TDA Progress Report 42-113

May 15, 1993

Uncorrectable Sequences and Telecommand

L. Ekroot, R. McEliece,[1] S. Dolinar, and L. Swanson
Communications Systems Research Section

The purpose of a tail sequence for command link transmission units is to fail to decode, so that the command decoder will begin searching for the start of the next unit. A tail sequence used by several missions and recommended for this purpose by the Consultative Committee on Space Data Standards is analyzed. A single channel error can cause the sequence to decode. An alternative sequence requiring at least two channel errors before it can possibly decode is presented. (No sequence requiring more than two channel errors before it can possibly decode exists for this code.)

I. Introduction

When a *command link transmission unit* (CLTU) consisting of many *codeblocks* is received by a spacecraft, the command decoder verifies that each codeblock is a valid codeword and accepts it, or that it is a slightly corrupted codeword and corrects it, or that it is too far from a valid codeword and rejects it. Rejecting a codeblock causes the receiver to give up on the unit and begin searching for the start of the next unit. At the end of the CLTU, there is a *tail sequence* designed to be rejected as a codeword, sending the decoder into a "search mode." This article analyzes the performance of the tail sequence recommended by the Consultative Committee on Space Data Standards (CCSDS) and used by several missions. So instead of the usual question about a code, i.e., how many errors can the code correct or detect, the question here is how many errors can occur before an uncorrectable sequence becomes correctable.

II. Analysis of Uncorrectable Sequences

In order for a sequence to be uncorrectable, it must be far enough from a codeword to cause the decoder to not decode. At the very least, it must differ from the nearest codeword in more positions than the decoder is able to correct. However, channel errors can make such a sequence decodable. The more errors that must occur before the sequence becomes correctable, the less likely it is that the sequence will accidentally decode. In order to maximize the number of channel errors before the sequence will decode, it is necessary to characterize and find sequences that are as far away from codewords as possible.

[1] Consultant, California Institute of Technology, Engineering Department.

A code is designed to have codewords that are maximally far away from each other. If, as a simple example, a new code is created by using a subset of the codewords, then the unused codewords are still far away from the codewords of the new code. Intuitively, these unused codewords are candidates for uncorrectable sequences.

The codes discussed in this article include the perfect (63,57) Hamming code, the (63,56) expurgated Hamming code, and shortened versions of the expurgated code. The codewords for the shortened codes correspond to subsets of the codewords of the (63,56) code, which are themselves the even codewords from the (63,57) code. By making use of the larger code's properties and keeping track of what happens as the codeword sets get smaller, sequences that are uncorrectable in each of the smaller codes can be found.

A. The Perfect (63,57) Hamming Code

The generator polynomial for the (63,57) code is the sixth-degree primitive polynomial

$$g_p(x) = x^6 + x + 1 \qquad (1)$$

The (63,57) code is perfect and has minimum distance 3, i.e., every binary sequence of length 63 is either a codeword or is Hamming distance one away from exactly one codeword. For a perfect code, there are no holes left when the space of binary sequences is filled with balls of radius one centered at the codewords.

The even-weight words are a promising subset to use for a new code, leaving the odd-weight words as candidates for uncorrectable sequences. In fact the (63,56) code described in the next section uses the even-weight words.

B. The (63,56) Expurgated Hamming Code

The generator polynomial for the (63,56) code is given by

$$g(x) = x^7 + x^6 + x^2 + 1 = (x+1)(x^6 + x + 1) \qquad (2)$$

Since the generator polynomial is the product of $x + 1$ and the generator polynomial $g_p(x)$ of the perfect (63,57) code, the (63,56) code consists of only the even-weight codewords from the perfect (63,57) code. This code has minimum distance four and can correct at most one error.

Note that any odd-weight codeword in the (63,57) code is exactly Hamming distance three away from the nearest even-weight codewords, and the even-weight codewords are the codewords of the (63,56) code. Any odd-weight binary sequence of length 63 differs from a nearest codeword by either three bits or one bit. Similarly, any even-weight sequence is either a codeword or is Hamming distance two from the nearest codewords.

The following example illustrates how the concepts of distance relate to a sequence that is not a codeword in the perfect (63,57) code. The sequence selected for the example plays a role in the CCSDS recommendations for telecommand; the description of that role is deferred to Section III.

Example 1: The length 63 sequence

01010101010101010101010101010101010101010101010101010101 1010101

has even weight, and is therefore either a codeword in the (63,56) code or two away from a codeword. The two-part *syndrome* $\begin{pmatrix} s_1(\mathbf{r}) \\ s_2(\mathbf{r}) \end{pmatrix}$ of a sequence $\mathbf{r} = r_{N-1}r_{N-2}\cdots r_1r_0$ is given by

$$\begin{pmatrix} s_1(\mathbf{r}) \\ s_2(\mathbf{r}) \end{pmatrix} \equiv \begin{pmatrix} \sum_{i=0}^{N-1} r_i\alpha^i \pmod{\alpha^6 + \alpha + 1} \\ \sum_{i=0}^{N-1} r_i\alpha^i \pmod{\alpha + 1} \end{pmatrix} \qquad (3)$$

where α is a root of the generator polynomial $g(x)$, and N is the length of the codewords. The two-part syndrome tells if the sequence is a codeword, how to correct it if it is distance one from a codeword, or that it is not near enough to a single codeword. Specifically, if the syndrome is $\binom{0}{0}$, then the sequence is a codeword; if the syndrome is $\binom{0}{1}$, then the sequence has odd weight and differs from the nearest codeword in three positions; if the syndrome is $\binom{\alpha^j}{1}$, the sequence has odd weight and differs from the nearest codeword in the jth position; if the syndrome is $\binom{\alpha^j}{0}$, the weight is even and the sequence differs from the nearest codewords in two positions.

For this example sequence, the top part of the syndrome (modulo $\alpha^6 + \alpha + 1$) is

$$s_1(\mathbf{r}) \equiv \sum_{i=0}^{N-1} r_i \alpha^i \pmod{\alpha^6 + \alpha + 1} \tag{4}$$

$$= 1 + \alpha^2 + \alpha^4 + \alpha^6 + \alpha^7 + \alpha^9 + \alpha^{11} + \cdots + \alpha^{61} \tag{5}$$

$$\equiv 1 + \alpha^2 + \alpha^4 + (\alpha + 1) + \frac{\alpha^7 + \alpha^{63}}{1 + \alpha^2} \tag{6}$$

$$= \alpha + \alpha^2 + \alpha^4 + \frac{\alpha\alpha^6 + 1}{1 + \alpha^2} \tag{7}$$

$$\equiv \frac{(\alpha + \alpha^2 + \alpha^4)(1 + \alpha^2)}{1 + \alpha^2} + \frac{\alpha(\alpha + 1) + 1}{1 + \alpha^2} \tag{8}$$

$$= \frac{\alpha + \alpha^2 + \alpha^4 + \alpha^3 + \alpha^4 + \alpha^6 + \alpha^2 + \alpha + 1}{1 + \alpha^2} \tag{9}$$

$$= \frac{\alpha^3 + \alpha^6 + 1}{1 + \alpha^2} \tag{10}$$

$$\equiv \frac{\alpha^3 + (1 + \alpha) + 1}{1 + \alpha^2} \tag{11}$$

$$= \frac{\alpha^3 + \alpha}{1 + \alpha^2} \tag{12}$$

$$= \alpha \tag{13}$$

where Eqs. (6), (8) and (11) follow from equivalence modulo $\alpha^6 + \alpha + 1$. The bottom part of the syndrome (modulo $\alpha + 1$) is

$$s_2(\mathbf{r}) \equiv \sum_{i=0}^{N-1} r_i \alpha^i \pmod{\alpha + 1} \tag{14}$$

$$\equiv \text{weight}(\mathbf{r}) \pmod{2} \tag{15}$$

$$\equiv 32 \pmod{2} \tag{16}$$

$$\equiv 0 \tag{17}$$

The syndrome $\binom{\alpha}{0}$ indicates that the sequence is not a codeword in the (63,56) code, and is two away from the nearest codewords. A single error in any bit except r_1 will make the sequence differ from a codeword by one, and thus decodable.[1]

If instead a sequence is considered that is an odd-weight word in the perfect (63,57) code, it is three away from the codewords of the (63,56) code. This means that two errors must occur before it becomes decodable by a single-error–correcting (63,56) decoder. Such a sequence would be a better choice for a tail sequence because it is more resistant to accidental decoding in the presence of errors.

In selecting a particular sequence for the command coding application, the effects on the distance and uncorrectability properties as the code is shortened must be taken into account.

C. Shortening the (63,56) Code

Select a subset of a code, where all the codewords in the subset have zeros in some specified positions. Since all of the codewords in the subset have zeros in the specified positions, those positions carry no information and can be ignored. The resulting set of codewords forms a ***shortened code***. Shortening cannot decrease the minimum distance, and will only increase the minimum distance if the code is shortened severely.[2]

For this application, shortening will be done by taking only the codewords which have zeros in the leftmost or first m positions. Note that for each shortened word there is a corresponding full-length word that has zeros in the first m positions.

Example 2: Consider the sequence of length 55

010101010101010101010101010101010101010101010101 1010101

It corresponds to the full-length sequence

00000000010101010101010101010101010101010101010101010101 1010101

with weight 28.

The top part of the syndrome for the full-length sequence is

$$s_1(\mathbf{r}) \equiv \sum_{i=0}^{N-1} r_i\alpha^i \pmod{\alpha^6+\alpha+1} \tag{18}$$

$$= 1+\alpha^2+\alpha^4+\alpha^6+\alpha^7+\alpha^9+\alpha^{11}+\cdots+\alpha^{53} \tag{19}$$

$$\equiv 1+\alpha^2+\alpha^4+(1+\alpha)+(\alpha^7+\alpha^9+\alpha^{11}+\cdots+\alpha^{53}) \tag{20}$$

$$= \alpha+\alpha^2+\alpha^4+\frac{\alpha^7+\alpha^{55}}{1+\alpha^2} \tag{21}$$

$$\equiv \alpha+\alpha^2+\alpha^4+\frac{\alpha(\alpha+1)+\alpha(\alpha+1)^9}{(1+\alpha)^2} \tag{22}$$

[1] If the error is in r_1, the syndrome becomes $\binom{0}{1}$, indicating that it is three away from a codeword.

[2] For the (63,56) code, it can be shown that as long as the shortened length is greater than 32, the minimum distance will remain 4, and the maximum distance of any sequence to the nearest codewords will remain 3.

$$= \alpha + \alpha^2 + \alpha^4 + \frac{\alpha(1 + (\alpha + 1)^8)}{1 + \alpha} \tag{23}$$

$$= \alpha + \alpha^2 + \alpha^4 + \frac{\alpha(1 + \alpha^8 + 1)}{1 + \alpha} \tag{24}$$

$$\equiv \alpha + \alpha^2 + \alpha^4 + \frac{\alpha\alpha^2(\alpha + 1)}{1 + \alpha} \tag{25}$$

$$= \alpha + \alpha^2 + \alpha^4 + \alpha^3 \tag{26}$$

$$= \alpha(\alpha + 1)^3 \tag{27}$$

$$\equiv \alpha^{19} \tag{28}$$

The bottom part of the syndrome is zero since the weight is even. The two-part syndrome $\binom{\alpha^{19}}{0}$ indicates that the sequence is not a codeword, and that it is two away from the nearest codewords. A single error can make the sequence differ from a codeword by one, and thus decodable.

D. Finding a Good Uncorrectable Sequence

The concepts in Section II.B and Section II.C lead to the definition of a *good uncorrectable sequence* as one for which it and all the desired truncations of it are maximally distant from the codewords in the corresponding code, i.e., an odd-weight codeword in the perfect (63,57) code. The syndrome for good uncorrectable sequences is $\binom{0}{1}$. The analysis below shows that an uncorrectable sequence can be chosen so that, when it is truncated by octets, its syndrome does not change, and therefore it does not become correctable.

If a given sequence is truncated by m bits, and if it is desired that the syndrome not be changed by the truncation, then the smallest possible m is eight. This is because in order to not change either part of the syndrome, the truncated bits must correspond to a polynomial that is zero modulo both $\alpha + 1$ and $\alpha^6 + \alpha + 1$; the lowest order nonzero polynomial satisfying that requirement is $g(\alpha) = \alpha^7 + \alpha^6 + \alpha^2 + 1$. Therefore the shortest such nonzero sequence is 11000101, which has length eight.

There are engineering reasons for truncating by octets in the application considered in Section III. Also, bit synchronization requirements often make it preferable to have many transitions in the sequence, and thus a mostly zeros sequence is undesirable. For the remainder of the article, it is assumed that shortening will be done only by multiples of eight bits, and that octets of zeros are not of interest.

Since the bottom part of the syndrome must be 1 for the sequence and all of its truncations, and the truncated bits 11000101 have even weight, the nontruncated part of the sequence must have odd weight. Since the top part of the syndrome must be 0 for the sequence and all of its truncations, the nontruncated part of the sequence must correspond to a polynomial which is zero modulo $\alpha^6 + \alpha + 1$. The shortest such sequence is 1000011.

A simple construction of a good uncorrectable sequence is a concatenation of octets of the form 11000101 with the seven bits 1000011 in the rightmost positions. This is not the only good uncorrectable sequence, but it does have good distance and bit synchronization properties. The syndrome for this sequence is confirmed in the next example.

Example 3: Consider

11000101 11000101 11000101 11000101 11000101 11000101 11000101 1000011

This sequence has weight 31, so the bottom part of the syndrome $s_2(\mathbf{r})$ is 1. The top part of the syndrome (modulo $\alpha^6 + \alpha + 1$) for the full-length sequence is

$$s_1(\mathbf{r}) \equiv \sum_{i=0}^{N-1} r_i \alpha^i \quad (\text{mod } \alpha^6 + \alpha + 1) \tag{29}$$

$$= 1 + \alpha + \alpha^6 + \sum_{k=0}^{6} \alpha^{7+8k}(1 + \alpha^2 + \alpha^6 + \alpha^7) \tag{30}$$

$$= 1 + \alpha + \alpha^6 + \sum_{k=0}^{6} \alpha^{7+8k}(1 + \alpha + \alpha^6)(1 + \alpha) \tag{31}$$

$$\equiv 0 \tag{32}$$

Thus, the sequence is three away from the nearest codewords. A single error cannot make this sequence decodable.

III. Command Link Coding

The CCSDS recommendation uses a tail sequence that is specially constructed to not decode. By not decoding, it causes the receiver to begin searching for the next CLTU. It will be shown that the sequence that has been recommended is not the best sequence in terms of distance, and a better one will be given.

In order to apply the results of Section II to the command coding problem, the operations of the CLTU must be detailed. Refer to [1] for more detailed explanations of what is summarized here.

A. Telecommand Codeblock

The telecommand codeblock has K information bits, 7 *inverted* parity check bits, and a fill bit for a total length of L. Because of the code selected and the desire to shorten in units of 8 bits, the number of information bits and block lengths considered are $K = 32$, 40, 48, and 56, and $L = 40$, 48, 56, and 64, respectively.

B. Command Link Transmission Unit

The CLTU consists of

(1) a 16-bit start sequence, namely 1110101110010000,

(2) a number of telecommand codeblocks (the information may be padded with fill to make the information into a multiple of K bits), and

(3) a tail sequence which is a sequence of bits the same length as a codeblock and designed to be uncorrectable. The idea is to cause the receiver to stop decoding and begin looking for the start sequence of the next CLTU.

C. Tail Sequences

The CCSDS recommendations specify that the tail sequence $\mathbf{t}$ be a sequence of alternating zeros and ones beginning with a zero. Ignoring the fill bit, and noting that the parity bits are inverted, it can be seen that for a block length L of 64, this corresponds to the sequence in Example 1. As illustrated in the example, this sequence has distance properties such that a single channel error can, and almost certainly will, make the sequence decodable.

For the shorter block lengths $L = 40$, 48, and 56, the sequences of alternating zeros and ones beginning with a zero have corresponding full-length sequences with zeros filled in the first positions. All three of these corresponding sequences have even weights, and have nonzero syndromes. The calculation for $L = 56$ is done in Example 2.

If instead the sequence

$$11000101\ 11000101\ 11000101\ 11000101\ 11000101\ 11000101\ 11000101\ 0111100\ 0$$

is used as the tail sequence, it corresponds (when the fill bit is removed and the parity bits are inverted) to the sequence in Example 3. If one channel error occurs, then this sequence will still be uncorrectable.

D. Augmentation Using the Fill Bit

The CCSDS recommendations propose to use the fill bit as a flag to tell the decoder to operate in error-detect mode only. This augmentation is only suggested for use with the tail sequence. This improves the probability of spotting the tail sequence by not allowing the decoder to correct any errors when the fill bit is 1. In this mode, two errors are sufficient to make the CCSDS tail sequence (of alternating zeros and ones) decodable, while a minimum of three errors is required to make the sequence presented here decodable.

IV. Conclusions

The analysis in this article shows that there are uncorrectable sequences that can tolerate one more channel error than the CCSDS tail sequence before becoming decodable. It is also shown that this property may be preserved for the shortened as well as the full-length codes recommended by the CCSDS. A sequence satisfying these requirements should be considered for the role of the CCSDS tail sequence since it is more resistant to channel errors than the proposed tail sequence, and still has many transitions to aid bit synchronization.

Tables of probabilities of missing the tail sequence, the operation of the decoder on the shortened codes, and the proof that no sequence requiring at least three channel errors before it can possibly decode exists for this code may be the subject of future work.

Reference

[1] NASA, Consultative Committee on Space Data Standards, *Telecommand Part 1 Channel Service*, Blue Book, 201.0-B-1, Washington, D.C., January 1987.