

Technical University of Denmark



Improved decoding for a concatenated coding system

Paaske, Erik

Published in:
I E E E Transactions on Communications

Link to article, DOI:
[10.1109/26.58746](https://doi.org/10.1109/26.58746)

Publication date:
1990

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Paaske, E. (1990). Improved decoding for a concatenated coding system. I E E E Transactions on Communications, 38(8), 1138-1144. DOI: 10.1109/26.58746

DTU Library

Technical Information Center of Denmark

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Improved Decoding for a Concatenated Coding System Recommended by CCSDS

ERIK PAASKE

Abstract—The concatenated coding system recommended by CCSDS uses an outer (255,233) Reed–Solomon code based on 8 b symbols, followed by the block interleaver and an inner rate $\frac{1}{2}$ convolutional code with memory 6. Viterbi decoding is assumed. Two new decoding procedures based on repeated decoding trials and exchange of information between the two decoders and the deinterleaver are proposed. In the first one, where the improvement is 0.3–0.4 dB, only the RS decoder performs repeated trials. In the second one, where the improvement is 0.5–0.6 dB, both decoders perform repeated decoding trials and decoding information is exchanged between them.

I. INTRODUCTION

THE Consultative Committee for Space Data Systems (CCSDS) has recommended a concatenated coding system as a standard for telemetry channel coding [1]. The coding system has an outer (255, 223) Reed–Solomon code based on 8 b symbols, followed by a block interleaver and an inner rate $\frac{1}{2}$ convolutional code with memory $M = 6$. Viterbi decoding is assumed.

In this paper, we consider ways of achieving an extra (de)coding gain by exchanging information between the two decoders and by allowing repeated decoding trials which use information from the other decoder. We consider only the system recommended by CCSDS, except that we extend the maximum interleaving degree to $I = 16$.

It is intuitively clear that the full error correction capability of a concatenated code is not attained if no information is exchanged between the inner and outer decoders or the deinterleaver and outer decoder. Several approaches to utilize the full power of the code have been taken, by Lee [2] among others, but promising results, i.e., more than 0.3 dB, was achieved only with a complex real-time minimum byte error algorithm used on unit memory codes. Furthermore, we do not know of any results specifically obtained for the system recommended by CCSDS.

In our study, two approaches have been investigated. The first one uses the Viterbi decoder only once, and can be performed off line at a later time if only the nondecoded Reed–Solomon words are kept together with the error positions of the already-decoded words. With a frame rejection probability, $P_{FR} < 10^{-2}$, a 0.3–0.4 dB improvement was achieved. The second method requires the entire input stream to be kept since it uses partly repeated Viterbi decoding with forced (known) states and list-of-2 Viterbi decoding. A 0.5–0.6 dB improvement was achieved with $P_{FR} < 10^{-2}$.

The paper is organized as follows. In Section II, we give a short description of the system as it was implemented by the European Space Agency (ESA) for the Giotto Mission to Halley's Comet, and which is the implementation we focus on. In Section III, we treat the theory and strategy behind our first approach, with more details included in Appendix A, and in a similar way, we treat our second approach in Section IV, with more details included in Appendix B. In Section V, we report on simulation results for both approaches, and

Paper approved by the Editor for Coding Theory and Applications of the IEEE Communications Society. Manuscript received January 6, 1989; revised March 24, 1989. This work was supported in part by the Danish Council of Scientific and Technical Research under Grant 16-3656.E-047.

The author is with the Institute of Circuit Theory and Telecommunication, Technical University of Denmark, DK-2800 Lyngby, Denmark.
IEEE Log Number 9037473.

Section VI is devoted to an unsolved problem for which a solution might bring concatenated coding a large step forward.

II. DESCRIPTION OF CONCATENATED CODING SYSTEM

Although the inner and outer codes have been specified in the CCSDS recommendation [1] and a 32 b synchronization word is also foreseen, there are different philosophies to follow when it comes to the actual implementation of the decoder. As already mentioned, we have used the system which was actually implemented by ESA to support the Giotto Mission to Halley's Comet and which is shown in Fig. 1.

The system transmits in frames with a 32 b sync word at the beginning of each frame. The data including the sync word¹ are encoded by the Reed–Solomon (RS) encoder and interleaved according to the scheme shown in Fig. 2. The RS symbols are 8 b bytes, and thus the interleaving scheme can be regarded as a byte-oriented interleaving scheme. In the system implemented by ESA, the maximum interleaving depth is $I = 8$, but in this paper, we shall extend interleaving depths up to $I = 16$, mainly because I turns out to be an important parameter in our decoding process.

The RS-encoded data from the interleaver are convolutionally encoded with a rate $\frac{1}{2}$, 64-state code, and then passed on to the PSK modulator.

The channel is assumed to contribute additive white Gaussian noise (AWGN) to the signal.

At the ground station, the eight-level soft decision data from the demodulator are fed to the format synchronizer (FS), which resolves frame synchronization and phase ambiguity. Unlike NASA, which places the FS after the Viterbi decoder (VD), ESA has chosen to place it in front of the VD. This position has some advantages. One is that the FS can operate on soft decision data and the noise can still be considered to be AWGN, which is not the case after the VD. This implies that the almost optimum detection algorithm developed by Massey [3] and Tolstrup [4] can be used, and thus a very efficient frame synchronization can be achieved. (In fact, the FS implemented by ESA still achieves fast and reliable frame synchronization down to $E_b/N_0 = 0$ dB.) But once frame synchronization is achieved, the node synchronization problem for the VD is also solved. Thus, we shall assume perfect synchronization henceforth.

The received data are then passed on to the VD, the deinterleaver DI, and finally the RS decoder (RSD) which can correct up to $t = 16$ byte errors. By definition, a decoding error occurs when the RSD finds a codeword other than the transmitted codeword; this is in contrast to a decoding failure, which occurs when the RSD fails to find any codeword at all. For the (255, 223) RS code, decoding errors occur with a probability less than $1/t! \sim 10^{-14}$ [5]. This means that we can consider the decoded data from the RSD to be virtually error-free because a decoding failure appears with a probability around $1 \cdot 10^{-14}$ if more than 16 errors occur. In other words, the RSD can be used to correct $t \leq 16$ errors and detect $t > 16$ errors. Simulation shows that for $I \geq 8$ and $E_b/N_0 > 2.5$ dB, decoding failures are extremely rare.

When the decoders operate in what we shall denote normal mode, the data pass through each box only once, and there is no additional information exchange between the three boxes. In the next

¹In a revised issue of the CCSDS recommendation, the sync word is no longer included in the RS codeword.

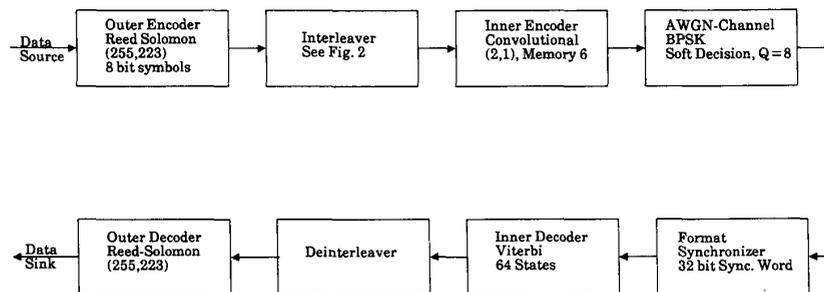


Fig. 1. Concatenated coding system as implemented by ESA.

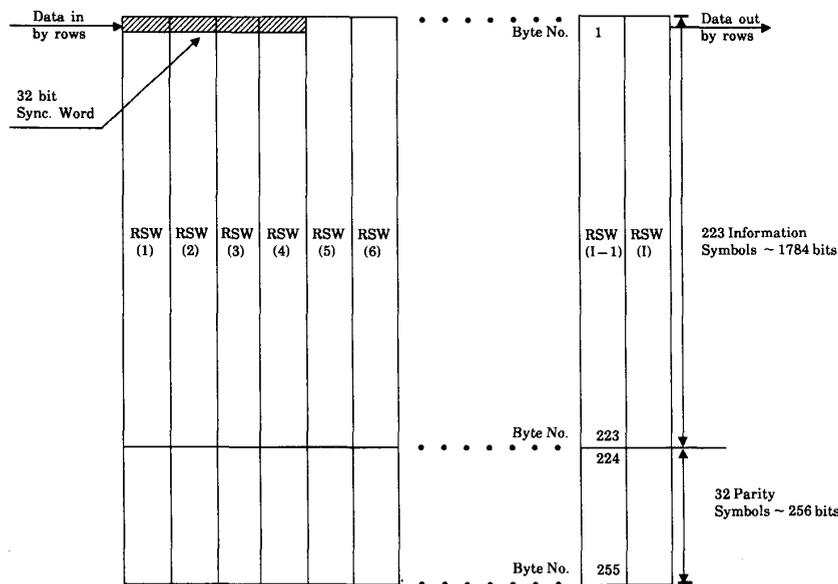


Fig. 2. Interleaving scheme.

two sections, we shall consider two alternatives where more than one decoding trial is performed if necessary, and where we exploit the possibility of additional information exchange among the VD, DI, and RSD.

III. DECODING WITH REPEATED RS-DECODING TRIALS ONLY

Using this alternative, the additional storage capacity can be kept very small, and still the decoding can be improved by 0.3-0.4 dB. We achieve this by letting the VD operate in normal mode, while the RSD is allowed to perform repeated trials on the Viterbi decoded data, operating as an error-and-erasure decoder. Thus, no additional storage requirements are needed for the input data to the VD, and the postdecoding needs only additional storage for the parity bytes of the RSW which are not decoded and the error positions of those which are already decoded.

A. Strategy and Procedure

The strategy is based on the ability of the RSD to decode e errors and s erasures as long as $2e + s \leq 32$. Thus, we can expand the error-correcting capability of the code if we can transform errors to erasures. However, since we do not know the error positions for sure, we might also erase a correct byte by accident. We therefore denote erasures that hit erroneous bytes as good erasures (GE) and those hitting correct bytes as bad erasures (BE). The decoding possibility is, of course, only improved if the number of GE exceeds the number of BE, and therefore we need erasure procedures (EP) where this situation prevails. We have used four such procedures, and three of

them use the fact that errors appearing after the VD tend to occur in bursts. In the following we let $RSW(i)$ denote the i th codeword in the interleaving scheme (see Fig. 2), and the argument i is always taken modulo the interleaving degree I . We notice that an error burst of length $l + 1$ bytes starting in byte position k in $RSW(i)$ will in $RSW(i + j)$ affect byte position k for $i + j \leq I$ and $i + j \leq i + l$, and byte position $k + 1$ for $I < i + j \leq i + l$. For convenience, we therefore define byte position k in $RSW(i + j)$ to be the k th byte if $i + j \leq I$ and the $(k + 1)$ th byte if the argument should be reduced modulo I , i.e., if $i + j > I$.

EP1: Assume that $RSW(i + 1)$ to $RSW(i + l - 1)$ have not been decoded, while $RSW(i)$ and $RSW(i + l)$, $l \leq I$ have been, and that errors have been corrected in byte position k for both. Erase position k in $RSW(i + 1)$ to $RSW(i + l - 1)$. We denote such erasures as double-sided erasures (DSE), and notice that the likelihood for a burst error to have occurred in the erased position is very high. Simulations show that the probability for a DSE to be a GE is around 0.96.

EP2: Assume that $RSW(i)$ have been decoded, while $RSW(i + 1)$ or $RSW(i - 1)$ have not. Erase all error positions from $RSW(i)$ in $RSW(i + 1)$ or $RSW(i - 1)$, and denote such erasures as single-sided erasures (SSE). Simulations show that the probability for an SSE to be a GE is around 0.60.

EP3: Assume that $RSW(i)$ has been decoded (e errors corrected), while $RSW(i + 1)$ or $RSW(i - 1)$ have not, and assume that s_1 DSE's can be obtained by EP1. Combine the s_1 DSE with the selection of s_2 erasures chosen among the $e - s_1$ possible SSE's. The optimum choice of the number s_2 is treated in Appendix A, but once the

number s_2 is chosen, it makes, in principle, no difference whether the erasures are selected at random or by a systematic procedure.

EP4: Assume that at least one of the nondecoded RSW's has 17 errors. Select (at random or systematically) two byte positions and make them erasures. In an RSW with 17 errors, the probability for such erasures to be GE is, of course, 1/15.

The postdecoding procedure can now be formulated as follows.

Postdecoding Procedure 1:

- 1) Set i_d equal to the number of decoded RSW's.
- 2) If $i_d = I$, go to 8).
- 3) If $i_d = 0$, go to 7).
- 4) Perform decoding using EP1. If successful, go to 1).
- 5) Perform decoding using EP2. If successful, go to 1).
- 6) Perform decoding using EP3. If successful, go to 1).
- 7) Perform decoding using EP4. If successful, go to 1).
- 8) Stop.

In the procedure, the sentence "Perform decoding using EP..." should be understood in the following way. Make erasures on the first nondecoded RSW using the EP mentioned and perform decoding trials. In case the RSW is decoded, the step is terminated successfully and we go to 1). If the RSW is not decoded, we try with the next possible nondecoded RSW. The procedure is repeated until we have either a successful termination with a decoded RSW or an unsuccessful termination, which means that all possible nondecoded RSW's have been tried or, in the case of EP3 and EP4, have been tried T_{\max} times, each with different selections of erasures. The limit T_{\max} ensures termination in the cases where decoding is impossible, and reduces the cost for the cases where an overwhelming amount of trials would be needed.

B. Example

In this example, we shall show how to correct a frame containing 12 RSW's with a total of 239 byte errors. The error pattern is shown in Fig. 3. After the normal decoding, only RSW(3) is decoded, so the postdecoding starts by setting $i_d = 1$ and moving to step 4). As appears from Fig. 3, no burst has a length more than 12 bytes (the longest burst appears in position 52/53), but EP1 will (erroneously) assume a longer burst in position 66/67. Since at least two GE's are needed, the step terminates unsuccessfully. Now, using EP2 on RSW(2), we transform the error pattern of this word to 8 errors and 16 erasures, which can just be corrected. Returning to step 4) does not contribute new possibilities, but step 5) with EP2 on RSW(4) terminates successfully, and we can increase i_d to 3. However, in the next attempt, both steps 4) and 5) terminate unsuccessfully and we go to step 6). We use EP3 on RSW(1) and RSW(5). There are no DSE's, and the optimum choice for s_2 is 16 for RSW(1) as well as for RSW(5). The step terminates successfully after only five trials on each of the two possible RSW's since the error pattern of RSW(5) is transformed to 8 errors and 16 erasures. In the following attempts, RSW(6) and RSW(7) will be decoded by step 5) such that we now have $i_d = 6$.

We are now in a position where steps 4) and 5) cannot bring us any further. In principle, RSW(1) and RSW(8) can be corrected by step 6), but in both cases, this would require that at least four out of five BE's were eliminated by the selection procedure. This did not happen, and the step was terminated unsuccessfully after $T_{\max} = 500$ trials on each of the two possible RSW's.

In step 7), RSW(9) was decoded after only 89 decoding trials on each of the six nondecoded RSW's.

After decoding RSW(9), we have seven DSE's for RSW(8) [which is still not enough], but using EP2 with SSE from RSW(9) as well as RSW(7), the error pattern is finally transformed to one error and 29 erasures.

The next three codewords are decoded in step 6). For RSW(10) and RSW(11), elimination of one out of three and two out of six BE's is necessary, and this was performed in 6 and 17 trials, respectively. Then RSW(1) was decoded because we had the luck to eliminate four out of five BE's using only 35 trials. One reason for this to happen

might be that we have now four DSE's, and then a smaller number to select among.

Finally, RSW(12) is decoded in step 5).

The entire frame was decoded using 1705 decoding trials, including the 1000 unsuccessful decoding trials on RSW(1) and RSW(8) in step 6) and the 530 trials used in step 7). One should notice, however, that the frame chosen for this example is, indeed, a very nasty one ($E_b/N_0 = 1.9$ dB).

IV. DECODING WITH REPEATED VD-DECODING TRIALS

In case there is a possibility to store the entire soft quantized bitstream leaving the FS, one can perform repeated VD-decoding trials whereby the decoding can be improved by 0.5–0.6 dB. We achieve this by letting the VD and RSD perform repeated decoding trials, with the VD constrained in such a way that the symbols which are actually already decoded by the RSD must be part of the decoded path, and by using a list-of-2 decoding structure in case of decoding failures for all RSW's in the first pass.

A. Strategy and Procedure

As mentioned above, the strategy is based on two principles. The first one uses the fact that if the VD knows beforehand a number of states that the correct path must include, then the entire decoded path and, in particular, the bits in the neighborhood of the known states become more reliable. When decoding is based on a minimum metric, the practical way to realize decoding of such a path between two known states can be performed by initializing the known starting state with a zero metric and all other states with a large metric, and then performing the backsearch from the known ending state only. We shall denote this repeated VD with forced states (VDFS). A similar feedback from the outer decoder was used by Lee [2], but apparently without any use of the ending state. Unfortunately, VDFS can only be used if at least one RSW is decoded.

The second principle is, therefore, used when no RSW is decoded in the normal mode. In this case, we perform a list-of-2 VD (VDL2), but the method used is somewhat different from one suggested by Forney [6], and which turned out to be far too complex to be practical. Our aim has not been to implement a maximum likelihood algorithm, but to find an algorithm which is easy to implement and almost as effective. We have, therefore, chosen to trace back all paths merging with the decoded paths at states where the metric difference between the two paths is zero, and use such paths as alternative paths. Such a procedure may, however, result in several alternatives for a single RS symbol (see, e.g., Fig. 4), but in order not to complicate the RSD procedure too much, we are interested in an algorithm resulting in at most one alternative for an RS symbol. We therefore include the condition that for each RS symbol, only the last found alternative byte is kept. As appears from Fig. 4, this condition may imply that only part of an alternative path should be kept.

Having performed the VDL2, we are in a position where alternative bytes exist for some of the RS symbols. These alternative bytes can be used either to point out some erasure positions in the RSW or to substitute some bytes in the path from the normal mode VD. In the first case, we have again GE and BE, but in the latter case, we can have good substitutions (GS), i.e., correct alternative bytes substituting erroneous bytes, neutral substitutions (NS), i.e., erroneous alternative bytes substituting erroneous bytes and bad substitutions (BS), i.e., erroneous alternative bytes substituting correct bytes.

Simulations show that the probability of a GS is around 0.34 and for a NS around 0.30, implying that the probability of a GE becomes 0.64. In Appendix B, we have treated how to find the optimum number n_e to select among N erasures or the optimum number n_s to select among N substitutions, but it is not obvious whether erasures or substitutions should be used. The right thing to do would probably be, for a hypothesized number of errors, to calculate $P(y \geq \gamma)$ and $P(z \geq \delta)$ (see Appendix B) and choose the method which results in the largest probability of a decoding success. Generally speaking, the erasure method wins when t is only slightly greater than 16,

Byte No.	RSW NO.											
	1	2	3	4	5	6	7	8	9	10	11	12
1											*	
-												
3												*
4	*	*					*	*	*			
-												
18							*	*	*	*	*	*
19	*			*	*							
20									*			
-												
23	*	*	*		*	*	*	*				
-												
26	*	*										
-												
33							*	*	*	*	*	
-												
39												*
40	*											
41					*							
-												
46			*	*								
-												
51		*	*	*	*							
52							*	*	*	*	*	*
53	*	*	*	*	*	*						
54											*	
-												
57			*	*	*	*	*	*	*	*	*	*
58	*											
-												
62			*	*	*	*						
-												
65											*	*
66	*	*	*									
67		*	*	*	*							
-												
71	*	*	*									
-												
74				*	*	*	*	*				
-												
79							*	*	*			
80												*
81	*	*										
82									*	*		
-												
84										*	*	
-												
91							*	*	*	*		
-												
95			*	*	*	*			*	*	*	
-												
97							*					
-												
99												*
100	*											
101									*			
102								*	*	*	*	
-												
107									*			
-												
110				*	*							
111		*	*	*								
-												
113			*	*	*	*	*	*	*	*		

Byte No.	RSW NO.											
	1	2	3	4	5	6	7	8	9	10	11	12
115									*	*	*	*
116	*	*	*									*
-												
125	*	*										
-												
127												*
128	*	*										
129			*	*								
-												
135				*	*							
-												
144				*	*							
-												
147										*	*	*
148	*											
149			*	*	*	*	*					
-												
155		*										
156												*
157	*	*										
-												
159	*	*										
160										*	*	*
161	*											
162				*	*	*						
-												
164				*	*	*	*	*	*	*	*	*
165	*											
-												
177									*	*	*	*
178	*	*	*	*	*	*	*					
-												
180			*	*			*	*				
-												
184	*	*										
-												
192											*	*
-												
198							*	*	*	*	*	*
199			*	*	*							
-												
206				*	*	*						
-												
208										*	*	
-												
214												*
215	*											
-												
222				*	*	*	*					
-												
225										*	*	*
226	*					*	*					
227						*	*	*	*	*	*	*
-												
242		*	*	*	*	*	*	*				
-												
247		*	*	*	*	*	*	*				
-												
250						*	*					
251					*	*						
TOTAL NO. OF BYTE ERRORS	22	18	16	17	22	19	19	23	17	22	22	22

Fig. 3. Pattern of byte errors for frame in example.

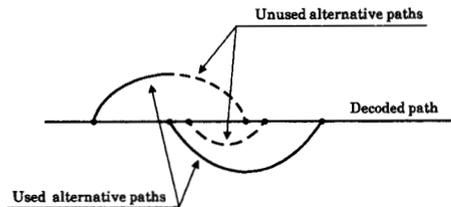


Fig. 4. List-of-2 Viterbi decoding with used and unused alternative paths.

TABLE I
SIMULATION RESULTS FOR CONCATENATED CODING SYSTEM WITH INTERLEAVING
DEGREE $I = 8$ AND POSTDECODING ACCORDING TO PROCEDURES 1 AND 2

E_b/N_0 in dB	Byte Error Rate after VD	Decoding Failures								
		Without postdecoding		With Postdecoding						
		% of Frames	% of RSW	Procedure 1 ; $T_{max} = 500$		Procedure 2 ; $T_{max} = 2000$				
% of Frames	% of RSW			Average No. of RSD-Trials per Word	% of Frames and RSW	Average No. of VDFS per Frame	Average No. of VDL2 per Frame	Average No. of RSD-Trials per Word		
1.8	$8.2 \cdot 10^{-2}$	99.6	84.1	—	—	—	12.8	1.86	0.62	514.3
1.9	$6.9 \cdot 10^{-2}$	95.2	60.1	26.8	16.3	112.4	1.0	1.43	0.17	36.5
2.0	$5.9 \cdot 10^{-2}$	77.8	33.8	5.2	2.2	21.5	0	0.92	0.04	3.87
2.1	$4.9 \cdot 10^{-2}$	44.0	12.7	0.4	0.2	2.30	0	0.45	0	1.13
2.2	$4.1 \cdot 10^{-2}$	18.0	3.5	0	0	1.12	0	0.18	0	1.04
2.3	$3.4 \cdot 10^{-2}$	8.8	0.7	0	0	1.01	0	0.04	0	1.01
2.4	$2.7 \cdot 10^{-2}$	0.6	0.1	0	0	1.00	0	0.01	0	1.00
2.5	$2.3 \cdot 10^{-2}$	0	0	0	0	1.00	0	0	0	1.00

while substitution becomes the better method for $t \geq 20$. There is, however, another argument which favors the substitutions, namely, that the probability of undetected errors in decoded RSW is increased substantially if the erasure method is used, but only slightly if substitutions are used. We therefore decided to use only substitutions, although this might slightly increase the number of decoding trials needed.

The postdecoding procedure can now be formulated as follows.

Postdecoding Procedure 2:

- 1) Set i_d equal to the number of decoded RSW's.
- 2) If $i_d = I$, go to 8).
- 3) If $i_d > 0$, go to 6).
- 4) Perform VDL2.
- 5) Perform RSD with substitutions as described in Appendix B. If successful with one RSW, go to 6); else go to 8).
- 6) Perform repeated VDFS.
- 7) Perform normal RSD's on all the remaining nondecoded RSW's. If successful with at least one RSW, go to 1); else go to 4).
- 8) Stop.

V. SIMULATION RESULTS

The usefulness of the two postdecoding procedures suggested in Sections III and IV was verified by computer simulations. The results are presented in Tables I-III. For each E_b/N_0 value and each interleaving degree, the decoding of 4000 RSW's was simulated, i.e., 500 frames with interleaving degree $I = 8$, 334 frames with $I = 12$, and 250 frames with $I = 16$.

A. Simulations for Procedure 1

For ease of programming some shortcuts were made compared to the procedure outlined in Appendix A for EP3 and EP4. These shortcuts have the effect that the expected number of trials is slightly increased compared to the procedure outlined in Appendix A, and therefore, a small improvement compared to our simulation results might be possible.

The maximum number of decoding trials for each RSW was chosen to be $T_{max} = 500$. The limit was based on some preliminary simulations which showed that most of the RSW's which were not decoded within 500 trials were either impossible to decode or would require an overwhelming amount of trials, implying a large increase in the average number of trials.

TABLE II
SIMULATION RESULTS FOR CONCATENATED CODING SYSTEM WITH INTERLEAVING
DEGREE $I = 12$ AND POSTDECODING ACCORDING TO PROCEDURES 1 AND 2

E_b/N_0 in dB	Byte Error Rate after VD	Decoding Failures									
		Without postdecoding		With Postdecoding							
		% of Frames	% of RSW	Procedure 1 ; $T_{max} = 500$		Procedure 2 ; $T_{max} = 2000$					
% of Frames	% of RSW			Average No. of RSD-Trials per Word	% of Frames and RSW	Average No. of VDFS per Frame	Average No. of VDL2 per Frame	Average No. of RSD-Trials per Word			
1.8	$8.3 \cdot 10^{-2}$	100.0	86.1	—	—	—	—	6.6	2.94	0.44	152.2
1.9	$7.0 \cdot 10^{-2}$	98.2	61.6	34.6	15.2	104.3	0.3	1.93	0.08	11.6	
2.0	$5.8 \cdot 10^{-2}$	86.8	32.0	6.3	1.6	12.7	0	1.08	0.01	1.41	
2.1	$4.9 \cdot 10^{-2}$	57.2	12.9	0	0	1.50	0	0.60	0	1.13	
2.2	$4.1 \cdot 10^{-2}$	21.2	3.1	0	0	1.05	0	0.21	0	1.03	
2.3	$3.4 \cdot 10^{-2}$	4.5	0.5	0	0	1.01	0	0.05	0	1.01	
2.4	$2.8 \cdot 10^{-2}$	0.9	0.1	0	0	1.00	0	0.01	0	1.00	
2.5	$2.3 \cdot 10^{-2}$	0	0	0	0	1.00	0	0	0	1.00	

TABLE III
SIMULATION RESULTS FOR CONCATENATED CODING SYSTEM WITH INTERLEAVING
DEGREE $I = 16$ AND POSTDECODING ACCORDING TO PROCEDURES 1 AND 2

E_b/N_0 in dB	Byte Error Rate after VD	Decoding Failures								
		Without Postdecoding		With Postdecoding						
		% of Frames	% of RSW	Procedure 1 ; $T_{max} = 500$		Procedure 2 ; $T_{max} = 2000$				
% of Frames	% of RSW			Average No. of RSD-Trials per Word	% of Frames and RSW	Average No. of VDFS per Frame	Average No. of VDL2 per Frame	Average No. of RSD-Trials per Word		
1.8	$8.2 \cdot 10^{-2}$	100.0	84.1	—	—	—	2.4	3.78	0.28	71.5
1.9	$6.9 \cdot 10^{-2}$	99.6	59.5	41.4	14.1	99.1	0	2.24	0.04	3.8
2.0	$5.8 \cdot 10^{-2}$	94.0	32.2	8.4	2.0	16.6	0	1.23	0	1.38
2.1	$4.9 \cdot 10^{-2}$	70.8	13.3	0.4	0.1	1.5	0	0.75	0	1.14
2.2	$4.1 \cdot 10^{-2}$	32.4	3.7	0	0	1.08	0	0.32	0	1.04
2.3	$3.4 \cdot 10^{-2}$	6.4	0.6	0	0	1.01	0	0.06	0	1.01
2.4	$2.8 \cdot 10^{-2}$	1.2	0.1	0	0	1.00	0	0.01	0	1.00
2.5	$2.3 \cdot 10^{-2}$	0.4	0.0	0	0	1.00	0	0.0	0	1.00

As appears from Tables I-III, a 0.3-0.4 dB improvement seems possible. This improvement (for $I = 12$ and $I = 16$) can be obtained if the RSD can operate around 50% faster than the normal transmission speed, and some overhead to guide the postdecoding procedure is available. We notice also that the influence of the interleaving degree is mainly to reduce the average number of RSD trials. Finally, we would like to point out that a large improvement can also be observed in the cases where "normal decoding" fails, consider, e.g., $I = 8$, $E_b/N_0 = 2.0$ dB where the number of nondecoded RSW's is decreased from 33.8% to 2.2% and (maybe more important) the number of frames with decoding failures from 77.6% to 5.2%.

B. Simulations for Procedure 2

As appears from Procedure 2, VDFS is used if some but not all RSW's are successfully decoded. The simulations showed that, normally, at least one (and in most cases, several) more RSW's could be decoded following a VDFS. In fact, this happened in all but one case. Therefore, this method seems to function very well as soon as one RSW is decoded.

The VDL2 was simulated as described in Section IV. We used metrics which are equivalent to the integer symbol metrics for eight-level quantization proposed in [7] and we also had the option to trace back paths emerging with the decoded path in states where the metric difference was $\leq \Delta$. Simulations showed, however, that $\Delta = 0$ resulted in a suitable number of alternative bytes, and that the GS/BS proportion was (as expected) much better with $\Delta = 0$ than with $\Delta = 2$. Furthermore, the achievable gain became marginal at the cost of a drastic increase in the number of RSD trials if $\Delta > 0$.

For the RSD in step 5), for ease of programming, we made some shortcuts compared to the procedure outlined in Appendix B. The only effect is a slight increase in the average number of decoding trials for our simulation results. In contrast to Procedure 1, the larger value of $T_{max} = 2000$ was beneficial since more frames were decoded with

a moderate increase in the average number of RSD trials, and the latter number is still far below the value for Procedure 1.

As appears from Tables I-III, an improvement of 0.5-0.6 dB can be achieved, and we get, as expected, the best results with the largest interleaving degree, mainly because the probability that at least one RSW has $t \leq 16$ increases with increasing interleaving degree. Thus, $I = 16$ results in a small increase in the average number of VDFS's, but fewer VDL2's and a substantial reduction in the average number of RSD trials.

Finally, we shall mention that for $E_b/N_0 = 1.8$ dB and $I = 16$, all the remaining frames can be decoded if we use VDL2 with $\Delta = 2$ as well as $\Delta = 0$ and allow an unlimited number of RSD trials, but three frames require more than 10^6 trials. Corresponding improvements can be obtained for $I = 12$ and $I = 8$ at the cost of a substantial increase in the number of RSD trials.

VI. AN UNSOLVED PROBLEM

The decoding described in this paper is based on the possibility of performing repeated RSD trials. Consider the case with substitutions of alternative bytes. If an RSD trial ends with a decoding failure, our decoding situation is not improved, and we do not even know whether the substitutions increased or decreased the number of errors. If, on the other hand, such information could be obtained, a straightforward procedure where only one substitution was added at a time could be used, and the error-correction capability of the code could be increased. Thus, we can phrase the question in the following way. Assume that an RSW has $t > 17$ errors such that a decoding failure results, and assume that one byte is substituted before a second decoding trial. Is it possible by observing the differences in the decoding process for the two trials to extract information as to whether t was increased, decreased, or unchanged?

VII. CONCLUSION

To improve the coding gain for the concatenated coding system recommended by CCSDS, two procedures have been described. Procedure 1 can improve the gain 0.3-0.4 dB. Only one Viterbi decoding and only a slight increase in the storage is required for off-line decoding. Procedure 2 can improve the gain 0.5-0.6 dB. It requires storage of the entire received bitstream (quantized to eight levels) and repeated Viterbi decoding, but has a smaller average number of Reed-Solomon decoding trials. Increasing the interleaving degree generally improves performance.

We believe that the procedures proposed here can be of great value for many deep space missions, in particular, when the link margin is small as, e.g., for the Voyager-Uranus encounter where the link margin was only 0.6 dB.

APPENDIX A

The implementation of EP3 presupposes the selection of s_2 erasures among $e - s_1$ possible SSE's. In this Appendix, we treat the optimum choice of the number s_2 . Considering the high probability of GE among DSE's, we shall assume that all the s_1 DSE's are GE. The problem then becomes to choose $n = s_2$ erasures among $N = e - s_1$ where, on the average, $M = pN$ are GE and $N - M$ are BE. Among the n we choose, there are α GE's and $n - \alpha$ BE's. Then decoding is obtained if

$$y = \alpha - (n - \alpha) = 2\alpha - n \geq \gamma = 2t - 32 - s_1 \quad (A1)$$

where t is the number of errors in the RSW. Thus, we want to maximize $P(y \geq \gamma)$.

Let x denote the number of GE's; then x follows the hypergeometric distribution

$$P(x = \alpha) = \frac{\binom{M}{\alpha} \binom{N-M}{n-\alpha}}{\binom{N}{n}} \quad (A2)$$

For this distribution, we have [8]

$$E[x] = np \quad (A3)$$

$$V[x] = \sigma^2[x] = np(1-p)(1-(n-1)/(N-1)) \quad (A4)$$

$$E[y] = n(2p-1) \quad (A5)$$

$$\sigma[y] = 2\sqrt{np(1-p)(1-(n-1)/(N-1))} \quad (A6)$$

Now, using the approximation [8]

$$P(y \geq \gamma) \approx 1 - \Phi\left(\frac{\gamma - E[y]}{\sigma[y]}\right) \quad (A7)$$

where Φ is the Gaussian distribution function, we can calculate $\delta P(y \leq \gamma)/\delta n$ and obtain a maximum for

$$s_2 = n = \left\lceil \frac{N\gamma}{2(\gamma - M) + N} \right\rceil \quad (A8)$$

where $\lceil x \rceil$ means the integer, which differs from x with the smallest value and which makes $s_1 + s_2$ even and less than N . The reason for an even integer follows from (A1), and we also notice that N erasures have been tried in an earlier decoding step.

For given values of N , M , and γ , we can, of course, find the probability $P(y \geq \gamma)$, and the number of decoding trials $T_{90}(s_2)$ we need to secure that the probability of a decoding success exceeds 90% within $T_{90}(s_2)$ trials.

We are now in a position to describe an approximate procedure to perform decoding using EP3.

Assume $t = 17$ errors in the RSW we work on. Let $M = pN$ with $p = 0.6$, and find s_2 from (A8) for M , $M \pm 1$, and $M \pm 2$. For each value of s_2 , calculate $T_{90}(s_2)$ and perform $T_{90}(s_2)$ trials with a selection of s_2 erasures. If unsuccessful, go to the next possible RSW. Also, if the number of trials for a given RSW ever exceeds T_{\max} , then go to the next possible RSW. If unsuccessful in all the possible RSW's, increase t by 1 and repeat the procedure until an RSW is decoded or T_{\max} is reached for all possible RSW's.

As mentioned earlier, T_{\max} ensures termination, but an alternative would be to let this limit work on the total number of decoding trials within a *frame*.

For the implementation of EP4, we select two erasures among the 255 RS symbols. Thus, the probability of two GE's becomes

$$\binom{17}{2} / \binom{255}{2}$$

and performing, e.g., up to 500 trials leaves us with a decoding failure probability of only 12% if $t = 17$ errors occur.

On the other hand, if we assume $t = 18$ and select four erasures, then about 120 000 trials may be required to end up with the same decoding failure probability.

APPENDIX B

If a VDL2 has been performed and the alternative bytes are used to create erasures, we again have to choose n out of N bytes with $M = pN$ GE's. Thus we have as in Appendix A that decoding is achieved if

$$y = 2\alpha - n \geq \gamma = 2t - 32. \quad (B1)$$

If the N alternative bytes are used to substitute already decoded bytes, we have $M = p_1N$ GS's, $R = p_2N$ NS's, and $N - M - R$ BS's. If we choose n bytes, we get α GS's, β NS's, and $n - \alpha - \beta$ BS's. Then decoding is achieved if

$$z = \alpha - (n - \alpha - \beta) = 2\alpha + \beta - n \geq \delta = t - 16 \quad (B2)$$

and we therefore wish to maximize $P(z \geq \delta)$.

Letting x denote the number of GS's and y the number of NS's, we get

$$P(x = \alpha, y = \beta) = \frac{\binom{M}{\alpha} \binom{R}{\beta} \binom{N-M-R}{n-\alpha-\beta}}{\binom{N}{n}}. \quad (\text{B3})$$

Tedious but straightforward calculations show that

$$E[z] = n(2M + R - N)/N \quad (\text{B4})$$

$$\sigma[z] = \sqrt{n(N-n)A} \quad (\text{B5})$$

where A does not depend on n .

Using again

$$P(z \geq \delta) \approx 1 - \Phi\left(\frac{\delta - E[z]}{\sigma[z]}\right) \quad (\text{B6})$$

we can find the value of n which maximizes $P(z \geq \delta)$:

$$n = \frac{N\delta}{2(\delta - M) - R + N}. \quad (\text{B7})$$

An appropriate method for decoding with substitutions now becomes the following. Assume $t = 17$. Let $M = 0.34N$ and $R = 0.30N$ and find n from (B7) for $M, M \pm 1, M \pm 2$, and $R, R \pm 1, R \pm 2$. Calculate $T_{90}(n)$ and perform $T_{90}(n)$ trials with a selection of n substitutions for each value of n . If unsuccessful, go to the next possible RSW. Also, if the number of trials for a given RSW ever exceeds T_{\max} , then go to the next possible RSW. If unsuccessful in all possible RSW's, increase t by 1 and repeat the procedure until an RSW is decoded or T_{\max} is reached for all RSW's.

ACKNOWLEDGMENT

The author would like to thank L. Prol-Hansen, European Space Operations Center, Darmstadt, West Germany, for having initiated this study, and his student V. H. Højrup for carrying out some of the preliminary simulations in his master's thesis. Also, Prof. J. Juste-

sen, Institute of Telecommunication, Technical University of Denmark, is acknowledged for several fruitful discussions.

REFERENCES

- [1] Consultative Committee for Space Data Systems, "Recommendation for space data systems standard, telemetry channel coding," CCSDS 101.0-B-2, *Blue Book*, issue 2, Jan. 1987.
- [2] L.-N. Lee, "Concatenated coding systems employing a unit-memory convolutional code and a byte-oriented decoding algorithm," *IEEE Trans. Commun.*, vol. COM-25, pp. 1064-1074, Oct. 1977.
- [3] J. L. Massey, "Optimum frame synchronization," *IEEE Trans. Commun.*, vol. COM-20, pp. 115-119, Apr. 1972.
- [4] P. T. Nielsen, "Some optimum and suboptimum frame synchronizers for binary data in Gaussian noise," *IEEE Trans. Commun.*, vol. COM-21, pp. 770-772, June 1973.
- [5] R. J. McEliece and L. Swanson, "On the decoder error probability for Reed-Solomon codes," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 701-703, Sept. 1986.
- [6] G. D. Forney, Jr., "Convolutional codes II. Maximum likelihood decoding," *Inform. Contr.*, vol. 25, pp. 222-266, July 1974.
- [7] J. A. Heller and I. M. Jacobs, "Viterbi decoding for satellite and space communication," *IEEE Trans. Commun. Technol.*, vol. COM-19, pp. 835-848, Oct. 1971.
- [8] N. L. Johnson and S. Kotz, *Discrete Distributions*. Boston, MA: Houghton Mifflin, 1969.



Erik Paaske was born in Odder, Denmark, on April 14, 1938. He received the M.S. and Ph.D. degrees in electrical engineering from the Technical University of Denmark, Lyngby, in 1963 and 1969, respectively.

From 1963 to 1965 he was employed by F. L. Smidth and Company A/S, working with factory automation. Since 1965 he has been at the Institute of Circuit Theory and Telecommunication, Technical University of Denmark. In 1969 he became an Associate Professor, teaching and doing research on information theory and coding.