Chapter 7

Block Codes

In this chapter we give a brief introduction to block codes. Basically the codes consist of a number of vectors. The goal is to have as many vectors as possible in the code but with each vector as far apart from every other vector as possible. This is equivalent to finding a communication system with high efficiency (bits/second/Hz) and low error probability (for a given signal-to-noise ratio). There is a tradeoff between these two parameters of a code. The first section deals with quantifying this tradeoff.

Decoding a general block code involves finding the codeword closest (in some sense) to the received vector. For most block codes the decoding complexity is very large. For example the Reed-Solomon codes used in compact disc players have as many as $(2^8)^{28} = 2^{224} \approx 10^{67}$ codewords. Thus comparing the received vector with all possible codewords is not practical. Because of this the codes usually considered have some structure that makes the decoding less complex. So instead of consider all possible block codes we consider certain subsets of block codes. The first subset of block codes we consider is linear codes. We show how to decode linear code with less complexity (for high rates) than general block codes. Next we examine cyclic codes which have even less decoding complexity than linear codes (when using bounded distance decoding). We also look at some nonlinear codes that can be decoded without too much difficulty because of their relative small size. Among these are the Nordstrom-Robinson code and codes for QAM type of signaling.

Def: An $[M,n]$ block code over $\{0,1,\ldots,q-1\}$ is a set of M vectors of length n with each component being in the alphabet $\{0,1,\ldots,q-1\}$.

Def: The Hamming weight of a vector $\mathbf{x}$ is the number of nonzero components in $\mathbf{x}$. This is denoted by $\omega_H(\mathbf{x})$.

Def: The Hamming distance between two vectors $\mathbf{x}$ and $\mathbf{y}$ is the Hamming weight of their difference $d_H(\mathbf{x},\mathbf{y}) = \omega_H(\mathbf{x}-\mathbf{y})$.

Def: The minimum distance of a code is the minimum Hamming distance between two distinct codewords.

Thm: A code with minimum distance d_{min} can correct all error patterns with e or fewer errors provided $2e+1 \leq d_{\min}$.

Proof: Clearly if a code has distance $2e+1$ and e or fewer errors occurred the received vector will be closer to the transmitted codeword than any other codeword and will be decoded correctly.

Def: The rate (in bits/channel use) of a $[M,n]$ block code is $r = \frac{\log_2 M}{n}$.

We would like codes with large distances and many codewords. However, there is a fundamental tradeoff between the number of vectors and the minimum distance between vectors. This tradeoff is examined in the next section.

1. Bounds on Distance and Rate of Codes

In this section we examine the tradeoff between the number of vectors in a vector space and the minimum distance between the vectors.

1. Hamming bound: If a code can correct e errors ($d_{\min} \geq 2e+1$) the spheres of radius e around each codeword must be disjoint. The union of all of these spheres must contain less than 2^n points (the total number of vectors of length n over $\{0,1\}$).

$$\underset{\substack{\uparrow \\ \text{number of} \\ \text{codewords}}}{M} \quad \left(\underbrace{1+\binom{n}{1}+\cdots+\binom{n}{e}}_{\substack{\text{number of points in} \\ \text{sphere of radius } e}}\right) \quad \leq \quad \underbrace{2^n}_{\substack{\text{total number of} \\ \text{points in space.}}}$$

2. Gilbert bound: Consider the densest packing of spheres of radius e in the space of vectors of length n over $\{0,1\}$. Let $x_0, x_1, \ldots, x_{M-1}$ be the center of the spheres. Consider the spheres of radius $2e$ around these points. These spheres may not be disjoint but must completely fill the space (if not there would be a point with distance $\geq 2e$ from $x_0, \ldots, x_{M-1}$ which could be added to the packing thus making it not the densest packing).

$$M\left(1+\binom{n}{1}+\cdots+\binom{n}{2e}\right) \geq 2^n$$

3. Singleton bound (for linear codes): $d_{\min} \leq n-k+1$

Since a linear code is a k dimensional subspace there is a generator matrix G of the form $G = [I_k A]$. Two codewords can possibly differ in only *one* place in the first k components and disagree in at most $n-k$ places in the last $n-k$ components. Thus $d_{\min} \leq n-k+1$.

Def: A code with $d_{\min} = n-k+1$ is called a maximum-distance separable code. (MDS code)

4. Plotkin bound: $d_{\min} \leq \frac{m}{m-1}\binom{n}{2}$ (No proof).

5. Elias bound (will be given only in its asymptotic form).

Asymptotic forms of the bounds:

Let $n, M, d \to \infty$ such that $\frac{\log_2 M}{n} \to R$ and $\frac{d_{\min}}{n} \to \delta$ then the above bound can be written as:

1. Hamming bound: $R \leq 1 - H_2(\delta/2)$
2. Gilbert bound: $R \geq 1 - H_2(\delta)$
3. Singleton bound: $R \leq 1 - \delta$
4. Plotkin: $R \leq 1 - 2\delta$ $(0 \leq \delta \leq 1/2)$
5. Elias: $R \leq 1 - H_2(\omega_0)$, $\omega_0 = \frac{1}{2}\left[1 - \sqrt{1-2\delta}\right]$
6. McEliece et. al. bound

$$R \leq min_{0 \leq u \leq 1-2\delta}\{1 - h(u^2) - h(u^2 + 2\delta u + 2\delta)\}$$

where $h(x) = H_2(\frac{1}{2}(1-\sqrt{(1-x)})$ and $H_2(p) = -p\log_2(p) - (1-p)\log_2(1-p)$.

The graphs of these bounds are shown below. The Gilbert bound guarantees existence of codes on or above the lower line while the other bounds are upper limits on the rate of a code for a given distance.

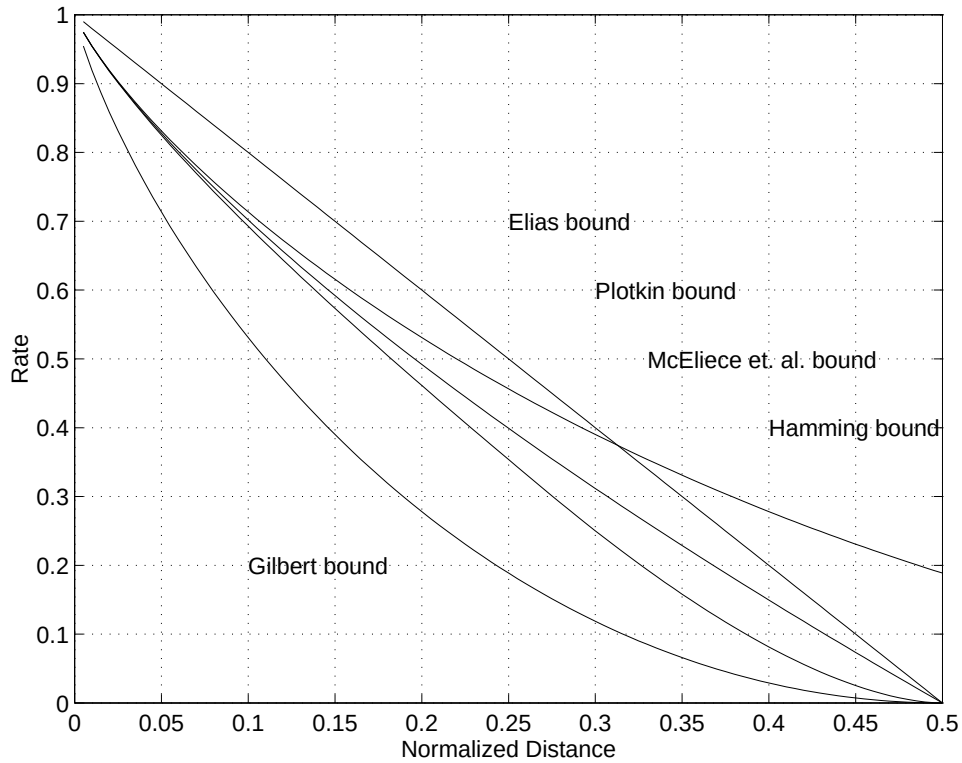


Figure 7.1: Bounds on the Size of Block Codes

2. Linear Codes

Def: An (n,k) linear code over $\{0,1,\ldots,q-1\}$ is a k dimensional subspace of the n dimensional vector space of vectors with components in $\{0,1,\ldots,q-1\}$. (q now must be a prime number or a power of a prime number)

Equivalent definition: An (n,k) linear code over $\{0,1,\ldots,q-1\}$ is a $[q^k,n]$ block code for which the sum of any two codewords is also a codeword. (If q is a prime addition is done "$\bmod q$". If q is not a prime more complicated addition is required).

Def: Let C be a linear code. A matrix G whose rowspace equal C is called a generator matrix for C. (The rowspace of a matrix is the set of vectors that are linear combinations of the rows of the matrix). G is a $k \times n$ matrix

Def: Let G be an (n,k) linear code. A $(n-k \times n)$ matrix H such that $H\mathbf{x}^T = 0$ for all $\mathbf{x} \in C$ is called the parity check matrix for C.

Fact: If G is of the form $G = [I_k A]$ then H is of the form $H = [-A^T I_{n-k}]$ where I_k is the $k \times k$ identity matrix and A is a $k \times (n-k)$ matrix.

Def: The repetition code is a linear $(n,1)$ code with two codewords

$$C = \{(0000\ldots00),(1111\ldots11)\}$$

The rate of the code is $1/n$ bits/channel use.

Def: The following codes are called "trivial codes".

1. $C = \{\mathbf{x}_0\}$ for any x_0.
2. $C =$ repetition code.
3. $C = \{\mathbf{x} : x \text{ is a vector of length } n\} =$ the whole space.

3. Decoding Block Codes

The operation of decoding a block code is
Decoding Linear Codes on a BSC:
Let $\mathbf{x}$ be the transmitted codeword; $\mathbf{x} \in C$.
Let $\mathbf{e}$ be the error pattern.
Let $\mathbf{y}$ be the received vector.

$$\mathbf{y} = \mathbf{x} + \mathbf{e}$$

Decoding Algorithm:

1. Compute $\mathbf{s}^T = H\mathbf{y}^T$
2. Find the set of vectors such that $H\mathbf{e}^T = \mathbf{s}^T$ (called a coset).
3. Choose the vector $\mathbf{e}_0$ in the set that is most probable (minimum Hamming weight if channel is BSC).
4. Output the codeword $\hat{\mathbf{x}} = \mathbf{y} - \mathbf{e}_0$.

Def: A $[M,n]$ code with minimum distance $d_{\min}$ is said to be perfect if

$$M\left(\sum_{i=0}^{e}\binom{n}{i}\right) = 2^n$$

where

$$e = \lfloor \frac{d_{\min}-1}{2} \rfloor$$

Fact: Besides the trivial codes the only perfect codes are the Hamming codes and Golay codes. (Repetition is perfect only when n is odd).

Hamming codes: $n = 2^m - 1$, $k = 2^m - 1 - m$, $d_{\min} = 3 \Rightarrow e = 1$, $M = 2^k = 2^{2^m-1-m}$, $M(1+\binom{n}{1}) = 2^{2^m-1-m}(1+2^m-1) = 2^{2^m-1-m}\,2^m = 2^{2^m-1} = 2^n$

Golay codes: a) $n = 23$, $k = 12$ $d_{min} = 7$.

$$1 + \binom{23}{1} + \binom{23}{2} + \binom{23}{3} = 2^{11}\,2^{12} \cdot 2^{11} = 2^{23} = 2^n$$

b) $\boxed{q = 3}$ $\quad n = 11 \quad k = 6 \;\; d_{\min} = 5 \;\; e = 2$

$$1 + (q-1)\tbinom{n}{1} + (q-1)^2\tbinom{n}{2} = q^5 = 3^5$$
$$1 + 2\tbinom{11}{1} + 4\tbinom{11}{2} = 3^5$$
$$\underbrace{3^6}_{M}\,3^5 = 3^{11} = q^n$$

Note: Showing that $M\sum_{i=0}^{e}(q-1)^i\binom{n}{i} = 2^n$ is *not* sufficient to show existence of a perfect code but only the possibility of existence.

Fact: The only binary MDS code are the repetition codes. Reed-Solomon codes are MDS codes but $q > 2$ for Reed-Solomon codes.

Def: The dual of a linear code C with generator matrix G and parity check matrix H is a linear code with generator matrix H and parity check matrix G.

$$\underset{(n,n-k)\text{ code}}{C^\perp} \text{ is dual code of } \underset{(n,k)\text{ code}}{C}$$

Example: (6,3) Code

(6,3) Code

$M = 8$, $n = 6$, $k = 3$.

$$\begin{aligned} \mathcal{C} &= \{(000000),(100101),(010111),(001011), \\ &\quad (110010),(101110),(011100),(111001)\} \end{aligned}$$

Generator Matrix

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

Parity Check Matrix

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

Example of Encoding
Let $\mathbf{m} = (011)$ then $\mathbf{c} = \mathbf{m}G = (011100)$.
Example of Decoding
Let $\mathbf{r} = (011001)$. Compute $\mathbf{s}^T = H\mathbf{r}^T$

$$\mathbf{s}^T = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Coset leader = (100000) thus an error is most likely to have occurred in the first position. Thus the decoder will decide that the transmitted codeword is (111001) and the channel made one error.

The Hamming Code

The Hamming code has the following parity check matrix

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

and generator matrix

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

Let

$$\begin{aligned} \mathbf{v}_1 &= (1,0,0,0,1,0,1) \\ \mathbf{v}_2 &= (0,1,0,0,1,1,1) \\ \mathbf{v}_3 &= (0,0,1,0,1,1,0) \\ \mathbf{v}_4 &= (0,0,0,1,0,1,1) \end{aligned}$$

These are a set of linear independent vectors that generate the code.

An alternative set of linear independent vectors that generate the code is

$$\begin{aligned}
\mathbf{v}_1\prime &= (1,0,1,1,0,0,0) \\
\mathbf{v}_2\prime &= (0,1,0,1,1,0,0) \\
\mathbf{v}_3\prime &= (0,0,1,0,1,1,0) \\
\mathbf{v}_4\prime &= (0,0,0,1,0,1,1)
\end{aligned}$$

To show that these are basis vectors for the code we need to show that they can be generated by the previous basis vectors and are linearly independent. It is easy to see that

$$\begin{aligned}
\mathbf{v}_1\prime &= \mathbf{v}_1 + \mathbf{v}_3 + \mathbf{v}_4 \\
\mathbf{v}_2\prime &= \mathbf{v}_2 + \mathbf{v}_4 \\
\mathbf{v}_3\prime &= \mathbf{v}_3 \\
\mathbf{v}_4\prime &= \mathbf{v}_4
\end{aligned}$$

To show that they are linearly independent consider

$$a_1\mathbf{v}_1\prime + a_2\mathbf{v}_2\prime + a_3\mathbf{v}_3\prime + a_4\mathbf{v}_4\prime, \qquad a_i \in \{0,1\}$$

The only way to get 0 in the first component from this linear combination is if $a_1 = 0$. The only way to get zero in the second component from this linear combination is if $a_2 = 0$. The only way to get zero in the last component from this linear combination is if $a_4 = 0$. Finally with $a_1 = a_2 = a_4 = 0$ the only way for the result to be 0 is if a_3 is also 0. Thus these vectors are linearly independent. These basis vectors are shifts of each other. Also a cyclic shift of $\mathbf{v}_4\prime$ is $\mathbf{v}_1\prime + \mathbf{v}_3\prime + \mathbf{v}_4\prime$. Since the codewords are linear combinations of these basis vectors, a cyclic shift of a codeword is also a linear combination of these basis vectors and thus also a codeword.

The codewords that this code generate are

$$\begin{array}{ll}
(0,0,0,0,0,0,0) & (1,1,1,1,1,1,1) \\
 & \\
(1,0,0,0,1,0,1) & (0,1,0,0,1,1,1) \\
(1,1,0,0,0,1,0) & (1,0,1,0,0,1,1) \\
(0,1,1,0,0,0,1) & (1,1,0,1,0,0,1) \\
(1,0,1,1,0,0,0) & (1,1,1,0,1,0,0) \\
(0,1,0,1,1,0,0) & (0,1,1,1,0,1,0) \\
(0,0,1,0,1,1,0) & (0,0,1,1,1,0,1) \\
(0,0,0,1,0,1,1) & (1,0,0,1,1,1,0)
\end{array}$$

This code is called a cyclic code because every cyclic shift of a codeword is also a codeword. The minimum distance of this code is 3 (the minimum weight of any nonzero codeword). Thus this code is capable of correcting 1 error. If we consider for each codeword the number of received vectors that are decoded into it, then because it corrects any single error, there are 7 received vectors that differ from the codeword in one position. In addition, if the received vector is the codeword itself, then it will be decoded into codeword. Thus there are 8 received vectors that are decoded into each codeword. There are 16 codewords. This then accounts for $8 \times 16 = 128$ received vectors. But there are only $128 = 2^7$ possible received vectors. Thus there are no vectors outside of the single error correction decoding region.

Because this is a single error correcting code, the coset leaders must be all error patterns of weight 1. The coset

leaders are then very easy to identify. They are

Syndrome	Coset Leader
$(0,0,0)$	$(0,0,0,0,0,0,0)$
$(0,0,1)$	$(0,0,0,0,0,0,1)$
$(0,1,0)$	$(0,0,0,0,0,1,0)$
$(0,1,1)$	$(0,0,0,1,0,0,0)$
$(1,0,0)$	$(0,0,0,0,1,0,0)$
$(1,0,1)$	$(1,0,0,0,0,0,0)$
$(1,1,0)$	$(0,0,1,0,0,0,0)$
$(1,1,1)$	$(0,1,0,0,0,0,0)$

Thus to correct a error for this code, compute the syndrome and then identify which column of the matrix H is that syndrome (H contains all possible nonzero columns of length 3). The column that is the syndrome is the place a single error occurred. A double error will never be corrected for this code.

4. Cyclic Codes

Def: A group G is a set of objects for which an operation $\star$ is defined that satisfies

1. If $a \in G, b \in G$ then $a \star b \in G$
2. If $a \in G, b \in G, c \in G$ then $(a \star b) \star c = a \star (b \star c)$
3. $\exists$ an identity element $i \in G$ such that for any $a \in G \;\; i \star a = a \star i = a$
4. Every element $a \in G$ has an inverse $a_i \in G$ such that

$$a \star a_i = a_i \star a = i$$

Def: A group is called Abelian if for any $a \in G, b \in G \; a \star b = b \star a$

Example 1. $G = \{0,1,2,\ldots,q-1\} \;\; \star = + \bmod q (k_i = q - k_i)$ (Abelian)
Example 2.

$\star$	a	b	c
a	a	b	c
b	b	c	a
c	c	a	b

Def: A ring R is a set of elements with two operations ($+$ and $\cdot$) such that

i) R is an Abelian group under addition
ii) $a \in R, b \in R \Rightarrow a \cdot b \in R$
iii) $a, b, \; \in R \Rightarrow a \cdot (b \cdot c) = (a \cdot b) \cdot c$
iv) $a, b, c \; \in R \Rightarrow a \cdot (b + c) = a \cdot b + a \cdot c$

Ex: $R = \{0,1,2,3\}$ is a ring with addition mod 4 and multiplication mod 4.

$+$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

$\cdot$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

Note: 2 does not have an inverse under multiplication for this example.

Def: A field F is a set of elements with two operations ($+$ and $\cdot$) such that

i) F is an Abelian group under $+$ (identity $= 0$)

ii) $F^* = \{x \in F, x \neq 0\}$ is an Abelian group under $\cdot$

iii) $(a+b)c = ac + bc = c(a+b)$

Example: $F = \{0,1,2\}$

	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

$+$

	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

$\cdot$

Example: $F = \{0,1\}$

	0	1
0	0	1
1	1	0

$+$

	0	1
0	0	0
1	0	1

$\cdot$

Example $F = \{0,1,a,b\} = \{(00),(01),(10),(11)\} = \{0,1,x,x+1\}(x^2 = x+1)$

	0	1	a	b
0	0	1	a	b
1	1	0	b	a
a	a	b	0	1
b	b	a	1	0

$+$

	0	1	a	b
0	0	0	0	0
1	0	1	a	b
a	0	a	b	1
b	0	b	1	a

$\cdot$

Def: A finite field is a field with a finite number of elements.

Fact: For any prime power p^m there is up to an isomorphism only one finite field of size p^m (1-1 mapping).

The finite field of size q is denoted by $GF(q)$ (after Galois who was founder).

Def: An ideal over a ring R is a subset S of R such that

i) $f \in S, g \in S \Rightarrow f - g \in S$

ii) $f \in s, g \in R \Rightarrow fg \in S$

Def: A cyclic code C is a subspace of a n-dimensional vector space over $GF(q)$ such that if

$$\begin{aligned} \underline{c} &= (c_0, \ldots, c_{n-1}) \in C \text{ then} \\ \tilde{c} &= (c_1, \ldots, c_{n-1}, c_0) \in C \text{ also } . \end{aligned}$$

Fact: A cyclic code is an ideal . . . . .

Thm: A cyclic code over $GF(q)$ of block length n is a set of polynomials that satisfy: 1) $a(x),\ b(x) \in C \Rightarrow \alpha a(x) + \beta b(x) \in C\ \alpha, \beta \in GF(q)\ a(x) \in C \Rightarrow x^i a(x) \bmod x^n - 1 \in C$.

Thm: A cyclic code consists of all multiples of the *unique* monic polynomial $g(x)$ of smallest degree $\bmod (x^n - 1)$.

Example: (7,4) Hamming code $g(x) = 1 + x^2 + x^3$

$$1 + x + \ldots + x^5 + x^6 = (x^2 + x^3 + x^5)g(x) \bmod x^7 - 1$$

Examples of cyclic codes

1. Hamming code over $GF(2)$:
2. Golay code: $g(x) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1\ \ n = 23$
3. BCH codes: $n = 2^m - 1,\ \ n - k \geq mt,\ \ d_{\min} = 2t + 1$
4. Maximal length shift register codes (dual of Hamming code)

$$n = 2^m - 1\ \ k = m$$

(Also is Hadamard code with first bit dropped)

(Also is Simplex code)

Code	n	k	d_{min}	Rate (r)
Hamming Code	7	4	3	0.5714
Hamming Code	15	11	3	0.7333
Hamming Code	31	26	3	0.8387
Hamming Code	63	58	3	0.9206

Error probability for codes: Use union bound or union-Chernoff bound.

5. Reed-Solomon codes

These are codes over $GF(2^m)$.

$$d_{\min} = N - K + 1 \ \ (\because \text{MDS code})$$

These codes are used in compact digital discs, spread-spectrum systems, computer memories.

Example: Consider the (4,2) Reed Solomon code over GF(5)=$\{0,1,2,3,4\}$. The generator polynomial is

$$g(x) = (x-1)(x-2) = x^2 - 3x + 2 = x^2 + 2x + 2$$

The codewords are

$$\begin{aligned}
(0+0x)g(x) &= 0x^3 + 0x^2 + 0x + 0 \\
(1+0x)g(x) &= 0x^3 + 1x^2 + 2x + 2 \\
(2+0x)g(x) &= 0x^3 + 2x^2 + 4x + 4 \\
(3+0x)g(x) &= 0x^3 + 3x^2 + 1x + 1 \\
(4+0x)g(x) &= 0x^3 + 4x^2 + 3x + 3 \\
(0+1x)g(x) &= 1x^3 + 2x^2 + 2x + 0 \\
(1+1x)g(x) &= 1x^3 + 3x^2 + 4x + 2 \\
(2+1x)g(x) &= 1x^3 + 4x^2 + 1x + 4 \\
(3+1x)g(x) &= 1x^3 + 0x^2 + 3x + 1 \\
(4+0x)g(x) &= 1x^3 + 1x^2 + 0x + 3 \\
(0+2x)g(x) &= 2x^3 + 4x^2 + 4x + 0 \\
(1+2x)g(x) &= 2x^3 + 0x^2 + 1x + 2 \\
(2+2x)g(x) &= 2x^3 + 1x^2 + 3x + 4 \\
(3+2x)g(x) &= 2x^3 + 2x^2 + 0x + 1 \\
(4+2x)g(x) &= 2x^3 + 3x^2 + 2x + 3 \\
(0+3x)g(x) &= 3x^3 + 1x^2 + 1x + 0 \\
(1+3x)g(x) &= 3x^3 + 2x^2 + 3x + 2 \\
(2+3x)g(x) &= 3x^3 + 3x^2 + 0x + 4 \\
(3+3x)g(x) &= 3x^3 + 4x^2 + 2x + 1 \\
(4+3x)g(x) &= 3x^3 + 0x^2 + 4x + 3 \\
(0+4x)g(x) &= 4x^3 + 3x^2 + 3x + 0 \\
(1+4x)g(x) &= 4x^3 + 4x^2 + 0x + 2 \\
(2+4x)g(x) &= 4x^3 + 0x^2 + 2x + 4 \\
(3+4x)g(x) &= 4x^3 + 1x^2 + 4x + 1 \\
(4+4x)g(x) &= 4x^3 + 2x^2 + 1x + 3
\end{aligned}$$

The codewords are easily seen to be cyclic shifts of the following vectors

$$\begin{array}{c} 0\,0\,0\,0 \\ 0\,1\,2\,2 \\ 0\,2\,4\,4 \\ 0\,3\,1\,1 \\ 0\,4\,3\,3 \\ 1\,3\,4\,2 \\ 1\,4\,1\,4 \\ 2\,3\,2\,3 \end{array}$$

Note that because this is a linear code the minimum distance is the minimum weight which is 3. Thus this code can correct one error. Suppose the code is used on a channel and suppose one error is made. Then

$$r(x) = c(x) + e(x)$$

where

$$e(x) = e_j x^j$$

This implies the error has magnitude e_j and occurred in the j-th position. Because $c(x) = g(x)i(x)$ for some polynomial $i(x)$ it is clear that

$$\begin{aligned} r(1) &= g(1)i(1) + e(1) = e(1) = e_j \\ r(2) &= g(2)i(2) + e(2) = e(2) = 2^j e_j \end{aligned}$$

Thus the magnitude of the error is just $r(1)$ and the location of the error can be determined from

$$\log_2\left(\frac{r(2)}{r(1)}\right)$$

For example suppose we receive

$$r(x) = 4x^3 + 3x^2 + 4x^1 + 1$$

Then $r(1) = 4 + 3 + 4 + 1 = 2$ and $r(2) = 2 + 2 + 3 + 1 = 3$. Thus the error magnitude is 2. The error location is determined from

$$r(2)/r(1) = 3/2 = 3 \cdot 2^{-1} = 3 \cdot 3 = 4 = 2^2$$

Thus the error polynomial is $e(x) = 2x^2$. Thus

$$c(x) = r(x) - e(x) = 4x^3 + 3x^2 + 4x^1 + 1 - 2x^2 = 4x^3 + 1x^2 + 4x^1 + 1$$

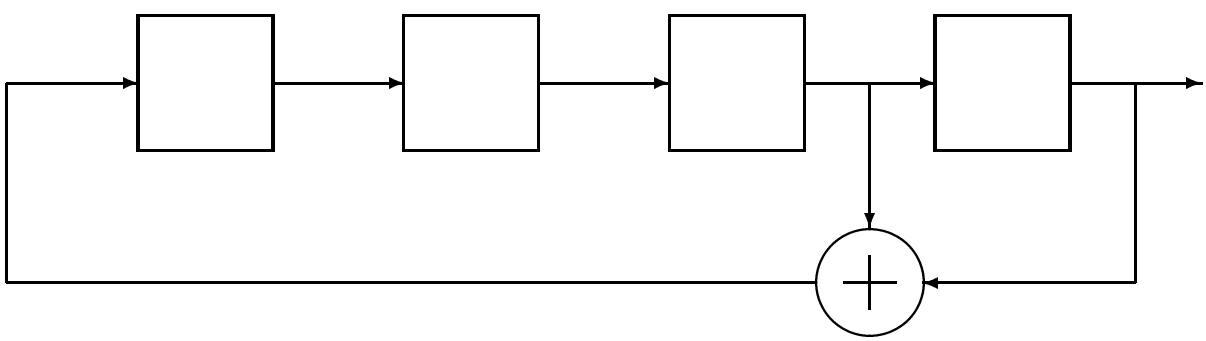
which is indeed a codeword. Here is $GF(2^4)$ using primitive polynomial x^4+x+1.

$\alpha^{-\infty}$	0
α^0	1
α^1	x
α^2	x^2
α^3	x^3
α^4	$x+1$
α^5	x^2+x
α^6	x^3+x^2
α^7	x^3+x+1
α^8	x^2+1
α^9	x^3+x
α^{10}	x^2+x+1
α^{11}	x^3++x
α^{12}	x^3+x^2+x+1
α^{13}	x^3+x^2+1
α^{14}	x^3+1

6. Maximal length sequences (m-sequences)

Maximal length shift register sequences are used in many applications including spread-spectrum systems. They are usually not used as error control codes but as signaling waveforms. The usefulness stems from the nice auto-correlation property m-sequences posses.

Below we show an example of a sequence produced by a shift register of length 4 with feedback. As can be seen, the sequence is periodic with period 15.



0 0 0 1	0 1 0 1
1 0 0 0	1 0 1 0
0 1 0 0	1 1 0 1
0 0 1 0	1 1 1 0
1 0 0 1	1 1 1 1
1 1 0 0	0 1 1 1
0 1 1 0	0 0 1 1
1 0 1 1	0 0 0 1

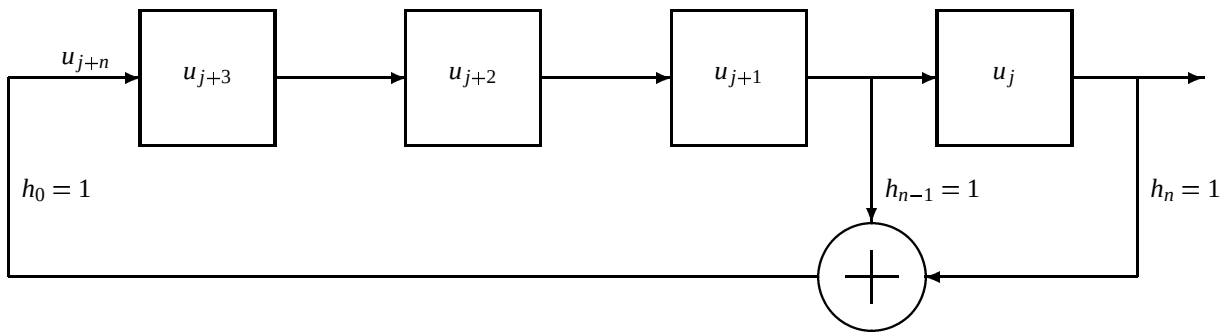
The output of the shift register is the periodic sequence.

1 0 0 0 1 0 0 1 1 0 1 0 1 1 1

In general the feedback connection can be described by a polynomial $h(x)$. Let $h(x) = h_0x^n + h_1x^{n-1} + \cdots + h_{n-1}x + h_n$ with $h_i \in \{0,1\}$. The taps of the shift register that are fed back correspond to $h_i = 1$. We require $h_n = h_0 = 1$ (otherwise we could make the shift register shorter). The sequence at the output of the shift register can be expressed as

$$u_{j+n} = h_1u_{j+n-1} + \cdots + h_{n-1}u_{j+1} + h_nu_j$$

as can be seen below.



The period of the sequence is denoted by N. If the length of the shift register is n the longest period possible for the sequence is $2^n - 1$. This is because the shift register could be in one of 2^n states. However, the all zero state would only generate the all zero sequence. Thus a nonzero sequence could correspond to the shift register going through all the $2^n - 1$ nonzero states before repeating.

Properties of m-sequences

1. The sequence length is $2^n - 1$.

2. The number of ones in the sequence is 2^{n-1} and the number of zeros is $2^{n-1} - 1$.

3. (Shift and Add) The sum of any two (distinct) shifts of a single m-sequence is a different shift of the same sequence.

   To see this consider a sequence out of the shift register with initial state $a_1, ..., a_n$. The sequence generated is

$$\begin{aligned} a_{n+1} &= h_1a_n + \cdots + h_{n-1}a_2 + h_na_1 \\ a_{n+j} &= h_1a_{j+n-1} + \cdots + h_{n-1}a_{j+1} + h_na_j. \end{aligned}$$

   Consider another sequence started in a different initial state $b_1, ..., b_n$. The sequence generated is

$$\begin{aligned} b_{n+1} &= h_1b_n + \cdots + h_{n-1}b_2 + h_nb_1 \\ b_{n+j} &= h_1b_{j+n-1} + \cdots + h_{n-1}b_{j+1} + h_nb_j. \end{aligned}$$

   The mod 2 sum of these two sequences is $c_{j+n} = a_{j+n} + b_{j+n}$ and is exactly the sequence from the same shift register by starting in state $a_1 + b_1, ..., a_n + b_n$ where the addition is mod 2.

4. The (periodic) autocorrelation function of the ± 1 sequence obtained by the transformation $v_j = (-1)^{u_j}$ is two valued. That is

$$\theta_v(l) = \sum_{i=0}^{N-1} v_iv_{i+l} = \begin{cases} N, & l = 0 \bmod (N) \\ -1, & l \neq 0 \bmod (N) \end{cases}$$

   This is obvious if $l = 0$ for then $v_iv_{i+l} = 1$ so the sum is N. If $l \neq 0$ then the result is obtained by realizing that

$$v_iv_{i+l} = (-1)^{u_i}(-1)^{u_{i+l}} = (-1)^{u_i+u_{i+l}}.$$

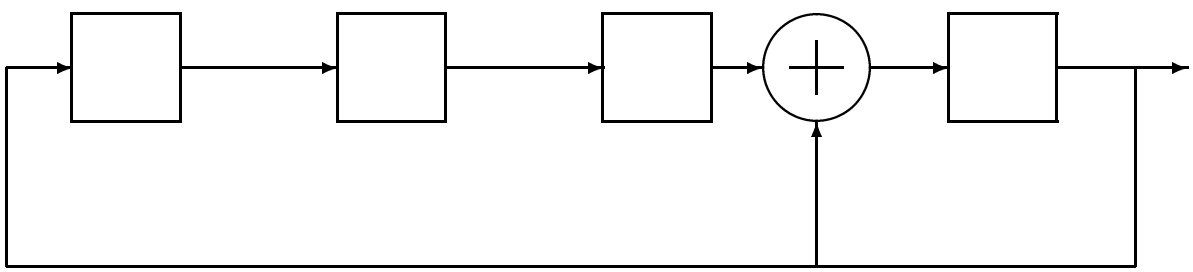
Clearly the sum on the right matters only mod 2. Since $l \neq 0$, $u_i + u_{i+l} = u_{i+k}$ for some integer k. (This is using the shift and add property) . So for $l \neq 0$

$$\theta_v(l) = \sum_{i=0}^{N-1} (-1)^{u_{i+k}} = 2^{n-1}(-1) + (2^{n-1} - 1)(1) = -1.$$

5. The number of distinct m-sequences is of length $2^n - 1$ is $\phi(2^n - 1)/n$ where $\phi(x)$ is Euler phi-function (also called totient function) $\phi(L)$ is the number of positive integers less than or equal to L that are relatively prime (greatest common divisor is 1) to L.

$2^n - 1$	Number of Sequences
15	2
31	6
63	6
255	16
1023	60

Alternative Implementation

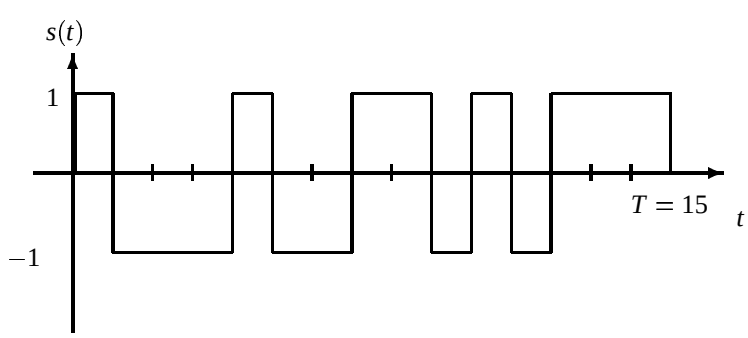


This implementation avoids doing a large number of mod 2 additions of the previous implementation.

Below we show a waveform consisting of a sequence of pulses of amplitude ± 1 determined from an m-sequence and the continuous time autocorrelation function

$$\hat{s}(\tau) = \int_{-\infty}^{\infty} s(t)s(t-\tau)dt.$$

It should be noted that the continuous time autocorrelation function defined above is the same as the discrete periodic correlation when the argument τ of the continuous correlation function is an integer multiple of the duration of a single bit. In between the autocorrelation function varies linearly.



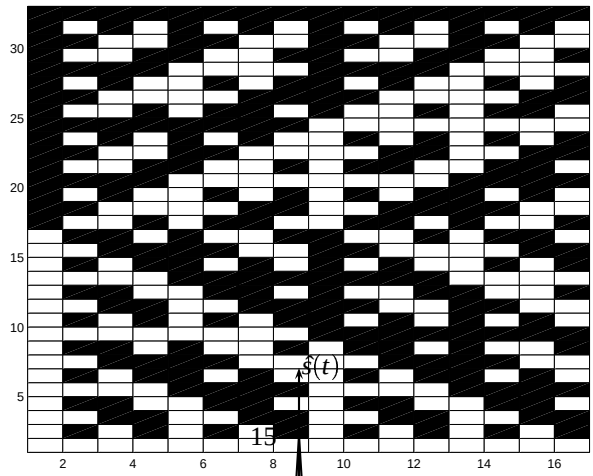
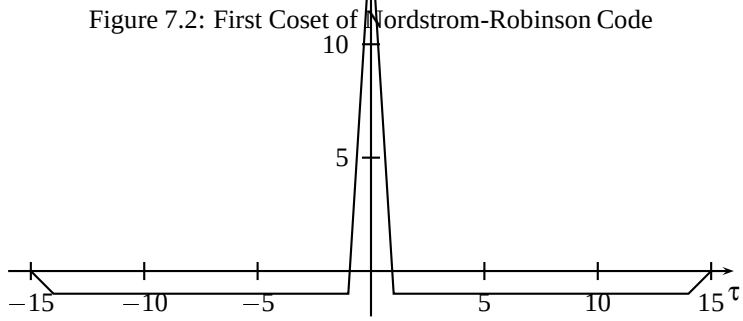


Figure 7.2: First Coset of Nordstrom-Robinson Code



7. Nordstrom-Robinson Code

The Nordstrom-Robinson code is a nonlinear code formed by starting with a biorthogonal code of length 16 (with 32 codewords) and adding 7 translates of the code. The biorthogonal code has minimum distance 8 and 32 codewords. It is remarkable that the Nordstrom Robinson code has minimum distance 6 and 256 codewords. These translates (or cosets) are shown below. The first coset is the biorthogonal coded derived from a Hadamard matrix.

The Nordstrom-Robinson code can be easily decoded with soft-decision decoding in spite of the fact that it is a nonlinear code. A simple algorithm to decode it is to take the received vector and apply a fast Hadamard transform to determine the correlation of the received vector with the 32 codewords in the first coset. Then translate the received vector using the first (or any) codeword of the first coset and reapply the fast Hadamard transform. This gives the correlations with the first coset. Continue for each coset and then choose the signal with the largest correlation as the transmitted codeword.

8. Codes for Multiamplitude signals

In this section we consider coding for multiamplitude signals. The application of this is to bandwidth constrained channels. We can consider as a baseline a two dimensional modulation system transmitting a 2400 symbols per second. If each symbol represents 4 bits of information then the data rate is 9600 bits per second. So we would like to have more signals per dimension in order to increase the data rate. However, we must try to keep the signals as far apart from each other as possible (in oder to keep the error rate low). So an increase of the size of the signal constellation for fixed minimum distance would likely increase the total signal energy transmitted.

The codes (signal sets) constructed are not linear in nature so the application of linear block codes is not very productive.

32-ary signal sets Consider a 32-ary QAM signal set shown below. The average energy is 20. The minimum distance is 2 and the rate is 5 bits/dimension. Clearly this is a nonlinear code in that the sum of two codewords is not a codeword.

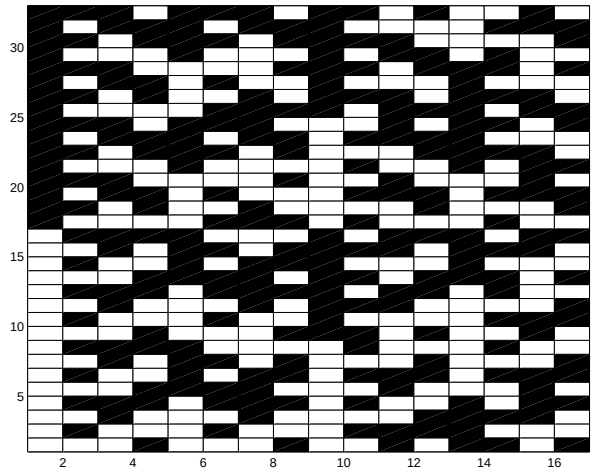


Figure 7.3: Second Coset of Nordstrom-Robinson Code

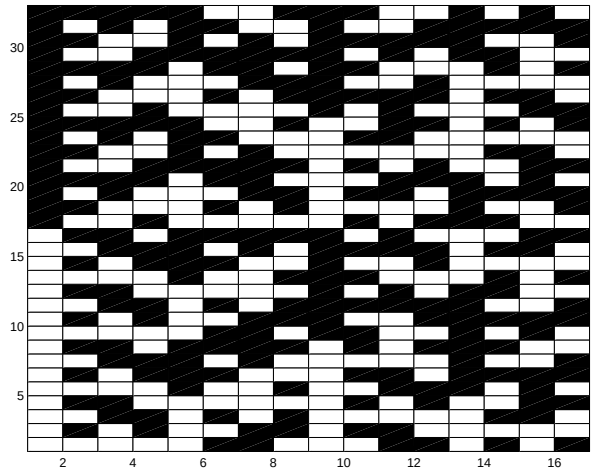


Figure 7.4: Third Coset of Nordstrom-Robinson Code

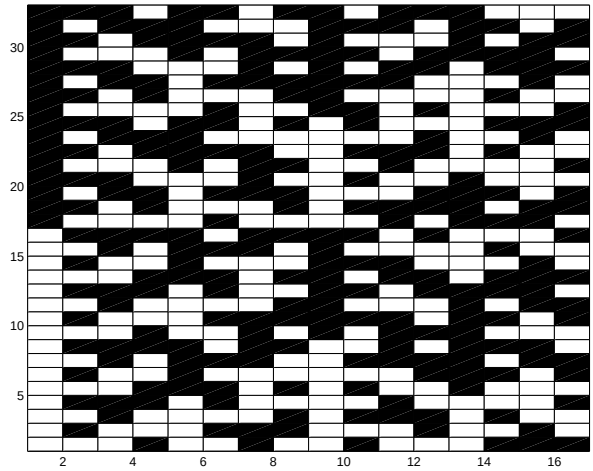


Figure 7.5: Fourth Coset of Nordstrom-Robinson Code

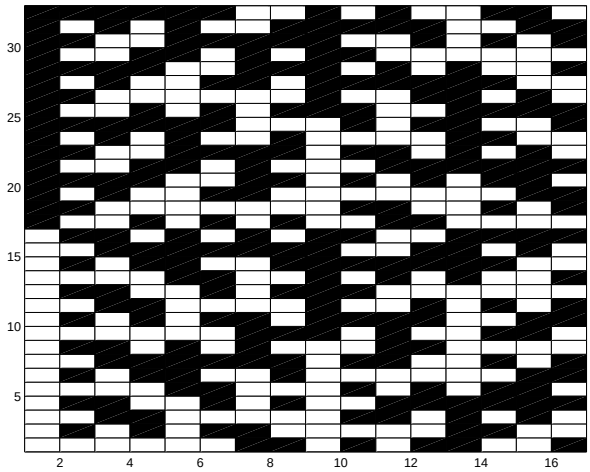


Figure 7.6: Fifth Coset of Nordstrom-Robinson Code

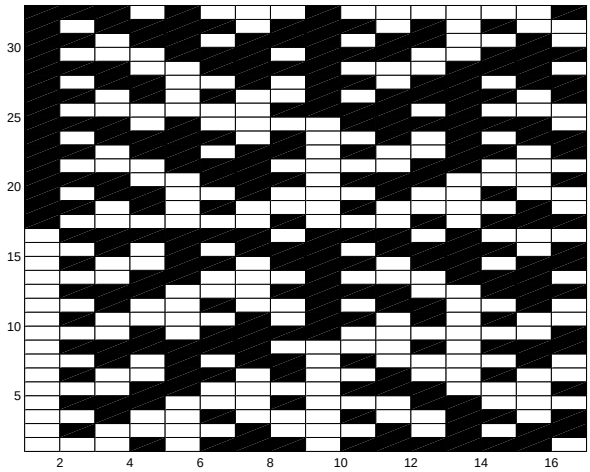


Figure 7.7: Sixth Coset of Nordstrom-Robinson Code

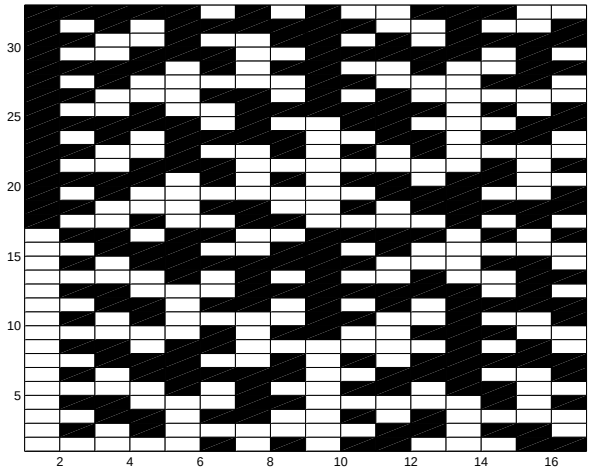


Figure 7.8: Seventh Coset of Nordstrom-Robinson Code

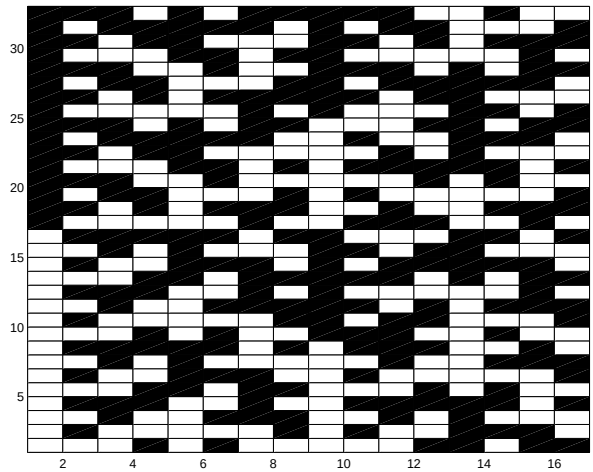
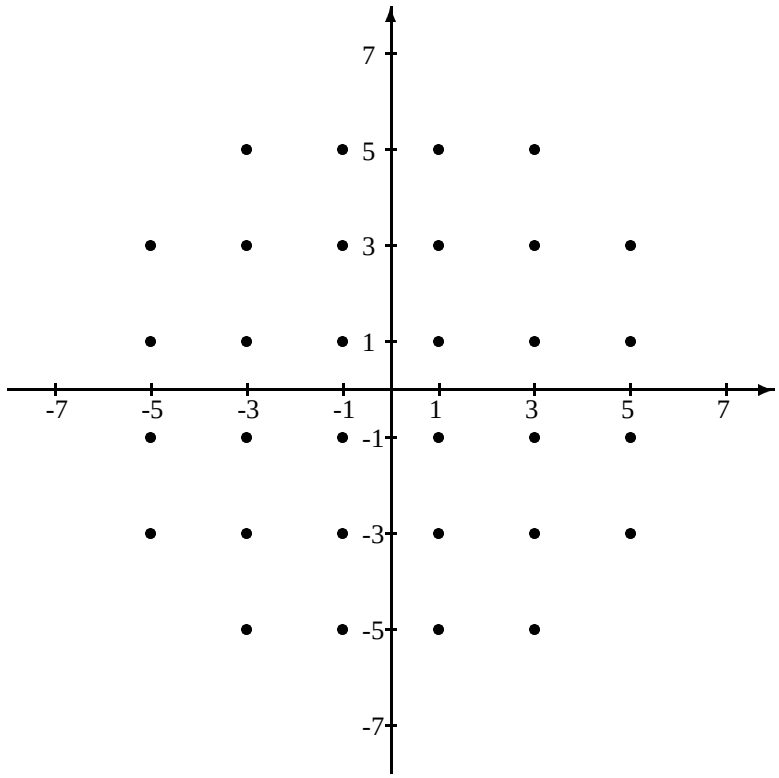
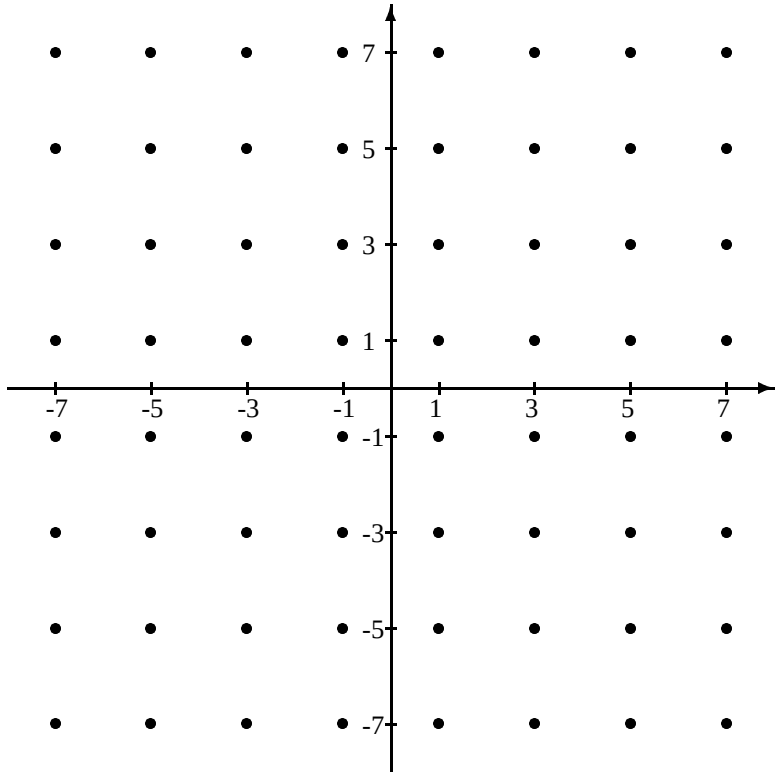


Figure 7.9: Eighth Coset of Nordstrom-Robinson Code

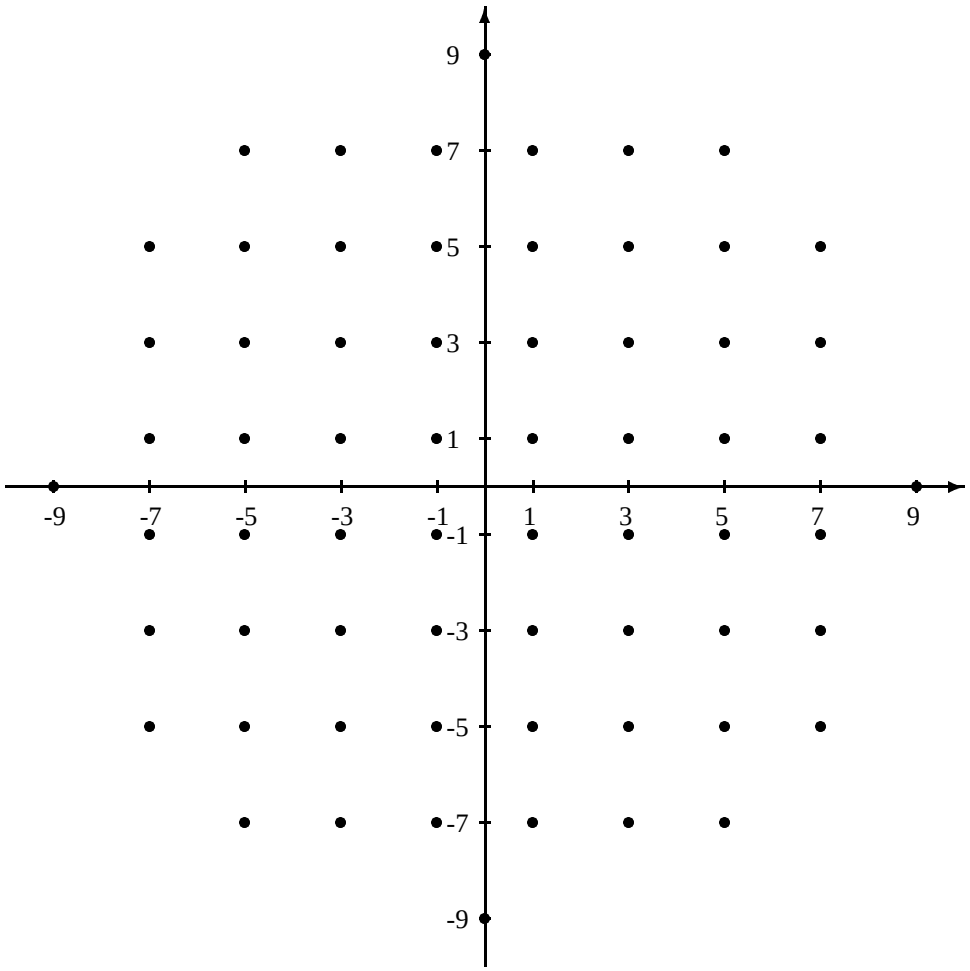


There are several possible ways to improve the performance of the constellation. First, one could add redundancy (e.g. use a binary code and make hard decisions and use the code to correct errors). However, this improved performance is at the expense of lower data rate (we must transmit the redundant bits). A second possible way of improving the performance (adding redundancy) is to increase the alphabet size of the signal set but then only allow certain subsets of all possible signal sequences. We will show that considerable gains are possible with this approach. So first we consider expanding the constellation size.

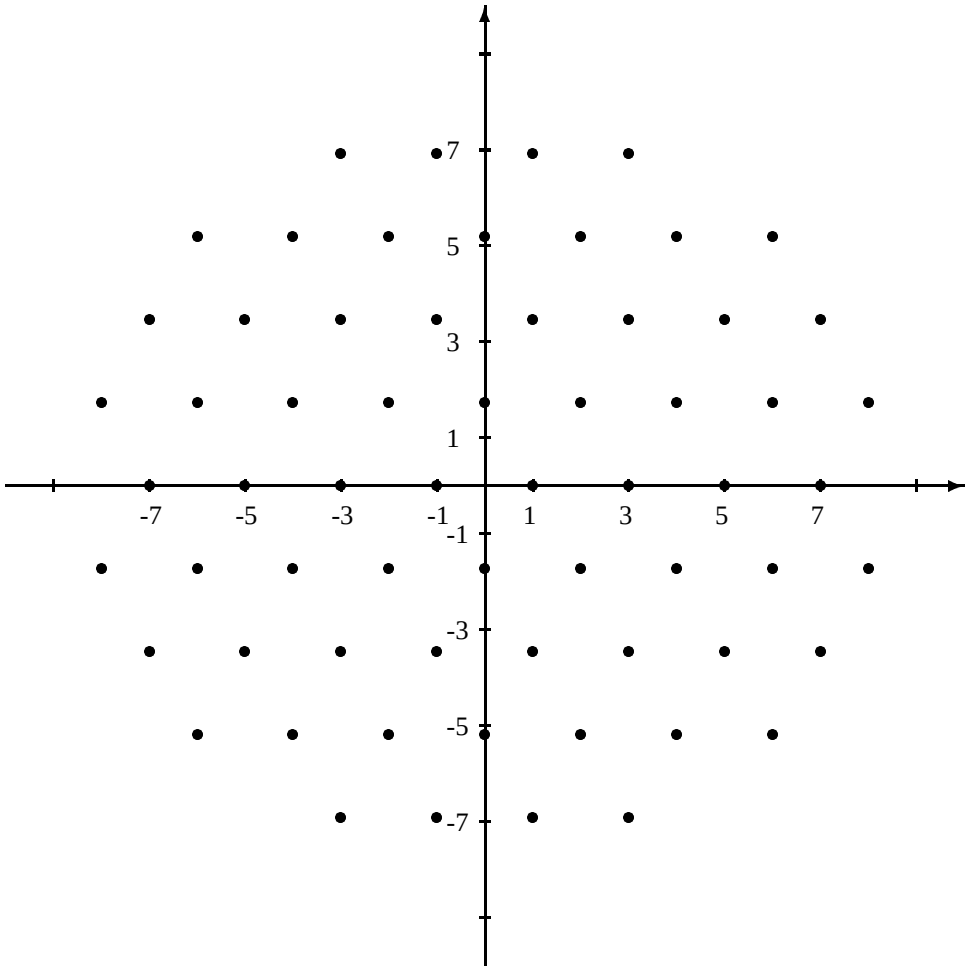
64-ary signal sets Consider a 64-ary QAM signal set shown below. The average energy is 42.


7
5
3
1
-7
-5
-3
-1
1
3
5
7
-1
-3
-5
-7

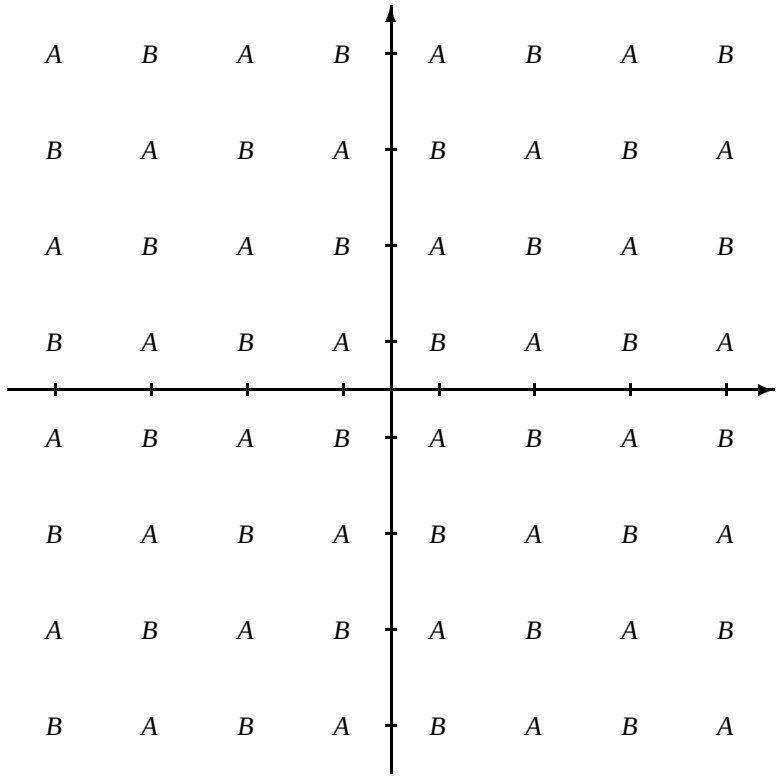
Modified QAM (used in Paradyne 14.4kbit modem). This has average energy of 40.9375.



The following hexagonal constellation has energy 35.25 but each interior point now has 6 neighbors compared to the four neighbors for the rectangular structures.



Consider now coding for 64QAM (and comparing it to an uncoded 32QAM signal set). Consider the following block code. Divide the points in the constellation into two subsets called A and B as shown below.



The code is then described by vectors of length two where it is required that the components come from the same set. Thus we can either have two signals from subset A or two signals from subset B. The Euclidean distance is calculated as follows. Consider the following two codewords.

$$\mathbf{c}_0 = (a_0, a_0)$$

and

$$\mathbf{c}_1 = (a_0, a_1)$$

where $a_0, a_1 \in A$ and $a_0 \neq a_1$. Then

$$d_E^2(\mathbf{c}_0, \mathbf{c}_1) \geq 8$$

Similar calculation holds for points in subset B. Also consider

$$\mathbf{c}_0 = (a_0, a_1)$$

and

$$\mathbf{c}_1 = (b_0, b_1)$$

where $a_i \in A$ and $b_i \in B$. Then

$$d_E^2(\mathbf{c}_0, \mathbf{c}_1) \geq 8$$

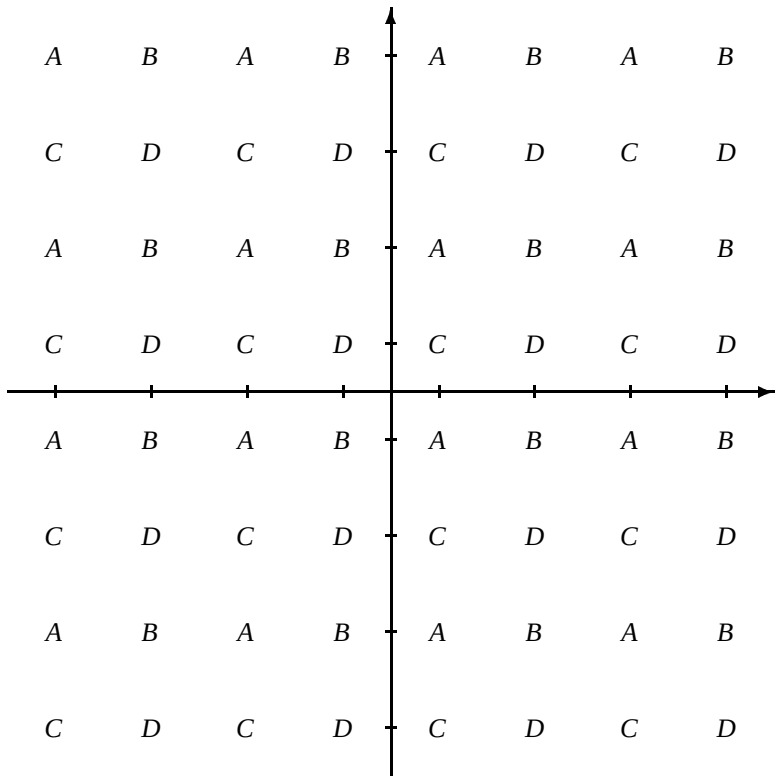
Thus the distance between two points is twice the distance of the original signal set. The original signal set transmitted 6 bits/2 dimensions or 3 bits/dimension. The new signal set transmits 11 bits/4 dimensions or 2.75 bits/dimension. The original signal set has on the average 3.4375 nearest neighbors per signal point. We calculate the number of nearest neighbors for the code as follow. Consider the nearest neighbors to the codeword (a_0, a_1) where a_0 is an interior point of the constellation and is in subset A. Then a nearest neighbor is of the form (a_2, a_1). There are four choices for a_2. This is the same as the original constellation. Now consider a_0 to be one of the exterior points (but not a corner point). Then there are only two nearest neighbors (as opposed to three for the

original constellation). Now consider a_0 to be a corner point. Corner points have only one nearest neighbor. Thus the average number of nearest neighbors is calculated to be

$$\frac{1}{64}[36 \times 4 + 24 \times 2 + 4 \times 1] = 3.0625.$$

Thus we have gained a factor of two in Euclidean distance compared to 64QAM and have reduced the average number of nearest neighbors.

Consider now further dividing the constellation.



The minimum distance between points in subset A is now 4 (or a minimum distance squared of 16). A block code for this signal partition consists of codewords of the form

$$\begin{gathered}
(A,A,A,A)\\
(A,A,D,D)\\
(A,D,A,D)\\
(A,D,D,A)\\
(D,A,A,D)\\
(D,A,D,A)\\
(D,D,A,A)\\
(D,D,D,D)\\
(C,C,C,C)\\
(C,C,B,B)\\
(C,B,C,B)\\
(C,B,B,C)\\
(B,C,C,B)\\
(B,C,B,C)
\end{gathered}$$

$$(B,B,C,C)$$
$$(B,B,B,B)$$

That is the components are either all from the sets A and D or all from the sets C and B. The number of times from any set is even. The minimum distance of this code/modulation is determined as follows. Two codewords of the form (A,A,A,A) but differing in exactly one position has squared Euclidean distance of 16. Two codewords for the form (A,A,A,A) and (A,A,D,D) have squared Euclidean distance of 8+8=16. Two codewords of the form (A,A,A,A) and (B,B,B,B) have squared Euclidean distance of 4+4+4+4=16. Thus it is easy to verify the minimum squared Euclidean distance of this code is 16 or 4 times larger than 64 QAM. The number of bits per dimension is calculated as 4 bits to determine a codeword and 4 bits to determine which point in a subset to take. Thus to chose the four subsets requires 16 bits. Thus we have 16+4=20 bits in 8 dimensions or a rate of 2.5 bits per dimension. We could compare this to a 32QAM system which also has 2.5 bits/dimension. The minimum distance squared of 32QAM is 4 and the signal power is 20 (compared to 42 for 64QAM). Thus we have increased the signal power by a factor of 2 but have increased the squared-Euclidean distance by a factor of 4. The net "coding gain" is 2 or 3dB. (Can you calculate the number of nearest neighbors?)

Thus when comparing a coded system with a certain constellation and an uncoded system with some other constellation the coding gain is defined as

$$\text{Coding Gain} = \frac{d_{E,c}^2/P_c}{d_{E,u}^2/P_u}$$

where $P_c(P_u)$ is the power (or energy) of the coded (uncoded) signal set and $d_{E,c}(d_{E,u})$ is the corresponding Euclidean distance.

9. Minimum Bit Error Probability Decoding

In this section we consider the problem of minimum bit error probability decoding for a block code. For simplicity consider a (3,2) single parity check code for which the codewords are

Data	*Bits*	*Coded*		*Bits*	*Coded*		*Bits*
d_1	d_2	d_1	d_2	q	b_1	b_2	p
0	0	0	0	0	$+1$	$+1$	$+1$
0	1	0	1	1	$+1$	-1	-1
1	0	1	0	1	-1	$+1$	-1
1	1	1	1	0	-1	-1	$+1$

Denote the data bits by d_1, d_2 and the parity bit $q = d_1 \oplus d_2$. Assume the data bits are equally likely. The coded bits from the field $\{0,1\}$ are mapped into ± 1 via $0 \to 1$ and $1 \to -1$. Denote the coded bits by b_1, b_2, p_1. Let $c = \sqrt{E}(b_1, b_2, p_1)$. For an additive white Gaussian noise channel the received signal (after demodulation) is

$$r = (r_1, r_2, r_3) = c + n$$

where the noise vector n is a sequence of three i.i.d. Gaussian distributed random variables with mean 0 and variance $N_0/2$. The decision rule that minimizes the codeword error probability based on observing r is to compute the distance between the observation r and each codeword and then determine the codeword closest in Euclidean distance to the received vector. The data bits corresponding to the closest codeword are then the data bits for which the probability of choosing the wrong *sequence* of data bits is minimum. This is sometimes referred to as maximum likelihood sequence detection.

On the other hand we may wish to minimize the probability of choosing the wrong bit for each of the two bits. To do this we need to start with the optimal bit decision rule. For the first bit, the minimum bit error probability rule is to compute the maximum a posteriori probability as follows.

$$
\begin{aligned}
\Lambda &= \log\left[\frac{P(b_1=+1|r)}{P(b_1=-1|r)}\right] \\
&= \log\left[\frac{p(r|b_1=+1)P(b_1=+1)/p(r)}{p(r|b_1=-1)P(b_1=-1)/p(r)}\right] \\
&= \log\left[\frac{p(r|b_1=+1)P(b_1=+1)}{p(r|b_1=-1)P(b_1=-1)}\right] \\
&= \log\left\{\left[\frac{P(b_1=+1)}{P(b_1=-1)}\right]\left[\frac{p(r|b_1=+1,b_2=+1)P(b_2=+1)+p(r|b_1=+1,b_2=-1)P(b_2=-1)}{p(r|b_1=-1,b_2=+1)P(b_2=+1)+p(r|b_1=-1,b_2=-1)P(b_2=-1)}\right]\right\}
\end{aligned}
$$

Now consider the calculation of these terms. First, for equally likely data bits the first part cancels. Since if b_1 and b_2 are known so is the parity p we have that

$$
\begin{aligned}
p(r|b_1=1,b_2=1) &= p(r_1,r_2,r_3|b_1=1,b_2=1,p=1) \\
&= p(r_1|b_1=+1)p(r_2|b_2=+1)p(r_3|p=1).
\end{aligned}
$$

Similarly, since if $b_1=1$ and $b_2=-1$ then $p=-1$ and thus

$$
\begin{aligned}
p(r|b_1=+1,b_2=-1) &= p(r_1,r_2,r_3|b_1=+1,b_2=-1,p=-1) \\
&= p(r_1|b_1=+1)p(r_2|b_2=-1)p(r_3|p=-1).
\end{aligned}
$$

Let

$$
L_1=\log(\frac{P(b_1=+1)}{P(b_1=-1)}),\quad L_2=\log(\frac{P(b_2=+1)}{P(b_2=-1)}).
$$

Then since $P(b_1=+1)+P(b_1=-1)=1$

$$
P(b_1=+1)=\frac{e^{L_1}}{1+e^{L_1}},\quad P(b_1=-1)=\frac{1}{1+e^{L_1}}
$$

In addition let

$$
\begin{aligned}
L(r_1) &= \log(\frac{p(r_1|b_1=+1)}{p(r_1|b_1=-1)}=\log\left(\frac{\frac{1}{\sqrt{\pi N_0}}\exp\{-(r_1-\sqrt{E})^2/N_0\}}{\frac{1}{\sqrt{\pi N_0}}\exp\{-(r_1+\sqrt{E})^2/N_0\}}\right)=4r_1\sqrt{E}/N_0 \\
L(r_2) &= \log(\frac{p(r_2|b_2=+1)}{p(r_2|b_2=-1)}=4r_2\sqrt{E}/N_0 \\
L(r_3) &= \log(\frac{p(r_3|p=+1)}{p(r_3|p=-1)}=4r_3\sqrt{E}/N_0.
\end{aligned}
$$

Then we can write the log of the aposteriori probability ratio as

$$
\begin{aligned}
\Lambda &= \log\left[\frac{P(b_1=+1|r)}{P(b_1=-1|r)}\right] \\
&= \log\left\{\left[\frac{P(b_1=+1)}{P(b_1=-1)}\right]\right. \\
&\quad \left.+\log\left[\frac{p(r_1|b_1=+1)p(r_2|b_2=+1)p(r_3|p=+1)P(b_2=+1)+p(r_1|b_1=+1)p(r_2|b_2=-1)p(r_3|p=-1)P(b_2=-1)}{p(r_1|b_1=-1)p(r_2|b_2=+1)p(r_3|p=-1)P(b_2=+1)+p(r_1|b_1=-1)p(r_2|b_2=-1)p(r_3|p=+1)P(b_2=-1)}\right]\right\}
\end{aligned}
$$

$$
\begin{aligned}
&= L_1 + L(r_1) + \log\left[\frac{p(r_2|b_2=+1)p(r_3|p=+1)P(b_2=+1)+p(r_2|b_2=-1)p(r_3|p=-1)P(b_2=-1)}{p(r_2|b_2=+1)p(r_3|p=-1)P(b_2=+1)+p(r_2|b_2=-1)p(r_3|p=+1)P(b_2=-1)}\right] \\
&= L_1 + L(r_1) + \log\left[\frac{\frac{p(r_2|b_2=+1)}{p(r_2|b_2=-1)}\frac{p(r_3|p=+1)}{p(r_3|p=-1)}\frac{P(b_2=+1)}{P(b_2=-1)}+1}{\frac{p(r_2|b_2=+1)}{p(r_2|b_2=-1)}\frac{P(b_2=+1)}{P(b_2=-1)}+\frac{p(r_3|p=+1)}{p(r_3|p=-1)}}\right] \\
&= L_1 + L(r_1) + \log\left(\frac{\left[e^{L(r_2)}e^{L_2}e^{L(r_3)}+1\right]}{\left[e^{L(r_2)}e^{L_2}+e^{L(r_3)}\right]}\right)
\end{aligned}
$$

The first term above is the apriori information about the data bit of interest. The second term is the information about the data bit based solely on the first received sample. The last term is the information about the first bit obtained from the observations of the second bit and the parity. Thus the optimal decision rule is to minimize the probability of error for the first bit is to compute Λ and compare Λ with 0. If it is larger the decision is that $b_1 = +1$ while if it is less than zero the decision is that $b_1 = -1$.

For equal probable data bits $L_1 = L_2 = 0$ and the log likelihood ratio become

$$
\Lambda = L(r_1) + \log\left(\frac{\left[e^{L(r_2)+L(r_3)}+1\right]}{\left[e^{L(r_2)}+e^{L(r_3)}\right]}\right)
$$

Consider now a more complicated code, namely a product code that has 4 information bits and 4 parity bits as shown in the array below.

b_1	b_2	p_1
b_3	b_4	p_2
p_3	p_4	

The observation consists of the the transmitted data in additive white Gaussian noise.

$$
\begin{aligned}
r_1 &= \sqrt{E}b_1 + n_1 \\
r_2 &= \sqrt{E}b_2 + n_2 \\
r_3 &= \sqrt{E}b_3 + n_3 \\
r_4 &= \sqrt{E}b_4 + n_4 \\
r_5 &= \sqrt{E}p_1 + n_5 \\
r_6 &= \sqrt{E}p_2 + n_6 \\
r_7 &= \sqrt{E}p_3 + n_7 \\
r_8 &= \sqrt{E}p_4 + n_8
\end{aligned}
$$

Based on the above it is straight forward to generalize the results above for this new code. Instead of doing this we describe a suboptimal method for decoding. The algorithm is iterative.

- Step1: Begin with $L_{V,1} = L_{V,2} = L_{V,3} = L_{V,4} = 0$.
- Step 2: Compute estimates for the likelihoods for b_1, b_2, b_3, b_4 as follows:

$$
L_{H,1} = \log\left(\frac{\exp\{L(r_2)+L_{V,2}+L(r_5)\}+1}{\exp\{L(r_2)+L_{V,2}\}+\exp\{L(r_5)\}}\right)
$$

$$
L_{H,2} = \log\left(\frac{\exp\{L(r_1)+L_{V,1}+L(r_5)\}+1}{\exp\{L(r_1)+L_{V,1}\}+\exp\{L(r_5)\}}\right)
$$

$$L_{H,3} = \log\left(\frac{\exp\{L(r_4)+L_{V,4}+L(r_6)\}+1}{\exp\{L(r_4)+L_{V,4}\}+\exp\{L(r_6)\}}\right)$$

$$L_{H,4} = \log\left(\frac{\exp\{L(r_3)+L_{V,3}+L(r_6)\}+1}{\exp\{L(r_3)+L_{V,3}\}+\exp\{L(r_6)\}}\right)$$

- Step 3: Using the likelihood computed in step 2, compute

$$L_{V,1} = \log\left(\frac{\exp\{L(r_3)+L_{H,3}+L(r_7)\}+1}{\exp\{L(r_3)+L_{H,3}\}+\exp\{L(r_7)\}}\right)$$

$$L_{V,2} = \log\left(\frac{\exp\{L(r_4)+L_{H,4}+L(r_8)\}+1}{\exp\{L(r_4)+L_{H,4}\}+\exp\{L(r_8)\}}\right)$$

$$L_{V,3} = \log\left(\frac{\exp\{L(r_1)+L_{H,1}+L(r_7)\}+1}{\exp\{L(r_1)+L_{H,1}\}+\exp\{L(r_7)\}}\right)$$

$$L_{V,4} = \log\left(\frac{\exp\{L(r_2)+L_{H,2}+L(r_8)\}+1}{\exp\{L(r_2)+L_{H,2}\}+\exp\{L(r_8)\}}\right)$$

- Step 4: Repeat steps 2 and 3 for some number of times.

- Step 5. Compute decisions for the bits based on (assuming equally likely apriori probabilities)

$$L(r_1)+L_{H,1}+L_{V,1}$$

For this code we can bound the bit error probability with maximum likelihood decoding if we know the weights of all the codewords.

Number of Codewords	Weight of Codeword	Number of nonzero Information bits
1	0	0
4	3	1
4	4	2
1	4	4
4	5	3
2	6	2

$$P_{e,b} \leq Q(\sqrt{\frac{2E_b*1.5}{N_0}})+3Q(\sqrt{\frac{2E_b*2}{N_0}})+3Q(\sqrt{\frac{2E_b*2.5}{N_0}})+Q(\sqrt{\frac{2E_b*3}{N_0}})$$

In this figure the dots are the bit error probability for maximum likelihood decoding (minimum codeword error probbility), the red line is the upper bound on bit error probability, the blue line is the error probability with one iteration of the iterative decoding while the black line coresponds to four iterations.

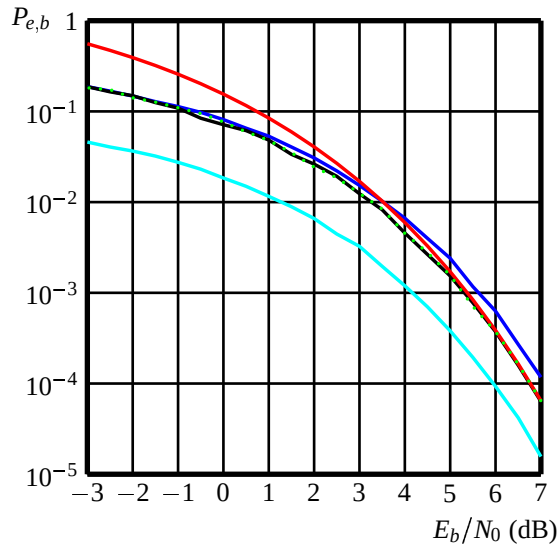


Figure 7.10: Bit error probability for a product code with iterative decoding (1 iteration and 4 iterations) compared to ML decoding and the upper bound on bit error probability for ML decoding

10. Problems

1. Consider the following code (signal set) of length 4 for a binary symmetric channel (i.e. $A = B = \{0,1\}$) with crossover probability p : $\mathbf{x}_0 = 0000$, $\mathbf{x}_1 = 0011$, $\mathbf{x}_2 = 1100$, $\mathbf{x}_3 = 1111$. Suppose the codewords (signals) are transmitted with unequal probabilities,

$$\begin{aligned} P\{\mathbf{x}_0\} &= 1/2 \\ P\{\mathbf{x}_1\} &= 1/8 \\ P\{\mathbf{x}_2\} &= 1/8 \\ P\{\mathbf{x}_3\} &= 1/4 \end{aligned}$$

   Find a decoding rule that minimizes the average error probability.

2. Consider the binary input AWGN channel with 'hard decisions' on the output of the channel. That is a decision of 0 is made if the output is greater than zero and a decision of 1 is made if the output is less than zero. This channel is shown below. The noise is a zero mean Gaussian random variable with variance $N_0/2$. Make a sketch of the minimum required E_b/N_0 for error probability arbitrarily small vs. code rate (in bits/channel use or bits/dimension). The code rate is between 0 and 1. What is the loss in E_b/N_0 in making a hard decision at low rates? (Hint: Consult the figures handed out in class and a table of the Q function. The capacity of a binary symmetric channel with crossover probability p is $1 - H_2(p)$. See also problem 5.15 of text.

   (b) Plot the point corresponding to no coding and error probability 10^{-5}.

   (c) Plot the point corresponding to a repetition code of length 3 and error probability 10^{-5}. Be careful to take into account the rate of the code in your calculation of E_b/N_0.

3. In this problem we will compare the E_b/N_0 for fixed error probability of long repetition codes with BPSK modulation with <u>soft</u> and <u>hard</u> decision decoding. You will need to use the central limit theorem (a version of this is stated below).

   Central Limit Theorem:

Let X_k be independent random variables with common distribution $F(x)$ such that

$$E[X_k] = 0 \quad E[X_k^2] = \sigma^2 > 0 \quad E[|X_k|^3] = \rho < \infty$$

and let F_n be the distribution of the normalized sum

$$(X_1 + X_2 + \ldots + X_n)/\sigma\sqrt{n}$$

Then for all x and n

$$|F_n(x) - \Phi(x)| \leq \frac{3\rho}{\sigma^3\sqrt{n}}$$

(Feller, An Introduction to Probability Theory and Its Applications, vol. 2, pp. 542).

The soft decision decoder makes a decision based on the the *sum* of the n outputs of the BPSK demodulator (without binary quantizer). Compute the error probability for the soft decision decoder. Your answer should be in terms of the Q function and E_b/N_0.

The hard decision decoder first makes a hard decision on each on the n demodulator outputs and then makes a optimum decision (minimum Hamming distance) as to which codeword was transmitted based on the quantized outputs. Use the above theorem to get an approximation to the error probability (for large n) of the hard decision decoder. Your answer should again be in terms of the Q function and E_b/N_0. What is the loss in signal-to-noise ratio in making a hard decision for repetition codes on the binary AWGN? (Hint: For small x, $1 - 2Q(x) \approx \frac{2x}{\sqrt{2\pi}}$).

4. (a) For the cyclic Hamming code with generator matrix

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

when used on the binary AWGN channel ($0 \to +\sqrt{E}$, $1 \to -\sqrt{E}$) decode the following received vector using soft decision (minimum error probability) decoding.

$$r = (-0.9,\ -0.8,\ +0.1,\ -0.1,\ -0.6,\ +0.7,\ +0.6).$$

If the same code is used on the hard decision channel and the received vector is a binary quantized version of r find the most likely transmitted codeword.

(b) Find the Union bound on the error probability for Hamming codes with soft decision and hard decision decoding.

(c) Find an exact expression for the probability of decoding error for the hard decision decoder.

(d) Find an exact expression for the average probability of an information bit being in error for the Hamming code (with hard decisions).

(e) Make a plot of the answers to (b)-(d) vs E_b/N_0 for E_b/N_0 (dB) between 0 and 15 dB.

5. (a) Find the Union bound to the codeword error probability for linear codes when used on the binary AWGN channel with soft decisions. Express your answer in terms of the Q function, E_b/N_0 and A_i where A_i is the number of codewords with Hamming weight i.

(b) Further upper bound the error probability by using the bounds

$$Q(x) \leq \frac{1}{2}e^{-x^2/2} \quad x \geq y \text{ implies } e^{-x} \leq e^{-y}$$

to obtain a bound of the form

$$P_e < K \exp\{-Q_a E_b/N_0\}$$

where Q_a, called the quality of the code, is a function only of the minimum distance and rate of the code. (Alternatively use the Union-Bhattacharyya bound).

(c) Compute the quality parameter of the (23,12) Golay code, the (n,1) repetition codes, and the (15,11) Hamming code. Note: This bound is tight (close to the actual answer) only for large values of E_b/N_0.

6. Consider the cyclic code of length 15 with generator polynomial

$$g(x) = x^{13} + x^{12} + x^{10} + x^9 + x^7 + x^6 + x^4 + x^3 + x + 1.$$

(a) List the codewords of the code.

(b) Find a generator matrix for the code.

(c) Find a parity check matrix for the code.

(d) Decode the following received vector using minimum (Hamming) distance decoding.

$$r = (1,\ 0,\ 1,\ 1,\ 1,\ 1,\ 0,\ 0,\ 1,\ 0,\ 0,\ 1,\ 0,\ 0,\ 1)$$

7. Let C_1 be an $(n_1,\ k_1)$ binary linear code with minimum distance d_1 and generator matrix G_1. Let C_2 be an $(n_2,\ k_2)$ binary linear code with minimum distance d_2 and generator matrix G_2. Consider the code with generator matrix G given below.

$$G = \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix}$$

What is the length, dimension, and minimum distance of this new code? Justify your answers.

8. The following code is used in conjunction with Binary Frequency Shift Keying.

$$C = \{(0,0,0,0),(0,0,1,1),(1,1,0,0),(1,1,1,1)\}.$$

That is, a 0 is sent using one frequency and a 1 is sent using another frequency. Assume that the two frequencies are orthogonal (over a single bit duration). Assume that the phase of the received signal is independent from bit to bit. Let $y_{0,l}$, $l = 1,2,3,4$ and $y_{1,l}$, $l = 1,2,3,4$ be the outputs of the noncoherent matched filters for the two frequencies.

(a) If noncoherent demodulation is used determine the optimum receiver/decoder. That is, how should the receiver process the 8 variable to optimally determine which codeword was sent.

(b) Determine bounds on the error probability with the optimum receiver and a suboptimum square-law combining type of receiver.

(c) Determine he receiver that minimizes the probability of error for the first information bit.

9. (a) Consider a binary error correcting code on length n that can correct e errors. By considering the volume of the nonoverlapping decoding regions find a bound on how many codewords there could possibly be.

(b) Consider the linear code of length 15 with 11 information bits and 4 parity bits.

How many codewords are there?

The generator matrix is

$$G = \begin{bmatrix} 100000000001111 \\ 010000000001110 \\ 001000000001101 \\ 000100000001100 \\ 000010000001011 \\ 000001000001010 \\ 000000100001001 \\ 000000010000111 \\ 000000001000110 \\ 000000000100101 \\ 000000000010011 \end{bmatrix}$$

The parity check matrix is given below.

$$H = \begin{bmatrix} 111111100001000 \\ 111100011100100 \\ 110011011010010 \\ 101010110110001 \end{bmatrix}$$

The received vector is $r = (011111000000000)$. Determine the most likely information bits transmitted. Is it ever possible to correct more than one error with this code?

10. Consider a (40,16) product code with 16 information bits arranged in a 4 by 4 array. Each row and column are protected by a (7,4) Hamming code with generator matrix

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

and parity check matrix

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

The overall product code is a (40,16) code. Derive an iterative algorithm for decoding the product code when used on an AWGN channel. Simulate the performance with 1,2, and 4 iterations. Plot the bit error probability versus signal-to-noise ratio for signal-to-noise ratios (E_b/N_0) between -3dB and 5dB. Remember that in this case $E_b = 40E/16$ where E is the energy used for each coded bit. Compare your results to the union bound for maximum likelihood decoding.