

## IEEE802.16e 対応 LDPC デコーダの高性能化に関する検討

我喜屋 良太<sup>†</sup> 和田 知久<sup>‡</sup>

<sup>†</sup> 琉球大学理工学研究科 〒903-0213 沖縄県西原町字千原1番地

<sup>‡</sup> 琉球大学工学部情報工学科 〒903-0213 沖縄県西原町字千原1番地

E-mail: <sup>†</sup> ryosan@lsi.ie.u-ryukyu.ac.jp, <sup>‡</sup> wada@ie.u-ryukyu.ac.jp

あらまし 本稿では、IEEE802.16e用の低密度パリティチェック符号(LDPC符号)の復号器の高速化を目的として、部分並列手法を提案する。特に計算量の多い符号化率 1/2 に注目して提案方式の説明を行う。提案手法では検査行列のモデルマトリックス Hbm の行と列の処理順序を入れ替える事により、効率よく行処理・列処理に更新値を使用できるようにし、エラー訂正性能の向上を実現した。

キーワード Low-Density-Parity-Check (LDPC)符号, WiMAX, IEEE802.16e,

## Investigation on high performance LDPC decoder for IEEE802.16e

Ryota GAKIYA<sup>†</sup> and Tomohisa WADA<sup>‡</sup>

<sup>†</sup> Graduate School of Engineering and Science, University of the Ryukyus

1 Senbaru, Nishihara-tyo, Okinawa, 903-0213 Japan

<sup>‡</sup> Faculty of Information Engineering, University of the Ryukyus 1 Senbaru, Nishihara-tyo, Okinawa, 903-0213 Japan

E-mail: <sup>†</sup> ryosan@lsi.ie.u-ryukyu.ac.jp, <sup>‡</sup> wada@ie.u-ryukyu.ac.jp

**Abstract** In order to realize high speed and high performance Low Density Parity Check (LDPC) Decoder for IEEE 802.16e, in another word "Mobile WiMAX", A Partial Parallel processing method is proposed. In addition, Parity Check model matrix Hbm's rows and columns are inter-replaced. The replacement of the rows and columns make best use of the updated values which is used in LDPC decoding algorithms.

**Keyword** Low-Density-Parity-Check (LDPC) code, WiMAX, IEEE802.16e

### 1. はじめに

低密度パリティチェック(LDPC)符号は、欧州の衛星デジタル放送規格である DVB-S2, IEEE802.3, IEEE802.16e 等への採用が決まっており、シャノン理論限界にせまる次世代の誤り訂正符号として注目されている。本稿では 2005 年 12 月に標準化が完了された IEEE802.16e の LDPC 符号について着目した。モバイル用途の規格であるためできるだけコストを抑えつつ良い特性を得られる LDPC 復号器のアルゴリズムを検討することを目的とする。

### 2. LDPC 符号の復号法

LDPC 符号は、2 元の  $m \times n$  検査行列 H によって定義される符号である。コードワード c と検査行列 H の積  $cH$  が 0 になる特性を持つ。LDPC 符号の復号法には、確率領域 sum-product 復号法と、対数領域 sum-product 復号法がある [1]。本質的にはどちらも等価な復号法であるが、ここではハードウェア実装に相性の良い対数ドメインの復号法を対象とする。

#### 2.1. sum-product 復号法

以下に対数尤度比 (log likelihood ratio) に基づく sum-product アルゴリズムについて説明する。

- ステップ 1(初期化) 検査行列  $H=[H_{mn}]$  に対して  $H_{mn}=1$  を満たす全ての組  $(m,n)$  に対して対数事前値比  $\beta_{mn}=0$  とする。また、反復回数のカウンタとする変数  $l=1$  とし、反復最大回数を変数  $l_{max}$  に設定する。

- ステップ 2(行処理)  $m=1,2,\dots,M$  の順に  $H_{mn}=1$  を満たす全ての組  $(m,n)$  に対して、次の更新式を利用して対数外部値比  $\alpha_{mn}$  を更新する。

$$\alpha_{mn} = \left( \prod_{n' \neq n} \text{sign}(\lambda_{n'} + \beta_{mn'}) \right) f \left( \sum_{n' \neq n} f(\lambda_{n'} + \beta_{mn'}) \right)$$

ここで、

$$\text{sign}(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

であり、関数  $f(x)$  は、

$$f(x) = \ln \frac{\exp(x)+1}{\exp(x)-1}$$

と定義される。この関数  $f$  を Gallager の  $f$  関数と呼ぶ。

2 値入力 AWGN 通信路の場合には第  $n$  シンボルの対数尤度比は

$$\begin{aligned} \lambda_n &= \ln \frac{P(y_n|x_n=0)}{P(y_n|x_n=1)} \\ &= \ln \frac{1/(\sqrt{2\pi\sigma^2}) \exp(-(y_n-1)^2/(2\sigma^2))}{1/(\sqrt{2\pi\sigma^2}) \exp(-(y_n+1)^2/(2\sigma^2))} \\ &= -(y_n-1)^2/(2\sigma^2) + -(y_n+1)^2/(2\sigma^2) \\ &= 2y_n/\sigma^2 \end{aligned}$$

により定義される。

- ステップ 3(列処理)  $n=1,2,\dots,N$  の順に  $H_{mn}=1$  を満たす全ての組  $(m,n)$  に対して、次の更新式を利用して  $\beta_{mn}$  を更新する。

$$\beta_{mn} = \lambda_n + \sum_{m' \in B(n), m'} \alpha_{m'n}$$

- ステップ 4(一時推定語の計算)  $n=1,2,\dots,N$  について

$$\hat{c}_n = \begin{cases} 0, & \text{if } \text{sign} \left( \lambda_n + \sum_{m' \in B(n)} \alpha_{m'n} \right) = 1 \\ 1, & \text{if } \text{sign} \left( \lambda_n + \sum_{m' \in B(n)} \alpha_{m'n} \right) = -1 \end{cases}$$

を計算する。

- ステップ 5(パリティ検査) 一時推定語が符号語になっているかどうかを検査する。もし、

$$(\hat{c}_1, \dots, \hat{c}_n)$$

$$(\hat{c}_1, \dots, \hat{c}_n) H^T = 0$$

を満たせば、 $(\hat{c}_1, \dots, \hat{c}_n)$  を推定語として出力し、

アルゴリズムを終了する。

- ステップ 6(反復回数のカウント) もし  $l < l_{\max}$  ならば  $l$  をインクリメントしてステップ 2 に戻る。  $l = l_{\max}$  ならば、 $(\hat{c}_1, \dots, \hat{c}_n)$  を推定語として出力し、アルゴリズムを終了する。

## 2.2. min-sum アルゴリズム

前節の sum-product アルゴリズムで実装を行おうと

すると、Gallager の  $f$  関数が問題になる。 $f$  関数の計算コストが高いため、ルックアップテーブルを用いたり、線形補間などで対応するが、行処理回路を配置した数だけそれを用意しなくてはならない。

そこで、 $f$  関数を近似して行う min-sum アルゴリズムがある。

$$\begin{aligned} & f \left( \sum_{n' \in A(m), n} f(|\lambda_{n'} + \beta_{mn'}|) \right) \\ & \equiv f \left( f \left( \min_{n' \in A(m), n} |\lambda_{n'} + \beta_{mn'}| \right) \right) \\ & = \min_{n' \in A(m), n} |\lambda_{n'} + \beta_{mn'}| \end{aligned}$$

よって、

$$\alpha_{mn} = \left( \prod_{n' \in A(m), n} \text{sign}(\lambda_{n'} + \beta_{mn'}) \right) \min_{n' \in A(m), n} |\lambda_{n'} + \beta_{mn'}|$$

を用いる。これは、 $f$  関数の性質  $f=f^{-1}$  を利用している。

近似を行う事により、復号特性の劣化が予想されるが、文献[3]、[4]のように簡単な処理を行うことにより、sum-product 復号法と同程度の特性を得ることができる modified-min-sum アルゴリズムが知られている。本稿では文献[2]のようにハードウェア実装を行いやすいよう  $\beta$  の更新値を 0.75 倍にして行った。

$$\beta_{mn} = 0.75 \times \left( \lambda_n + \sum_{m' \in B(n), m'} \alpha_{m'n} \right)$$

## 3. IEEE802.16e

IEEE802.16e は、移動端末向けの高速度無線通信規格のことで、モバイル WiMAX とも呼ばれる。固定無線通信の WiMAX を移動中でも通信を快適に行うための規格である。

2005 年 12 月に標準化が完了され、時速 120km 以下の移動環境でも安定した通信が行えるようになり、通信速度は最大 75Mbps 程度、3~5km 程のエリアをカバーする規格である。

### 3.1. IEEE802.16e 用 LDPC 符号の構成法

ここでは IEEE802.16e 用 LDPC 符号について今回必要な部分の詳細を述べる。LDPC 符号では  $m \times n$  の検査行列  $H$  で定義される符号である。 $n$  は送信語のビット数で、 $m$  はパリティチェックビット数である。情報ビット数は  $k=n-m$  となる。検査行列  $H$  は以下の用に定義される。



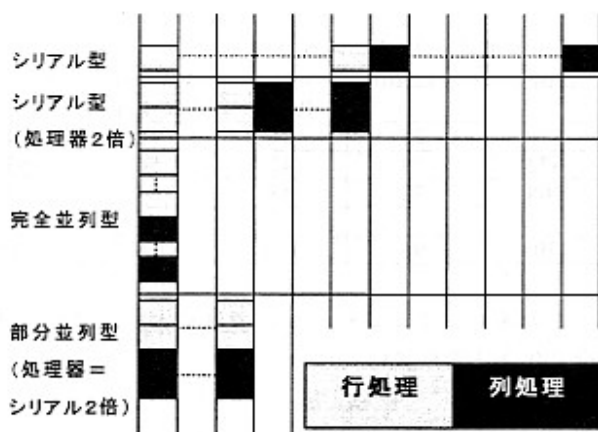


図 1：繰り返し回数 1 回あたりの実行時間

シリアル型は行処理を全部行って後に列処理を行うため 1 回の繰り返しに符号長が大きければ大きいほど時間を要してしまう。そこで、行処理回路、列処理回路を 2 倍に増やすことで、実行時間を 1/2 にすることができる。

完全並列型の場合、全ての行・列に対し処理器が設けられるため、1 回の計算で繰り返し 1 回分を計算することができる。

部分並列型の場合、この図ではシリアル型の処理器 2 倍と同じ処理器と仮定して示している。行処理、列処理が並列に実行される為、シリアル型より効率よく計算時間の削減が行える。

本稿のターゲットである IEEE802.16e 用 LDPC 符号の最大サイズは、 $2304 \times 1152$  である。その為、完全並列型で実装しようとする于行処理回路が 1152 個、列処理回路が 2304 個必要になる。

以上のことから、本稿では部分並列型を使用することにする。

### 5. 提案手法

本稿のターゲットである LDPC 符号は、もともと  $z \times z$  に分割された検査行列を用いているので、それを元に部分並列化を行う。

行処理は奇数ブロック、列処理は偶数ブロック、その後、行処理は偶数ブロック、列処理は奇数ブロックを行う。これを単純部分並列型とする。

以下に Hbm の値が 0 又は、正の値の部分に着色した図を示す。

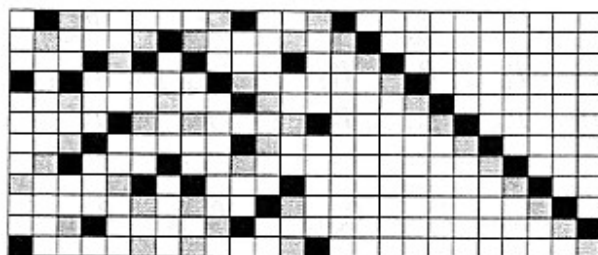


図 2：符号化率 1/2 の Hbm

ここで、黒く示した部分は行処理、列処理が同時に行われてしまう部分になっている。つまり、行処理、列処理で計算された更新値をうまく活用できない場合がある。

そこで、その問題を解決するために、以下を提案する。

まず始めに Hbm の行を [1,3,5,12,7,9,11,2,4,6,8,10] の順番に置換をする。そうすると図 3 のようになる。

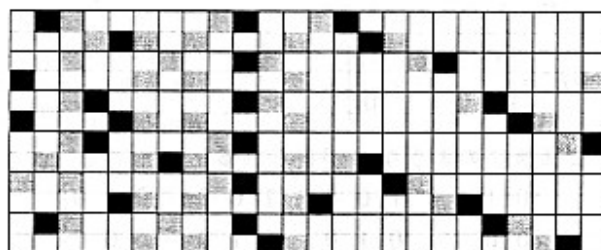


図 3：行置換後の Hbm

こうすることにより、2 行ブロック毎に見ると同じ列の位置に要素が存在する場所が無くなっている。つまり、2 行ブロック毎に上下に重ならなくなった事により、奇数行・偶数行ブロックで出力される更新値を同時に利用する列処理がなくなる。

次に列を次の順に置き換えて説明する。

[2,3,4,7,9,10,11,13,17,19,20,23,1,5,6,8,12,14,15,16,18,21,22,24]

実際にはリストの前半部分、後半部分において列処理を行うため、入れ替えは行わない。なお、行処理は先ほどと同じく奇数行、偶数行ブロックで実行する。

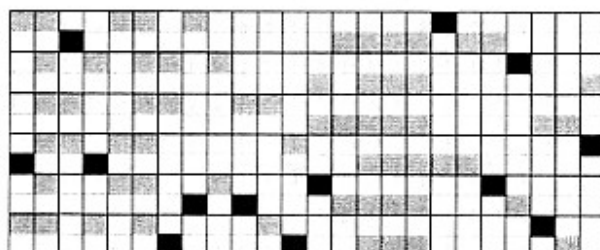


図 4：列置換後の Hbm

入れ替え後、行処理が奇数ブロックを処理する場合、列処理は後半半分を行い、行処理が偶数ブロックを処理する場合、列処理は前半部分を行う。

これを行うことにより、黒で示される行処理・列処理が同時に行われる部分を減らす事ができ、計算された更新値をうまく活用できないケースを減らすことで、繰り替えしの復号処理の性能向上、すなわち少ない繰り返し回数での復号処理を実現することができる。

## 6. シミュレーション

単純に並列化を行った mod-min-sum アルゴリズム、提案手法、そして性能を比較するためにシリアル型の mod-min-sum の 3 つの手法を matlab 上で作成し比較検討を行った。

### 6.1. matlab によるシミュレーション結果

$z=96$ , 符号化率 1/2 の Hbm を使用し測定を行った。

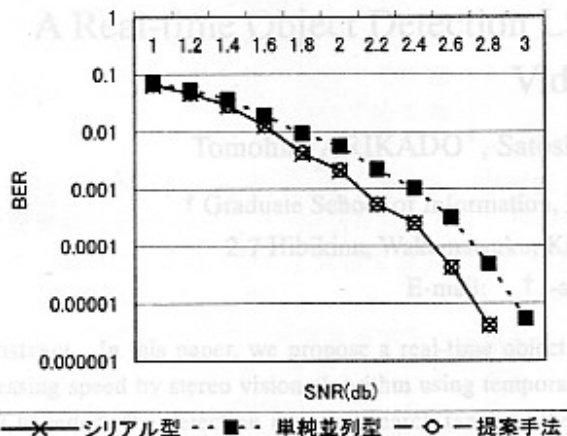


図 5 : 繰り返し 10 回での各手法の BER

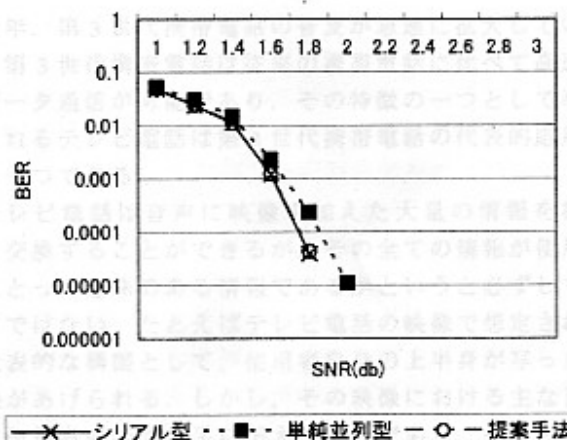


図 6 : 繰り返し 20 回での各手法の BER

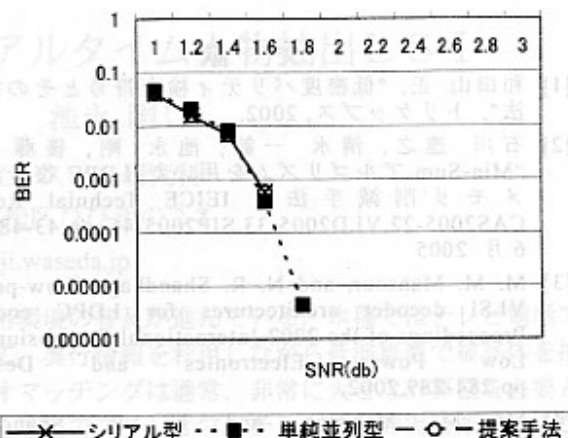


図 7 : 繰り返し 30 回での各手法の BER

IEEE802.16e 用 LDPC 符号の部分のみのシミュレーションを行うため、AWGN+BPSK 通信路での通信と仮定し、符号長 2304, SNR は 1~3db でシミュレーションを行った。図 5 は繰り返し回数 10 回, 図 6 は繰り返し回数 20 回, 図 7 は繰り返し回数 30 回での実行結果である。まず、繰り返し回数 10 回では、提案型はシリアル型とほぼ同等の性能を実現していることがわかる。これは単純な並列手法と比べて BER=10<sup>-5</sup> において約 0.3db の性能向上を実現している。また、図 6 に示すように繰り返し回数 20 回の際には 10<sup>-5</sup> において約 0.2db 改善することができた。図 7 に示すように 30 回程度の繰り返しを行うと BER 特性はほぼ限界値に達しており差はなくなっている。したがって、提案手法は BER の限界値を実現するために、20 回程度の繰り返し回数でほぼ収束しており、これは単純並列手法の 30 回に比べて繰り返し回数を大幅に削減している。

## 7. まとめ

本稿では、LDPC 符号について述べ、LDPC 符号の復法において、検査行列の行を入れ替えても復号する際何も影響が無い事を利用して IEEE802.16e 用の部分並列実装のための高性能化を目指すアルゴリズムを提案し検討した。シミュレーションにより繰り返し回数 10 回時に 0.3db ほど改善することができた。また、BER 限界値を達成するために、単純並列手法では 30 回程度かかる繰り返し回数を、20 回程度に削減し、処理の高速化を実現した。

今回は符号化率が一番低いつまり、計算コストがかかる R=1/2 について検討を行った。しかし、その他の符号化率の Hbm に対しても考える必要があり、2/3B-code では同様の手法がかなり有効であると思わ

れる。今後の課題としては、2/3A-code, 3/4A-code B-codeについても検討を行い、実装を行っていきたい。

### 文 献

- [1] 和田山 正, "低密度パリティ検査符号とその復号法", トリケップス, 2002.
- [2] 石川 達之, 清水 一範, 池永 剛, 後藤 敏, "Min-Sum アルゴリズムを用いた LDPC 復号器のメモリ削減手法", IEICE Technical Report CAS2005-22, VLD2005-33, SIP2005-46, pp.43-48 6月 2005
- [3] M. M. Mansour, and N. R. Shandhag; "Low-power VLSI decoder architectures for LDPC codes," Proceedings of the 2002 International Symposium on Low Power Electronics and Design, pp.284-289, 2002.
- [4] M. M. Mansour, and N. R. Shandhag; "High-throughput LDPC decoders," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume 11, pp.976-996, 2003.