i = X_j$, then $(C_i)^2=(G/n_i)\ \|X_i\|^2\ C_i$.

For a proof see Blake and Mullin [12].

Furthermore special structures of the configuration matrix may uniquely characterize the group code.

Theorem 9.(Blake)

Let us consider the configuration matrix C of an [M,n] code in which all entries of the first row are distinct.

Then C is the configuration matrix of a group code if and only if:

i) its rows are permutations of the first one;

ii) M is a power of 2, i.e. $M=2^s$;

iii) in the decomposition

$$C= \Sigma\ c_i\ P_i$$

the matrices P_i are permutation matrices of order two and commute with each other.

Moreover $n\geq s$ and the group generating the code is commutative of type $(1,1,\ldots,1)$.

Now we can devise a general theorem concerning the conditions for a given Gram matrix to be the configuration matrix of a group code. However the formulation of such general conditions may be quite unsati-

sfactory, because they lack either classical mathematical fascination or practical utility. It is a challenging question to find more pleasant and possibly useful conditions.

Theorem 10.

A Gram matrix C is the configuration matrix of a group code if and only if

i) rows of C are permutations of the first one;

ii) a matrix J, all entries of which are 1s and the order of which is not greater than (M-1)!, exists such that the matrix $C' = C \otimes J$ commutes with all matrices of a right regular representation of a group G.

See [10] for a proof.

We stop here the presentation of Slepian's group codes. In the next section we shall consider an extension that will include multilevel codes which share, of course, the same underlying property of symmetry.

V - GENERALIZED GROUP ALPHABETS

The class of multidimensional alphabets is introduced. Special instances of these codes have been widely used for designing multidimensional signals in combined modulation and coding. Their structure is very rich in symmetries and, as far as we know, most of the signal constellations in actual use, either equienergetic or not, belong to this family.

Definition 3.

Consider a set of K n-vectors $\underline{X} = \{X_1, \ldots, X_K\}$, called the initial set, and L orthogonal n x n matrices $S_1, \ldots, S_L$ that form a representation S(G) of the group G. The set of vectors $S(G)X_1, \ldots, S(G)X_K$ obtained from the action of S(G) on the vectors of the initial set is called a Generalized Group Alphabet, and from now on shortened to GGA.

Definition 4.

A GGA is called separable if the vectors of the initial set are transformed by S(G) into either disjoint or coincident vector sets, i.e.,

$$S(G)X_j \cap S(G)X_k = \begin{cases} \emptyset & j \neq k \\ S(G)X_j & j = k \end{cases}$$

Since an orthogonal matrix transforms a vector into one with the same length, the signals associated with a GGA have as many energy levels as there are in the initial set.

Definition 5.

A GGA is called regular if the number of vectors in each subalphabet $S(G)X_j$, $j=1,\ldots,K$, does not depend on j, i.e., each vector of the initial set is transformed by S(G) into the same number of distinct vectors. A regular GGA is called strongly regular if each set $S(G)X_j$ contains exactly L distinct vectors.

The following result stems directly from the definitions.

Theorem 11.

The number M of vectors in a regular GGA is a multiple of K. If GGA is strongly regular, then M=KL.

We consider now some distance properties of the elements of a GGA. Choose a partition of a GGA into m subsets $\underline{Z}_1$, $\underline{Z}_2,\ldots,$ $\underline{Z}_m$. For each subset $\underline{Z}_i$, we can define the intradistance set as the set of all the Euclidean distances among pairs of vectors in $\underline{Z}_i$. For any pair of distinct subsets $\underline{Z}_i$, $\underline{Z}_j$, we define their interdistance set as the set of all the Euclidean distances between a vector in $\underline{Z}_i$ and a vector in $\underline{Z}_j$.

Definition 6.

The partition of a separable GGA into m subsets $\underline{Z}_1,\ldots,\underline{Z}_m$ is called fair if all the subsets are distinct, include the same number of vectors and their intradistance sets are equal.

We shall now present a constructive method to generate fair partitions of a GGA. Consider the generating group S(G) of the GGA, one of its subgroups, say S(H), and the partition of S(G) into left cosets of S(H). We have the following result.

Theorem 12.

If the left cosets of the subgroup S(H) are applied to the initial set of a strongly regular GGA, this procedure results in a fair partition of the GGA. Under the same hypotheses, if S(H) is a normal subgroup, then left and right cosets give rise to the same fair partition.
For a proof see [11].

The condition of strong regularity of the GGA can be removed: but in this case it may happen that different cosets generate the same element of the partition. Hence, some of the cosets must be removed from consideration. Moreover, notice that if S(H) is a normal subgroup of S(G), then we do not need to distinguish between left or right coset partitions. On the contrary, if S(H) is not normal, the partitions obtained from right cosets may not be fair, as it can be shown by a counterexample. In some cases, we are interested in further partitioning every element $\underline{Z}_i$ in the same number of subsets. This leads to the concept of a chain partition, that is the GGA is partitioned in subsets which in turn are partitioned in the same number of sub-subsets, and so on. We call level of a subset in the chain partition the number of inclusions beetwen the given subset and the whole group code.

Definition 7.

The chain partition of a separable GGA is called fair if any two elements of the partition at the same level of the chain include the same number of vectors and have equal intradistance sets.

For fair chain partitions we have the following theorem.

Theorem 13.

Consider a strongly regular GGA, and a chain of subgroups of its generating group S(G), that is

$$S(H_1) \subset S(H_2) \subset S(H_3) \subset \ldots \subset S(H_s) = S(G) \quad .$$

Use H_{s-1} and its left cosets to generate a partition of GGA. Then, use H_{s-1} and its left cosets in H_s to further partition all the sets of the previous partition. Repeat the procedure with H_{s-2}, and so on, until H_1 and its left cosets in H_2 are used. The resulting chain partition of GGA is fair.

A theorem concerning the interdistance sets sheds some further light on the symmetry properties of GGA's.

Theorem 14.

Let H be a normal subgroup of G. The partition of a strongly regular GGA obtained by applying the left cosets of H to the initial set $\underline{X}$ has the following property: the interdistance set associated with any two cosets, say S_1H and S_2H, is a function only of the coset S_3H, where $S_3 = S_1^T S_2$, and not of S_1, S_2 separately.
For a proof see [11].

We conclude this section by showing how GGAs, in particular group codes, can be used in conjunction with error control codes to exploit the channel capacity further. We shall illustrate first the joint use of multidimensional alphabets and block codes, thus we will describe how the signal alphabets are paired to convolutional (trellis) codes.

Imai and Hirakawa [33] and recently Ginzburg [31] have described constructions which make it possible to design set of signals with a regular structure and with an arbitrary minimum distance as insured by the algebraic properties of block codes. Ginzburg's construction considers L block encoders $C_1, C_2, \ldots, C_L$ which accept source symbols, and output L blocks $(q_{1i}, q_{2i}, \ldots, q_{Ni})$, i=1,...,L, of N symbols each. The modulator f maps each L-tuple $(q_{j1}, \ldots, q_{jL})$, j=1,...,N, into the vector

$$X_j = f(q_{j1}, \ldots, q_{jL}), \quad j = 1, \ldots, N$$

chosen from a GGA of $M = M_1 \ldots M_L$ elements. This mapping is obtained as follows. In GGA we define a system of L partitions such that each class of the ℓ-th partition includes M_ℓ classes of the $(\ell-1)$-th partition. Each class will consist of $M(\ell) = M_1 M_2 \ldots M_\ell$ signals. By numbering the classes of the $(\ell-1)$-th level occurring in a class of the ℓ-th level we can obtain a one-to-one mapping of the set of classes of the $(\ell-1)$-th partition onto the set of integers $\{0, \ldots, M_\ell - 1\}$. Therefore, if q_{ij} are chosen in the set $\{0, \ldots, M_\ell - 1\}$, $\ell = 1, \ldots, L$, any L-tuple $(q_{j1}, \ldots, q_{jL})$ defines a unique value of the j-th elementary signal $X_j = f(q_{j1}, \ldots, q_{jL})$.

We shall now see how an Ungerboeck code can be designed using GGA. The procedure suggested in [47] and called "mapping by set partitioning", can be achieved by the notion of fair partition, which represents a systematic generalization of that concept.

Each coded symbol depends on $k+\nu$ source bits, namely the block $\tau = (a_1, \ldots, a_k)$ of k bits generated by the source, plus ν bits preceding this block. The ν bits determine one of the $N = 2^\nu$ states of the encoder, say $\sigma = (a_{k+1}, \ldots, a_{k+\nu})$, $a_n = 0, 1$. The encoder state for the next coded symbol is obtained by shifting the a_n's k places to the right, dropping the right-most k bits and inserting on the left the most recent k source bits. The encoded symbol X_j, which is an element of a GGA, depends on τ and σ and, in this framework, the encoding procedure

can be described using a trellis and by assigning to the branches outgoing from each node the set of symbols obtained from a fair partition of a GGA.

VI - THE INITIAL VECTOR PROBLEM

The minimum distance is a relevant factor to define the code performance on noisy channels because it is a fact that distant signals are hard to confuse as an effect of the noise. Moreover monotone decreasing functions of the minimum distance constitute an upper bound to the error probability. It follows that codes with large minimum distances are desirable, and in particular the choice of Slepian's group codes with the greatest minimum distance leads to the initial vector problem which is also interesting from a geometrical point of view.

The initial vector problem for group codes can be stated as follows: given a finite group S(G) of orthogonal matrices that generates a group code [M,n] by operating on an initial unit vector X, among all such vectors X find out the vector X_o for which the minimum distance is the greatest possible. We have to find the maximum of the minimum of the distances, i.e. to determine a kind of saddle point with respect to the continuous variable X and discrete variable g:

$$\max_{X} \left[\min_{g \neq g'} d(D(g')X, D(g)X) \right]$$

where the maximum is taken over all the vectors of R^n with the constraints $\|X\|=1$ and $S(H)X=X$. S(H) is a subgroup of S(G), possibly $H=\{e\}$. At the present time no general solution is known. The problem has been solved for many classes of group codes and for codes generated by special representations. Djokovic and Blake, [25], settled the case of full homogeneous component; Downey and Karlof found all the optimal group codes in three dimensions [28]; Biglieri and Elia identified the

optimal initial vector for Variant I permutation codes, [9], and showed that for cyclic codes [8] as well as for abelian codes the optimal initial vector is obtained by solving a linear programming problem. Nevertheless, the evidence so far is that the problem cannot have, in general, a closed form solution.

We do not digress on the meaning of "solution", but we adopt the pragmatic view that for practical purposes any kind of numerical solutions should be regarded as a valid one.

For computational approaches the initial vector problem can be stated, in general, as a mathematical problem with a quadratic objective subjected to quadratic constraints, [37].

Let d_o^2 be the minimum square distance. The optimal initial vector X_1 is the solution to:

$$d_o^2 = \text{Max Min } d^2(D(g)X_1, X_1)$$

where the maximum is taken over all unit vectors and the minimum is on all elements $g \in G$ different from the identity.

For any unit vector X and unitary matrix D(g), we have

$$d^2(D(g)X,X) = 2 - 2(D(g)X,X).$$

Thus maximizing the minimum distance is equivalent to minimizing the maximum inner product. We may assume the maximum inner product positive and equal to r^2. Let $Y=(1/r)X_1$. Then, for all non identity elements of G, $(D(g)Y,Y) \leq 1$ and $(Y,Y)=1/r^2$. Hence Y is a solution to:

Find Max (Y,Y)

subject to $(D(g)Y,Y) \leq 1$ whenever g is not the identity in G.

The problem of the initial set of vectors for GGA is more complicated, of course, than for group codes because more than one vector is to be found and different objectives may motivate the choice. In this case one formulation of the initial set vector problem is the following:

> Given S(G) find a set $\{X_1, \ldots, X_K\}$ of K n-dimensional vectors with average square norm equal to E, such that the generated GGA is regular and such that the minimum distance is as large as possible.

Here we do not treat the subject further, as the discussion would be very long. For example GGA used in conjuction with error control codes hopefully must have the maximum possible minimum intradistance associated to a given fair partition.

In this context the open problems are countless; the few known solutions either are heuristic or obtained by hand manipulations. Much work must still be done.

VII - THE CONSTRUCTIVE VIEW

One important intent of the group code theory is to produce good point constellations for the design of digital signals to be used in data transmission, vector quantization, pattern recognition or in many other fields. A second and ambitious objective of this theory is the systematic classification and construction of all regular point constellations in n-dimensional spaces. Before discussing the capabilities of the constructive methods of group coding theory, we present three interesting point constellations that have large minimum distances and provide a good instance of this matter.

The first example is given by the [8,3] group code which is the classical constellation shown in Fig.2, (edges connect points at minimum distance), that has a minimum distance slightly greater than the cube. It is generated by the action of the representation of the cyclic group C_8.

The group is generated by:

$$D(g) = (-1)^h \oplus \begin{vmatrix} \cos(\pi h/4) & \sin(\pi h/4) \\ -\sin(\pi h/4) & \cos(\pi h/4) \end{vmatrix}$$

The initial vector is $(\sqrt{(\ 1/(2\sqrt{2}+1))},\ \sqrt{(\ 2\sqrt{2}/(2\sqrt{2}+1))},\ 0)$

The minimum distance is $d^2_{min} = 4/(2 + 1/\sqrt{2}) > 4/3$

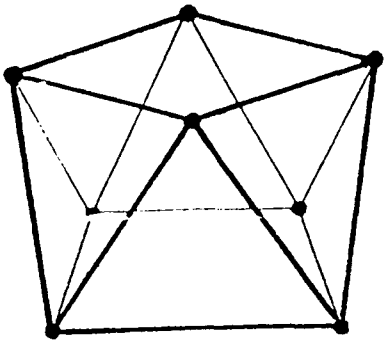


Fig.2

The second example is a not regular and not equienergetic GGA having 14 points in 3 dimensions. The configuration shown in Fig.3, is generated by the action of a representation of the group of the cube

$$C_2 \times C_2 \times C_2 \times S_3 \ .$$

The initial set is $\{(u, 0, 0), (v, v, v)\}$, where

$$v = \sqrt{7\ (7 - 2\sqrt{2})/123} \qquad u = \sqrt{7\ (13 + 8\sqrt{2})/123}$$

The minimum distance is $d^2_{min} = 28\ (7 - 2\sqrt{2})/123 = 0.9496$ and it is significantly greater than 0.93386, the minimum distance of the best known spherical 14 point configuration.

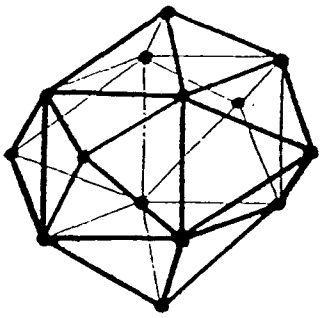


Fig.3

Finally, the third and last example is the [16,4] group code generated by the action of a representation of the abelian group $C_2 \times C_8$. The configuration is shown in Fig.4. The representation is generated by

$$D(g) = (-1)^k \otimes (-1)^{h+k} \otimes \begin{pmatrix} \cos(\pi h/4) & \sin(\pi h/4) \\ -\sin(\pi h/4) & \cos(\pi h/4) \end{pmatrix} \qquad \begin{matrix} k=1,2 \\ \\ h=1,\ldots,8 \end{matrix}$$

The initial vector is $(\sqrt{((\sqrt{2}-1)/2)}, \sqrt{((\sqrt{2}-1)/2)}, \sqrt{(2-\sqrt{2})}, 0)$.

The minimum distance is $d^2_{min} = 2\,(2-\sqrt{2}) = 1.1716$

Note that one of the most used point constellations, the two dimensional 16-QAM has minimum square distance 2/5= 0.4.

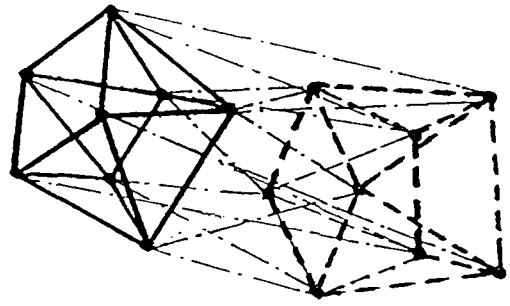


Fig.4

The ingredients involved in the constructive aspect of group codes are groups, matrices and imagination. Four remarkable achievements are particularly important:

1) an old theorem by Jordan stating that the number of finite groups with trivial maximal normal abelian subgroup, which have an irreducible representation of dimension n, is finite and upper bounded by $b(n)= n!\ 6^{\pi(n)\,(n-1)+2}$, where $\pi(n)$ counts the number of primes less than n;
2) the recent classification of all finite simple groups;
3) the fact that the number of finite groups of given order is finite;

4) the complete classification of all commutative groups as well as their representations.

Finite simple groups, Galois' fundamental discovery, are instrumental in building up all other groups and their representations. Abelian groups together with finite group having trivial center can be used to classify all groups which have a representation in n-dimensional spaces. In this context it is useful to recall the outstanding theorem of the classification of finite groups, completed in 1981. This theorem resulted from the global efforts of several hundred mathematicians from all-over the world over a period of 100 years. It is remarkable by itself and relevant to the classification of group codes.

Theorem 15. [20]

The finite simple groups are to be found among:

i) the cyclic groups C_p of prime order p.

ii) the alternating groups A_n of degree n at least 5.

iii) the Chevalley groups

iv) the Tits group

v) the 26 sporadic simple groups.

The Mathieu group, usually denoted by M_{24}, played a central role in the discovery of all 26 sporadic groups. M_{24} is also important in the theory of error-correcting codes, because it is the automorphism group of the Golay code (24,12,8), the only binary perfect multiple error correcting code; see [39,49,21].

Even if it is not necessary to resort to the above definitive theorem, simple groups play a basic role in group codes.

Theorem 16.

Let us consider a [M,n] group code generated by a group G through its representation D(g). If M is a prime number then the group code is generated by a cyclic subgroup C_M of G.

Theorem 17.

No [M,n] group codes exists if M is an odd prime and n is odd.

Theorem 18.

A [M,n] group code can be constructed using representations of a cyclic group provided that either

i) n is even and M>2

or

ii) n is odd and M is even.

Theorem 19.

The number of [M,n] group codes, generated by irreducible representations of groups with trivial maximal normal abelian subgroup is finite and bounded by a function of n alone.

Concluding this section we remark that the problem of the existence of group codes for every M and n is very interesting as it concerns the existence of regular configurations of points on n-dimensional spheres, and generalizes the vertex configurations of regular polytopes.
We can summarize the results as follows:

a)	n even	$M \geq n+1$	at least one group code generated by a cyclic group of order M exists
b)	n odd,	M even $\geq n+1$	at least one group code generated by a cyclic group of order M exists
c)	n even	M odd prime	only one group code generated by the cyclic group of order M exists
d)	n odd,	M odd prime	no group code exists
e)	n = 3,	any M	all group codes have been classified by Downey and Karlof. No group codes with M odd exist.

The definitive classification of all group codes is far from complete, so that many open problems and conjectures still deserve attention. Most of these problems are appealing and may produce beautiful results. We recall, by way of sample, two interesting problems that are still open:

- One group code in dimension 5 with M=15 is known to exist, [26]. It is conjectured that it is the only group code in five dimensional space with an odd number of points.
- Brauer [15] and his school have reached the classification of all groups having an irreducible representation in dimension 4 and 5. It would be interesting to find out all group codes in dimension 4 (the useful dimension for today's applications).
  The determination of all group codes [M,5] would also be interesting as well as the classification of [M,7]. The latter is possible due to the complete list of groups with irreducible representation in dimension 7 obtained by Wales [50, 51, 52].

VIII. CONCLUSIONS

The impact of ancient and modern mathematical concepts on manipulation, transmission and storing of information has made a science of fine, intelligent but scattered techniques.

In this paper we reported on group code theory as an application of general results originated from the ancient geometry. The geometric view provides the appropriate framework for dealing with digital signal processing, signal design, vector quantization and in general communication systems. To enhance the importance of this concept in communication we also considered the combination of these alphabets with block or trellis codes. We have not described the interesting connection of lattices, group codes and combined modulation and coding, this beautiful subject is thoroughly developed in the fundamental book [21] by

Conway and Sloane.

In this paper no essentially new results were proposed. However we hope that the presentation of a topic which is earning a prominent position with increasing applications in the new global communication system will be of some interest, especially to young researchers who are looking for fruitful areas of research with high scientific content and useful applications.

We think that group code theory, which may be credited of a long history dated back to ancient regular polyhedra, is a good example of Feller's conception of mathematics [56]. In fact we wish to conclude with Feller's words:

> "The manner in which mathematical theories are applied does not depend on preconceived ideas: it is a purposeful technique depending on, and changing with experience".

REFERENCES

[1] D.Slepian, "Bounds on Communication", Bell System Technical Journal, vol.42, May 1963, pp.681-707.

[2] D. Slepian, "A Class of Binary Signaling Alphabets", BSTJ, n.35, pp.203-234, January 1956.

[3] D.Slepian, "Group codes for the Gaussian channel", Bell System Technical Journal, vol.47, April 1968, pp.575-602.

[4] D.Slepian, "On neighbor Distances and Symmetry in Group Codes", IEEE Trans. on Information Theory, vol.IT-17, September 1971, pp.630-632.

[5] D. Slepian, "Permutation Modulation", Proc. of the IEEE, March 1965.

[6] D. Slepian, "Some comments on Fourier Analysis, Uncertainty and Modelling", SIAM Rew., vol.25, n.3, July 1983.

[7] E.Biglieri amd M.Elia, "On the existence of group codes for the Gaussian channel", IEEE Trans. on Inform. Theory, vol.IT-18, May 1972, pp.399-402.

[8] E.Biglieri, and M.Elia, "Cyclic-group codes for the Gaussian channel", IEEE Trans. on Inform. Theory, vol.IT-22,n.5, September 1976, pp.624-629.

[9] E.Biglieri and M.Elia, "Optimum Permutation Modulation Codes and Their Asymptotic Performance",IEEE Trans. on Information Th., vol.IT-22, n.6, November 1976, pp.751-753.

[10] E.Biglieri and M.Elia, "Configuration matrices of Group Codes for the Gaussian Channel", Int. Symp. on Inform. Theory, Cornell, USA, November 1977.

[11] E.Biglieri and M.Elia, "Multidimensional Modulation and Coding for Bandlimited Channels",IEEE Trans. on Inform. Theory, to be published.

[12] I.F.Blake and R.C.Mullin, "The Mathematical Theory of Coding", Academic Press, New York, 1975.

[13] I.F.Blake, "Distance properties of Group Codes for the Gaussian Channel", SIAM Journal of Applied Math., vol.23, No.3, 1972.

[14] I.F.Blake, "Configuration matrices of group codes", IEEE Trans. on Inform. Theory, vol.IT-20, n.1, January 1974, pp.95-100.

[15] R. Brauer, "Uber endliche lineare Gruppen von Primzahlgrad", Mathematical Annalen, 169, 1967, pp.73-96.

[16] W.Burnside, "Theory of groups of Finite Order", Dover, New York, 1955.

[17] M.Burrow, "Representation Theory of Finite Groups", Academic Press, New York, 1965.

[18] A.R.Calderbank and J.E.Mazo, "A new description of trellis codes", IEEE Trans. on Inform. Theory, vol.IT-30, November 1984, pp.784-791.

[19] A.R.Calderbank, and N.J.A.Sloane, "Four-Dimensional Modulation with an Eight-State Trellis Code", AT&T Tech. Journal, Vol.64, No.5, May-June 1985, pp.1005-1018.

[20] J.H.Conway, R.T.Curtis, S.P.Norton, R.A.Parker, R.A.Wilson, "ATLAS of finite groups", Clarendon Press, Oxford, 1985

[21] J.H.Conway and N.J.A. Sloane, "Sphere-packing, Lattices and Groups", Springer Verlag, New York, 1987, to appear.

[22] H.M.S. Coxeter, "Regular Polytopes", Dover, New York, 1973.

[23] H.M.S. Coxeter, "Regular Complex Polytopes", Cambridge University press, London, 1974.

[24] C.W.Curtis and I.Reiner, "Representations Theory of Finite Groups and Associative Algebras", Wiley, New York, 1966.

[25] D.Djokovic and I.Blake, "An Optimization problem for Unitary and orthogonal Representations of Finite Group", Trans. of the American Math. Soc. 164, 1972.

[26] C.P. Downey and J.K. Karlof, "On the Existence of [M,n] Group Codes for the Gaussian Channel with M and n Odd", IEEE Trans. Inform. Theory, vol.IT-23, no.4, July 1977, pp.500-503.

[27] C.P. Downey and J.K. Karlof, "Odd Group Codes for the Gaussian Channel", SIAM J. Appl. Math., vol.34, no.4, June 1978 pp.715-720.

[28] C.P. Downey and J.K. Karlof, "Computational Methods for Optimal [M,3] Group Codes for the Gaussian Channel", Utilitas Mathematica, vol. 18, March 1980, pp.51-70.

[29] C.P. Downey and J.K. Karlof, "Group Codes for the Gaussian Broadcast Channel with two receivers", IEEE Trans. Inform. Theory, vol.IT-26, no.4, July 1980, pp.406-411.

[30] L.E. Franks, "Signal Theory", Prentice Hall, 1969.

[31] V.V.Ginzburg,"Mnogomerniye signaly dlya nepreryvnogo kanala" Problemy Peredaci Informacii, n.1, 1984, pp.28-46, (in Russian).

[32] I.Hargittai, "Symmetry: Unifying Human Understanding", Pergamon, 1986.

[33] H.Imai, S.Hirakawa, "A new multilevel coding method using error-correcting codes", IEEE Trans. on Inform. Theory, vol.IT-23, 1977, pp.371-377.

[34] I. Ingemarsson, "Commutative group codes for the gaussian channel", IEEE Trans. on Inform. Theory, vol. IT-19, pp.215-219.

[35] I. Ingemarsson, "On the structure of group codes for the Gaussian channel", Report LiTH-ISY-I-0782, Linkoping University, Sweden, 1986.

[36] I.Jacobs, "Comparison of M-ary modulation systems", Bell System Technical Journal, vol.46, May-June 1967, pp.843-864

[37] J.K. Karlof, "Permutation Codes for the Gaussian Channel", Report of Dpt. Math. Sciences, University of North Carolina, Wilmington, 1987.

[38] R. McEliece, "The Theory of Information and Coding", Addison Wesley, 1977.

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

[40] A. Papoulis, "The Fourier Integral and its Applications", McGraw-Hill, 1962.

[41] A. Papoulis, "Signal Analysis", New York, McGraw-Hill, 1977.

[42] W.W. Peterson and E.J. Weldon, "Error-Correcting Codes", MIT Press, Cambrdge, 1981.

[43] C.E. Shannon, "A Mathematical Theory of Communications", BSTJ, vol.27, 1948, pt.I pp.379-423, pt.II pp.623-656.

[44] C.E. Shannon, "Probability of Error for Optimal Codes in a Gaussian channel", BSTJ, vol.38, May 1959, pp.611-656.

[45] Shu Lin and D.J. Costello, "Error Control Coding: Fundamentals and Applications", Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[46] N.J.A.Sloane, "Tables of Sphere Packing and Sperical Codes", IEEE Trans. on Inform. Th., vol.IT-27, n.3, May 1981, pp.327-338.

[47] G.Ungerboeck,"Channel coding with multilevel/phase signals" IEEE Trans. on Inform. Theory, vol.IT-28, January 1982, pp.55-67.

[48] B.L.van der Waerden, "Modern Algebra", vol.1/2, Ungar, New York, 1953.

[49] J.H.Van Lint, "Introduction to Coding Theory", New York, Springer Verlag, 1982.

[50] D.B.Wales, "Finite Linear Groups of prime degree", Canadian Journal of Mathematics, 21, 1969, pp.1025-1041.

[51] D.B.Wales, "Finite Linear Groups of degree seven, I", Canadian Journal of Mathematics, 21, 1969, pp.1042-1066.

[52] D.B.Wales, "Finite Linear Groups of degree seven, II", Pacific Journal of Mathematics, vol.34, N.1, 1970, pp.207-235.

[53] H.Weyl, "Symmetry", Princeton University Press, Princeton, 1952.

[54] J.Wozencraft and I.Jacobs, "Principles of Communication Engineering", Wiley, New York, 1965.

[55] E.Zehavi and J.K.Wolf, "On the performance evaluation of trellis codes", IEEE Trans. on Information Theory, vol.IT-33, n.2, March 1987,pp.196-202.

[56] W.Feller, "An Introduction to Probability Theory and its Applications", vol. I, Wiley, New York, 1968.

Appendix D

Reprinted from

Renato M. Capocelli
Editor

Sequences

Combinatorics, Compression, Security, and Transmission

Springer-Verlag
New York Berlin Heidelberg
London Paris Tokyo Hong Kong

A Note on Addition Chains and some Related Conjectures *

M. Elia and F. Neri
Dipartimento di Elettronica – Politecnico di Torino
I - 10129 Torino – Italy

Abstract

Addition chains are finite increasing sequences of positive integers, useful for the efficient evaluation of powers over rings. Many features of addition chains are considered, and some results related to the still open Scholz-Brauer conjecture are presented.

1 Introduction

In many fields, such as number theory, cryptography, computer science, or numerical analysis, an efficient computation of

$$x^n = xx\ldots x \tag{1}$$

is often required, where n is a positive integer ($n \in \mathcal{Z}$) and x can belong to any set $\mathcal{R}$ (usually a ring) in which an associative multiplication with identity is defined. It was at once observed that the computation of (1) can be obtained through a sequence

$$x, x^2, x^{a_2}, \ldots, x^{a_i}, \ldots, x^n$$

where each element x^{a_i} is the product of two previous ones. It turns out that the nth-power of x can be associated to the sequence of integers

$$1 = a_0 < a_1 < a_2 < \ldots < a_r = n \tag{2}$$

with the property that, for every i, a couple (j, k) can be found, such that

$$a_i = a_j + a_k, \qquad i > j \geq k.$$

*This work was financially supported in part by the United States Army through its European Research Office, under grant n. DAJA45-86-C-0044.

The sequence (2) is called *addition chain* for n. Without loss of generality, the a_i's are assumed to be sorted in ascending order, and with no duplications.

The problems typical of the evaluation of powers have been thoroughly discussed by Knuth [1] and by Borodin and Munro [2]. In particular [1] reports on many problems that are still open and that deserve attention both as research problems and for their importance in many applications.

Let us now recall some examples where the evaluation of powers is a crucial point.

- First of all, the present day widely discussed public key cryptographic scheme proposed by Rivest, Shamir and Adleman [3], requires the search for two large (several hundreds digits) prime numbers p and q, and the evaluation of powers in Z_{pq}, the ring of the residues modulo pq.

- As a second example let us consider the computation of inverses in finite field GF(q); it is well known [4] that the inverse of every non-zero element is given by

$$a^{-1} = a^{q-2}$$

and in many applications the size of q makes this computation as heavy as those required in the previous example.

- As a third example, let us consider the computation of roots in finite fields. Given $a \in \mathrm{GF}(q)$, let k be the root index; we want to compute the expression

$$b = a^{\frac{1}{k}}$$

whenever it exists. A sufficient condition for the existence is that k has an inverse into the ring Z_{q-1}, i.e. there exists an integer $f(k)$ such that

$$kf(k) = 1 \bmod q - 1.$$

Under this condition we have

$$b = a^{f(k)}.$$

If k has not an inverse in Z_{q-1} then more tests on a are needed to know whether its k-th root exists.

- As a final example, the generation of pseudo-random sequences $x_0, x_1, \ldots, x_n, \ldots$ by the purely multiplicative congruential method, using the iterative relation

$$x_{n+1} = ax_n \bmod m,$$

requires multipliers a that are primitive elements in Z_m in order to generate sequences with maximum period. The test for a number to be primitive

may consists in raising the number being tested to quantities related to the factors of $\varphi(m)$ [1].

In many interesting cases these exponents have the same order of magnitude of m, hence they are rather sizable for non trivial periods. Moreover all the operations must be done fully exploiting the finite size registers of the underlying machine if long periodicity is desired (see [5]), so that even the simple multiplication can be fairly costly.

2 Power Evaluations

In this section we discuss the direct and simplest approaches to power evaluations, since they give insight to more tricky theoretical problems.

Several schemes have been proposed and compared, in order to minimize the efforts (i.e. number of multiplications) for evaluating (1), but it seems that none can be definitively preferable in the general case. The choice of a method instead of the other is affected by a number of constraints, aims or available resources, namely:

- the order of magnitude of the exponent n;
- the availability of storage for precomputed tables;
- whether the situation calls for
  1. independent evaluations of the power (1);
  2. evaluations of several powers of the same base x;
  3. evaluations of several powers to the same exponent n.

In this paper we do not pursuit a complete comparison of all these different situations, but we will be interested mainly on the minimum number of products necessary to evaluate (1). In other words we will restrict our attention to the study of the function $l(n)$, defined as

$$\textit{minimum number of products for evaluating the } n\textit{-th power in an associative ring.} \tag{3}$$

At a first sight a very economical evaluation of (1) is obtained by the binary decomposition of the exponent n, which leads to a number of multiplications upper bounded by $2\lfloor \log_2 n \rfloor$. The same decomposition implies the simple but tight lower bound $\lceil \log_2 n \rceil$. Most considerations about the evaluation of powers concern the estimation of tighter upper bounds.

[1] φ is the Euler totient function.

2.1 The right to left binary method

If we write

$$n = \sum_{i=0}^{t} b_i 2^i, \qquad b_i \in \{0, 1\}, \tag{4}$$

where $t = \lfloor \log_2 n \rfloor$, the power (1) can be computed as

$$x^n = \prod_{i=0}^{t} \left(x^{2^i}\right)^{b_i}. \tag{5}$$

Given that the b_i's can be only 0 or 1, raising to b_i is straightforward. We shall call this approach *right to left binary method.*

In (5) t multiplications are required to evaluate the powers

$$x^{2^i}, \qquad i = 1, 2, \ldots, t \tag{6}$$

and one more multiplication is needed for every non zero b_i, $i < t$, leading to a total of

$$\lfloor \log_2 n \rfloor + \nu(n) - 1$$

multiplications, where $\nu(n)$ is the number of 1's in the binary representation of n. The storage required by an implementation of the binary method (5) can be reduced to three memory cells: one to hold the successive powers (6), another to hold n during its decomposition, and an accumulator for the result.

The right to left binary method can be generalized to an m-ary method in the following way [6]. Let

$$t = \lfloor \log_m n \rfloor \tag{7}$$

and consider the m-ary decomposition of the exponent n

$$n = \sum_{i=0}^{t} d_i m^i, \qquad d_i \in \{0, 1, \ldots, m-1\}. \tag{8}$$

This decomposition can be rewritten as

$$n = \sum_{i \in J_1} m^i + 2 \sum_{i \in J_2} m^i + \ldots + (m-1) \sum_{i \in J_{m-1}} m^i, \tag{9}$$

where J_j denotes the set of indices such that the coefficients d_i in (8) are equal to j.

The right to left m-ary method can be described by the following procedure.

$$\begin{aligned}
&\textbf{Step 1.}\ \text{COMPUTE AND STORE} \\
&\qquad x^m, x^{m^2}, x^{m^3}, \ldots, x^{m^t}; \\
&\textbf{Step 2.}\ \text{FOR EVERY } j \in 1 \ldots m-1 \\
&\qquad \text{COMPUTE } \tilde{x}_j = x^{j \sum_{i \in J_j} m^i} \\
&\textbf{Step 3.}\ \text{COMPUTE (1) AS} \\
&\qquad \prod_{j=1}^{m-1} \tilde{x}_j
\end{aligned} \tag{10}$$

Step 1 of procedure (10) requires at most $t\,l(m)$ multiplications, if $l(m)$ is the minimum number of multiplications for raising a number to its m-th power: actually, in the average, not all the terms in Step 1 will be necessary. Raising to j in Step 2 requires $l(j)$ multiplications, while the remaining operations in Steps 2 and 3 can be carried out with no more than $t-1$ multiplications. The total number of multiplications is bounded by

$$t\,l(m) + t - 1 + \sum_{j=2}^{m-1} l(j). \tag{11}$$

2.2 The left to right binary method

Another way of computing (1) is to rewrite the exponent n from (4) by Horner's rule for evaluating polynomials

$$n = b_0 + 2(b_1 + 2(b_2 + 2(b_3 + 2(\ldots + 2b_t)\ldots))).$$

We shall refer to this approach as *left to right binary method*, since a left to right scanning of n's binary representation is required.

The left to right binary method, extended to an m-ary method, is described by the following procedure, based upon the decomposition (8).

$$\begin{aligned}
&\textbf{Step 1.}\ \text{COMPUTE AND STORE} \\
&\qquad x, x^2, x^3, \ldots, x^{m-1}; \\
&\textbf{Step 2.}\ \text{LET } i = t; \\
&\qquad \text{START WITH } x^{d_t}; \\
&\textbf{Step 3.}\ \text{REPEAT} \\
&\qquad\quad \text{LET } i = i - 1; \\
&\qquad\quad \text{RAISE TO THE } m\text{-TH POWER}; \\
&\qquad\quad \text{IF } d_i \text{ IS NOT 0} \\
&\qquad\qquad \text{MULTIPLY BY } x^{d_i}; \\
&\qquad \text{UNTIL } i = 0;
\end{aligned} \tag{12}$$

Table 1: Upper bounds to the number of multiplications in computing (1).

base m	right to left procedure (10)	left to right procedure (12)
2	$2\lfloor \log_2 n \rfloor - 1$	$2\lfloor \log_2 n \rfloor$
3	$3\lfloor \log_3 n \rfloor$	$3\lfloor \log_3 n \rfloor + 1$
4	$3\lfloor \log_4 n \rfloor + 2$	$3\lfloor \log_4 n \rfloor + 2$
5	$4\lfloor \log_5 n \rfloor + 4$	$4\lfloor \log_5 n \rfloor + 3$
6	$4\lfloor \log_6 n \rfloor + 7$	$4\lfloor \log_6 n \rfloor + 4$
7	$5\lfloor \log_7 n \rfloor + 10$	$5\lfloor \log_7 n \rfloor + 5$
8	$4\lfloor \log_8 n \rfloor + 14$	$4\lfloor \log_8 n \rfloor + 6$

Note that a certain amount of storage is necessary for the quantities computed in the first step of the above procedure; moreover the representation base m of n must be available in a left to right order.

Step 1 of procedure (12) requires at most $m-2$ multiplications; actually the x^{d_i} do not need to be computed for those values of d_i not present in the decomposition (8). Each iteration of Step 3 requires at most $l(m)+1$ multiplications, the $+1$ is present only if the i-th d_i is not 0. The total number of multiplications is bounded by

$$m - 2 + t\,(l(m) + 1). \tag{13}$$

2.3 Bounds for l(n)

A lot of work concerns the search of tight bounds for $l(n)$. By comparing the bounds (11) and (13), Table 1 can be built, where t is expressed as in (7). The order of magnitude of the exponent n can be seen to affect the choice of the base m; the optimal m increases with n. As an example, the base 4 should be preferred to the base 2 whenever $n > 128$. Moreover, those bases that are powers of 2 appear somehow optimal, since they lead to comparatively small coefficients for $\lfloor \log_m n \rfloor$ in Table 1.

Even if the left to right m-ary method seems to behave better for large bases m, a careful inspection of the bounds (11) and (13) shows that the bound (11) is weaker, since Steps 1 and 2 of procedure (10) are open to several optimizations both in the case of few and the case of many terms in the decomposition (8).

When $p(>1)$ powers of the same base x are to be evaluated, the right to left method becomes advantageous. In this case, in fact, the precomputations in

Step 1 of both procedures (10) and (12) can be executed only once, so that the bounds

$$t\,l(m) + p\left(t - 1 + \sum_{j=2}^{m-1} l(j)\right) \tag{14}$$

for the right to left method, and

$$m - 2 + p\,t\,(l(m) + 1) \tag{15}$$

for the left to right method, can be derived. The bound (14) is tighter than (15), since the coefficient of p is smaller.

It is known that the bounds presented above are asymptotically (for large n's) equivalent. Considering the left to right binary method, we can write

$$\lceil \log_2 n \rceil \leq l(n) \leq \lfloor \log_2 n \rfloor + \nu(n) - 1. \tag{16}$$

Since $\nu(n) \leq \lceil \log_2 n \rceil$, and

$$\lfloor \log_2 n \rfloor + \lceil \log_2 n \rceil \leq 2\lfloor \log_2 n \rfloor + 1,$$

the bounds (16) can be rewritten as

$$\lceil \log_2 n \rceil \leq l(n) \leq 2\lfloor \log_2 n \rfloor. \tag{17}$$

Considering the m-ary methods, and substituting $t = \lfloor \log_m n \rfloor$ in (11), the number $l(n)$ of multiplications for raising to n, is bounded by the number of multiplications required by the m-ary method which, for $m = 2^s$, takes the form

$$l(n) \leq \left(1 + \frac{1}{s}\right) \lfloor \log_2 n \rfloor + 2^s. \tag{18}$$

If we let $s = \log_2 \log_2 n - 2\log_2 \log_2 \log_2 n$, (18) becomes

$$l(n) \leq \left(1 + \frac{1}{\log_2 \log_2 n}\right) \log_2 n + o\left(\frac{\log_2 n}{\log_2 \log_2 n}\right). \tag{19}$$

This result is due to Brauer [7] and reported by Knuth [1, page 451, Theorem D]. It is as tight as possible because of a probabilistic asymptotic upper bound to $l(n)$, due to Erdos [8], which asserts that the probability that

$$l(n) \leq \log_2 n + (1 - \epsilon)\left(\frac{\log_2 n}{\log_2 \log_2 n}\right) \tag{20}$$

is definitively less than 1 for any $\epsilon > 0$, or, equivalently, that there always are n's for which the inequality (20) is reversed.

Also the lower bound $\lceil log_2 n \rceil$ can be stressed; in fact Schonhage [9] has shown that the following lower bound holds for every n

$$l(n) \geq \log_2 n + \log_2 \nu(n) - 2.13 \qquad (\nu(n) > 4).$$

3 Addition Chains

Addition chains are the tool for solving the problem of computing (1) for a given n with the minimum number of multiplications. Note that this problem is only a particular case of problem (1), in the sense that nothing is said about the cost of deriving $l(n)$; and this cost can exceed by far the cost of computing (1) by anyone of the previously quoted methods. Nevertheless addition chains are useful to the evaluation of powers both from the theoretical standpoint and when several quantities need to be raised to a same fixed exponent.

Addition chains have been formally defined in the introduction as sequences of integers

$$1 = a_0 < a_1 < a_2 < \ldots < a_r = n$$

with the property that, for every i, a couple (j, k) can be found, such that

$$a_i = a_j + a_k, \qquad i > j \geq k. \tag{21}$$

It turns out that if r is the minimum number for which there exists an addition chain of length r for n, then this addition chain is a solution to the problem stated at the beginning, and $l(n) = r$.

It is convenient to define two special classes of addition chains. A *star chain* is defined as in (21) with the stronger constraint $j = i - 1$. An l^0-*chain* is an addition chain with some ***marked*** elements; the condition is that in (21) a_j is the largest marked element less that a_i. It can be shown that

$$l(n) \leq l^0(n) \leq l^*(n), \tag{22}$$

where $l^0(n)$ and $l^*(n)$ are defined in a way similar to $l(n)$, respectively for l^0-chains and star chains.

A lot has been written about addition chains (see [1] for a presentation of the main results), but the problem of finding $l(n)$ is not completely settled, in the sense that $l(n)$ is not known for all n's.

Bounds for the function $l(n)$ were shown in the previous Section.

3.1 Functions Related To Addition Chains

Many interesting functions are related to $l(n)$; here we consider two such functions which are defined as follows.

$$c(r) = \text{minimum integer } n \text{ that } l(n) = r \tag{23}$$

$$d(r) = \text{number of solutions in } n \text{ to the equation } l(n) = r \tag{24}$$

For a generic n, for which $l(n) = r$, the following bounds hold

$$2^{r/2} \leq c(r) \leq 2^r; \tag{25}$$

the upper bound is straightforward from the definition (23) of $c(r)$, while the lower bound comes from the upper bound in (17). Using the results shown in Table 1, the lower bound can be tightened to the form $2^{F(r)}$, with $F(r) = ar + b$; as an example, exploiting the decomposition to the base 3, we obtain $a = 0.53$ and $b = 0$, which is always tighter than $2^{r/2}$.

Moreover, the same lower bound can be significantly improved using (19); in fact, after some algebraic manipulations, we can obtain the asymptotic bounds

$$2^{r-\frac{r}{\log_2 r}+o\left(\frac{r}{\log_2 r}\right)} \leq c(r) \leq 2^r. \tag{26}$$

From this and the previous relations the asymptotic behavior of the function $c(r)$ will be

$$c(r) = 2^r + o(2^r).$$

From (25), and from the definition of $d(r)$, the following inequality can be stated

$$d(r) < 2^r - c(r) + 1;$$

hence

$$d(r) + c(r) < 2^r + 1.$$

It is likely to conjecture that $d(r)$ behaves asymptotically as an r-th power of 2:

$$d(r) = O\left(2^{d_r r}\right),$$

where d_r is a constant close to 1.

The known values of $c(r)$ and $d(r)$ for small values of r, taken from Knuth [1], are shown in Table 2 where, for sake of comparison, some of the bounds derived in this Section are also reported.

4 The Scholz-Brauer Conjecture

A famous problem concerning addition chains is the Scholz-Brauer conjecture [10]. This conjecture refers to the chains for $2^n - 1$, which are of special interest, since they are the worst case for the binary method (their binary representation is a string of 1's). Let us call a number n satisfying the inequality

$$l(2^n - 1) \leq n - 1 + l(n), \tag{27}$$

where $l(n)$ is defined in (3), a *SB-number*.

Table 2: $c(r)$, $d(r)$ and related bounds.

r	$2^{r/2}$	$2^{F(r)}$	$d(r)$	$c(r)$	$\nu(c(r))$	2^r
1	1.41	2	1	2	1	2
2	2	2.83	2	3	2	4
3	2.82	4	3	5	2	8
4	4	5.66	5	7	3	16
5	5.66	8	9	11	3	32
6	8	11.31	15	19	3	64
7	11.31	16	26	29	4	128
8	16	22.63	44	47	5	256
9	22.63	32	78	71	4	512
10	32	45.25	136	127	7	1024
11	45.25	64	246	191	7	2048
12	64	101.6	432	397	5	4096
13	90.5	161.3	772	607	7	8192
14	128	256	1382	1087	7	16384
15	181.0	406.4	2481	1903	9	32768
16	256	645.1		3583	11	32768
17	362.0	1024		6271	9	32768
18	512	1625		11231	11	32768

The longstanding Scholz-Brauer conjecture states that

all positive integers are SB-numbers.

In the following, it will be shown that (27) holds for infinitely many n's. Let us recall some of the properties of $l(n)$, reported from [1]; they will be useful in the sequel.

$$l(nm) \leq l(n) + l(m); \tag{28}$$

$$l(2^a) = a; \tag{29}$$

$$l(2^a + 2^b) = a + 1 \qquad \text{if } a > b \geq 0; \tag{30}$$

$$l(2^a + 2^b + 2^c) = a + 2 \qquad \text{if } a > b > c \geq 0 \tag{31}$$

(this is Theorem B in [1]);

$$a + 2 \leq l(2^a + 2^b + 2^c + 2^d) \leq a + 3 \qquad \text{if } a > b > c > d \geq 0,$$

where $n = 2^a + 2^b + 2^c + 2^d$ is said to be *special* (see [1, p.449]) if the lower bound holds with equality (this is called Theorem C in [1]);

$$l^0(2^n - 1) \leq n - 1 + l^0(n); \tag{32}$$

this implies that the Scholz-Brauer conjecture holds for l^0-chains (the result, due to Hansen, is called Theorem G in [1]).

Lemma 1 *If $l(n) = l^*(n)$ then n is a SB-number.*

Proof - Straightforward from (22) and (32).
□

Lemma 2 *For every integers a and k, the following inequality holds*

$$l\left(\frac{2^{k2^a}-1}{2^k-1}\right) \leq k2^a - k + a. \tag{33}$$

Proof - It is direct to verify (33) for $a = 0$. Now let us suppose (33) is satisfied for $a-1$; thus, using (28) and (29), we have

$$\begin{aligned} l\left(\frac{2^{k2^a}-1}{2^k-1}\right) &= l\left(\left(\frac{2^{k2^{a-1}}-1}{2^k-1}\right)\left(2^{k2^{a-1}}+1\right)\right) \leq \\ &\leq l\left(\frac{2^{k2^{a-1}}-1}{2^k-1}\right) + l\left(2^{k2^{a-1}}+1\right) \leq \\ &\leq k2^{a-1} - k + a - 1 + k2^{a-1} + 1 \leq \\ &\leq k2^a - k + a. \end{aligned}$$

The validity of (33) for every a follows from the induction principle.
□

Note that the recursive argument used in the proof above also defines, in case of k=1, an addition chain which contains numbers of the form

$$2^\ell\left(2^{2^h}-1\right) \qquad 0 \leq \ell \leq 2^h; \quad 1 \leq h \leq a-1. \tag{34}$$

For later use, we state this point as a Corollary.

Corollary 1 *There exists an addition chain for $2^{2^a}-1$ of length 2^a-1+a, such that it contains the numbers (34). This addition chain has the form*

$$\ldots, (2^{2^h}-1), 2\left(2^{2^h}-1\right), \ldots, 2^{2^h}\left(2^{2^h}-1\right), (2^{2^{h+1}}-1), \ldots$$

Note that

$$2^{2^h}(2^{2^h}-1) + (2^{2^h}-1) = 2^{2^{h+1}} - 2^{2^h} + 2^{2^h} - 1 = 2^{2^{h+1}} - 1.$$

Theorem 1 *For every positive integer n the inequality*

$$l(2^n - 1) \leq n - 2 + \nu(n) + \lfloor \log_2 n \rfloor \tag{35}$$

holds.

Proof - By decomposing n into its binary representation as in (4), we can write

$$2^n - 1 = 2^{\sum_{j=0}^{t-1} b_j 2^j}(2^{b_t 2^t} - 1) + 2^{\sum_{j=0}^{t-2} b_j 2^j}(2^{b_{t-1} 2^{t-1}} - 1) + \ldots + (2^{b_0} - 1) =$$

$$= \sum_{j=0}^{t} 2^{\sum_{i=0}^{j-1} b_i 2^i}(2^{b_j 2^j} - 1). \tag{36}$$

Applying Corollary 1 it can be seen that all the $\nu(n)$ terms in the summation but the first are in the chain for $(2^{2^t} - 1)$, whose length, according to Lemma 2, is bounded by $2^t - 1 + t$. Since the first factor in the first term can be expressed as 2^{n-2^t}, it accounts for at most $n - 2^t$ multiplications. Combining these two contributions with the $\nu(n) - 1$ additional multiplications required by the $\nu(n)$ not zero terms in the decomposition (36), the Theorem is proved.
□

Corollary 2 *If $l(n) = \lfloor \log_2 n \rfloor + \nu(n) - 1$ then n is a SB-number.*

Theorem 2 *Every n such that $\nu(n)$ is not greater than 4 is a SB-number.*

Proof - The proof of Theorem 2 is given separately for the four cases $\nu(n) = 1, \ldots, 4$.

Case $\nu(n) = 1$ - Proved in Lemma 2 with $k = 1$.

Case $\nu(n) = 2$ - It must be shown that, for every integer a and b such that $a > b \geq 0$, the following inequality holds

$$l(2^{2^a + 2^b} - 1) \leq 2^a + 2^b + a.$$

We can write

$$2^{2^a + 2^b} - 1 = 2^{2^b}(2^{2^a} - 1) + 2^{2^b} - 1.$$

From Corollary 1 we know that $2^{2^b} - 1$ belongs to the addition chain ending in $2^{2^a} - 1$, so that, using Lemma 2 we have

$$l(2^{2^a + 2^b} - 1) \leq l(2^{2^a} - 1) - 2^b + 1 \leq 2^a + 2^b + a.$$

Case $\nu(n) = 3$ - It must be shown that, for every a, b and c such that $a > b > c \geq 0$, the following inequality holds

$$l(2^{2^a+2^b+2^c} - 1) \leq 2^a + 2^b + 2^c + a + 1.$$

In a way similar to the case $\nu(n) = 2$, using Corollary 1 and Lemma 2, the proof stems from the equality

$$2^{2^a+2^b+2^c} - 1 = 2^{2^b+2^c}(2^{2^a} - 1) + 2^{2^c}(2^{2^b} - 1) + 2^{2^c} - 1.$$

Case $\nu(n) = 4$ - Two subcases must be considered: $l(n) = a + 3$ and $l(n) = a + 2$. In the first case the proof follows from Theorem 1. In the second case it follows from Exercise 13 in [1, p. 463] — showing that n has a star chain so that Lemma 1 applies — and (32).

□

4.1 Generalizing the Scholz-Brauer Conjecture

The numbers n with all 1's in their binary representation behave much better than bound (19). In fact for numbers of the form $2^n - 1$, since $\log_2 n \geq \nu(n) - 1$, the inequality (35) can be rewritten as

$$l(2^n - 1) \leq n - 1 + c \log_2 n, \tag{37}$$

where c is a convenient constant $1 \leq c \leq 2$. The second term at the right hand side of (20), in this case, has the form

$$\frac{\log_2(2^n - 1)}{\log_2 \log_2(2^n - 1)} \approx \frac{n}{\log_2 n}$$

and, for large n's, the inequality

$$c \log_2 n < \frac{n}{\log_2 n}$$

holds.

Improvements on the upper bound for $l(n)$ are shown by numbers which have some regular patterns in their binary representation. As an example we consider the following Theorem

Theorem 3 *For every positive integer M of the form*

$$M = \sum_{i=0}^{t-1} 2^i + 2^t \sum_{i=0}^{s-t} b_{i+t} 2^i = (2^t - 1) + 2^t M_1 \tag{38}$$

the following upper bound holds

$$l(M) \leq s - 2 + \nu(M_1) + \nu(t) + \lfloor \log_2 t \rfloor \tag{39}$$

Proof - The proof, applying Theorem 1, is straightforward.
□

Along the same lines, if we consider numbers of the form

$$M = 1 + 2^k + 2^{2k} + \cdots + 2^{(t-1)k},$$

then for $t = 2^A$, Lemma 2 shows that

$$l(M) \leq kt - k + l(t)$$

and the following Theorem 4 shows that this inequality also holds for every t such that $\nu(t) \leq 3$.

Theorem 4 *For every integers k and t, such that $\nu(t)$ is not greater than 3, the following inequality holds*

$$l\left(\frac{2^{kt}-1}{2^k-1}\right) \leq kt - k + A + l(t). \tag{40}$$

Proof - The proof is given separately for the three cases $\nu(t) = 1, 2, 3$.

Case $\nu(t) = 1$ - Proved in Lemma 2.

Case $\nu(t) = 2$ - Let $t = 2^A + 2^B$, with $A > B \geq 0$. We can write

$$l\left(\frac{2^{k(2^A+2^B)}-1-2^{k2^B}+2^{k2^B}}{2^k-1}\right) = l\left(\frac{2^{k2^B}-1}{2^k-1} + 2^{k2^B}\frac{2^{k2^A}-1}{2^k-1}\right). \tag{41}$$

Due to (28) and (29), and to Lemma 2,

$$\begin{aligned} l\left(2^{k2^B}\frac{2^{k2^A}-1}{2^k-1}\right) &\leq l\left(2^{k2^B}\right) + l\left(\frac{2^{k2^A}-1}{2^k-1}\right) \leq \\ &\leq k2^B + k2^A - k + A \end{aligned}$$

Since the addition chain for $\frac{2^{k2^A}-1}{2^k-1}$ contains $\frac{2^{k2^B}-1}{2^k-1}$, due to Corollary 1, and only one more product is needed for the two terms inside the right hand side of (41), we can write

$$\begin{aligned} l\left(\frac{2^{k(2^A+2^B)}-1}{2^k-1}\right) &\leq k2^B + k2^A - k + A + 1 = \\ &= k(2^A + 2^B) - k + (A+1). \end{aligned}$$

Case $\nu(t) = 3$ - Let $t = 2^A + 2^B + 2^C$, with $A > B > C \geq 0$. We can write

$$l\left(\frac{2^{kt}-1}{2^k-1}\right) = l\left(2^{k(2^B+2^C)}\frac{2^{k2^A}-1}{2^k-1} + 2^{k2^C}\frac{2^{k2^B}-1}{2^k-1} + \frac{2^{k2^C}-1}{2^k-1}\right).$$

In a way similar to the case $\nu(t) = 2$, using (28) and (30), and Lemma 2, we can obtain

$$l\left(\frac{2^{k(2^A+2^B+2^C)}-1}{2^k-1}\right) \leq k(2^A + 2^B + 2^C) - k + A + 2.$$

□

We can now propose a generalization of the Scholz-Brauer conjecture in the form

for every k and for every n the following inequality

$$l\left(\frac{2^{kn}-1}{2^k-1}\right) \leq kn - k + l(n)$$

holds.

Note that, for $k = 1$, it reduces to the original conjecture.

5 Conclusions

Knuth reports that $1 \leq n \leq 18$ and sporadic 20, 24 and 32 are SB-numbers with equality satisfied; moreover he has shown by computer search that $l(n) = l^*(n)$ for all integers less than 12509. As a consequence of Lemma 1, 12509 can be assumed to be the first non SB-number.

An infinity of SB-numbers exists but it is an open question to prove the Scholz-Brauer conjecture either in the generalized form or not.

Finally, as a consequence of the results presented in this paper, an even more interesting open question seems to be *find the smallest value of c such that (37) holds for every n.*

References

[1] D. E. Knuth, *The Art of Computer Programming*, vol. II, Addison-Wesley, Reading Massachussetts, 1981, pp. 441-466.

[2] A. Borodin, I. Munro, *The Computational Complexity of Algebraic and Numeric Problems*, American Elsevier Pub., New York, 1975.

[3] R. Rivest, **A. Shamir, L. Adleman,** A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Comm. of ACM,* vol. 21, Feb. 1978, pp. 120-126.

[4] R. J. McEliece, *Finite Fields for Computer Scientists and Engineers,* Kluwer Academic Pub., Boston, 1987.

[5] M. Elia, F. Neri, Generation of Pseudorandom Independent Sequences, *Proceedings of the IASTED International Symposium MIC '86,* Innsbruck (Austria), Feb. 18-21, 1986, M. H. Hanza ed., Acta Press, pp. 25-28.

[6] A. Chi-Chih Yao, On the Evaluation of Powers, *SIAM J. Comput.,* Vol. 5, No. 1, Mar. 1976, pp. 100-103.

[7] A. Brauer, *Bull. Amer. Math. Soc.*, 45 (1939), pp. 736-739.

[8] P. Erdos, Remarks on Number Theory, III: On Addition Chains, *Acta Arithm.*, 6 (1960), pp. 77-81.

[9] A. Schonhage, *Theoretical Comp. Sci.*, 1 (1975), pp. 1-12.

[10] A. Scholz, *Jahresbericht der Deutschen Mathematiker-Vereinigung,* (II), 47 (1937), pp. 41-42.

Appendix E

A Note on the Complete Decoding of Kerdock Codes *†

M. Elia, C. Losana and F. Neri
Dipartimento di Elettronica
Politecnico di Torino – Italy

Abstract

A representation of the Kerdock code $K(m)$ is given that allows instantaneous encoding and the use of different complete decoding strategies. Applications to error correction and to vector quantisation are described. The particularly interesting code $K(4)$ is thoroughly analysed and the associated bit error rate on the binary symmetric channel is found in closed form.

1 Introduction

Kerdock codes $K(m)$ are nonlinear codes having many interesting properties, such as high error correcting capabilities, high symmetry and beautiful descriptions. They may be viewed in some way as dual codes of Preparata codes $P(m)$, another noteworthy class of nonlinear codes. The code $K(4)$ is very interesting because besides the relatively high rate 1/2, it coincides with the Preparata code $P(4)$, so that it looks like a sort of self-dual nonlinear code.

The most obvious application of Kerdock codes is their use as channel codes in communication systems. $K(4)$ may also be viewed as the Nordstrom-Robinson code N_{16}, and used as a vector quantizer for encoding random waveforms such as in the case of speech Linear Predictive Coding (LPC) at the rate of 1/2 bit per sample [3]. In a similar way Kerdock codes

†This paper was presented at IEEE International Symposium on Information Theory, Kobe, JAPAN, June 1988.

*This work was financially supported in part by the United States Army through its European Research Office, under grant n. DAJA45-86-C-0044.

i	A_i
0	1
$2^{m-1} - 2^{m/2-1}$	$2^m(2^{m-1} - 1)$
2^{m-1}	$2^{m+1} - 2$
$2^{m-1} + 2^{m/2-1}$	$2^m(2^{m-1} - 1)$
2^m	1

Table 1: Weight distribution for $K(m)$.

allow the decoding from data produced by soft demodulation. In both vector quantization and soft data decoding the problem is to minimize an objective function, which most frequently is taken to be the squared-error distortion.

In Section 2 a systematic representation of $K(m)$ is given that allows instantaneous encoding and the application of different strategies for a complete decoding, as it will be described in Section 3. The application of $K(4)$ to vector quantization will also be described in Section 3. A short analysis of the computational complexity pertaining to the above applications of $K(m)$ will be given in Appendix A. Finally, Section 4 reports some results on the performance evaluation of $K(4)$ used as a channel code on the Binary Symmetric Channel (BSC).

2 A representation for $K(m)$

In this Section we briefly recall the formal definition of Kerdock codes in order to introduce a systematic encoding scheme. We also collect some of its general properties for easy reference.

The Kerdock code $K(m)$, m even, is a nonlinear code consisting of the Reed-Muller code of parameters $(2^m, m+1, 2^{m/2})$ and $2^{m-1} - 1$ cosets of $\mathcal{R}(1,m)$ in $\mathcal{R}(2,m)$. $K(m)$ is also denoted by $[2^m, 2^{2m}, 2^{m-1} - 2^{m/2-1}]$. Important features of any code are the weight and the distance distributions. The weight distribution of a $[n, M, d]$ code is the set $\{A_i\}_{i=0}^n$, where A_i denotes the number of codewords of weight i, while the distance distribution is the set $\{B_i\}_{i=0}^n$, where $M\,B_i$ is the number of ordered pairs of codewords such that the distance between them is i. Linear codes have $B_i = A_i$ and the same property is shown by Kerdock codes. The weight and distance distribution of $K(m)$, taken from [1], is given in Table 1.

Let $c_{\mathcal{R}}$ be a vector of $\mathcal{R}(1,m)$. For later use it is convenient to interpret

$c_{\mathcal{R}}$ according to the following decomposition

$$c_{\mathcal{R}} = (\underset{\leftarrow m+1 \rightarrow}{x} \mid \underset{\leftarrow m-1 \rightarrow}{z_1} \mid \underset{\leftarrow 2^m-2m \rightarrow}{z_2}). \qquad (1)$$

Let w_i be a coset leader that performs a translation of $\mathcal{R}(1,m)$ to generate a codeword c of $\mathcal{K}(m)$, i.e.

$$c = w_i + c_{\mathcal{R}};$$

the code $\mathcal{K}(m)$ is the union of disjoint cosets of $\mathcal{R}(1,m)$, written as follows

$$\mathcal{K}(m) = [w_1 + \mathcal{R}(1,m)] \cup [w_2 + \mathcal{R}(1,m)] \cup \ldots \cup [w_{2^m} + \mathcal{R}(1,m)]. \qquad (2)$$

The definition of $\mathcal{K}(m)$ strongly lies on the choice of the w_i's, $i = 1, \ldots, 2^{m-1}$, which may be obtained by means of primitive idempotents for length $2^{m-1}-1$, or by using simplectic forms to define a convenient set of boolean functions. A very simple construction of $\mathcal{K}(4)$ is given in [2] were the cosets leaders are defined through simplectic forms, very easy to obtain, on four variables.

For easy reference, it is convenient to introduce a binary matrix $W^{(m)}$, built with the coset leaders w_i written by rows, an example of which will be given in (5).

We use a systematic $\mathcal{R}(1,m)$ code and its translates, given by coset leaders w_i of a special form, to generate $\mathcal{K}(m)$. The following two Lemmas are the formal support to this representation.

Lemma 1 *In every coset of a systematic linear (n,k,d) code there exists exactly one word with k consecutive zeros in information positions.*

Proof - Let $(i \mid a)$ denote a codeword of a systematic (n,k,d) code, where i is the subvector of information bits. This subvector ranges over the whole set of possible 2^k bit patterns. Given a coset leader $(x \mid y)$, there exists one and only one code vector $(x \mid z)$ such that

$$(x \mid z) + (x \mid y) = (o \mid z + y)$$

is the unique coset's element with k zeros in information positions.
□

Now let $i = (\underbrace{i_1}_{\leftarrow m+1 \rightarrow} \mid \underbrace{i_2}_{\leftarrow m-1 \rightarrow})$ be a vector of $2m$ information bits.

Lemma 2 *In the matrix $W^{(m)}$ there exists a submatrix made of $m-1$ columns whose rows are the 2^{m-1} different binary sequences of $m-1$ bits.*

Proof - It is known [9] that Kerdock codes can be viewed as systematic codes. This means that all the different patterns of $2m$ bits must appear in the $2m$ information positions of the 2^{2m} codewords. As already mentioned, these codewords can be viewed as translations of $\mathcal{R}(1,m)$ due to the 2^{m-1} coset leaders w_i. These leaders, by Lemma 1, can be chosen to have $m+1$ zeros in the $m+1$ information positions of $\mathcal{R}(1,m)$:

$$w_i = (0 \mid x_i \mid y_i),$$

and the codewords of $\mathcal{R}(1,m)$ can be taken in systematic form:

$$c_{\mathcal{R}} = (i_1 \mid z_1 \mid z_2).$$

Therefore every codeword of $K(m)$ will be of the form

$$(i_1 \mid z_1 + x_i \mid z_2 + y_i).$$

According to the observation above, all the subvectors $(i_1 \mid z_1 + x_i)$ must be different for different pairs i_1 and i. In particular

$$(i_1 \mid z_1 + x_i) \neq (i_1 \mid z_2 + x_j)$$

for every $i \neq j$. That is $x_i \neq x_j$ for every $i \neq j$ or, in other words, all the subvectors x_i in the 2^{m-1} vectors w_i's are distinct and range over all the different binary sequences of $m-1$ bits.

□

As a consequence of Lemmas 1 and 2, w_i may be taken of the form

$$w_i = (0 \mid i_2 \mid y),$$

so that the codewords will result of the form

$$c = (\underbrace{i_1}_{\leftarrow m+1 \rightarrow} \mid \underbrace{z_1 + i_2}_{\leftarrow m-1 \rightarrow} \mid \underbrace{z_2 + y}_{\leftarrow 2^m - 2m \rightarrow}). \qquad (3)$$

As noted in [9] the Kerdock code could also be viewed as a strictly systematic code at the cost of loosing the orderly representation reported above.

i	L_i
0	1
1	16
2	120
3	112
4	7

Table 2: Weight distribution of ML correctable error patterns for $K(4)$.

2.1 Application to $K(4)$

The above results applied to $K(4)$ let the generating matrix of the underlying $\mathcal{R}(1,4)$ code be written in the form

$$G_{\mathcal{R}} = (G_1 \mid G_2 \mid G_3) = \begin{pmatrix} 10000 & 111 & 01101001 \\ 01000 & 110 & 11010101 \\ 00100 & 101 & 10110011 \\ 00010 & 011 & 10001111 \\ 00001 & 000 & 01111111 \end{pmatrix}, \tag{4}$$

and correspondently the matrix of coset leaders in the form

$$W^{(4)} = \begin{pmatrix} 00000 & 000 & 00000000 \\ 00000 & 001 & 11100101 \\ 00000 & 010 & 10111001 \\ 00000 & 100 & 11001011 \\ 00000 & 011 & 01010011 \\ 00000 & 101 & 00011101 \\ 00000 & 110 & 00100111 \\ 00000 & 111 & 11111110 \end{pmatrix}. \tag{5}$$

A very interesting feature is the fact that suitable translations of $K(4)$ cover the whole vector space $\mathrm{GF}(2)^{16}$ without overlapping. In fact 256 correctable error patterns ℓ_i have been found by computer search such that the translates $\ell_i + K(4)$ do not overlap and cover the whole space $\mathrm{GF}(2)^{16}$. The weight distribution $\{L_i\}_{i=0}^{2^m}$ of the correctable error patterns is reported in Table 2, where L_i denotes the number of error patterns of weight i. This property shows that Standard Array decoding is possible for $K(4)$, as it will be described in the Section 3.

From Table 2 the fact that $K(4)$ is not quasi-perfect can also be observed.

The property reported above for $K(4)$, can be conjectured to hold for all Kerdock codes $K(m)$:

> *suitable translations of $K(m)$ cover the vector space $GF(2)^{2^m}$ without overlapping.*

2.2 Encoding

The representation (3) shows that instantaneous encoding is possible. In fact the first $m+1$ bits can be transmitted while they enter the encoder. At the $(m+1)$-th bit the remaining parity check bits for the $\mathcal{R}(1,m)$ code, i.e. the vectors z_1 and z_2, can be computed. As the remaining $m-1$ information bits enter the encoder, they are summed with the entries of vector z_1 and transmitted. At that point the coset leader w_i (hence the vector y) is known, such that the remaining parity check bits can be computed as $z_2 + y$ and transmitted.

3 Decoding and Quantization

In this Section some procedures for decoding Kerdock codes and for performing the vector quantization based on Kerdock codes are described. As a consequence of the representation introduced in Section 2, the problem of decoding Kerdock codes may be formulated as follows:

> *given a received word r, find the pair $[\hat{w}_i, \hat{c}_{\mathcal{R}}]$ made of a coset leader and a Reed-Muller codeword, such that the decoded codeword $\hat{c} = \hat{w}_i + \hat{c}_{\mathcal{R}}$ satisfies the chosen decoding criterion.*

Several decision rules may be considered, their main difference lying in the manner adopted to resolve ties whenever more than $\lfloor \frac{d-1}{2} \rfloor$ errors are detected, since these codes are not perfect. In particular two strategies deserve special interest: the Maximum Likelihood (ML) and the Minimum Correction (MC) rules. They are defined as follows.

Maximum Likelihood rule: r is decoded as the codeword $\hat{c}$ that maximizes the conditional probability $p\{c \mid r\}$. On BSC this rule coincides with the minimum distance decoding, i.e. r is decoded as the codeword $\hat{c}$ corresponding to the minimum distance.

Minimum Correction rule: r is decoded as the codeword at the minimum distance if the distance is less or equal to $\lfloor \frac{d-1}{2} \rfloor$; otherwise the information bits are extracted from the received word without any correction attempt.

Four algorithms for decoding Kerdock codes are described in the following, based on the arithmetic in GF(2). They show increasing memory requirements and decreasing computational complexity. Let us remind that the Hamming weight $\mathrm{wt}(x)$ of a vector x is the number of its nonzero components and let us introduce the vector r, decomposed as in (1)

$$r = (r_1 \mid r_2 \mid r_3),$$

that will be referred to as the received vector.

Algorithm 1 (Minimum distance decoding) - The codewords c_i are stored in a table. For every received vector r, the 2^{2m} Hamming distances $y_i = \mathrm{wt}(r - c_i)$, $i = 1, \ldots, 2^{2m}$, are computed and the minimum $\hat{y}_i$ is found. Ties are resolved by random equiprobable choices. The decoded bits $\hat{i}$ are recovered from the corresponding codeword $\hat{c}_i$.

Algorithm 2 (Syndrome decoding: ML rule) - $H_{\mathcal{R}}$, the parity check matrix of $\mathcal{R}(1,m)$, the vectors $z_i = H_{\mathcal{R}} w_i$, $i = 1, \ldots, 2^{m-1}$, and the vectors $u_j = H_{\mathcal{R}} \ell_j$, $j = 1, \ldots, 2^{2^{m-1}}$, are stored. For every r the syndrome $s = H_{\mathcal{R}} r$ is computed. The pair $[\hat{u}_j, \hat{z}_i]$ that sums to s is then found and $\hat{\ell}_j$ is recovered from $\hat{u}_j$. The received vector r is finally decoded as $\hat{c} = r + \hat{\ell}_j$ and the information bits $\hat{i}$ are recovered from $\hat{c}$.

This algorithm must be restricted to $K(4)$, as it takes advantage from the fact that Standard Array decoding is possible (see Section 2.1). As already mentioned, we conjecture that it may also be applied to decode $K(m)$, for every m even.

Algorithm 3 (Syndrome decoding: MC rule) - This is the previous scheme adapted to the MC decoding rule.

$H_{\mathcal{R}}$, parity check matrix of $\mathcal{R}(1,m)$, a table $\mathcal{T}$ of the syndrome vectors $H_{\mathcal{R}} e$ associated to error patterns e of weight not greater than $\left\lfloor \frac{2^{m/2-1}-1}{2} \right\rfloor$, and $G_{\mathcal{R}}$, generating matrix of $\mathcal{R}(1,m)$, are stored. For every r the syndrome $s = H_{\mathcal{R}} r$ is computed. The error pattern $\hat{e}$ is searched in $\mathcal{T}$, using the entry s. If it is found, r is decoded as $\hat{c} = r + \hat{e}$ and the information bits $\hat{i}$ are recovered from $\hat{c}$. Otherwise the first $m+1$ information bits $\hat{i}_1 = r_1$ are taken unmodified, the vector $\hat{a} = G_2^T \hat{i}_1$ is computed and the remaining $m-1$ information bits are obtained as $\hat{i}_2 = r_2 + \hat{a}$.

Algorithm 4 (Tabular decoding) - A table $\mathcal{T}_1$ of the indices j's associated to the error patterns ℓ_j's for every $\mathbf{r} \in \mathrm{GF}(2)^{2^m}$ and a table $\mathcal{T}_2$ of the error patterns ℓ_j's, $j = 1, \ldots, 2^{m-1}$ are stored. For every $\mathbf{r}$ the index j is obtained using Table $\mathcal{T}_1$. The error pattern ℓ_j is read in Table $\mathcal{T}_2$ using j, in order to compute $\hat{\mathbf{c}} = \mathbf{r} + \ell_j$. The information bits $\hat{\mathbf{i}}$ are finally recovered from $\hat{\mathbf{c}}$.

Vector quantization is a field where $K(4)$ has found a valuable application. Let us formally recall the vector quantization problem with minimum squared-error distortion. Let $\boldsymbol{x} = (x_1, ..., x_n) \in R^n$ be the input to the vector quantizer and let $\{\boldsymbol{c}_i\}_{i=1}^N$ be the set of codewords. The problem may be formulated as follows:

> *find the codeword $\boldsymbol{c}_i$ among the N possible ones which minimizes the squared error*
>
> $$\| \boldsymbol{x} - \boldsymbol{c}_i \|^2 = \boldsymbol{x}^T \boldsymbol{x} - 2\boldsymbol{x}^T \boldsymbol{c}_i + \boldsymbol{c}_i^T \boldsymbol{c}_i, \qquad (6)$$
>
> *where if $\boldsymbol{c}_i^T \boldsymbol{c}_i$ is independent of i then the minimum distance is achieved by the codeword $\boldsymbol{c}_i$ yielding the largest scalar product $y_i = \boldsymbol{x}^T \boldsymbol{c}_i$.*

The most efficient algorithms known today for performing the vector quantization using $K(4)$ are based on the Hadamard Transform (HT), whose definition, for easy reference, will now be recalled.

Let H_n denote an Hadamard matrix in Sylvester form, which is a $n \times n$ matrix of $+1$'s and -1's with the property that the scalar product of any two distinct rows is 0. Thus H_n must satisfy the relation

$$H_n H_n^T = nI,$$

where I is the $n \times n$ identity matrix.

An n-dimensional column vector $\boldsymbol{y}$ is called the HT of the vector $\boldsymbol{x}$ if it is obtained multiplying the vector $\boldsymbol{x}$ by an Hadamard matrix, i.e.

$$\boldsymbol{y} = H_n \boldsymbol{x}.$$

In this context we shall consider the i-th binary codeword of $K(m)$ as a vector of $+1$'s and -1's, with $+1$'s replacing 0's and -1's replacing 1's. It is easy to see that this will replace the usual vector sum over GF(2) with the dot component-wise product of integer vectors, hereafter denoted $\odot$.

The scalar and dot products are compatible in the sense that the following property holds

$$x^T(y \odot z) = (x \odot y)^T z. \tag{7}$$

Vector quantization using $K(m)$ requires , by direct application of (6), the computation of 2^{2m} scalar products

$$y_i = x^T c_i, \qquad i = 1, \ldots, 2^{2m}, \tag{8}$$

and $2^{2m} - 1$ comparisons to search the minimum $\hat{y}_i$.

Applying the property (7), y_i may be computed as

$$y_i = x^T(w_j \odot c_{\mathcal{R}}) = (x \odot w_j)^T c_{\mathcal{R}}. \tag{9}$$

As noted in [1,2,3], the 2^{m+1} codewords of $\mathcal{R}(1,m)$ can be grouped to form a Hadamard matrix H_{2^m} and its negative $-H_{2^m}$. Therefore the y_i's can be computed as 2^m-dimensional HT's of the 2^{m-1} vectors $(x \odot w_j)$. Moreover only 2^{2m-1} comparisons are necessary to find the maximum scalar product; the search can be limited to the absolute values $\mid x^T c_i \mid$ and the proper codeword can then be chosen according to the sign of $x^T c_i$.

The above observations can be also applied to the minimum distance decoding of soft data. The computation of Algorithm 1 may be performed by executing the HT's of the received vector r and the companion vectors $r \odot w_j$, $j = 2, \ldots, 2^{m-1}$. In the following, two algorithms that implement the decoding along these lines are described.

Algorithm 5 (HT decoding: ML rule) - The matrix $W^{(m)}$ of the coset leaders and the Hadamard matrix H_{2^m} in Sylvester form are stored. For every r, the 2^{2m} scalar products $y_i = r^T c_i$, $i = 1, \ldots, 2^{2m}$, are computed by performing the 2^m HT's $H_{2^m}(r \odot w_i)$ and $-H_{2^m}(r \odot w_i)$, $i = 1, \ldots, 2m$. The maximum $\hat{y}_i$ is then found, resolving ties by random equiprobable choices. The received vector r is finally decoded as the codeword $\hat{c}_i$, from which the information bits $\hat{i}$ are recovered.

Algorithm 6 (HT decoding: MC rule) - The matrices $W^{(m)}$, H_{2^m} and $G_{\mathcal{R}}$, generating matrix of of $\mathcal{R}(1,m)$, are stored. For every received vector r, the 2^{2m} scalar products $y_i = r^T c_i$, $i = 1, \ldots, 2^{2m}$, are computed by performing the 2^m HT's, $H_{2^m}(r \odot w_i)$ and $-H_{2^m}(r \odot w_i)$, $i = 1, \ldots, 2m$. The maximum $\hat{y}_i$ is then searched. If there are no ties, r is decoded as the codeword $\hat{c}_i$, from which the information bits $\hat{i}$ are recovered. Otherwise the first $m+1$ information bits are taken unmodified ($\hat{i}_1 = r_1$), the vector $\hat{a} = G_2^T \hat{i}_1$ is computed and the remaining $m - 1$ information bits are taken as $\hat{i}_2 = r_2 + \hat{a}$.

m	Direct application $2^{3m-1} - 2^{2m-1}$	Direct FHT $m2^{2m-1}$	Proposed scheme $2^m[3 + (m-2)2^{m-1}]$
4	1,920	512	304
6	129,024	122,88	8,384
8	8,355,840	262,144	197,376

Table 3: Complexity figures for vector quantization with $K(m)$.

An efficient method for computing the HT's required by the above Algorithms is reported in Appendix A, together with computational complexity remarks. The resulting complexity figures are summarized in Table 3.

4 Bit and Word Error Probabilities for $K(4)$

In general it is very hard to compute either bit error rate or word error rate for nonlinear codes. For $K(4)$, however, such a computation is feasible because its structure is very similar to that of linear codes. In fact, as previously observed, the decoding can be organized as a Standard Array, since the translates of $K(4)$ by the correctable error patterns do not overlap and cover the whole vector space of dimension 16 over GF(2). In this case (see [1,6]) the bit error probability p_b and word error probability p_w after complete decoding on the BSC can be expressed as a polynomial in the raw bit error rate p of the BSC:

$$\frac{1}{8}\sum_{i=0}^{16} E_i\, p^i,$$

where the coefficients E_i are reported in Table 4. They have been computed from the Standard Array according to a counting scheme proposed in [1]. Interesting are the asymptotic expressions $p_b \asymp \frac{1464}{8}p^3$ and $p_b \asymp \frac{1329}{8}p^3$ for the ML and MC decoding respectively, as p tends to zero. From these relations it follows that, at least asymptotically, the MC rule is superior to the ML rule as far as the bit error rate is concerned. On the other hand, as expected, the word error probability, whose asymptotic expressions are $p_w \asymp 448p^3$ and $p_w \asymp 504p^3$, shows a better asymptotic expression in the ML case.

i	E_i - bit		E_i - word	
	ML	MC	ML	MC
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	1464	1329	3584	4032
4	−12635	−10856	−32088	−37912
5	52116	40497	140448	175000
6	−130242	−81309	−388080	−512288
7	211196	65510	744480	1047104
8	−218250	110310	−1036728	−1565760
9	117176	−409012	1069376	1753808
10	22836	675936	−820512	−1485792
11	−99288	−704544	463680	950880
12	88496	496376	−187880	−454104
13	−43680	−233184	51744	157528
14	12480	66912	−8688	−37696
15	−1664	−8960	672	5600
16	0	0	0	−392

Table 4: Coefficients for bit and word error rate computation of $K(4)$.

5 Conclusions

In this paper we have dealt with many different properties of Kerdock codes.

A description of Kerdock codes that allows instantaneous encoding was given. This approach leads to the application of two different decoding strategies, i.e. the well known Maximum Likelihood criterion and another one that we have called Minimum Correction rule. Referring to $K(4)$ it has been shown that a Standard Array can be built by translating the set of codewords without overlapping. From the inspection of this Standard Array it turns out that K(4) is not quasi-perfect (see also Table 2). The same Standard Array allows the computation of the bit error rate for $K(4)$ on the binary symmetric channel, with respect to both ML and MC decoding strategies: in this particular case MC is asymptotically superior.

Finally it has been analized a scheme suitable both for decoding and for vector quantization based on $K(m)$. Based upon Hadamard Transforms, it shows very low computational complexity figures. Table 3 compares the number of sums required by the proposed scheme with the standard FHT and the direct application of Algorithm 1 above.

References

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

[2] **J. H. vanLint,** *Coding, Decoding and Combinatorics,* Applications of Combinatorics, R. J. Wilson editor, Shiva Pub., 1982, pp. 67-74.

[3] **J. Adoul,** Fast ML Decoding Algorithm for the Nordstrom-Robinson Code, *IEEE Trans. on Inform. Th.,* vol. IT-33, N. 6, Nov. 1987, pp. 931-933.

[4] **A. M. Kerdock,** A class of low-rate nonlinear codes, *Info. and Control,* 20 (1972), pp. 182-187.

[5] **J. H. Conway and N. J. A. Sloane,** Soft decoding techniques for codes and lattices, including the Golay code and the Leech lattice, *IEEE Trans. Inform. Theory,* vol. IT-32, Jan. 1986, pp. 41-50.

[6] **M. Elia,** A note on the computation of bit error rate for binary block codes, *Journal of Linear Algebra and its Applic.,* Wisconsin, vol. 98, Jan. 1988, pp. 199-210.

[8] **A. Borodin, I. Munro,** *The Computational Complexity of Algebraic and Numeric Problems,* American Elsevier Pub., New York, 1975.

[9] **J. Mykkeltveit,** A note on Kerdock codes, *JPL Technical Report* 32-1526, vol. IX, Jet Propulsion Labs, Pasadena (CA), 1972, pp. 82-83.

A Complexity of soft data decoding and vector quantization based upon $K(m)$

Every dissertation on decoding complexity suffers the lacking of suitable measures of complexity. However for most practical applications the number of arithmetical operations (in any field), the number of logical operations and the amount of storage required can be taken as meaningful figures. In the following we estimate the complexity, in terms of number of arithmetic sums, for decoding and vector quantizing based upon $K(m)$.

In [3], by referring to a definition of $K(4)$ as $\mathcal{N}_{16}$, it is shown that in the vector quantization problem, the nearest neighbor codeword can be found with 30 additions and 128 comparisons. By using similar arguments, based on a variant of the Fast Hadamard Transform (FHT), and using our representation of Kerdock codes, we will introduce a generalization to $K(m)$, that shows the same complexity figure in the case $m = 4$.

It was already shown in Section 3 that the vector quantization problem can be solved with the computation of 2^{m-1} Hadamard transforms of dimension 2^m. The complexity of this computation stems from the following observations, motivated in [1,2,3].

1. The HT of dimension 2^m may be computed by evaluating HT's of smaller dimension. In fact the matrix H_{2^m} may always be written as

$$H_{2^m} = \begin{pmatrix} H_{2^{m-1}} & H_{2^{m-1}} \\ H_{2^{m-1}} & -H_{2^{m-1}} \end{pmatrix}.$$

   This means that a HT of dimension 2^m may be computed by performing two HT's of dimension 2^{m-1} and operating $2 \cdot 2^{m-1}$ sums, and every HT of dimension 2^{m-1} may by obtained from two HT's of dimension 2^{m-2} and $2 \cdot 2^{m-2}$ sums, and so on.

   This observation allows a decomposition of the Sylvester-type matrix H_{2^m} in terms of H_4 submatrices, where the matrix H_4 has the structure

$$H_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

As an example

$$H_{16} = \begin{pmatrix} H_4 & H_4 & H_4 & H_4 \\ H_4 & -H_4 & H_4 & -H_4 \\ H_4 & H_4 & -H_4 & -H_4 \\ H_4 & -H_4 & -H_4 & H_4 \end{pmatrix}.$$

2. The vectors $\boldsymbol{x}$ and $\boldsymbol{w}_i$ can be partitioned into subvectors of dimension 4

$$\boldsymbol{x} = (\boldsymbol{x}_1 \mid \boldsymbol{x}_2 \mid \ldots \mid \boldsymbol{x}_{2^{m-2}})$$

and

$$\boldsymbol{w}_i = (\boldsymbol{w}_{1i} \mid \boldsymbol{w}_{2i} \mid \ldots \mid \boldsymbol{w}_{2^{m-2}i}),$$

and their dot product may be performed independently in each single part

$$\boldsymbol{x} \odot \boldsymbol{w}_i = (\boldsymbol{x}_1 \odot \boldsymbol{w}_{1i} \mid \boldsymbol{x}_3 \odot \boldsymbol{w}_{3i} \mid \ldots \mid \boldsymbol{x}_4 \odot \boldsymbol{w}_{4i}).$$

Note that the action of $\boldsymbol{w}_{ji}$ on $\boldsymbol{x}_j$ is to change the sign of some entry.

3. The HT's of (a_1, a_2, a_3, a_4) and $(a_1, a_2, a_3, -a_4)$ require 12 sums

$$\begin{aligned} &(a_1 + a_2) + (a_3 + a_4) \\ &(a_1 + a_2) + (a_3 - a_4) \\ &(a_1 + a_2) - (a_3 + a_4) \\ &(a_1 + a_2) - (a_3 - a_4) \\ &(a_1 - a_2) + (a_3 + a_4) \\ &(a_1 - a_2) + (a_3 - a_4) \\ &(a_1 - a_2) - (a_3 + a_4) \\ &(a_1 - a_2) - (a_3 - a_4). \end{aligned}$$

4. Given $\boldsymbol{a}^T = (a_1, a_2, a_3, a_4)$, the HT of a vector derived from $\boldsymbol{a}$ by an even number of the sign changes can be obtained from the HT of $\boldsymbol{a}$ by simple permutations and sign changes. If we call $\boldsymbol{b}^T = (b_1, b_2, b_3, b_4)$ the HT of $\boldsymbol{a}$, we have

$$\begin{aligned} (\ a_1,\ a_2,\ a_3,\ a_4) &\leftrightarrow (\ b_1,\ b_2,\ b_3,\ b_4) \\ (-a_1, -a_2, -a_3, -a_4) &\leftrightarrow (-b_1, -b_2, -b_3, -b_4) \\ (-a_1, -a_2,\ a_3,\ a_4) &\leftrightarrow (-b_3, -b_4, -b_1, -b_2) \\ (\ a_1,\ a_2, -a_3, -a_4) &\leftrightarrow (\ b_3,\ b_4,\ b_1,\ b_2) \end{aligned}$$

$$
\begin{aligned}
(-a_1,\ a_2, -a_3,\ a_4) &\leftrightarrow (-b_2, -b_1, -b_4, -b_3)\\
(\ a_1, -a_2,\ a_3, -a_4) &\leftrightarrow (\ b_2,\ b_1,\ b_4,\ b_3)\\
(-a_1,\ a_2,\ a_3, -a_4) &\leftrightarrow (-b_4, -b_3, -b_2, -b_1)\\
(\ a_1, -a_2, -a_3,\ a_4) &\leftrightarrow (\ b_4,\ b_3,\ b_2,\ b_1).
\end{aligned}
$$

5. Due to the form of the matrix $W^{(m)}$, for each block of 4 columns the computation of at least a couple of HT's as in Point 3 above must be done. No other transforms are required due to the observation in Point 4. The total number of sums is therefore

$$12\,\frac{2^m}{4} = 3 \cdot 2^m.$$

6. Due to Point 1 above, the combination of subtransforms to produce $H_{2^m}(\boldsymbol{x} \odot \boldsymbol{w}_i)$ requires the following number of sums

$$2 \cdot 2^{m-1} + 2^2\, 2^{m-2} + \ldots + 2^{m-2}(2^2 = 4) = (m-2)\, 2^m.$$

7. The number of 4-dimensional HT's to be computed is 2^{m-2}, so that the total number of sums is

$$3 \cdot 2^m + 2^{m-1}(m-2)\, 2^m = 2^m\,[3 + (m-2)2^{m-1}].$$

The final expression $2^m\,[3 + (m-2)2^{m-1}]$ gives the number of sums that are sufficient for decoding and vector quantizing based upon $\mathcal{R}(m)$.

Appendix F

Multiplication in Galois Fields $GF(2^m)$ *

Michele Elia and Daniele Vellata
Dipartimento di Elettronica - Politecnico di Torino
I - 10129 Torino – Italy

Abstract

Many data encrypting and data encoding techniques operate in Galois fields and require that the basic arithmetical operations of sum and product be performed as quickly as possible. Here we propose three schemes for computing products in $GF(2^m)$, to be considered alongside the known ones.

1 Introduction

The decoding of multiple error-correcting cyclic codes, [1,2,7], and the encrypting of streams of digital data [4,5] usually operate in appropriate Galois fields. The efficiency of the basic field operations of sum and product is crucial to enable the execution of these processes without affecting the overall system performance. In particular the product of field elements seems to be the most critical operation.

Recently hardware implementations, [8,9], of finite field multipliers have been proposed, which are based on known algorithms [10,11,12]. All these algorithms use, to a different extent, linear feedback shift registers, [6].

Here we introduce some alternative schemes for computing products in $GF(2^m)$ which we believe to be new and that in several cases outperform the known algorithms. In particular:

- **Algorithm I** is the direct interpretation of the product definition; at the cost of some storage it works fast and with no limitations as to the field definition or order.

*This work was financially supported in part by the United States Army through its European Research Office, under grant n. DAJA45-86-C-0044.

- **Algorithm II** is reminescent of the $FastFourierTransform$ speeding up principle, needs less memory than the previous one, but requires a more complex implementation.

- **Algorithm III** is based on a special form of the primitive polynomial defining the basis element for $GF(2^m)$, performs very well but is limited to special values of m.

One of the main concerns in this kind of problems is the balance among different resources or performance requirements. These problems will be shortly debated in the final section. While in the next section we will recall, for sake of easy reference, some useful notations and we will introduce the necessary definitions.

2 Field element representation

An element α of $GF(2^m)$ can be represented either as a power of a primitive element η, that is

$$\alpha = \eta^{L_\eta(\alpha)}$$

where $L_\eta(\alpha)$ denotes an integer number called logarithm of α to base η, or as a polynomial in β of degree $m-1$, where β is root of a polynomial $g(x)$ irreducible over $GF(2)$ of degree m, that is

$$\alpha = \sum_{i=0}^{m-1} \alpha_i \beta^i \qquad \alpha_i \in GF(2)$$

For later use we define the polynomial $\alpha(x)$ associated to α:

$$\alpha(x) = \sum_{i=0}^{m-1} \alpha_i x^i.$$

Note that $\alpha = \alpha(\beta)$.

It is commonly believed that exponential representations are better for computing products while polynomial representations are better for computing sums in $GF(2^m)$, [2]. Really, the matter is slightly different, because multiplication of numbers in the exponential representation requires the execution of sums of integers modulo $2^m - 1$ with the waste of time due to carry propagation. Moreover, in many applications the polynomial representation is unavoidable.

Now let us recall how the product of two numbers α and γ in $GF(2^m)$ may be computed when polynomial representation is used. Writing

$$\delta = \alpha\gamma = \sum_{i=0}^{m-1} \gamma\beta^i \alpha_i$$

we see that δ is computed by summing up $\gamma\beta^i$ whenever $\alpha_i = 1$. The addend $\gamma\beta^i$ can be obtained as the content of a linear feedback shift register, having characteristic polynomial $g(x)$, starting with initial content γ, and performing i steps. Along the same line of the well known algorithm used to evaluate the product of integers, we may represent the products $\gamma\beta^i$, $i = 0, \ldots, n-1$ as an array of dots with the convention that each dot corresponds to a product $\gamma^j\beta^i$. With abuse of language we say that dots on the same column must be added modulo 2 and finally the stream of dots of length $2n-2$ must be reduced to a stream of the n rightmost positions. See fig.1.

Equivalently these operations may be described using the polynomial representation so that the product $\alpha\gamma$ can be computed by first executing the product $\alpha(x)\gamma(x)$ and then reducing the result *modulo* $g(x)$

$$\alpha(x)\gamma(x) = \sum_{i=0}^{2m-2} a_i x^i = \delta(x) + g(x)p(x), \tag{1}$$

hence setting $x = \beta$ to get

$$\alpha\gamma = \delta = \sum_{i=0}^{m-1} \alpha_i \beta^i$$

where $\alpha = \alpha(\beta)$, $\gamma = \gamma(\beta)$ and $\delta = \delta(\beta)$.
Observe that, for later use, the product $\alpha(x)\gamma(x)$ can be written as:

$$\alpha(x)\gamma(x) = a(x) \dotplus x^m b(x). \tag{2}$$

where

$$a(x) = \sum_{i=0}^{m-1} a_i x^i \qquad ; \qquad b(x) = \sum_{i=0}^{m-2} a_{i+m} x^i.$$

3 Algorithms

In this section we describe the algorithms in an abstract form so that the presentation is not influenced by present-day technology. Comparisons with recent implementations will be given in the final section.

Algorithm I. Using equation (1) and substituting $x = \beta$, we get

$$\alpha\gamma = \sum_{i=0}^{2m-2} \alpha_i x^i \mid_{x=\beta} = \sum_{i=m}^{2m-2} \alpha_i \beta^i + \sum_{i=0}^{m-1} \alpha_i \beta^i. \tag{3}$$

In this expression only the sum from m to $2m-2$ must be processed. The reduction process can be very fast if we have previously stored the following $m-1$ powers of β

$$\beta^j = \sum_{i=0}^{m-1} c_{ij} \beta^i \qquad m \leq j \leq 2m-2 \qquad c_{ij} \in GF(2)$$

which allows us to compute δ in a straightforward way

$$\delta = \sum_{i=0}^{m-1} \alpha_i \beta^i + \sum_{i=m}^{2m-2} \alpha_j \sum_{i=0}^{m-1} c_{ij} \beta^i \quad .$$

Algorithm II. To describe this algorithm, which requires less stored data but is slower than Algorithm I, let us consider the first sum in equation (2) and let us suppose that the following power of β is known

$$\beta^n = \sum_{i=0}^{m-1} b_i \beta^i \tag{4}$$

where $n = m - 1 + \lfloor \frac{m-2}{2} \rfloor$. Noting that

$$\beta^{n+j} = \sum_{i=0}^{m-1} b_i \beta^{i+j}$$

the (n+j)-th power can be obtained by shifting j times to the left the sequence $(b_{m-1}, b_{m-2}, ..., b_0)$ $\qquad m - 1 - \lfloor \frac{m-2}{2} \rfloor > j > 0$.

The powers of β whose exponent is between n and $2m-2$, do not need to be stored. In fact using equation (3) and the above observations, the first sum in equation (1) can be reduced to the sum having the maximum power of β less or equal to $n-1$. The next steps consist in repeating these operations successively on the powers of β of exponent $m - 1 + \lfloor \frac{m-2}{2} \rfloor$, $m - 1 + \lfloor \frac{1}{2} \lfloor \frac{m-2}{2} \rfloor \rfloor$, ... , $m - 1 + \lfloor 1/2 \lfloor ...1/2 \lfloor (m-2)/2 \rfloor ... \rfloor$, the number of iterations being $\lfloor log_2 m - 2 \rfloor$.

Algorithm III. This algorithm is based on special forms of the generating polynomial $g(x)$. Here we consider fields that have elements associated to irreducible trinomials of the form $g_\ell(x) = x^m + x^\ell + 1$, where $2 < \ell \leq \lfloor \frac{m}{2} \rfloor$. The cases $g_1(x) = x^m + x + 1$ and $g_2(x) = x^m + x^2 + 1$ will be considered separately both to start and to explain the procedure. In case of $g_1(x)$, we can recast equation (2) as follows

$$\alpha(x)\gamma(x) = a(x) + (1+x)b(x) + (x^m + x + 1)b(x)$$

so that substituting $x = \beta$, we get δ as

$$\delta = a(\beta) + (1+\beta)b(\beta)$$

which is computed in two steps with no storage.

Also $g_2(x)$ presents the same behaviour; in fact we have

$$\begin{aligned} \alpha(x)\gamma(x) &= a(x) + b(x) + x^2[b(x) + b_{m-2}x^{m-2}] + b_{m-2}(x^2+1) + \\ &+ b_{m-2}(x^m + x^2 + 1) + (x^m + x^2 + 1)b(x) \end{aligned} \tag{5}$$

so that substituting $x = \beta$, we get δ as

$$\delta = a(\beta) + b(\beta) + \beta^2[b(\beta) + b_{m-2}\beta^{m-2}] + b_{m-2}(\beta^2 + 1) \tag{6}$$

which is computed in two steps with no storage.

In general we have

$$\alpha(x)\gamma(x) = a(x) + (1 + x^\ell)b(x) + (x^m + x^\ell + 1)b(x).$$

This equation can be conveniently rewritten as

$$\begin{aligned} \alpha(x)\gamma(x) &= a(x) + b(x) + (x^m + x^\ell + 1)b(x) + \sum_{i=0}^{m-2} b_i x^{i+\ell} = \\ &= a(x) + b(x) + (x^m + x^\ell + 1)b(x) + \sum_{i=0}^{m-1-\ell} b_i x^{i+\ell} + \\ &+ (x^m + x^\ell + 1)\sum_{j=0}^{\ell-2} b_{j+m-\ell}x^j + (x^\ell + 1)\sum_{j=0}^{\ell-2} b_{j+m-\ell}x^j \end{aligned} \tag{7}$$

so that substituting $x = \beta$, we get δ as

$$\delta = a(\beta) + b(\beta) + \sum_{i=0}^{m-1-\ell} b_i \beta^{i+\ell} + (\beta^\ell + 1)\sum_{j=0}^{\ell-2} b_{j+m-\ell}\beta^j. \tag{8}$$

which is computed in two steps with no storage.

	Alg. I		Alg. II		Alg. III		Alg. SR		Alg. STP	
m $GF(2^m)$	PS	NS	PS	NS	PS	NS	PS	NS	PS	NS
4	12	3	4	3	0	2	0	6	0	4
8	56	3	16	4	0	2	0	14	0	8
16	240	3	48	5	0	2	0	30	0	16

Table 1: Algorithm comparissons

- PS indicates the required storage measured in bits;
- NS indicates the number of steps between input and output;
- SR stays for Shift Register;
- STP stays for Scott-Tavares-Peppard.

4 Conclusions

This paper presents three schemes for computing products in $GF(2^m)$. The algorithms are not strictly comparable as far as they make use of different resources. As a matter of fact special algorithms for performing products in finite fields have been proposed in the scientific literature. In particular the Massey and Omura multiplier utilizes the normal basis representation of the field elements, while the Berlekamp multiplier uses both the standard and dual bases representations: for both algorithms it is difficult to change the polynomial which generates the field. The algorithm proposed by Scott, Tavares and Peppard, which has been hardware implemented, does not present the previous limits and can be compared with the ones proposed here.
For sake of comparison Table 1 shows for the mentioned algorithms the amount of required storage and the number of steps between input and output.

The facts emerging from this table were confirmed by both software pro-

gramming and hardware implementations. From both programming semplicity and execution time points of view, Algorithm III is undisputably preferable. Its limits stem from the fact that neither primitive irreducible trinomials are available for every m, nor it is known whether an infinite number of such primitive trinomials does exist.

References

[1] **E.R. Berlekamp,** *Algebraic Coding Theory,* McGraw-Hill Book Company, New York, 1968.

[2] **F.J. MacWilliams and N.J.A. Sloane,** *The Theory of Error-Correcting Codes,* Elsevier, New York, 1976.

[3] **W. Diffie and M.E. Hellman, New directions in Cryptography,** IEEE Transactions on Information Theory, vol.IT-22, November 1976, pp.644-654.

[4] **D. Denning,** *Cryptography and Data Security,* Addison-Wesley, Reading MS, 1983.

[5] **N. Koblitz,** *A Course in Number Theory and Cryptography,* Springer-Verlag, New York, 1987.

[6] **R.J.McEliece,** *Finite Fields for Computer Scientists and Engineers,* Kluwer Academic Press, Boston, 1987.

[7] **M. Elia,** *Alegebraic decoding of the (23,12,7) Golay Code,* IEEE Transactions on Information Theory, vol.IT-33, January 1987, pp.150-151.

[8] **B.B.Zhou,** *A new Bit-serial Multiplier over* $GF(2^m)$, IEEE Transactions on Computers, vol.C-37, No.6, June 1988, pp.749-751.

[9] **I.S.Hsu, T.K. Troung, L.J.Deutsch and I.S. Reed,** *A Comparison of VLSI Architecture of Finite Field Multipliers Using Dual, Normal or Standard Bases,* IEEE Transactions on Computers, vol.C-37, No.6, June 1988, pp.735-739.

[10] **E.R. Berlekamp,** *Bit-serial Reed-Solomon encoders,* IEEE Transactions on Information Theory, vol.IT-28, November 1982, pp.869-874.

[11] **C.C.Wang, T.K. Troung, H.M. Shao, L.J.Deutsch, J.K. Omura and I.S. Reed,** *VLSI Architecture for computing multiplications and inverses in* $GF(2^m)$, IEEE Transactions on Computers, vol.C-34, August 1985.

[12] **P.A. Scott, S.E. Tavares and L.E. Peppard,** *A fast multiplier for* $GF(2^m)$, IEEE Journal on Selected Areas in Communications, vol.SAC-4, January 1986.

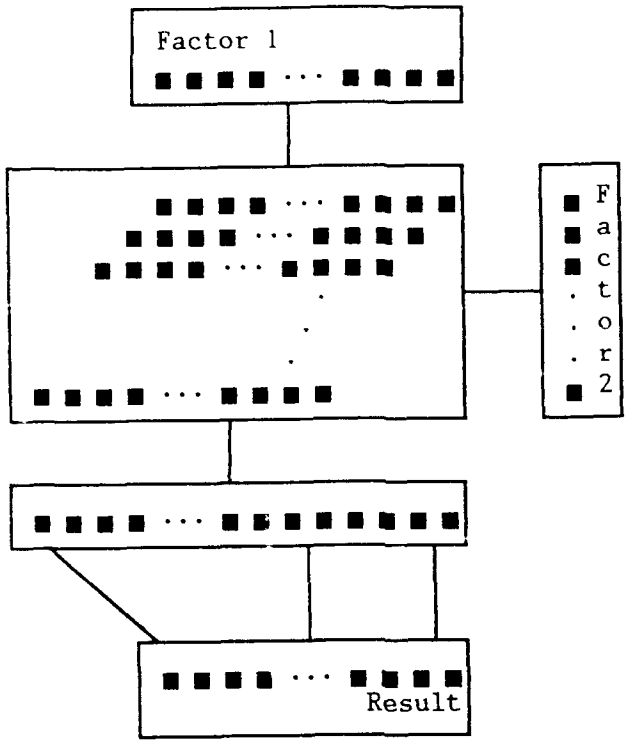


Fig.1 - General scheme of multiplier

Appendix G

On the Concatenation of Binary Linear Codes

Michele Elia * and Fabio Neri
Dipartimento di Elettronica - Politecnico di Torino
I - 10129 Torino – Italy

Abstract

Many recent applications of error-correcting codes, especially in the case of transmission over very noisy channels, have been based on concatenation to achieve high performances. This paper considers concatenation of linear block and convolutional codes, presenting some considerations on the bit error rate computation after complete decoding. The fact that code concatenation is not a commutative operation is discussed.

1 Introduction

The use of error-control codes is steadily increasing in a variety of digital systems like digital recording [2], satellite links [4,5,6] and HF mobile radio transmissions. In many of these applications coding is unavoidable to achieve high performances and often simply to allow the system to work.

In most situations the symbol error probability and the transmission rate are conflicting targets, and the application of efficient and flexible codes is necessary. In these cases the choice of the code is conditioned by two constraints, namely the complexity of the receiving devices and the decoding delay. Code concatenation seems to offer a good compromise in terms of the constraints above.

Furthermore, several applications require uneven protection of the information symbols. This is the case, for example, of packetized information transmissions, where the protocol information carried by packets often requires better protection than the information part. Unequal error protection is a target easily pursued with code concatenation.

*This work was financially supported by the United States Army through its European Research Office, under grant n. DAJA45-86-C-0044.

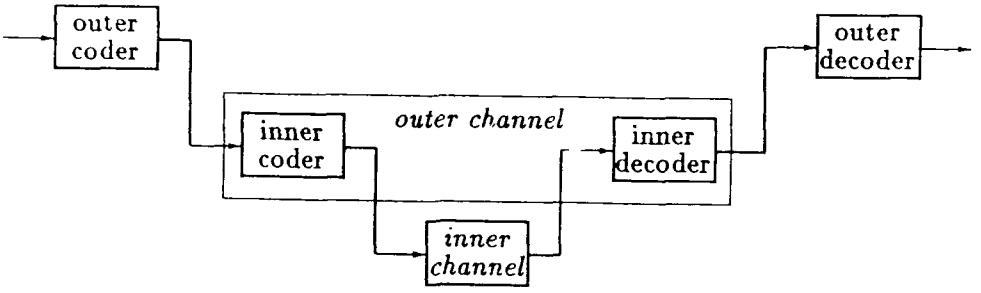


Figure 1: *Channel model with code concatenation*

It must be observed that the concatenation of codes does not give optimal performances as promised by Shannon's bounds: in general concatenated codes are not as powerful as the best single-stage code with the same rate. However multistage decoding presents a reduced complexity. Moreover in some interesting practical cases, concatenation yields performances that are not improved by any known single code.

This paper presents some features that are peculiar to code concatenation. It is structured as follows. Section 2 describes the model of code concatenation assumed in the paper. Section 3 recalls some relevant results on the computation of symbol error probabilities, while Section 4 presents error probability results for the case of code concatenation. Finally, some applications of concatenated codes are described in Section 5.

2 Channel model for code concatenation

Let's consider the transmission chain resulting from the concatenation of two codes. The model of the system is shown in Fig. 1, where two main parts can be identified: the inner and the outer channel. The *inner channel* is a discrete channel resulting either from a chain of modulator + physical channel + demodulator or from another embedded coding section. The inner channel is supposed to be a memoryless binary symmetric channel (BSC), characterized by an error probability p. The code directly facing the inner channel is called *inner code*.

The *outer channel* is the discrete channel resulting by the chain inner coder + inner channel + inner decoder, considered as a unit.

The inner code may be either a convolutional or a block code, sometimes jointly used with modulation, providing either a hard or a soft output. Let $(n_{IN},\ k_{IN},\ d_{IN})$ denote the parameters of the inner code, where d_{IN} is the minimum distance in the case of block codes and the free distance in the case of convolutional codes. Let $R_{IN} = k_{IN}/n_{IN}$ be the inner code rate.

The outer code usually is a block code. Let $(n_{OUT},\ k_{OUT},\ d_{OUT})$ be the parameters of the outer code, with the same conventions as for inner codes. Let $R_{OUT} = k_{OUT}/n_{OUT}$ be the outer code rate.

Code concatenation reduces the net information rate; the overall rate results in

$$R = R_{IN}\ R_{OUT}.$$

The *decoding delay* $\mathcal{D}$ is an important parameter used to evaluate the performance of codes. It is defined as the number of symbols passed between the instant that an information symbol enters the encoder and the instant that the same symbol comes out from the decoder. Concatenation usually increases this figure, but other processing operations, like interleaving or signal propagation, affect delay even more. Here we consider only the net delay introduced by the co-decoding operations. It is direct to verify that

$$\mathcal{D} = \min_{h\ integer} \{h\ n_{IN} \mid \quad h\ n_{IN} \geq n_{OUT}\}.$$

3 Error Probability I – Basic results

In this section we recall definitions as well as results concerning the computation of the symbol error probability. Let's consider $[n, M, d]$ block codes, n being the dimension, $M = 2^k$ the number of code vectors, and d the minimum distance.

For every binary block code, linear or nonlinear, the bit error rate P_{symb} is defined as [9]

$$P_{\text{symb}} = \frac{1}{kM} \sum_{i=1}^{k} \sum_{j=1}^{M} \text{Prob}\{\hat{x}_i \neq x_{j_i} \mid \boldsymbol{x}_j \ \text{ was sent}\} \tag{1}$$

where the code vectors $\boldsymbol{x}_j = (x_{j_1}, x_{j_2}, \ldots, x_{j_k})$ are equally likely, $\hat{\boldsymbol{x}} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_k)$ is the decoded vector, and k is the number of information bits per codeword. In the case of linear codes on a BSC with error probability p, Equation (1) can be written in the form

$$P_{\text{symb}} = \frac{1}{k} \sum_{e} f(e)\ p^{\text{wt}(e)}(1-p)^{n-\text{wt}(e)} \tag{2}$$

where $f(e)$ is the number of incorrect information bits after decoding with the assumption that the all zero code vector was transmitted, and the summation is extended over all the 2^n binary n-tuples. wt(o) is the Hamming weight function: wt(x) is the number of nonzero bits in x. For computational purposes equation (2) can be rewritten in the form

$$P_{\text{symb}} = \sum_{i=0}^{n} B_i \; p^i (1-p)^{n-i} = \sum_{i=0}^{n} E_i \; p^i \tag{3}$$

where

$$B_i = \frac{1}{k} \sum_{\text{wt}(e)=i} f(e) \tag{4}$$

and $\sum_{\text{wt}(e)=i}$ sums $f(e)$ for all error patterns e of Hamming weight i.

Now let $B(X,Y)$ denote the generating polynomial of the B_i's, i.e.

$$B(X,Y) = \sum_{i=0}^{n} B_i \; X^i Y^{n-i};$$

thus we can write

$$P_{\text{symb}} = B(p, 1-p). \tag{5}$$

Moreover let $W(X,Y)$ denote the weight enumerator polynomial, i.e.

$$W(X,Y) = \sum_{i=0}^{n} A_i \; X^i Y^{n-i}$$

where A_i is the number of code vectors whose Hamming weight is i. It can be shown that $B(X,Y)$ can be mechanically derived from $W(X,Y)$, by means of a linear operator, which admits an explicit representation as antisymmetric homogeneous differential operator in the algebra $\mathrm{Z}[[X,Y]]$, see [14].

The closed expression of symbol error rates for many interesting block codes are now known and reported in the literature. Unfortunately this is not true for convolutional codes, although the bit error rate (BER) asymptotic expressions are known for many interesting codes of both kinds.

BER curves. The curves of the bit error rate (BER) versus the error probability of the BSC p show a threshold phenomenon for most commonly used codes: there is a fast transition between the region where the code

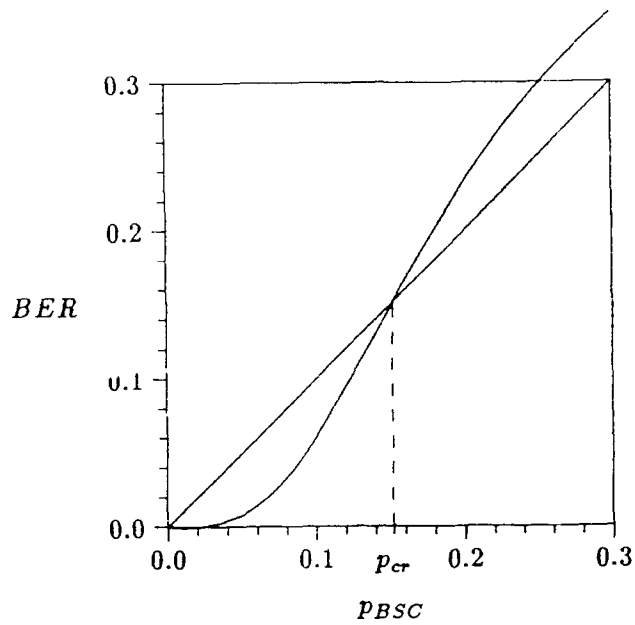


Figure 2: *BER versus raw bit error probability for (23,12,7) Golay code*

reduces the channel noise and the region where the code is useless. Fig. 2 depicts such a phenomenon in the case of the Golay (23,12,7) code.

It must be noted that the symbol error rate on the BSC, even if derived under the assumption of a stationary behavior of the BSC characteristics, gives useful indications on the behavior in a time varying environment: if the dynamic in the time varying environment is limited within a given range, the performance is described by the image of this range, see Fig. 3.

Asymptotic expressions for BER. In the case of binary block codes the asymptotic expressions for the BER take the form

$$P_{\text{symb}} \asymp B\, p^{\lfloor \frac{d-1}{2} \rfloor + 1} \tag{6}$$

where d is the minimum distance of the code and B is a suitable constant. Table 1 shows the asymptotic BER expressions for some interesting block codes.

In the case of convolutional codes, it has been shown [10] that the bit error probability can be written in the form

$$P_{\text{symb}} = C\,[p(1-p)]^{d_f/2} + O\left(p^{\frac{d_f}{2}+\frac{1}{2}}\right) \tag{7}$$

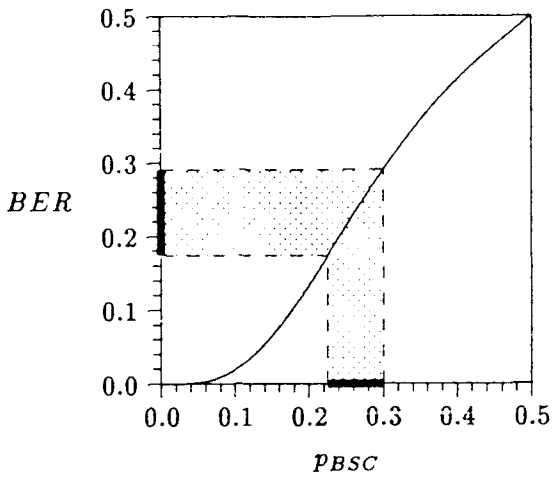


Figure 3: *Bit error performances for time varying channels. The case of BCH (15,5,7) code.*

code	(n, k, d)	BER_{asymp}	p_{cr} on BSC
1	(7,4,3)	$9\ p^2$	0.2115
2	(15,7,5)	$115\ p^3$	0.2444
3	(15,5,7)	$469\ p^4$	0.3233
4	(23,12,7)	$2695\ p^4$	0.1522
5	(24,12,8)	$1771\ p^4$	0.1905

Table 1: *Values of asymptotic BER and p_{cr} for some binary block codes*

L	d_f	BER	p	C
3	5	10^{-6}	$4.0\ 10^{-3}$	0.988
4	6	10^{-6}	$7.0\ 10^{-3}$	2.915
8	10	10^{-6}	$1.5\ 10^{-2}$	1317.

Table 2: *Estimations of the constant C by simulation. Legenda:*
L = constraint length of the rate 1/2 convolutional code
d_f = free distance
p = BSC raw error probability

$$P_{\text{symb}} \asymp C\ p^{d_f/2}. \tag{8}$$

where d_f is the free distance of the particular code and C is a constant that is normally difficult to evaluate. Values of C estimated by simulation for some codes are reported in Table 2.

Critical error probabilities. The critical error probability, another interesting feature of binary codes, is defined as follows.

Definition 1 - *The critical error probability p_{cr} for a binary code is defined as the minimum error probability of a binary symmetric channel (BSC) for which the bit error probability after complete decoding is not greater than the raw error probability of the channel.*

It is immediately apparent that it is not convenient to use the code whenever the error probability on the BSC is greater than the critical error probability: in such a case, in fact, the use of the code leads to worse error performances than no coding at all. Table 1 shows the critical error probabilities for some interesting linear codes.

It might be useful, in order to select among alternative codes over a BSC, to define the *relative critical error probability* as follows.

Definition 2 - *The relative critical error probability p_{rcr} for a pair of binary codes is defined as the minimum error probability of a binary symmetric channel at which the bit error probabilities after complete decoding for the two codes are equal.*

Note that the critical error probability may be also considered for concatenated codes: in fact one of the advantages deriving from "good" concatenations is the increase of the resulting critical error probability, maintaining at the same time good code performances for p below such limit.

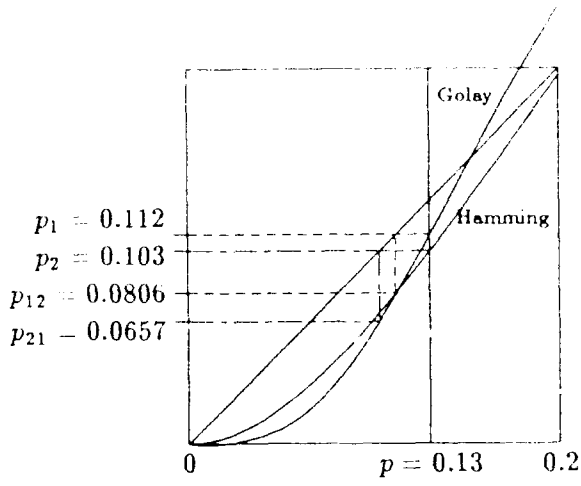


Figure 4: *Asymmetry of code concatenation*

4 Error Probability II – Concatenated codes

Recalling the code concatenation model given in Section 2, the inner code may be viewed as a mechanism that transforms the error probability p_i of the inner channel to $p_o = f_i(p_i)$, the bit error probability of the outer channel. The outer code performs a similar operation by transforming p_o to the error probability of the concatenated system $p_s = f_o(p_o)$. The resulting transformation is

$$p_s = f_o(f_i(p_i)).$$

Due to the non-linearity of the $f_i(\circ)$ and $f_o(\circ)$ functions, this is in general different from

$$p_s' = f_i(f_o(p_i)).$$

Therefore the optimal concatenation of two codes, under the only constraint of achieving the minimum symbol error probability at a given rate, depends in general on the order in which the two codes operate.

A sketch of how the concatenation of two codes depends on the order is shown in Fig. 4. In the figure, for a given value of the inner channel error probability — $p_i = p = 0.13$ — and two particular codes with similar rates — code 1 is the Golay code (23,12,7) and code 2 is the Hamming (7,4,3) code —, it is shown that two different values of the overall error probability p_s can be obtained. In particular, when code 1 is chosen as the inner code,

$p_o = p_1 = 0.112$ and $p_s = p_{12} = 0.0806$, while, when code 2 is the inner code, $p_o = p_2 = 0.103$ and $p_s = p_{21} = 0.0657$.

It should be remembered that often the need for the high error correcting capabilities of concatenated codes arises at high channel error probability ($1 \div 5\ 10^{-2}$), where the asymptotic expressions do not hold.

In many applications, codes are used in the presence of sufficiently small channel error probabilities, so that the polynomial expressions introduced in the previous section can be substituted with their asymptotic form for p tending to zero: this means that we can consider only the term of the polynomials where p has the smallest exponent.

Under the assumption that p is small enough for the asymptotic expressions to hold, we may develop some considerations that allow the comparison, in terms of the symbol error probability, of the two possible concatenation orders for a couple of codes.

Let $p_1 = A_1 p^{n_1}$ and $p_2 = A_2 p^{n_2}$ be respectively the bit asymptotic error probabilities of codes C_1 and C_2. If the concatenation shows C_1 as the outer and C_2 as the inner code, we obtain the asymptotic BER

$$p_{12} = A_1\ A_2^{n_1}\ \ p^{n_1\ n_2}$$

while C_2 outer and C_1 inner gives

$$p_{21} = A_2\ A_1^{n_2}\ \ p^{n_1\ n_2}.$$

In the particular case $n_1 = n_2 = n$, it is straightforward to see that the best asymptotic performances are obtained when the inner code has a lower asymptotic error probability, i.e. when it has a lower value for the coefficient A.

Table 3 shows the asymptotic expressions of the bit error rate for the two possible concatenation orders for some couples of codes (taken from Table 1).

5 Applications

In this section we consider some widely used code concatenations and we evaluate their performances.

Order of concatenation			Order of concatenation
1-2	$119025\ p^6$	$83835\ p^6$	2-1
1-3	$1979649\ p^8$	$3077109\ p^8$	3-1
2-5	$6.39\ 10^{11}\ p^{12}$	$3.10\ 10^{11}\ p^{12}$	5-2
3-5	$4.38\ 10^{15}\ p^{16}$	$8.57\ 10^{13}\ p^{16}$	3-5

Table 3: *Comparison of code concatenation orders*

5.1 Rate 1/2 convolutional and Reed-Solomon code chain

One of the first proposed code's chain, see [1], consists of a convolutional code of rate 1/2 and constraint length 7 with Viterbi decoding as the inner code and a Reed-Solomon (RS) code over GF(2^8) as the outer code.

The maximum free distance for noncatastrophic convolutional codes with rate 1/2 and constraint length 7 is 10, see [21, page 251]. The asymptotic BER for this convolutional code is, from (8), $C\,p^5$. The constant C has been estimated by simulation, using the TOPSIM III simulator [20].

5.2 Uneven error protection - the LPC case

Several applications require uneven protection of the information to be transmitted. This is the case, for example, of packetized data transmissions, where the protocol information carried by packets requires better protection than the information part. A typical situation is the transmission of voice digitized according to the Linear Predictive Coding (LPC) [22] approach.

The LPC-10 is a US Government standard that allows the transmission of digitized voice at 2.4 Kbit/s. The speech signal is segmented in contiguous talkspurts of 22.5 ms, called frames. Each frame is coded into a 54 bit packet. Frames can be of two kinds: voiced and unvoiced. Voiced frames are reconstructed at the receiver by filtering a basic waveform with a filter whose coefficients are estimated by the transmitter by means of the LPC covariance analysis algorithm. These coefficient are transmitted inside the 54 bit packets. Unvoiced frames carry little information content, so that they are rebuilt as filtered noise, with a lower order filter whose parameters require less bits in the 54 bits packets; the remaining bits are used for error protection of the bit stream. One bit is used to discriminate between voiced and unvoiced packets.

It should be clear that LPC packets contain various kinds of information

(filter coefficients, other LPC parameters, voiced/unvoiced flag, error protection, etc.) whose need for error protection varies: it is very important, for example, not to spoil the content of the voiced/unvoiced flag, while an error in a low order bit of a filter coefficient is much less significant. In this case it is desirable to have a better error protection on some bits of the packets: this target is easily pursued with code concatenation. The inner code could be applied only to those parts requiring more protection, while the outer code could protect the whole bit stream.

5.3 The Compact Disc audio system

In the Compact Disc audio system error protection is achieved by the use of two chained Reed-Solomon (RS) codes [2]. A (32,28,5) RS code is used as the inner code and a (28,24,5) RS code is the outer code, both are over GF(2^8); detected errors in the inner codes are erasures for the outer code. The concatenation order has been chosen out of several constraints, but it can be shown to be not optimal.

The two RS codes show the same minimum distance, hence the same exponent for p in (6). The coefficient B in the same equation is greater for the (32,28) than for the (28,24) code, since the former offers similar error correcting capabilities on a larger number of information symbols. But in Section 4 it has been shown that, under these conditions, the inner code should be the one with the smaller coefficient B: this implies that the reverse concatenation order would lead to better symbol error performances.

5.4 Comparison of code concatenation with single stage codes

As final example let us compare an instance of concatenated codes with several single codes with comparable rate. It seems that concatenated codes outperforms any known single code even at the relatively small rate of 0.38. The codes are listed below and the results are summarized in Table 4.

1. Concatenation of inner Hamming $(15, 11, 3)$ code and outer Golay $(23, 12, 7)$ code
2. Concatenation of outer Hamming $(15, 11, 3)$ code and inner Golay $(23, 12, 7)$ code
3. Single BCH $(33, 13, 10)$ code

	Codes inner outer	Rate	Dec. delay	Asymptotic BER	p_{cr}
1	(15,11,3) (23,12,7)	0.383	30	$839,548,980\ p^8$	0.137
2	(23,12,7) (15,11,3)	0.383	23	$171,588,966\ p^8$	0.124
3	(33,13,10)	0.394	33	$35,960\ p^5$	0.290
4	(39,15,10)	0.385	39	$73,815\ p^5$	0.311
5	(55,21,15)	0.382	55	$178,181,640\ p^8$	0.279

Table 4: *Comparisons among error-control schemes*

4. Single BCH $(39, 15, 10)$ code

5. Single BCH $(55, 21, 15)$ code

The asymptotic BER was derived according to the approach described in [14].

The decoding complexity may be hard to define, due to the fact that an efficient decoding algorithm is not always available. Referring to the decoding schemes available today, the codes used in the concatenations 1. and 2. can be decoded with the very efficient error trapping procedure devised by Kasami [16], while for the other single stage codes the known, [24] and [23], complete decoding procedure, is direct computation of the minimum distance. It turns out that the latter codes are incomparably more difficult to decode.

6 Conclusions

The use of error-control codes calls for a compromise between efficiency and complexity. The original scheme proposed by Forney [1] of concatenating two or more codes and modulation provides the proper answer to the problem: it is now well accepted that concatenation van be advantageously applied to manage many interesting situations, and sometimes it can be the only solution with affordable complexity.

The decreasing cost of digital circuits allows to foresee that cost-effective applications of codes will be even more widespread in the near future. Code concatenation yields cheaper implementations together with high performances. Many authors support the opinion that code concatenation is not only a trick, due to our limited knowledge of codes' structure, to achieve good performances; rather it is an effective way to obtain high performance

with limited complexity. This opinion is upheld by the proof, see [25], that the general decoding problem, also for linear codes, is NP-complete.

References

[1] **D. Forney,** *Concatenated Codes,* Research Monograph no. 37, MIT Press, Cambridge, MA, 1966

[2] **H. Hoeve, J. Timmermans and L. B. Vries,** Error Correction and Concealment in the Compact Disc System, *Philips tech. Rev.,* vol. 40, no. 6, 1982, pp. 166-172

[3] **A.M. Michelson and A.H. Levesque,** *Error-Control Techniques for Digital Communication,* Wiley, New York, 1985.

[4] **Kasami and Shu Lin,** *A Cascaded Coding Scheme for error Control and its Performance Analysis,* IBM Workshop on Error Control Coding, San Jose, Sept. 1987.

[5] **A.M. Michelson, et al.,** *Performance Results for a Concatenated Code Design on the DPSK Channel,* MILCOM 1985, Boston, Oct.20-23, vol 1, pp.236-240.

[6] **Boyd, et al.,** *A Concatenated Coding Approach for High Data rate Applications,* NTC 1977, Los Angeles, vol 3, pp.36: 2-1/2-7.

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

[8] **E. R. Berlekamp,** *Algebraic Coding Theory,* McGraw-Hill Book Company, New York, 1968

[9] **F. J. MacWilliams and N. J. A. Sloane,** *The Theory of Error-Correcting Codes,* Elsevier, New York, 1976

[10] **R. J. McEliece,** *Finite Fields for Computer Scientists and Engineers,* Kluwer Academic Press, Boston, 1987

[11] **R. J. McEliece,** *The Theory of Information and Coding,* Addison Wesley, 1977

[12] **M. Elia,** Algebraic decoding of the (23,12,7) Golay Code, *IEEE Transactions on Information Theory,* vol. IT-33, Jan. 1987, pp. 150-151

[13] **E. R. Berlekamp,** Bit-serial Reed-Solomon encoders, *IEEE Transactions on Information Theory,* vol. IT-28, Nov. 1982, pp. 869-874

[14] **M. Elia,** A note on the computation of bit error rate for binary block codes, *Journal of Linear Algebra and its Applic.,* vol. 98, Jan. 1988, pp. 199-210

[15] **L. A. Dunning,** Encoding and Decoding for the Minimization of Message Symbol Error Rates in Linear Block Codes, *IEEE Transactions on Information Theory,* vol. IT-33, no. 1, Jan. 1987, pp. 91-104

[16] **Shu Lin and D. J. Costello,** *Error Control Coding: Fundamentals and Applications,* Prentice-Hall, Englewood Cliffs, New Jersey, 1983

[17] **W. W. Peterson and E. J. Weldon,** *Error-Correcting Codes,* MIT Press, Cambridge, MA, 1981

[18] **M. Elia,** Symbol error rate of binary block codes, *Trans. Ninth Prague Conference on Inform. Th., Statist. Dec. Functions, Random Processes,* Prague, pp. 223-227, June 1982

[19] **M. Elia and G. Prati,** On the Complete Decoding of Binary Linear Codes, *IEEE Trans. on Inform. Th.,* vol. IT-31, no. 4, July 1985, pp. 518-520

[20] **M. Ajmone-Marsan et al.,** Digital Simulation of Communication Systems with TOPSIM, *IEEE Journal Select. Areas in Communications,* vol. SAC-2, no. 1, Jan. 1984, pp. 42-50

[21] **A. J. Viterbi and J. K. Omura,** *Principles of Digital Communication and Coding,* Mc-Graw Hill Book Company, 1979

[22] **P.E. Papamichalis,** *Practical Approaches to Speech coding,* Prentice-Hall, Englewood Cliffs, New Jersey, 1987

[23] **J.H. Van Lint, R.M. Wilson,** On the Minimum Distance of Cyclic Codes, *IEEE Trans. on Inform. Th.,* vol. IT-32, no. 1, July 1986, pp. 518-520

[24] **P. Bours, J.C.M. Janssen, M. van Asperdt, H.C.A. van Tilborg,** Algebraic Decoding beyond e_{BCH} of some binary cyclic codes, when $e > e_{BCH}$ *International Colloquium on Coding Theory, Osaka, 1988*

[25] **E.R. Berlekamp, R.J.McEliece, H.C.A. van Tilborg** On the Inherent Intractability of Certain Coding Problems *IEEE Trans. on Inform. Th.,* vol. IT-24, no. 3, May 1978, pp. 384-386