

Reed-Solomon Codes and the Exploration of the Solar System*

by

Robert J. McEliece and Laif Swanson

1. Introduction.

The exploration of the solar system by unmanned spacecraft is one of the triumphs of the 20th century. The dramatic photographs of Mercury, Venus, Mars, Jupiter, Saturn, Uranus, and Neptune transmitted by spacecraft with romantic names like *Mariner*, *Voyager*, *Viking*, etc. over distances of hundreds of millions, even billions, of miles, have made these planets, which were previously known to us only as fuzzy telescopic images in textbooks, as real to us in the 1990's as, say, the Himalayas, the Sahara Desert, or Antarctica.

In a book devoted to Reed-Solomon codes, it is surely appropriate to include a chapter on deep space applications, since error-control coding in general, and Reed-Solomon coding in particular, has been part of the communications technology of planetary exploration, almost from the beginning. So in this article, we will trace the use of RS codes in space applications from “prehistoric” times (about 1970) to the present -- and into the future!

First, some general remarks. In deep-space communication, the channel is, to a very close approximate ion, a power-limited, wideband, additive Gaussian channel. (See [36], Chapter 2, or [30], Chapter 4, for good descriptions of this model.) RS codes are not effective *directly* on such a channel, for two main reasons. First, RS codes are best at combatting bursts of errors, but the Gaussian channel is memory less. Second, there is no known practical way to “soft decode” RS codes, and the 2 dB loss resulting from hard quantization prior to decoding is intolerable in all but a few applications.

Nevertheless, RS codes can be effectively used indirectly on the space channel, as “outer” codes in concatenated systems. The general idea of concatenation, which was introduced by Forney in 1967 [15], is shown in Figure 1. The idea is to use an “inner” encoder-decoder pair directly adjacent to the unreliable channel over which reliable communication must be achieved. No matter how well it is designed, the inner decoder will occasionally make errors, which will normally be bursty and hard-quantized. It is the job of the outer code to correct these errors. As Forney was the first to observe, Reed-Solomon codes are ideal choices for the outer code, since they are naturally able to correct complex,

* This chapter was written at the California Institute of Technology's Jet Propulsion Laboratory, under contract with the National Aeronautics and Space Administration.

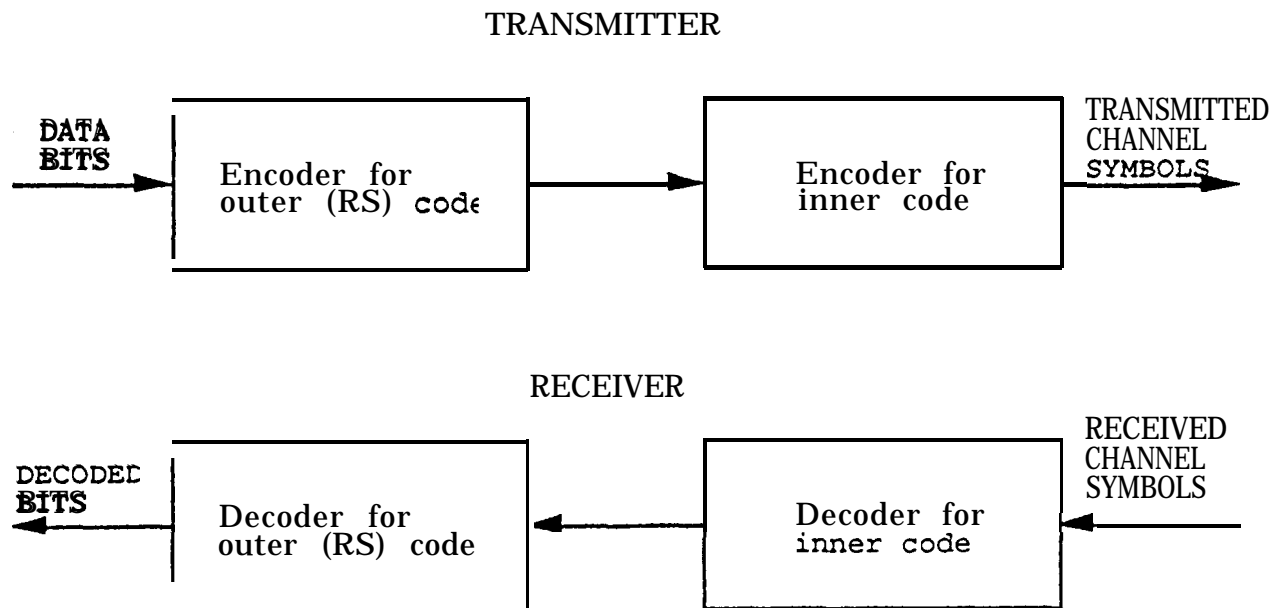


Figure 1. A General Concatenated Coding System.

bursty, error patterns. They turn out to be so ideal, in fact, that other codes are rarely used as outer codes in concatenated systems.

Shortly after the appearance of Forney's work, Odenwaller, under the direction of Viterbi [22], realized that concatenation could be used to great advantage on the space channel, if the inner code is a convolutional code decoded with Viterbi's algorithm, and the outer code is RS (possibly with the addition of an interleaver), as shown in Figure 2. In Figure 3, we see typical performance curves for the space channel that illustrate the advantages of using RS codes. Without the outer RS code (the "unconcatenated" curve in Figure 3), the tradeoff between bit signal-to-noise ratio and decoder error probability is relatively shallow, whereas when the outer RS code is added to the system, the resulting curve is comparatively steep, so that if a decoded bit error probability of 10^{-5} or less is required, the concatenated system is markedly superior to the unconcatenated one. Note however, that if the desired bit error probability is 10^{-3} or greater, concatenation offers no significant advantage. As we will see below, this fact, together with the fact that *uncompressed images*, which until recently comprised the bulk of the data returned by planetary probes, are usually acceptable if the bit error probability is 10^{-2} or less, explains why RS codes made a relatively late appearance in space communication systems.

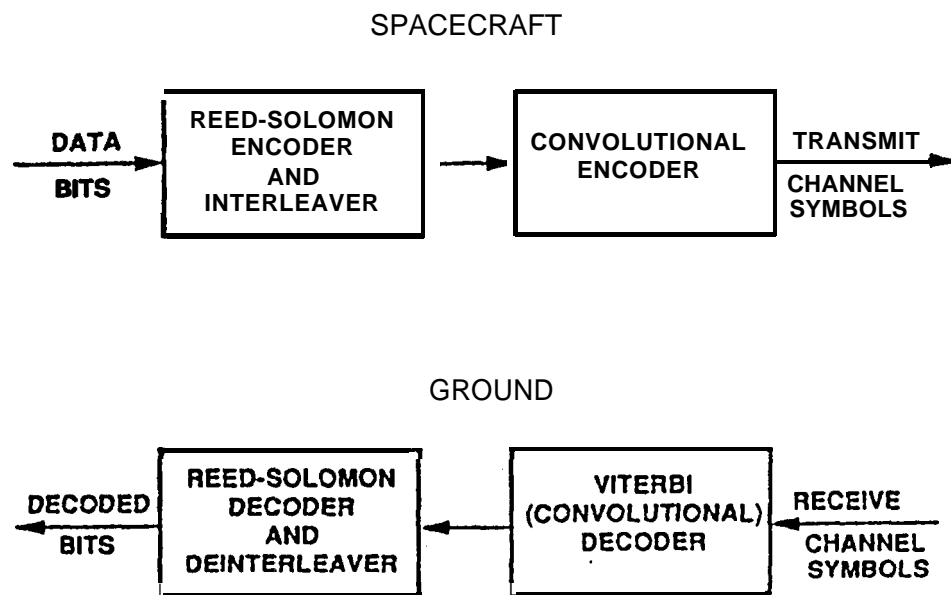


Figure 2. An Odenwalder-Viterbi Concatenated Coding System for the Space Channel

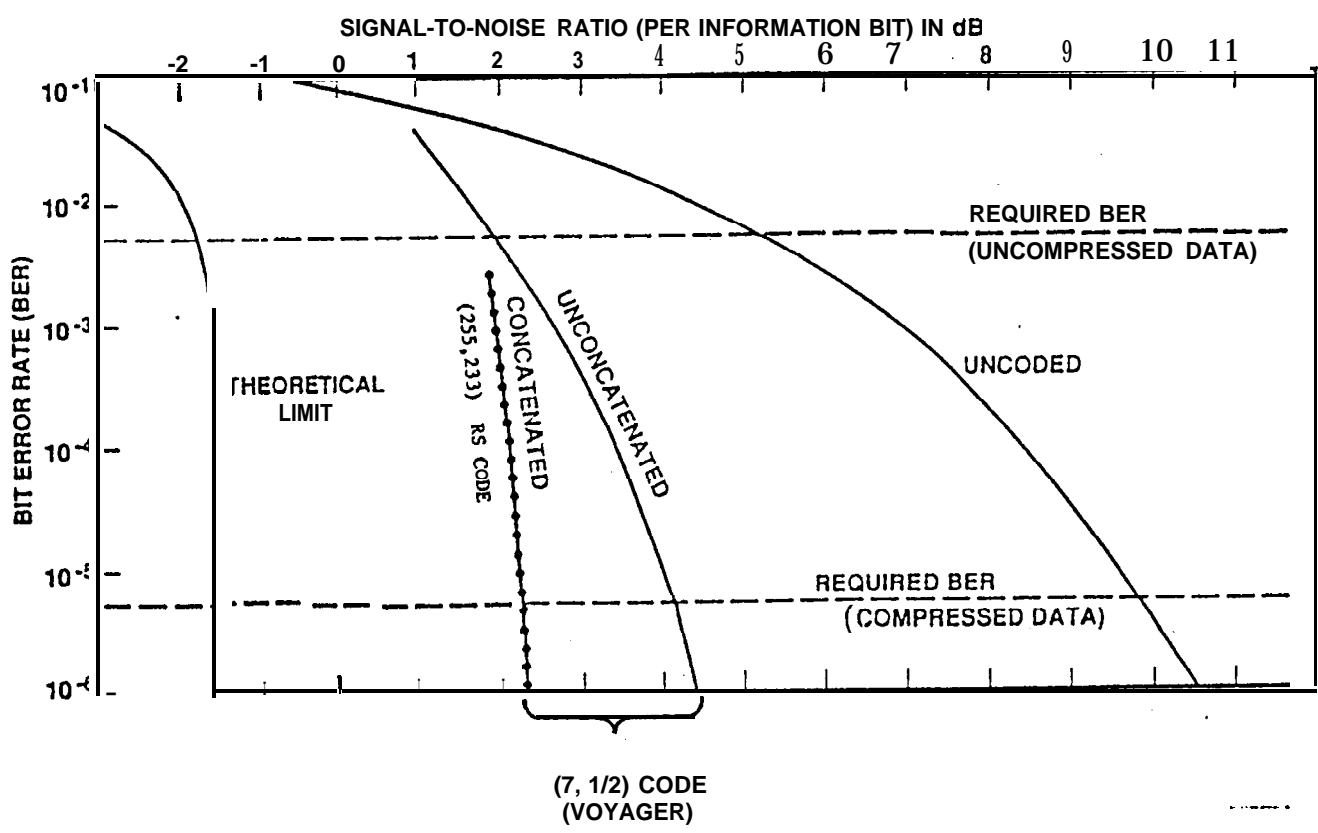


Figure 3. Typical Performance Curves for Concatenated and Unconcatenated Coding Systems for the Space Channel.

2. Prehistory — the 1971 Mariner Mars Mission.

If we stretch our definitions a little, we can argue that the first space application of RS codes was on NASA's 1971 *Mariner* Mars orbiter mission, which was launched on May 30, 1971. On that mission, the main downlink code was the (32,6) biorthogonal code, which was decoded using a fast Hadamard transform, or "Green Machine" decoder [29]. The bulk of the data returned by Mariner 71 was in the form of digital images of the surface of Mars, for which a decoded error probability of 5×10^{-3} was acceptable. However, the spacecraft also returned data from another experiment, the infrared interferometer spectrometer (IRIS), which required a bit error probability almost two orders of magnitude smaller. Since the IRIS data comprised only a small fraction of the total data delivered by Mariner, it obviously would have been wasteful to transmit the entire data stream at a bit SNR large enough to produce a decoder error probability of 5×10^{-5} . A solution to this dilemma, devised by Dorsch and Miller [14], was to use concatenation only on the IRIS data. Their idea was to use a concatenation system of the general type depicted in Figure 1, with the inner code being the (32, 6) biorthogonal code, and the outer code being a (6,4) RS code over $GF(2^6)$. Dorsch and Miller rightly called the outer code a "generalized Hamming code," since with redundancy 2 it could only correct one error, but they also observed that "the described code is a specific case of a Reed-Solomon code," which is of course also true. Thus in a sense, this concatenated system was the first space-borne use of Reed-Solomon codes. However, the first "full-blown" use of RS concatenated coding on a space mission was on the *Voyager* mission, which we describe in the next section.

3.1. The Voyager Mission–History.

Multi-error-correcting Reed-Solomon codes were used for the first time in deep-space exploration in the spectacularly successful *Voyager* mission, which began in the Summer of 1977 with the launch of twin spacecraft (Voyager 1 and Voyager 2) from Cape Kennedy, towards the outer planets Jupiter and Saturn. (See Murray [21] for an insider’s reminiscences about this historic mission.) Earlier deepspace missions like *Pioneer*, *Mariner*, *Viking*, and indeed *Voyager* itself at Jupiter and Saturn, used sophisticated error-correction but had no need for Reed-Solomon codes, because their digital images were not *compressed* prior to transmission. At Uranus and Neptune, however, *Voyager*, transmitted some (though not all) of its images in compressed format, which made RS coding essential. Let us see why this was so.

A *Voyager* full-color image is digitized by the spacecraft’s imaging hardware into three 800 x 800 arrays of eight-bit, pixels, or $3 \times 800 \times 800 \times 8 = 15,360,000$ bits. In an *uncompressed* spacecraft telecommunication system, these bits are transmitted, one by one, to earth, where the image is reconstructed. Of course, if some of the received bits are in error, the quality and scientific usefulness of the image is degraded, and early studies by planetary scientists established 5×10^{-3} as the maximum bit error probability acceptable for images from NASA planetary missions.* Thus when telecommunications engineers designed the error-control coding for these missions, they invariably sought to maximize the “coding gain” at a decoded bit error probability of 5×10^{-3} . For example, in the baseline *Voyager* telecommunication system, which uses a $K = 7$, rate 1/2 convolutional code (originally suggested by Odenwalder [22], [23]), the coding gain at $P_b = 5 \times 10^{-3}$ is about 3.5 dB (see Figure 3).

Odenwalder (op.cit.) also showed that by concatenating the $K = 7$, rate 1/2 code with an outer RS code, as shown in Figure 2, the coding gain for low bit error rates can be improved considerably. For example, in Figure 3, we see the comparative performance of the baseline (7, 1/2) convolutional code and the same code, concatenated with a (255, 223) RS code (assuming an interleaving depth large enough that the RS symbol errors can be assumed independent). We see that while the concatenated system is only slightly superior to the baseline system at $P_b = 5 \times 10^{-3}$ (about 0.2 dB), at smaller values of P_b , the concatenated system is markedly superior. For example, at $P_b = 10^{-6}$, the concatenated system is about 2.5 dB superior, which implies that at $P_b = 10^{-6}$, the concatenated system can transmit at a 78% higher data rate than the baseline system. However, as we have seen, planetary missions only required $P_b = 5 \times 10^{-3}$, and so these potential gains at lower values of P_b were apparently of no practical value. Shortly after Odenwalder’s work appeared, however, an important breakthrough in data compression occurred that changed

* Unaccountably, in 1977 Murray and Burgess [20], recalling the 1973 *Mariner 10* mission to Mercury, wrote that “Three errors in 100 bits had been established by the imaging team years earlier as an acceptable level.” Presumably it should read “three errors in 1000 bits.”

this situation dramatically.

It had been realized since the early 1960s that planetary images were extremely redundant, and that far fewer than 15 million bits should suffice to represent one of them. However, the known techniques for reducing this redundancy were too complex to be implemented onboard a spacecraft. But this situation changed in the early 1970's, when Robert Rice at Caltech's Jet Propulsion Laboratory devised a data compression algorithm that typically compressed a planetary image by a factor of 2.5, with no loss of fidelity, and which was simple enough to be implemented in Voyager's software (see [31], [32], [33], [34]). (In modern data compression parlance, Rice's algorithm could be called "line-by-line adaptive entropy coding of the pixel differences.") A factor of 2.5 achieved by data compression translates to 4 dB in system gain, a figure which would be difficult to obtain in any other way. Still, conservative spacecraft engineers judged the Rice algorithm too risky for the all-important basic mission to Jupiter and Saturn, although they were willing to include it as part of a backup system in case the primary communication link failed, and as a way of enhancing the hoped-for "extended mission" to distant Uranus and Neptune. Even so, there was a stumbling block.

The stumbling block was that Rice's *decompression* algorithm, like most decompression algorithms, is quite sensitive to bit errors. If a compressed line contains even one bit error, Rice's algorithm will, as a rule, garble the line beyond recognition. Thus it was determined that the venerable value of $P_b = 5 \times 10^{-3}$ was no longer acceptable; a much lower value was required, a value that could only be achieved efficiently using concatenation with RS codes, as prescribed by Odenwalder! After considerable study, which took into account the fact that decoder errors, when they occur, tend to occur in bursts, it was determined that $P_b = 10^{-6}$ was necessary for Rice-compressed planetary images. A glance at Figure 3 shows that the RS concatenated system requires an E_b/N_0 of 2.8 dB to achieve a $P_b = 10^{-6}$, whereas the baseline $K = 7$, rate 1/2 system requires 2.6 dB for $P_b = 5 \times 10^{-3}$. Thus the net energy cost of going from $P_b = 5 \times 10^{-3}$ to $P_b = 10^{-6}$ is 0.2 dB, which means that using the Rice compression algorithm together with the concatenated RS/convolutional system results in a net gain of about 3.8 dB over the baseline system. Indeed, this system was implemented on *Voyager*, and in Figure 4 we see the first deep-space photograph ever sent using Reed-Solomon technology.

In the next section, we outline the operational details of *Voyager's* RS-enhanced coding system.

Figure 4. Uranus, as seen through the eyes of Irv and Gus,
April 5, 1985, from a distance of over 200 million miles.

3.2. The Voyager Mission. Operational Details.

The *Voyager* RS code is a (255, 223) code over the field $\text{GF}(2^8)$. The field is represented by the primitive polynomial $m(x) = x^8 + x^4 + x^3 + x^2 + 1$, which is the first degree 8 polynomial listed in the famous tables of Peterson-Weidon ([27]). The generator polynomial for the code is

$$g(x) = (x - \alpha)(x - \alpha^2) \cdots (x - \alpha^{32}),$$

where α is a zero of $m(z)$. The code is interleaved to a depth of 4, which means that the overall performance will not be quite as good as that shown in Figure 3, which is for infinite interleaving. A careful study, however, shows that for this system, the loss due to finite interleaving is only 0.02 dB at a decoder error probability of 10^{-6} [6].

The *Voyager* on-board RS encoder is a special-purpose hardware device, built from several dozen SS1 space-qualified CMOS parts [11]. In essence, it is a hardware implementation of the usual systematic shift-register encoder as depicted, say, in [10], sec. 4.3, requiring 32 hard-wired Galois field multipliers, corresponding to the coefficients of $1, z, \dots, z^{31}$ in $g(x)$.

The *Voyager* ground-based decoder was built by Charles Lahmeyer of the Jet Propulsion Laboratory ([17]). It implements the RS decoding algorithm outlined in [18], table 8.6. In particular, it uses the Euclidean algorithm to solve the key equation, then a Chien search to locate the errors, and finally the usual " $w(z) / \sigma'(x)$ " formula to evaluate the errors. The actual physical device is a special-purpose circuit built from discrete-component TTL logic, primarily 74S Schottkey logic. The decoder uses no microprocessors. Instead, processing is done by dedicated "nanosequencers," which are special purpose microprogrammable controllers. During the Uranus encounter and Neptune encounters, the *Voyager* data rate was a maximum of 44.8 Kbps, but the decoder was capable of running at speeds up to one megabit/sec. [17].

4.1. The Galileo Mission—History

Galileo is a two-and-a-half ton NASA spacecraft which was launched towards Jupiter in October 1989. It will arrive in late 1995, and will then begin a two-year study of the Jovian atmosphere, satellites, and surrounding magnetosphere. In the summer of 1995, a probe will detach itself from the main body of the spacecraft, and in December 1995 this probe will plunge into Jupiter's stormy atmosphere, from which it will bravely return data until its inevitable destruction a few hours later. Much, though not all, of *Galileo's* data will be protected by Reed-Solomon codes.

Unfortunately, the *Galileo* mission has suffered two major operational misfortunes, both of which have caused engineers to make significant alterations in the communications technology. The first of these was the Challenger disaster (January 28, 1986), which delayed the launch of *Galileo* by more than three years. The coding system on the pm-*Challenger Galileo* mission was virtually identical to that of *Voyager*, viz. a $K = 7$, rate $1/2$ convolutional code concatenated with a $(255, 223)$ RS code over $GF(256)$. The only significant difference was that for unavoidable engineering reasons, the interleaving depth on Galileo was only 2, vs. 4 for Voyager.

The launch delay and propulsion restrictions caused by the *Challenger* accident resulted in both a longer travel time to Jupiter, and a less favorable planetary geometry. Because of this, there was the potential for a significant loss of data return. This potential loss was however partially compensated for by a last-minute decision by spacecraft engineers to include an enhanced error-correction system on the Galileo spacecraft. Instead of the Voyager-like, NASA standard $K = 7$, rate $1/2$, convolutional code, a $K = 15$, rate $1/4$ code was proposed, adopted, and a corresponding encoder was built into the spacecraft prior to launch [12], [35]. But no changes were made to the RS part of the coding system, and so we will not discuss this interesting system further.

However, a further calamity was to befall Galileo, and RS codes were heavily involved in this story, which we shall tell in the next section.

4.2. The Galileo 'S-Band Mission.'

In April, 1991, the high-gain, or "X-band" spacecraft antenna, which had been "furled" like an umbrella for launch, failed to unfurl properly when commanded to do so. Repeated attempts to open the antenna failed, and mission managers declared the X-band antenna dead.

Because of the failure of Galileo's high-gain antenna, it became necessary to use the low-gain, or S-band antenna, whose gain was 40 dB less than the high-gain antenna. This reduced the useful data rate from Galileo from 100,000 bits per second to only 10 bits per second! It fell to JPL engineers, including coding specialists, to increase this data rate by making post-launch enhancements to the Galileo communications system. The major system-level enhancement was the addition of 15:1 image data compression [5], [7]. As with the Voyager system, the presence of data compression means that the required decoded bit error probability is very small, this time on the order of 10^{-7} , and a RS-convolutional concatenated coding system is indicated. However, the importance of every tenth of a dB to the success of the mission; and the fact that the received data rate is so small, motivated JPL coding engineers to propose a very elaborate, high-performance system, which we will describe briefly. The key idea of these coding enhancements is *redecoding*, which means making several decoding passes through the data. This idea seems to have originated independently in ([8], [9]), and [24].

The Galileo "S-Band" coding system is of the same general form as the original Galileo system, but both the inner codes and outer codes are somewhat modified. The original Galileo $K = 15$, rate $1/4$ code could not be used because it was inextricably linked to the crippled X-band antenna. Furthermore, the only transmission path through the S-band antenna passes through a hard-wired NASA standard $K = 7$, rate $1/2$ code. In order to obtain the coding gains achievable from a long constraint length convolutional code, coding engineers were forced to program Galileo's on-board computers to encode a $K = 11$, rate $1/2$ convolutional code, which, when then cascaded with the hardware $K = 7$ code, formed a $K = 14$, rate $1/4$ convolutional code [5], [28].

The outer code for the Galileo S-band mission is Reed-Solomon, over the field $GF(2^8)$, interleaved to depth 8. However, the 8 codewords in each interleaved block don't all have the same redundancy (see Figure 5)

The redundancies chosen for the eight RS codewords in each interleaved block are, as shown, 100, 10, 32, 10, 60, 10, 32, 10, which works out to an average redundancy of 33 symbols per codeword. The codewords with higher redundancy are called "strong" codewords, and those with lower redundancy are called "weak" codewords. (In fact there is a 256th byte in each RS block, which is a synch marker, so that each S-band Galileo data frame is actually a 256×8 array of bytes, consisting of 8 bytes of synch marker, 1776 information bytes, and 264 bytes of RS parity.)

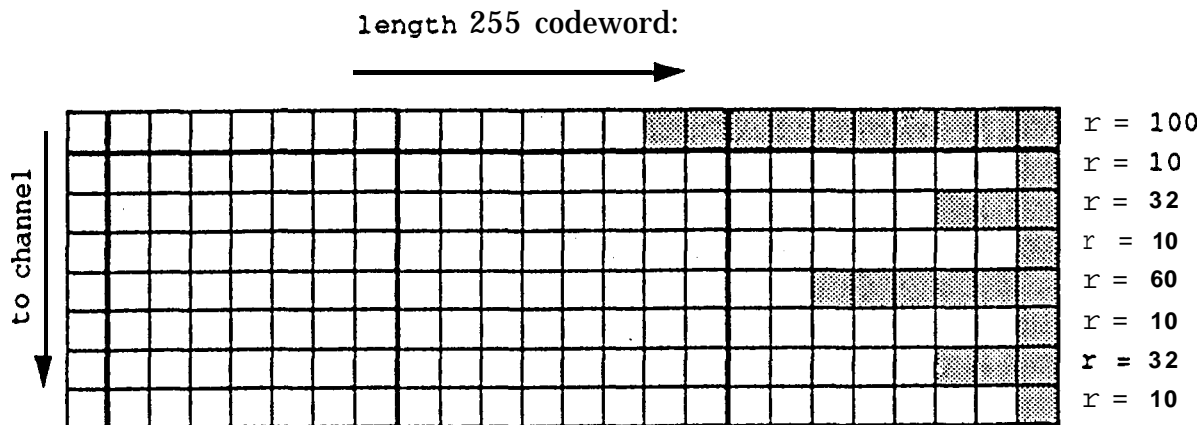


Figure5. The Galileo S-Band interleaved RS block.

The idea is that in the presence of a long burst (or several long bursts) of errors from the inner Viterbi decoder, the strongest RS codeword will be very likely to decode, even though some or all of the weaker codewords may not. Once the RS decoding is complete, the Viterbi decoder then makes a second pass, aided this time by the sure knowledge of those bits decoded by the RS decoder. This knowledge is used to force the Viterbi decoder to consider only paths which are consistent with the known bits, i.e., the paths are “pinned down” at certain locations. (In fact, since a state of the the inner code is specified by 13 bits, and since typically no two consecutive eight-bit RS symbols will be known, the full Viterbi decoder state will typically not be known. Still, the partial state information provided by the RS decoder allows the Viterbi decoder to discard many formerly attractive paths.) Because any long Viterbi decoder bursts have been broken up by the states which are pinned, or partially pinned, by the knowledge provided by the RS decoder, the Viterbi decoder can do a better job of decoding on the second pass.

After the second pass of the Viterbi decoder, RS decoding is repeated. This time, the hope is that the second strongest RS word will decode, so that even more knowledge of the correct bits can be used by the Viterbi on a *third* pass. Finally, after four passes through both decoders, the process stops.

Another Galileo decoding enhancement, involving only the RS decoder, is *error forecasting*. If one word in an interleaved RS block decodes, but others do not, the corrections made by the decoded word can be used predict, or *forecast*, locations of some of the errors in the adjacent undecidable RS words. If sufficiently many of the erroneous symbols in the uncorrectable RS words are successfully forecasted (erased), then some or all of these

words may decode on a second try, since any RS code can correct twice as many erasures as errors. However, in the *Galileo* scheme, the “pinning” of states shortens the Viterbi bursts enough so that *error forecasting* is of relatively little value. The entire variable redundancy, quadruple-pass scheme, adopted for use on the *Galileo* S-band mission, gains 0.53 dB over the “plain vanilla”, constant redundancy, one-pass, RS concatenated scheme, at a decoder error probability of 2×10^{-7} .

Although at the time this article was being written (Summer 1993), no documentation for the *Galileo* decoder, as eventually implemented, yet existed, in an earlier article, [13], a similar, but simpler, system was analyzed in detail. In that system, the redundancies chosen for the 8 RS codewords in each interleaved block were 64, 20, 20, 20, 64, 20, 20, 20, an average of 31 symbols per codeword, and only two decoding passes were used. The performance of this scheme, and three simpler schemes, taken from [13], is summarized in Table 1.

Table 1. Performance of Four RS-Viterbi Concatenated Systems with decoded 8-bit symbol error probability of 2×10^{-7} .

Decoding passes	1	1	2	2
Error forecasting	No	Yes	No	Yes
Redundancy profile	(32)	(44, 28, 28, 28)	(66, 20, 22, 20)	(64, 20, 20, 20)
E_b/N_0 (dB) required	1.17	0.98	0.78	0.76

In Table 1, we see comparisons of four possible schemes of the type described above. In each case, the inner code is the $K = 14$, rate 1/4 “cascaded” convolutional code described above, and the RS codewords are length 255, and interleaved to depth 8. The four options listed in Table 1 correspond, roughly, to the four combinations of yes-no answers to the two questions (1) Is error forecasting used when decoding a RS block, and (2) Is Viterbi redecoding done? The first column corresponds to a “plain vanilla” system with constant RS redundancy 32, no error forecasting, and only one decoding pass. On the bottom line, we see that the value of E_b/N_0 required for this system to achieve a decoded (8-bit) symbol decoded error probability of 2×10^{-7} , is 1.17 dB. In the second column, variable redundancy RS codes (44, 28, 28, 28, 44, 28, 28, 28) plus error forecasting, but only one pass, yields an improvement of 0.19 dB. In the third column, we see the performance of a system with variable redundancy (66, 20, 22, 20, 66, 20, 22, 20) but no error forecasting, but with two passes through the decoders. This yields a further improvement of 0.20 dB. Finally, in column 4, we see the performance of a two-pass, error-forecasting system, with redundancy profile (64, 20, 20, 20, 64, 20, 20, 20). The improvement over column three is only 0.02 dB, but is nevertheless positive, and was judged to be worth the slight increase in complexity over system 3. The overall improvement over the plain vanilla system is thus seen to be

1.17 – 0.76 = 0.41 dB, compared to the 0.53 dB improvement for the full-blown Galileo system, described above.

Encoding, both for the $K = 11$, $r = 1/2$ convolutional code, which cascades with the hardware $K = 7$, $r = 1/2$ code to form the inner $K = 14$, $r = 1/4$ convolutional code, and of the various interleaved RS codes, will be done in software by Galileo's Command Data Subsystem, which consists primarily of eight space-qualified RCA 1802 microprocessors. These same processors, will, in addition, perform the on-board data compression. (All this is in addition to the tasks they were programmed to do before the high-gain antenna failed!)

On the ground, decoding and redecoding for both the RS and convolutional codes will be done in software, by a Sun SC 1000 workstation. The RS decoding program was written by Todd Chauvin, using the time-domain RS errors-and-erasures decoding algorithm described on p. 155 of [19]. The Euclidean algorithm is used to solve the key equation.

In summary, the star-crossed *Galileo* mission has provided a once-in-a-lifetime opportunity for coding engineers to pull out all the stops and design what is arguably the highest performance, highest complexity error-control coding system ever built, and Reed-Solomon codes form a central part of this system.

5. The CCSDS Standard.

By now the use of Reed-Solomon on spacecraft telemetry systems has become relatively routine, and so it is not surprising that a committee has written standards. Indeed, in May 1984, the Consultative Committee for Space Data Systems, representing the space agencies for most of the world (including NASA and ESA, the European Space Agency), issued an official recommendation for a telemetry channel coding standard, [2], [3], [4], which has since been adopted for use by numerous planetary missions, including NASA's Mars Observer (to Mars: was launched in September 1992, arrived in August 1993), *Cassini* (to Saturn: will be launched in 1997, arrive 2004), the joint NASA/ESA *Ulysses* mission (to the Sun's polar regions: was launched October 1990, will arrive June 1994), and ESA missions *Giotto** (1985-1986 mission to Halley's comet), *Huygens* (the Titan probe which will fly aboard *Cassini*), and *Cluster* and *Soho* (both spacecraft in the International Solar and Terrestrial Physics program).

The CCSDS recommended coding standard is twofold: a convolutional coding system without concatenation, and a convolutional coding system with concatenation. The

* In fact, *Giotto* was launched on July 2, 1985, and arrived at Halley's comet in the Spring of 1986, and so it was transmitting RS encoded data from deep space almost as soon as *Voyager* did. (See Figure 4.)

unconcatenated system contains no surprises: the recommendation is for the venerable $K = 7$, rate $1/2$ code that has been used so many times before. The *concatenated* system, however, does contain some surprises. The recommended RS code is a (255, 223) code over $GF(2^8)$, with recommended interleaving depths of 1, 2, 3, 4, and 5. However, (cf. Section 3.2), the field $GF(2^8)$ is to be represented by the polynomial $m(x) = x^8 + x^4 + x^3 + x^2 + 1$, rather than the expected $x^8 + x^4 + x^3 + x^2 + 1$, and the generator polynomial for the CCSDS's (255,223) code is

$$g(z) = \prod_{j=1}^{143} (z - \alpha^{11j}),$$

where α is a primitive root in the field $GF(2^8)$, i.e, a root of the equation $m(x) = 0$. This choice of parameters is the result of work done by Belekamp and Perleman ([1], [25], [26]), in the early stages of the Galileo project. In particular, Berlekamp discovered a way to simplify the encoding of RS codes using bit-serial arithmetic, relative to the so-called *dual basis* for $GF(2^8)$ relative to the "standard" basis $\{1, \alpha, \dots, \alpha^7\}$. The particular choice of field representation and generator polynomial given above, recommended by the CCSDS, was motivated by a desire to minimize the encoder hardware for a bit-serial, dual-basis, encoder.

In 1990, Paaske [24] discussed the possibility of using a "two pass" decoding strategy on the CCSDS standard system. This work had a substantial influence on the final design of the Galileo S-band coding system, even though that system does not conform to the CCSDS standard.

6. Summary and Conclusions.

Coding has been an essential part of space communications systems for thirty years. *Reed-Solomon* coding for this application is a relative newcomer, but for the past decade it too has been an integral part of space exploration. Space applications represent some of the earliest, and most important uses of these powerful codes. We have seen that in deep-space applications, RS codes are always used as outer codes in concatenated systems of the type originally proposed by Forney and Odenwalder, when the required decoded bit or symbol probability is of the order 10^{-5} or less, as is usually dictated by the presence of data compression. Despite the fact that the theory of RS codes is quite mature, we have seen that the needs of space communication have driven research in RS codes into directions that would probably not have occurred otherwise, e.g., bit serial encoders, variable redundancy interleaving, and so on.

As planetary missions become increasingly sophisticated and cost-constrained, and in particular, as data compression becomes standard practice, we may be sure that Reed-Solomon codes, by now old friends to the communication system designers, will find their way into the farthest reaches of the solar system—and beyond!

References.

1. E. R. Berlekamp, "Bit Serial Reed-Solomon Encoders," *IEEE Trans. Inform. Theory*, vol. IT-28 (Nov. 1982), pp. 869-874.
2. Recommendation for Space Data System Standards: Telemetry Channel Coding, issue 1. Washington, D. C.: May, 1984.
3. Consultative Committee for Space Data Systems "Blue Book" no. CCSDS 101. O-B-2. NASA: Washington, D. C.. 1987.
4. Consultative Committee for Space Data Systems "Blue Book" no. CCSDS 101.O-B-3. NASA: Washington, D. C.. 1992.
5. K.-M. Cheung, D. Divsalar, S. Dolinar, I. Onyszchuk, F. Pollara, and L. Swanson, "Changing the Coding System on a Spacecraft in Flight," *Abst. 1993 Int'l. Symp. Inform. Theory*, p. 381.
6. K.-M. Cheung and S. J. Dolinar, Jr., "Performance of Galileo's Concatenated Codes with Nonideal Interleaving," *JPL TDA Progress Reports*, vol. 42-95 (Nov. 1988), pp. 148-152.
7. K.-M. Cheung, and K. Tong, "Proposed Data Compression Schemes for The Galileo S-Band Contingency Mission." *Proc. 1993 Space and Earth Science Data Compression Workshop* (Snowbird, Utah, April 1993) (NASA Conf. Pub. 3191), pp. W--109.
8. O. Collins, *Coding Beyond the Computational Cutoff Rate*. Pasadena California: Caltech Ph.D. thesis, May, 1989.
9. O. Collins and M. Hinzla, "Determinate-state Convolutional Codes." *JPL TDA Progress Reports*, vol. 42-107 (Nov. 1991), pp. 36-56.
10. D. J. Costello and S. Lin, *Error Control Coding*. Englewood Cliffs, N. J. : Prentice-Hall, 1983.
11. J. P. deVries. *Voyager: Functional Requirement Voyager Dual Processor Data System*. NASA Jet Propulsion Laboratory document D-238. Pasadena, Calif.: October 1982.
12. S. Dolinar, "A New Code for Galileo," *JPL TDA Progress Reports*, vol. 42-93 (May. 1988), pp. 83-96.
13. S. Dolinar and M. Belongie, "Enhanced Decoding for the Galileo S-Band Mission." *JPL TDA Progress Report 42-114* (August 1993), in press.

14. B. Dorsch and W. H. Miller. "Error Control Using a Concatenated Code," Washington, D. C.: NASA Technical Note D-5775, June 1970.
15. G. D. Forney, *Concatenated Codes*. Cambridge, Mass.: MIT Press, 1967.
16. E. Hilbert, J. Lee, E. Popard, and R. F. Rice, "Reed-Solomon Coding for Galileo Project." Jet Propulsion Laboratory, document GLL-625-303. Pasadena, Calif.: April, 1979.
17. C. R. Lahmeyer, *Reed-Solomon Decoder Manual*. NASA Jet Propulsion Laboratory document D-1665. Pasadena, Calif.: April 1985.
18. R. J. McEliece, *The Theory of Information and Coding*. Reading, Mass.: McGraw-Hill, 1977.
19. R. J. McEliece, "The Decoding of Reed-Solomon Codes," *JPL TDA Progress Reports*, vol. 42-95 (Nov. 1988), pp. 153-167.
20. B. C. Murray and E. Burgess. *Flight to Mercury*. New York: Columbia University Press, 1977.
21. B. C. Murray. *Journey into Space*. New York: Norton, 1989.
22. J. P. Odenwalder, *Optimal Decoding of Convolutional Codes*, U. C.L.A. Ph.D. dissertation, Systems Science dept., 1970.
23. J. P. Odenwalder, et. al. *Hybrid Coding Systems Study Final Report*. NASA dot. no. NASA-CR-1 14486. San Diego, Cal.: Linkabit, 1972.
24. E. Paaske, "Improved Decoding for a Concatenated Coding System Recommended by CCSDS," *IEEE Trans. Comm.* vol 38 (August 1990), pp. 1138-1144.
25. M. Perlman. *Space Flight Operations Center Reed-Solomon Subsystem (RSS) Mappings Between Codewords of Two Distinct (N, K) Reed-Solomon Codes over $GF(2^J)$* . NASA Jet Propulsion Laboratory document D-8351 Change 1. Pasadena, Calif.: Sept. 1991.
26. M. Perlman and J.-J. Lee. *Reed-Solomon Encoders—Conventional vs. Berlekamp's Architecture*. NASA Jet Propulsion Laboratory Publication no. 82-71, November 1982..
27. W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes* (2nd Ed.). Cambridge, Mass.: MIT Press, 1972.
28. F. Pollara and D. Divsalar, "Cascaded Convolutional Codes," JPL TDA Progress

- Reports, vol. 42-110 (Aug. 19921), pp. 202-207.
29. E. C. Posner, 'Combinatorial Structures in Planetary Reconnaissance," pp. 15-46 in *Error Correcting Codes*, H. B. Mann. Ed. New York: Wiley & Sons, 1968.
 30. J. G. Proakis. *Digital Communications*, 2nd ed. New York: McGraw-Hill, 1989.
 31. R. F. Rice. *Channel Coding and Data Compression System Considerations for Efficient Communication of Planetary Imaging Data*. NASA Jet Propulsion Laboratory Tech. Memo. 33-695. Pasadena, Calif.: Sept. 1974.
 32. R. F. Rice. 'An Advanced Imaging communication System for Planetary Exploration," SPIE Seminar proc. vol. 66 (Aug. 1975), pp. 70-89.
 33. R. F. Rice. *Potential End-to-End Imaging Information Rate Advantages of Various Alternative Communication Systems*. NASA Jet Propulsion Laboratory Publ. 78-52. Pasadena, Calif.: June 1978.
 34. R. F. Rice, E. Hilbert, J.-J. Lee, and A. Schlutsmeyer, "Block Adaptive Rate Controlled Image Data Compression," Proc. 1979 ITC, pp. 53.5.1-53.5.6.
 35. L. Swanson, 'A New Code for Galileo," Abst. 1988 Int'l. Symp. Inform. Th'y., p. 94.
 36. A. J. Viterbi and J. K. Omura. *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1979.