International Journal of Computer Trends and Technology- volume4Issue3- 2013

A FEC Decoding in LTE and WiMAX Systems

G.Vamsikrishna[1], T.PavanKumar[2], V.Srikanth[3]

[1]*M.Tech, Department of CSE, K.L.University, Andhra Pradesh, India*
[2]*Assoc.professor, Department of CSE, K.L.University, Andhra Pradesh, India*
[3]*professor, Department of CSE, K.L.University, Andhra Pradesh, India*

Abstract:

Recent wireless systems such as EDGE, WIMAX, LTE are using LDPC, tail biting and turbo convolution codes as the forward error correction codes (FEC) for the data and overhead channels. Therefore many decoding algorithms are introduced for decoding these codes. Using different decoding approaches lead to different hardware architectures. As, in new wireless systems these codes work side by side a single universal decoder which is efficient in handling decoding of all the codes.

***Keywords*—**EDGE, WIMAX, LTE, FEC.

I. INTRODUCTION

In this paper we are mainly dealing with forward error correction (FEC) which is a technique used for controlling data errors in unreliable or noisy channels .In this the sender encodes the data using a error correction code(ECC) which allows the receiver to correct the error occurred during the transmission and to recover the data correctly.

The two categories of FEC codes are block codes and convolutional codes as follows.

- Block codes work on fixed-size blocks (packets) of bits or symbols of predetermined size. Practical block codes can generally be decoded in polynomial time to their block length.
- Convolutional codes work on bit or symbol streams of arbitrary length. They are most often decoded with the Viterbi algorithm, though other algorithms are sometimes used. Viterbi decoding allows asymptotically optimal decoding efficiency with increasing constraint length of the convolutional code, but at the expense of exponentially increasing complexity.

Most convolutional decoding algorithms are based upon either algebraically calculating the error pattern or using trellis graphical representation. With the advent in turbo coding a new decoding approach was introduced know as iterative decoding which is a framework based on bipartite graphs for the description of LDPC codes and decoding via BP algorithm.

Turbo code were introduces in 1993 which is a parallel concatenation of two encoders inter connected by an interleaver .The decoding of the code is an iterative process in which each component decoder takes the information provided by the previous block decoder. Later this iterative process decoding process is extended to serially concatenate convolutional codes.

The success of turbo codes made rediscovery of another code with similar performance and characteristics. These LDPC codes implement sparse parity-check matrices H.LDPC codes are well represented by bipartite graphs where a set of variable nodes represent the code words and a set of nodes represent the parity-check which defines the code. In these codes the number of edges will be the same for regular data were as the degree of edges change based on the distribution in irregular codes.

Both turbo codes and LDPC codes are being studied for past fifteen years. The relation between these codes is unclear until Mackay claimed that turbo codes are LDPC codes. McEliece showed that the decoding algorithms used for decoding these codes fall in to the same category as BP and Bayesian network.

In LTE and WIMAX systems, the decoding algorithms proposed are based on Viterbi and MAP algorithms for decoding the tail-biting and turbo codes. There are many other algorithms proposed for decoding tail-biting convolutional codes and turbo codes. The reduced wrap-around Viterbi algorithm was proposed to decode tail-biting convolution codes in WIMAX system to reduce the number of iterations performed for decoding and memory usage. Another decoding algorithm such as double trackback and bidirectional Viterbi algorithm were proposed for decoding in LTE.

II. CONVOLUTIONAL CODES

Convolutional codes are type of channel codes which are error correction codes .these codes permit reliable communication of an information sequence over a channel that adds noise, introduces bit errors, or otherwise distorts the transmitted signal. Elias introduced convolutional codes in 1955. These codes have found many applications, including deep-space communications and voice based modems. Convolutional codes continue to play a role in low-latency applications such as speech transmission and Constituent codes in Turbo codes.

A. Convolutional encoding

The binary convolutional codes are one of the most popular forms of binary error correcting codes that have found numerous applications. To convolutionally encode data, using k memory registers holding 1 input data bit. Initially all registers start with 0.The encoder use n modulo-2 adder implemented using single XOR gate (where the logic is: 0+0 = 0, 0+1 = 1, 1+0 = 1, and 1+1 = 0).An input bit is applied into the first register using the remaining vales and the generator polynomials, the encoder will output n bits. We perform right shift by one bit a time i.e. m_1 moves to m_0, m_0 movies to m_{-1} waits for the remaining input bits. The process id continued until all the registers have returned to zero state.

The figure below is a rate 1/3 (*m*/*n*) encoder with constraint length (*k*) of 3. Generator polynomials are *G*1 = (1, 1, 1), *G*2 = (0, 1, 1), and *G*3 = (1, 0, 1). Therefore, output bits are calculated (modulo 2) as follows:

n1 = m1 + m0 + m-1
n2 = m0 + m-1
n3 = m1 + m-1.

B. Convolutional decoding

Binary convolutional codes are one of the popular convolutional codes which are used in numerous applications. Convolutional codes are referred as tail-biting when the starting state of the encoder is equal to the ending state and the state value need not be zero state. Tail-biting can be represented by the parity-check and generator matrices.

1) Parity check matrix for tail-biting codes:

In order to represent the tail-biting codes using tanner graph for applying the BP algorithm we need to calculate the generator and parity check matrices. The matrix representation can be illustrated by a simple example:

Example 1: Consider the convolutional code with rate R = k/n = 1/2, where k represents the number of input bits and n the output bits. Assume that the information sequence is
x = (x0, x1, x2...).
The encoder will convert the input in to the following sequences
y(0) = (y(0) 0 , y(0) 1 , y(0) 2 , ...) and
y(1) = (y(1) 0 , y(1) 1 , y(1) 2 , ...) .
If there are multiple input streams, we can refer to a single interleaved input
x = (x (0) 0, x (1) 1 ...).
Also, the output streams are multiplexed to create a single data stream
y= (y(0) 0 y(1) 0 , y(0) 1 y(1) 1 , ...)
Where **y** is the convolutional code word. In addition, each element in the interleaved output stream **y** is a linear combination of the elements in the input stream
x = (x(0) 0 , x(1) 0 , ..., x(k−1) 0 , x(0) 1 , x(1) 1 , ..., x(k−1)..).

An impulse response g(j) is obtained from the encoder by applying a single 1 at the input and a string of zeros, where strings of zeros are applied to all the remaining inputs. The impulse responses for the encoder in our example are

$$g^{(0)} = (1011),$$
$$g^{(1)} = (1101).$$

These impulsive responses obtained are the values of the generator matrix. These code words are similar to the generator polynomials and code words of a cyclic code.
The generator sequences can be expressed in the
Following general form:

$$y_t^{(j)} = \sum_{i=0}^{m-1} xt - 1 g\, t\, (j) \quad .$$

These expressions can be represented as a matrix multiplication, providing a generator matrix equal to that of the block codes. The deference between these two types of codes arises at the input length where block have a fixed length whereas the convolutional codes are semi-infinite as there is no constraint on the input length. The generator and parity check matrices will be

$$
\mathrm{G} = \begin{bmatrix}
G_m & .. & .. & & G_0 & G_1 & .. & .. & G_{m-1} \\
G_{m-1} & G_{m-2} & & & & G_0 & & & G_{m-2} \\
G_{m-2} & G_{m-1} & \cdots & & & & & & \cdots\ \cdots\ \cdots \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\
 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\
G_0 & G_1 & .. & .. & G_{m-1} & G_m & & .. & .... \\
 & G_0 & & G_1 & & & & G_m & \\
 & & & \cdots & \cdots & \cdots & & & \\
 & & & \cdots & \cdots & \cdots & & & \\
 & & & & & G_0 & G_1 & .... & G_m
\end{bmatrix}
$$

$$
\mathbf{G}_l = \begin{bmatrix}
g_{1,l}^{(1)} & g_{1,l}^{(2)} & \cdots & g_{1,l}^{(n)} \\
g_{2,l}^{(1)} & g_{2,l}^{(2)} & \cdots & g_{2,l}^{(n)} \\
\vdots & \vdots & & \vdots \\
g_{k,l}^{(1)} & g_{k,l}^{(2)} & \cdots & g_{k,l}^{(n)}
\end{bmatrix}
$$

Each block of rows in the matrix G is obtained by a circular shift by n positions of the previous block. The parity-check matrix of a rate k/n tail-biting convolutional code with constraint length m is

$$
\mathrm{m}= \begin{bmatrix}
P_0^T|I & \cdots\cdots & P_m^T|O & P_{m-1}^T|O & \cdots\ .. & P_1^T|O \\
P_1^T|I & \cdots\cdots & P_{m-1}^T|O & P_m^T|O & \cdots\ .. & P_2^T|O \\
 & & \ddots & & & \\
\cdots\cdots\cdots & & P_1^T|I & P_0^T|I & \cdots\cdots & P_m^T|O \\
P_m^T|O & \cdots\cdots & P_1^T|O & P_0^T|O & \cdots\ \cdots & \\
 & & \ddots & & & \\
 & \cdots & \cdots & & \cdots\cdots\ \cdots & \\
 & P_m^T|O & P_{m-1}^T|O & \cdots\ \cdots & P_1^T|O & P_1^T|I
\end{bmatrix}
$$

Let us consider an example of binary (12, 6) fed to an encoder. Then the tail-biting construction gives a binary (12, 6) code with generator and parity check matrices.

The generator matrix G obtained will be

$$
\mathrm{G} = \begin{bmatrix}
11\ 01\ 10\ 11\ \ 00\ 00 \\
00\ 11\ 01\ 10\ 11\ 00 \\
00\ 00\ 11\ 01\ 10\ 11 \\
11\ 00\ 00\ 11\ 01\ 10 \\
10\ 11\ 00\ 00\ 11\ 01 \\
01\ 10\ 11\ 00\ 00\ 11
\end{bmatrix}
$$

The parity check matrix H

$$
\mathrm{H} = \begin{bmatrix}
11\ 00\ 00\ 11\ 01\ 10 \\
10\ 11\ 00\ 00\ 11\ 01 \\
01\ 10\ 11\ 00\ 00\ 11 \\
11\ 01\ 10\ 11\ 00\ 00 \\
00\ 11\ 01\ 10\ 11\ 00 \\
00\ 00\ 11\ 01\ 10\ 11
\end{bmatrix}
$$

When looking at the matrix H obtained from applying tail-biting convolutional code, we know that the matrix H is similar to that of the H matrix of the irregular LDPC codes in which the number of non-zero elements are not fixed.

III. CODING STRUCTURE IN WIMAX AND LTE

To address the low and high rate requirements of LTE, the 3rd Generation Partnership Project (3GPP) working group undertook a rigorous study of advanced channel coding candidates such as tail-biting convolutional and turbo codes for low and high data rates, respectively. We study the application of BP decoder for the turbo codes in LTE system. Here we focus on the FCH which has shorter payload size of 12 and 24 bits.

1) Tail-biting convolutional code in WIMAX:

In this session we discuss about the frame control header structure. The WiMAX use the OFDMA for the data transmission in the physical layer, the pay load will be 12 or 24 bits as stated and smallest unit for data transmission in sub channel. A sub channel contains 96 coded bits which translates to a 48 bits minimal information block. At present the FCH payload bits are repeated to meet the minimum number of encoder bits. The generator polynomials of WiMAX tail-biting convolutional code are
g1 = (1011011) and
g2 = (1111001)

2) Turbo code in LTE system:

The 3GPP turbo code is a systematic parallel concatenated convolutional code (PCCC) with two 8-state constituent encoders and one turbo code internal interleaver. Each constituent encoder is independently terminated by tail bits. For an input block size of K bits, the output of a turbo encoder consists of three length K streams, corresponding to the systematic and two parity bit streams (referred to as the “Systematic”, “Parity 1”, and “Parity 2” streams in the following), respectively, as well as 12 tail bits due to trellis termination. Thus, the actual mother code rate is slightly lower than 1/3. In LTE, the tail bits are multiplexed to the end of the three streams, whose lengths are hence increased to (K + 4) bits each [5]. The transfer function of the 8-state constituent code for the PCCC is:

G (D) =$[1, \frac{g1(d)}{g0(d)}]$

Where
g0 (D) = 1 + D2 + D3, (17)
g1 (D) = 1 + D + D3.

IV. CONCLUSION

In this paper, we demonstrated the use of decoding tail biting convolutional and turbo codes using the BP algorithm. By using this algorithm decoding the tail biting convolutional code in WiMAX system Speeds the error correction and reduces the decoding complexity when compared to the ML-Viterbi algorithms. In addition the BP algorithm has only an initial Decoding delay that avoids the intermediate decoding delays those usually occur. With the reduced decoding complexity of the BP decoder the end to end efficiency is increased with minimal hardware effort.

V. RFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, ―Near Shannon limit error-correcting coding and decoding: Turbo codes, ‖ IEEE Intl. Conf. on Commun. vol. 2, Geneva, Switzerland, pp. 1064-1070, 1993.

[2] R. M. Tanner, ―A recursive approach to low complexity codes, ‖ IEEE Trans. Inform. Theory, vol. 27, no. 5, pp. 533-547, 1981.

[3] G. D. Forney, Jr., ―Codes on graphs: normal realizations, ‖ IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 520–548, 2001.

[4] H. H. Ma and J. K. Wolf, ―On Tail Biting Convolutional Codes, ‖ IEEE Trans. On Commun., vol. 34, no. 2, pp. 104-111, 1986.

[5] N. Wiberg, "Codes and Decoding on General Graphs," Linkoping Studies in Science and Technology, Dissertation No. 440, Linkoping University, -Linkoping, Sweden, 1996.

[6] R. J. McEliece, D. MacKay, and J.-Fu Cheng, "Turbo Decoding as an Instance of Pearl's "Belief Propagation" Algorithm," IEEE Trans. On Commun., vol. 16, no. 2, pp. 140-152, 1998.

[7] G. Colavolpe, "Design and performance of turbo Gallager codes," IEEE Trans. On Commun., vol. 52, no. 11, pp. 1901-1908, 2004.

[8] T. T. Chen, and S-He Tsai, "Reduced-Complexity Wrap-Around Viterbi Algorithms for Decoding Tail-Biting Convolutional Codes," 14th European Wireless Conference, Jun. 2008.

[9] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved LDPC Codes Using Irregular Graphs," IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 585-598, 2001.

[10] Giulietti A., "Turbo Codes: Desirable and Designable," IKluwer Academic
Publishers, ISBN: 1-4020-7660-6, 2004.
[11] P. Elias, "Coding for Noisy Channels," IRE Conv. Rec., vol. 3, pt. 4, pp. 37–46, 1955.

[12] A. J. Viterbi, "Convolutional Codes and their Performance in Communication Systems," IEEE Trans. On Commun. vol. 19, no. 15, pp. 751-772, 1971.

[13] C. Pollitt, D. Declercq, and T. Lestable, "Efficient Decoding of Turbo Codes with Nonbinary Belief Propagation," EURASIP Journal on Wireless Communications and Networking, vol. 2008, no. 473613, 2008.

[14] A. Refaey, S. Roy, and P. Fortier, "On the Application of BP Decoding to Convolutiona and Turbo Codes," Asilomar Conference on Signals, Systems, and Computers, Nov. 2009.

[15] P. Stahl, J. B. Anderson, and R. Johannesson, "Optimal and near-optimal encoders for short and moderate-length tail-biting trellises," IEEE Trans. Inform. Theory, vol. 45, no. 7, pp. 2562-2571, 1999.

[16] W. Yu, M. Ardakani, B. Smith, and F. FKschischang, "Complexityoptimized low-density parity-check codes for Gallager decoding algorithm B," IEEE International Symposium on Information Theory (ISIT) s, Sep. 2005.14