Generalized t-Designs and Weighted Majority Decoding*

R. SAFAVI-NAINI AND IAN F. BLAKE

Department of Electrical Engineering, University of Waterloo, Waterloo, Ontario, N2L 3G1 Canada

Several authors have investigated the presence of combinatorial structures, most notably t-designs, among the supports of code words of a given weight. Their application to majority decoding has also received attention. In this paper generalized t-designs, in which differing block sizes and block multiplicities are allowed, are considered. A simple method for determining such designs from the supports of the code words of binary linear codes is established. Using this method a constructive proof of the Rudolph-Robbins theorem, that every binary linear code may be one-step weighted majority decoded, is given.

1. INTRODUCTION

Connections between coding theory and combinatorial designs have been studied since the early work of Paige (1956) and Bose (1961), among others. Such investigations have taken two approaches. In one approach the structure of good codes is examined to determine the presence of combinatorial designs. The alternative approach is to use the linear span of the incidence matrix of a combinatorial configuration, such as a finite geometry or block design, as the dual code. In both cases the presence of such structures led to majority logic-type decoding algorithms.

One of the most useful and important methods for determining the existence of t-designs among the supports of code words of a given weight is the Assmus-Mattson theorem (1969a). This theorem led to the discovery of new 5-designs. Subsequently many other workers, including Delsarte (1973) and MacWilliams and Sloane (1977), contributed important results to the problem. Goethals (1970) and Assmus and Mattson (1969b) independently discovered a one-step majority decoding algorithm for the extended binary (24, 12) Golary code, using the fact that its code words of a given weight support a 5-design. A similar algorithm was later found for the (48, 24) binary quadratic residue code by Assmus and Mattson (1970).

Delsarte (1973) introduced the notion of a q-ary t-design in relation to q-ary codes. In the case \( q = 2 \) a q-ary t-design is an ordinary t-design. These designs

* This work was sponsored by NRC Grant A-7382.
and their relation to code structure were further investigated by Goethals and van Tilborg (1975) and Assmus et al. (1976) and this last work used the q-ary t-designs to decode certain codes.

Ng (1970) noticed that in certain cases one-step majority decoding could be effected by using nonorthogonal parity checks and weighting the zero parity check. Rudolph and Robbins (1972) extended this idea and proved the remarkable fact that any linear binary code can be one-step weighted majority decoded. The proof, however, was nonconstructive and, for a particular code, it remained a problem to determine a set of parity check equations with associated weightings to achieve such a one-step decoding.

The purpose of the present paper is twofold. First, the notion of a generalized t-design, allowing different block sizes and block multiplicities is introduced. Second, these designs will be used to give a constructive proof of the Rudolph-Robbins theorem. It should be noted that this use of the term generalized t-design differs from that in Assmus et al. (1976). In coding terms the block sizes will be the weights of the code words used and the multiplicities assigned to the blocks will be the weights of the parity checks. In the case where all the multiplicities are positive integers, they can be viewed as block repetitions. This notion of generalized t-designs is perhaps too general to be of significant interest in itself. When viewed in the light of other structures such as orthogonal arrays and linear codes, interesting results can be obtained.

For the remainder of this section, the necessary coding theory prerequisites will be briefly stated. The next section will review the notions of ordinary, q-ary, and generalized t-designs in order to place these latter designs and their properties in perspective. Methods for obtaining generalized t-designs from linear codes are discussed in Section 3 where several theorems on their construction and characterization are given. Their use in weighted one-step majority decoding is demonstrated in Section 4 where the constructive proof of the Rudolph-Robbins theorem is given.

Many of techniques used in this paper are due to Delsarte (1973) and for completeness we briefly introduce some of the notation. Let \( F_q \) be the finite field of order \( q \) and \( F_q^n \) the vector space of \( n \)-tuples over \( F_q \). For \( x, y \in F_q^n \), the weight of \( x \), \( w(x) \), is the number of its coordinate positions nonzero and the distance between \( x \) and \( y \), \( d(x, y) \), is the number of positions in which \( x \) and \( y \) differ, \( w(x - y) \). An arbitrary subset \( C \) of \( F_q^n \) is called a (nonlinear) code. The minimum distance \( d \) of \( C \) is defined by

\[
d = \min_{x, y \in C} d(x, y).
\]

If \( |C| = M \) then the distance distribution of \( C \) is defined to be the \((n + 1)\)-tuple \( B = (B_0, B_1, \ldots, B_n) \), where

\[
B_i = (1/M) |\{(x, y) \mid x, y \in C, d(x, y) = i\}|.
\]
Notice that $B_0 = 1$ and that $d$ is the smallest value of $i$, $i \geq 1$, for which $B_i \neq 0$. We define the parameter $s$ of $C$ to be the number of nonzero distances of $C$:

$$s = |\{i | B_i \neq 0, i = 1, 2, \ldots, n\}|.$$

For fixed positive integers $n$ and $\lambda$, the Krawtchouk polynomial of degree $k$ is defined by

$$P_k(x) = \sum_{j=0}^{k} (-1)^{\lambda} x^{k-j} \frac{x}{j!} \left(\frac{n-x}{k-j}\right),$$

where $C_k = \frac{(x+1) \cdots (x+j)}{j!}$. When it is desired to explicitly show the dependence on $n$, we will use the notation of MacWilliams and Sloane (1977) and write the polynomial $P_k(x; n)$. These polynomials have many interesting properties and we note in particular that for $\lambda = 1$

$$\binom{n}{i} P_i(i) = \binom{n}{j} P_j(j).$$

To an $(n+1)$-tuple of rational numbers $A = (A_0, A_1, \ldots, A_n)$ define

$$A_k = \sum_{i=0}^{n} A_i P_k(i)$$

and the $(n+1)$-tuple $S = (S_0, S_1, \ldots, S_n)$ is called the MacWilliams transform of $A$. The MacWilliams transform of the distance distribution of a code $C$, $B = (B_0, B_1, \ldots, B_n)$ is such that $S_i \geq 0$ (Delsarte, 1972). The parameters $s'$ and $d'$ are defined from the MacWilliams transform by

$$s' = |\{i | S_i \neq 0, i = 1, 2, \ldots, n\}|$$

and

$$d' = \min_{S_i \neq 0} i,$$

respectively. Thus each code $C$ has four fundamental parameters $d$, $s$, $d'$, and $s'$.

If $C \subseteq F_q^n$ is a $k$-dimensional subspace then $C$ is referred to as a linear $(n, k)$ code. In this case the weight enumerator is

$$A(x, y) = \sum_{i=0}^{n} A_i x^i y^{n-i},$$

where $A_i$ is the number of code words of weight $i$ and $A_i = B_i$. The dual of the linear $(n, k)$ code $C$ is $C'$ where

$$C' = \left\{ y \in F_q^n | (x, y) = \sum_{i=1}^{n} x_i y_i = 0 \right\}.$$
and is a linear \((n, n - k)\) code. Denote the weight enumerator of \(C'\) by

\[
A'(x, y) = \sum_{i=0}^{n} A'_i x^i y^{n-i}.
\]

It was shown by Delsarte (1972) that \(A'_k = |C| A'_k\), i.e., the weight enumerator of \(C'\) is a scalar multiple of the MacWilliams transform of the weight enumerator of \(C\). Consequently the parameter \(d'\) of the linear code \(C\) is the minimum distance of \(C'\) and \(s'\) is the number of nonzero weights of \(C'\).

Following MacWilliams and Sloane (1977) the nonzero weights of \(C\) will be denoted by \(\tau_1, \tau_2, \ldots, \tau_s\) and those of \(C'\) by \(\sigma_1, \sigma_2, \ldots, \sigma_{s'}\). The remainder of the paper will deal only with binary linear codes although some of the results will also be valid for nonlinear codes.

As a final point of notation for \(x = (x_1, x_2, \ldots, x_n) \in F_2^n\) let \(\text{supp}(x) = \{i \mid x_i \neq 0\}\). For \(x, y \in F_2^n\), \(x\) is said to cover \(y\) if \(x_i = y_i\) for all \(i \in \text{supp}(y)\).

A particularly useful equation, which we refer to as Delsarte's relation, states that for a linear code \(C\)

\[
\delta(x) = \sum_{y \in C} \alpha(w(y)) \langle x, y \rangle = |C| \sum_{i=0}^{n} \alpha_i B'(x, i),
\]

where \(\alpha_i\) is the \(i\)th coefficient in the expansion of \(\alpha(x)\), a polynomial of degree at most \(n\), in the basis of Krawtchouk polynomials; \(B'(x, i)\) is the number of words of \(C'\) at distance \(i\) from \(x\); and, in the binary case of interest here, \(\langle x, y \rangle = (-1)^{(i, y)}\), \((x, y)\) the usual inner product on \(F_2^n\).

2. Generalized t-Designs

An ordinary \((t-(n, k, \lambda_t))\)-design, or \(t\)-design, on the set \(X = \{1, 2, \ldots, n\}\) is a collection of distinct \(k\)-sets of \(X\) called blocks, such that every \(t\)-set of \(X\) is contained in precisely \(\lambda_t\) of the \(k\)-sets of \(D\). If \(\lambda_t = 1\) the design is called a Steiner system. A \(t\)-design is also an \(i\)-design, \(0 \leq i \leq t\). The complement of a \(t\)-design, obtained by replacing each \(k\)-set with its complement in \(X\) to give blocks of size \((n - k)\) is also a \(t\)-design. If every \(k\)-set of \(X\) is a block the design is called trivial.

One of the most important theorems for obtaining \(t\)-designs from codes is the Assmus–Mattson theorem, the binary version of which can be stated as follows:

**Theorem 2.1** (Assmus and Mattson, 1969). Suppose that the number of nonzero weights of the binary linear code \(C'\) which are less than or equal to \(n - t\) is itself less than or equal to \(d - t\). Then each weight of \(C\) supports a \(t\)-design and each weight less than or equal to \((n - t)\) of \(C'\) supports a \(t\)-design.
Another theorem, clearly related to the above but different in its statement, was obtained by MacWilliams and Sloane (1977).

**Theorem 2.2** (MacWilliams and Sloane, 1977). *Let $C$ be a binary linear code with parameters $d$, $s$, $d'$, and $s'$*. Let $\bar{s}$ be $s$ if $A_n = 0$ and $s - 1$ otherwise and let $\bar{s}'$ be $s'$ if $A'_n = 0$ and $s' - 1$ otherwise. *If either $\bar{s} < d'$ or $\bar{s}' < d$ then the code words of weight $w$ in $C$ (and $C'$) form a $t$-design where $t = \max(d' - \bar{s}, d - \bar{s}')$, provided that $t < d$.*

It was observed recently (Safavi-Naini and Blake, 1978) that the proviso $t < d$ in Theorem 2.2 is unnecessary since this is always the case. Furthermore, it was shown that Theorems 2.1 and 2.2 are equivalent in the sense that the value of $t$ obtained in each theorem is the same.

One can also obtain $t$-designs from codes over $\mathbb{F}_q$ by considering the supports of code words of a given weight as the design blocks. The $q$-ary version of the Assmus–Mattson theorem, which will not be given here, covers this situation.

Perhaps a more natural combinatorial structure to investigate in codes over $\mathbb{F}_q$, $q > 2$, is the $q$-ary $t$-design, first defined by Delsarte (1973). A set $D$ of elements of $\mathbb{F}_q^n$ forms a $q$-ary $t$-design with parameters $\lambda_t$, $t$, $k$, and $n$, if every element has weight $k$ and if the number of elements of $D$ covering $x \in \mathbb{F}_q^n$, $w(x) = t$, is $\lambda_t$, a constant independent of $x$. It can be shown that a $q$-ary $t$-design is also a $q$-ary $i$-design, $0 \leq i \leq t$. When $q = 2$, the notion of a $q$-ary $t$-design reduces to that of an ordinary $t$-design. The following theorem, contained in Safavi-Naini and Blake (1978) is largely due to Delsarte (1973), Goethals and van Tilborg (1975), and Assmus et al. (1976).

**Theorem 2.3.** *Let $C$ be a $q$-ary code containing the all-zero code word with the parameters $d$, $s$, $d'$, and $s'$ such that $s' \geq d/2$. Then for each weight $k$ the code-words of $C$ form a $q$-ary $t$-design where $t = \max(d - s', d' - s)$.*

The theorem appears as an analog of Theorem 2.2, recognizing that code words of weight $n$ in the $q$-ary case may not be scalar multiples of each other and hence such vectors cannot be deleted as in the binary case, i.e., $s$ cannot in general be replaced by $\bar{s}$ in the $q$-ary case.

We now introduce generalized $t$-designs, written $G$ $t$-designs.

**Definition.** Let $D$ be a set of distinct subsets of $X = \{1, 2, \ldots, n\}$, not necessarily of the same size. To the block $d \in D$ we assign the real number $m_d$, the multiplicity of the block $d$. The set $D$, together with the assigned multiplicities, is called a $G$ $t$-design if for any $t$-set $y$ of $X$,

$$
\sum_{d \in D : y \subseteq d} m_d = \lambda_t,
$$

where $\lambda_t$ is a real number, independent of $y$. 
Notice that we do not insist that the multiplicities be positive or integer, although cases where they are all positive and integer will have a physical interpretation. The $G_t$-design can be identified with its incidence matrix $D$ where the row of $D$ corresponding to $d \in D$ will have the multiplicity $m_d$ "associated" with it. For convenience we refer to $d \in D$ as the block and $d \in D$ as its incidence vector, a binary $n$-tuple.

Denote by $S$ the set of block sizes of $D$ i.e., $S = \{ \tau \mid \tau \leq \tau_i, d \in D \} = \{ \tau_1, \tau_2, \ldots, \tau_s \}$, $\tau_i < \tau_{i+1}$, $i = 1, 2, \ldots, s - 1$. If $m_d = 1, d \in D$ and $|S| = s = 1$ the $G_t$-design is an ordinary $t$-design. Many of the properties of ordinary $t$-designs do not carry over to $G_t$-designs. We note in particular that a $G_t$-design is not necessarily a $G_i$-design, $0 \leq i \leq t$. The complement of a $G_t$-design, $D^c = \{X\setminus d, d \in D\}$ where the block $X\setminus d$ is assigned the multiplicity $m_d$, is not necessarily a $G_t$-design. These comments are illustrated in the following example.

**Example 2.1.** The set of incidence vectors, $D$, 

\[
\begin{array}{cccccc}
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 \\
\end{array}
\]

with each row assigned a multiplicity of 1, describes a $G_3$-design which is not a $G_2$-design but is a $G_1$-design. The complementary design is neither a $G_3$-design nor a $G_2$-design but is a $G_1$-design.

The $G_t$-design $D$ is called homogeneous if it is a $G_i$-design for each $i = 0, 1, \ldots, t$ and such a design will be referred to as a $HG_t$-design. Some of the properties of these designs are similar to those of ordinary $t$-designs and to verify this we require information on two parameters of the design. For the $HG_t$-design let $\lambda_i, i = 0, 1, \ldots, t$, denote the number of blocks of $D$, including multiplicities, which contain a given $i$-set and by a simple counting argument

\[
\lambda_i = \left[ \sum_{d \in D} \binom{d_i}{i} \right] \binom{n}{i}, \quad i = 0, 1, \ldots, t.
\]
Let $\lambda_{i,j}$ denote the number of blocks, again including multiplicities, that contain a given $i$-set, say $v_i$, but contain no elements of a given $j$-set, $v_j$, $v_i \cap v_j = \emptyset$, $i + j \leq t$. By an inclusion–exclusion argument

$$\lambda_{i,j} = \sum_{k=0}^{j} (-1)^k \binom{j}{k} \lambda_{i+j-k}, \quad i + j \leq t,$$

and this quantity depends only on $i$ and $j$. Denote by $D^c$ the complement of the $HG_t$-design $D$, where the complement of $d \in D$ retains the multiplicity $m_d$. Since $\lambda_{i,j}^c = \lambda_{i,t}$ and $\lambda_{i}^c = \lambda_{t,0}$, where the superscript $c$ refers to the complemented design $D^c$, it follows that $D^c$ is also a $HG_t$-design.

**Example 2.2.** The following incidence vectors each with multiplicity 1 form a $HG_3$-design:

\[
\begin{array}{cccccccc}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 \\
\end{array}
\]

with parameters $\lambda_0 = 14$, $\lambda_1 = 7$, $\lambda_2 = 3$, and $\lambda_3 = 1$.

The incidence vectors of this example are actually the code words of weights 3 and 4 in a realization of the Hamming $(7, 4)$ code. Such examples can be readily constructed from codes by considering orthogonal arrays. A binary $M \times n$ incidence matrix $0$ with the property that any $T$ of its columns contain every ordered binary $T$-tuple exactly $\mu$ times is called a binary orthogonal array of strength $T$. A code $C$, either linear or nonlinear, is an orthogonal array of strength $d' - 1$. Clearly if we assign a multiplicity of one to each row of an orthogonal array, of strength $T$, the result is a $HG_T$-design. Since the dual of the Hamming $(7, 4)$ code has minimum distance 4 and since the all-ones
incidence vector can be deleted without affecting the \( t \)-design property, it follows that the weight 3 and 4 code words form a \( HG \ 3 \)-design as claimed in the example. The relationship between orthogonal arrays and \( G \ t \)-designs is explored further in Safavi-Naini and Blake (in press).

3. Generalized \( t \)-Designs from Codes

The relationships between orthogonal arrays, and \( G \ t \)-designs were explored in Safavi-Naini and Blake (in press). We will interpret one of the main results of that work in terms of linear codes. For the linear code \( C \) let \( A'_\sigma \) denote the set of words in the dual code \( C' \) of weight \( \sigma \). Assign to each such word, viewed as the incidence vector of a \( \sigma \)-set of \( X = \{1, 2, \ldots, n\} \), a multiplicity \( m_\sigma \) (the same for each word of weight \( \sigma \)). The collection of words with their associated multiplicities is then written as \( \sum_{\sigma \in S'} m_\sigma A'_\sigma \) where \( S' \) is the set of nonzero weights in \( C' \). In terms of linear codes, Theorem 3.2 of Safavi-Naini and Blake (in press) can be recast as follows:

**Theorem 3.1.** Let \( C \) be a binary linear \((n, k)\) code with distance \( d \) and \( f(x) \) a polynomial of degree \( r \), \( r < d' - 1 \). Then \( \sum_{\sigma \in S'} f(\sigma) A'_\sigma \) is a \( HG \ (d' - 1 - r) \)-design.

A useful characterization of \( HG \ t \)-designs is contained in the following lemma.

**Lemma 3.2.** The linear combination \( \sum_{\sigma \in S} m_\sigma A_\sigma \) of the code \( C \) is a \( HG \ t \)-design iff for any \( x \in \mathbb{F}_2^n \), \( w(x) < t \), \( \sum_{y \in C} m_y \langle x, y \rangle \) is a function of \( w(x) \) only, independent of the particular \( x \).

The proofs of the theorem and lemma are omitted. The parameters of the design of Theorem 3.1, which will be useful in the next section, are easily calculated as

\[
\lambda_i = \left( \sum_{\sigma \in S'} f(\sigma) \binom{\sigma}{i} A'_\sigma \right)/\binom{n}{i}, \quad i = 0, 1, \ldots, d' - 1 - r. \tag{3.1}
\]

Similarly we now find the quantities \( \lambda_{i,j} \).

**Lemma 3.3.** For the \( HG \ (d' - 1 - r) \)-design \( \sum_{\sigma \in S'} f(\sigma) A_\sigma \)

\[
\lambda_{i,j} = \sum_{\sigma \in S'} f(\sigma) A_\sigma \binom{n - i - j}{\sigma - i} / \binom{n}{\sigma}. \tag{3.2}
\]
Proof. As in the case of ordinary t-designs
\[ \lambda_{i,j} = \sum_{k=0}^{j} (-1)^k \binom{j}{k} \lambda_{i+k} \]
for \( i + j \leq t = d' - 1 - r \). Substituting (3.1) into this expression
\[ \sum_{k=0}^{j} (-1)^k \binom{j}{k} \sum_{\sigma \in S'} f(\sigma) A_\sigma \left( \binom{\sigma}{i+k} / \binom{n}{i+k} \right) \]
\[ = \sum_{\sigma \in S'} f(\sigma) A_\sigma \left( \binom{\sigma}{i} / \binom{n}{i} \right) \sum_{k=0}^{j} (-1)^k \binom{j}{k} \binom{\sigma - i}{k} / \binom{n - i}{k} \]
\[ = \sum_{\sigma \in S'} f(\sigma) A_\sigma \left( \binom{n - i - \sigma}{j} / \binom{n}{j} \right), \]
where use has been made of Eq. (7.1) in Gould (1972). This last expression is easily simplified to
\[ \lambda_{i,j} = \sum_{\sigma \in S'} f(\sigma) A_\sigma \left( \binom{n - i - j}{\sigma - i} / \binom{n}{\sigma} \right) \]
as required.

There are several interesting corollaries to Theorem 3.2 which are briefly considered.

Corollary 3.4. Let \( P_i(x; n) = P_i(x) \) be the Krawtchouk polynomial of degree i. Then \( \sum_{\sigma \in S} P_{\sigma}(r) A_\sigma \) is a HG \( (d' - 1 - i) \)-design with parameters \( \lambda_j \) which are zero for \( j < i \). If \( i > (d' - 1)/2 \) then \( \lambda_j = 0, j = 0, 1, \ldots, d' - 1 - i \).

Proof. Notice that for convenience the HG design is constructed on the code \( C \) itself rather than \( C' \) and so \( d \) is replaced by \( d' \). By direct enumeration we have
\[ \binom{n}{j} \lambda_j = \sum_{\sigma \in S} \binom{\tau}{j} P_{\sigma}(\tau) A_\sigma, \quad j = 0, 1, \ldots, d' - 1 - i. \] (3.3)
As a polynomial of degree \( j \), we can write
\[ \binom{x}{j} = \frac{x(x-1) \cdots (x-j+1)}{j!} = \sum_{i=1}^{j} \alpha_{ij} x^i. \]
Also from the recurrence relation for Krawtchouk polynomials (MacWilliams and Sloane, 1977, p. 152),

\[(k + 1) P_{k+1}(x) = (n - 2x) P_k(x) - (n - k + 1) P_{k-1}(x),\]

it follows that

\[xP_l(x) = \sum_{k=0}^{l} \beta_{l,k} P_k(x), \quad l \geq 1,\]

and, by induction

\[x^r P_l(x) = \sum_{k=\max(l-r,0)}^{l-r} \beta_{l-r,k} P_k(x).\]

Using these expressions in (3.3) yields

\[\binom{n}{j} \lambda_j = \sum_{\tau \in S} \left\{ \sum_{k=0}^{j} \alpha_{k,j} \tau^k \right\} P_i(\tau) A_{\tau} = \sum_{k=0}^{j} \alpha_{k,j} \sum_{\tau \in S} \tau^k P_i(\tau) A_{\tau}

\]

\[= \sum_{k=0}^{j} \alpha_{k,j} \sum_{\tau \in S} \left\{ \sum_{m=\max(i-k,0)}^{i+k} \beta_{l,m}^{(k)} P_m(\tau) \right\} A_{\tau}

\]

\[= \sum_{k=0}^{j} \alpha_{k,j} \sum_{m=\max(i-k,0)}^{i+k} \beta_{l,m}^{(k)} \left\{ \sum_{\tau \in S} P_m(\tau) A_{\tau} \right\}.

\]

The expression in the brackets is the MacWilliams transform of the weight distribution of \(C\) and since \(m \leq i + k \leq i - j \leq d' - 1\) we have

\[\sum_{\tau \in S} P_m(\tau) A_{\tau} = |C| A'_m = |C| \delta_{m0}.

\]

Consequently we have

\[\binom{n}{j} \lambda_j = |C| \sum_{k=0}^{j} \alpha_{k,j} \sum_{m=\max(i-k,0)}^{i+k} \beta_{l,m}^{(k)} \delta_{m0}\]

and if \(i > j\) we must have \(\lambda_j = 0\) since in this case \(m\) cannot be zero. To conclude the corollary note that if \(i > d' - 1 - i\) or \(i > (d' - 1)/2\) then \(\lambda_j = 0, j \leq i,\) i.e., all the parameters of the design are zero in this case.

**Example.** As an example of this unusual corollary consider the extended binary \((24, 12)\) Golay code. It has minimum distance 8 and is self-dual. Each nonzero weight in the code supports a 5-design. To satisfy the conditions of
the corollary we choose \( i = 4 \) and consider the HG 3-design \( \sum_{\tau \in S} P_4(\tau; n) A_\tau \).

Since

\[
P_4(x; n) = 10626 - 4064x + (1660/3) x^2 - 32xt + (2/3) x^4,
\]

the design can be expressed as

\[
A_8 + (66) A_{12} + (-126) A_{16} + (10626) A_{24},
\]

and it is readily calculated that \( \lambda_4 = 0, i = 1, 2, 3 \). In this case however, since each weight supports a 5-design, the linear sum is a HG 5-design (a fact neither predicted nor prevented by Theorem 3.1) and we find that \( \lambda_4 = 2171 \) and \( \lambda_5 = 3967 \).

From Eqs. (3.1) and (3.2) the dependence of the parameters \( \lambda_j \) of the design on the weight enumeration of the code is shown explicitly. However, this is misleading as these parameters can, in fact, be calculated without any reference to the weight enumeration of the code as the following lemma shows.

**Lemma 3.5.** Let \( f(x) \) be a polynomial of degree \( i \). The parameter \( \lambda_j \) of the HG \( (d' - 1 - i) \)-design \( \sum_{\tau \in S} f(\tau) A_\tau \) is given by

\[
\lambda_j = |C| \sum_{l=0}^{n} \binom{n}{l} \binom{i}{j} f(l).
\]

**Proof.** From (3.1)

\[
\binom{n}{j} \lambda_j = \sum_{\tau \in S} \binom{\tau}{j} f(\tau) A_\tau, \quad j = 0, 1, ..., d' - 1 - i, \tag{3.4}
\]

and note that \( \binom{\tau}{j} f(x) \), as a polynomial of degree \( i + j \), can be expressed as

\[
\binom{x}{j} f(x) = \sum_{k=0}^{\frac{i+j}{2}} \Psi_k \binom{x}{k}.
\]

Substituting this into (3.4) gives

\[
\binom{n}{j} \lambda_j = \sum_{\tau \in S} \left\{ \sum_{k=0}^{\frac{i+j}{2}} \Psi_k \binom{\tau}{k} \right\} A_\tau = \sum_{k=0}^{\frac{i+j}{2}} \Psi_k \left\{ \sum_{\tau \in S} \binom{\tau}{k} A_\tau \right\}. \tag{3.5}
\]

Now the code \( C \) is an orthogonal array of strength \( d' - 1 \) and so, for any \( k \leq d' - 1 \), by a counting argument, we have

\[
\sum_{\tau \in S} \binom{\tau}{k} A_\tau = |C| 2^{-k} \binom{n}{k}.
\]
Using this expression in (3.5) and noting that 

\[
\binom{n}{k} \lambda_j = \sum_{k=0}^{i+j} \Psi_k | \mathcal{C} | 2^{-k} \binom{n}{k} = | \mathcal{C} | 2^{-n} \sum_{k=0}^{i+j} \Psi_k 2^{-k} \binom{n}{k} 
\]

\[
= | \mathcal{C} | 2^{-n} \sum_{k=0}^{i+j} \Psi_k \left\{ \sum_{l=0}^{n} \binom{n}{l} \binom{l}{k} \right\} = | \mathcal{C} | 2^{-n} \binom{n}{l} \sum_{k=0}^{n} \left\{ \sum_{l=0}^{i+j} \Psi_k \binom{l}{k} \right\}
\]

\[
= | \mathcal{C} | 2^{-n} \sum_{l=0}^{n} \binom{n}{l} \binom{l}{j} f(l)
\]

as we were required to show.

The following example illustrates the construction of some HG $t$-designs from a certain class of codes.

**Example.** Extremal codes (Mallows and Sloane, 1973) are a class of binary self-dual codes with parameters $n = 24m$, $d = 4(m + 1)$, $s = 4m$, and $\tau_s = n$, with all nonzero code word weights divisible by 4. Each nonzero weight supports a 5-design and the complete code is an orthogonal array of strength $4m + 3$. Define the subset $R_{l,r}$ of $S$, the set of code word weights with the all-ones vector deleted, as

\[
R_{l,r} = \{\tau_l, \tau_{l+1}, \ldots, \tau_{l+r-1}\}
\]

and let

\[
g_{l,r}(x) = \prod_{\tau \in S \setminus R_{l,r}} (x - \tau) = \prod_{\tau \in S \setminus R_{l,r}} (x - \tau)
\]

which is a polynomial of degree $4m - 1 - r$. From Theorem 3.1, \( \sum_{\tau \in S} g_{l,r}(\tau) A_\tau \) is a HG $t$-design for $t = r + 4$ and the only weights with nonzero multiplicities assigned are those of $R_{l,r}$. For $\tau \in R_{l,r}$

\[
g_{l,r}(\tau) = \prod_{\sigma \in S \setminus R_{l,r}} (\tau_l - \tau) = \prod_{j=1}^{l-1} (d + 4(i - 1) - (d + 4(j - 1)))
\]

\[
\times \prod_{j=l+1}^{4m-1} (d + 4(i - 1) - (d + 4(j - 1)))
\]

\[
= 4^{4m-1-r} \prod_{j=1}^{l-1} (i-j) \prod_{j=l+1}^{4m-1} (i-j)
\]

\[
= 4^{4m-1-r}(-1)^{l+r} \left( \binom{r-1}{i-1} \right) \left( \binom{4m-1}{i} i(r-1)! \right).
\]

Note that the polynomials of (3.6), for $l = 1, 2, \ldots, s - r + 1$, always define
HG \((r + 4)\)-designs on consecutive weights and that the sign of the multiplicities attached to the weights is always the same. These polynomials can be defined for any code, of course, and the same observations made.

It has been observed that the individual weights of a binary linear code support a \(t\)-design for \(t = \max(d' - \bar{s}, d - \bar{s}')\) but it is not known whether this is the maximum \(t\) possible or not. For a certain class of HG designs the following result is of interest. We adopt the notation that for a set \(R \subseteq S\) the polynomial \(h_R(x)\) is defined by

\[
h_R(x) = \prod_{\tau \in R} (x - \tau), \quad \bar{R} = \bar{S} \setminus R.
\]

**Theorem 3.6.** For the binary linear code \(C\) let \(d' = 2e' + 1\) and \(r < e' - 1\). Then the HG \((d' - 1 - r)\)-design \(\sum_{\tau \in \bar{S}} h_{\bar{R}}(\tau) A_{\tau}, \ |R| = r\), cannot be a HG \((d' - r)\)-design.

**Proof.** Recall Delsarte’s relation

\[
\hat{\delta}(x) = \sum_{y \in C} a(w(x)) \langle x, y \rangle = |C| \sum_{i=0}^{n} \alpha_i B'(x, i).
\]

For \(a(x) = h_R(x)\) and \(x \in F_2^n\), \(w(x) = d' - r\), since \(a(x)\) is a polynomial of degree \(r\), \(\alpha_i = 0, i > r\), and

\[
\hat{\delta}(x) = |C| \alpha_r B'(x, r)
\]

and \(B'(x, r)\) is the number of code words of weight \(d'\) of \(C'\) covering \(x\). If \(\sum_{\tau \in \bar{S}} h_{\bar{R}}(\tau) A_{\tau}\) is a HG \((d' - r)\)-design (by construction it is only a HG \((d' - 1 - r)\)-design), then by Lemma 3.2, \(B'(x, r)\) is a constant depending only on \(d' - r\) and not on \(x \in F_2^n\). Consequently the code words of weight \(d'\) in \(C'\) form a \((d' - r)\)-design. By a simple distance argument it follows that

\[
d' - 1 - r \leq e' + 1 \quad \text{or} \quad r \geq e' - 1
\]

contrary to assumption, which completes the proof.

The following theorem establishes a result which constrains the maximum \(t\) for which each weight class supports an ordinary \(t\)-design.

**Theorem 3.7.** Let \(C\) be a linear code for which \(d' - \bar{s} > d - \bar{s}'\). A necessary and sufficient condition that none of the weight classes of \(C\) supports a \(t\)-design for \(t > d' - \bar{s}\) is that the minimum weight class of \(C'\) supports a \((d' - \bar{s})\)-design (and not a \(t\)-design for \(t > (d' - \bar{s})\)).

**Proof.** For the polynomial

\[
h_{s_1}(x) = \prod_{\tau \in \bar{S} \setminus \{d\}} (x - \tau),
\]
the Delsarte relation can be written as

$$\delta(x) = \sum_{y \in C} h_{\tau_i}(y) \langle x, y \rangle = h_{\tau_i}(\tau_i) \sum_{y \in A_{\tau_i}} \langle x, y \rangle$$

$$= |C| \sum_{j=0}^{\bar{s}} h_j B'(x, j) - h_{\tau_i}(n)(-1)^{w(x)} A_n .$$

If $w(x) \leq d' - \bar{s}$ then, since $h_{\tau_i}(x)$ is a polynomial of degree $\bar{s} - 1$ and $h_j = 0$, $j > \bar{s} - 1$, the Delsarte relation is

$$\delta(x) = |C| h_{w(x)} - h_{\tau_i}(n)(-1)^{w(x)} A_n$$

as the all-zero code word is the only one at distance less than or equal to $\bar{s} - 1$ from $x$. If $w(x) = d' - \bar{s} + 1$ then

$$\delta(x) = |C| (h_{w(x)} + h_{\bar{s}-1} B'(x, \bar{s} - 1)) - h_{\tau_i}(n)(-1)^{w(x)} A_n ,$$

(3.7)

where $d' - \bar{s} + 1 < \bar{s} - 1$ or $d'/2 < \bar{s} - 1$, otherwise

$$\delta(x) = |C| h_{\bar{s}-1} B'(x, \bar{s} - 1) - h_{\tau_i}(n)(-1)^{w(x)} A_n .$$

(3.8)

Suppose now the code words of weight $d'$ in $C'$ support a $(d' - \bar{s})$-design. If the code words of weight $\tau_i$ in $C$ support a $t$-design, $t > (d' - \bar{s})$ then, from Lemma 3.2, $\delta(x)$ is a constant, depending only on $w(x) = d' - \bar{s} + 1$ and independent of $x$, and so $B'(x, \bar{s} - 1)$, which is the number of code words of $C'$ of weight $d'$ which cover $x$, is also constant. Thus these code words support a $(d' - \bar{s} + 1)$-design which is a contradiction and the code words of weight $\tau_i$ in $C$ cannot support a $(d' - \bar{s} + 1)$-design.

Conversely, suppose the code words of weight $d'$ in $C'$ support a $(d' - \bar{s} + 1)$-design. In this case, $B'(x, \bar{s} - 1)$, the number of code words of weight $d'$ in $C'$ which cover $x$, is a constant, implying that $\delta(x)$ of either (3.7) or (3.8) is a constant. By Lemma 3.3, since

$$\sum_{y \in C, w(y) = \tau_i} \langle x, y \rangle$$

is a constant, the code words of weight $\tau_i$ in $C$ support a $(d' - \bar{s} + 1)$-design for $\tau_i \in S$, and the proof is complete.

**Corollary 3.8.** In a linear code $C$, if one weight supports a $(d' - \bar{s} + 1)$-design then all weights in both $C$ and $C'$ support $(d' - \bar{s} + 1)$-designs.

The proof of the corollary follows immediately from that of Theorem 3.7. To this point all HG designs have been polynomial in the sense that the multiplicities associated with weight $\tau$ in the HG $t$-design are found by evaluating
a polynomial of degree at most \(d' - 1 - t\). The following theorem indicates a set of conditions for which this is always true.

**Theorem 3.9.** Let \( C \) be a linear code such that the code words of weight \(d'\) in \( C' \) support a \( t \)-design for \( t = d' - \bar{s} \), assumed to be greater than or equal to \( d - \bar{s} \), and for no larger value of \( t \). Then for any HG \( t \)-design \( \sum_{\tau \in S} m_\tau A_\tau \), there exists a polynomial \( f(x) \) of degree at most \(d' - 1 - t\) such that \( f(\tau) = m_\tau \) for all \( \tau \in \bar{S} \).

**Proof.** Let \( \sum_{\tau \in R} m_\tau A_\tau \), \( |R| = r \), denote a HG \( t \)-design on \( r \) weights and let \( f(x) \) be the characteristic polynomial of the design, i.e., \( f(\tau) = 0 \), \( \sigma \in \bar{S}\setminus R \) and \( f(\tau) = m_\tau \), \( \tau \in R \). Using Delsarte's relation we enumerate the various possibilities:

1. \( \omega(x) \leq d' - \bar{s} : \delta(x) = |C| f_{\omega(x)} - f(n)(-1)^{\omega(x)}A_n \),
2. \( \omega(x) = d' - \bar{s} + 1 : \delta(x) = |C| (f_{\omega(x)} + f_{d-1}B'(x, \bar{s} - 1)) - f(n)(-1)^{\omega(x)}A_n \),
3. \( \omega(x) = d' - \bar{s} + 2 : \delta(x) = |C| (f_{\omega(x)} + f_{d-2}B'(x, \bar{s} - 2)) + f_{d-1}B'(x, \bar{s} - 1)) - f(n)(-1)^{\omega(x)}A_n \),
4. \( \omega(x) < \frac{d'}{2} : \delta(x) = |C| (f_{\omega(x)} + \sum_{j=0}^{d'-\omega(x)) - d'} f_{d'-\omega(x)+j} \times B'(x, d' - \omega(x) + j)) - f(n)(-1)^{\omega(x)}A_n \),
5. \( \frac{d'}{2} \leq \omega(x) \leq t : \delta(x) = |C| \sum_{j=0}^{\frac{d'}{2} - \omega(x) - d'} f_{d'-\omega(x)+j} \times B'(x, d' - \omega(x) + j)) - f(n)(-1)^{\omega(x)}A_n \),

where \( f_i \) is the \( i \)-th coefficient in the expansion of \( f(x) \) in the basis of Krawtchouk polynomials. In (ii), \( B'(x, \bar{s} - 1) \) is the number of code words of weight \(d'\) in \( C'\) which cover \( x \), \( \omega(x) = d' - \bar{s} + 1 \). Since

\[
\delta(x) = \sum_{y \in C} f(\omega(y)) \langle x, y \rangle = \sum_{\tau \in R} f(\tau) \sum_{y \in C \atop \omega(y) = \tau} \langle x, y \rangle
\]

is a constant for \( \omega(x) \leq t \), depending only on \( \omega(x) \) and not on \( x \), and since for \( \omega(x) = d' - \bar{s} + 1\), \( B'(x, \bar{s} - 1) \) is not a constant as the words of weight \(d'\) in \( C'\) do not form a \( (d' - \bar{s} + 1) \)-design, it follows that \( f_{d-1} \) must be zero as the remaining terms in (ii) are independent of \( x \). Similarly from (iii), as \( B'(x, \bar{s} - 2) \) is the number of words of weight \(d'\) in \( C'\) covering \( x \), \( \omega(x) = d' - \bar{s} + 2 \) we must have \( f_{d-2} = 0 \). Repeating the argument yields that \( f_{d-i} = 0 \),

643/42/3-2
Thus the degree of \( f(x) \) is at most \( d' - 1 - t \) and \( \sum_{\tau \in \mathcal{R}} m_{\tau} A_{\tau}, | R | = r \) is a polynomial HG \( t \)-design, i.e., there does in fact exist a polynomial \( f(x) \), of degree at most \( d' - 1 - t \), such that \( f(\tau) = m_{\tau}, \tau \in \mathcal{S} \).

4. WEIGHTED MAJORITY DECODING AND GENERALIZED \( t \)-DESIGNS

In this section we show how the \( G \) \( t \)-designs can be used to effect a majority decoding algorithm for any binary linear code thereby giving a constructive proof to the Rudolph–Robbins theorem. This was in fact the primary motivation for considering \( G \) \( t \)-designs. To begin with, a previously known majority decoding algorithm (Goethals, 1970; Assmus and Mattson, 1967; and Rahman, 1975) for the binary self-dual \((48, 24)\) quadratic residue code using only the ordinary \( 5 \)-design supported by the code words of minimum weight is considered. The algorithm is incomplete in that it is only able to correct four errors not in the position being corrected whereas the code is capable of correcting five such errors. A similar algorithm capable of correcting these five errors, using a \( G \) \( 6 \)-design on the code words of weight 12 and 16 will then be described. The fact that the code is self-dual plays no significant role in the basic argument. The argument is finally extrapolated to the general case to give the proof of the Rudolph–Robbins theorem.

The \((48, 24)\) QR code has minimum distance 12 and every nonzero code word weight is divisible by 4. It has eight nonzero weights, including the all-ones code word and \( \bar{s} = 7 \). From Theorem 2.1 or 2.2 the code words of each weight support a \( t \)-design where \( t = d' - \bar{s} = 12 - 7 = 5 \). In particular there are 19,256 code words of weight 12 and these form a \( 5 \)-design with parameters \( \lambda_0 = 17,296, \lambda_1 = 4,324, \lambda_2 = 1,012, \lambda_3 = 220, \lambda_4 = 44, \) and \( \lambda_5 = 8 \). For a code word \( x \) of any linear code \( C, (x, y) = \sum_{i=1}^{n} x_i y_i = 0 \) iff \( y \in C' \). In particular if \( y \in C' \) is chosen so that \( y_1 = 1 \) we obtain \( x_1 = \sum_{i=2}^{n} x_i y_i \). This fact will be the basis of the majority decoding algorithm. If a code word \( C \) is transmitted and in transmission incurs errors, the received word \( r \) can be expressed as \( r = c + e \), where \( e \) is the error vector. For any \( y \in C' \), \( (r, y) = (e, y) \) and if \( y_1 = 1 \), \( e_1 = \sum_{i=2}^{n} r_i y_i \) is an estimate of the first error position. The majority decoding scheme then is to determine a set of words in \( C', \{ y^{(i)}, i = 1, 2, \ldots, M \} \), each with a one in the first coordinate position, with the property that the number of check equations \( (r, y^{(i)}) \) which yield 1 is greater than \( M/2 \) if \( e_1 = 1 \), and is less than \( M/2 \) if \( e_1 = 0 \), provided that \( e \) or fewer errors have been made in transmission. Such a set of parity checks can be determined by considering the combinatorial structure of the code.

For the \((48, 24)\) QR code note that there are 4324 code words containing a 1 in the first position. Suppose that \( i \) errors have been made in transmission, \( 0 \leq i \leq 4 \), and that none of these appear in the first position. The number of parity checks that yield a 1 is then
while the number that yield a zero is
\[ \sum_{j=0}^{t} (-1)^j \binom{i}{j} \lambda_{j+1, i-j}. \]

As long as the difference No. zeros \(= No. ones = \sum_{j=0}^{t} (-1)^j \binom{i}{j} \lambda_{j+1, i-j} > 0 \), the algorithm will yield the correct result. If \(i + 1\) errors have been made in transmission, \(0 \leq i \leq 3\), of which one error is in the position being corrected, then as long as the difference No. ones \(= No. zeros = \sum_{j=0}^{t} (-1)^j \binom{i}{j} \lambda_{j+1, i-j} > 0 \) the algorithm will again yield the correct estimate for \(e_i\). Consequently, a necessary and sufficient condition for the algorithm to yield the correct estimate is that
\[ \sum_{j=0}^{t} (-1)^j \binom{i}{j} \lambda_{j+1, i-j} > 0, \quad i = 0, 1, \ldots, e. \]

Since \(\lambda_{j+1, i-j} = \sum_{k=0}^{i-j} (-1)^k \binom{i-j}{k} \lambda_{j+1+k}\), this expression can be reduced to
\[ C(i) = \sum_{j=0}^{t} (-1)^j \binom{i}{j} \sum_{k=0}^{i-j} (-1)^k \binom{i-j}{k} \lambda_{j+1+k} \]
\[ = \sum_{j=0}^{t} \sum_{k=0}^{i-j} (-1)^{i+j+k} \binom{i}{j} \binom{i-j}{k} \lambda_{j+1+k}. \]

Note that to determine these parameters for a \(t\)-design it is necessary that \(j + i \leq t\) implying that at most \((t-1)\) errors can be corrected. For the \((48, 24)\) QR code the function \(C(i)\) can be calculated for \(i = 0, 1, 2, 3, 4\) \((t = 5)\) with the results:

<table>
<thead>
<tr>
<th>(i)</th>
<th>(C(i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4324</td>
</tr>
<tr>
<td>1</td>
<td>2300</td>
</tr>
<tr>
<td>2</td>
<td>936</td>
</tr>
<tr>
<td>3</td>
<td>540</td>
</tr>
<tr>
<td>4</td>
<td>228</td>
</tr>
</tbody>
</table>

Clearly the function \(C(i)\) satisfies the required condition and the algorithm is able to correct four errors. We are unable to use the algorithm to correct the five errors that the code is capable of correcting since the code words of each weight only form five designs.
This problem can be overcome by using the $G_t$-designs of the previous section. In this case we require a $G_6$-design to correct five errors and this can be accomplished by assigning multiplicities to the two lowest weights of 12 and 16 by the polynomial

$$f(x) = \prod_{j=5}^{9} (x - 4j),$$

i.e., the code words of weight 12 are assigned the multiplicity $f(12) = 8.12.16.20.24$ and those of weight 16 the multiplicity $f(16) = \frac{1}{6}f(12)$, using the result of Theorem 3.1. It is readily seen that we can accomplish the same effect by assigning the multiplicities 6 and 1 to the weights 12 and 16, respectively.

For this 6-design the parameters $\lambda_i$ are given by the expression

$$\lambda_i = \frac{6 \binom{12}{i} A_{12} + \binom{16}{i} A_{16}}{\binom{48}{i}}, \quad A_{12} = 17,296, \quad A_{16} = 535,095,$$

and $\lambda_0 = 638,871$, $\lambda_1 = 204,309$, $\lambda_2 = 62,997$, $\lambda_3 = 18,645$, $\lambda_4 = 5,269$, $\lambda_5 = 1,413$, and $\lambda_6 = 357$. In using the majority decoding algorithm it is again only necessary that $C(i) > 0$ for $0 \leq i \leq 5$ in order to correct five errors, the full capability of the code. The tabulation of the function $C(i)$ is:

<table>
<thead>
<tr>
<th>$i$</th>
<th>$C(i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>204,309</td>
</tr>
<tr>
<td>1</td>
<td>78,315</td>
</tr>
<tr>
<td>2</td>
<td>26,901</td>
</tr>
<tr>
<td>3</td>
<td>7,915</td>
</tr>
<tr>
<td>4</td>
<td>1,813</td>
</tr>
<tr>
<td>5</td>
<td>235</td>
</tr>
</tbody>
</table>

and the algorithm does indeed correct the five errors. The important point to note in this example is that in order to prove the algorithm corrects five errors we required a $G_6$-design and considering only the vectors of minimum weight, an ordinary 5-design was not sufficient.

It is a relatively straightforward matter to extend the method of this example to give a proof of the Rudolph–Robbins theorem. The proof is constructive in the sense that it gives necessary and sufficient conditions on a polynomial to yield a $HG(e + 1)$-design which can be used in the error correction algorithm. It is then a simple matter to construct a polynomial to satisfy these conditions.
THEOREM 4.1 (Rudolph–Robbins, 1972). Let $C$ be a binary linear $(n, k)$ code with minimum distance $d = 2e + 1$. Then $C$ is one-step weighted majority decodable.

Proof. As in the previous example assume the first position is being weighted majority decoded with a set of parity checks. As before, to obtain a correct decision when up to $e$ errors are made it is necessary that

$$
\sum_{j=0}^{i} (-1)^j \binom{i}{j} \lambda_{j+1, i-j} > 0, \quad i = 0, 1, \ldots, e. \tag{4.1}
$$

To construct the $HG (e + 1)$-design to achieve the decoding, we first extend $C$ to an $(n + 1, k)$ linear code with minimum distance $d = 2e + 2$, still capable of correcting only $e$ errors. The dual code $C'_e$ is an orthogonal array of strength $2e - 1$. If $f(x)$ is a polynomial of degree $e$ then from Theorem 3.1, $\sum_{\sigma \in S} f(\sigma) A'_e(\sigma)$ is a $HG (e + 1)$-design. We now investigate conditions on this polynomial which will ensure the design can be used in majority decoding. From Lemma 3.3 we have

$$
\lambda_{i,j} = \sum_{\sigma \in S'} f(\sigma) A'_e(\sigma) \binom{n + 1 - i - j}{\sigma - i} / \binom{n + 1}{\sigma}
$$

and substituting this into Eq. (4.1) shows that for correct decisions by majority decoding we require that

$$
\sum_{j=0}^{i} (-1)^j \binom{i}{j} \sum_{\sigma \in S'} f(\sigma) A'_e(\sigma) \binom{n + 1 - (j + 1) - (i - j)}{\sigma - (j + 1)} / \binom{n + 1}{\sigma}
$$

$$
= \sum_{\sigma \in S'} f(\sigma) \left[ A'_e(\sigma) / \binom{n + 1}{\sigma} \right] \sum_{j=0}^{i} (-1)^j \binom{i}{j} \binom{n - i}{\sigma - 1 - j} > 0.
$$

By some rearranging of terms this last condition can be written as

$$
\sum_{\sigma \in S'} f(\sigma) \left[ A'_e(\sigma) / \binom{n + 1}{\sigma} \right] \sum_{j=0}^{i} (-1)^j \binom{\sigma - 1}{j} \binom{n - (\sigma - 1)}{i - j}
$$

$$
= \left[ 1 / \binom{n}{i} \right] \sum_{\sigma \in S'} g(\sigma) P_i(\sigma - 1; n),
$$

where

$$
g(\sigma) = f(\sigma) A'_e(\sigma) / \binom{n + 1}{\sigma} = f(\sigma) A'_e(\sigma) n + 1
$$

and an alternative expression for the Krawtchouk polynomial of degree $i$ is used (MacWilliams and Sloane, 1977, p. 130). For correct decisions by majority
decoding then it is sufficient to show the existence of a polynomial \( f(x) \) of degree \( e \) such that

\[
\sum_{\sigma \in S'} \sigma A_{\sigma} P_i(\sigma - 1; n) > 0, \quad i = 0, 1, \ldots, e. \tag{4.2}
\]

Using the fact (MacWilliams and Sloane, 1977, p. 153) that

\[
P_j(\sigma - 1; n) = \sum_{t=0}^{j} P_t(\sigma; n + 1)
\]

and letting

\[
f(\sigma) = \sum_{j=0}^{e} f_j P_j(\sigma; n + 1),
\]

substituting into (4.2) yields

\[
\sum_{\sigma \in S'} \sigma A_{\sigma} \sum_{j=0}^{e} f_j P_j(\sigma; n + 1) \left( \sum_{t=0}^{i} P_t(\sigma; n + 1) \right) > 0. \tag{4.3}
\]

Since \( P_j(\sigma; n + 1) P_t(\sigma; n + 1) \) is a polynomial of degree \( j + l \) we can write

\[
P_j(\sigma; n + 1) P_t(\sigma; n + 1) = \sum_{k=0}^{j+l} P_k^1 P_k(\sigma; n + 1)
\]

and substituting into (4.3) gives the condition

\[
\sum_{\sigma \in S'} \sigma A_{\sigma} \sum_{j=0}^{e} f_j \sum_{l=0}^{i} \sum_{k=0}^{j+l} P_k^1 P_k(\sigma; n + 1) > 0, \quad i = 0, 1, \ldots, e,
\]

or

\[
\sum_{l=0}^{i} \sum_{j=0}^{e} f_j \sum_{k=0}^{j+l} \left\{ \sum_{\sigma \in S'} \sigma A_{\sigma} P_k(\sigma; n + 1) \right\} > 0, \quad i = 0, 1, \ldots, e. \tag{4.4}
\]

Using the recurrence relationship (MacWilliams and Sloane, 1977, p. 152)

\[
2\sigma P_k(\sigma; n + 1) = -(k + 1) P_{k+1}(\sigma; n + 1) + (n + 1) P_k(\sigma; n + 1) - ((n + 1) - k + 1) P_{k-1}(\sigma; n + 1)
\]

it follows that

\[
\sum_{\sigma \in S'} \sigma A_{\sigma} P_k(\sigma; n + 1)
\]

\[
= \frac{1}{2} [-(k + 1) A_{k+1} + (n + 1) A_k - ((n + 1) - k + 1) A_{k-1}],
\]
where \( A_i, \ i = 0, 1, ..., n + 1 \) is the weight distribution of the extended code. Since the minimum distance of the code is \( 2e + 2 \), for \( 0 < k \leq 2e \) we have

\[
\sum_{\sigma \in S'} \alpha A'_\sigma P_k(\sigma; n + 1) = 0,
\]

while for \( k = 0 \)

\[
\sum_{\sigma \in S'} \alpha A'_\sigma P_0(\sigma; n + 1) = \sum_{\sigma \in S'} \alpha A'_\sigma = \frac{|C'|}{2} (n + 1).
\]

(4.5)

Since \( k \leq j + l \leq e + l \leq 2e \), substituting (4.5) into (4.4) gives

\[
\frac{|C'|}{2} (n + 1) \sum_{j=0}^{e} \sum_{l=0}^{i} f_j p_{jl} > 0, \quad i = 0, 1, ..., e.
\]

(5.1)

But \( p_{jl} = \delta_{j,l} \) and \( i \leq e \) and it follows that

\[
\frac{|C'|}{2} (n + 1) \sum_{j=0}^{e} \sum_{l=0}^{i} p_{jl} = \frac{|C'|}{2} (n + 1) \sum_{j=0}^{i} f_j > 0, \quad i = 0, 1, ..., e,
\]

i.e., the polynomial \( f(x) \) may be used for majority decoding if

\[
\sum_{j=0}^{i} f_j > 0, \quad i = 0, 1, ..., e.
\]

This condition is trivially satisfied if \( f_i > 0, i = 0, 1, ..., e \) and the proof is complete.

5. Comments

Generalized \( t \)-designs, while of interest in their own right as a combinatorial structure, have provided a constructive solution to the Rudolph–Robbins majority decoding theorem. This solution will in general be very inefficient and seldom used in practice. The question remains as to whether some other approach to majority decoding using weighted nonorthogonal parity checks will give a solution which is competitive with other methods.

Received: September 15, 1978; Revised: March 2, 1979

References


