A New Code for Galileo

S. Dolinar
Communications Systems Research Section

Over the past six to eight years, an extensive research effort was conducted to investigate advanced coding techniques which promised to yield more coding gain than is available with current NASA standard codes. The delay in Galileo's launch due to the temporary suspension of the shuttle program provided the Galileo project with an opportunity to evaluate the possibility of including some version of the advanced codes as a mission enhancement option. A study was initiated last summer to determine if substantial coding gain was feasible for Galileo and, if so, to recommend a suitable experimental code for use as a switchable alternative to the current NASA-standard code. The Galileo experimental code study resulted in the selection of a code with constraint length 15 and rate 1/4. The code parameters were chosen to optimize performance within cost and risk constraints consistent with retrofitting the new code into the existing Galileo system design and launch schedule. The particular code was recommended after a very limited search among good codes with the chosen parameters. It will theoretically yield about 1.5 dB enhancement under idealizing assumptions relative to the current NASA-standard code at Galileo's desired bit error rates. This ideal predicted gain includes enough cushion to meet the project's target of at least 1 dB enhancement under real, non-ideal conditions.

I. Introduction

After Galileo's launch was postponed by the suspension in the shuttle schedule, the Galileo telemetry link became the subject of extensive activity. The new Galileo trajectory will result in reduced telemetry margins at encounter. On the other hand, the delayed launch schedule affords an opportunity to seek countervailing enhancements. Of the many possible enhancements, one of the easiest to implement is a change in the convolutional code.

Convolutional codes have been used on deep space probes for many years. A convolutional code is described by its constraint length K, its rate $r = 1/N$,[1] and N "connection vectors" of K bits each. The encoder (which is the hardware necessary on the spacecraft) consists of a shift register of length K with N sets of taps. This means that a new encoder is a fairly small change on the spacecraft because its complexity is only linear in K and N. On the other hand, the decoder's complexity is roughly proportional to 2^K, so a change to a longer constraint length adds significantly to the problem of decoding on the ground.

[1]In general, convolutional codes can have rational rates k/N, but rates $1/N$ are the simplest and the most interesting for the Galileo application.

Over the past six to eight years, a great deal of research has been conducted in pursuit of advanced convolutional codes with rates $1/N$, $N \leq 6$, and constraint lengths $K \leq 15$ [1]–[5]. These parameters were chosen so that decoders working at typical deep space data rates could be implemented with foreseeable technological advances.

During the summer and fall of 1987, a search was conducted for an advanced convolutional code which could be used by Galileo as an experimental mission enhancement option. Such a code is limited by the bandwidth of Galileo's radio modulator to rates $1/N$ with $N \leq 4$. A constraint length 15, rate 1/4 code which will allow more than 1 dB of gain over the current $K = 7$, $r = 1/2$ convolutional code was found. This article describes the code search, the problems, and the result. The experimental code has constraint length 15 and rate 1/4, and it is specified by its four connection vectors written in octal form as 46321, 51271, 63667, 70535. For symbol synchronization purposes, alternate output symbols produced by this code are to be inverted, as is the case with the current NASA-standard (7,1/2) code.

An explanation of the connection vector notation and a detailed connection diagram for the Galileo experimental encoder are given in Section VII of this article. The simulated performance of this code and a comparison with the performance of the current NASA-standard code are given in Sections V and VI.

II. Background: An Extensive Previous Code Search Effort

Over the past six to eight years, the DSN undertook a research effort to study advanced coding techniques which promised to yield more coding gain than is available using current NASA-standard codes. The goal for the research effort was approximately 2 dB of extra coding gain over present performance in typical deep space missions such as Voyager and Galileo. This represents about half of the maximum possible theoretical gain between current performance and Shannon's ultimate limit on any code's performance. The potential application of the new codes would be deep space communications in the "far future."

The quest for 2 dB of coding gain took off in several directions from current codes. The research focused on the same basic concatenation of a Reed–Solomon outer code with a convolutional inner code, but the code parameters were allowed to vary to levels not feasible when the present NASA standards were developed. The research effort studied the effects of increasing the constraint length and decreasing the code rate of the convolutional code, and increasing the symbol size and optimizing the code rate of the Reed–Solomon code. Due to a higher predicted payoff in performance versus complexity, a significant advance in convolutional code parameters was attempted, whereas the Reed–Solomon code parameters were varied only slightly from those of the present Reed–Solomon code used on Voyager and Galileo.

The quest was declared a success when some codes were found which surpassed the 2-dB goal, with the best code improving performance by 2.11 dB. To achieve this amount of gain, the convolutional code constraint length had to be increased to $K = 15$ and the code rate decreased to $r = 1/6$, with a 10-bit (1,023,959) Reed–Solomon code used as the outer code. In addition, lengthy code searches were conducted using bounds, approximations, and simulations to search for good convolutional codes within any given code space. The code space corresponding to any given constraint length K and code rate $r = 1/N$ contains an astronomical number of possible codes (2^{KN}). Therefore, intelligent search techniques had to be developed in order to locate a "best code" when only a tiny fraction of all possible codes could be evaluated explicitly. Some of the work that went into this research is described in [1]–[4].

Table 1 shows the end result of the code search effort. The performance of the best code found during the code search is listed along with four other significant benchmarks: (1) uncoded performance; (2) the performance of the current NASA-standard (7,1/2) convolutional code by itself (unconcatenated); (3) the performance of the current NASA-standard convolutional code concatenated with the current Voyager/Galileo Reed–Solomon code; and (4) Shannon's limit on the performance of any code of arbitrary complexity. All figures are for communication at a desired bit error rate of 10^{-6} over a Gaussian channel using BPSK signals. Table 1 shows that the best code from the search comes within 2.01 dB of the Shannon limit. In comparison, the current Voyager/Galileo concatenated code misses this ultimate limit by 4.12 dB, and the current convolutional code alone misses by 6.37 dB (at a 10^{-6} bit error rate). To complete the perspective, uncoded performance at a 10^{-6} bit error rate is 12.13 dB worse than the Shannon limit. Thus, the best code discovered from the code search realizes just over half the maximum possible gain (measured in decibels) theoretically obtainable beyond the current concatenated code.

III. Potential Application of Advanced Codes to Galileo

As a result of recent advances in VLSI technology that make long constraint length Viterbi decoders feasible, the use of advanced codes in deep space communications is no longer reserved for the "far future." The delay in Galileo's launch due to the temporary suspension of the shuttle program provided the Galileo project with an opportunity to evaluate the possibility of including some version of the advanced codes as a mission enhancement option. A study was begun last summer to determine if substantial coding gain was feasible for Galileo and, if so, to recommend a suitable experimental code to be available as a switchable alternative to the current NASA-standard Galileo code.

IV. The Choice of Code Parameters for Galileo

The first thing that became apparent in the potential Galileo application was that the best code found during the original code search could not be directly applied to Galileo. The bandwidth expansion necessary for a rate 1/6 convolutional code simply could not be accommodated in a Galileo retrofit. In addition, there was a hard constraint on keeping the 8-bit (255,223) Reed–Solomon outer code. Therefore, a new abbreviated code search was launched to investigate how much of the 2.11 dB of potential gain could be obtained within the Galileo constraints.

The constraints imposed by Galileo were (1) an 8-bit (255,223) Reed–Solomon outer code; (2) a convolutional inner code with a rate no lower than 1/4 and preferably 1/2; and (3) a convolutional inner code with a constraint length small enough to guarantee that a practical Viterbi decoder could be built within the next few years. Within these constraints, the objective was to pick the code parameters that would achieve the maximum coding gain at minimal cost and risk to the project.

The earlier research had taken a giant leap forward to very advanced codes. Galileo now needed an estimate of gains that could come from smaller leaps. The initial phase of the code study for Galileo was to make rough estimates of the relative payoff for changing the code parameters unit by unit.

The first step in this initial phase was to gather data from previous code search studies and form extrapolations to roughly estimate the variation of performance with each of the code parameters. Approximate curves are shown in Fig. 1. The points on these curves represent best codes found according to various sources [3]–[5]. It is seen that an approximate gain of a little over 0.1 dB can be expected for each unit increase in K. Another 0.6 to 0.9 dB can be expected for decreasing the code rate from 1/2 to 1/4, and about 0.2 to 0.3 dB for further decreasing the code rate from 1/4 to 1/6. The 10-bit Reed–Solomon outer code is worth about 0.25 dB relative to the 8-bit code. Together, these components account for the 2.11-dB predicted gain of the $K = 15, r = 1/6$ convolutional code, 10-bit Reed–Solomon code concatenation versus the current NASA-standard $K = 7, r = 1/2$ convolutional code, 8-bit Reed–Solomon code concatenation.

The second step in the initial phase was to evaluate the costs and risks to the Galileo project of selecting particular sets of code parameters. The costs associated with varying each of the code parameters can be placed in three basic categories: (1) effects on encoder complexity; (2) effects on decoder complexity; and (3) effects on other spacecraft and ground systems.

Encoder complexity grows only modestly with constraint length K and reciprocal code rate $N = 1/r$. The encoder needs a shift register of length K and N sets of taps (exclusive-OR gates). For good codes, the total number of taps is roughly proportional to N. Technology was immediately available to build any encoder with $K \leqslant 15$ and $r \geqslant 1/6$, so encoder complexity was not a major driver in the selection of code parameters.

In contrast, decoder complexity grows exponentially with constraint length K. Memory and computational requirements for a Viterbi decoder increase as 2^K. Viterbi decoders operating at 100 kbps are not currently available for values of K greater than about 10. However, 100-kbps decoders for $K = 13$ to 15 should be achievable within the next two years based on VLSI technology.

Decoder complexity increases only modestly with reciprocal code rate N. Decoder memory requirements are independent of N, and computational requirements increase only linearly with N. Thus, decoder complexity was not a limiting factor in the choice of code rate.

Even though decreasing the code rate has relatively minor effects on the encoder and decoder, it has significant effects on other spacecraft and ground systems, especially in a retrofitted system like the one proposed for Galileo, whose original system design was based on old assumptions about the best codes available. The channel symbol rate increases linearly with reciprocal code rate N for a fixed information bit rate. Therefore, bandwidth requirements increase linearly with N, the smallest suitable subcarrier frequency is proportional to N, and symbol clock speeds are proportional to N. The symbol signal-to-noise ratio is less than the bit signal-to-noise ratio by $10 \log_{10} N$ dB, and symbol tracking might be impaired.

Increasing constraint length has no such side effects on other spacecraft and ground systems. The costs of choosing a long constraint length code are confined to increased complexity in the encoder and decoder, primarily the decoder.

The selection of code parameters for Galileo thus boiled down to the resolution of two major issues: how much bandwidth expansion Galileo could tolerate and what constraint length would permit low-risk decoder development before Galileo's encounter at Jupiter. To answer these questions, code performance evaluations were needed for the combinations of code parameters most acceptable (in terms of costs) in the Galileo application.

The third and final step in the initial phase of the study was to perform new cursory code searches in the code spaces $K = 13$, 15 and $r = 1/2$, 1/4. Figures 2 and 3 show some typical performance results from this very abbreviated study. The data points plotted in Figs. 2 and 3 represent simulated code performance under idealizing assumptions for a representative "best found code" for each combination of code parameters. The idealizing assumptions are described in the next section. At interesting levels of bit error rate (10^{-6} to 10^{-5} for concatenated coding and 5×10^{-3} for nonconcatenated coding), these curves indicate that 0.8 to 0.9 dB of performance gain is obtained by going from rate 1/2 to rate 1/4 and a little over 0.1 dB by going from constraint length 13 to constraint length 15. The project evaluated these predicted incremental performance gains relative to the costs and risks of selecting different constraint lengths and code rates.

The constraint length issue was resolved in favor of going all the way to the maximum constraint length considered in the study, $K = 15$. Current strides in the development of the BIG Viterbi Decoder[2] for the DSN indicate that this assumption was realistic. A prototype of a $K = 15$ Viterbi decoder should be completed by late 1990.

The project's position on code rate was to stay with rate 1/2 if at all possible, because this would confine changes on the spacecraft to the new encoder (plus a switch for invoking the experimental code option). However, given the large payoff in coding gain for going from rate 1/2 to 1/4, the required spacecraft modifications were evaluated for rates down to 1/4. Code rates lower than 1/4 could not be considered, because Galileo's Radio Frequency Subsystem could not tolerate more than a doubling of the subcarrier frequency. Eventually, the decision was made to select a code with rate 1/4, because this was the only alternative that would guarantee a worthwhile performance gain of at least 1 dB. This choice of code rate means that the spacecraft needs not only an optional experimental encoder but also a mechanism for doubling the subcarrier frequency and symbol clock rate whenever the optional encoder is invoked. Appropriate modifications to Galileo's Telemetry Modulation Unit were approved by the project in order to support the experimental rate 1/4 code.

[2]Work on this decoder will be reported in subsequent issues.

With the code parameters selected at $K = 15$, $r = 1/4$, approximately 1.5 dB of theoretical relative coding gain was the goal. Some portion of the theoretical gain will undoubtedly be whittled away due to such things as extra radio loss and the effects of non-ideal interleaving. A practical target was set for at least 1 dB of real net gain from the experimental code.

V. The Abbreviated Code Search for Galileo

The original code search investigated thousands of codes with constraint lengths 13, 14, 15 and code rates 1/4, 1/5, 1/6. Randomly selected codes were subjected to a quick initial evaluation based on free distance calculations and transfer function bounds on the error probability. Full Viterbi decoder simulations were then performed for just a handful of best candidates from the initial evaluation.

The best code found during the original study was a constraint length 15, rate 1/6 code with connection vectors 46321, 51271, 63667, 70535, 73277, 76513. The original work only cursorily examined codes with the particular combination of parameters $K = 15$, $r = 1/4$, and thus a new code search was needed to recommend an experimental code for Galileo.

Due to the short time constraints on inputs to the project's decision to implement an experimental code, the Galileo code search could not be as thorough as the search conducted during the original work. A decision was made to focus on (15, 1/4) codes whose four connection vectors are subsets of the six connection vectors of the best (15,1/6) code found during the original study. The rationale for this choice is the inverse of the "good codes generate good codes" notion [1] that guided much of the original work. All 15 of these "subset codes" were given a preliminary evaluation using free distance calculations and transfer function bound techniques. A few non-subset codes were also tested, but none performed better than the best of the subset codes.

All codes were tested over a very narrow range of bit-energy to noise-spectral-density ratio (E_b/N_0) between 0.3 dB and 0.5 dB.[3] This range of E_b/N_0 was selected to approximately satisfy Galileo's dual bit error rate requirements of

[3]All values quoted for E_b/N_0 in this report refer to the bit-energy to noise-spectral-density ratio for the convolutional code alone. For concatenated coding, this is equivalent to treating all of the Reed–Solomon code's symbols as information symbols.

5×10^{-3} for the convolutional code alone, and 10^{-5} for the convolutional code concatenated with Galileo's (255,223) 8-bit Reed–Solomon code. All error performance comparisons were based on ideal assumptions, and they do not include allowances for real-system degradations due to radio loss, finite quantization of decoder inputs, or non-ideal interleaving of Reed–Solomon symbols.

The specific idealizing assumptions used for all the Galileo-directed Viterbi decoder simulations were (1) essentially no quantization of metrics (precision limited only by the 64-bit floating point precision of the computer); (2) 128-bit memory path truncation length; and (3) ideal interleaving of Reed–Solomon symbols. Metric quantization is actually 3 bits for the current NASA-standard (7,1/2) MCD (Maximum-likelihood Convolutional Decoder), and it was assumed to be 4 bits for most of the original code search work. For the current study, it was felt that simulating an ideal soft-decision Viterbi decoder (i.e., no quantization of metrics) would produce a fairer comparison of relative code performance, because there was no time to optimize the choice of quantization levels for any tested code. Memory path truncation length is infinite for an ideal Viterbi decoder and 64 bits for the current NASA-standard (7,1/2) MCD. A rough rule of thumb developed for the smaller constraint length codes is that the truncation length should be about 5 times the constraint length [6] in order for the decoder to approach ideal performance. This factor of 5 may have to be increased somewhat for the low signal-to-noise ratios typical of a concatenated system. A 128-bit truncation length was judged adequate to cover constraint lengths through $K = 15$. Galileo's 8-bit Reed–Solomon symbols are interleaved to a depth of only two symbols, and this is grossly insufficient for removing intersymbol dependencies. Average lengths of error bursts from (15,1/4) decoders are approximately 30 bits at Galileo's required performance levels. However, without a good model for estimating Reed–Solomon code performance in the face of intersymbol dependencies, the fairest approach to evaluating relative code performance was to compare all codes on the basis of an ideal interleaving assumption.

The results of the preliminary evaluations of the 15 subset codes are shown in Table 2. The codes are listed in order of predicted merit at $E_b/N_0 = 0.5$ dB, based on an approximate error probability lower bound derived from a transfer function expansion.[4] The table also shows the free distance for each code, a parameter which determines code performance at extremely low error probabilities or high values of E_b/N_0. Somewhat surprisingly, at the low E_b/N_0 required for Galileo, some of the poorest subset codes under the maximum free distance criterion are some of the best codes under the transfer function bound criterion (and vice versa). Unfortunately, neither criterion provides a definitive indicator of relative code performance in this range of E_b/N_0.

Four of the subset codes were selected for further testing via full simulation of the Viterbi decoder. The four selected codes included both of the top-ranked codes under the two criteria discussed above. The Viterbi decoder simulation would provide a definitive comparison of code performance if it could run long enough. Its major shortcoming is the inordinate amount of computer time required for each simulation. Decoding 200,000 bits for one (15,1/4) code at one value of E_b/N_0 consumes about 60 hours of CPU time on a Sun 3/260 computer.

The results of the full simulation tests of the four candidate subset codes are shown in Table 3. The table also lists full simulation results for two non-subset codes. One of these was suggested in an earlier study [3] as a perturbation of the subset code with maximum free distance, and the other is derived from a good (13,1/4) code.

The codes in Table 3 were tested at the low end of the desired range in E_b/N_0 (0.3 dB) in order to maximize the number of error samples for a valid statistical comparison. For comparison purposes, all codes were subjected to the same extremely long sequence of 800,000 noise samples.

Table 3 lists both the simulated bit error rate and the simulated symbol error rate for 8-bit Reed–Solomon symbols. The symbol error rate is the predominant factor that determines the performance of each candidate convolutional code when concatenated with Galileo's Reed–Solomon outer code. Symbol error rates of 2.1×10^{-2} to 2.5×10^{-2} correspond to concatenated code bit error rates of 10^{-6} to 10^{-5}, under the assumption of ideal interleaving of Reed–Solomon symbols.

Each estimate of bit error rate or symbol error rate in Table 3 was based on 200,000 decoded bits. Due to the propensity of convolutional decoder errors to occur in long bursts, each (15,1/4) code simulated at 0.3 dB only produced slightly more than 100 independent error bursts. The corresponding 1σ accuracy in the bit error rate and the symbol error rate estimates is about 10 to 15 percent.

The two subset codes that were top-ranked according to one of the two criteria for preliminary code evaluation produced more simulated bit errors and more simulated symbol

[4] D. Divsalar and S. Dolinar, "All $K = 15$, $r = 1/4$ Codes Derived From the Best $K = 15$, $r = 1/6$ Code," JPL IOM 331-87.2-324 (internal document), Jet Propulsion Laboratory, Pasadena, California, July 27, 1987.

errors than the two tested subset codes that were not top-ranked on either preliminary scale. The performance of the latter two codes could not be distinguished with any degree of statistical significance; the code with the higher free distance (d_{free} = 35) was nominated for the original experimental encoder breadboard design and is the code ultimately selected for implementation on Galileo. The two non-subset codes listed in Table 2 did not fare as well as the two best subset codes. Approximately a dozen additional non-subset codes suggested from various sources were partially simulated, but these simulations were aborted long before the decoding of 200,000 bits when it became apparent that their performance was inferior to that of the recommended code.

A programming glitch was discovered in the random number generator which was used for the results in Table 3. It is not known whether the inadvertently programmed random number generator has desirable statistical properties. Two of the codes listed in Table 3 were retested with a properly reprogrammed random number generator. These two codes were the recommended code for Galileo and the code with the largest free distance. The results of the tests with the new random number generator are shown in Table 4. The table shows simulation results at E_b/N_0 = 0.3 dB and 0.5 dB for both codes, and at E_b/N_0 = 0.1 dB and 0.7 dB for the recommended code only. The error rate estimates are based on 200,000 decoded bits for E_b/N_0 = 0.3 dB and 0.5 dB; 100,000 bits for E_b/N_0 = 0.1 dB; and 400,000 bits for E_b/N_0 = 0.7 dB. The corresponding 1σ accuracy in the error rate estimates is about 10 to 15 percent for E_b/N_0 = 0.1 dB and 0.3 dB, and 15 to 20 percent for E_b/N_0 = 0.5 dB and 0.7 dB.

The recommended code won all comparative simulation tests for bit error rate performance with both random number generators, though many differences were not statistically significant individually. It won all but two symbol error rate comparisons, losing by statistically insignificant margins to one code in Table 3 and a different code at one value of E_b/N_0 in Table 4. Bit error rate comparisons were weighted more heavily than symbol error rate comparisons, due to Galileo project emphasis on improving non-concatenated performance, and due to the dubious validity of the ideal interleaving assumption that was necessary to directly link convolutional code symbol error rates with concatenated code bit error rates. All in all, the recommended code seems to be a solid though not absolutely optimum code that will function well in its intended application on Galileo.

VI. Comparison With the Current NASA Standard (7,1/2) Code

Table 4 also lists simulation test results obtained with the reprogrammed random number generator for the current NASA-standard (7,1/2) convolutional code at values of E_b/N_0 ranging from 1.05 dB to 2.55 dB. These error rate estimates are based on the same idealizing assumptions used for simulating the (15,1/4) codes, and they are several tenths of a decibel more optimistic than actual MCD (Maximum-likelihood Convolutional Decoder) performance. The table shows that at Galileo's desired bit error rate of 5×10^{-3} or symbol error rate of 2.5×10^{-2}, the recommended (15,1/4) code offers about 1.5 dB of improvement over the NASA-standard (7,1/2) code under the same ideal circumstances for both codes.

VII. Galileo Experimental Code Specification

The recommended experimental code for Galileo has constraint length 15 (memory length 14) and rate 1/4. The connection vectors for the four output symbols per input information bit are represented in octal form as 46321, 51271, 63667, 70535. Equivalently, in binary form these are 100110011010001, 101001010111001, 110011110110111, 111000101011101. Each of the four output binary code symbols per input information bit is obtained as a modulo-2 sum of the current information bit and a subset of the 14 previous information bits corresponding to the positions of the 1's in the appropriate connection vector. The leftmost connection vector bit indicates a connection to the current information bit, and each succeeding bit indicates a connection or non-connection to the next most recent information bit.

A connection diagram for the Galileo code is shown in Fig. 4. In addition to the basic code connections, this diagram also indicates that the first and third output symbols should be inverted. The alternate symbol inversion has no effect on code performance and is included solely to aid symbol synchronization by ensuring that the encoder will not produce long runs of 0's or 1's in response to constant input. Alternate symbol inversion avoids arbitrarily long runs of all 0's or all 1's in response to constant input as long as the parity of successive connection vectors does not alternate odd-even-odd-even [7]. Since the experimental code has three connection vectors with odd parity and one with even parity, the parity cannot alternate odd-even-odd-even, and the ordering of the output symbols is inconsequential.

VIII. Postscript: An Issue of Code Transparency

Shortly after the choice of experimental code for Galileo, the question of code transparency arose with respect to the selected code. The present NASA-standard (7,1/2) convolutional code has a useful property that makes it transparent to inversion of input data bits. A binary code is transparent to input inversion if complementing every bit in the input data stream causes every symbol in the output encoded symbol stream to be complemented as well. A necessary and sufficient condition for the transparency of a rate $1/N$ convolutional code is that all N of its connection vectors have odd parity.

The code transparency feature is useful because a maximum-likelihood decoder encountering an accidentally inverted, transparently coded symbol stream will decode the exact complement of the maximum-likelihood data stream, and the true maximum-likelihood data bits can be easily recovered by removal of an overall binary ambiguity. In contrast, an accidentally inverted symbol stream encoded by a non-transparent code will produce decoded output from which the true maximum-likelihood data bits cannot be directly recovered.

Galileo's biphase modulated symbol stream possesses an inherent sign ambiguity that must be resolved at some point. If the symbols are encoded by a non-transparent code, ambiguity removal must precede decoding. Transparently coded symbols, on the other hand, may be decoded either before or after the ambiguity is resolved.

Code transparency was not a constraining criterion during either the original extensive code search or the limited Galileo-directed search reported here. Transparent codes are the exception rather than the rule. For example, only one-sixteenth of all (15,1/4) convolutional codes are transparent. As a result, not many tested long constraint length codes happen to be transparent. Not one of the (15,1/4) codes evaluated during the Galileo-directed search is transparent. In particular, the recommended experimental code choice for Galileo is not transparent.

A transparent code was constructed by making a tiny modification to the recommended experimental code. The motivation for this particular modification was a potential small performance advantage suggested by certain symmetry considerations as applied to the first three bits and last three bits of the recommended code's connection vectors. The transparent code differs from the recommended code by just one bit in one of its four connection vectors. The connection vectors for the transparent code are 46321, 51273, 63667, 70535.

The transparent code has been subjected to simulation tests under the same ideal assumptions that applied to the other codes tested and reported above, using the reprogrammed random number generator only. The simulation results for the transparent code are listed in Table 5 alongside the comparable results for the non-transparent recommended Galileo experimental code.

The transparent code outperformed the recommended code at two of four tested values of E_b/N_0 if measured by simulated Reed–Solomon symbol error rate, or at one of four values of E_b/N_0 if measured by simulated bit error rate. No other code previously tested had ever outscored the recommended code in more than one category. However, all differences in simulated error rates for the two codes were statistically insignificant.

Proper perspective for interpreting the simulation results can be obtained only by evaluating the raw estimated error rates with respect to the statistical uncertainties in the simulation predictions. Figures 5 and 6 plot the simulated bit and symbol error rates for the transparent and non-transparent codes. In these figures, the performance traces for both codes are plotted as wide swaths indicating $\pm 1\sigma$ uncertainty around the nominal error rate estimates listed in Table 5. Estimates of the statistical uncertainty of the simulation results are based on a geometric model for Viterbi decoder error bursts [5].

The plots in Figs. 5 and 6 show no discernible performance difference between the transparent and non-transparent codes over the tested range of E_b/N_0. All of the small numerical differences in Table 5 pale in significance next to the statistical uncertainties inherent in the simulation. Furthermore, even these small differences tend to cancel each other for these two codes. Each code's performance swaths are intertwined both above and below the other's over the full tested range of E_b/N_0. The bottom-line conclusion to be drawn from Figs. 5 and 6 is that the Galileo project's choice between the transparent and the non-transparent code had to be based on factors other than simulated code performance. The decision of the Galileo project was that transparency was an insufficient reason to change the selected experimental code.

IX. Summary

The Galileo experimental code search resulted in the code selection depicted in Fig. 4. The code parameters were chosen to optimize performance within cost and risk constraints consistent with retrofitting the new code into the existing Galileo system design and launch schedule. The particular code was recommended after a very limited search among good codes with the chosen parameters. The recommended code is a solid though not absolutely optimum code that will function well in its intended application on Galileo. It will theoretically yield about 1.5 dB of improvement under idealizing assumptions relative to the current NASA-standard code at Galileo's desired bit error rates. This ideal predicted gain includes enough cushion to meet the target of at least 1 dB of enhancement under real, non-ideal conditions. The (non-transparent) experimental code can be trivially modified to form a transparent code essentially equivalent in performance, but this modification was not approved for Galileo.

References

[1] P. J. Lee, "New Short Constraint Length, Rate $1/N$ Convolutional Codes Which Minimize Required E_b/N_0 for Given Bit Error Rate," *TDA Progress Report 42-77*, January–March 1984, Jet Propulsion Laboratory, Pasadena, California, pp. 41–56, May 1984.

[2] P. J. Lee, "A Very Efficient Transfer Function Bounding Technique on Bit Error Rate for Viterbi Decoded, Rate $1/N$ Convolutional Codes," *TDA Progress Report 42-79*, July–September 1984, Jet Propulsion Laboratory, Pasadena, California, pp. 114–123, November 1984.

[3] J. H. Yuen and Q. D. Vo, "In Search of a 2-dB Coding Gain," *TDA Progress Report 42-83*, July–September 1985, Jet Propulsion Laboratory, Pasadena, California, pp. 26–33, November 1985.

[4] P. J. Lee, "Further Results on Rate $1/N$ Convolutional Code Constructions with Minimum Required SNR Criterion," *IEEE Transactions on Communications*, vol. COM-34, no. 4, pp. 395–399, April 1986.

[5] R. L. Miller, L. J. Deutsch, and S. A. Butman, *On the Error Statistics of Viterbi Decoding and the Performance of Concatenated Codes*, JPL Publication 81-9, Jet Propulsion Laboratory, Pasadena, California, September 1, 1981.

[6] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*, New York: McGraw-Hill, pp. 258–259, 1979.

[7] L. D. Baumert, R. J. McEliece, and H. C. A. van Tilborg, "Symbol Synchronization in Convolutionally Coded Systems," *IEEE Transactions on Information Theory*, vol. IT-25, no. 3, pp. 362–365, May 1979.

Table 1. The quest for coding gain

Code	Required E_b/N_0 for 10^{-6} BER, dB	Coding non-optimality relative to Shannon limit, dB	Coding gain relative to current concatenated code, dB
Uncoded	10.54	12.13	–
Current convolutional (7,1/2)	4.78	6.37	–
Current concatenated (7,1/2) + 8-bit R–S	2.53	4.12	0.00
Code search best code (15,1/6) + 10-bit R–S	0.42	2.01	2.11
Shannon limit (unknown code of arbitrary complexity)	−1.59	0.00	4.12

Table 2. Preliminary evaluation of subset codes

Code connection vectors	d_{free}	Approximate BER bound at 0.5 dB
(51271, 70535, 73277, 76513)*	32	4.9×10^{-4}
51271, 63667, 70535, 76513	34	5.4×10^{-4}
(51271, 63667, 70535, 73277)*	34	5.4×10^{-4}
(46321, 51271, 63667, 70535)*	35	5.7×10^{-4}
46321, 51271, 63667, 76513	36	6.2×10^{-4}
46321, 63667, 70535, 73277	35	6.2×10^{-4}
46321, 51271, 70535, 76513	34	6.3×10^{-4}
46321, 51271, 70535, 73277	34	6.4×10^{-4}
51271, 63667, 73277, 76513	35	6.8×10^{-4}
46321, 51271, 73277, 76513	36	7.3×10^{-4}
63667, 70535, 73277, 76513	34	7.7×10^{-4}
46321, 63667, 70535, 76513	37	7.7×10^{-4}
46321, 63667, 73277, 76513	36	8.2×10^{-4}
46321, 70535, 73277, 76513	36	8.3×10^{-4}
(46321, 51271, 63667, 73277)*	38	8.6×10^{-4}

*Codes selected for full simulation.

Table 3. Simulation results for candidate codes

Code connection vectors	d_{free}	Simulated bit error rate at 0.3 dB	Simulated symbol error rate at 0.3 dB
(46321, 51271, 63667, 70535)*†	35	1.06×10^{-2}	2.45×10^{-2}
(51271, 63667, 70535, 76513)*	34	1.07×10^{-2}	2.37×10^{-2}
(46321, 51271, 63667, 73277)*	38	1.11×10^{-2}	2.60×10^{-2}
46321, 51271, 63667, 73257	37	1.12×10^{-2}	2.64×10^{-2}
45177, 53365, 62313, 75655	35	1.16×10^{-2}	2.67×10^{-2}
(51271, 70535, 73277, 76513)*	32	1.20×10^{-2}	2.74×10^{-2}

*Subset codes from Table 2.
†Recommended (15, 1/4) experimental code for Galileo.

Table 4. Simulation results with reprogrammed random number generator

Code connection vectors	d_{free}	Simulated bit error rate	Simulated symbol error rate
(46321, 51271, 63667, 70535)†	35	1.77×10^{-2} @ 0.1 dB	4.22×10^{-2} @ 0.1 dB
		9.5×10^{-3} @ 0.3 dB	2.24×10^{-2} @ 0.3 dB
		4.2×10^{-3} @ 0.5 dB	1.08×10^{-2} @ 0.5 dB
		2.2×10^{-3} @ 0.7 dB	5.1×10^{-3} @ 0.7 dB
46321, 51271, 63667, 73277	38	9.5×10^{-3} @ 0.3 dB	2.16×10^{-2} @ 0.3 dB
		4.9×10^{-3} @ 0.5 dB	1.16×10^{-2} @ 0.5 dB
(171, 133)*	10	3.68×10^{-2} @ 1.05 dB	8.53×10^{-2} @ 1.05 dB
		1.37×10^{-2} @ 1.55 dB	3.41×10^{-2} @ 1.55 dB
		1.11×10^{-2} @ 1.65 dB	2.77×10^{-2} @ 1.65 dB
		9.4×10^{-3} @ 1.75 dB	2.37×10^{-2} @ 1.75 dB
		7.6×10^{-3} @ 1.85 dB	1.91×10^{-2} @ 1.85 dB
		5.9×10^{-3} @ 1.95 dB	1.51×10^{-2} @ 1.95 dB
		4.4×10^{-3} @ 2.05 dB	1.14×10^{-2} @ 2.05 dB
		1.3×10^{-3} @ 2.55 dB	3.2×10^{-3} @ 2.55 dB

*Current NASA standard (7,1/2) code.
†Recommended (15,1/4) experimental code for Galileo.

Table 5. Performance comparison of Galileo experimental code (non-transparent) and modified code (transparent)

Code connection vectors	d_{free}	Simulated bit error rate	Simulated symbol error rate
(46321, 51271, 63667, 70535)†	35	1.77×10^{-2} @ 0.1 dB 9.5×10^{-3} @ 0.3 dB 4.2×10^{-3} @ 0.5 dB 2.2×10^{-3} @ 0.7 dB	4.22×10^{-2} @ 0.1 dB 2.24×10^{-2} @ 0.3 dB 1.08×10^{-2} @ 0.5 dB 5.1×10^{-3} @ 0.7 dB
(46321, 51273, 63667, 70535) *	36	1.87×10^{-2} @ 0.1 dB 8.5×10^{-3} @ 0.3 dB 4.6×10^{-3} @ 0.5 dB 2.3×10^{-3} @ 0.7 dB	4.45×10^{-2} @ 0.1 dB 2.04×10^{-2} @ 0.3 dB 1.07×10^{-2} @ 0.5 dB 5.3×10^{-3} @ 0.7 dB

*Modified (15,1/4) code (transparent).

†Recommended (15,1/4) experimental code for Galileo (non-transparent).

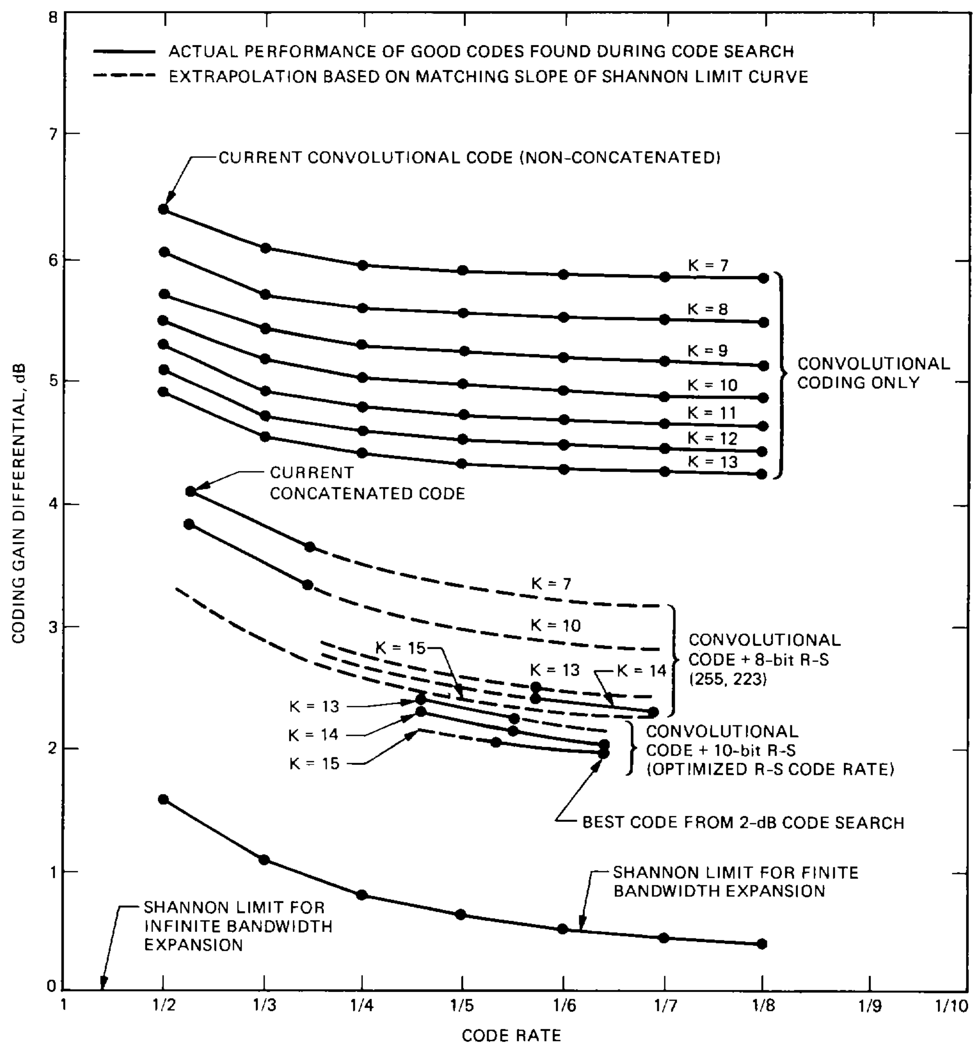

Fig. 1. Coding gain differential relative to the Shannon limit for infinite bandwidth expansion at a bit error rate of 10^{-6}

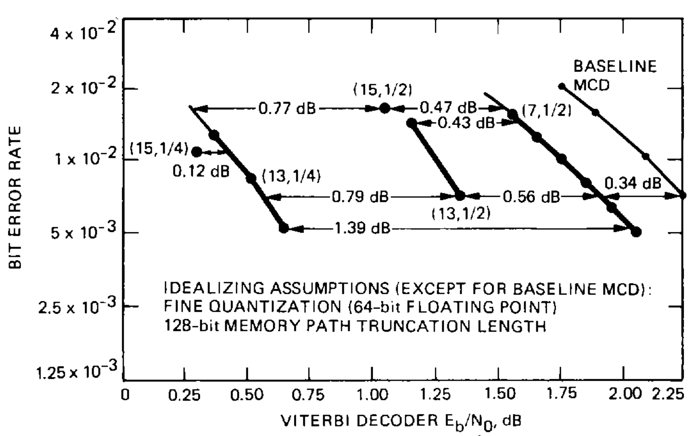

Fig. 2. Best codes found during code parameter selection phase of study: performance of convolutional codes alone (unconcatenated)

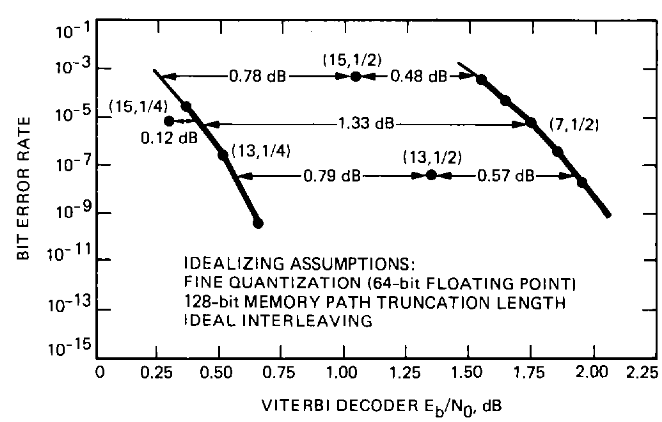

Fig. 3. Best codes found during code parameter selection phase of study: performance of convolutional codes concatenated with (255,223) Reed-Solomon code

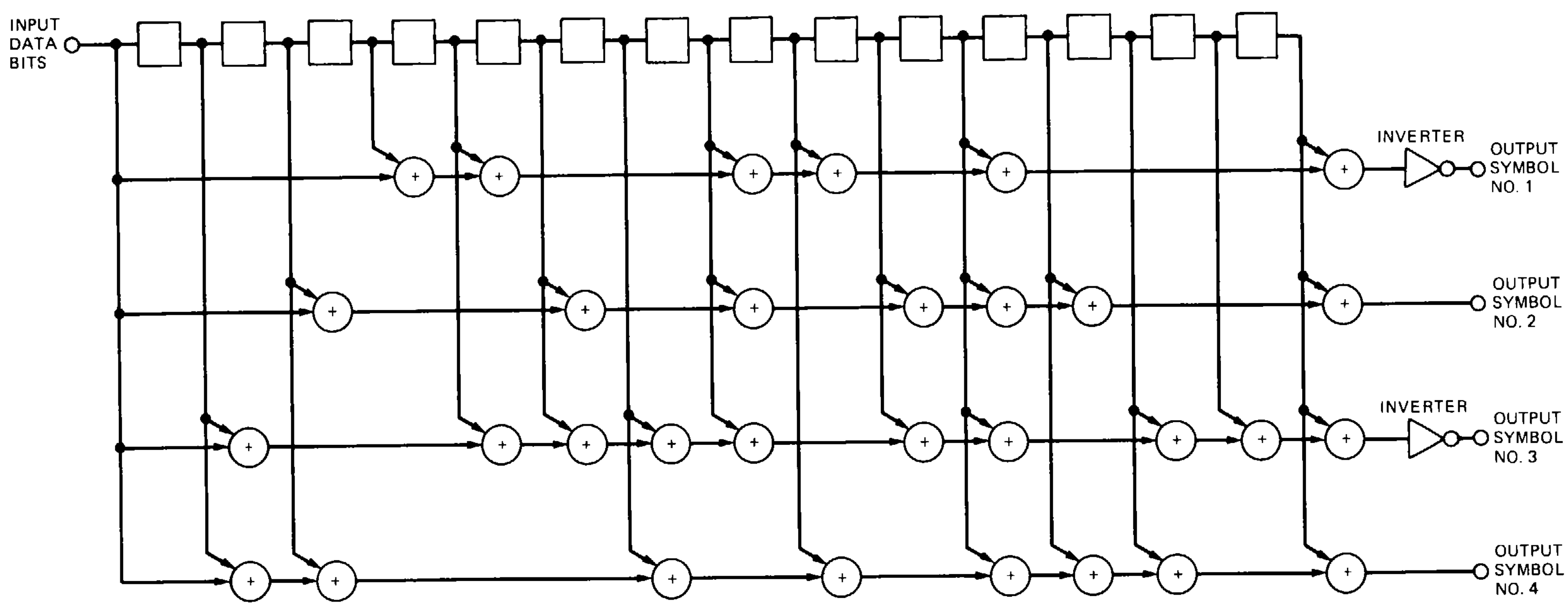

Fig. 4. Encoder connection diagram for recommended Galileo experimental code

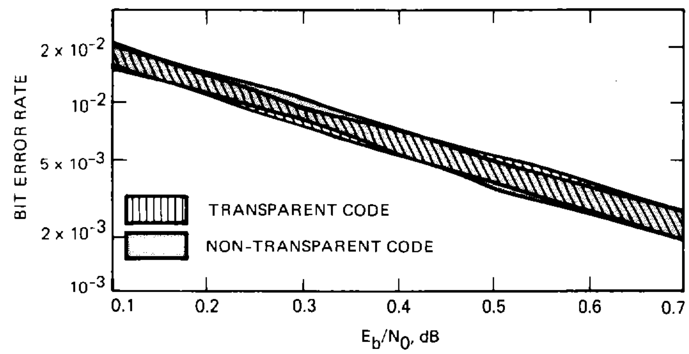

Fig. 5. Simulated bit error rate for Galileo experimental code alternatives (± 1-sigma uncertainty swath indicated for both codes)

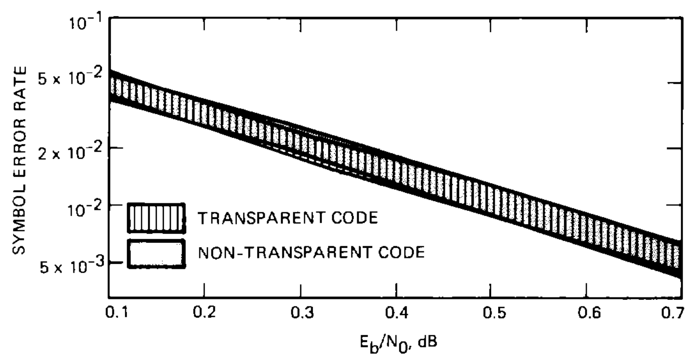

Fig. 6. Simulated symbol error rate for Galileo experimental code alternatives (± 1-sigma uncertainty swath indicated for both codes)