REDUCED COMPLEXITY MAXIMUM LIKELIHOOD DECODING OF LINEAR BLOCK CODES

S.M. Elengical, F. Takawira and H. Xu

School of Electrical, Electronic and Computer Engineering, University of KwaZulu-Natal, Howard College Campus, King George V Avenue, Durban 4041, South Africa

Abstract: This paper proposes a reduced complexity Maximum-Likelihood (ML) decoding Algorithm for Linear Block Codes based on the Kaneko decoder and incorporating ruling out conditions for useless iteration steps. The proposed decoding scheme is evaluated over the Additive White Gaussian Noise (AGWN) channel using Binary Phase Shift Key (BPSK) signalling by simulation. Simulations results show that for negligible performance loss, there is significant reduction in the complexity of decoding.

Key words: MLD Decoding, Reduced Complexity Decoding, Linear Block Codes.

I. INTRODUCTION

LINEAR block codes have proven to be efficient codes that provide good performance at large block lengths. The performance of large linear block codes can approach the Shannon Limit of a given transmission channel as explained in [2]. Decoding of such linear block codes has been done by soft-decision decoding. This method of decoding provides improved decoding as compared to hard decision decoding as it utilizes probabilistic information of the received sequence at the decoder. Maximum Likelihood (ML) Decoding is an optimized decoding procedure that decodes a received sequence to an output codeword such that the probability of a received codeword, given a transmitted sequence is as high as possible. There has been a number of decoding procedures proposed that offer ML decoding to linear block codes [6]. It has been shown that the performance of a given coding system improves with increase in code length. However the complexity required for performing ML decoding escalates exponentially with increasing code length.

The decoder proposed by Kaneko, et al in [1] performs ML decoding on linear block codes, with ML decoding defined such that the most likely candidate transmitted codeword is included in the list of codewords analyzed for decoding. Therefore the decoder always converges to the most likely codeword.

Given that all codewords have equal probability of being transmitted, the complexity required for decoding grows exponentially with code length n. In the Kaneko decoder the approach to reducing complexity is to find an efficient technique to generate codewords that will contain with high probability the most likely transmitted codeword. The Kaneko decoder generates a set of codewords such that the probability that, the most likely candidate codeword sent is contained in the set of candidate codewords is 1. The algorithm generates a larger set of codewords when a noisy sequence is received and smaller sets when cleaner sequences are received. Thus the average decoding complexity is reduced without loss of performance as compared to fixed codeword set decoders. This paper proposes to reduce the complexity further, by ruling out useless iterative steps in the decoding procedure and thus reducing the total number of iterations during decoding.

II. DECODING PROCEDURE

A. The Kaneko Decoder

The Kaneko decoder uses a calculated reliability sequence for the decoding of received data. At the receiver the demodulator generates the reliability sequence $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ from the received sequence $y = (y_1, y_2, \ldots, y_n)$, where y_i is a received signal when x_i is transmitted. The reliability α_i matched to the AWGN channel is the bit-log-likelihood ratio

$$\alpha_i = K \bullet \ln \frac{p(y_i \mid 1)}{p(y_i \mid 0)}, \quad i = 1, 2, \ldots, n \tag{1}$$

where K is an arbitrary positive constant and $p(y_i|x_i)$ is the probability of receiving y_i when x_i is transmitted. The hard decision sequence $y^H = (y^H_1, y^H_2, \ldots, y^H_n)$ of y is then

$$y_i^H = \begin{cases} 0, & \alpha_i = 0, \\ 1 & \alpha_i > 0, \end{cases} \quad i = 1, 2, \ldots, n \tag{2}$$

where α_i indicates the reliability of y_i^H if the codeword is transmitted using BPSK signalling across an AWGN Channel. The elements with the least reliability are considered to be the elements with the most probability of error.

Let U be the set of all positions of a codeword, i.e., $U = \{1, 2, \ldots, n\}$. Now divide the set U into S_x and S_x^c for

Copyright © 2004 IEEE: An earlier version of this paper was first published in AFRICON '04, 15-17 September 2004, Gaborone, Botswana

the codeword x. If $x_i = y^H_i$ then the position i belongs to S_x, otherwise the position i belongs to S_x^c, i.e., $S_x = \{i|x_i = y^H_i,\ i \in U\}$ and $S_x^c = \{i|x_i \neq y^H_i,\ i \in U\}$. Therefore, $U = S_x + S_x^c$.

The Maximum Likelihood Metric (MLM) of the received sequence is then defined as

$$l(y,x) = \sum_{i \in S_x^c} |\alpha_i| \tag{3}$$

where $S_x^c = \{i|x_i \neq y^H_i,\ i \in U\}$. The main aim of the Kaneko decoder is to find the codeword x from the received sequence y, such that the value of $l(y,x)$ is minimized. This value of $l(y,x)$ is calculated for a generated set of codewords based on the received information sequence. Once the MLM is calculated for a candidate codeword, x, we look for the codeword with the lowest MLM among the previously generated candidate codewords and determine that the codeword with the lowest MLM is most likely. If x satisfies equation (4), then we can determine that x is the most likely codeword given the received sequence and can terminate the algorithm. This early termination condition is shown in the equation below.

$$l(y,x) < \sum_{i=1}^{d-\left\lfloor \frac{m_0+m}{2} \right\rfloor} |\alpha_{S_x}(i)| \tag{4}$$

where $m_0 = \| S^c_{x_0} \|$ and $m = \| S^c_x \|$, x_0 is the output of the algebraic decoder when y^H is the input sequence. Here $\|S\|$ stands for the number of elements in the set S.

The Kaneko decoder calculates this MLM for $i = 1$ to $2^T - 1$ decoded codewords. Where T is the number of least reliable bit positions in the received sequence assumed to be in error, i.e. the positions where the amplitudes of the received sequence are smallest. For each of the iterations, an error pattern is generated and added to the hard-decision received word, which is then decoded using an algebraic decoder producing a codeword.

The number of error patterns that are generated depends on the equation given below.

$$l(y,x) < \sum_{i=1}^{d-\left\lfloor \frac{m_0+m}{2} \right\rfloor - t - 1} |\alpha_{S_x}(i)| + \sum_{i=1}^{t=1} |\alpha_u(j+i)| \tag{5}$$

The number of error bits that are generated is determined by the value of j, which satisfies equation (5) calculated during the initial iteration. The Kaneko decoder thus calculates the number of bits assumed to be in error, and then performs a number of decoding iterations that is limited by the number of bits assumed to be in error. If the most likely codeword is decoded and satisfies equation (4) during the iterative process, the decoder terminates and outputs the codeword.

B. Ruling Out Condition

The Kaneko decoder has improved decoding complexity in that the iterations are terminated once the condition of optimality is satisfied by a decoded codeword. Each of the iterations performs algebraic decoding of the received sequence plus a generated error pattern. This error pattern generated is depended on the iteration number as well as the reliability of the received sequence. It can be shown that for a received sequence, decoding of a number of error patterns will result in the same codeword. This implies that some error patterns generated will be useless as they do not produce any new information. In [5] a set of ruling out conditions were proposed, which rule out iterations that are considered useless. These conditions make use of the fact that the error patterns are generated in increasing binary order with the bit number increasing with increasing reliability of the received bit information.

At each stage of the iterative process, when a new nonzero test error pattern is generated, the condition given in the equation below is tested.

$$L(u_{best}) \leq \underline{L}[U_t(e)(u_1, d_{\min})] \tag{6}$$

where u_l is the latest decoded candidate codeword and u_{best} is the best among all the candidate codewords that have been generated previously. If equation (6) holds, the current iteration is skipped and the next candidate error pattern is generated until the iteration limit is reached. Otherwise, algebraic decoding is performed on the received sequence plus candidate error pattern. $L(u)$ is a correlation discrepancy of the given vector u and is defined as

$$L(u) \triangleq \sum_{i \in D_1(u)} |y_i| \tag{7}$$

where $D_1(u) \triangleq \{i : u_i \neq y_i^H,\ \text{and } 1 \leq i \leq n\}$.

The useless iteration bound is defined as

$$\underline{L}[U_t(e)(u_1, d_{\min})] = \sum_{j=1}^{m} |r_{i_j}| + \sum_{i \in D(u_1)} |r_i| \tag{8}$$

Where m is the number of bits in error, i.e. the number of 1s in the candidate error pattern.

The condition given in equation (6) is tested at the start of each iteration of the Kaneko decoder; if the condition is satisfied, the particular iteration is skipped.

The Proposed Decoding algorithm is given below.

begin
  i := 1;
  T: = n;
  l(y,x) : = ∞
while i < 2^T-1 do
    begin
    Algebraically decode $y^H + e^{(i)}$
    If decoder successfully finds x and $l(y,x) < l(y,\hat{x})$ then

```
        begin
            If x satisfies eq (4) then exit else
            Calculate T from eq (5)
            end
        i : = i +1;
        end
       Calculate the ruling out condition eq (6)
       if eq (6) is satisfied
            i = i +1; (skip iteration)
       end
    end
    end
 Exit (Codeword decoded is x̂ )
 End
```

III. SIMULATION

Computer simulations were performed for binary antipodal signals over the additive white Gaussian noise (AWGN) channel using BCH (15, 5, 7) code. The received signal is given by $y_i = \sqrt{E_s} + z_i$ when $x_i = 0$ and $y_i = -\sqrt{E_s} + z_i$ when $x_i = 1$, where E_s is the energy per bit of the channel input and z_i is identically distributed Gaussian random variables with mean 0 and variance $\sigma^2 = N_0/2$, and N_0 is the noise spectral density. The SNR for the channel is given as $\gamma = E_s/N_0$ and the SNR per transmitted information bit is $\gamma_b = \gamma \cdot n/k$. The SNR range considered is from 0dB to 6dB.

Simulations were averaged with a minimum of 200000 samples and at least 150 frame errors per sample point. The decoders compared were the original Kaneko decoder and the proposed decoder with Ruling-Out useless Iterative Steps Conditions. The time complexity of the decoder is defined as the number of iterations that the decoder performs during the decoding process. Initially, the decoder performs algebraic decoding of the received sequence. This decoding is not considered to be part of the iterative process and is not counted as an iteration.

The simulation results show a marked improvement in the complexity of the proposed algorithm as compared to the original Kaneko decoder. The frame error rate in Fig.1 shows that the performance of the proposed decoder matches that of the Kaneko decoder with negligible performance loss at high SNR and also has some performance gain at low SNR. The bit error rate comparison in Fig.2 shows comparable performances of the original and proposed decoding systems, with the proposed decoder having negligible performance loss at high SNR.

From the complexity comparison graphs in Fig.3 it can be seen that the original Kaneko decoder goes through at least one iterative step during the decoding procedure. The proposed decoder however does not always perform this decoding step. This is due to the fact that there is an initial algebraic decoding step that performs optimal decoding at the outset, which then eliminates the need for further decoding iterations.

The complexity comparison in Fig.4 shows an improvement of decoding complexity of at least 58% which then increases with increasing SNR. The reduction in the number of iterations is low at lower SNR due to the fact that at lower SNR there are more received sequences in error that cannot be corrected by single decoding by the algebraic decoder and therefore requires iterations of the decoder. At higher SNR, the algebraic decoder is able to correct a majority of the errors of the received sequence and thus the decoder does not have to perform any iteration. This is seen from fig 3, as the SNR reaches 5dB, the average number of iterations approaches 0 as the algebraic decoder corrects the majority of errors. The percentage reduction in number of iterations therefore approaches 100% as SNR increases as can be seen from Fig 4.

IV. CONCLUSION

This paper proposes a reduced complexity Maximum-Likelihood (ML) decoding Algorithm for Linear Block Codes based on the Kaneko decoder and incorporating ruling out conditions for useless iteration steps. The simulation results for (15, 5, 7) code show a marked improvement in the complexity of the proposed algorithm as compared to the original Kaneko decoder. The frame error rate shows that the performance of the proposed decoder matches that of the Kaneko decoder. The bit error rate shows comparable performances of the original and proposed decoding systems.

From the complexity comparison graphs it can be seen that the proposed decoder has lower complexity of decoding at lower SNRs compared to the Kaneko decoder and has comparable complexity of decoding at higher SNRs. The proposed decoder has lower complexity that makes it preferable for implementation on practical communication systems that have limited processing power. This complexity advantage does not negatively affect the ML decoding performance of the decoder.

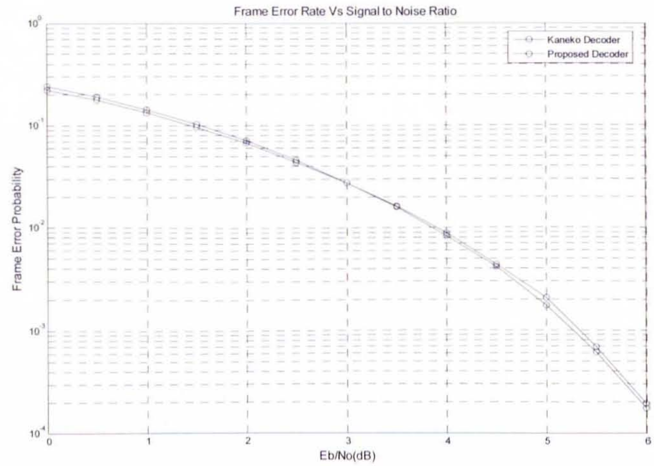

Fig.1 Frame Error Probability comparison between the Kaneko Decoder and the Proposed Decoder

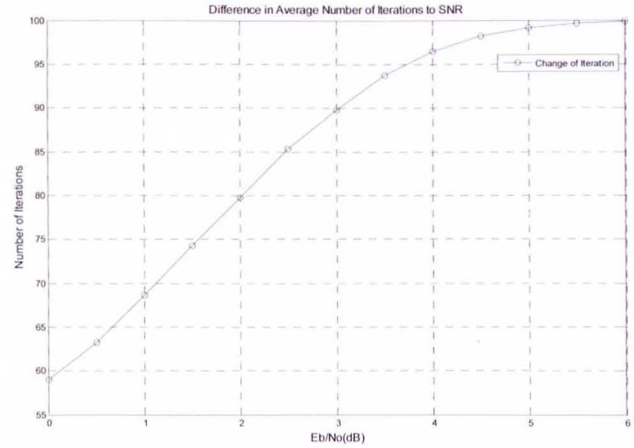

Fig.4 Percentage Reduction in Number of Iterations in Proposed Decoder as compared to the Kaneko Decoder

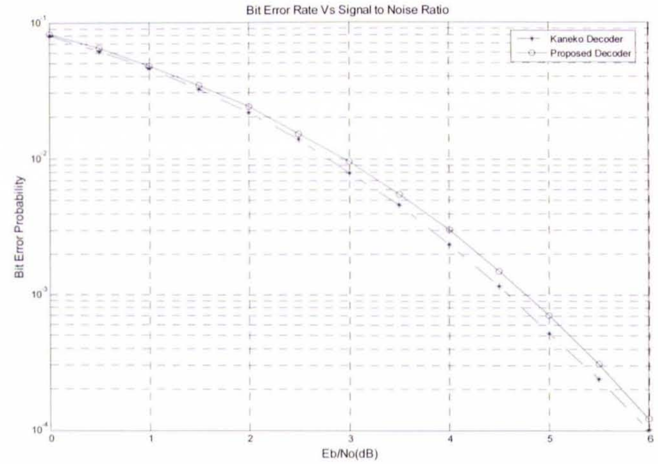

Fig.2 Bit Error Probability Comparison between the Kaneko Decoder and Proposed Decoder

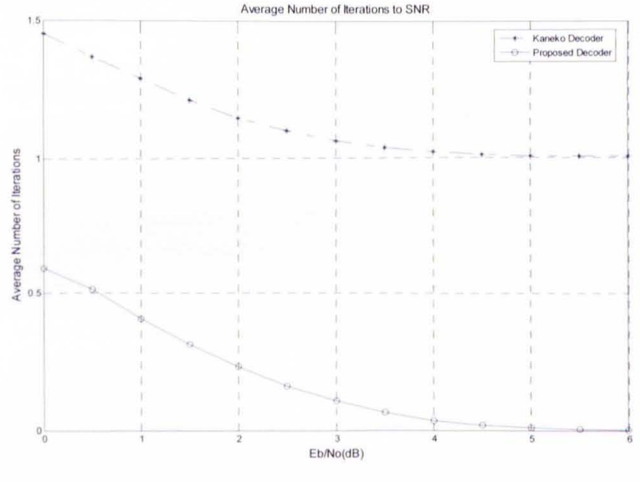

Fig.3 Comparison of average number of iterations of Kaneko Decoder and Proposed Decoder

REFERENCES

[1] Toshimitsu Kaneko, "An efficient Maximum-Likelihood-Decoding Algorithm for Linear Block Codes with Algebraic Decoder", IEEE Transaction on INF Theory Vol. 40, No 2, pp 320 -327 March 1994

[2] Simon Haykin, "Communication Systems", John Wiley & Sons, Inc, ISBN 0-471-57176-8, 1994

[3] Kasami, Y. Tang, T Koumoto & T. Fujiwara, "Sufficient Conditions for Ruling-Out Useless Iterative Steps in a Class of Iterative Decoding Algorithms", IEICE Transactions Fundamentals Vol. E82-A, No 10 pp 2061-2073, October 1999

[4] T Koumoto, T Takata, T Kasami & S Lin, "Conditions for Reducing the Number of Iterations of Iterative Decoding", Tech Report of IEICE, 2001

[5] Toshimitsu Kaneko, "An Improvement of Soft-Decision Maximum-Likelihood Decoding Algorithm Using Hard-Decision Bounded-Distance Decoding", IEEE Transaction on INF Theory Vol. 43, pp 1314 -1319, July 1997

[6] G.D. Forney Jr., "Generalised Minimum Distance Decoding", IEEE Trans. on Information Theory, Vol. 12, pp 125-131, April 1966