Linear Block Codes

Definition: A code is said to be linear if the sum of any two codewords is also a codeword.

The minimum distance of a linear code is much easier to compute than the minimum distance of an arbitrary code.

Proposition: The minimum Hamming distance of a linear code is the minimum Hamming weight of any nonzero codeword.

$$\begin{aligned} d_{\min} &= \min_{\mathbf{c}_1 \neq \mathbf{c}_2} d_H(\mathbf{c}_1, \mathbf{c}_2) \\ &= \min_{\mathbf{c}_1 \neq \mathbf{c}_2} w_H(\mathbf{c}_1 - \mathbf{c}_2) \\ &= \min_{\mathbf{c} \neq 0} w_H(\mathbf{c}) \end{aligned}$$

Linear Block Codes

Proposition: For a linear code the number of codewords, A_d, distance d from codeword c_i does not depend on the codeword c_i. We say the code is geometrically uniform.
As a result the error probability given codeword c_i is transmitted does not depend on which codeword the error probability is conditioned on.

Proposition: For a linear code the number of codewords, A_d, distance d from codeword c_i does not depend on the codeword c_i. We say the code is geometrically uniform.

Example 1: Consider the code

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

- From codeword $(0,0,0,0)$ there are two codewords distance 2, and one codeword distance 4.
- From codeword $(1,1,0,0)$ there are two codewords distance 2, and one codeword distance 4.

As a result the error probability given codeword c_i is transmitted does not depend on which codeword the error probability is conditioned on.

Wayne E Stark: It doesn't matter where you start for a linear code all the code words the distance structure is the same, and sometimes we call this.

Vector spaces

- Consider the set of binary vectors of length n with components 0 and 1.
  - For example vectors of length 4 include $\{(0,0,0,0),(0,0,0,1),(0,0,1,0),...,(1,1,1,1)\}$.
- Define addition of vectors as component wise mod 2 addition.

$$\begin{array}{r} (0,1,0,1) \\ +(1,0,0,1) \\ \hline (1,1,0,0) \end{array}$$

- Scalar multiplication is multiplication by 0 or 1.

$$0 \times (1,0,1,1) = (0,0,0,0), \quad 1 \times (1,0,1,1) = (1,0,1,1)$$

Vector spaces

- A vector space is a set of vectors and two operations (addition and multiplication) which is
  - closed under addition,
  - each vector has an additive inverse and
  - satisfies the associative, commutative and distributed properties.
- The set of binary vectors of length n with addition done component-wise mod 2 addition and multiplication done in a scalar sense satisfies these properties.

Linear code as subspace

- A subspace of a vector space is a subset of the vector space that also satisfies the properties of a vector space.
- In particular it is closed under addition (the addition of any two elements in the subspace is also an element in the subspace).
- A subspace C of a vector space V can be "generated" from a set of basis vectors by forming all linear combinations of the basis vectors.

Linear code as subspace

- A linear code can be generated by a linear combination of $k = \log_2 M$ basis codewords.
- A generator matrix, G, contains these k basis codewords.
- The particular linear combination used depends on the information bits to be transmitted. Because there are k basis vectors there are also k information bits.
- An (n, k) linear code is a mapping from k information bits to one of 2^k codewords of length n.

Example: (6, 3) Code

$M = 8$, $n = 6$, $k = 3$. $= \#$ of basis vectors

$$\mathcal{C} = \{(000000), (100101), (010111), (001011), (110010), (101110), (011100), (111001)\}$$

Generator Matrix

$$G = \begin{matrix} m_1 \\ m_2 \\ m_3 \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Here is how a codeword is generated. Assume the message is $m = 101$. Then

$$c = mG = [101] \underset{3\times 6}{\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}} = [101110]$$

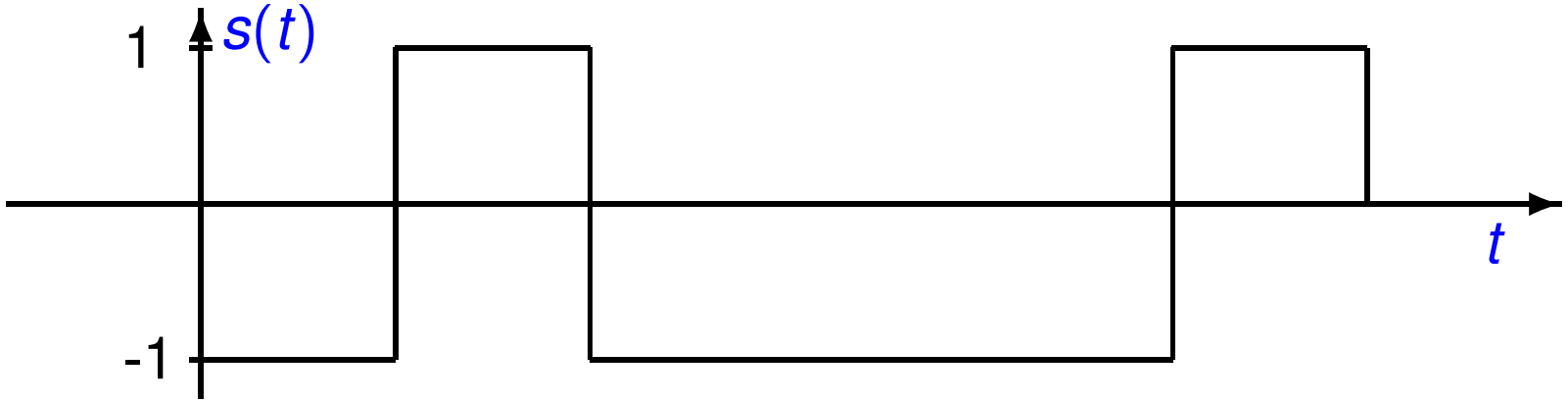
Thus the encoding is a simple matrix multiplication.

Coding and Modulation

If this codeword was to be transmitted we would convert the 0/1 vector into a +1/-1 vector as follows

$$(1,0,1,1,1,0) \rightarrow (-1,+1,-1,-1,-1,+1)$$

which for (baseband) signalling would be a waveform like shown below.



Parity Checks

- For a linear code there is a matrix called the parity check matrix H.
- All codewords $\mathbf{c}$ must satisfy the parity check equations

$$H\mathbf{c}^T = 0^T = \begin{pmatrix} 0 \\ 0 \\ . \\ . \\ 0 \end{pmatrix}$$

- The parity check matrix can be obtained from the generator matrix.
- From the generator matrix a set of equations relating the code symbols to the information symbols is straight forward.

Parity Checks

- For a systematic code the codeword contains k positions that are the information bits. The remaining positions are parity checks. There are $n-k$ parity check equations.
- The parity check matrix is a $(n-k) \times n$ matrix.
- Since $Hc^T = 0^T$, the minimum number of columns of H that can be summed to 0 is also the minimum weight of the code.

Parity Check for the (6, 3) Code

For the example above we can write the equations

$$
\begin{aligned}
c_1 &= m_1 \\
c_2 &= m_2 \\
c_3 &= m_3 \\
c_4 &= m_1 + m_2 = c_1 + c_2 \\
c_5 &= m_2 + m_3 = c_2 + c_3 \\
c_6 &= m_1 + m_2 + m_3 = c_1 + c_2 + c_3
\end{aligned}
$$

$c_1 + c_2 + c_4 = 0$

$c_1 + c_2 - c_4 = 0$

$c_2 + c_3 + c_5 = 0$

$c_1 + c_2 + c_3 + c_6 = 0$

Rewriting the last three equations we obtain

$$
\begin{aligned}
c_1 + c_2 + c_4 &= 0 \\
c_2 + c_3 + c_5 &= 0 \\
c_1 + c_2 + c_3 + c_6 &= 0
\end{aligned}
$$

Parity Check for the (6, 3) Code

$K = \log_2 m$

These can be written in matrix form as

$$Hc^T = 0$$

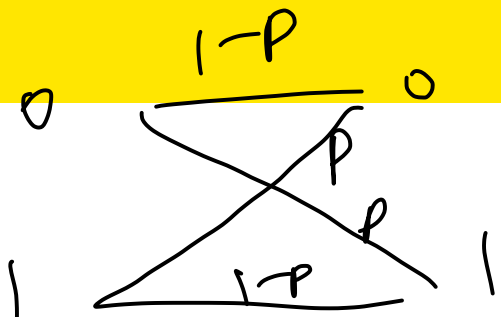
The matrix H is known as the parity check matrix. For the above code the parity check matrix is

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad 3 \times 6$$

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad 6 \times 1 \qquad = \begin{bmatrix} 1+1 \\ 0 \\ 1+1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad 3 \times 1$$

For example the codeword (100101) satisfies this.

Channel



- If the codeword is transmitted over a binary symmetric channel the received vector can be represented as

$$\mathbf{r} = \mathbf{c} + \mathbf{e}$$

where $\mathbf{e}$ is an error vector and the addition is done mod two.

- For example suppose the codeword (1,0,1,1,1,0) was transmitted over the channel and the noise in the channel causes the third bit to be in error.
- Then $\mathbf{e} = (0, 0, 1, 0, 0, 0)$.

Decoding

$M = 2^K$ = # of codewords

e ?

- Decoding is done by noticing that

$$\mathbf{s}^T = H\mathbf{r}^T = H(\mathbf{c}+\mathbf{e})^T = H\mathbf{c}^T + H\mathbf{e}^T = H\mathbf{e}^T$$

- Because there are more unknowns in the above equation (n unknowns and $n-k$ equations) we can not uniquely determine $\mathbf{e}$ from this.
- In fact there are 2^k different vectors $\mathbf{e}$ which satisfy the above equation.
- We would like to find the most likely such vector.

$He^T = s^T \Rightarrow e' = e + c$ where c is a codeword is also a sdution to $He'^T = s^T$

Example of Decoding

- Let $\mathbf{r} = (011001)$.
- Compute $\mathbf{s}^T = H\mathbf{r}^T$

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

- The error patterns that satisfy this equation are

$$(100000), (000101), (101011), (010010)$$
$$(001110), (011001), (110111), (111100)$$

Example of Decoding

$\hat{c} = r + e = 111001$ codeword

- The most likely error pattern is the one with the fewest errors (assuming a BSC with crossover probability less than 1/2).
- Thus the most likely error pattern is $\mathbf{e} = (100000)$.
- The vector $\mathbf{s}$ is called the syndrome.
- The most likely error vector that yields a given syndrome is called the coset leader.
- 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.

Standard Array

- The standard array is a table containing all possible error vectors.
- Each row of the array corresponds to an error vector that yields the same syndrome.
- The first column of each row is the error vector, called the coset leader, that has the smallest weight.
- In some cases there may be two error patterns with the same (smallest) weight.
- In this case either one could be choosen as the coset leader.

Standard Array Decoding

- Using the standard array, decoding is done as follows
  - First find the synrdome $s^T = Hr^T$.
  - Second, find the corresponding coset leader for that syndrome.
  - Third, subtract the coset leader from the received vector to determine the most likely transmitted signal.

Standard Array for Decoding (6, 3) code

S	Coset Leader							
000	000000	100101	010111	110010	001011	101110	011100	111001
101	100000	000101	110111	010010	101011	001110	111100	011001
111	010000	110101	000111	100010	011011	111110	001100	101001
011	001000	101101	011111	111010	000011	100110	010100	110001
100	000100	100001	010011	110110	001111	101010	011000	111101
010	000010	100111	010101	110000	001001	101100	011110	111011
001	000001	100100	010110	110011	001010	101111	011101	111000
110	101000	001101	111111	011010	100011	000110	110100	010001

S = Syndrome

2^{n-k}

$n = 2^k$

3 way tie

ask for retransmission

Standard Array

- Note that the first row in the standard array (with syndrome 0^T) is just the set of codewords (vectors satisfying $Hr^T = 0^T$).
- The other rows are called cosets and are obtained by adding a vector (not a codeword or a vector previously found) to the first row.
  - While the first row (the codewords) are a linear subspace, the cosets are not a linear subspace but a translation of a linear subspace.
- The coset leader is the vector in a coset with the smallest weight.
- The columns (except for the first column corresponding to the syndrome) are the decoding regions of each codeword that is at the top of the column.

Standard Array

of information bits

- The generator matrix G for a linear code has k rows of length n.
- The parity check matrix H for a linear code has $n-k$ rows of length n.
- A code is said to be systematic if the first k columns of the generator matrix is an identity matrix. In this case the last $n-k$ columns of the parity check matrix forms an identity matrix.
- A systematic code has the information bits (symbols) "in the clear" in each codeword. That is, in each codeword there are k positions that are the information bits.
- The complexity of decoding using the standard array is 2^{n-k} because you need to store the most likely error pattern for each possible syndrome and there are 2^{n-k} possible syndromes.

If G is of the form

$$G = [I_{k\times k} \mid A]$$

where A is a $k \times (n-k)$ array then

$$H = \left[A^T \mid I_{(n-k)\times(n-k)}\right].$$

Example 5: $n = 15, k = 5, n-k = 10$.

$$G = \left[\begin{array}{ccccc|cccccccccc} 1&0&0&0&0&1&1&1&0&1&1&0&0&1&0\\ 0&1&0&0&0&0&1&1&1&0&1&1&0&0&1\\ 0&0&1&0&0&1&1&0&1&0&1&1&1&1&0\\ 0&0&0&1&0&0&1&1&0&1&0&1&1&1&1\\ 0&0&0&0&1&1&1&0&1&1&0&0&1&0&1 \end{array}\right]$$

$$H = \left[\begin{array}{ccccc|cccccccccc} 1&0&1&0&1&1&0&0&0&0&0&0&0&0&0\\ 1&1&1&1&1&0&1&0&0&0&0&0&0&0&0\\ 1&1&0&1&0&0&0&1&0&0&0&0&0&0&0\\ 0&1&1&0&1&0&0&0&1&0&0&0&0&0&0\\ 1&0&0&1&1&0&0&0&0&1&0&0&0&0&0\\ 1&1&1&0&0&0&0&0&0&0&1&0&0&0&0\\ 0&1&1&1&0&0&0&0&0&0&0&1&0&0&0\\ 0&0&1&1&1&0&0&0&0&0&0&0&1&0&0\\ 1&0&1&1&0&0&0&0&0&0&0&0&0&1&0\\ 0&1&0&1&1&0&0&0&0&0&0&0&0&0&1 \end{array}\right].$$

Short Table of Codes

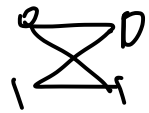
n	k	d_{min}	t	r	Code Type
7	4	3	1	0.571	Hamming (BCH)
8	4	4	1	0.500	Extended Hamming
15	11	3	1	0.733	Hamming (BCH)
15	7	5	2	0.467	BCH
15	5	7	3	0.333	BCH
15	1	15	7	0.067	Repetition (BCH)
23	12	7	3	0.478	Golay
24	12	8	3	0.500	Extended Golay

Hamming code →

t → e

Decode

Hard decision decoding



Hamming code

$H = [h_0, h_1, \cdots h_{n-1}]$

all possible columns of size $n-k$ except all 0

Short Table of Codes (cont.): BCH Codes

Hamming code →

n	k	d_{min}	t	r
31	26	3	1	0.839
31	21	5	2	0.677
31	16	7	3	0.516
31	11	11	5	0.355
31	6	15	7	0.194
31	1	31	15	0.032

repetition code

Hamming code

all possible columns of size n-k except all zeros

$H = [h_0, h_1, \cdots h_{n-1}]$

Short Table of Codes (cont.): BCH Codes

n	k	d_{min}	t	r
63	57	3	1	0.905
63	51	5	2	0.810
63	45	7	3	0.714
63	39	9	4	0.6190
63	36	11	5	0.5714
63	30	13	6	0.4762
63	24	15	7	0.3810
63	18	21	10	0.2857
63	16	23	11	0.2540
63	10	27	13	0.1587
63	7	31	15	0.1111
63	1	63	31	0.0159

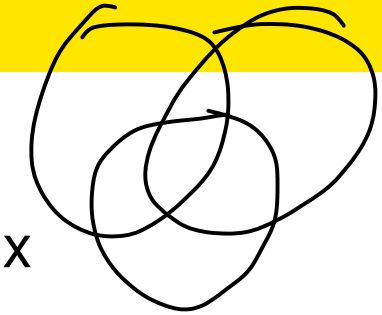
Codes used in DVB

n	k	d_{min}	t	r	Code Type
3240	3072	25	12	0.9481	BCH
5400	5232	25	12	0.9689	BCH
6480	6312	25	12	0.9741	BCH
7200	7032	25	12	0.9767	BCH
9720	9552	25	12	0.9827	BCH
10800	10632	25	12	0.9844	BCH
11880	11712	25	12	0.9859	BCH
12600	12432	25	12	0.9867	BCH
13320	13152	25	12	0.9874	BCH
14400	14232	25	12	0.9883	BCH

$m = 2^{14232}$

Complexity $= (n-k)^2$ — for bounded distance

The Hamming Code



The Hamming code has the following parity check matrix

$$H = \left[\begin{array}{cccc|ccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right]$$

and generator matrix

$$G = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

The Hamming Code

Let

$$\begin{aligned} \mathbf{v_1} &= (1,0,0,0,1,0,1) \\ \mathbf{v_2} &= (0,1,0,0,1,1,1) \\ \mathbf{v_3} &= (0,0,1,0,1,1,0) \\ \mathbf{v_4} &= (0,0,0,1,0,1,1) \end{aligned}$$

not unique

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

$$\begin{aligned} \mathbf{v_1'} &= (1,0,1,1,0,0,0) \\ \mathbf{v_2'} &= (0,1,0,1,1,0,0) \\ \mathbf{v_3'} &= (0,0,1,0,1,1,0) \\ \mathbf{v_4'} &= (0,0,0,1,0,1,1) \end{aligned}$$

The Hamming Code

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

$$\begin{aligned} \mathbf{v_1}' &= \mathbf{v_1} + \mathbf{v_3} + \mathbf{v_4} \\ \mathbf{v_2}' &= \mathbf{v_2} + \mathbf{v_4} \\ \mathbf{v_3}' &= \mathbf{v_3} \\ \mathbf{v_4}' &= \mathbf{v_4} \end{aligned}$$

To show that they are linearly independent consider

$$a_1\mathbf{v_1}' + a_2\mathbf{v_2}' + a_3\mathbf{v_3}' + a_4\mathbf{v_4}', \qquad a_i \in \{0, 1\}$$

The Hamming Code

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.

The Hamming Code

These basis vectors are cyclic shifts of each other. Also a cyclic shift of $\mathbf{v_4}'$ is $\mathbf{v_1}' + \mathbf{v_3}' + \mathbf{v_4}'$. 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 Hamming Code

For a linear code, the minimum distance = minimum weight codeword (except 0)

The codewords in this code are

$(0,0,0,0,0,0,0)$	$(1,1,1,1,1,1,1)$
$(1,0,0,0,1,0,1)$	$(0,1,0,0,1,1,1)$
$(1,1,0,0,0,1,0)$	$(1,0,1,0,0,1,1)$
$(0,1,1,0,0,0,1)$	$(1,1,0,1,0,0,1)$
$(1,0,1,1,0,0,0)$	$(1,1,1,0,1,0,0)$
$(0,1,0,1,1,0,0)$	$(0,1,1,1,0,1,0)$
$(0,0,1,0,1,1,0)$	$(0,0,1,1,1,0,1)$
$(0,0,0,1,0,1,1)$	$(1,0,0,1,1,1,0)$

weight 3

weight 4

info bit

- The above code is called a cyclic code because every cyclic shift of a codeword is also a codeword.

Geometry of the Hamming Code

- The minimum distance of this code is 3 (the minimum weight of any nonzero codeword).
- Thus this code is capable of correcting 1 error.
- Consider the number of received vectors that are decoded into a given codeword
  - 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.

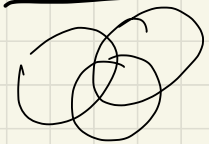
Coset Leaders of the Hamming Code

- 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)

- So the column of the parity check matrix that matches the syndrome indicates the position where the single error occurred.

codeword error prob for Hamming code



If 2 or more errors are made,

then decoder will output an incorrect codeword

$n=3$

$d_{min}=3 \Rightarrow$

$e=1$

# of vectors in a decoding region is

$1+7=8$

1 codeword, 7 vectors that different in 1 position

# of vectors in all decoding regions

$=16 \times 8 = 128 = 2^7$ = all possible vectors

(perfect code)

If 2 or more errors are made, then decoder outputs incorrect codeword

$$P(\text{codeword error}) = P(\text{2 or more errors in 7 positions})$$

$$= \sum_{l=2}^{7} P(l \text{ errors})$$

$$= \sum_{l=2}^{7} \binom{7}{l} p^l (1-p)^{7-l} \quad \text{(codeword error prob)}$$

What is Bit error probability? (Info bits)

Simplified Decoding of the Hamming Code

- 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.

Bit Error Probability of the Hamming Code

The bit error probability of the Hamming code can be calculated (for hard decision decoding) as

$$\begin{aligned} P_b &= 9p^2(1-p)^5 + 19p^3(1-p)^4 + 16p^4(1-p)^3 \\ &\quad + 12p^5(1-p)^2 + 7p^6(1-p) + p^7 \end{aligned}$$

To see how the first term is calculated consider the all zeros codeword as the transmitted codeword.

- There are seven codewords that are distance 3 from the all zeros codeword. One of these is the codeword (1,0,0,0,1,0,1). There are three vectors that are weight 2 and distance 1 from this codeword that would cause 1 error in four information bits.
- Another codeword is (1,1,0,0,0,1,0). There are three vectors that are weight 2 and distance 1 from this codeword that would cause 2 errors in four bits.

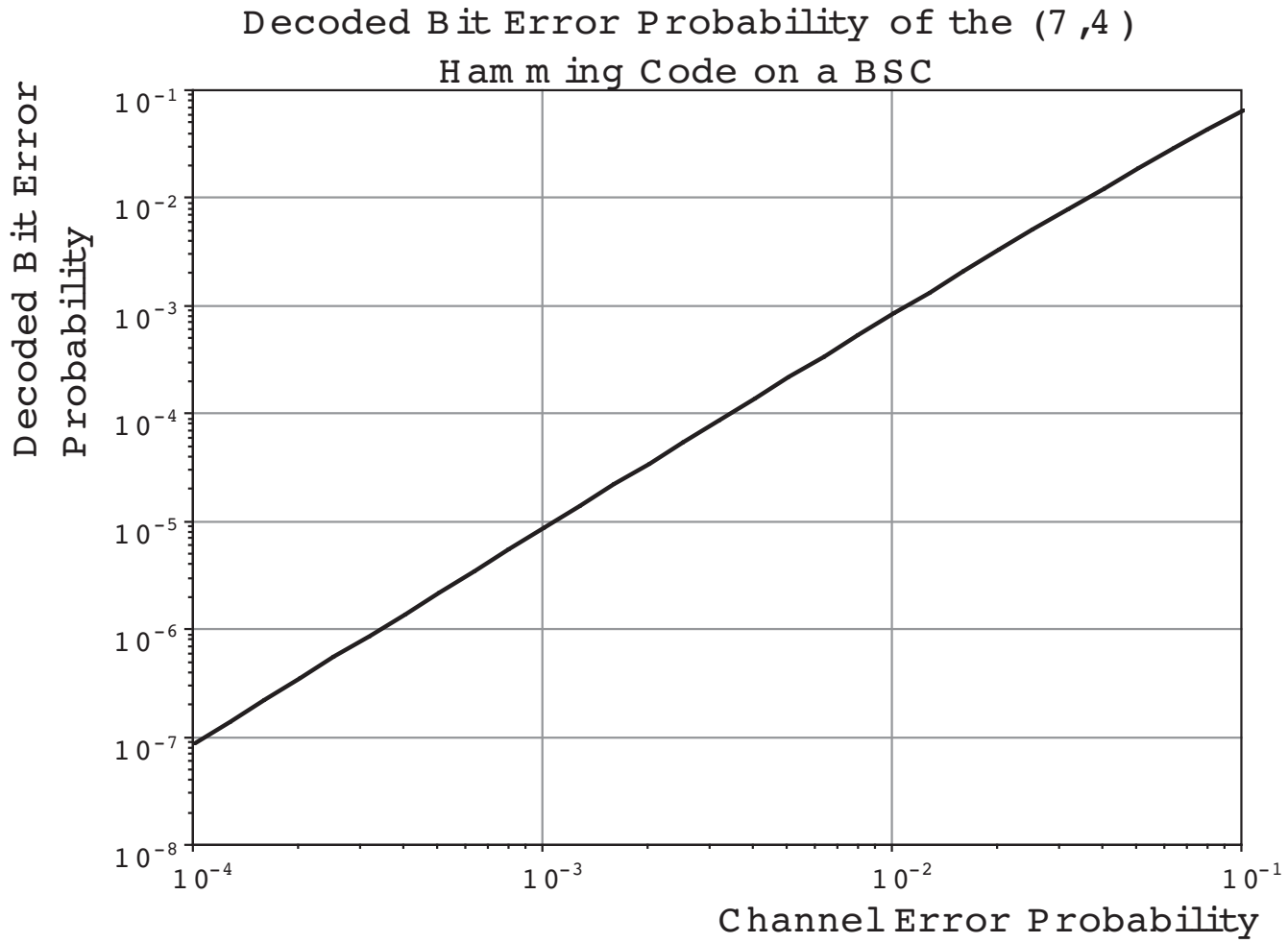
Bit Error Probability of the Hamming Code

- Continuing in this manner we get

$$3(1) + 3(2) + 3(2) + 3(3) + 3(2) + 3(1) + 3(1) = 36$$

  as the number of a bit errors caused by a weight two error pattern.
- Since there are four bits transmitted we get $\frac{36}{4} p^2 (1-p)^5 = 9p^2(1-p)^5$.

Bit Error Probability of the Hamming Code



Codeword Error Probability of The Hamming Code

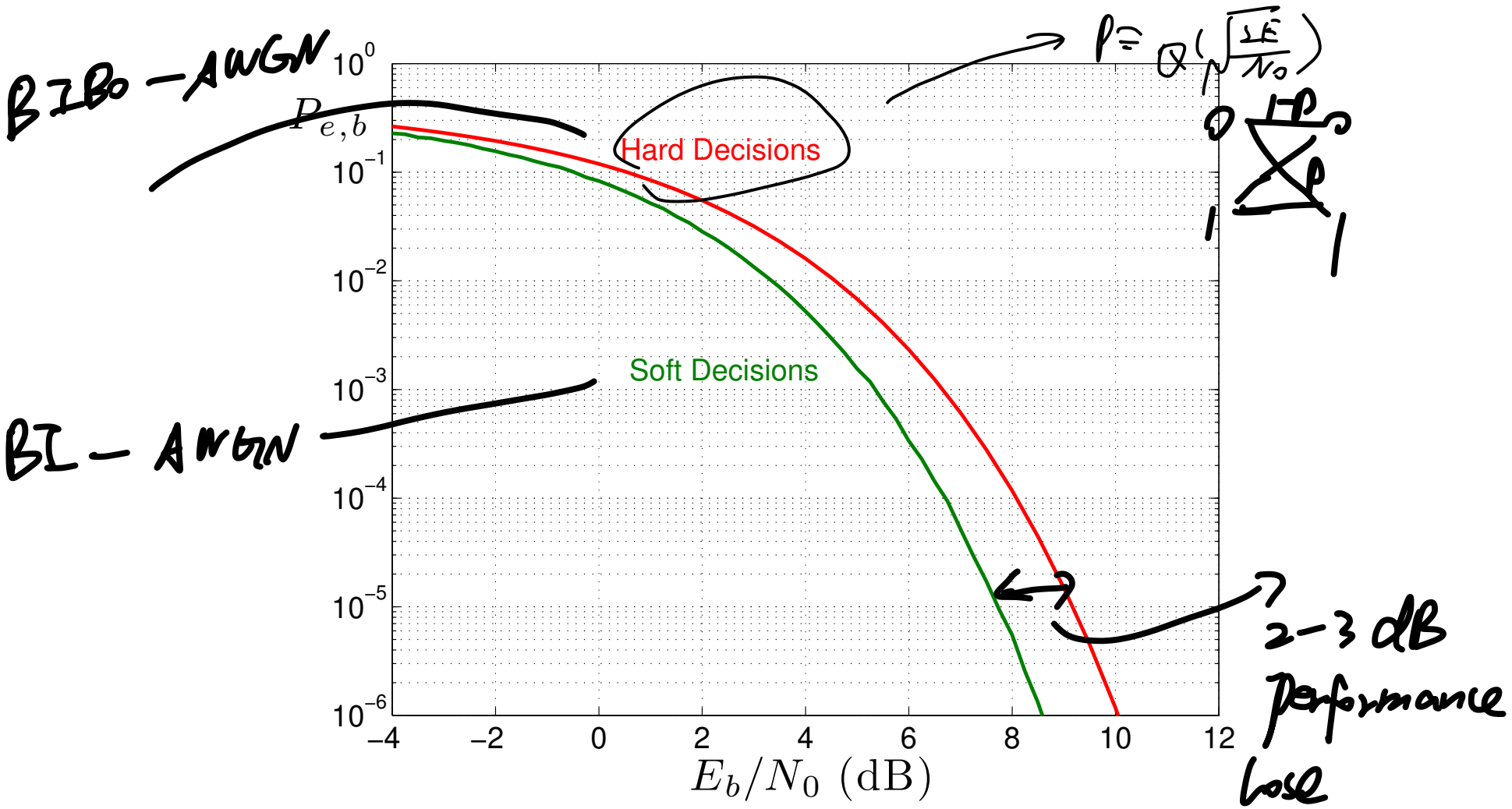
AWGN Soft decision

The bound on the codeword error probability for soft decision decoding is

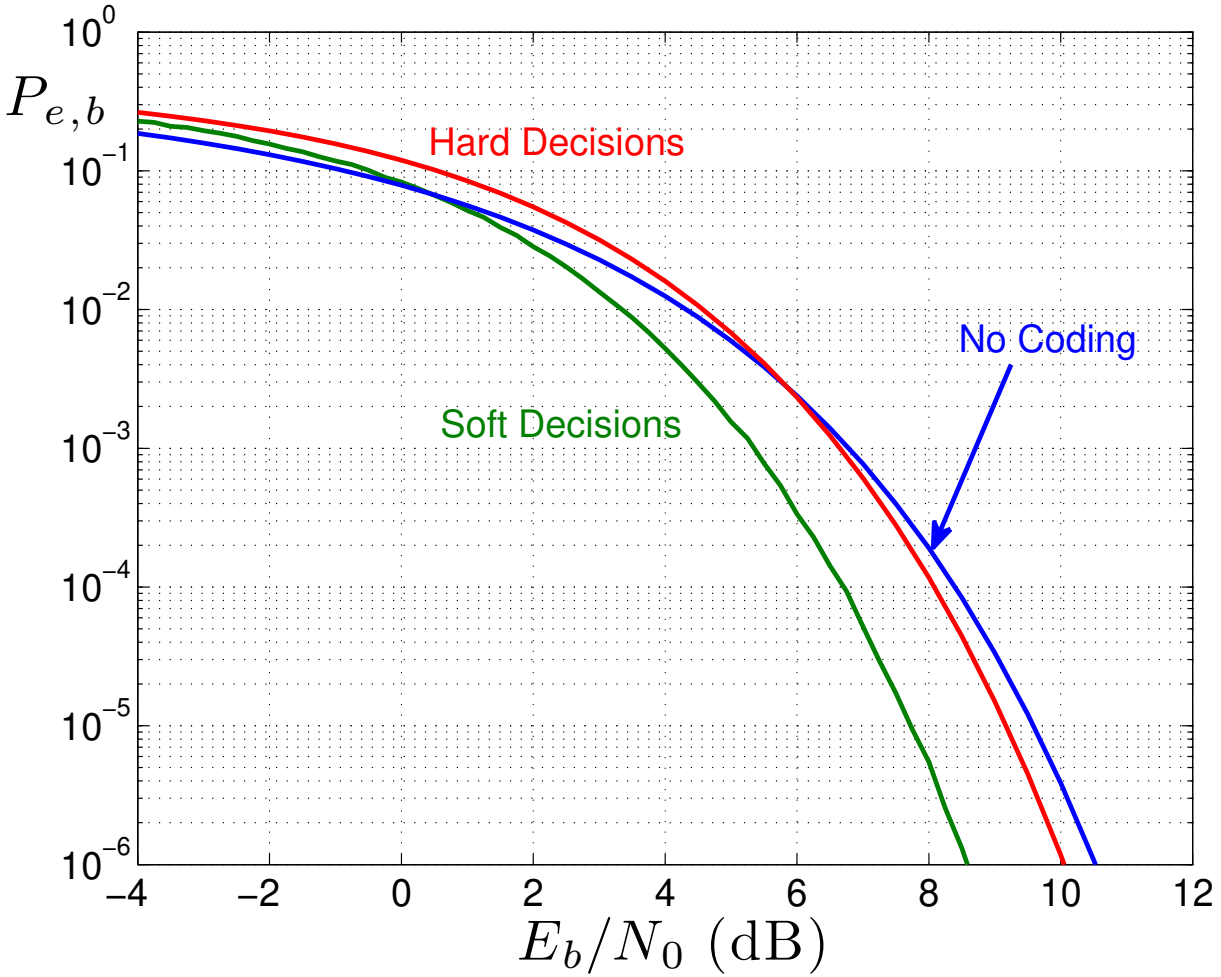
$$P_e \leq \underbrace{7Q\left(\sqrt{\frac{6E}{N_0}}\right) + 7Q\left(\sqrt{\frac{8E}{N_0}}\right) + Q\left(\sqrt{\frac{14E}{N_0}}\right)}_{\text{Union Bound}} \leq \overbrace{7D^3 + 7D^4 + D^7}^{\text{Union-Bhattacharyya Bound}}$$

where $D = e^{-E/N_0}$ and $E = 4E_b/7$.

Bit Error Probability of The Hamming Code

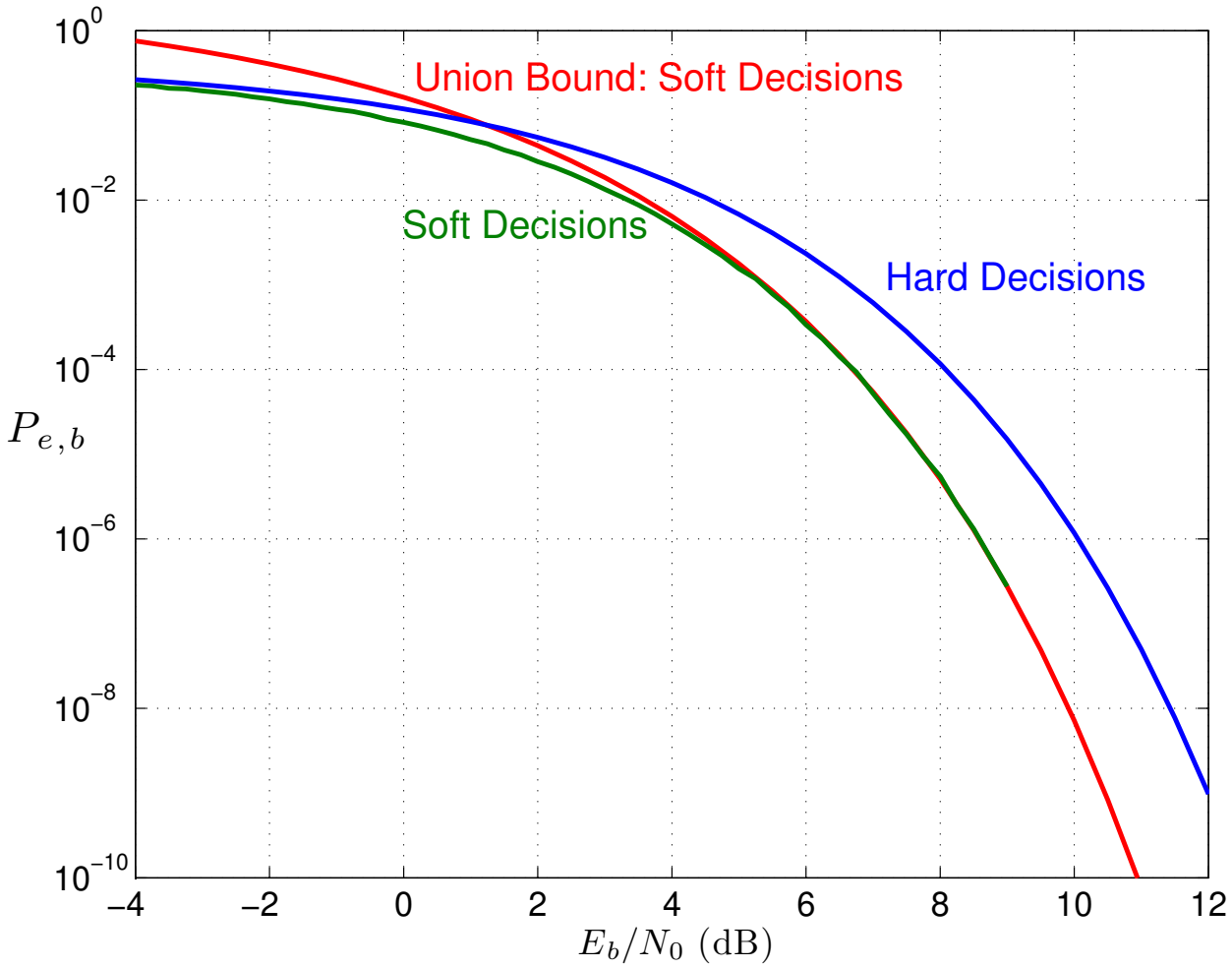


Bit Error Probability of The Hamming Code



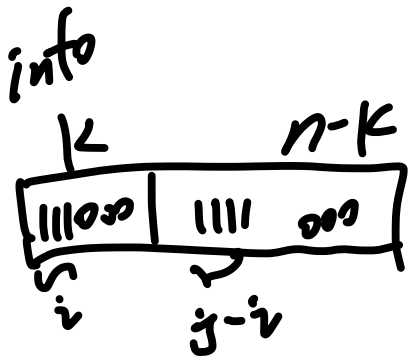
Bit Error Probability of The Hamming Code

BIBO-AWGN (Hard Decisions) vs. BI-AWGN (Soft Decisions)



General Bit Error Probability Bounds

Let $A_{i,j}$ be the number of codewords with i information bits being 1 and weight j. The probability of bit error can be bounded as



$$P_{e,b} \leq \sum_{i=1}^{k}\sum_{j=1}^{n} \frac{i}{k} A_{i,j} P_2(j)$$

$$= \sum_{j=1}^{n}\left(\sum_{i=1}^{k} \frac{i}{k} A_{i,j}\right) P_2(j)$$

$$= \sum_{j=1}^{n} w_j P_2(j)$$

Union Bound on codeword error prob

$$P_{e,c} \leq \sum_{d=d_{min}}^{n} A_d P_2(d)$$

where

$$w_j = \sum_{i=1}^{k} \frac{i}{k} A_{i,j}.$$

Golay Code

12 information bit Correct 3 errors

- The Golay code has length 23, dimension 12 and distance 7.
- In the Golay code there are 253 vectors that are distance 78 from the all zero codeword, 506 vectors that are distance 8, 1288 that are distance 11, 1288 that are distance 12, 506 that are distance 16, 253 that are distance 17 and 1 that is distance 23.
- The extended Golay code adds a single parity check to each codeword to make sure the number of ones is even.
- The extended Golay code has length 24, dimension 12 and distance 8.
- In the extended Golay code there are 759 vectors that are distance 8 from the all zero codeword, 2576 that are distance 12, 759 that are distance 16 and 1 that is distance 24.
- The code can correct 3 errrors. Note that $d_{min} r = 8(.5) = 4 = 6.02$dB.

$P_{e,c} \leq f(d_{min} r)$

Geometry of the (23, 12) Golay Code

- The Golay code has distance 7 and can correct 3 errors.
- There are $\binom{23}{0} + \binom{23}{1} + \binom{23}{2} + \binom{23}{3} = \underline{\quad 2^{11} \quad}$ vectors in the decoding region of each codeword. 2048
- There are $\underline{2^{12}}$ codewords.
- The total number of vectors in all the decoding regions is the number of vectors in each decoding region of each codeword times the number of codewords = $(2^{12})(\underline{\quad 2^{11} \quad}) = \underline{\quad 2^{23} \quad}$.
- There are $\underline{\quad 0 \quad}$ vectors not in the decoding region of any codeword.

$\Rightarrow$ this is a perfect code

Generator for the (23,12) Golay Code

$$G = \left[\begin{array}{c|c} I_{12} & \\ \hline \begin{array}{c} 100000000000 \\ 010000000000 \\ 001000000000 \\ 000100000000 \\ 000010000000 \\ 000001000000 \\ 000000100000 \\ 000000010000 \\ 000000001000 \\ 000000000100 \\ 000000000010 \\ 000000000001 \end{array} & \begin{array}{c} 100010110111 \\ 111001010101 \\ 101111000110 \\ 110100001111 \\ 101001101011 \\ 100111011001 \\ 100111101100 \\ 111010011010 \\ 111110100001 \\ 101100111100 \\ 110101110010 \\ 111111111111 \end{array} \end{array}\right]$$

Parity Check for the (23,12) Golay Code

$$H = \left[\begin{array}{c|c} 11111111111 & 100000000000 \\ 01010011101 & 010000000000 \\ 01101001110 & 001000000000 \\ 00110100111 & 000100000000 \\ 10100111100 & 000010000000 \\ 01101110001 & 000001000000 \\ 10001010111 & 000000100000 \\ 11000101011 & 000000010000 \\ 00011111010 & 000000001000 \\ 11110010010 & 000000000100 \\ 10111001001 & 000000000010 \\ 11011100100 & 000000000001 \end{array}\right]$$

(23,12) Golay Code

weight enumerator

j	A_j	j	A_j	j	A_j	j	A_j
0	1	6	0	12	1288	18	0
1	0	7	253	13	0	19	0
2	0	8	506	14	0	20	0
3	0	9	0	15	506	21	0
4	0	10	0	16	253	22	0
5	0	11	1288	17	0	23	1

Using A_j can find the union bound on codeword error probability

(24,12) Golay Code

The (24,12) Golay code has an extra parity check added to every codeword in the (23,12) code. Every odd weight code word in the (23,12) Golay code becomes an even weight codeword in the (24,12) Golay code. The even weight codewords in the (23,12) code do not change weights. The weight enumerator for the (24,12) code is shown below.

j	A_j	j	A_j	j	A_j	j	A_j	j	A_j
0	1	5	0	10	0	15	0	20	0
1	0	6	0	11	0	16	759	21	0
2	0	7	0	12	2576	17	0	22	0
3	0	8	759	13	0	18	0	23	0
4	0	9	0	14	0	19	0	24	1

using

(24,12) Golay Code $A_{i,j}$

using this to calculate union bound on bit error prob

# of code bit = 1

$i \backslash j$	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	11	0	0	0	1
2	0	0	0	0	0	0	0	0	66	0	0	0	0
3	0	0	0	0	0	0	0	0	165	0	0	0	55
4	0	0	0	0	0	0	0	0	275	0	0	0	20
5	0	0	0	0	0	0	0	0	165	0	0	0	616
6	0	0	0	0	0	0	0	0	66	0	0	0	792
7	0	0	0	0	0	0	0	0	11	0	0	0	616
8	0	0	0	0	0	0	0	0	0	0	0	0	220
9	0	0	0	0	0	0	0	0	0	0	0	0	55
10	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0

# of infor bit = 1

(24,12) Golay Code $A_{i,j}$

$i \backslash j$	13	14	15	16	17	18	9	20	21	22	23	24
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	11	0	0	0	0	0	0	0	0
6	0	0	0	66	0	0	0	0	0	0	0	0
7	0	0	0	165	0	0	0	0	0	0	0	0
8	0	0	0	275	0	0	0	0	0	0	0	0
9	0	0	0	165	0	0	0	0	0	0	0	0
10	0	0	0	66	0	0	0	0	0	0	0	0
11	0	0	0	11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1

(24,12) Golay Code w_j

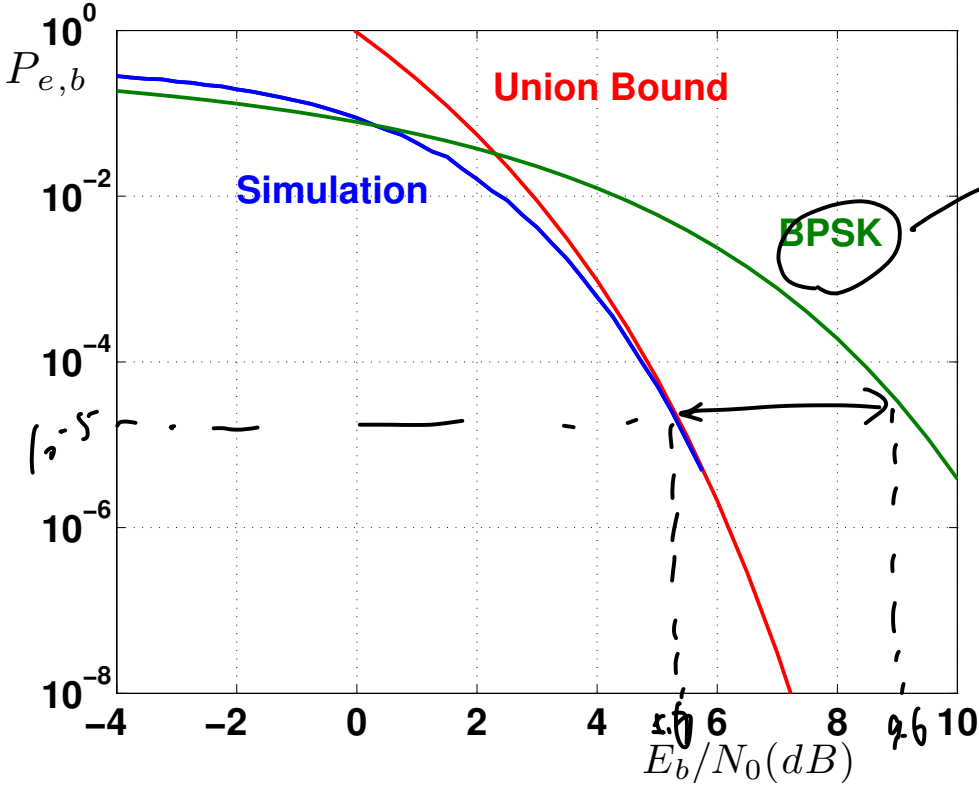
j	w_j	j	w_j	j	w_j	j	w_j
0	0	7	0	13	0	19	0
1	0	8	253	14	0	20	0
2	0	9	0	15	0	21	0
3	0	10	0	16	506	22	0
4	0	11	0	17	0	23	0
5	0	12	1288	18	0	24	1
6	0						

Bit error probability of (24,12) Golay Code Performance

rate = $\frac{1}{2}$

trade off energy and data rate and complexity

ending and decoding

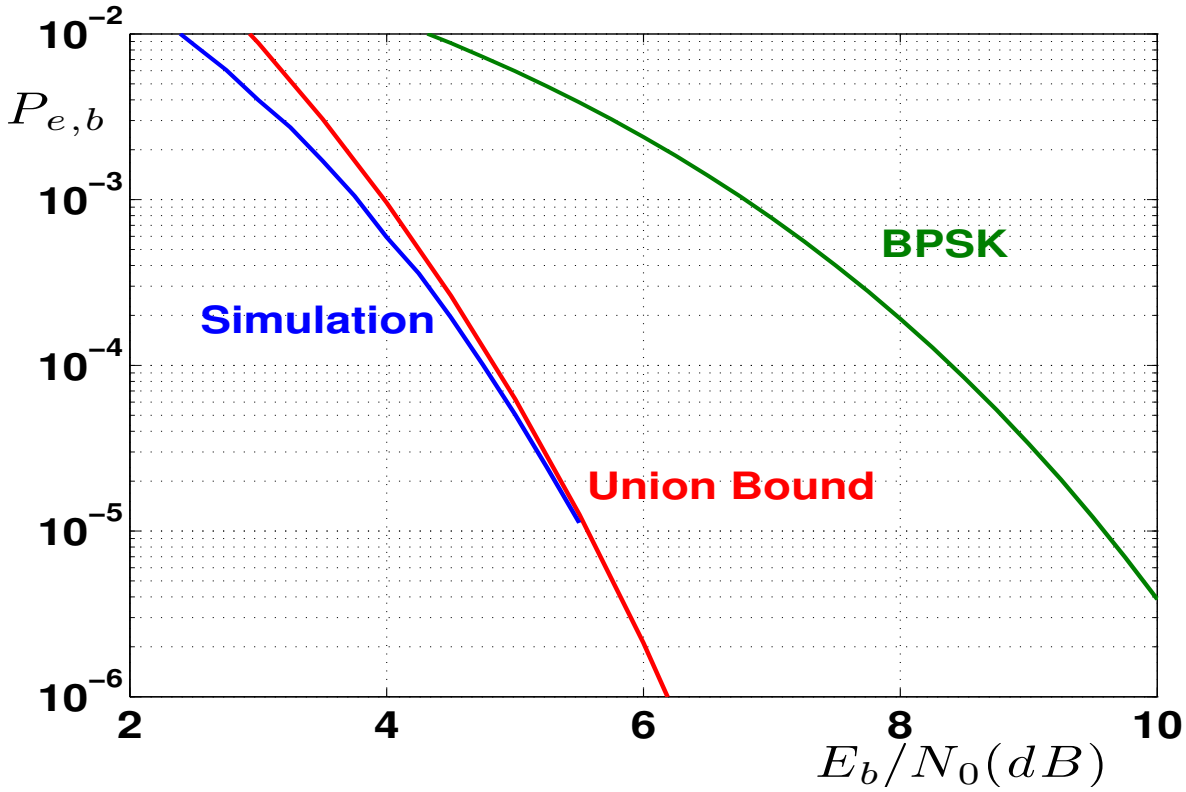


no coding

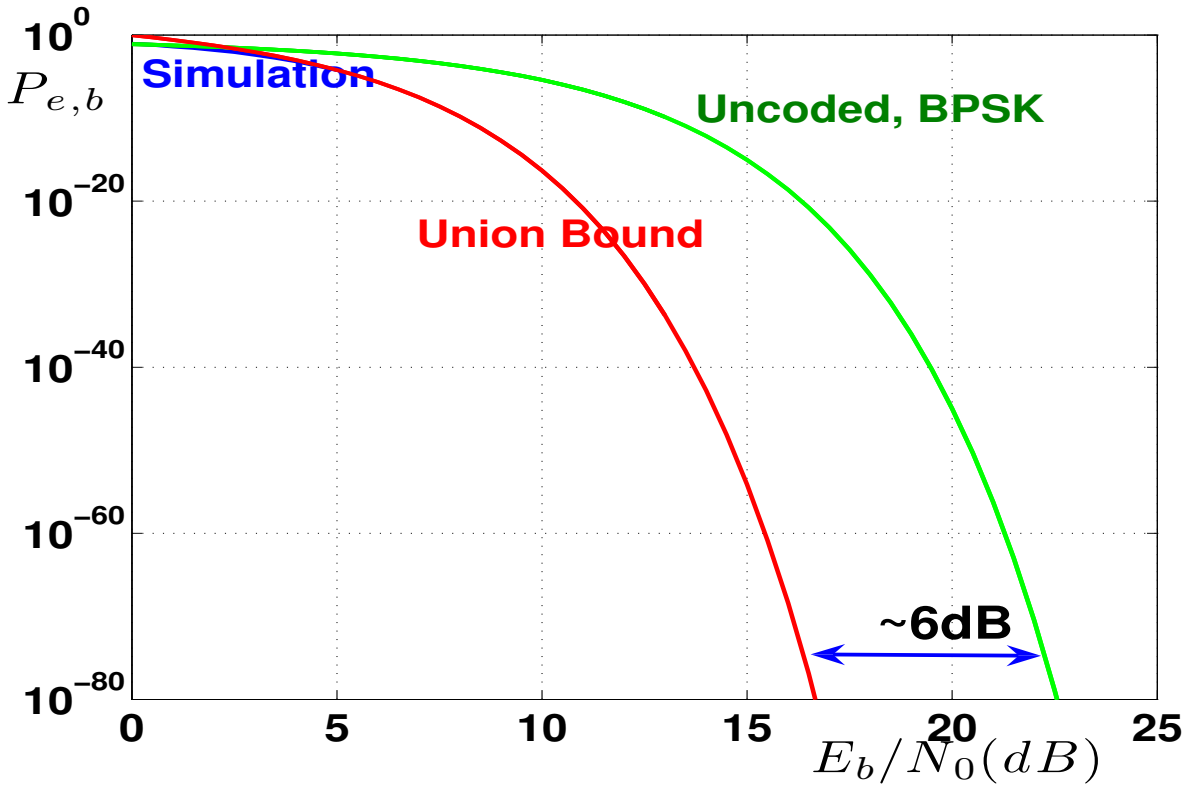
~ 4 dB

Save 4 dB energy at the expense of decreasing the data rate, but increasing complexity

Golay Code Bit Error Probability



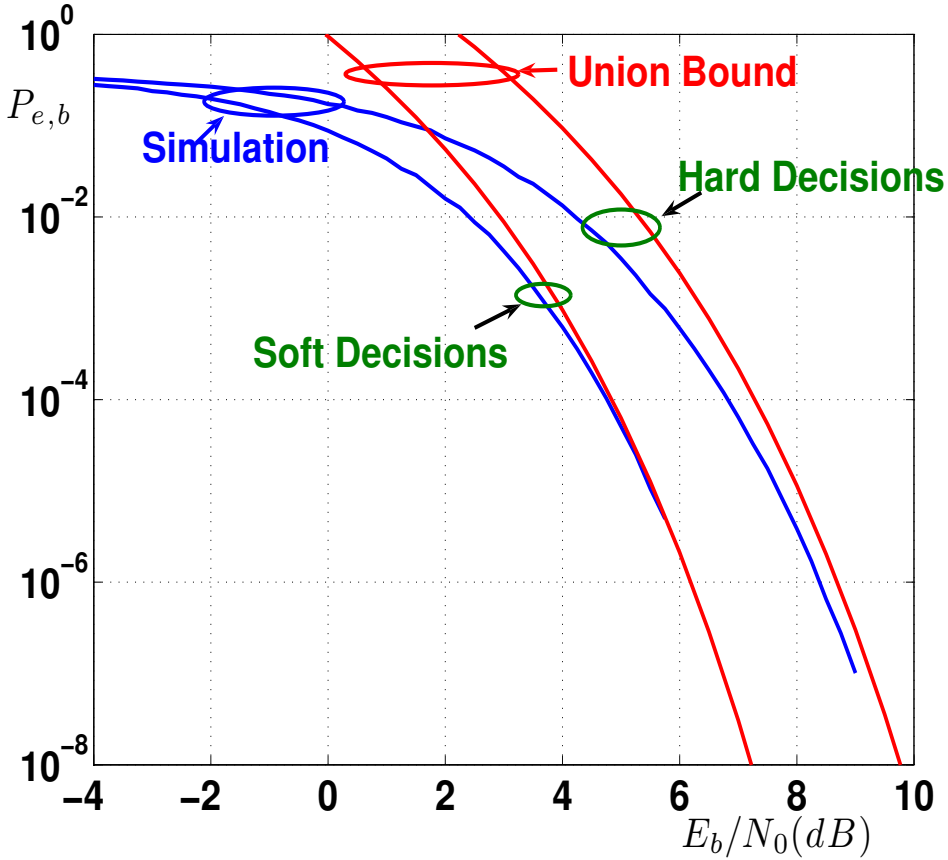
Golay Code Bit Error Probability



Golay Code Bit Error Probability: BIBO-AWGN vs BI-AWGN

Hard decisions

Soft decisions



complexity

n block code 2^{n-k} linear code

Cyclic Codes

cyclic shift

$(c_0, c_1, \ldots, c_{n-1}) \rightarrow (c_{n-1}, c_0, \ldots, c_{n-2})$

Definition: A linear cyclic code is a linear block code such that every cyclic shift of a codeword is also a codeword

Notation: It is convenient to represent all codewords as polynomials

Example: If $c = (c_0, c_1, \ldots, c_{n-1})$ is a codeword of length n with $c_i \in \{0, 1\}$ we will write this as

$$c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$$

$$xc(x) = c_0 x + c_1 x^2 + \cdots + c_{n-2} x^{n-1} + c_{n-1} x^n$$

$$c_0 x + c_1 x^2 + \cdots + c_{n-2} x^n + c_{n-1} x^n + c_{n-1} - c_{n-1}$$

$$xc(x) = \underbrace{c_{n-1} + c_0 x + c_1 x^2 + \cdots + c_{n-2} x^{n-1}}_{\text{cyclic shift of } c(x)} + c_{n-1}(x^n - 1)$$

Cyclic Codes

Claim:

A cyclic shift of a codeword $c(x)$ corresponds to finding the remainder of $xc(x)$ divided by $x^n - 1$, that is

$$\frac{xc(x)}{x^n - 1} = c_{n-1} + \underbrace{\frac{c_{n-1} + c_0 x + \cdots + c_{n-2}x^{n-1}}{x^n - 1}}_{\text{Remainder Term}}.$$

Cyclic Codes

Proof:

Let $c_1(x) = c_{n-1} + c_0 x + \cdots + c_{n-2}x^{n-1} = xc(x) - c_{n-1}(x^n - 1)$. Then $c_1(x)$ is the remainder when $c(x)x$ is divided by $x^n - 1$. That is,

$$xc(x) = c_1(x) - c_{n-1}(x^n - 1)$$

where $c_1(x)$ is a polynomial of degree $n-1$ or less. Thus $[xc(x) \equiv c_1(x)] \bmod (x^n - 1)$ i.e. $xc(x) - c_1(x)$ is a multiple of $x^n - 1$.

Polynomial Representation Arithmetic

- All polynomials have binary coefficients. For example $a(x) = 1x^4 + 0x^3 + 1x^2 + 1x + 1$.
- Multiplication of polynomials is done as usual with mod two addition of coefficients. For example

$$\begin{aligned} a(x) &= 1x^4 + 0x^3 + 1x^2 + 1x + 1 \\ b(x) &= 1x^4 + 1x^3 + 0x^2 + 0x + 1 \\ \hline a(x) + b(x) &= 0x^4 + 1x^3 + 1x^2 + 1x + 0 \\ a(x)b(x) &= 1x^8 + 1x^7 + 1x^6 + 0x^5 + 1x^4 + 1x^3 + 1x^2 + 1x + 1 \end{aligned}$$

Polynomial Representation Arithmetic

- Consider a polynomial $g(x)$ of degree $n-k$ and a polynomial $m(x)$ of degree $k-1$

$$\begin{aligned} g(x) &= g_0 + g_1 x + \cdots + g_{n-k} x^{n-k} \\ m(x) &= m_0 + m_1 x + \cdots + m_{k-1} x^{k-1} \\ & \qquad m_i \in \{0,1\}, \quad g_i \in \{0,1\} \end{aligned}$$

k coef

generator polynomial

message polynomial

- Consider all polynomials of the form

$$c(x) = m(x)g(x)$$

codeword poly

degree n-1 => n coef

for some $m(x)$, where $g(x)$ is fixed.

- There are 2^k such polynomials because there are 2^k such $m(x)$. ($c_1 = c_2 \Longrightarrow m_1 = m_2$)

Polynomial Representation of Binary Cyclic Codes

Claim:

If $x^n - 1$ is a multiple of $g(x)$ ($g(x)$ is a divisor of $x^n - 1$) then set of polynomials generated from $m(x)g(x)$ is a cyclic code

Proof:

Consider any codeword $c(x) = m(x)g(x)$. A cyclic shift of $c(x)$ produces

$$c_1(x) = xc(x) - (x^n - 1)c_{n-1}$$

multiple of $g(x)$

Since $c(x)$ is a multiple of $g(x)$ and $x^n - 1$ is a multiple of $g(x)$ so is $c_1(x)$. Thus a cyclic shift of any codeword is a codeword.

Polynomial Representation of Binary Cyclic Codes

Consider $n = 7$. The factorization of $x^7 - 1$ over polynomials with binary coefficients is

$$x^7 - 1 = (x-1)(x^3 + x^2 + 1)(x^3 + x + 1)$$

We now consider three examples of cyclic codes of length 7. The first one is the (7,4) Hamming code, the second one is the simple single parity check code and the last one is the repetition code.

Polynomial Representation of Binary Cyclic Codes: Example 1:

$m(x)=0 \Rightarrow c(x)=0$

$$
\begin{aligned}
g(x) &= (1+x^2+x^3) \\
m(x) &= 1 \Longrightarrow 1+x^2+x^3 \text{ is a codeword} \\
(1011000) &= c_4 \\
m(x) &= 0 \Longrightarrow 0 \text{ is a codeword} \\
(0000000) & \\
m(x) &= 1+x \\
\Longrightarrow c(x) &= (1+x^2+x^3)(1+x) \qquad x^3+x^3 = 0 \mod 2 \\
&= 1+x^2+x^3+x+x^3+x^4 \\
&= 1+x+x^2+x^4 = c_{11} \\
m(x) &= 1+x+x^3 \\
c(x) &= (1+x^2+x^3)(1+x+x^3) \\
&= 1+x+x^2+x^3+x^4+x^5+x^6 = c_{15}.
\end{aligned}
$$

Polynomial Representation of Binary Cyclic Codes: Example 2:

Let

$$g(x) \quad = \quad (1 + x + x^3).$$

This is also a (7,4) Hamming code. In fact it can be found by taking the reverse of every codeword in the previous example.

Polynomial Representation of Binary Cyclic Codes: Example 3:

Let

$$g(x) = (1 + x).$$

Codewords are of the form

$$c(x) = m(x)(x + 1).$$

Notice that $c(1) = m(1)0$ implies that

$$c(1) = 0$$

so that

$$c_0 + c_1 + \cdots + c_{n-1} = 0$$

or in other words the number of ones in any codeword is even. Clearly any cyclic shift of a codeword is another codeword. This is the $(n, n-1)$ single (even) parity check code.

Polynomial Representation of Binary Cyclic Codes: Example 4:

$(x^7-1) = (x-1)(x^3+x^2+1)(x^3+x+1)$

Let

$$g(x) = (x^3 + x^2 + 1)(x^3 + x + 1) = (1 + x + x^2 + ... + x^6)$$

$n-k = 6$

$k = 1$

$M(x) = m_0$

The two codewords are

$$c(x) = 0$$

and

$$c(x) = g(x) = 1 + x + x^2 + ... + x^{n-1}$$

This is the $(n, 1)$ repetition code. It is clear that every codeword is a cyclic shift of another codeword.

$(0\ 000\cdots 0)$

$(1\ 111\cdots 1)$

Example

We now consider an example of cyclic codes of length $n = 15$. The factorization of $x^{15} - 1$ over polynomials with binary coefficients is

$$x^{15}-1 = (x-1)(x^2+x+1)(x^4+x+1)(x^4+x^3+1)(x^4+x^3+x^2+x+1).$$

Bluetooth uses a shortened Hamming code. The original Hamming code is a (15,11) code and this is shortened (set one information bit to zero) to a (15,10) code. The generator polynomial is $(x-1)(x^4+x+1)$.
Codes used in QR images.

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

This is a (15,5) code with distance 7 that can correct all patterns of 3 errors.

Code 5 is the previous example

$M = 32$, $n = 15$, $d_{H,min} = 7$, $r = \log_2(32)/15 = 1/3$

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

Codeword	$m(x)g(x)$	*Codeword*	$m(x)g(x)$
000000000000000	$0g(x)$	000011101100101	$x^4 g(x)$
000111011001010	$x^3 g(x)$	000100110101111	$(x^3 + x^4)g(x)$
001110110010100	$x^2 g(x)$	001101011110001	$(x^2 + x^4)g(x)$
001001101011110	$(x^2 + x^3)g(x)$	001010000111011	$(x^2 + x^3 + x^4)g(x)$
011101100101000	$xg(x)$	011110001001101	$(x + x^4)g(x)$
011010111100010	$(x + x^3)g(x)$	011001010000111	$(x + x^3 + x^4)g(x)$
010011010111100	$(x + x^2)g(x)$	010000111011001	$(x + x^2 + x^4)g(x$
010100001110110	$(x + x^2 + x^3)g(x)$	010111100010011	$(x + x^2 + x^3 + x^4)g(x)$
111011001010000	$g(x)$	111000100110101	$(1 + x^4)g(x)$
111100010011010		111111111111111	
110101111000100		110110010100001	
110010100001110		110001001101011	
100110101111000		100101000011101	
100001110110010		100010011010111	
101000011101100		101011110001001	
101111000100110		101100101000011	$(1 + x + x^2 + x^3 + x^4)g(x)$

Golay Code Example (also used in QR images)

Consider $n = 23$. The factorization of $x^{23} - 1$ over polynomials with binary coefficients is

$$x^{23}-1 = (x-1)(x^{11}+x^{10}+x^6+x^5+x^4+x^2+1)(x^{11}+x^9+x^7+x^6+x^5+x+1)$$

We now consider an example of cyclic codes of length 23.

$$\begin{aligned} g(x) &= (x+1)(x^{11}+x^9+x^7+x^6+x^5+x+1) \\ &= x^{12}+x^{11}+x^{10}+x^9+x^8+x^5+x^2+1. \end{aligned}$$

This is a (24,12) code with distance 8 that can correct all patterns of 3 errors (and detect patterns of 4 errors). In the code used in QR only 6 information bits are used (rather than 12) and 18 coded bits are transmitted. This is called a shortened code. Basically we set 6 of the information bits to zero.

Polynomial Representation of Binary Cyclic Codes

$(31, 26)$ Hamming code

Consider $n = 31$. The factorization of $x^{31} - 1$ over polynomials with binary coefficients is

$$\begin{aligned} x^{31} - 1 &= (x-1)(x^5 + x^2 + 1)(x^5 + x^3 + 1) \\ &\quad \times (x^5 + x^4 + x^3 + x^2 + 1)(x^5 + x^3 + x^2 + x + 1) \\ &\quad \times (x^5 + x^4 + x^2 + 1)(x^5 + x^4 + x^3 + x + 1) \end{aligned}$$

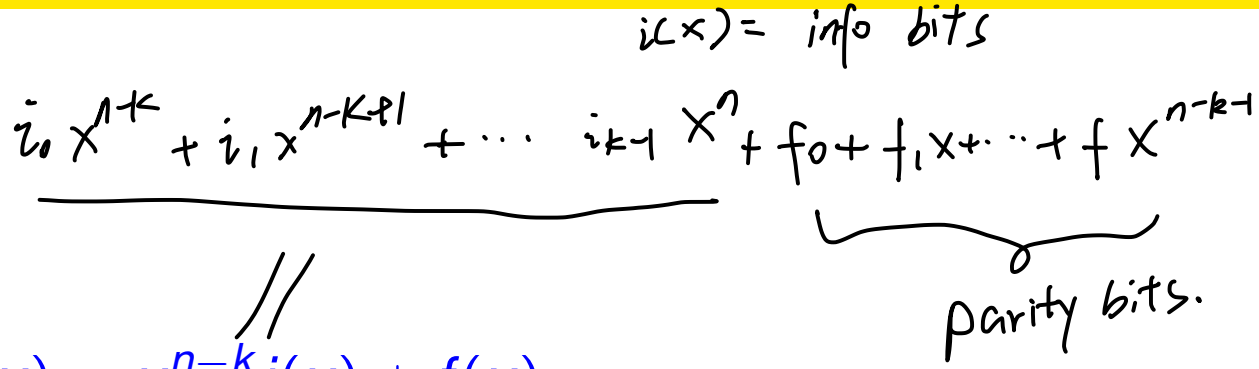
Systematic Encoding of Cyclic Codes

- For cyclic codes the codewords are polynomial multiples of the generator polynomial $c(x) = g(x)m(x)$. not systematic
- However with this approach the information bits are not directly "visible" in the codeword. That is, we can not look at a codeword and identify a component of the codeword as an information bit.
- If the information is directly visible (or in the clear) in the codeword, then the code is said to be **systematic**.
- To get a systematic encoding for a cyclic code we first we let the high order terms of the polynomial representation of the codeword be the information polynomial.
- We then choose the lower order terms to guarantee that the codeword is a multiple of $g(x)$.

$c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$

info bits

Systematic Encoding of Cyclic Codes



$$c(x) = x^{n-k} i(x) + f(x)$$

Since we require $c(x)$ to be a multiple of $g(x)$ we must get 0 when we divide by $g(x)$. That is

$$\text{Remainder}[c(x), g(x)] = 0$$

where $\text{Remainder}[a(x), b(x)]$ is the remainder after dividing $a(x)$ by $b(x)$.

Systematic Encoding of Cyclic Codes

Thus

$$\text{Remainder}[x^{n-k}i(x) + f(x), g(x)] = 0$$

$$\text{Remainder}[x^{n-k}i(x), g(x)] - \text{Remainder}[f(x), g(x)] = 0$$

Since $g(x)$ has degree $n-k$ and $f(x)$ has degree less than $n-k$ $\text{Remainder}[f(x), g(x)] = f(x)$. Thus

$$f(x) = -\text{Remainder}[x^{n-k}i(x), g(x)].$$

So encoding is done by dividing $i(x)x^{n-k}$ by $g(x)$ and finding the remainder.

Example

$C(x) = m(x) \cdot g(x)$

Suppose $n = 15$, $k = 11$,

Divide by $g(x)$

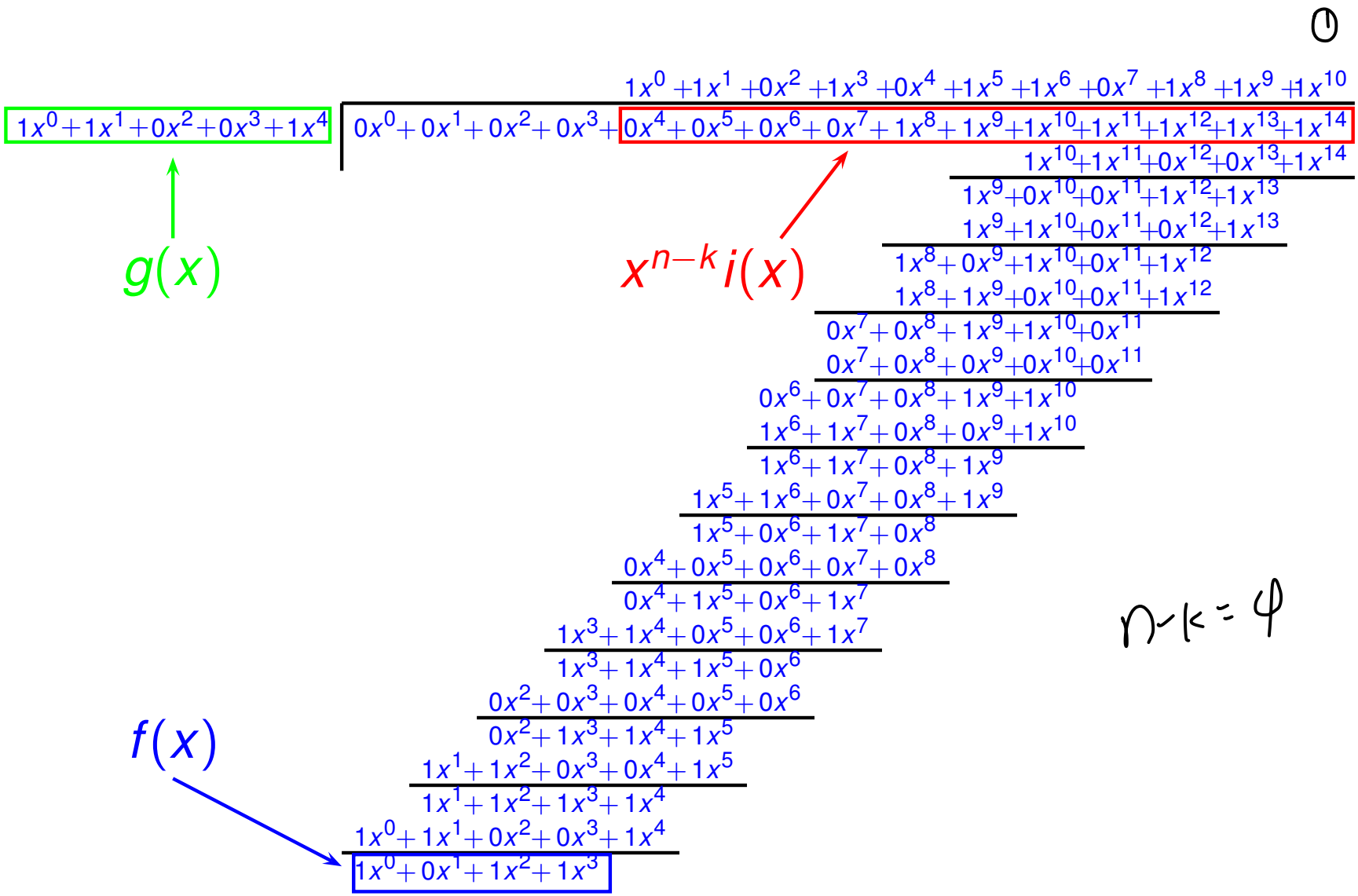
$$
\begin{aligned}
g(x) &= 1 + x + x^4 \\
i(x) &= x^4 + x^5 + x^6 + x^7 + x^8 + x^9 + x^{10} \\
i(x)x^{n-k} &= i(x)x^4 \\
&= 0x^4 + 0x^5 + 0x^7 + 1x^8 + 1x^9 + 1x^{10} + 1x^{11} + 1x^{12} + 1x^{13} + 1x^{14}
\end{aligned}
$$

To determine $c(x)$ we find the remainder when we divide $i(x)x^{n-k}$ by $g(x)$. Long division of polynomials.

DVB Standard Cyclic Code

$$
\begin{aligned}
g_1(x) &= 1 + x^2 + x^3 + x^5 + x^{16} \\
g_2(x) &= 1 + x + x^4 + x^5 + x^6 + x^8 + x^{16} \\
g_3(x) &= 1 + x^2 + x^3 + x^4 + x^5 + x^7 + x^8 + x^9 + x^{10} + x^{11} + x^{16} \\
g_4(x) &= 1 + x^2 + x^4 + x^6 + x^9 + x^{11} + x^{12} + x^{14} + x^{16} \\
g_5(x) &= 1 + x + x^2 + x^3 + x^5 + x^8 + x^9 + x^{10} + x^{11} + x^{12} + x^{16} \\
g_6(x) &= 1 + x^2 + x^4 + x^5 + x^7 + x^8 + x^9 + x^{10} + x^{12} + x^{13} + x^{14} + x^{15} + x^{16} \\
g_7(x) &= 1 + x^2 + x^5 + x^6 + x^8 + x^9 + x^{10} + x^{11} + x^{13} + x^{15} + x^{16} \\
g_8(x) &= 1 + x + x^2 + x^5 + x^6 + x^8 + x^9 + x^{12} + x^{13} + x^{14} + x^{16} \\
g_9(x) &= 1 + x^5 + x^7 + x^9 + x^{10} + x^{11} + x^{16} \\
g_{10}(x) &= 1 + x + x^2 + x^5 + x^7 + x^8 + x^{10} + x^{12} + x^{13} + x^{14} + x^{16} \\
g_{11}(x) &= 1 + x^2 + x^3 + x^5 + x^9 + x^{11} + x^{12} + x^{13} + x^{16} \\
g_{12}(x) &= 1 + x + x^5 + x^6 + x^7 + x^9 + x^{11} + x^{12} + x^{16} \\
g(x) &= g_1(x)g_2(x)g_3(x)g_4x)g_5(x)g_6(x)g_7(x)g_8(xg_9(x)g_{10}(x)g_{11}(x)g_{12}(x)
\end{aligned}
$$

Cyclic Redundancy Check (CRC) for Error Detection

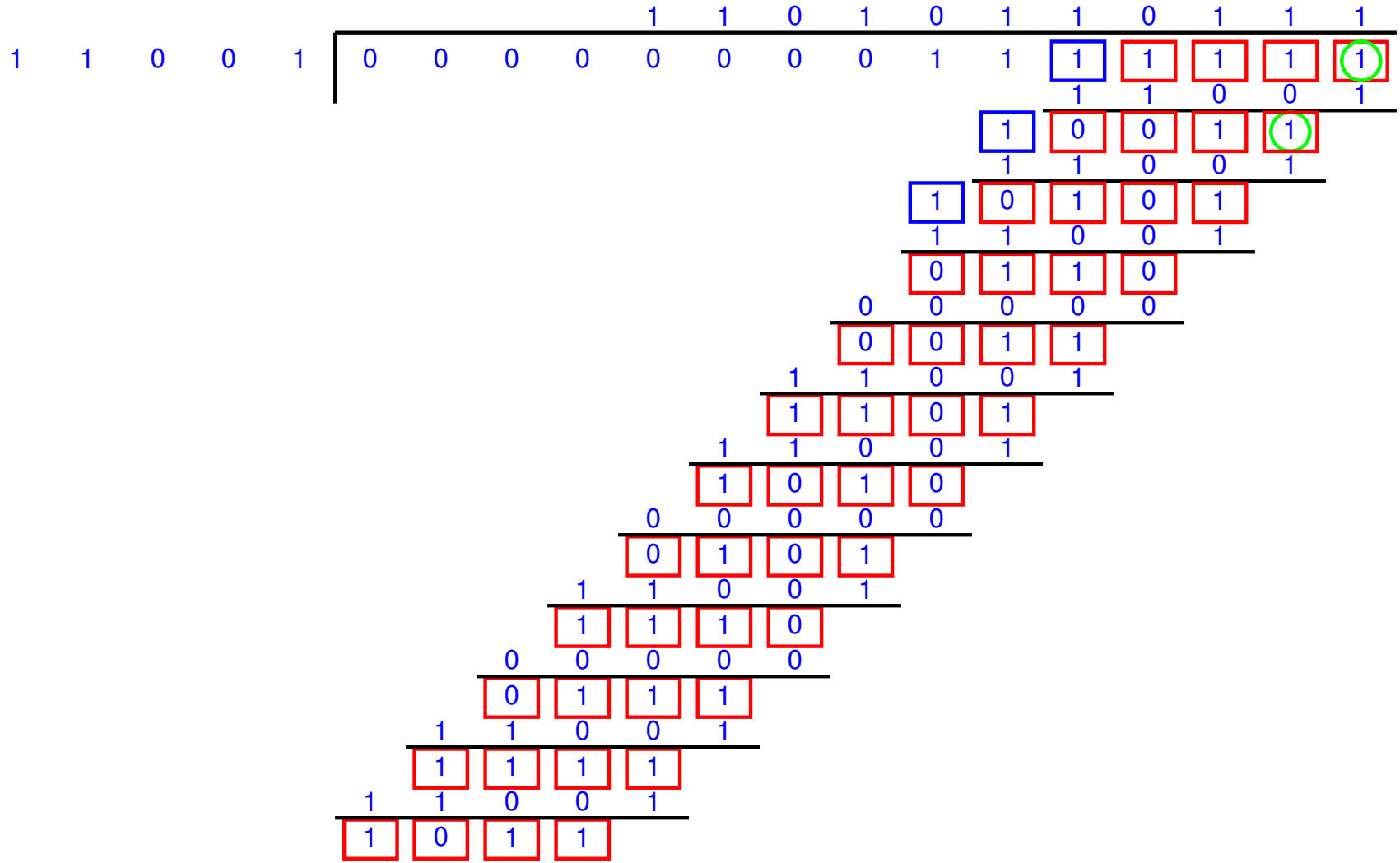


Systematic Encoding of Cyclic Codes

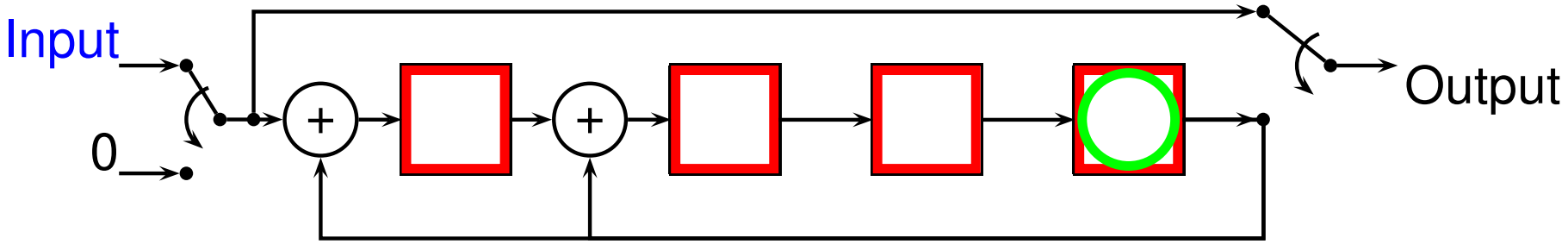
$$c(x) = \underbrace{1 + x^2 + x^3}_{f(x)} + \underbrace{x^8 + x^9 + x^{10} + x^{11} + x^{12} + x^{13} + x^{14}}_{i(x)x^{n-k}}$$

Systematic Encoding of Cyclic Codes

Dropping all x's but keep the places



4 bit Cyclic Code Encoder



- Shift register initialized to all zeros.
- Input information bits into shift register while also storing as output bits.
- After first information bit is in far right memory element run the shift register till the last information bit has been put into left most shift register element.
- Then continue operation of shift register while putting in $n-k$ zeros.
- After the $n-k$ zeros have been put into the shift register the contents of the shift register are the redundant bits and can be appended to the information bits already stored.

Cyclic Redundancy Check (CRC) for Error Detection

- Cyclic codes are used to detect errors in transmitted packets.
- A generator polynomial determines the redundant bits that need to be added to a set of information bits.
- If at the receiver the parity check equations are not satisfied then an error is detected and the packet may be discarded as being erroneous.
- Recalculating the parity checks is simply a matter of reencoding the information and checking that the parity symbols are idential to what was received.
- Some typical generator polynomials are

Cyclic Redundancy Check (CRC) for Error Detection

Type	$g(x)$
CRC-32	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$
CRC-24	$x^{24}+x^{23}+x^{14}+x^{12}+x^8+1$
CRC-16	$x^{16}+x^{15}+x^2+1$ (ANSI)
CRC-16	$x^{16}+x^{12}+x^5+1$ (CCITT X-25)
CRC-8	$x^8+x^7+x^6+x^4+x^2+1$
CRC-4	$x^4+x^3+x^2+x+1$