Cascaded Convolutional Codes

F. Pollara and D. Divsalar
Communications Systems Research Section

Due to the hardware design of Galileo's Command and Data Subsystem (CDS), the channel code usable in an S-band (2290–2300 MHz) mission must include the NASA standard (7,1/2) convolutional code. Galileo's hardware encoder for the (15,1/4) code is not usable in S-band mode. However, the need for higher coding gain dictates the use of long constraint length convolutional codes. Theoretical results show how a large subclass of such codes is realizable by using a software encoder in the CDS cascaded with the hardware encoder for the NASA standard code.

I. Introduction

Several options for improving Galileo's telemetry downlink performance at Jupiter if the high-gain antenna fails to deploy were evaluated in the *Galileo Options Study*[1] sponsored by the Telecommunications and Data Acquisition (TDA) Office. Specific recommendations were developed in the subsequent *Galileo S-Band Mission Study*.[2]

In this article, the authors describe one of the proposed options to improve Galileo's S-band (2290–2300 MHz) downlink performance based on the use of advanced long constraint length convolutional codes.

The Command and Data Subsystem (CDS) of Galileo provides two output paths to the Modulation/Demodulation Subsystem (MDS): a low-rate telemetry output (40 bps) and a high-rate telemetry output (10 bps to 134.4 kbps). The low-rate output is directly connected to the low-gain antenna path. The high-rate output may use the low-gain antenna only through a hardware (7,1/2) convolutional encoder, as shown in Fig. 1. Galileo's hardware encoder for the (15,1/4) code is not usable in S-band mode.

One of the options to improve Galileo data return through its low-gain antenna is to use advanced channel coding techniques, including long constraint length convolutional codes. This could be achieved by uploading a software (15,1/4) encoder in the CDS and using the low-rate output connected to the S-band telemetry path. However, the encoder output rate would be fixed to 40 symbols per second, which is not compatible with the desire for higher rates.

The only way to send S-band telemetry at higher rates is to use the high-rate CDS output, which then forces the use of the hardware (7,1/2) convolutional encoder. Therefore, methods are to be sought for realizing long constraint

[1] L. Deutsch, *Galileo Options Study* (internal document), Jet Propulsion Laboratory, Pasadena, California, November 5, 1991.

[2] L. Deutsch and J. Marr, *Galileo S-Band Mission Study Final Report* (internal document), Jet Propulsion Laboratory, Pasadena, California, March 2, 1992.

length convolutional codes by cascading a software encoder with the existing hardware (7,1/2) encoder.

II. Cascaded Convolutional Codes

The best solution would be to find a method for bypassing the (7,1/2) hardware encoder by realizing an inverse software encoder preceding it. Any desired code would then be realizable in software.

It is well known that a noncatastrophic encoder has a feed-forward inverse [1]. Denote N as the (7,1/2) hardware encoder on Galileo with generator polynomials g_0 and g_1, and N^{-1} as its inverse. Then it is possible to undo the operation of N, as shown in Fig. 2, where M is a multiplexer and the relative symbol rates are shown below each connection.

For the (7,1/2) standard NASA code having generator polynomials $g_0 = 1 + x + x^2 + x^3 + x^6$ and $g_1 = 1 + x^2 + x^3 + x^5 + x^6$, the feed-forward inverse is delay-free, since the greatest common divisor (GCD) $(g_0, g_1) = 1$, and is given by $f_0 = 1 + x + x^2 + x^3 + x^4$ and $f_1 = x^2 + x^4$, since GCD $(g_0, g_1) = g_0 f_0 + g_1 f_1 = 1$. The feed-forward inverse can be used to recover the information sequence a from the encoder output y. This inverse cannot be used in the Galileo CDS configuration since the output of the hardware encoder is not accessible.

However, the problem of constructing a preinverse of N such that the sequences w and y are identical, as shown in Fig. 3, has no solution in general. This is clear from an information-theory point of view, since w can be any binary sequence while the sequence y is restricted to being a code word of the specific code in use.

A. Structure and Design of Cascaded Codes

An alternative method for realizing a longer constraint length code with some form of processing preceding the hardware encoder is shown in Fig. 4.

The structure shown in Fig. 4 is just an example of one of the possible structures for obtaining a rate 1/4 code equivalent to code D, shown in Fig. 5. This structure cannot obtain all desired codes but just a certain subclass of codes.

A simplified strategy for designing a (15,1/4) cascaded code is to assume that a code C specified through f_0 and f_1 is given and then compute the resulting equivalent code specified by h_0, h_1, h_2 and h_3. One has

$$y_0(x) = a(x) f_0(x)$$

$$y_1(x) = a(x) f_1(x)$$

$$z(x) = y_0(x^2) + x y_1(x^2)$$

$$u_0(x) = z(x) g_0(x)$$

$$u_1(x) = z(x) g_1(x)$$

$$w(x) = u_0(x^2) + x u_1(x^2)$$

which gives

$$w(x) = a(x^4)[f_0(x^4) + x^2 f_1(x^4)][g_0(x^2) + x g_1(x^2)]$$

and, from Fig. 5,

$$q(x) = s_0(x^4) + x s_1(x^4) + x^2 s_2(x^4) + x^3 s_3(x^4)$$

$$s_0(x) = a(x) h_0(x)$$

$$s_1(x) = a(x) h_1(x)$$

$$s_2(x) = a(x) h_2(x)$$

$$s_3(x) = a(x) h_3(x)$$

which gives

$$q(x) = a(x^4)[h_0(x^4) + x h_1(x^4) + x^2 h_2(x^4) + x^3 h_3(x^4)]$$

Since one wants

$$q(x) = w(x)$$

to hold, the condition becomes

$$[f_0(x^4) + x^2 f_1(x^4)][g_0(x^2) + x g_1(x^2)] =$$

$$[h_0(x^4) + x h_1(x^4) + x^2 h_2(x^4) + x^3 h_3(x^4)] \qquad (1)$$

By expanding the left-hand side of Eq. (1) and identifying terms of equal power, one can find polynomials h_0, h_1, h_2, and h_3 satisfying this equation. The memory of a convolutional code is the maximum degree among its generator polynomials. If one defines by m_C, m_N, and m_D the memories of codes C, N, and D, Eq. (1) implies that

$$2m_D = 2m_C + m_N$$

The same result applies to the respective constraint lengths, since for these codes the constraint length K is equal to $m + 1$.

B. Example

In this example, it is assumed that g_0, g_1, f_0, and f_1 are given and a solution for h_0, h_1, h_2, and h_3 is sought. In particular, if one chooses

$$f_0(x) = 1 + x^9 + x^{11}$$

$$f_1(x) = 1 + x^3 + x^6 + x^9 + x^{11}$$

one gets $h_0 = 57347, h_1 = 71526, h_2 = 02245$, and $h_3 = 52207$, in octal representation. This yields a rate $r = 1/4$ code with memory $m = 14$ (constraint length $K = m + 1 = 15$), i.e., a (15,1/4) code with free distance $d_f = 30$, while the experimental code on Galileo had $d_f = 35$ with the same parameters. Searching over f_0 and f_1 may yield better codes.

A more explicit solution for Eq. (1) can be found by defining g_0 and g_1 in terms of their even and odd parts

$$\begin{aligned} g_0(x) &\triangleq g_{0e}(x^2) + x g_{0o}(x^2) \\ g_1(x) &\triangleq g_{1e}(x^2) + x g_{1o}(x^2) \end{aligned} \qquad (2)$$

Then it follows that

$$\left.\begin{aligned} h_0(x) &= f_0(x) g_{0e}(x) + x f_1(x) g_{0o}(x) \\ h_1(x) &= f_0(x) g_{1e}(x) + x f_1(x) g_{1o}(x) \\ h_2(x) &= f_0(x) g_{0o}(x) + f_1(x) g_{0e}(x) \\ h_3(x) &= f_0(x) g_{1o}(x) + f_1(x) g_{1e}(x) \end{aligned}\right\} \qquad (3)$$

As a verification, the results obtained in the previous example can be reproduced by using this explicit solution.

Convolutional codes with high coding gain, including the original (15,1/4) Galileo code, are such that the first and last coefficient of all generator polynomials are equal to 1, i.e., $h_{i,j} = 1, i = 0, 1, 2, 3, j = 0, 14$, where $h_i(x) \triangleq \sum_{j=0}^{14} h_{i,j} x^j$. Therefore, it is interesting to determine whether a cascaded code having this property exists. From Eqs. (3) one has

$$f_{0,0} g_{0e,0} = h_{0,0}$$

$$f_{0,0} g_{1e,0} = h_{1,0}$$

$$f_{0,0} g_{0o,0} + f_{1,0} g_{0e,0} = h_{2,0}$$

$$f_{0,0} g_{1o,0} + f_{1,0} g_{1e,0} = h_{3,0}$$

where $f_i(x) \triangleq \sum_{j=0}^{11} f_{i,j} x^j$ and $g_{ip}(x) \triangleq \sum_{j=0}^{6} g_{ip,j} x^j$. In order to get $h_{i,0} = 1$ and $i = 0, 1, 2, 3$, one should have

$$g_{0e,0} = g_{1e,0} = 1 \text{ and } g_{0o,0} = g_{1o,0}$$

Also, from Eq. (3) one has the following conditions, on the coefficients of x^{14}

$$f_{0,11} g_{0e,3} + f_{1,11} g_{0o,2} = h_{0,14}$$

$$f_{0,11} g_{1e,3} + f_{1,11} g_{1o,2} = h_{1,14}$$

$$f_{1,11} g_{0e,3} = h_{2,14}$$

$$f_{1,11} g_{1e,3} = h_{3,14}$$

In order to get $h_{i,14} = 1$ and $i = 0, 1, 2, 3$, one should have

$$g_{0e,3} = g_{1e,3} = 1 \text{ and } g_{0o,2} = g_{1o,2}$$

But, for the (7,1/2) NASA code, one has

$$g_{0e,0} = g_{1e,0} = 1$$

$$g_{0o,0} = 1 \text{ and } g_{1o,0} = 0$$

$$g_{0e,3} = g_{1e,3} = 1$$

$$g_{0o,2} = 0 \text{ and } g_{1o,2} = 1$$

which implies $g_{0o,0} \neq g_{1o,0}$ and $g_{0o,2} \neq g_{1o,2}$. Thus, it is impossible to get $h_{i,0} = 1$ for all i's and/or $h_{i,14} = 1$ for all i's. When the NASA code is used, the following is possible

$$h_{0,0} = h_{1,0} = 1 \text{ and } h_{2,0} \neq h_{3,0}$$

and

$$h_{0,14} \neq h_{1,14} \text{ and } h_{2,14} = h_{3,14} = 1$$

III. Conclusion

A method is presented for realizing long constraint length convolutional codes as a cascade of two codes including the NASA standard (7,1/2) code. This analysis shows that a large class of codes can be realized using this construction method. These results led to the inclusion of one of these cascaded codes in the design described in the *Galileo S-Band Mission Study.*

Reference

[1] J. L. Massey and M. K. Sain, "Inverses of Linear Sequential Circuits," *IEEE Transactions on Computers*, vol. C-17, pp. 330–337, April 1968.

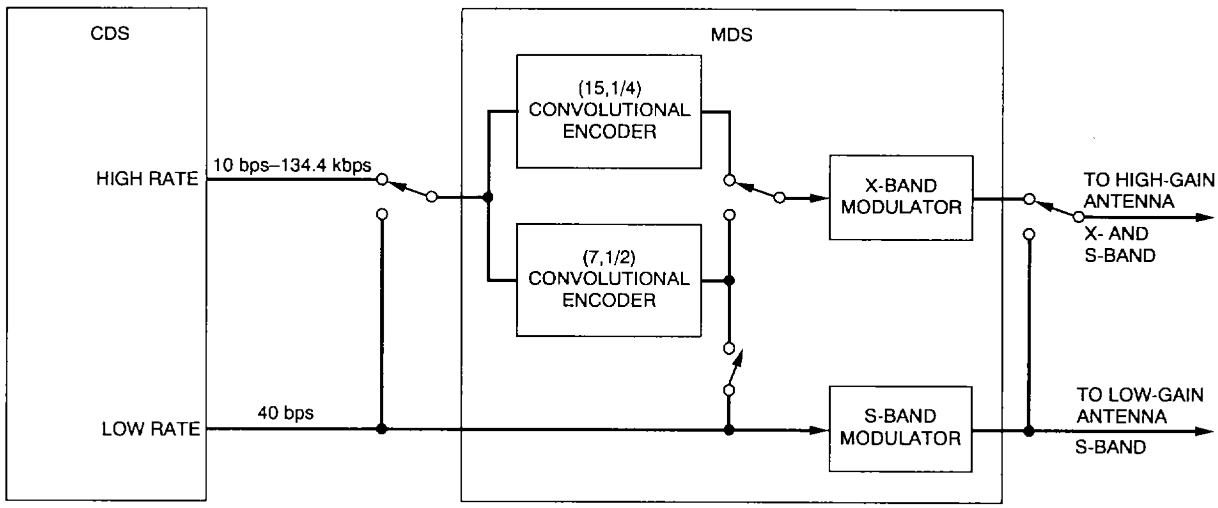

Fig. 1. Functional telemetry data flow.

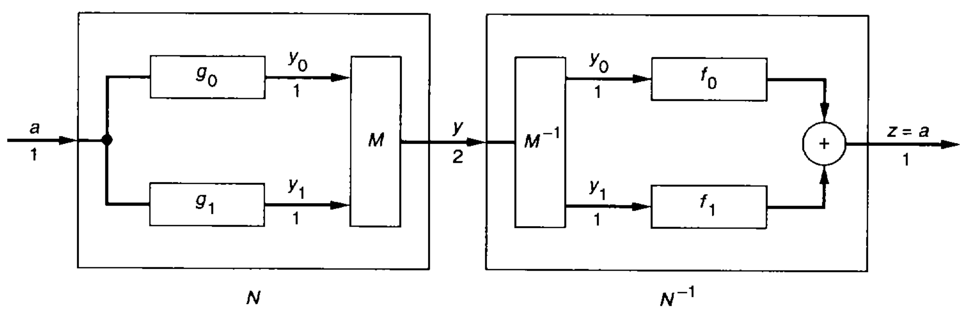

Fig. 2. Feed-forward inverse of code *N*.

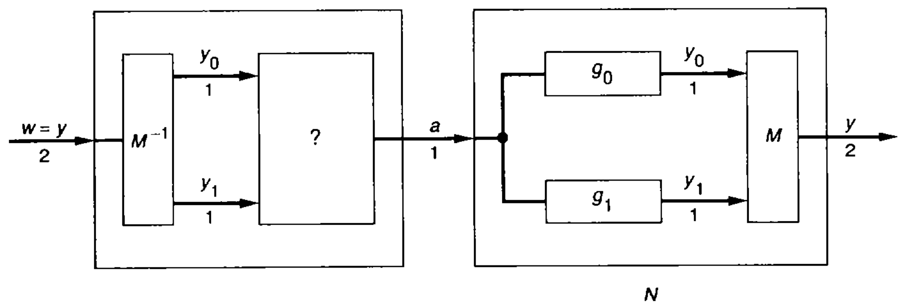

Fig. 3. Preinverse of code *N*.

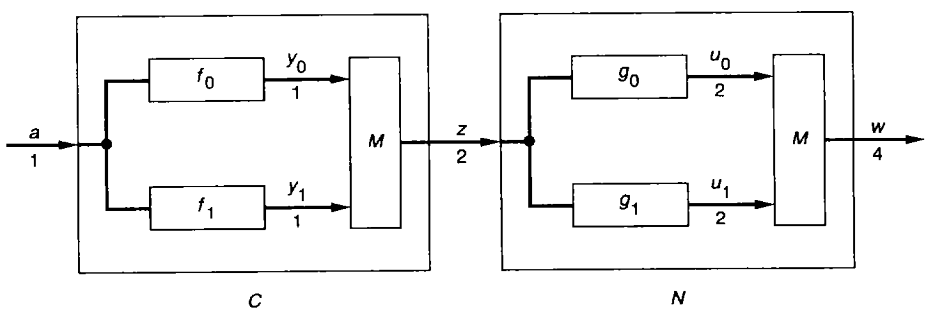

Fig. 4. Alternate structure for proposed Galileo code (rate = 1/4).

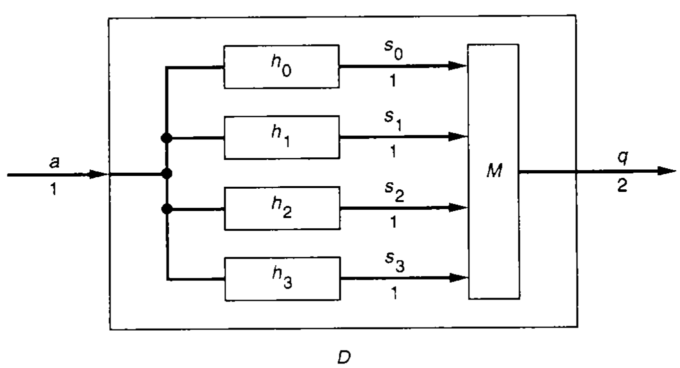

Fig. 5. Code equivalent to the cascades of codes *C* and *N*.