Golay Codes

The Golay codes are examples of perfect codes; they were discovered by the Swiss mathematician and information theorist, Marcel J. E. Golay in 1949. A binary Golay code is a type of linear error-correcting code used in digital communications.

The Voyager 1 and 2 spacecraft were launched towards Jupiter and Saturn in 1977. This code was used in the encoding and decoding of the general science and engineering (GSE) data for the missions.

In 1949 Marcel Golay noticed that:

$$\binom{23}{0} + \binom{23}{1} + \binom{23}{2} + \cdots + \binom{23}{12} = 2^{11}.$$

It indicated to him that the possibility of a (23, 12) perfect binary code existed that could correct 3 or fewer errors. This led to the binary form of the Golay code. It is one of the few examples of a nontrivial perfect code. This is the only known code capable of correcting any combination of three or fewer random errors in a block of 23 elements.

♣ **Binary Golay Codes.** There are two closely related binary Golay codes. The extended binary Golay code G_{24} is a $[\,24\,,\,12\,,\,8\,]$ code that encodes 12 bits of data in a 24-bit word in such a way that any 3-bit errors can be corrected or any 7-bit errors can be detected. The other, the perfect binary Golay code, G_{23} is a $[\,23\,,\,12\,,\,7\,]$ code that has codewords of length 23 and is obtained from the extended binary Golay code by deleting one coordinate position (conversely, the extended binary Golay code is obtained from the perfect binary Golay code by adding a parity bit).

The fact that the extended Golay code is used more often that the Golay code, first we define the extended Golay code.

♠ **Binary Extended Golay Codes.** It is a linear code generated by the 12×24 matrix $G_{24} = [\,I_{12}\,|\,A\,]$, where I_{12} is the 12×12 identity matrix and

$$A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \\ A_8 \\ A_9 \\ A_{10} \\ A_{11} \\ A_{12} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

Notice that the second row of the matrix A is obtained from the first row of A:

$$A_1 = [1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0],$$

by moving its first component to the last position. The same way each row of A is a right-shift of the previous row, except the last row.

- Another generator matrix for G_{24} is $[\, A \,|\, I_{12} \,]$.
- The code G_{24} is self-dual. Thus its parity-check matrix is $H = G_{24}^t$.
- The code G_{24} has no codeword of weight 4, so the distance of G_{24} is $d = 8$.
- The code G_{24} is an exactly three-error-correcting code.
- It is also common to call any binary linear code equivalent to the above code (moving some columns of G_{24} in different positions), an extended Golay code.
- The weight of every codeword in G_{24} is a multiple of 4.

All codewords in G_{24} have even weights.
Here is the weight distribution:

Weight	0	4	8	12	16	20	24
Words	1	0	759	2576	759	0	1

Here is another 12×12 matrix generating an Extended binary Golay code:

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

Notice that B is a symmetric matrix.

$$|C| \leq \frac{2^n}{\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t}}$$

Perfect Codes. A code C of odd distance $d = 2t + 1$ is called *perfect*, if C attains the Hamming bound. This means that

$$|C| = \frac{2^n}{\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t}}$$

♠ **Binary Golay Codes.** Hamming codes and Golay codes are the only non-trivial perfect codes.

To construct a Golay code, we use the vectors $u = 1\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 0$ and $J = 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1$, and $P = (p_{ij})$, the 11×11 full-cycle permutation matrix, i.e. the only non-zero entries are:

$$p_{2,1} = p_{3,2} = p_{i+1,i} = \ \cdots\ \cdots\ = p_{n,n-1} = p_{1,n} = 1\,.$$

Then define the matrix B as follows:

$$B = \begin{bmatrix} u & 1 \\ uP & 1 \\ uP^2 & 1 \\ uP^3 & 1 \\ uP^4 & 1 \\ uP^5 & 1 \\ uP^6 & 1 \\ uP^7 & 1 \\ uP^8 & 1 \\ uP^9 & 1 \\ uP^{10} & 1 \\ \\ J & 0 \end{bmatrix} = \begin{bmatrix} 1\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 0 & 1 \\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 1 & 1 \\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1 & 1 \\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 0 & 1 \\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1 & 1 \\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1 & 1 \\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1 & 1 \\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0 & 1 \\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 0 & 1 \\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 0 & 1 \\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1 & 1 \\ \\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 & 0 \end{bmatrix}$$

The binary linear code with generator matrix $\left[I_{12} \,|\, \widehat{B} \right]$ is called the binary Golay code and will be denoted by G_{23}.

Alternatively, the binary Golay code can be defined as the code obtained from the extended Golay code by deleting the last coordinate of every codeword.

Let B be the 12×11 matrix obtained from the matrix A by deleting its last column.

$$B = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad \text{or} \quad \widehat{B} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

- The code G_{23} is a perfect three-error-correcting code.
- The extended code of G_{23} is G_{24}.
- The distance d of G_{23} is 7.

♣ **Ternary Golay Codes.** In addition to binary Golay codes there are also ternary Golay codes.

The Extended Ternary Golay code is a $[12, 6, 6]$ code is the following matrix:

$$G_{12} = [\, I_6 \,|\, T_{6,6} \,] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 2 & 2 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 2 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 2 & 1 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 2 & 2 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 1 & 0 \end{bmatrix}$$

Notice that the matrix $T_{6,6}$ is a symmetric matrix. Any linear code that is equivalent to the above code (moving some columns of G_{12} in different positions) is also called an extended ternary Golay code. This code is also self-dual.

The ternary Golay code G_{11} is the code obtained by puncturing G_{12} (usually in the last coordinate). The parameters of this perfect code are $[11, 6, 5]$ and it is generated by the following matrix:

$$G_T = [\, I_6 \,|\, T_{6,5} \,] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 2 & 2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 2 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 2 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 1 \end{bmatrix}$$

♣ **Algoritm for Decoding Extended Golay Code.** The decoding algorithm shown below, consists in determining the error pattern $\mathbf{u} = \mathbf{v} + \mathbf{w}$, where $\mathbf{w}$ denotes the word received and $\mathbf{v}$, the nearest to $\mathbf{w}$. In the content of the algorithm $\mathbf{wt(x)}$ denotes the weight of the vector $\mathbf{x}$, (i.e. the number of "ones" contained in $\mathbf{x}$). After determining $\mathbf{u}$ we assume that the corrected received word will be $\mathbf{v = w + u.}$ Here are the steps of the algorithm:

STEP 1. Compute the syndrome:

$$\mathbf{s = mod\,(\,w\,G^{\,t}\,,\,2\,)}$$

STEP 2. If the weight of $\mathbf{s}$ is less than or equal to $\mathbf{(3)}$, then the error pattern is:

$$\mathbf{u} = [\,\mathbf{s}\,,\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\,].$$

STEP 3. If the weight of $\mathbf{s}$ is greater than $\mathbf{(3)}$, but for some $\mathbf{i = 1, 2, \ldots, 12}$, the weight of $\mathbf{s + A_i}$ is less than or equal to $\mathbf{(2)}$, then the error pattern is $\mathbf{u} = [\,\mathbf{s + A_i}\,, \mathbf{e_i}\,]$, where $\mathbf{e_i}$ is the $\mathbf{i^{th}}$ row of the 12×12 identity matrix.

Step 4. Compute the second syndrome $\mathbf{s_2 = s\,A}$.

Step 5. If the weight of $\mathbf{s_2}$ is less than or equal to $\mathbf{(3)}$, then the error pattern is:
$\mathbf{u} = [\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\quad\mathbf{0}\,,\mathbf{s_2}\,]$.

STEP 6. If the weight of $\mathbf{s_2}$ is greater than **(3)**, but for some $\mathbf{i = 1, 2, \ldots, 12}$, the weight of $\mathbf{s_2 + A_i}$ is less than or equal to **(2)**, then the error pattern is $\mathbf{u = [\, e_i \,, \, , s_2 + A_i \,]}$,.

STEP 7. If $\mathbf{u}$ is not yet determined then request retransmission.

The algorithm is explained with the use of MATLAB.
First we create the full cycle permutation matrix P:

```
>> echo on; clc
>> I10 = eye(10);  Z10 = zeros(1, 10);  P = [Z10  1; I10  Z10'];
P =
     0     0     0     0     0     0     0     0     0     0     1
     1     0     0     0     0     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     0     0     0
     0     0     1     0     0     0     0     0     0     0     0
     0     0     0     1     0     0     0     0     0     0     0
     0     0     0     0     1     0     0     0     0     0     0
     0     0     0     0     0     1     0     0     0     0     0
     0     0     0     0     0     0     1     0     0     0     0
     0     0     0     0     0     0     0     1     0     0     0
     0     0     0     0     0     0     0     0     1     0     0
     0     0     0     0     0     0     0     0     0     1     0
```

Using the vector $\mathbf{x = A_1}$, we define the matrices **A**, **G**, and **H**.

```
>>  x = [1   1   0   1   1   1   0   0   0   1   0];
>>  for i = 1:11,  A11(i,1:11) = [x * P^(i-1)];  end;
>>  J11 = ones(11,1);  A = [A11  J11; J11'  0]
A =
     1     1     0     1     1     1     0     0     0     1     0     1
     1     0     1     1     1     0     0     0     1     0     1     1
     0     1     1     1     0     0     0     1     0     1     1     1
     1     1     1     0     0     0     1     0     1     1     0     1
     1     1     0     0     0     1     0     1     1     0     1     1
     1     0     0     0     1     0     1     1     0     1     1     1
     0     0     0     1     0     1     1     0     1     1     1     1
     0     0     1     0     1     1     0     1     1     1     0     1
     0     1     0     1     1     0     1     1     1     0     0     1
     1     0     1     1     0     1     1     1     0     0     0     1
     0     1     1     0     1     1     1     0     0     0     1     1
     1     1     1     1     1     1     1     1     1     1     1     0
>>  I = eye(12);  G = [I A];  H = G';  J = ones(12, 1);
```

◇ **Examples.** The algorithm will be explained with the help of five different examples.

The following definitions, notations, and concepts will be used:

- The syndrome **s** of the received word **w** is obtained as **s = w * H** or **s = w * G'**.
- **ns = s * J** gives us the number of "ones" in **s**.
- **S = J * s** is the 12×12 matrix, where all rows are equal to **s**.
- To find **nr**, first we define 12×12 matrix **R = mod (S + A , 2)**, where its $\mathbf{i^{th}}$ row represents the weight of $\mathbf{s + A_i}$. Then **nr = R * J** .

- **s2 = s * A** gives us the second syndrome.

Example 1. the weight of **s** is less than or equal to **(3)**.

The received codeword:

>> **wa1** = [**1 0 1 1 1 1 1 0 1 1 1 1**]; **wa2** = [**0 1 0 0 1 0 0 1 0 0 1 0**];

>> **wa** = [**wa1**, **wa2**]

wa = *Columns* 1 *through* 12

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

Columns 13 *through* 24

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

STEP 1. Compute the syndrome:

>> **sa** = **mod**(**wa** * **H**, **2**)

sa = 1 0 0 0 0 0 0 0 0 0 0 1

STEP 2. Finding the weight of the syndrome:

>> **wta** = **sa** * **J**

wta = 2

STEP 3. Since the weight of **sa** is less than or equal to 3, the error pattern is:

>> **ua** = [**sa**, **zeros**(**1**, **12**)]

ua = *Columns* 1 *through* 12

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

Columns 13 *through* 24

0 0 0 0 0 0 0 0 0 0 0 0

>> **va** = **mod**(**wa** + **ua**, **2**)

va = *Columns* 1 *through* 12

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

Columns 13 *through* 24

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

The codeword sent was:

>> **va0** = **va**(**1** : **12**)

*va*0 = 0 0 1 1 1 1 1 0 1 1 1 0

Example 2. The weight of **s** is greater than **(3)**, but the weight of $\mathbf{s} + \mathbf{A_i}$ is less than or equal to **(2)**.

The received codeword:

>> **wb1** = [**0 0 1 0 0 1 0 0 1 1 0 1**]; **wb2** = [**1 0 1 0 0 0 1 0 1 0 0 0**];

>> **wb** = [**wb1**, **wb2**]

STEP 1. Compute the syndrome:

>> **sb** = **mod**(**wb** * **H**, **2**)

sb = 1 1 0 0 0 1 0 0 1 0 0 1

STEP 2. Finding the weight of the syndrome:

$>>$ $\mathbf{nb} = \mathbf{sb} * \mathbf{J}$

$nb =$ 5

STEP 3. Since the weight of **sb** is greater than **3**, we compute the matrix **Rb**.

$>>$ $\mathbf{Sb} = \mathbf{J} * \mathbf{sb}\,; \mathbf{Rb} = \mathbf{mod}\,(\mathbf{Sb} + \mathbf{A}, \mathbf{2})$

STEP 4. Find the weight of each row of **Rb**

$>>$ $\mathbf{nrb} = (\mathbf{Rb} * \mathbf{J})'$

$nrb =$ 4 6 8 4 2 8 6 6 6 6 6 8

STEP 5. The weight of the fifth row is less than or equal to **2**, so the error pattern is:

$>>$ $\mathbf{ub} = [\,\mathbf{mod}\,(\,\mathbf{Rb}\,(\,\mathbf{5}\,,:\,),\mathbf{2}\,),\mathbf{I}\,(\,\mathbf{5}\,,:\,)\,]$

$ub =$ *Columns* 1 *through* 12

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

Columns 13 *through* 24

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

$>>$ $\mathbf{vb} = \mathbf{mod}\,(\mathbf{wb} + \mathbf{ub}, \mathbf{2})$

$vb =$ *Columns* 1 *through* 12

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

Columns 13 *through* 24

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

The message sent was:

$>>$ $\mathbf{vb0} = \mathbf{vb}(\mathbf{1} : \mathbf{12})$

$vb0 =$ 0 0 1 0 0 1 0 1 1 1 1 1

Note. Since the algorithm is designed for **IMLD** and G corrects all the error patterns of weight at most **3**, only one row of **Rb** may have weight less than or equal to **3**.

Example 3. Computing the second syndrome $\mathbf{s_2} = \mathbf{s}\,\mathbf{A}$.

The received codeword:

$>>$ $\mathbf{wc1} = [\,\mathbf{1}\ \ \mathbf{1}\ \ \mathbf{1}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\,]\,;\ \ \mathbf{wc2} = [\,\mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{1}\ \ \mathbf{0}\ \ \mathbf{1}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{0}\ \ \mathbf{1}\ \ \mathbf{0}\ \ \mathbf{1}\,]$

$>>$ $\mathbf{wc} = [\,\mathbf{wc1}, \mathbf{wc2}\,]\,;$

STEP 1. Compute the syndrome:

$>>$ $\mathbf{sc} = \mathbf{mod}\,(\,\mathbf{wc} * \mathbf{H}\,,\,\mathbf{2}\,)$

$sc =$ 1 1 0 0 0 0 0 1 0 1 0 1

STEP 2. Find the weight of the syndrome:

$>>$ $\mathbf{nc} = \mathbf{sc} * \mathbf{J}$

$nc =$ 5

STEP 3. Since the weight of **sc** is greater than **3**, we compute the matrix **Rc**.

$>>$ $\mathbf{Sc} = \mathbf{J} * \mathbf{sc}; \mathbf{Rc} = \mathbf{mod}\,(\mathbf{Sc} + \mathbf{A}\,,\,\mathbf{2});$

STEP 4. Find the weight of each row of **Rc**.

$>>$ $\mathbf{nrc} = (\mathbf{Rc} * \mathbf{J})'$

$nrc =$ 4 8 4 4 4 4 8 6 6 6 8 8

STEP 5. All the rows of **Rc** have weight greater than **2**. We need to find the second syndrome:

$>>$ **sc2** = **mod** (**sc** $*$ **A**, **2**)

$sc2 =$ 0 0 0 0 0 0 0 1 1 1 0 0

STEP 6. Compute the weight of the second syndrome:

$>>$ **nc2** = **sc2** $*$ **J**

$nc2 =$ 3

STEP 7. The weight of **sc2** is less than or equal to **3**. The error pattern is:

$>>$ **uc** = [**zeros**(**1**, **12**), **sc2**]

Columns 1 *through* 12

0 0 0 0 0 0 0 0 0 0 0 0

Columns 13 *through* 24

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

$>>$ **vc** = **mod** (**wc** + **uc**, **2**)

$vc =$ *Columns* 1 *through* 12

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

Columns 13 *through* 24

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

The codeword sent was:

$>>$ **vc0** = **vc**(**1** : **12**)

$vc0 =$ 1 1 1 0 0 0 0 0 0 0 0 0

Example 4. There is a row **i** of **A** such that **s2** + $\mathbf{A_i}$ has a weight less than or equal to **2**. The received codeword:

$>>$ **wd1** = [**0 0 0 1 1 1 0 0 0 1 1 1**]; **wd2** = [**0 1 1 0 1 1 0 1 1 0 0 0**];

$>>$ **wd** = [**wd1 wd2**];

STEP 1. Compute the syndrome:

$>>$ **sd** = **mod** (**wd** $*$ **H** , **2**)

$sd =$ 1 0 1 1 0 1 1 0 1 0 1 0

STEP 2. Finding the weight of the syndrome:

$>>$ **nd** = **sd** $*$ **J**

$nd =$ 7

STEP 3. Since the weight of **sd** is greater than **3**, we compute the matrix **Rd**.

$>>$ **Sd** = **J** $*$ **sd**; **Rd** = **mod** (**Sd** + **A** , **2**)

STEP 4. We find the weight of each row of **Rd**

$>>$ **nrd** = (**Rd** $*$ **J**)′

$nrd =$ 8 4 8 6 6 8 4 8 8 4 6 4

STEP 5. All the rows of **Rd** have weight larger than **2**. We need to find the second syndrome:

>> **sd2** = **mod** (**sd** ∗ **A** , **2**)

*sd*2 = 1 1 1 0 0 1 1 1 1 1 0 1

STEP 6. Finding the weight of the second syndrome:

>> **nd2** = **sd2** ∗ **J**

*nd*2 = 9

STEP 7. The weight of **sd2** is larger than **3**, so we compute the matrix **Rd2**.

>> **S2** = **J** ∗ **sd2** ; **Rd2** = **mod** (**S2** + **A** , **2**) ; **nrd2** = (**Rd2** ∗ **J**)′

*nrd*2 = 6 8 6 2 4 6 6 4 6 4 6 4

STEP 8. Since the weight of the fourth row is less than or equal to **2**, the error pattern is:

>> **ud** = [**I** (**4** , :), **mod** (**Rd2** (**4** , :) , **2**)]

ud = *Columns* 1 *through* 12

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

Columns 13 *through* 24

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

>> **vd** = **mod** (**wd** + **ud**, **2**)

vd = *Columns* 1 *through* 12

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

Columns 13 *through* 24

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

The codeword sent was:

>> **vd0** = **vd**(**1** : **12**)

*vd*0 = 0 0 0 0 1 1 0 0 0 1 1 1

Example 5. Ask for retransmission.

The received codeword:

>> **we1** = [**1 1 1 1 1 1 0 0 0 0 0 0**] ; **we2** = [**1 1 1 0 0 0 0 1 1 0 0 0**] ;

>> **we** = [**we1** , **we2**] ;

STEP 1. Compute the syndrome:

>> **se** = **mod** (**we** ∗ **H** , **2**)

se = 1 0 0 0 1 0 0 1 0 0 1 0

STEP 2. Find the weight of the syndrome:

>> **ne** = **se** ∗ **J**

ne = 4

STEP 3. The weight of **se** is greater than **3**, so we compute the matrix **Re**.

>> **Se** = **J** ∗ **sd** ; **Re** = **mod** (**R** + **A** , **2**) ;

STEP 4. Find the weight of each row of **Re**.

>> **nre** = (**Re** ∗ **J**)′

nre = 7 5 7 9 5 3 9 7 7 7 7 7

STEP 5. All the rows of **Re** have weight greater than **2**. We need to find the second syndrome:

$>>$ **se2** = **mod** (**se** $*$ **A**, **2**)

$se2 =$ 0 1 0 1 1 0 1 0 0 0 0 0

STEP 6. Find the weight of the second syndrome:

$>>$ **ne2** = **se2** $*$ **J**

$ne2 =$ 4

STEP 7. The weight of **se2** is larger than **2**, so we compute the matrix **Re2**.

$>>$ **R** = **J** $*$ **se2**; **Re2** = **mod** (**R** + **A**, **2**); **nre2** = (**Re2** $*$ **J**)$'$

$nre2 =$ 5 7 7 7 9 7 7 9 3 7 5 7

STEP 8. All the rows of **Re2** have weight greater than **2**. This time we request retransmission.

Exercise. Decode each of the following received words that were encoded using G_{24}:

$wa = [0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \quad 1\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1]$

$wb = [0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \quad 1\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1]$

$wc = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0 \quad 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 1]$

$wd = [0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0 \quad 0\ 0\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 1]$

$we = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \quad 1\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1]$

♣ **Algoritm for Decoding Golay Code.** To decode a received word w, generated by a Golay code, we use the parity-check matrix of the extended code.

Let $w = [w_1, w_2, \ldots, w_{12}\ ,\ w_{13}, w_{14}, \ldots, w_{23}]$ be a received word.

Step 1.

(a_0) If the weight of w is odd then construct

$\widehat{w_0} = [w_1, w_2, \ldots, w_{12}\ ,\ w_{13}, w_{14}, \ldots, w_{23}, 0]$.

(a_1) If the weight of w is even then construct

$\widehat{w_1} = [w_1, w_2, \ldots, w_{12}\ ,\ w_{13}, w_{14}, \ldots, w_{23}, 1]$.

Step 2. Use the algorithm for the extended Golay code to decode $\widehat{w_0}$ or $\widehat{w_1}$.

Step 3. Remove the last digit of the decoded word.

Note. If w is a codeword, then the syndrome of $\widehat{w_0}$ or $\widehat{w_1}$ is the last row of the parity-check matrix H, (Why?)

Exercise. Decode each of the following received words that were encoded using G_{23}:

$wa = [1\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0 \quad 1\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1]$

$wb = [1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1 \quad 1\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 0]$

$wc = [1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0 \quad 1\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]$

$wd = [0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1]$

Bibliography

[1] D. R. Hankerson, D. G. Hoffman, D. A. Leonard, C. C. Lindner: Coding Theory and Cryptography – The Essentials, Second Edition, Revised and Expanded, Marcel Dekker Inc., 2000

[2] San Ling: Coding Theory: A First Course Paperback – First Edition, February 2004; ISBN-13: 978-0521529235 ISBN-10: 0521529239

[3] J.H. van Lint: Introduction to Coding Theory (Graduate Texts in Mathematics) – December 28, 1998

[4] Wade Trappe, Lawrence C. Washington: Introduction to Cryptography with Coding Theory (2nd Edition) – July 25, 2005