

Forward Error Correction and Pictures from Mars
David Garner, N6WY
N6WY@arrl.net

Abstract.

This paper illustrates how a sender can encode binary data words so that the data words can be sent over a weak noisy communication channel and a receiver can detect the correct data words. No abstract mathematical equations are used; just two numerical examples. The first example is a simple example. The second example is the encoding method used on Mariner 9 spacecraft to transmit pictures of the Martian surface as it orbited Mars. The Hadamard linear block code is used for both examples.

Introduction.

While doing research to reverse engineer the HF digital mode called Olivia, I developed two numerical examples (no strange mathematical symbols and abstract formulas.) that can explain to Radio Amateurs about linear Forward Error Correction (FEC) techniques for sending binary symbols over a radio link. Mathematicians and Communications Engineers may look at my examples with some disdain but my examples do illustrates how FEC works.

This paper covers the following:

- A short introduction of the various methods of Forward Error Correction for communications links.
- Background information about the Hadamard linear block code.
- A simple example of a “Covert Number Station” broadcasting random numbers.
- A non mathematical method for constructing a Hadamard matrix used for encoding and decoding Forward Error Correction Code Words.
- Background information and numerical examples for the code used on the Mariner 9 spacecraft. Mariner 9 mission was to orbit Mars and transmit pictures back to Earth. Forward Error Correction was used to send pictures of the Martian surface back to earth on a weak and noisy communication link.
- Bit Error Rate determination using Forward Error Correction.

Introduction to Forward Error Correction.

According to the BitcoinWiki, Forward Error Correction (FEC) is a technique used for controlling errors in data transmission over unreliable or noisy communication

channels. The central idea is that the sender encodes the message in a redundant way. The redundancy allows the receiver to detect a limited number of errors that may occur anywhere in the message. FEC gives the receiver the ability to correct errors without needing a reverse channel to request retransmission of data, but at the cost of a fixed, higher forward channel bandwidth. FEC is therefore applied in situations where retransmissions are costly or impossible, such as one-way communication links and when transmitting to multiple receivers in multicast. The maximum number of errors that can be corrected is determined by the design of the FEC code, so different forward error correcting codes are suitable for different conditions [1].

A simplistic example of FEC is to transmit each data bit 3 times, which is known as a (3,1) repetition code. This allows an error in any one of the three samples to be corrected by "majority vote". Though simple to implement and widely used, this triple modular redundancy is a relatively inefficient FEC.

The two main categories of FEC codes are block codes and convolutional codes.

There are several linear block codes such as Hamming codes, Reed-Solomon codes, Hadamard codes, Expander codes, Golay codes, and Reed-Muller codes.

In convolutional codes, the message comprises of data streams of arbitrary length and a sequence of output bits are generated by the sliding application of Boolean functions to the data stream. The convolutional codes can operate on a continuous string of data, whereas block codes operated on words. Convolutional codes also have memory—the behavior of the code depends on previous data [2].

Hadamard Code Background.

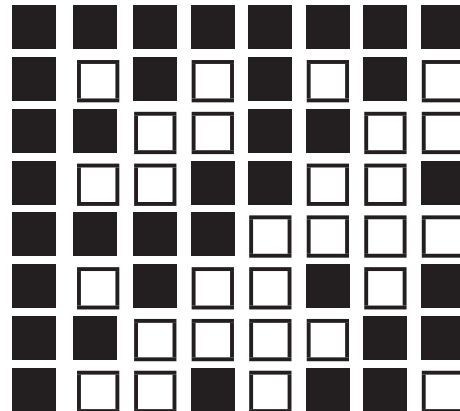
In this paper, I am limiting my discussion and examples to the Hadamard linear block code. The Hadamard code is also known under the names Walsh code, Walsh family, and Walsh–Hadamard code in recognition of the American mathematician Joseph Leonard Walsh [3]. Hadamard matrices are simple matrix structures and are used to generate the Hadamard code.

A Hadamard matrix is square, have entries +1 or -1 and have orthogonal row vectors and orthogonal column vectors. Figure 1 shows an 8 X 8 Hadamard matrix. Figure 2 is also an 8X8 Hadamard Matrix. The point I am trying to make is that you do not use the mathematical format for FEC.

Figure 1. 8X8 Hadamard Matrix

$$\begin{bmatrix}
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
 \hline
 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1
 \end{bmatrix}$$

Figure 2. Hadamard Matrix using Black and White Colors



Hadamard matrices have been actively studied by mathematicians for about a century and half and new uses are still being discovered. At the end of the referenced web page [3] is a list of practical applications for Hadamard matrices including an amateur radio protocol.

The Hadamard code is also used for code division multiple access (CDMA). In CDMA, the Hadamard code is referred to as Walsh Code, and is used to define individual communication channels. Each user will use a different Code Word, to modulate their signal. Because Walsh Code Words are mathematically orthogonal, a Walsh-encoded signal appears as random noise to a CDMA capable mobile terminal (cell phone), unless that terminal uses the same Code Word as the one used to encode the incoming signal [3].

Simple Example.

My simple example is not a very practical example because it corrects only one error. This example is of a covert number station that needs to broadcast what appears to be random numbers. This number station broadcasts only integers in the range of 0 to 7 using binary numbers. For this simple example, the broadcast station located in Cuba uses two different drum sounds in some music that starts the next segment of a radio program. A high drum sound is a one and a low sound is a drum zero.

Each Data Word contains three bits that is one number in the range of 0 through 7. To send eight different Data Words using the Hadamard Code, you need 8 different Code Words. An 8 X 8 Hadamard Matrix has 8 rows and each row is a Code Word. At this point in my simple example, I am not going to show how to make a Hadamard Matrix. Later, I will show a non mathematical way of creating a Hadamard Matrix of the size required for the Mariner 9 mission.

Since computers use 1's and 0's, I am going to change the -1's to 0's for the simple example Hadamard matrix as shown in Table 1. To the right of the matrix are the Data Words and to the left are the Data Values. Instead of sending 011 for the number 3, after Hadamard encoding the number 3 becomes 10011001.

Table 1. Encode Table		
Data Value	Hadamard matrix	Data Word
0	11111111	000
1	10101010	001
2	11001100	010
3	10011001	011
4	11110000	100
5	10100101	101
6	11000011	110
7	10010110	111

Table 2. Decode Table					
Data Value	Hadamard Matrix	RCW0 score	RCW1 score	RCW02 score	RCW3 score
0	11111111	4	3	2	3
1	10101010	4	5	4	3
2	11001100	4	5	4	5
3	10011001	8	7	6	5
4	11110000	4	3	4	3
5	10100101	4	5	6	7
6	11000011	4	5	6	5
7	10010110	4	3	4	5

As a side note, I did not need to assign the top row 0 and the bottom row 7. Any assignment would work as long as the receive stations know the assignment.

Using the Table 1 Encode Code Table, the covert number station is going to broadcast the Data Value 3, for four consecutive times. The Data Word that is

being encoded four times is 011 and the Transmitted Code Word that is being sent four times is 10011001.

The receive site has some interference and the agent in the field receives the four Code Words with some errors in the Code Words. With the help of Table 2, the field agent can decode the message. (Figure 3 defines the column headings used in Table 2.) The field agent needs four digits to identify the mission that are listed in the field agent mission book — the mission number book is not shown in this paper because it is Top Secret. The mission for this broadcast is number 3333.

Figure 3. Column Headings Definitions for Table 2

Send Data Value = 3 and Data Word = 011
Transmitted Data Code Word = 10011001
RCW_x = Received Code Word with x errors
RCW₀ = 10011001 (0 error.)
RCW₁ = 10001001 (1 error. red indicates received bit error)
RCW₂ = 10000001 (2 errors. red indicates received bit error)
RCW₃ = 10000101 (3 errors. red indicates received bit error)

The receive decode Hadamard matrix has the same values as the transmit encode Hadamard matrix. A simple scoring method identifies the Transmitted Code Word. The Received Code Word is given one point for each time it has the same bit value in the same position as a row in the Hadamard Matrix. The row with the highest score identifies the data value.

The first Received Code Word (RCW₀) has zero bit errors and has a perfect score for the Code Word that corresponds to the Data Value of 3. When a Received Code Word has a perfect match in the decode table all the other Received Code Words have a score that is equal to the distant value that a row in the Hadamard matrix has to all the other rows in the Hadamard matrix. The distant value is simply the number of bits that one row has that are different than any other row in the Hadamard Matrix. In standard coding theory notation for block codes, my simple example Hadamard code is a [8,3,4] where the 8 is block length (the number of bits in a Hadamard row), the 3 is the message length (the number of bits in the Data Word) and the 4 is the distance. The Mariner 9 code that is discussed later has a large block length compared to the message length that allows many bit errors in the Received Code Word.

The second Received Code Word (RCW1) has one bit error. My simple example using a Hadamard code [8,3,4] can only correct one bit error. The highest score in the decode table is for the Received Code Word for the Data Value of 3. Based on the score in the decode table, the field agent knows that the Code Word has one error and the Data Value is 3.

The third Received Code Word (RCW2) has two bit errors. Based on the score for RCW2 in the decode table, the field agent knows that RCW2 has two bit errors but does not know the Data Value. The field agent, who is in Oregon, checks the mission book and see that all missions that have the first three numbers of 335 and 336 are in Chicago. Using deduction, the field agent knows that the first three numbers in his mission are 333.

The fourth Received Code Word (RCW3) has three bit errors. The field agent does not know the RCW3 has three bit errors and using the scoring in the decode table the agent concludes the RCW3 has one bit error and he prepares to complete mission number 3335 instead of mission number 3333. FEC does have limitations.

How to construct a Hadamard Matrix.

A Hadamard matrix H of order n is an $n \times n$ matrix. My simple example used a Hadamard matrix of order 8 and is an 8×8 matrix. A Hadamard matrix can exist only if n is a multiple of 4. A 2×2 Hadamard matrix does exist and is a start point for building larger Hadamard matrices.

Figure 4 shows Hadamard matrices of order 2, 4, 8, and 16. The subscript on the H indicates the order of the Hadamard matrix. In the larger matrices, construction lines divide a matrix in 4 quadrants.

A Hadamard order 4 matrix can be made from the Hadamard order 2 matrix. To construct the Hadamard matrix of order 4, make a new Hadamard matrix of order 2 by changing the sign of every entry in the Hadamard order 2 matrix. Place that new Hadamard order 2 matrix in the lower right quadrant of the order 4 matrix. Copy the original order 2 matrix in the other 3 quadrants of the order 4 matrix. Similarly, a Hadamard order 8 matrix can be made from the Hadamard order 4 matrix and a Hadamard order 16 matrix can be made from the Hadamard order 8 matrix. A large Hadamard matrix can be made using this technique. There are easier ways to make Hadamard matrices using mathematics.

Figure 4. Hadamard Matrices Construction

$$\begin{aligned}
 H_2 &= \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\
 H_4 &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \\
 H_8 &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ \hline 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \\
 H_{16} &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \end{bmatrix} \\
 &\quad \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\ -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \end{bmatrix}
 \end{aligned}$$

Figure 2 is from reference [4]

Mariner 9 Background.

In 1971, the Mars Mariner 9 Spacecraft was the first man-made object to orbit a different planet and send back pictures of Mars that was 84 million miles from earth using a 20-watt transmitter. The pictures were black and white. Each Martian photograph required 4.5 million bits. The 4.5 million bits were split into 6 bit Data Words. Using 6 bits per Data Word, 64 different Data Words were created [5],[8].

Because of the weak signal from Mariner 9, the maximum useful data length of a Data Word with forward error correction was about 30 bits. The 5-repeat code was a possibility, having the advantage that it is very easy to implement, but it is only a capability of correcting 2 errors in an encoded word. To send a 6 bit Data Word with 5-repeat code becomes a 30 bits per Transmitted Code Word. (For sending a six bit Data Word like 010101, the data stream with 5-repeat code becomes 000001111100000111110000011111.) The Hadamard code was chosen for Mariner 9 because encoding a 6 bit Data Word into a 32 bit Transmitted Code Word gives a capability of correcting 7 errors at the received location [6].

A Hadamard order 5 matrix has 32 Code Words and has 32 bits per Code Word. The 32 bits per Code Word meets the Mariner 9 requirement but a total of 64 Code Words are required. A Hadamard order 6 matrix has 64 Code Words but the Mariner 9 communication link cannot support sending 64 bit Code Words. The solution is to use an augmented Hadamard code.

An augmented Hadamard code for Mariner 9 uses two different Hadamard order 5 matrices stacked together such that it becomes one matrix with 64 rows and 32 columns. The second matrix has 0's where the first matrix has 1's and the second matrix has 1's where the first matrix has 0's. All the rows in the augmented Hadamard code matrix of order n are different and will have the same distant value as the standard Hadamard code matrix of order n. In standard coding theory notation for block codes, the Mariner 9 augmented Hadamard code is [32,6,16] and the standard Hadamard code matrix of order 5 is [32,5,16]. Table 3 contains the augmented Hadamard code matrix that I am going to show FEC capability for Mariner 9. Table 3 also has the scores for Received Code Words. The Mariner 9 Code Words discussion is later in this paper.

Now before I proceed to show the decode capability of the [32,6,16], I need to comment about the design of Mariner 9. Processors and memory was not that great during the 1960s. The central processor for Mariner had a memory of 512 words [5]. Mariner 9 could not store 64, 32-bit Code Words. Mariner 9 did use a hardware design that was more economical in terms of speed, space, and weight. The hardware designed calculated a Transmitted Code Word from a Data Word rather than read it out of a stored array [7]. Amateur Radio HF modes that use Hadamard Code Words are not read out of a stored array but use a software program that calculates a Transmitted Code Word from each Data Word.

The earth receiving station for Mariner 9 required a fast decoding algorithm. The decision to use the FEC Hadamard code was based primarily on the decoding algorithm. A NASA engineer with the last name of Green designed what was called "The Green Machine". "The Green Machine" used a Fast Fourier Transform to calculate the Data Words from the Received Code Words [7].

Using Table Look-Up for Mariner 9 Code Words.

The earlier simple example had a small Code Word look-up table because the simple example had only 8 Data Words. The Mariner 9 has 64 Data Words and uses an augmented Hadamard code that requires two 32 X 32 Hadamard matrices. I copied the Mariner 9 Data Word to Code Word assignment from reference [8]. I added the score columns for two different Received Code Words for the table shown in Table 3. I had to use two pages for Table 3. Table 3a contains the Received Code Word assignments for the Data Words with value of 1 for the most significant bit. Table 3b contains the Received Code Word assignments for the Data Words with value of 0 for the most significant bit.

Figure 5 has the Column Headings Definitions for Table 3. Red bit values in the Received Code Words are bit errors. The values in the table that are blue indicates the Data Word with the most significant bit value of 1 and green indicates the Data Word with the most significant bit value of 0.

The Table 3 shows that the highest score indicates the correct Data Word for a Received Code Word that has less than 8 bit errors. If a Received Code Word has 8 or more bit errors, the correct Data Word cannot be determined. Every entry in a column does not need to be completed for determining the correct Received Code Word. A score of greater than 24 determines the correct Received Code Word. For example, if a cell in table 3 has a score of 25 then no cell in that column will have a score greater than or equal to 25.

Figure 5. Column Headings Definitions for Table 3

Send Data Words alternates between 100011 and 000011

Transmitted Code Words alternates between:

10101010010101011010101001010101
00000000111111110000000011111111

First Received Code Word is 1E0 and last is 0E8

1

1E0 = 0 errors for Received Code Word that has 1 as most significant bit

0

0E0 = 0 errors for Received Code Word that has 0 as most significant bit

The number above E indicates the value for the most significant bit and the number following E indicates the number of errors in the Received Code Word.

Received Code Words. Red bit values are bit errors.

1E0 = 10101010010101011010101001010101
0E0 = 00000000111111110000000011111111
1E1 = 10101010010101011010101001010100
0E2 = 00000000111111110000000011111100
1E3 = 10101010010101011010101001010010
0E4 = 00000000111111110000000011110000
1E5 = 10101010010101011010101001001010
0E6 = 00000000111111110000000011000000
1E7 = 10101010010101011010101000101010
0E8 = 00000000111111110000000000000000

Note: The position of any bit error does not affect the score

Table 99a. Score for Received Code Words

Data Word	Code Word	1	0	1	0	1	0	1	0	1	0
		E0	E0	E1	E2	E3	E4	E5	E6	E7	E8
111111	11111111111111111111111111111111	16	16	15	14	15	12	15	10	15	8
100000	10101010101010101010101010101010	16	16	17	16	19	16	21	16	23	16
110000	11001100110011001100110011001100	16	16	17	18	15	16	17	18	15	16
101111	10011001100110011001100110011001	16	16	15	16	15	16	15	16	15	16
111000	11110000111100001111000011110000	16	16	17	18	17	20	15	18	15	16
100111	10100101101001011010010110100101	16	16	15	16	13	16	13	16	15	16
110111	11000011110000111100001111000011	16	16	15	14	17	16	17	18	15	16
101000	10010110100101101001011010010110	16	16	17	16	17	16	15	16	15	16
111100	11111111000000001111111100000000	16	0	17	2	17	4	17	6	17	8
100011	10101010010101011010101001010101	32	16	31	16	29	16	27	16	25	16
110011	11001100001100111100110000110011	16	16	15	14	17	16	15	14	17	16
101100	10011001011001101001100101100110	16	16	17	16	17	16	17	16	17	16
111011	11110000000011111111000000001111	16	16	15	14	15	12	17	14	17	16
100100	10100101010110101010010101011010	16	16	17	16	19	16	19	16	17	16
110100	11000011001111001100001100111100	16	16	17	18	15	16	15	14	17	16
101011	10010110011010011001011001101001	16	16	15	16	15	16	17	16	17	16
111110	11111111111111110000000000000000	16	16	17	18	17	20	17	22	17	24
100001	10101010101010100101010101010101	16	16	15	16	13	16	11	16	9	16
110001	11001100110011000011001100110011	16	16	15	14	17	16	15	14	17	16
101110	10011001100110010110011001100110	16	16	17	16	17	16	17	16	17	16
111001	11110000111100000000111100001111	16	16	15	14	15	12	17	14	17	16
100110	10100101101001010101101001011010	16	16	17	16	19	16	19	16	17	16
110110	11000011110000110011110000111100	16	16	17	18	15	16	15	14	17	16
101001	10010110100101100110100101101001	16	16	15	16	15	16	17	16	17	16
111101	11111111000000000000000011111111	16	16	15	14	15	12	15	10	15	8
100010	10101010010101010101010110101010	16	16	17	16	19	16	21	16	23	16
110010	11001100001100110011001111001100	16	16	17	18	15	16	17	18	15	16
101101	10011001011001100110011010011001	16	16	15	16	15	16	15	16	15	16
111010	11110000000011110000111111110000	16	16	17	18	17	20	15	18	15	16
100101	10100101010110100101101010100101	16	16	15	16	13	16	13	16	15	16
110101	11000011001111000011110011000011	16	16	15	14	17	16	17	18	15	16
101010	10010110011010010110100110010110	16	16	17	16	17	16	15	16	15	16

Table 99b. Score for Received Code Words

Data Word	Code Word	1 E0	0 E0	1 E1	0 E2	1 E3	0 E4	1 E5	0 E6	1 E7	0 E8
010101	01101001100101101001011001101001	16	16	15	16	15	16	17	16	17	16
001010	00111100110000111100001100111100	16	16	17	18	15	16	15	14	17	16
011010	01011010101001011010010101011010	16	16	17	16	19	16	19	16	17	16
000101	00001111111100001111000000001111	16	16	15	14	15	12	17	14	17	16
010010	01100110100110011001100101100110	16	16	17	16	17	16	17	16	17	16
001101	00110011110011001100110000110011	16	16	15	14	17	16	15	14	17	16
011101	01010101101010101010101001010101	16	16	15	16	13	16	11	16	9	16
000010	000000001111111111111100000000	16	16	17	18	17	20	17	22	17	24
010110	01101001011010011001011010010110	16	16	17	16	17	16	15	16	15	16
001001	00111100001111001100001111000011	16	16	15	14	17	16	17	18	15	16
011001	01011010010110101010010110100101	16	16	15	16	13	16	13	16	15	16
000110	00001111000011111111000011110000	16	16	17	18	17	20	15	18	15	16
010001	01100110011001101001100110011001	16	16	15	16	15	16	15	16	15	16
001110	00110011001100111100110011001100	16	16	17	18	15	16	17	18	15	16
011110	01010101010101011010101010101010	16	16	17	16	19	16	21	16	23	16
000001	00000000000000001111111111111111	16	16	15	14	15	12	15	10	15	8
010100	01101001100101100110100110010110	16	16	17	16	17	16	15	16	15	16
001011	00111100110000110011110011000011	16	16	15	14	17	16	17	18	15	16
011011	01011010101001010101101010100101	16	16	15	16	13	16	13	16	15	16
000100	00001111111100000000111111110000	16	16	17	18	17	20	15	18	15	16
010011	01100110100110010110011010011001	16	16	15	16	15	16	15	16	15	16
001100	00110011110011000011001111001100	16	16	17	18	15	16	17	18	15	16
011110	01010101101010100101010110101010	0	16	1	16	3	16	5	16	7	16
000011	00000000111111110000000011111111	16	32	15	30	15	28	15	26	15	24
010111	01101001011010010110100101101001	16	16	15	16	15	16	17	16	17	16
001000	00111100001111000011110000111100	16	16	17	18	15	16	15	14	17	16
011000	01011010010110100101101001011010	16	16	17	16	19	16	19	16	17	16
000111	00001111000011110000111100001111	16	16	15	14	15	12	17	14	17	16
010000	01100110011001100110011001100110	16	16	17	16	17	16	17	16	17	16
001111	00110011001100110011001100110011	16	16	15	14	17	16	15	14	17	16
011111	01010101010101010101010101010101	16	16	15	16	13	16	11	16	9	16
000000	00000000000000000000000000000000	16	16	17	18	17	20	17	22	17	24

Bit Error Rate (BER).

The average Bit Error Rate (BER) can be calculated for FEC linear block codes by using the average number of bit correction required for each Received Code Word. For the table look-up method, the number of errors for the Received Code Word is the difference between the perfect score and the actual score. For the words in Table 3 that have the high score, there are 10, 32 bit Received Code Words and the average BER is:

$$\text{Average BER} = (0+0+1+2+3+4+5+6+7+8)/(10*32) = 36/320 = .11 = 11\%$$

I used the word with 8 errors because the Mariner 9 FEC can detect 8 bit errors but it can correct only 7 bit errors.

Summary.

The table look-up method is generally not used for Hadamard FEC encoding and decoding Data Words. Encoding and decoding can be quickly done using math and a small computer program. The Mariner 9 augmented Hadamard code is [32,6,16], where the 32 is block length (the number of bits in a Hadamard row), the 6 is the message length (the number of bits in the Data Word) and the 16 is the distance. For FEC linear block codes, the maximum number of errors that can be corrected is determined by half the distance minus 1. The maximum number of errors that can be detected is half the distance.

References

1. BitcoinWiki, 'Forward error correction', https://en.bitcoinwiki.org/wiki/Forward_error_correction, 2017.
2. BitcoinWiki, 'Convolutional code', https://en.bitcoinwiki.org/wiki/Convolutional_code#Popular_convolutional_codes, 2017.
3. Wikipedia, 'Hadamard code', https://en.wikipedia.org/wiki/Hadamard_code, 2020
4. math.uic.edu, 'Hadamard Matrices and Hadamard Codes', http://homepages.math.uic.edu/~leon/mcs425-s08/handouts/Hadamard_codes.pdf, date unknown
5. MIT, '6.02 Fall 2012 Lecture 6', https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-02-introduction-to-eecs-ii-digital-communication-systems-fall-2012/lecture-slides/MIT6_02F12_lec06.pdf, 2012
6. Malek, 'Coding Theory', https://nanopdf.com/download/coding-theory-hadamard-codes-california-state-university-east_pdf, 2015
7. UC Denver, 'Combinatorics in Space', <http://www-math.ucdenver.edu/~wcherowi/courses/m7409/mariner9talk.pdf>, date unknown
8. Anderson, University of Glasgow, 'ERROR – CORRECTING CODES' https://www.gla.ac.uk/media/Media_293296_smx.pdf, 2013