FEEDBACK DECODERS FOR NAVAL APPLICATIONS

J. A. HELLER
LINKABIT Corporation
San Diego, Calif.

R. D. PETERSON
Naval Electronics Laboratory Center
San Diego, California

Summary Feedback decoding is a simple technique for obtaining a significant improvement in performance on a wide range of binary channels. Code interleaving for burst error correction is particularly simple to implement with feedback decoding. A flexible feedback decoder built by LINKABIT Corporation was tested on a simulated shipboard satellite communication channel. The channel was characterized as being additive Gaussian with a superimposed 5% duty factor periodic burst during which the channel error rate is close to .5. The precise radar induced error burst locations are assumed unknown to the decoder. The simple feedback decoder, operated with the error bursts,provided performance equivalent to t at of an uncoded system operating against additive Gaussian noise alone.

I. Feedback Decoding of Convolutional Codes Viterbi decoding and sequential decoding are two powerful methods for attaining reliable and efficient communication on a variety of memoryless channels. Performance improvements are particularly striking when multi-level quantized received data is available to the decoder. More modest performance improvements can be obtained using an extremely simple class of decoders for convolutional codes called feedback decoders. Simple methods exist for implementing feedback decoders for binary input, binary output channels such as the binary symmetric channel. one well known type of feedback decoder is a "threshold decoder."[1]

In general, feedback decoder implementations have the added attraction that they can be made effective on burst-error channels, such as the periodic burst channel to be considered in this paper. Interleaving of data in the encoder and deinterleaving in the decoder can be performed in a straightforward manner, effectively breaking up error bursts and making the channel look memoryless to the decoder.

As an example, Figure 1 shows a portion of a rate 1/3 tree code starting with some fixed initial node a. The operation of a feedback decoder can be shown by observing how such a decoder might proceed in decoding this tree. Under the code tree in Figure 1 is a sample of hard quantized data that could be received corresponding to three branches worth of transmitted code symbols. Since the code in this example is of rate 1/3, three code symbols are received per branch. observing the received symbols we can quickly see that error(s) must have occurred since the received sequence does not correspond exactly to the code symbols on any path through the tree.

A feedback decoder might proceed in decoding the tree as follows. The decoder observes all paths to depth 2 into the code tree from the initial node, i.e. the paths contained in box A in the figure. The symbols on these path segments are compared with the first 6 received data symbols and the most likely path segment is chosen. When the binary symmetric channel is used,this amounts to choosing the path segment with the minimum Hamming distance to the received sequence. In Figure 1, path segment a-b-c is at distance 2 from the received sequence, and all other paths are at distance 3 or greater so this path is selected.

At this point the first information bit on the selected path is output by the decoder. The information bit sequence on path a-b-c is 10 so a 1 is the decoder output for the first branch level. The decoder now steps forward one branch to the node specified by the first decoder output (node b in Figure 1). It now treats node b as the new initial node and looks at all paths to depth 2 extending from it (box B). The most likely sub-path in box B is b-d-e, so the second decoder output is a “0.” The decoder now steps to node d. Decoding continues in this manner indefinitely.

This decoding procedure is called feedback decoding because decoding decisions at any given time affect decisions in the future. For instance, if the.first two received symbols would have been 0’s instead of 1’s in Figure 1, the first decoder output would have been “0” and the decoder would have proceeded to node f. The set of paths emanating from node f have entirely different symbols than those coming from node b. Thus future decisions can be different depending on whether a “0” or a “1” is the first decoder output. Decoder decisions are “fedback” to determine the next state of the decoder.

As another example to illustrate the implementational principals of the LINKABIT feedback decoders, Figure 2 shows a K = 3, rate 1/2, systematic convolutional coder followed by a channel. The channel outputs are then fed into a “syndrome calculator” which outputs one syndrome bit for each received information-parity pair. The syndrome bits and the received information bits are.fed into the feedback decoder for processing.

In this figure i_j is the j^{th} information bit generated by the source. P_j is he parity bit generated when i_j is first input to the coder. The channel is modeled by the Mod-2 addition of binary noise symbols e^i and e^P to the information and parity streams. Thus,

$$i_j' = i_j \oplus e_j^i$$

$$p_j' = p_j \oplus e_j^p$$

For instance, $e_j^i = 0$ implies correct reception of the j^{th} information bit while $e_j^i = 1$ causes i_j to be complemented, resulting in an error. A noisy channel is represented by error sequences, $\underline{e}^i$ and e^p, with a high density of 1's.

The syndrome calculator takes the received information bit sequence and passes it through a replica of the coder to generate p_j''. p_j'' is then added mod-2 to p_j' and the resulting syndrome bit s_j is sent to the decoder. It is instructive to observe the syndrome generating process in the absence of noise. If $e_j^i = 0$ and $e_j^p = 0$ for all j, the received information and parity bits are identical with those transmitted. This implies that $p_j'' = p_j$ since the coder replica at the receiver has the same inputs as the transmitting encoder.

Now since $p_j' = P_j$,

$$s_j = p_j'' \oplus p_j' = p_j \oplus p_j = 0$$

Thus, in the absence of noise, the syndrome bits are all zero regardless of the transmitted information bit sequence. Now since the syndrome is a linear function of the information bits and the noise, the effect of the noise on the syndrome is independent of the effect of the information bits. The information bits have been shown to have no effect on the syndrome, thus the syndrome is a function only of the noise.

The function of the decision logic in the decoder is to observe L consecutive syndrome bits s_j, ..., s_{j-L+1} (L = 4 in Figure 2) at a time, and determine a likely sequence of information and parity errors that could have caused the observed syndrome. If that sequence, contains an error in the oldest received information bit, i'_{j-L+1}, that bit is complemented and output by the decoder -- otherwise this bit is output uncomplemented. In addition, if it is determined that i'_{j-L+1} was likely in error, the syndrome is modified to remove the effects of the error. It is clear from the syndrome generator that in this example an error in s_{i-3} will cause s_{i-1}, s_{i-2}, and s_{i-3} to be complemented . Thus, the one bit correction output is "fedback" to the syndrome register; hence, present decisions affect future decisions.

The syndrome feedback decoder described here is completely analogous to the feedback decoder discussed in connection with Figure 1. Observation of an L bit syndrome and making a one information error decision is equivalent to comparing a received sequence with all paths to depth L into a code tree and deciding on the oldest information bit on the most likely path. Feeding back the decoder decision to the syndrome register to remove the effects of the error is equivalent to stepping forward on the branch corresponding to the most likely path and re-initiating the decoding procedure.

Threshold decoding is a special case of feedback decoding. In threshold decoding the decision logic of the decoder is constrained to be the equivalent of a single majority logic gate whose inputs are linear combinations of the observed syndrome bits. This constraint on the decision logic severely limits the class of codes which are efficiently threshold decodable. A much more efficient feedback decoder is one in which the decision logic is maximum likelihood. That is, the decision logic is unconstrained, and simply outputs corrections bits corresponding to the most likely L bit path through the code tree. Since the maximum likelihood decision function is always a fairly random appearing function of L input variables, a 2^L bit read-only-memory (ROM) is used for the decision function in the LINKABIT feedback decoders.

Obviously, decoder logic complexity grows exponentially in L. It is therefore highly desirable to use a code requiring a minimum value of L for a given error correction capability. Such codes, with up to 4 error correcting capability for rate 1/2, are tabulated by Bussgang (Ref. 2). Feedback decoders capable of correcting 2 and 3 errors, with L = 6 and 11 respectively, have been build by LINKABIT. The systematic code constraint lengths in these two cases are K = 6 and K = 10.

Interleaving for burst error correction can be easily incorporated into a feedback decoder by simply substituting D-stage shift registers for the individual flip-flops of Figure 2. This will cause code symbols affected by a given information bit to be spaced D symbols apart. If the channel bursts are less than D symbols in length, at most one symbol affected by any given information bit will be in error due to a single burst. Deinterleaving is provided by the D-stage registers in the decoder.

A combination switch selectable 2 and 3 error correcting feedback decoder, the L512N, was delivered to the Naval Electronics Laboratory Center by LINKABIT. The decoder also contained selectable degrees of interleaving to combat bursts.

II. Description of Problem A major problem encountered in placing a satellite receiving terminal aboard a Navy ship is the interference caused by nearby radars. Harmonics of a radar can fall in the pass band of a receiver and prevent communications during the period of the radar transmission. This channel is then characterized by radio frequency interference (RFI) in high level bursts of regular rate and duration. In between the bursts of RFI the channel is Gaussian with fairly good signal-to-noise ratios. Considering the detector as part of the channel, the channel can be described as binary symmetric with a crossover probability that is a two-valued function of time. During the period that the radar is transmitting the crossover probability is on the order of .5. A bit is as likely to be correct as incorrect. When the radar is not transmitting the crossover probability is determined by the E_b/N_o of a Gaussian channel.

A particular radar that causes interference is a pulse compression type radar that has a duty cycle of 5%. Assuming that communication is blocked 5% of the time the best error rate possible is .025.

The Navy's requirements stem directly from this radar interference problem. An error correcting system is necessary that will eliminate this burst of errors. Further requirements are that it be reliable, small in size and power consumption and inexpensive. Improved performance of the decoder in correcting randomly generated errors is desirable but not at an appreciable increase in price or complexity.

III Method of Measurement The decoder received from LINKABIT was tested at NELC as shown in Figure 3.

A test signal is generated in the bit source and is encoded and transmitted over a simulated channel. The channel is simulated by injecting errors in the bit stream by modulo 2 adding the output of an error generator to the bit stream. The output of the error generator is a binary waveform whose probability of being high during any bit interval is p. Therefore, the input error probability to the decoder is fixed by the error generator. The output error rate is measured by comparing the bit source with the decoder output.

Random independent errors are generated by use of a very long pseudo-random sequence generator. A large "and" gate looks for relatively rare occurrences such as n ones in a row. The probability of such an event is $1 \div 2^n$. By varying the number of inputs to the gate, the probability of an event can be selected. Independence of an event from bit to bit is obtained by clocking the shift register at a much higher rate than the data.

IV Performance Decoder performance over a binary symmetric channel is described by a graph relating probability of error input and probability of error output. Figure 4 shows the experimentally measured performance of the LINKABIT decoder in constraint length six and ten modes over a simulated binary symmetric channel. The error performance of the K = 10 code can be described as $2200\, p^4$ where p is the channel cross-over probability. The error performance of the K = 6 code is approximately $160\, p^3$.

The performance of the two coding systems over a Gaussian channel is shown in Figure 5. The performances of the systems are compared on the basis of equal signal energy per data bit. Ideal PSK detection is assumed. Coding gain at a 10^{-5} probability of error is 2.2 dB for the constraint length 10 LINKABIT system and 1.5 dB for the constraint length six LINKABIT system.

Of direct interest is the performance of the coding scheme over a channel that is periodically disturbed by RFI with a 5% duty cycle. The performance of the coded systems while operating in interleaved mode (256 bits) in the burst interference channel described is shown in Figure 6. The results were experimentally determined by simulating the burst interference channel. Note that the performance of the uncoded system is quite limited. No improvement in error performance would occur in the uncoded system until the signal-to-noise ratio was raised so high that it was positive during the interference bursts.

The performance of the coded system is the same as expected for an uncoded system with no bursts of RFI. Assume the link was designed for a received signal-to-noise ratio that yields adequate performance over a straight Gaussian link. The interleaved K = 10 coded system will yield essentially equal error performance over a Gaussian link which additionally is obliterated 5% of the time. The interleaved K = 6 coded system yields results that are approximately .2 dB poorer.

Reference

1. Massey, J. L., Threshold Decoding, MIT Press, Cambridge, Mass., 1963.

2. Bussgang, J. J., "Some Properties of Binary Convolutional Code Generators," IEEE Trans. on Information Theory, January, 1965.

3. Peterson, R. D., "The LINKABIT Corporation L512N Error Correcting Decoder," Naval Electronics Laboratory Technical Note TN 1838, 15 April 1971.

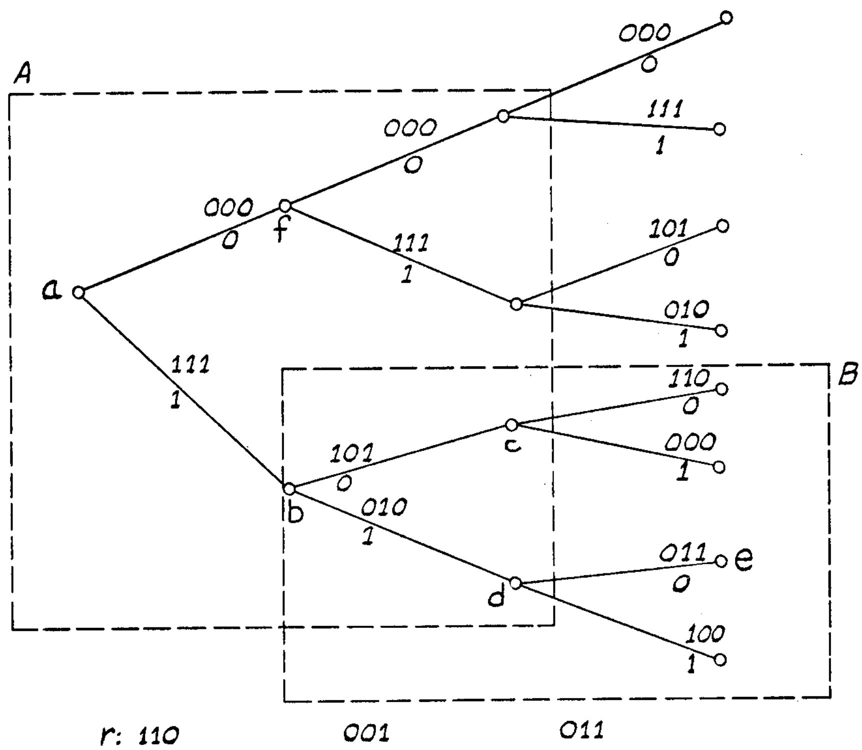

Figure 1. Tree Code and Sample Received Data Showing Operation of a Feedback Decoder.

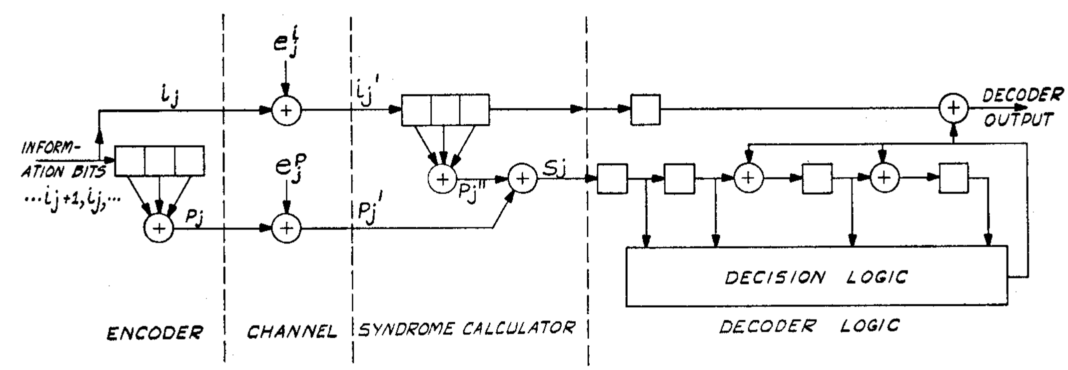

Figure 2. Communication System Using Syndrome Feedback Decoder.

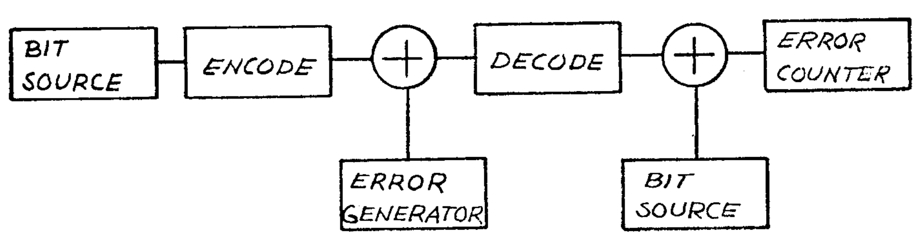

Figure 3. Decoder Test Configuration.

Figure 4.

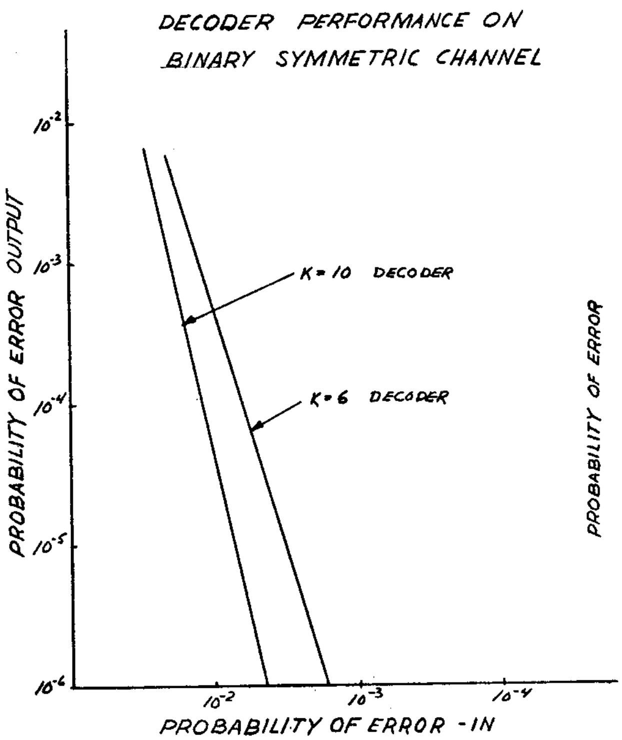

Figure 5.

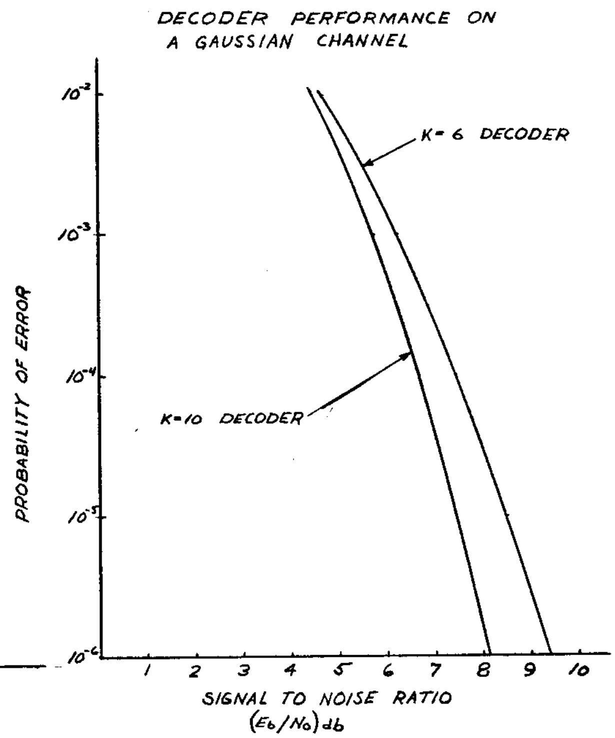

Figure 6.