Quick-Look Decoding Schemes for DSN Convolutional Codes

C. A. Greenhall and R. L. Miller
Communications Systems Research Section

The Galileo project will be tracked both by the Tracking and Data Relay Satellite System (TDRSS) and the DSN, whose (7, 1/2) convolutional codes differ in the order of the two symbols in each pair. To resolve this problem, we propose quick-look decoding of the TDRSS data. Quick-look decoding schemes requiring only simple shift registers are given for the DSN (7, 1/2) and (7, 1/3) convolutional codes. These schemes can be used when the communication channel is known to be virtually error free. The schemes not only decode the data, but can also detect symbol errors and the lack of node synchronization.

I. Introduction

The Galileo project will require both the Tracking and Data Relay Satellite System (TDRSS) and the DSN for tracking purposes. The TDRSS will be used for a few hours, the DSN for a few years. A problem has arisen because the TDRSS ground stations and the DSN stations have different Viterbi decoders. The connection vectors for the convolutional code that the TDRSS can decode are reversed from those of the DSN scheme.

This article presents one possible solution to this problem, viz. quick-look decoding. The following conditions are implicitly assumed throughout:

(1) The Galileo spacecraft is equipped with only a DSN-type encoder; TDRSS cannot process these data.

(2) The spacecraft-TDRSS communication channel is virtually error-free.

The second condition allows the possibility of quick-look decoding the data, without attempting to correct any errors. The advantage of quick-look decoding is that it involves only a couple of shift registers of length 7, as against the need to purchase a new Viterbi decoder.

Section II contains a brief description of what the design philosophy of a quick-look decoder should be. Section III presents a tutorial discussion of the theory behind quick-look convolutional decoders. Section IV contains decoding formulas for the DSN (7, 1/2) and (7, 1/3) convolutional codes. We have also included connection diagrams of these codes and their decoders.

II. Design Philosophy

A quick-look decoder should be a simple, fast algorithm that correctly decodes an error-free symbol stream. It does not attempt to correct errors, but may detect the presence of symbol errors or lack of bit synchronization. Nevertheless, such a decoder should not go looking for trouble. The less it knows, the better. The fewer symbols it has to look at, the fewer symbol errors it will see, and the fewer bit errors it will make. Finally, it must not propagate symbol errors indefinitely far down the decoded bit stream; each decoded bit should depend only on the last few symbols.

III. Theory

A. Inversion Formulas

The method of constructing quick-look decoders is taken from Massey and Sain (Ref. 1). The present exposition is self-contained. Let a convolutional code with constraint length $d+1$ and rate $1/r$, r an integer, be specified by *connection polynomials*

$$C_j(x) = c_{j0} + c_{j1}x + \cdots + c_{jd}x^d, \quad 1 \leq j \leq r$$

The coefficients are 0 or 1 and arithmetic is performed modulo 2. The sequences of information bits b_n and coded symbol vectors $(s_{1n}, \cdots, s_{rn})$, $-\infty < n < \infty$, are represented by the formal power series,

$$B(x) = \sum_{n=-\infty}^{\infty} b_n x^n$$

$$S_j(x) = \sum_{n=-\infty}^{\infty} s_{jn} x^n, \quad 1 \leq j \leq r$$

related by

$$S_j(x) = C_j(x) B(x), \quad 1 \leq j \leq r \tag{1}$$

In other words,

$$s_{jn} = c_{j0}b_n + c_{j1}b_{n-1} + \cdots + c_{jd}b_{n-d},$$

$$1 \leq j \leq r, \; -\infty < n < \infty$$

Knowing the $C_j(x)$ we must recover $B(x)$. Assume that the greatest common divisor (*gcd*) of $C_1(x), \cdots, C_r(x)$ is 1. By Euclid's Algorithm we can construct *inversion polynomials* $A_1(x), \cdots, A_r(x)$ of degree $< d$ such that

$$A_1(x)\,C_1(x) + \cdots + A_r(x)\,C_r(x) = 1 \tag{2}$$

From Eqs. (1) and (2) we have (dropping the x's now)

$$B = (A_1C_1 + \cdots + A_rC_r)B$$

$$B = A_1S_1 + \cdots + A_rS_r \tag{3}$$

for any bit sequence B. Hence the transformation defined by Eq. (1) is one-one and Eq. (3) gives its inverse. If

$$A_j(x) = a_{j0} + a_{j1}x + \cdots + a_{je}x^e, \quad 1 \leq j \leq r$$

then Eq. (3) says

$$b_n = \sum_{j=1}^{r} (a_{j0}s_{jn} + a_{j1}s_{j,n-1} + \cdots + a_{je}s_{j,n-e}),$$

$$-\infty < n < \infty \tag{4}$$

Conversely, if about the polynomials $C_1(x), \cdots, C_r(x)$ and $A_1(x), \cdots, A_r(x)$ we are given only that Eq. (3) inverts Eq. (1) for all B, then setting $B = 1$ gives Eq. (2) and the conclusion that $gcd\,(C_1, \cdots, C_r) = 1$. Thus:

> *An inversion formula of form (3) exists if and only if $gcd\,(C_1, \cdots, C_r) = 1$.*

B. Symbol Error Detection

Let $gcd\,(C_1, \cdots, C_r) = 1$. Assume that the formal power series $S_1, \cdots, S_r$ are a code stream, i.e., they satisfy Eq. (1) for some B. Then trivially we have

$$C_iS_j = C_jS_i, \quad 1 \leq i, j \leq r \tag{5}$$

We show that *Eq. (5) is necessary and sufficient for $S_1, \cdots, S_r$ to be a code stream.* We just saw that it is necessary. Assume that Eq. (5) holds. Define B by Eq. (3) as before. We show that Eq. (1) holds. It is enough to set $j = 1$. We have

$$\begin{aligned} C_1B &= C_1A_1S_1 + C_1A_2S_2 + \cdots + C_1A_rS_r \\ &= C_1A_1S_1 + A_2C_2S_1 + \cdots + A_rC_rS_1 \\ &= S_1 \end{aligned}$$

by virtue of Eqs. (5) and (2). Equation (5) thus yields parity checks of the received symbols.

IV. DSN Codes

We give decoding formulas and parity checks for the DSN (7, 1/2) and (7, 1/3) convolutional codes, whose connection diagrams are given in Figs. (1) and (2).

A. (7, 1/2) Code

The connection and inversion polynomials are

$$C_1(x) = 1 + x^2 + x^3 + x^5 + x^6$$

$$C_2(x) = 1 + x + x^2 + x^3 + x^6$$

$$A_1(x) = x^2 + x^4$$

$$A_2(x) = 1 + x + x^2 + x^3 + x^4$$

The quick-look inversion formula (Eq. (4)) is

$$b_n = s_{1,n-2} + s_{1,n-4} + s_{2,n} + s_{2,n-1} + s_{2,n-2} + s_{2,n-3} + s_{2,n-4} \tag{6}$$

in which b_n is the nth decoded bit and (s_{1n}, s_{2n}) is the nth symbol pair. Thus if we know (s_{1n}, s_{2n}) for $n \geqslant 0$ we can recover b_n for $n \geqslant 4$ by this method.

The parity check for detection of symbol errors or lack of bit synchronization (Eq. (5)) is

$$s_{1,n} + s_{1,n-1} + s_{1,n-2} + s_{1,n-3} + s_{1,n-6} + s_{2,n}$$
$$+ s_{2,n-2} + s_{2,n-3} + s_{2,n-5} + s_{2,n-6} = 0 \tag{7}$$

A short burst of parity errors in a good channel indicates one or more symbol errors; a long run of parity bits with a high proportion of errors indicates incorrect node synchronization.

Actual implementations of this code use trivial modifications of it: the DSN inverts the first symbol s_{1n}, and TDRSS transmits s_{1n} and s_{2n} in reverse order. In these situations, Eq. (6), Eq. (7), and Fig. 3 may have to be altered.

B. (7, 1/3) Code

To C_1 and C_2 of the previous code we adjoin a third polynomial:

$$C_3(x) = 1 + x + x^2 + x^4 + x^6$$

The three polynomials C_1, C_2, C_3 are pairwise relatively prime. Hence we can treat each of the three subcodes (C_i, C_j), $i < j$, separately. For these, Eq. (2) reads

$$(x^2 + x^4)\, C_1(x) + (1 + x + x^2 + x^3 + x^4)\, C_2(x) = 1 \tag{8}$$

$$(x + x^3)\, C_1(x) + (1 + x^2 + x^3)\, C_3(x) = 1 \tag{9}$$

$$(x^3 + x^4 + x^5)\, C_2(x) + (1 + x + x^4 + x^5)\, C_3(x) = 1 \tag{10}$$

Of course, Eq. (8) comes from the (7, 1/2) code. One could add these to get a single Eq. (3) with $r = 3$, but, according to our design philosophy, there is no point in doing this. The simplest of the above equations is Eq. (9); thus we may as well ignore the S_2 data entirely, use the inversion polynomials $x + x^3$ and $1 + x^2 + x^3$ on S_1 and S_3, and use only S_1 and S_3 for a parity check.

The quick-look inversion formula is

$$b_n = s_{1,n-1} + s_{1,n-3} + s_{3,n} + s_{3,n-2} + s_{3,n-3} \tag{11}$$

in which b_n is the nth decoded bit and (s_{1n}, s_{2n}, s_{3n}) the nth symbol triplet. The parity check is

$$s_{1,n} + s_{1,n-1} + s_{1,n-2} + s_{1,n-4} + s_{1,n-6} + s_{3,n}$$
$$+ s_{3,n-2} + s_{3,n-3} + s_{3,n-5} + s_{3,n-6} = 0 \tag{12}$$

Figure (4) gives the connection diagram of this decoder.

V. Conclusion

We have shown that both the (7, 1/2) and (7, 1/3) convolutional codes can be decoded using quick-look schemes. Such a scheme may be used by the Galileo project to work around the incompatibility between the TDRSS and DSN codes.

Reference

1. Massey, James L., and Sain, Michael K., Inverses of Linear Sequential Circuits. *IEEE Trans. Comput.*, Vol. C-17, 1968, 330–337.

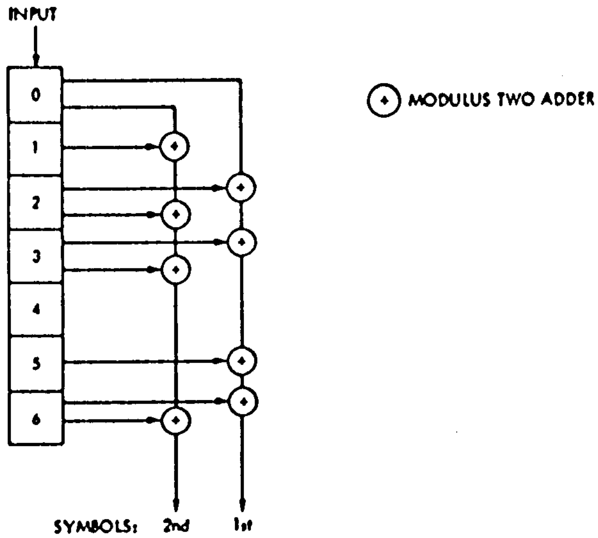

Fig. 1. Connection diagram of unmodified (7, 1/2) code, without symbol inversions or exchanges

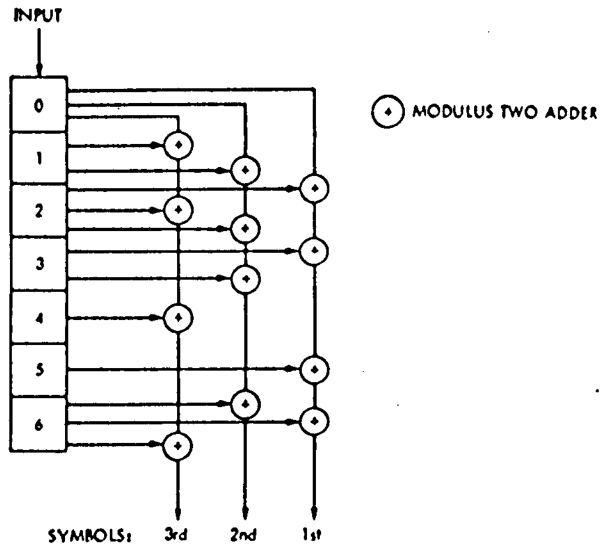

Fig. 2. Connection diagram of (7, 1/3) code

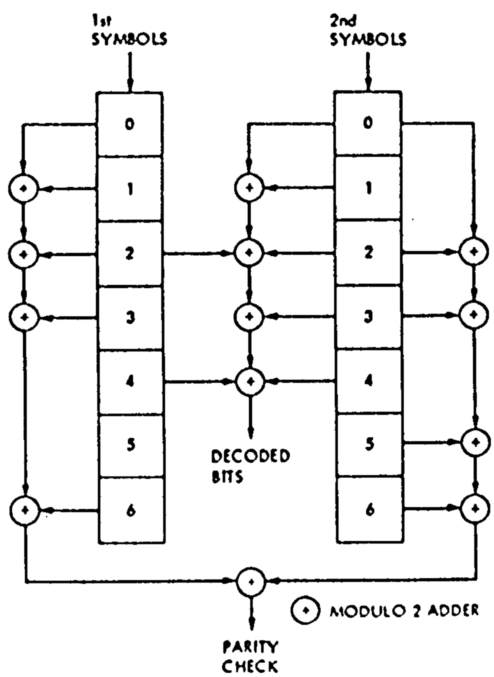

Fig. 3. Connection diagram of quick-look decoder for unmodified (7, 1/2) code, without symbol inversions or exchanges

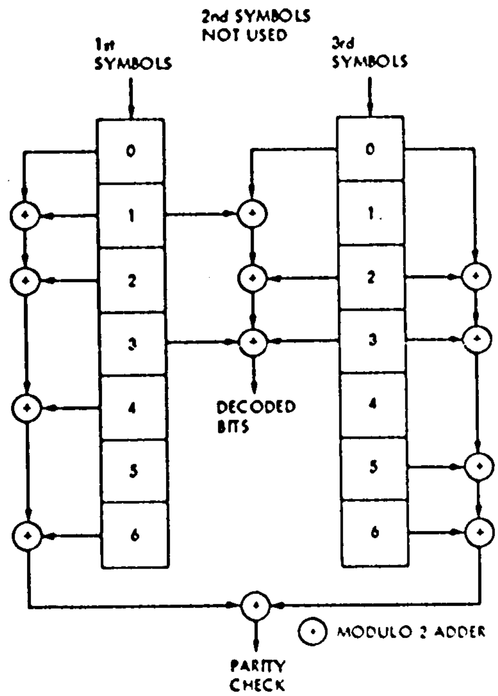

Fig. 4. Connection diagram of quick-look decoder for (7, 1/3) code