DECODING OF LINEAR BLOCK CODES USING SUM PRODUCT ALGORITHM

1KALYANI G, 2GOWTHAM RAJ M, 3AKILAN M, 4AKSHAY GOPIKUMAR MENON, 5GUGAN N, 6PARGUNARAJAN K

1,2,3,4,5Students of final year B.Tech, Department of Electronics & Communication Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, Tamil Nadu, India
6Asst. Professor, Department of Electronics & Communication Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, Tamil Nadu, India
E-mail: koopiganesan@yahoo.com, gowthamrm14@gmail.com, akilam1992@gmail.com, akshay@phenomenon.com, gugan.natarajan@gmail.com, k_pargunarajan@cb.amrita.edu

Abstract- In this paper, we exploit the usage of sum product algorithm for updating the reliability values of the received sequence at the receiver end. The reliability values based on the soft decision decoding scheme is utilized for proceeding in the further process of decoding. This method of decoding is more efficient in comparison to the hard decision process as the probability of every single bit as well as the parity bits are upgraded, thus taking into consideration the interconnections and dependency of every bits with its associated bits. Further it serves as a suboptimal decoding scheme, with reduced computations and searches in comparison to the optimal Maximum Likelihood decoding method that computes the probabilities based on log likelihood function. The comparison of the output obtained by using sum product algorithm or belief propagation is compared with the that obtained by ML scheme and the performance in an Additive White Gaussian Noise channel shows meager tradeoff in the error performance with reduced searches and computations.

Keywords- AWGN channel Additive White Gaussian Noise channel, BP Belief Propagation, ML Maximum Likelihood Decoding, SPA Sum Product Algorithm

I. INTRODUCTION

The process of transmitting message sequence through any channel involves the addition of noise, thus altering them at the receiver end. The process of decoding through the received bit stream to the most likely transmitted message is all that is involved in the decoding process, which may not be precisely the same that was transmitted, thus introducing error in the decoding scheme. This has a very high probability of increase as the Signal to Noise Ratio (SNR) decreases over the channel in which the message is transmitted. The process of deciphering the message at the receiver thus becomes tedious and involves a lot of computations, in terms of search space as well, as the number of bits processed at a time (length of message sequence encoded) increases. This results in the optimum methods, such as ML technique, quite complex. The process of sub optimal decoding techniques, based on reliability of the values can be used in order to meet this problem. The trade off in the error performance as a result of reduced complexity is quite less because of the fact of dependency on reliable bits in the received stream.

Soft decision decoding schemes, such as, Generalized Minimum Distance Decoding Algorithm (GMS), Chase Algorithm, Weighted Erasure Decoding methods, Probability First Search algorithm [1], based on the reliability values of the received sequence were an efficient method for decoding procedure. Belief Propagation can be utilized to update the reliability values of the received sequence before performing the hard decision decoding of the bits. This process of updating the values is done based on the interdependency of the bits involved in process of channel encoding to ensure further reliability in the process[2]. Thus a graphical representation of the various main variables as functions of various local variables can be good source of alternate view of the entire process with more clarity and understanding [1][2][3]. The process of belief propagation can be combined with any type of decoding scheme in order to achieve the desired result and performance at an early stage. The utilization of Factor Graph, a modified version of the trellis diagram can be viewed as an efficient way to graphically view the interdependency of reliability among the transmitted bits.

II. FACTOR GRAPH

Factor graphs play an important role in the graphical representation of the coding stratagem for a more precise and quite lucid description in comparison to the mathematical representation that is often used. A wider view of the entire process is presented by paving way for updating from the previous states already utilized and also to proceed in the right forward path. For a function that includes a lot of sub functions and variables, the representation in the form of factor graph have reduced the complexity that is involved when the same is represented in the form of mathematical equations and notions involving a lot of loops.

A tutorial to factor graphs and its application to sum product algorithm that is discussed briefly in this
paper, is a simple way to understand the various algorithms that have been developed in the field of communication and information theory. These algorithms involve a lot of global functions, which in turn depends on sub or local functions involving a lot of variables. The product and sum combinations of either these variables or one or more local functions constitute a major global function. The factorization of them with the help of factor graph, popularly known as bipartite graph, help in proper visualization of arguments of variables that constitute to form the functions (both local and global)\[2][3].

Assume for example a function that depends on five variables \((u, w, x, y, z)\), can be factorized as products of three other functions, given by the following equation

\[
f(u, w, x, y, z) = f_1(x, y, z) f_2(w, y) f_3(u, x)
\]  

In this function, to explain in simple terms, \(f_1, f_2, f_3\) are the local functions that depend on one or more of the variables \((u, w, x, y, z)\). The function \(f\) that in turn depends on these local functions, factorized as a product of them, denotes the global function. The factors are thus called the local functions and their product constitutes the global function. The factor graph representation of equation (1) is given below in Figure 1.

III. UPDATING OF RELIABILITY VALUES BASED ON BELIEF PROPAGATION

The application of sum product algorithm involves the following steps for decoding:
- Initial Step
- Horizontal Step
- Vertical Step
- Termination

These steps are discussed one by one below for a \((7, 4)\) Hamming Code.

Let \(c = \{c_1, c_2, c_3, \ldots, c_n\}\) be the transmitted valid codeword of length \(n\) after BPSK conversion. This codeword when transmitted through the AWGN channel encounters an addition of noise \((N)\) that results in the reception of \(r = \{r_1, r_2, r_3, \ldots, r_n\}\).

\[
r = c + N
\]

The generator matrix that is used in the encoding scheme plays a very important role in the factor graph representation for the updating scheme. For easier decoding let us consider the generator matrix \((G)\) to be in systematic form (which implies \(k\) symbols of message bits followed by \((n-k)\) parity bits or vice versa) thus having the dimension \(n\times k\).

The parity matrix \((H)\), of dimension \((n-k)\times n\), of the encoding scheme is given by the following expression:

\[
[H] = [-P \mid I]
\]

From the entries of the parity matrix the number of cycles involved in the process of coding can be evaluated and the graph is generally made into a cycle-free graph for the termination of the procedure at a faster rate and reducing the amount of complexity involved in the process. Even a graph that involves cycles can be made into a cycle free one by adhering to the principles and lemma to result in convergence of belief propagation algorithm [4][5][6]. The factor graph representation of the cycle free factorial graph is then constructed to update the reliability values of the received bits. A model factor representation, that is cycle free, of \((7, 4)\) Hamming code is represented below in Figure 2.
The entire procedure can be visualized using the above given factor graph notation that aids better understanding. As an initial step the probability of the bit that is intended to be updated is computed based on the channel specification. Thus in the case of an AWGN channel the probabilities are calculated based on the given formula in equation (4)

\[ p(x_i = b|r_i) = 1 / (1 + e^{-b*r_i*N_0}) \]  

(4)

where \( b \) represents whether the bit is ‘0’ or ‘1’ and \( r_i \) represents the soft decision input of the received bit. Likewise the probabilities of being a ‘0’ or ‘1’ of every single bit can be evaluated. Now, any particular bit’s probability is affected by the input from the check nodes to which they are connected, which in turn depends on the other variable nodes the corresponding check bits are associated with. A clear detail of these interdependency can be visualised from the factor graph representation the parity matrix with variable and check nodes in them.

Probabilities are initially sent to the various check nodes (j) from the selected variable node (i). This covers the initialization step of the decoding process using the sum product algorithm.

\[ q_{ij}(1) = p_i(1) \]  

(5)

\[ q_{ij}(0) = 1 - p_i(1) \]  

(6)

here the notation \( q \) is used for the probabilities form variable node (i) to the related check node(s), in this case only \( j \), and \( i \) & \( j \) denote the designations of the nodes.

To update the computed probabilities at every variable nodes inputs from the dependant check nodes are evaluated that constitutes the horizontal step. The input to the check nodes from the other associated variable nodes come into picture for this computation. The probability of the check node, at consideration, to be ‘0’ is given the below probability (i’ includes all other variable nodes excluding i)

\[ r_{ji}(0) = 0.5 + 0.5 \pi (1 - 2q_{i’j}(1)) \]  

(7)

Thus the inputs from other variable nodes to the check node are used for this computation. Here \( r \) denotes the probability from check node to the variable node. The probability of being ‘1’ can be calculated by subtracting the above probability from unity.

The algorithm obtains the name “sum product” as the expression (7) denotes the summation of product of the probabilities. Thus the computations involve summation of the product of probabilities. These probabilities can be now utilized to update the probabilities computed during the initial step of the procedure. These computations are far less in comparison to the computations required for MLD scheme that involves \( 2^n \) computations for every set of \( k \) bits in a \( (n, k) \) block coding technique. This can be done by the following procedure

\[ Q_i(1) = K_i * p_i * \pi r_{ji}(1) \]  

(8)

\[ Q_i(0) = K_i * p_i * \pi r_{ji}(0) \]  

(9)

The value of \( K \) is named as the scaling factor to sum up the probabilities to unity. The scaling factor can be evaluated from the new probabilities for ‘1’ and ‘0’ as

\[ K_i = 1 / (Q_i(0) + Q_i(1)) \]  

(10)

Thus the updated probability becomes the new probability of the variable node that is further used in the comparison of the probabilities of other variable nodes. This procedure can be repeated for various iterations for all of the bits until a convergence is made that can result in the termination of the updating procedure [4][5].

As the convergence procedure is made utilization to obtain termination the decision of the bit being a ‘0’ or ‘1’ can be computed using the decision making rule given in equation (11).

\[ \{ \begin{array}{l l} p_i = 0 & \text{if } Q_i(0) > Q_i(1) \\ p_i = 1 & \text{if } Q_i(1) > Q_i(0) \end{array} \]  

(11)

IV. APPLICATION OF SUM PRODUCT ALGORITHM

As discussed earlier this algorithm can be used along with other soft decision reliability based algorithm in order to increase the error performance of the sub optimal decoding scheme for an AWGN channel [6]. Rather the Hard Decision scheme, that makes use of a single threshold to determine whether the received bit is ‘0’ or ‘1’, using the soft input as such in order to update the probability based on the dependency on other bits as well can be much utilization to increase the error performance in any decoding algorithm. Though the computation of the sub optimal decoding scheme increases due to summation and products utilized the error performance increases.

Simulation result was carried out for a source of \( 10^5 \) bits in an AWGN channel for a snr (signal to noise ratio) range of 0dB to 10dB. The computations involved in decoding the received bits stream by comparing the computations and error performance with respect to ML decoding scheme presents a clear picture of the reduction in the computations and searches by trading off with the error performance of the system. The trade off can be further reduced by
Decoding of Linear Block Codes Using Sum Product Algorithm

working hand in hand with any reliability based decoding algorithm as mentioned earlier.

<table>
<thead>
<tr>
<th>Type of decoding scheme used</th>
<th>Number of computations involved for every 4 bits of message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML decoding scheme</td>
<td>128</td>
</tr>
<tr>
<td>Sum Product Algorithm</td>
<td>63</td>
</tr>
</tbody>
</table>

Table 1. Overview of the performances of MLD Vs SPA for (7, 4) Hamming code

<table>
<thead>
<tr>
<th>Type of decoding scheme used</th>
<th>Number of computations involved for 10^5 bits of message</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML decoding scheme</td>
<td>3200000</td>
</tr>
<tr>
<td>Sum Product Algorithm</td>
<td>1575000</td>
</tr>
</tbody>
</table>

Table 2. Overview of the performances of MLD Vs SPA for (7, 4) Hamming code involving a source of 10^5 bits

Graph 1. Illustrating the performance of MLD for (7, 4) Hamming code involving a source of 10^5 bits for snr range of 0dB to 10dB

Graph 2. Illustrating the performance of SPA for (7, 4) Hamming code involving a source of 10^5 bits for snr range of 0dB to 10dB

REFERENCES


★★★