SEQUENTIAL DECODING OF REED-SOLOMON CODES IN AN INCREMENTAL REDUNDANCY SYSTEM

Laurie L. Joiner and John J. Komo

Electrical and Computer Engineering
Clemson University, Clemson, South Carolina 29634 USA

ABSTRACT

A method of soft-decision decoding of Reed-Solomon codes is presented. This method is an iterative procedure where erasures only decoding is performed followed by a sequential procedure to determine the maximum likelihood estimate of the transmitted word. Results are given that compare this new method of soft decision decoding to more traditional soft decision decoding algorithms. The use of the sequential decoding method in an incremental redundancy system is discussed.

INTRODUCTION

Reed-Solomon codes have been widely used since their introduction in 1959. The algebraic properties of the codes have been used to derive efficient decoding algorithms. Most of the algorithms perform errors only decoding which is based on hard decisions. Here no information on the reliability of the received symbols is used by the decoder even though this information is available at the receiver. One method of using reliability information is to allow erasures and then perform errors and erasures decoding. Errors and erasures decoding has the advantage that more erasures can be corrected than errors. Specifically, an (n,k) Reed-Solomon code over a field of q elements can correct t_e errors and f_e erasures, where $n-k+1 \geq 2t_e+f_e+1$. Thus, erasures only decoding can correct twice as many errors as errors only decoding.

An errors and erasures decoding algorithm still does not use all of the information available to the receiver. One method of using the information available at the receiver is to let the receiver determine a reliability for each possible symbol based on the received signal. The decoder then uses the reliabilities to find the codeword that has the highest probability of being sent. This approach generates the codeword that has the maximum likelihood of being sent given the received word. Such decoding results in low probability of decoder error, but current implementations have a high cost of decoder complexity and time spent decoding a word. An algorithm is sought which performs maximum likelihood or near maximum likelihood decoding, but does so efficiently.

SEQUENTIAL DECODING

A new method of soft decision decoding of Reed-Solomon codes has been developed. The method is an iterative two part procedure. First a received word is decoded using erasures only decoding. If the resulting codeword has the best possible metric then decoding is complete. If the resulting codeword is not the best possible metric, then a sequential procedure is performed to produce a new candidate word and decoding is again attempted. This procedure is continued until a codeword has the best possible metric. The resulting codeword is the maximum likelihood estimate of the transmitted word.

For an (n,k) Reed-Solomon code the n soft decision outputs from the receiver (which are the inputs to the decoder) are ordered with β_i, i=1, ..., n, indicating the location with the ith best metric m_i. The locations with the best metrics are the locations which contain the most reliable symbols. Since Reed-Solomon codes are maximum distance separable (MDS), any k locations in the codeword can be taken as the data locations for decoding. The k locations in the received word with the best metrics, β_1, β_2, ..., β_k, are taken as the data locations for reencoding (equivalently, β_{k+1}, β_{k+2}, ..., β_n, are erasure locations for erasure only decoding). A stack is created using symbol values for a sequence of data locations which correspond to the metrics m_1, m_2, ..., m_k. After this stack is created, a new metric is calculated which is the sum of the metrics of the symbols in the codeword, including the parity symbols. If this metric is better than any metric currently in the stack, the decoding is compete. If not, the stack is extended until the top of the stack corresponds to an entry with all of the data locations. Again, a new metric is calculated which includes the metrics of the corresponding parity symbols. The search for a maximum likelihood estimate of the codeword is complete when a codeword has the best possible metric (a codeword is at the top of the stack).

The results of this decoding method produce a maximum likelihood estimate of the transmitted codeword. Since the stack is begun with the k most likely symbols, the sequential search is considerably reduced. The decoding that takes place with each iteration of the algorithm is erasures only decoding. Erasures only decoding can be performed quickly since only the error magnitudes and not the error locations must be found. A fast algorithm for performing the error magnitude calculations has been developed in [1].

This research was supported by the Army Research Office under grant number DAAH04-95-1-0247.

0-7803-4249-6/97/$10.00 © 1997 IEEE

For large codes and for channels with low signal-to-noise ratios, the decoding time for a received word can still be unfeasibly long because of the number of possible codewords that must be searched. The search can be shortened in two ways. First, the stack can be shortened by eliminating some of the least likely entries in the stack. This shortens the number of iterations that must be performed by eliminating some of the codeword possibilities. The resulting codeword is no longer a maximum likelihood estimate, but only a near maximum likelihood estimate. The search can also be shortened by adjusting the metric to give an advantage to the longer paths that have been explored, similar to the Fano metric [2]. With the original metric, a sequence with few symbols can be expanded if it has a metric that is only slightly better than a metric from a sequence that is of length n. By adding a bias to longer length sequences, they will be chosen over the smaller sequences of approximately the same metric value. This reduces the number of searches, but may result in picking a sequence that does not have the best possible metric. Again this gives near maximum likelihood performance.

EXAMPLE

Suppose an (8,4) extended Reed-Solomon code is transmitted over a Raleigh fading channel using 8-ary modulation. The following table lists the metric for each symbol at each of the n transmitted locations. The metrics are calculated from the difference of the magnitudes of the matched filter outputs for each symbol. A large difference between the largest filter output and the next largest filter output for any symbol location, results in a more negative metric value for the second most likely symbol. For this example, location 4 has the most reliable symbol (symbol 0) since it has the largest (most negative) second metric value (-41.43).

Location 0		Location 1	
Symbol Value	Metric Value	Symbol Value	Metric Value
0	0.00	0	0.00
α^5	-8.70	α^2	-31.38
α^4	-9.03	α^6	-31.49
α^3	-9.14	α	-31.68
α^2	-9.24	1	-33.05
α	-9.88	α^5	-34.01
1	-9.98	α^3	-34.01
α^6	-10.07	α^4	-34.19

Location 2		Location 3	
Symbol Value	Metric Value	Symbol Value	Metric Value
0	0.00	0	0.00
α	-0.51	α	-1.77
α^4	-1.27	α^6	-2.10
1	-1.44	1	-2.23
α^3	-1.46	α^4	-3.22
α^6	-1.64	α^2	-3.46
α^2	-1.82	α^3	-3.51
α^5	-2.28	α^5	-3.54

Location 4		Location 5	
Symbol Value	Metric Value	Symbol Value	Metric Value
0	0.00	α^3	0.00
α^5	-41.43	α	-1.13
α^3	-42.27	α^5	-1.64
1	-42.34	α^4	-1.91
α	-42.48	0	-2.08
α^6	-42.72	α^6	-2.11
α^2	-42.82	1	-2.82
α^4	-42.83	α^2	-3.48

Location 6		Location 7	
Symbol Value	Metric Value	Symbol Value	Metric Value
α	0.00	0	0.00
α^4	-1.23	α^6	-4.52
1	-2.60	α^3	-5.37
α^5	-2.63	1	-6.11
α^3	-3.44	α^2	-6.11
α^6	-3.44	α^5	-6.53
α^2	-3.61	α^4	-6.62
0	-3.64	α	-7.05

The data locations are chosen to be the four positions which have the most reliable symbols. Thus from the above table, the data locations are $\beta_1=4$, $\beta_2=1$, $\beta_3=0$, and $\beta_4=7$. The positions that are left, $\beta_5=3$, $\beta_6=6$, $\beta_7=5$, and $\beta_8=2$ are the least reliable positions and thus are erased in the decoder.

The stack is created by inserting symbol values for the most reliable symbol location (here, location 4). The top entry is extended by filling in values for the second most reliable location, location1. The process continues until the top entry in the stack has symbol values for all data locations. At the point where the top entry in the stack is of length four, the stack (the first 10 entries) is as follows:

Metric Value	β_1=4	β_2=1	β_3=0	β_4=7	β_5=3	β_6=6	β_7=5	β_8=2
0.00	0	0	0	0				
-4.52	0	0	0	α^6				
-5.37	0	0	0	α^3				
-6.11	0	0	0	1				
-6.11	0	0	0	α^2				
-6.53	0	0	0	α^5				
-6.63	0	0	0	α^4				
-7.05	0	0	0	α				
-8.70	0	0	α^5					

The top entry in the stack has all of its data locations filled in, so it is decoded and its new metric is calculated. No stack entries with metrics lower than the codeword will be considered, so the stack is shortened to

Metric Value	β_1=4	β_2=1	β_3=0	β_4=7	β_5=3	β_6=6	β_7=5	β_8=2
-4.52	0	0	0	α^6				
-5.37	0	0	0	α^3				
-5.72	0	0	0	0	0	0	0	0

The top two entries are decoded and their new metrics are calculated to be -13.08 and -13.81, respectively. After these decodings the stack becomes

Metric Value	β_1=4	β_2=1	β_3=0	β_4=7	β_5=3	β_6=6	β_7=5	β_8=2
-5.72	0	0	0	0	0	0	0	0
-13.08	0	0	0	α^6	α^2	α	1	α^5
-13.81	0	0	0	α^3	α^6	α^5	α^4	α^2

Since the top entry in the stack is a codeword, the decoding is complete and the decoded codeword is the all zero codeword.

If a bias had been added to the metrics, fewer decodings would have been performed. For example, if a bias equal to the code rate of 0.5 is added to each metric then the stack is the same except the metrics before any decodings become

Metric Value	β_1=4	β_2=1	β_3=0	β_4=7	β_5=3	β_6=6	β_7=5	β_8=2
2.00	0	0	0	0				
-2.52	0	0	0	α^6				
-3.37	0	0	0	α^3				
-4.11	0	0	0	1				
-4.11	0	0	0	α^2				
-4.53	0	0	0	α^5				
-4.63	0	0	0	α^4				
-5.05	0	0	0	α				
-7.20	0	0	α^5					
-8.53	0	0	α^4					

The metric for the all zero codeword is calculated as -1.72. This is the best metric in the stack, so the decoding process is complete. By adding the bias to the metrics, the number of decodings has been reduced from three to one.

RESULTS

Figure 1 shows the results of decoding a (32,12) extended Reed-Solomon code transmitted over a Raleigh fading channel. The system uses 32-ary MFSK modulation and noncoherent demodulation. Three of the curves are results from the sequential decoding method, with the stack size limited to 10 and 100 iterations. The metric used for all of the sequential decodings is based on the difference of the matched filter outputs. Of the two curves with the stack limited to 100 iterations, one uses the adjusted metric which has the added bias. Here a bias of 0.75 is used. For all of the sequential decodings, the main source of not decoding correctly is caused by decoder failure. If the number of iterations has reached the set limit (here 10 or 100 iterations), a decoder failure is declared. If decoder failures are not allowed, then the system can be forced to make a decision by choosing the top codeword entry in the stack. The other two curves are results from errors and erasures decoding using the reliability information to determine erasure locations. The ratio threshold test [3] (RTT) uses ratios of the matched filter outputs to determine which symbols should be erased. The sequential decoding algorithm for both 10 and 100 iterations has a lower probability of not decoding correctly than the ratio threshold test. The Bayesian erasure insertion test [4] uses filter outputs and knowledge of the channel to determine erasures. The sequential decoding algorithm that allows 100 iterations has a lower probability of not decoding correctly than the Bayesian technique.

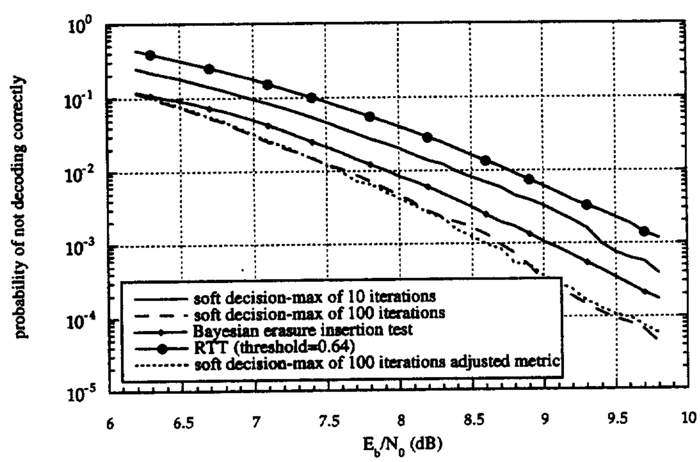

Figure 1
Soft Decision Decoding Versus Errors and Erasures Decoding

Figure 2 shows the reduction in the average number of decodings caused by using the metric that has been biased for the longer data sequences. Figure 1 shows that the probability of not decoding correctly is approximately the same for the two cases where the stack size is limited to 100 iterations, but the average number of decodings has been decreased considerably.

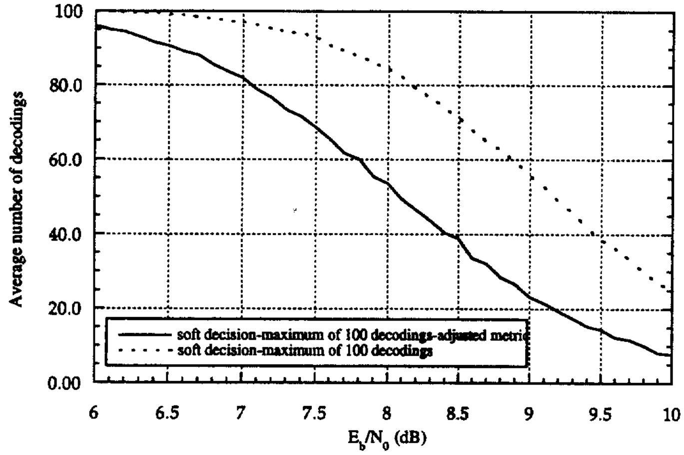

Figure 2
Average Number of Decodings Using Sequential Decoding Method

SEQUENTIAL DECODING IN AN INCREMENTAL REDUNDANCY SYSTEM

One application of the sequential decoding method is its use with a type-II hybrid ARQ protocol. In a type-II system, an (n,k) MDS code (such as an extended Reed-Solomon code) is chosen with rate less than one-half. This code should provide reliable communication on a channel with poor conditions. For this protocol, the k data symbols are encoded to a codeword of length n. Initially, only n/2 symbols are transmitted across the channel. The received symbols are decoded by filling in the missing symbols with erasures. If decoding is successful then a positive acknowledgment is sent. If decoding fails, then a negative acknowledgment is sent, and the other n/2 symbols (or some subset of these symbols) are transmitted. The process continues until decoding is successful with symbols being retransmitted if necessary. The advantage of this protocol is that if channel conditions are poor, then a powerful code can be used to correct transmission errors, but if channel conditions are good, then channel bandwidth is not wasted transmitting unneeded symbols.

Traditional decoding used with this protocol has two drawbacks. First, the reliability information known at the receiver is not used by the decoder which results in poorer performance than a soft decision decoding method. Second with every subsequent transmission of codeword symbols, the decoder must start the decoding process over, discarding calculations made in the previous decoding attempts. Since the sequential decoding method performs near maximum likelihood decoding, performance will be better using sequential decoding than traditional hard decision decoding. Also sequential decoding can use calculations from one transmission attempt on the next attempt. If, on a given transmission, the reliability of a decoded word is not high, a negative acknowledgment is sent and more symbols are transmitted. These symbols can be used to provide additional reliability information for the decoding attempts already performed. Thus additional transmissions do not lose all of the calculations already made.

CONCLUSIONS

A new method of soft decision decoding of Reed-Solomon codes has been presented. This method provides better performance than traditional errors and erasures decoding methods, and reduced complexity over existing maximum likelihood decoders. By adjusting the symbol metrics to add a bias for the longer length sequences, the average number of iterations has been reduced considerably without affecting performance.

REFERENCES

[1] J. J. Komo and L. L. Joiner, "Fast Error Magnitude Evaluations for Reed-Solomon Codes," in *Proc. 1995 IEEE International Symposium on Information Theory*, p. 416, Sept. 1995.

[2] S. Lin and D. J. Costello Jr. *Error Control Coding: Fundamentals and Applications*, Englewood Cliffs, NJ: Prentice Hall, 1983.

[3] A. J. Viterbi, "A robust ratio-threshold technique to mitigate tone and partial band jamming in coded MFSK systems," in *IEEE Military Commun. Conf. Rec.*, pp. 22.4.1-22.4.5, Oct. 1982.

[4] C. W. Baum and M. B. Pursley, "Bayesian methods for erasure insertion in frequency-hop communications with partial-band interference," *IEEE Trans. Commun.*, vol. COM-40, pp. 1231-1238, July 1992.