Serial Concatenated Systematic Convolutional Codes for Space Communications

A Letter
Submitted to the

*IEEE Transactions on Communications*

6 October 1995

by

Jean Yves Couleaud\textsuperscript{1} and Steven S. Pietrobon\textsuperscript{2}

Abstract

This letter describes the performance of the serial concatenation of two recursive systematic convolutional codes at very low $E_b/N_0$ iteratively decoded with the MAP (*maximum a posteriori*) algorithm.

Index Terms

Error correction, Turbo codes

---

This work was supported in part by the Institute for Telecommunications Research, University of South Australia and the Australian Research Council under an Australian Postdoctoral Research Fellowship.

\textsuperscript{1}Ecole Nationale Supérieure d’Ingénieurs de Constructions Aéronautiques, 49, Avenue Léon Blum, 31056 Toulouse Cedex, France.

\textsuperscript{2}Satellite Communications Research Centre, University of South Australia, The Levels SA 5095, Australia.
1 Introduction

The Galileo space probe on its way to Jupiter is currently limited to using its Low Gain Antenna (LGA) for transmitting data back to Earth (the 134,400 bits/s High Gain Antenna having failed to open). It is currently planned to use a serial concatenation of a memory 13, rate 1/4 convolutional code with a time varying GF$(2^8)$ Reed Solomon outer code of depth 8 and length 255. This scheme is mostly implemented in software and achieves a Bit Error Ratio (BER) of $10^{-7}$ at an $E_b/N_0$ of 0.58 dB [1] achieving a transmission rate up to 160 bits/s.

The recent discovery of “turbo” codes [2] has allowed schemes to perform up to 0.7 dB from Shannon capacity. Unfortunately these schemes cannot be used on Galileo since the LGA is hardwired to a standard memory 6, rate 1/2 non–systematic convolutional encoder with polynomials $g_0^1 = 171_8$ and $g_0^2 = 133_8$. In this letter, we present a modification of the turbo coding scheme by serially concatenating a rate 1/2 systematic convolutional code with an outer rate 1/2 systematic convolutional code with interleaving between the two codes [3]. The encoder is shown in Figure 1. Data are sent through an AWGN channel. For Galileo, the non–systematic encoder can be easily converted to systematic form by dividing the input sequence by either $g_0^1$ or $g_0^2$ (we chose $g_0^1$) using a linear feedback shift register. The outer code has $g_1^1$ as the divisor polynomial.

2 Decoding Scheme

A block diagram of the decoding scheme is shown in Figure 2. We chose to decode the outer code first and feed the $a$ posteriori information from both the information and parity bits we get from the MAP decoder [4] to the inner code. After transmitting $X_k^0, Y_k^0$ we receive the noisy $x_k^0, y_k^0$.

On the first iteration $z_k^0 = x_k^0$. $\Delta$ is the delay of each iteration.

2.1 Interleaver Design

Since MAP decoding is very sensitive to correlation in the noise, the interleaver we designed had to spread error bursts amongst the frame. In [3] the influence of the interleaver type for block interleaving, pseudo–random linear interleaving, and pseudo random interleaving for a
concatenated code was investigated. It was found that pseudo random linear interleaving gives the best performance, as in turbo codes.

The influence of the interleaver size for pseudo random interleaving was also investigated in [3]. It was found that for low SNR, interleaving becomes effective for sizes higher than 10,000 bits. In the next sections the interleaver size has been set to 16,384, almost the same as that used by Galileo.

2.2 Choice of Polynomials

Figure 3 shows the performance of using \( g_1^1/g_0^2 = 171/133 \) as the inner code and \( g_1^1/g_1^2 = 3/1, 5/7, 17/15 \) and \( 37/21 \) as the outer code at an \( E_b/N_0 = 1.0 \) dB. For low complexity codes around \(-2.5\) dB \( E_b/N_0 \) (where the inner code effectively operates in its first iterations), the individual codes performance reverses, i.e., the less complex the code the better the performance [3]. The 171/133 code performs very badly at this low of an \( E_b/N_0 \), so a more “powerful” less complex outer code needs to be used. As can be seen from Figure 3, the least complex two state code performs best.

This also explains why we choose to decode the outer code first, since it is more powerful in the first few iterations. However, as the number of iterations increases, the difference between starting with the inner or outer code becomes negligible.

If we choose polynomials other than the 171/133 NASA standard, we can greatly improve the performance of our scheme. For the 5/7-5/7 code a \( \text{BER} = 10^{-7} \) can be achieved at an \( E_b/N_0 \) of 0.32 dB after 20 iterations.

3 Effects of Puncturing

The achieved BERs for the previous codes using 171/133 as the inner code are very far from the performance of the scheme on Galileo as can be seen from Figure 3. In order to improve our results, we tried to puncture the outer code so as to “improve” the outer codes performance at the required operating point. Various puncturing matrices were tried with \( P = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \) performing the best with an overall code rate of 3/10. Rates lower or higher than this performed worse. Figure 4 shows that a \( \text{BER} = 10^{-7} \) is achieved at an \( E_b/N_0 \) of 0.57 dB after 15 iterations, similar to the
performance currently achieved by Galileo, but at a higher coding rate (0.3 compared to 0.219 for Galileo).

4 Conclusions

The coding schemes presented here give very high coding gains, but not as great as that achieved with parallel concatenated systematic convolutional codes. However, a code that could be used on Galileo (where parallel concatenated codes cannot be used) was found that is at least as good as the current code and which operates at a higher coding rate. Note that the analogy with turbo codes is obvious: in both cases the performance of the scheme is determined by the ability of the interleaver to randomise error bursts.

References


Figure 1: Encoder.

Figure 2: Decoder.

Figure 3: Performance of 171/133 based codes (1) Code 3/1–171/133 $E_b/N_0=0.5$ dB, (2) Code 3/1–171/133 $E_b/N_0=1.0$ dB, (3) Code 5/7–171/133 $E_b/N_0=1.0$ dB, (4) Code 17/15–171/133 $E_b/N_0=1.0$ dB, (5) Code 37/21–171/133 $E_b/N_0=1.0$ dB
Figure 4: Performance of the 3/1–171/133 code with punctured outer code (R=3/10).