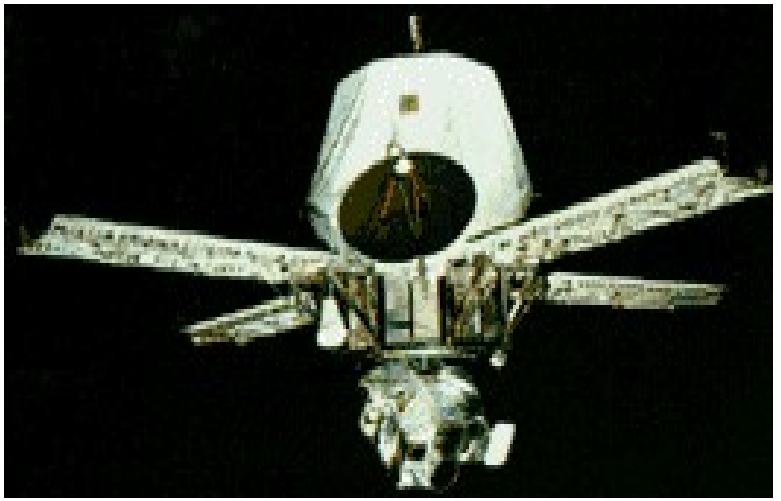
Combinatorics in Space

The Mariner 9 Telemetry System

Mariner 9 Mission



Launched: May 30, 1971
Arrived: Nov. 14, 1971
Turned Off: Oct. 27, 1972

Mission Objectives:
(Mariner 8): Map 70% of Martian surface.
(Mariner 9): Study temporal changes in Martian atmosphere and surface features.

Live TV

A black and white TV camera was used to broadcast “live” pictures of the Martian surface.

Each photo-receptor in the camera measures the brightness of a section of the Martian surface about 4-5 km square, and outputs a grayness value in the range 0-63. This value is represented as a binary 6-tuple.

The TV image is thus digitalized by the photo-receptor bank and is output as a stream of thousands of binary 6-tuples.

The Problem

The problem that arises comes from the difficulties inherent with the transmission of messages. More precisely, we wish to transmit a message and know that in the process of transmission there will be some altering of the message, due to weak signals, sporadic electrical bursts and other naturally occurring noise that creeps into the transmission medium. The problem is to insure that the intended message (our original transmission) is obtainable from whatever is actually received.

The Repeat Code

One simple approach to this problem is what is called a repeat code. For instance, if we wanted to send the message BAD NEWS, we could repeat each letter a certain number of times and send, say,

BBBBBAAAAADDDDD NNNNNEEEEEWWWWWSSSSS.

Even if a number of these letters got garbled in transmission, the intended message could be recovered from a received message that might look like

BBBEBFAAAADGDDD . MNNNTEEEEEWWWSWRRSSS,

by a process called ***majority decoding***, which in this case would mean that for each block of 5 letters the intended letter is the one which appears most frequently in the block.

Probability

The problem with this approach is economical, the repeat code is not very efficient. The increased length of the transmitted code, and thus the increased time and energy required to transmit it, is necessary in order to be able to decode the message properly, but how efficiently a coding procedure uses this increase depends upon the coding scheme. Suppose, in our example, that the probability that a letter is garbled in transmission is $p = 0.05$ and so $q = 1 - p = 0.95$ is the probability that a letter is correctly received. Without any coding, the probability of our 8 letter (spaces included) message being correctly received is

$$q^8 = (0.95)^8 = 0.66.$$

(In this calculation we are assuming that the error in transmitting a single symbol is independent of which position the symbol is in. This is a common simplifying assumption ... which may not be appropriate in real world situations.)

Probability

Using the repeat code, the probability of correctly decoding a given letter from a block of 5 symbols is

$$q^5 + 5q^4p + 10q^3p^2$$

since there are three ways to decode correctly: 1) all the symbols are correct, 2) one symbol is incorrect (5 ways this can happen) or 3) two symbols are incorrect (10 ways this can happen) [notice that these are just terms in the expansion of $(q+p)^5$]. So we obtain

$$(.95)^5 + 5(.95)^4(.05) + 10(.95)^3(.05)^2 = 0.9988$$

and thus the probability of getting the correct eight letter message after decoding is $(0.9988)^8 = 0.990$, clearly a great increase over the non-coded message (= 0.66), but this 1% probability of getting the wrong message might not be acceptable for certain applications.

Terminology

To increase the probability of decoding the correct message with this type of code we would have to increase the number of repeats - a fix which may not be desirable or even possible in certain situations. However, as we shall see, other coding schemes could increase the probability to 0.9999 without increasing the length of the coded message.

Before leaving the repeat codes to look at other coding schemes, let us introduce some terminology. Each block of repeated symbols is called a ***code word***, i.e., a code word is what is transmitted in place of one piece of information in the original message. The set of all code words is called a ***code***. If all the code words in a code have the same length, then the code is called a ***block code***. The repeat codes are block codes.

Detection and Correction

One feature that a useful code must have is the ability to detect errors. The repeat code with code words having length 5 can always detect from 1 to 4 errors made in the transmission of a code word, since any 5 letter word composed of more than one letter is not a code word. However, it is possible for 5 errors to go undetected (how?). We would say that this code is ***4-error detecting***. Another feature is the ability to correct errors, i.e., being able to decode the correct information from the error riddled received words. The repeat code we are dealing with can always correct 1 or 2 errors, but may decode a word with 3 or more errors incorrectly (how?), so it is a ***2-error correcting code***.

Coding Needed

In the Mariner 9 mission, without coding and a failure probability $p = 0.05$, 26% of the image would be in error ... unacceptably poor quality for TV transmission.

Any coding will increase the length of the transmitted message. Due to power constraints on board the probe and equipment constraints at the receiving stations on Earth, the coded message could not be much more than 5 times as long as the data.

Thus, a 6-tuple of data could be coded as a codeword of about 30 bits in length.

Other concerns

A second concern involves the coding procedure. Storage of data requires shielding of the storage media – this is dead weight aboard the probe and economics require that there be little dead weight. Coding should therefore be done “on the fly”, without permanent memory requirements.

Finally, decoding needs to be done rapidly. The Jet Propulsion Laboratory in Pasadena, California will process the signals and reconvert them to picture images for the press which will be gathered at JPL.

Besides this NASA priority, rapid decoding is needed so that feedback to the probe becomes viable – redirecting the camera based on what is seen.

The Code

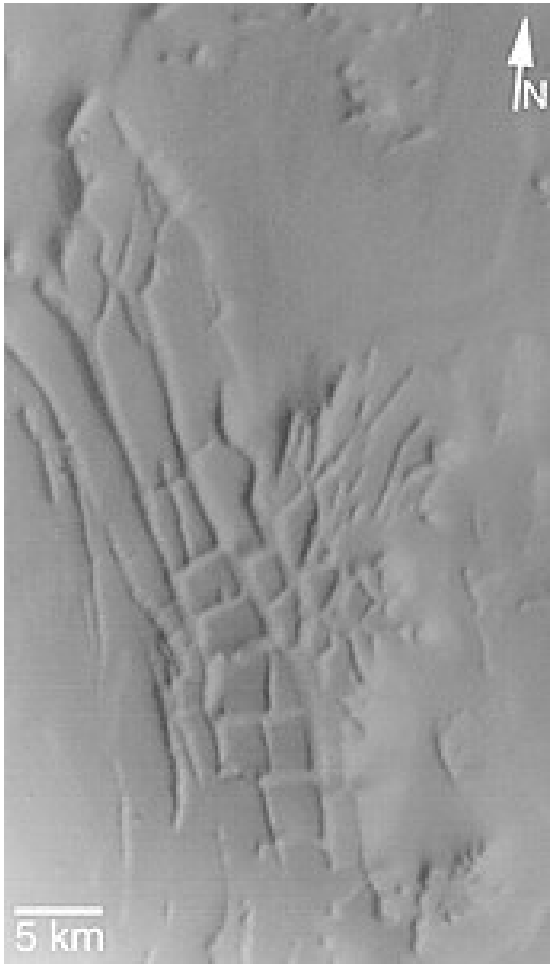
The 5-repeat code would satisfy the mission specs, but it is only 2-error correcting ... leaving 1% of the image in error.

The actual code selected is 7-error correcting and this reduced the probability of error in the image to only 0.01%.

The decision on which code to use was based primarily on the decoding algorithm. The algorithm was carried out by a fairly simple piece of specialized circuitry called “The Green Machine.”

The code selected was the (32,6)-Biorthogonal Reed-Muller Code. We will go through the construction and use of this code after looking at some of the results.

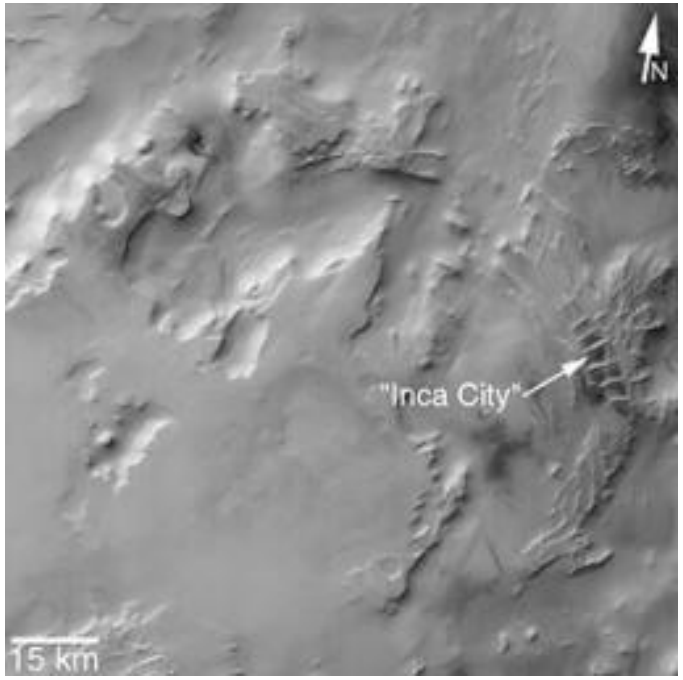
Results



Inca City:
-80 Lat., 64 Long.

"Inca City" is the informal name given by Mariner 9 scientists in 1972 to a set of intersecting, rectilinear ridges that are located among the layered materials of the south polar region of Mars. Their origin has never been understood; most investigators thought they might be sand dunes, either modern dunes or, more likely, dunes that were buried, hardened, then exhumed. Others considered them to be dikes formed by injection of molten rock (magma) or soft sediment into subsurface cracks that subsequently hardened and then were exposed at the surface by wind erosion.

Inca City

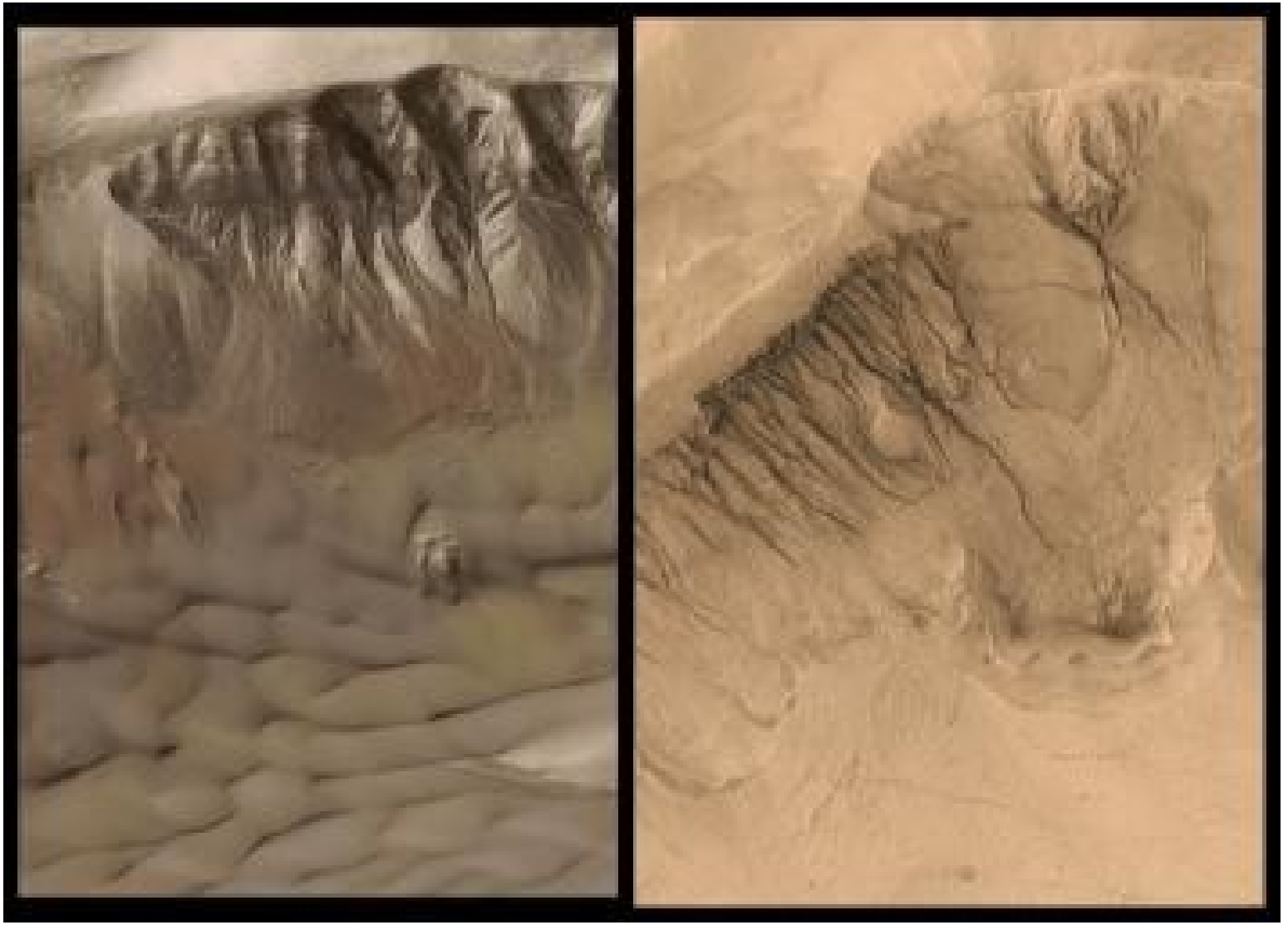


The Mars Global Surveyor (MGS) Mars Orbiter Camera (MOC) has provided new information about the "Inca City" ridges, though the camera's images still do not solve the mystery. The new information comes in the form of a MOC red wide angle context frame taken in mid-southern spring. The MOC image shows that the "Inca City" ridges, located at 82°S, 67°W, are part of a larger circular structure that is about 86 km (53 mi) across.

Inca City

It is possible that this pattern reflects an origin related to an ancient, eroded meteor impact crater that was filled-in, buried, then partially exhumed. In this case, the ridges might be the remains of filled-in fractures in the bedrock into which the crater formed, or filled-in cracks within the material that filled the crater. Or both explanations could be wrong. While the new MOC image shows that "Inca City" has a larger context as part of a circular form, it does not reveal the exact origin of these striking and unusual martian landforms.

Inca City



Hadamard Matrices

An $n \times n$ matrix $H = h_{ij}$ is an **Hadamard matrix of order n** if the entries of H are either +1 or -1 and such that $HH^T = nI$, where H^T is the transpose of H and I is the order n identity matrix. Put another way, a (+1,-1)-matrix is Hadamard if the inner product of two distinct rows is 0 and the inner product of a row with itself is n.

A few examples of Hadamard matrices are;

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

Hadamard Matrices

These matrices were first considered as Hadamard determinants. They were so named because the determinant of an Hadamard matrix satisfies equality in *Hadamard's determinant theorem*, which states that if $X = x_{ij}$ is a matrix of order n where $| x_{ij} | \leq 1$ for all i and j, then

$$| \det X | \leq n^{n/2}$$

It is apparent that if the rows and columns of an Hadamard matrix are permuted, the matrix remains Hadamard. It is also true that if any row or column is multiplied by -1, the Hadamard property is retained.[Prove this] Thus, it is always possible to arrange to have the first row and first column of an Hadamard matrix contain only +1 entries. An Hadamard matrix in this form is said to be ***normalized.***

Order of Hadamard Matrices

Theorem: *The order of an Hadamard matrix is 1,2 or 4n, n an integer.*

Proof: [1] is an Hadamard matrix of order 1 and the first example above is an Hadamard matrix of order 2. Suppose now that H is an Hadamard matrix of order h > 2. Normalize H and rearrange the first three rows to look like:

1 1	1 1	1 1	1 1
1 1	1 1	-1 .....-1	-1 .....-1
1 1	-1 .....-1	1 1	-1 .....-1
x	y	z	w

Where x,y,z,w are the numbers of columns of each type. As the order is h,

$$x + y + z + w = h$$

and taking the inner products of rows 1 and 2, 1 and 3, and, 2 and 3 we get

$$x + y - z - w = 0$$
$$x - y + z - w = 0$$
$$x - y - z + w = 0.$$

Solving this system of equations gives,

$$x = y = z = w = h/4.$$

Thus, the integer h must be divisible by 4.

Properties of Hadamard Matrices

Corollary: *If H is a normalized Hadamard matrix of order 4n, then every row (column) except the first has 2n minus ones and 2n plus ones, further n minus ones in any row (column) overlap with n minus ones in each other row (column).*

Proof: This is a direct result of the above proof since any two rows other than the first can take the place of the second and third rows in the proof. The same argument can be applied to the columns.

Hadamard matrices are known for many of the possible orders, the smallest order for which the existence of an Hadamard matrix is in doubt is currently 668 (A solution for the previous unknown case of 428 was announced by Kharaghani and Tayfeh-Rezaie in June 2004).

Construction of Hadamard Matrices

While there are a great many construction methods for Hadamard matrices, we will only consider one of the simplest, the direct product construction.

Construction - *Given Hadamard matrices H_1 of order n and H_2 of order m the direct product of these two matrices, represented by:*

$$\begin{array}{cccc} h_{11}H_2 & h_{12}H_2 & \cdots & h_{1n}H_2 \\ h_{21}H_2 & h_{22}H_2 & \cdots & h_{2n}H_2 \\ \cdots & \cdots & \cdots & \cdots \\ h_{n1}H_2 & h_{n2}H_2 & \cdots & h_{nn}H_2 \end{array}$$

where $H = |h_{ij}|$, is an Hadamard matrix of order nm.

Example

Let $H_1 = H_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}$.

The construction gives:

$$H_1 * H_2 = \begin{bmatrix} +H_2 & +H_2 \\ +H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}.$$

Back to Mariner 9

Recall that in the Mariner 9 mission, the data consisted of binary 6-tuples (64 grayness levels) and transmission restrictions permitted coding that would lengthen the transmitted words to about 30 bits.

The 5-repeat code would satisfy this condition, but it is only 2 error correcting.

The code chosen was a Reed-Muller code. The code words are 32 bits long and there are 64 of them. The code words are the rows of a 32×32 Hadamard matrix constructed by repeating our example construction 3 more times, and their negatives.

Error Correction

The ability of a code to correct errors is directly related to the "distance" between code words. We must make this concept precise.

The block codes we will talk about will be subsets of the set of all n-tuples whose coordinates come from an alphabet of size k. We will denote this large set by ***V(n,k)*** and will often think of it as a "vector space", but this is not accurate unless the alphabet is a field. In this more general setting, we do not do algebraic operations with our "vectors" since the alphabet need not have any algebraic properties. When we are being careful, we will use the term ***word*** instead of "vector".

Hamming Distance

The ***Hamming distance*** between two words in V(n,k) is the number of places in which they differ.

So, for example, the words (0,0,1,1,1,0) and (1,0,1,1,0,0) would have a Hamming distance of 2, since they differ only in the 1st and 5th positions. In V(4,4), the words (0,1,2,3) and (1,1,2,2) also have distance 2.

This Hamming distance is a ***metric*** on V(n,k), i.e., if d(x,y) denotes the Hamming distance between words x and y, then d satisfies:

1) $d(x,x) = 0$
2) $d(x,y) = d(y,x)$, and
3) $d(x,y) + d(y,z) \geq d(x,z)$. (triangle inequality)

Hamming Distance

The first two of these properties are obvious, but the triangle inequality requires a little argument (this is a homework problem).

Since we will only deal with the Hamming distance (there are other metrics used in Coding Theory), we will generally omit the Hamming modifier and talk about the ***distance*** between words.

Minimum Distance

The ***minimum distance*** of a code C is the smallest distance between any pair of distinct codewords in C. It is the minimum distance of a code that measures a code's error correcting capabilities. If the minimum distance of a code C is 2e + 1, then C is a 2e-error detecting code since 2e or fewer errors in a codeword will not get to another codeword and is an e-error correcting code, since if e or fewer errors are made in a codeword, the resulting word is closer to the original codeword than it is to any other codeword and so can be correctly decoded (***maximum-likelihood decoding***).

In the 5-repeat code of V(5,4) (codewords: 00000, 11111, 22222, and 33333) the minimum distance is 5. The code detects 4 or fewer errors and corrects 2 or fewer errors as we have seen.

Our Reed-Muller Code

From the properties of a Hadamard matrix, we can see that two different rows from the matrix will differ in exactly half of their positions (since the dot product will be 0). If we take one of the two rows and negate all its elements, the dot product will not change, so again the two rows will differ in exactly half of their elements. Finally, a row and its negation will differ in all positions. Thus, the minimum distance between these rows is half the length of the rows.

In our case, the code words are the rows of two 32 x 32 Hadamard matrices (one the negation of the other) so the minimum distance is 16, which makes this a 7-error correcting code.

Encoding

Encoding:

As there are 64 code words and 64 data types, any assignment of code word to data type will work, but the requirement that the encoding should require no memory meant that an arbitrary assignment would not do.

Using these special Hadamard matrices makes the code a 6 dimensional vector space, so there is a basis with 6 elements (any linear combination of which gives a code word). The data type 6-tuple is used to provide the coefficients for the linear combination of the basis vectors ... thus associating a unique code word to each data type.

This simple computation can be hard wired and requires no memory.

Decoding

Decoding:

As we have previously mentioned, the real reason for selecting this code was that it had a very fast decoding algorithm which we now describe.

First, convert all the code words (and the received vector) to ± 1 vectors by turning the 0's into -1's. Take the dot product of the received vector with each of the code words in turn. As soon as the result is 16 or greater, decode as that code word.

Suppose no errors have been made in transmission. Then the dot product of the received vector with itself will be 32 and with any other codeword will be 0 or -32.

Decoding

Decoding:

This follows since the distance between two code words is the weight of their difference (which is another code word) and so is either 0, 16 or 32. If 0, the code words are the same. If 32, the code words have no common component and the dot product of the ±1 form will be -32. In all remaining cases, 16 places are the same and 16 places are different, giving a dot product of 16 – 16 = 0.

For each error that occurs, the dot product will decrease by 2 (or increase by 2 from an incorrect codeword). If no more than 7 errors occur, the dot product with the correct code word decreases to at least 18 and the dot product with incorrect code words increases to at most 14 ... so correct decoding will occur. If 8 or more errors occur, there will be dot products of at least 16 but correct decoding is not possible.

The Green Machine

Decoding:

Even though this is a rapid decoding algorithm, the computations involved can be speeded up by a factor of 3 by using a Fast Fourier Transform for Abelian groups. This is what was actually done by the “green machine”.

Other Missions

The Voyager 1 & 2 spacecraft transmitted color pictures of Jupiter and Saturn in 1979 and 1980. Color transmission requires 3 times the amount of data, so a different code (the Golay (24,12,8) code) was used. It is only 3-error correcting, but its transmission rate is much higher. Voyager 2 went on to Uranus and Neptune and the code was switched to a Reed-Solomon code for its higher error correcting capabilities.