Simple Iterative Decoding of Product Codes Based on Table-Aided Scheme

Takayuki SHIMIZU* Kenya HORAI* Hisato IWAI* and Hideichi SASAOKA*

(Received January 19, 2008)

This paper proposes a simple iterative decoding algorithm of product codes by using table-aided soft-decision decoding, in which the table of error patterns corresponding to their syndromes is used to limit the number of the candidate codewords. In the proposed scheme, syndromes of row-vector and column-vector are calculated from the hard-decision sequence of a received sequence, and error patterns are calculated by the table-aided decoding. The error patterns are used to change the soft-input for the decoder into more reliable values. The changed soft-input is used as the soft-input for the next decoder. Iterating this process, the proposed scheme improves the performance of bit error rate remarkably. The proposed scheme also reduces the computational complexity by using the table-aided soft-decision decoding and by stopping the iteration of the decoding in the case that all error bits are considered to be corrected. The numerical simulations are carried out over the additive white Gaussian noise channel to investigate the bit error rate performance and how many the iterations are required until the itarative decoding is stopped. The results show the effectiveness of the proposed decoding algorithm.

Key words ： block codes, product codes, soft-decision decoding, table-aided decoding, iterative decoding

キーワード ： ブロック符号，積符号，軟判定復号，テーブル参照軟判定復号，反復復号

積符号のテーブル参照による簡易反復復号法の検討

清水崇之・宝来剣文・岩井誠人・笹岡秀一

1. はじめに

情報化社会の発展に伴い，様々な情報が大量にやり取りされるようになった．しかし，多くの場合，情報伝達は雑音のある通信路を介するため，送信した情報と受信した情報がすべて等しくなるとは限らない．そのため情報を誤りなく伝達するには通信路上で発生した誤りを受信側で訂正する誤り訂正符号が必要不可欠となる．

1993 年に Berrou らによりターボ符号 [1] が発表されて以降，Shannon 限界に迫る新しい方式として軟入力軟出力反復復号法は一躍注目を浴びることとなった．ターボ符号の特徴は符号化よりもむしろ復号方法にあり，前回の復号で得られた外部情報を次の復号の事前情報として用い，反復して復号を行うことにより，復号の精度を高めていく手法である．このような復号法はターボ復号と呼ばれる．

当初のターボ符号では，要素符号に畳み込み符号を用いたものであったが，その後，要素符号にブロック

* Department of Electronics, Doshisha University, Kyoto, 610-0321, Japan
Telephone:+81-774-65-6267, Fax:+81-774-65-6801, E-mail:iwai@mail.doshisha.ac.jp

符号を用いてターボ復号を行うブロックターボ符号も提案されている[2, 3, 4]．ブロックターボ符号の特徴として，高符号化率の符号を使用できることや，符号の最小ハミング距離が十分大きいため，ターボ符号で発生するエラーフロア現象が起こらないということが挙げられる．また，ブロックターボ符号には積符号が使われることが多い．ブロックターボ符号の復号法としては，Hagenauer らによって提案された BCJR アルゴリズム[5] を用いた復号法[2] や，Pyndiah によって提案された Chase 復号[6] を用いた復号法[3]，Fossorier らによって提案された順序統計量復号[7] を用いた復号法[4] などがある．この中でも，Pyndiah による Chase 復号を用いた方式は，復号性能と計算量との間で良いトレードオフを持つ復号法である．

しかしながら，ブロックターボ符号に対する従来の復号法では，最大事後確率 (MAP: Maximum A posteriori Probability) 復号に基づいて外部情報を計算しているため，計算量が膨大となる．

これらの問題に対し，本稿では複雑な処理を必要としない反復復号法を提案する．この方式では各行各列の軟出力計算にテーブル参照軟判定復号法[8] を用いる．シンドロームとそのシンドロームに対応した複数の誤りパターンの対応表を用いて誤りビットを検出し，そのビット位置に対応する軟入力を書き換える．この処理を繰り返し行うことで，復号の精度を上げていく．提案方式の有効性を示すために，計算機シミュレーションにより復号性能を評価する．加法的白色ガウス雑音（AWGN: Additive White Gaussian Noise）通信路を仮定してビット誤り率（BER: Bit Error Rate）特性を求め，簡易な処理で Chase 復号を用いた従来方式と同等の特性が得られることを示す．また，打ち切り条件を用いて反復処理を終了することにより，計算量の削減が可能であることを示す．

2. 積符号とその復号法

2.1 積符号の構成

積符号は複数の符号を組み合わせることによって，最小ハミング距離の大きな符号を簡単に生成することができる．本稿では，2 つの 2 元線形組織ブロック符号 $\mathcal{C}_1$ (行符号), $\mathcal{C}_2$ (列符号) を要素符号として構成される

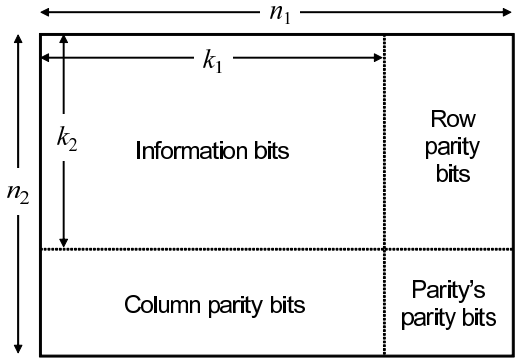


Fig. 1. Configuration of product codes.

Fig. 1 のような積符号を考える．それぞれの要素符号のパラメータを (n_1, k_1, d_1)，(n_2, k_2, d_2) とする．ただし，(n_i, k_i, d_i) $(i = 1, 2)$ は符号長，情報長，最小ハミング距離を表す．このとき，積符号 $\mathcal{P} = \mathcal{C}_1 \otimes \mathcal{C}_2$ の符号化は以下の手順で行われる．まず，情報ビットを k_2 行 k_1 列の行列の形式で配置する．次に，$1, \ldots, k_1$ 列目に対し，符号 $\mathcal{C}_2$ を用いて列方向に符号化する．最後に，$1, \ldots, n_2$ 行目に対し，符号 $\mathcal{C}_1$ を用いて行方向に符号化する．こうして生成された積符号 $\mathcal{P}$ のパラメータ (N, K, D) は，$N = n_1 n_2$, $K = k_1 k_2$, $D = d_1 d_2$ となる．また符号化率 R は，それぞれの要素符号の符号化率 $r_1 = k_1/n_1$, $r_2 = k_2/n_2$ を用いて $R = r_1 r_2$ と表される．なお，要素符号 $\mathcal{C}_1, \mathcal{C}_2$ の線形性から，符号化の順番を $\mathcal{C}_1, \mathcal{C}_2$ に変えても得られる符号語は同一である．したがって，積符号 $\mathcal{P}$ のすべての行は符号 $\mathcal{C}_1$ の符号語であり，すべての列は符号 $\mathcal{C}_2$ の符号語となっている．

2.2 積符号の反復復号法

積符号は従来から強力な復号性能を持つことが知られていたが，現実的な計算量で，その性能を引き出す復号法はなかった．しかし，反復復号であるターボ復号が提案されて以降，積符号に対してターボ復号を適用することにより，Shannon 限界に迫る復号性能が，現実的な計算量で得られること可能となった．以下では，そのターボ復号について説明する．

送信される符号語を $\boldsymbol{c} = (c_1, \ldots, c_N)$ とし，二位相偏移（BPSK: Binary Phase Shift Keying）変調により，符号語 $\boldsymbol{c}$ の各ビットが $\{0, 1\}$ から $\{-1, +1\}$ に

マッピングされ，送信系列 $\boldsymbol{x}$ として，通信路に送られるものとする．また，受信系列を $\boldsymbol{r} = (r_1, \ldots, r_N)$ とする．なお，この後の計算表記の簡単化のため単一インデックスを用いて表記している．このとき，すべての送信系列候補 $\boldsymbol{x}_i$ の中でブロック単位の事後確率 $P(\boldsymbol{x}_i|\boldsymbol{r})$ を最小とする送信系列候補を送信系列と推定する復号規範はブロック単位の MAP 復号と呼ばれる．ブロック単位の MAP 復号はブロック誤り率を最小化するという意味で最適な復号法である．しかし，ビット（シンボル）誤り率の最小化という観点からはブロック単位の MAP 復号は最適ではなく，シンボル単位の MAP 復号が最適となる．シンボル単位の MAP 復号は受信系列 $\boldsymbol{r}$ が与えられた下で，各送信候補ビット x_j $(1, \ldots, N)$ に関するシンボル単位の事後確率 $P(x_j|\boldsymbol{r})$ を最大とする $\hat{x}_j = \pm 1$ のいずれかを復号結果として出力する．すなわち，シンボル単位の事後確率 $P(x_j|\boldsymbol{r})$ の対数比である対数尤度比（LLR: Log-Likelihood Ratio）

$$L(x_j) = \log \frac{P(x_j = +1|\boldsymbol{r})}{P(x_j = -1|\boldsymbol{r})} \tag{1}$$

を何らかのアルゴリズムに基づいて計算し，$L(x_j) > 0$ ならば $\hat{x}_j = +1$，$L(x_j) < 0$ ならば $\hat{x}_j = -1$ と判定する（$L(x_j) = 0$ の場合は任意）．

線形符号にシンボル単位の MAP 復号をする場合，計算量は符号長に対して指数的に増大する．そのため，積符号のような符号長の長い符号に対してシンボル単位の MAP 復号を行うことは現実的に不可能である．それに対して，ターボ復号では，符号長が長く計算量が大きい符号を，計算量の比較的少ない複数の要素（積符号では行符号・列符号に対応する計算量の少ない復号器）に分解し，計算量の少ない復号器間の相互作用により復号性能を逐次的に向上させる手法がターボ復号である．

組織符号においては，事後 LLR $L(x_j)$ は以下の式で表現される [1]．

$$L(x_j) = L_{\mathrm{c}} r_j + L_{\mathrm{a}}(x_j) + L_{\mathrm{e}}(x_j) \tag{2}$$

ここで，各項は以下の通りである．

$L_{\mathrm{c}} r_j$：受信系列の各シンボル r_j から得られる通信路情報．AWGN 通信路では，E_{s}/N_0 をシンボル単位の信号対雑音電力比として以下の式で表わされる．

$$L_{\mathrm{c}} = 4E_{\mathrm{s}}/N_0 \tag{3}$$

$L_{\mathrm{a}}(x_j)$：x_j に関する事前情報．事前確率 $P(x_j)$ の対数尤度比で以下の式で表わされる．なお，初回の復号で事前確率が既知でない場合は $L_{\mathrm{a}}(x_j) = 0$ とする．

$$L_{\mathrm{a}}(x_j) = \log \frac{P(x_j = +1)}{P(x_j = -1)} \tag{4}$$

$L_{\mathrm{e}}(x_j)$：符号のパリティ検査ビットに関する拘束条件により x_j に関して得られる外部情報．

なお，$L_{\mathrm{c}} r_j + L_{\mathrm{a}}(x_j)$ は復号器への入力であるので軟入力，$L(x_j)$ は 復号器からの出力であるので軟出力と呼ばれる．

積符号のターボ復号器を Fig. 2 に示す．積符号では，行符号の復号器（行復号器）で計算された外部情報 $L_{\mathrm{e}}^{(\mathrm{row})}(x_j)$ を列符号の復号器（列復号器）に入力し，列復号器でのシンボル単位の MAP 復号における事前情報 $L_{\mathrm{a}}^{(\mathrm{col})}(x_j)$ として利用する．次に，列復号器で計算された外部情報 $L_{e}^{(\mathrm{col})}(x_j)$ を行復号器の事前情報 $L_{\mathrm{a}}^{(\mathrm{row})}(x_j)$ として再びシンボル単位の MAP 復号を行う．適当な反復の後，列復号器の事後 LLR $L^{(\mathrm{col})}(x_j)$ の符号により $\hat{x}_j$ を判定する．このように，計算量の小さい復号器間で外部情報を繰り返し受け渡しすることで，復号の精度を上げていくのがターボ復号の基本概念である．

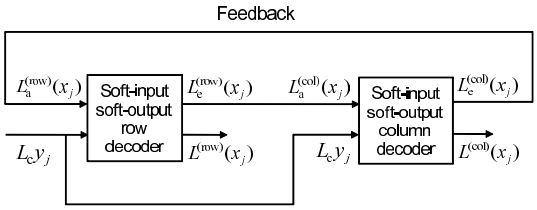


Fig. 2. Turbo decoder of product codes.

しかしながら，このターボ復号では各要素符号に対してシンボル単位の MAP 復号を行っているため，事後 LLR を計算するためには依然，膨大な計算量が必要となる．そこで，Chase 復号 [6] や順序統計量復号 [7] などの軟入力硬出力復号器を用いて近似的に MAP

復号を行い，計算量を削減する方式が提案されている[3, 4]．軟入力硬出力復号器を用いてターボ復号を行う方式では，軟入力硬出力復号を繰り返し行うことにより，事後 LLR の近似値を計算し，計算量を大幅に削減している．

3. テーブル参照軟判定復号法

3.1 最尤復号法

最尤（Maximum likelihood）復号法は，各符号語が等しい確率で送信される場合，ブロック誤り率を最小にする復号法であり，ブロック単位の MAP 復号と等価である．最尤復号法では，受信系列 $\boldsymbol{r}$ に対して，尤度関数と呼ばれる条件付き確率 $P(\boldsymbol{r}|\boldsymbol{x}_i)$ を最大とする符号語の送信系列 $\hat{\boldsymbol{x}}$ が送信されたと推定する．AWGN 通信路では，受信系列 $\boldsymbol{r}$ に対して，すべての符号語の送信系列 $\boldsymbol{x}_i$ とのユークリッド距離

$$d_{\mathrm{E}}(\boldsymbol{x}_i, \boldsymbol{r}) = \sqrt{||\boldsymbol{x}_i - \boldsymbol{r}||^2} \tag{5}$$

を求め，ユークリッド距離 $d_{\mathrm{E}}(\boldsymbol{x}_i, \boldsymbol{r})$ を最小にする送信系列 $\hat{\boldsymbol{x}}$ が送信された送信系列であると推定する．すなわち，AWGN 通信路では尤度関数 $P(\boldsymbol{r}|\boldsymbol{x}_i)$ 最大とユークリッド距離 $d_{\mathrm{E}}(\boldsymbol{x}_i, \boldsymbol{r})$ 最小は等価である．

3.2 テーブル参照軟判定復号法の原理

最尤復号はブロック誤り率を最小にできる復号法であるが，すべての符号語に対してユークリッド距離の比較を行うと，計算量が膨大になるという問題がある．この計算量を削減するために，テーブルを参照して復号を行う方式が提案されている[8]．

この方式で扱うテーブルとは，ハミング重みが $1, \ldots, d_{\mathrm{D}}$ の全誤りパターンと，そのシンドロームとの対応表である．このとき，d_{D} をテーブルの復号半径といい，テーブル容量と復号の計算量を決める重要なパラメータとなる．復号半径 d_{D} のテーブルを用いた場合，硬判定受信系列からハミング距離 d_{D} 以内の範囲が復号探索領域となる．通常，テーブルは，あらかじめ作成してメモリなどに保存しておき，復号を行うときに用いる．

テーブル参照軟判定復号法では，送信された符号語からテーブルの復号半径内に受信された硬判定系列の誤りを訂正する．テーブル参照軟判定復号法の手順

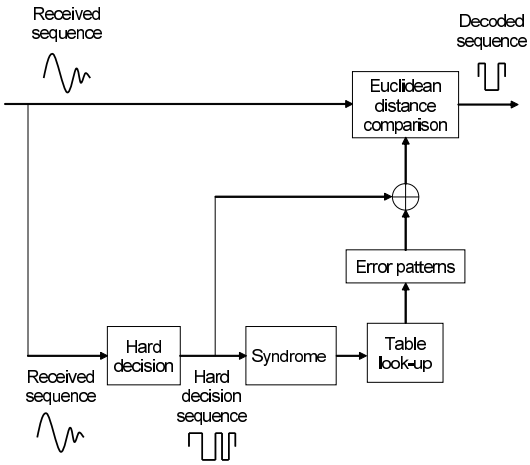


Fig. 3. Procedure of table-aided soft-decision decoding.

を Fig. 3 に示す．まず，復号の前にテーブルをあらかじめ作成しておく．なお，一度テーブルを作れば，あとは同じテーブルを繰り返し使用することができる．ここで，線形ブロック符号 $\mathcal{C} = (n, k, d_{\min})$ を用いるとする．ただし，$n, k, d_{\min}$ は，それぞれ符号長，情報長，最小ハミング距離である．送信される符号語を $\boldsymbol{c} = (c_1, \ldots, c_n)$ とし，BPSK 変調により，符号語 $\boldsymbol{c}$ の各ビットが $\{0, 1\}$ から $\{-1, +1\}$ にマッピングされ，送信系列 $\boldsymbol{x} = (x_1, \ldots, x_n)$ として，AWGN 通信路に送られるものとする．また，雑音系列を $\boldsymbol{b} = (b_1, \ldots, b_n)$ とすると，受信系列 $\boldsymbol{r}$ は $\boldsymbol{r} = (r_1, \ldots, r_n) = \boldsymbol{x} + \boldsymbol{b}$ と表される．硬判定受信系列を $\boldsymbol{y} = (y_1, \ldots, y_n)$ とし，$\boldsymbol{y}$ から得られるシンドロームを $\boldsymbol{s} = (s_1, \ldots, s_{n-k})$ とする．このとき，テーブルを参照することにより，シンドローム $\boldsymbol{s}$ に対応するハミング重みが $1, \ldots, d_{\mathrm{D}}$ の全誤りパターンが得られる．次に，これらの誤りパターンを硬判定受信系列 $\boldsymbol{y}$ に付加することで，復号半径 d_{D} 内の候補符号語をすべて得ることができる．最後に，これらの候補符号語と受信系列 $\boldsymbol{y}$ とのユークリッド距離を計算し，最もユークリッド距離の小さい候補符号語を復号語と推定する．ここで，符号語間の最小ハミング距離 $d_{\min}$ に対して，テーブルの復号半径を $d_{\mathrm{D}} = d_{\min} - 1$ とすれば，テーブル参照軟判定復号法は最尤復号法とほぼ同じ復号性能を持つ．

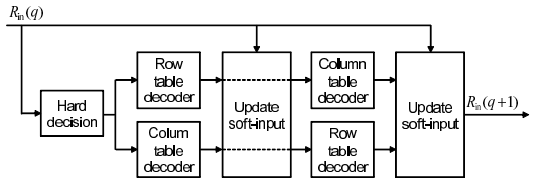


Fig. 4. Configuration of proposed decoding system.

4. 提案方式

4.1 提案方式の原理

本稿では，積符号に対してテーブル参照軟判定復号法を用いて反復復号を行う方式を提案する．テーブルを参照して得られた複数の誤りパターンから候補符号語を生成し，得られた候補符号語に対してユークリッド距離比較を行い，最も尤度の高い候補符号語または誤りパターンを求める．この処理を積符号の符号語のすべての行ベクトルと列ベクトルに対して行うことで，n_2 行 n_1 列の誤りパターンを 2 通り得ることができる．ここで，行ベクトルに対する処理で得られた誤りパターンを $\boldsymbol{E}_{\rm row}$，列ベクトルに対する処理で得られた誤りパターンを $\boldsymbol{E}_{\rm col}$ とする．

テーブル参照軟判定復号法により得られる誤りパターンは硬判定系列である．反復復号では復号器からの出力は軟判定系列であることが望ましいため，得られた誤りパターンをもとに軟出力を計算する工夫が必要となる．そこで本稿では，誤りパターンの誤りビット位置に対応する軟入力に対し，より確からしい値に書き換えていく方式を提案する．

提案方式の復号器の構成を Fig. 4 に示す．まず，入力された軟入力 $\boldsymbol{R}_{\rm in}(q)$ を硬判定し，硬判定系列 $\boldsymbol{W}$ を得る．そして，$\boldsymbol{W}$ の行ベクトル，列ベクトルに対してテーブル参照軟判定復号を適用し，誤りパターン $\boldsymbol{E}_{\rm row}$，$\boldsymbol{E}_{\rm col}$ を求める．誤りパターンの情報をもとに軟入力 $\boldsymbol{R}_{\rm in}(q)$ を書き換え，さらに，$\boldsymbol{W}\oplus\boldsymbol{E}_{\rm row}$ の列ベクトル，$\boldsymbol{W}\oplus\boldsymbol{E}_{\rm col}$ の行ベクトルに対してテーブル参照軟判定復号を適用し，それぞれ誤りパターン $\boldsymbol{E}'_{\rm col}$，$\boldsymbol{E}'_{\rm row}$ を求める．誤りパターンの情報をもとに，さらに軟入力 $\boldsymbol{R}_{\rm in}(q)$ を書き換え，軟出力 $\boldsymbol{R}_{\rm out}(q)=\boldsymbol{R}_{\rm in}(q+1)$ を得る．軟出力 $\boldsymbol{R}_{\rm out}(q)$ を次の復号の軟入力 $\boldsymbol{R}_{\rm in}(q+1)$ とし，同様の操作を繰り返す．適当な繰り返しのあと，最終的に軟出力を硬判定し，その硬判定系列を復号結果とする．

4.2 軟入力更新手法の検討

本方式では，テーブル参照軟判定復号により得られた誤りパターンをもとに軟入力を更新する．さらに，利用する誤りパターンのビット位置をその信頼度に応じて以下の 2 種類に分類する．

$$\boldsymbol{E}_{\rm and}={\rm AND}(\boldsymbol{E}_{\rm row},\boldsymbol{E}_{\rm col}) \tag{6}$$

$$\boldsymbol{E}_{\rm xor}={\rm XOR}(\boldsymbol{E}_{\rm row},\boldsymbol{E}_{\rm col}) \tag{7}$$

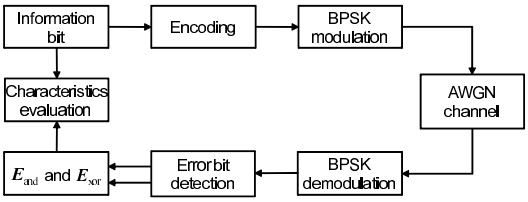
$\boldsymbol{E}_{\rm and}$ は，行ベクトル，列ベクトルそれぞれに対するテーブル参照軟判定復号で $\boldsymbol{E}_{\rm row}$ と $\boldsymbol{E}_{\rm col}$ のどちらにおいても検出された誤りビット位置を表し，$\boldsymbol{E}_{\rm xor}$ は，$\boldsymbol{E}_{\rm row}$ と $\boldsymbol{E}_{\rm col}$ のどちらか一方で検出された誤りビット位置を表している．この 2 種類の誤りビット位置に対し，それぞれ異なる方法で軟入力を更新する．

軟入力更新の手法を決定するために，$\boldsymbol{E}_{\rm and}$ および $\boldsymbol{E}_{\rm xor}$ に関するデータを取得し，それぞれで検出された誤りビット位置の特性を調べる．データ取得のためのシステムを Fig. 5 に示す．符号語は BPSK 変調されているものとし，通信路には AWGN 通信路を仮定した．受信系列に対して Fig. 5(b) のように行方向と列方向にテーブル参照軟判定復号法を行い，誤りパターン $\boldsymbol{E}_{\rm row}$ と $\boldsymbol{E}_{\rm col}$ を求める．さらに式 (6), (7) から $\boldsymbol{E}_{\rm and}$ と $\boldsymbol{E}_{\rm xor}$ を求め，特性評価を行う．

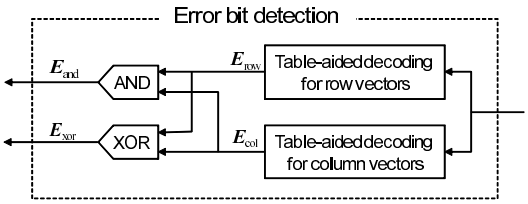
4.2.1 $\boldsymbol{E}_{\rm and}$ の特性

$\boldsymbol{E}_{\rm row}$ と $\boldsymbol{E}_{\rm col}$ の共通部分 $\boldsymbol{E}_{\rm and}$ と送信系列とを比較し，$\boldsymbol{E}_{\rm and}$ で検出された誤りビット位置に実際にどの程度誤りが生じているかを調べる．比較対象として，$\boldsymbol{E}_{\rm row}$ の誤りビット位置，$\boldsymbol{E}_{\rm col}$ の誤りビット位置についても調べる．試行回数を 10,000 回とし，その平均を求める．

BCH$(32,16,8)^2$ の積符号で取得した $\boldsymbol{E}_{\rm and}$ の特性を Fig. 6 に示す．Fig. 6 は，1 ビットあたりの電力対雑音電力密度比 $E_{\rm b}/N_0$ に対する検出された個数と実際に誤りが生じていた個数の割合を表している．Fig. 6 より， $\boldsymbol{E}_{\rm row}$ と $\boldsymbol{E}_{\rm col}$ のそれぞれ単独で検出された誤りビット位置と比較して，$\boldsymbol{E}_{\rm and}$ で検出された誤りビット位置は実際に誤りが生じている確率が高く，$E_{\rm b}/N_0=2$ dB 以上では 85%以上となっている．つ



(a) Simulation system for the evaluation of error bit detection performance



(b) Configuration of error bit detection

Fig. 5. Simulation system for the evaluation of error bit detection performance and configuration of error bit detection.

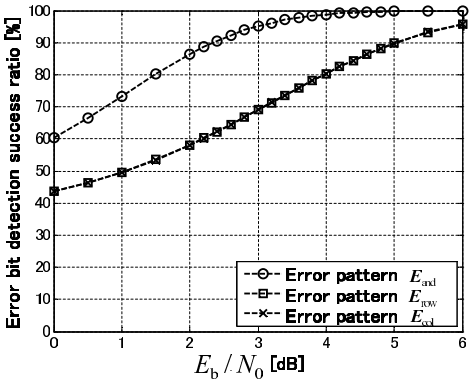


Fig. 6. Performance of error bit detection success ratio in various error patterns, $\boldsymbol{E}_{\rm and}$, $\boldsymbol{E}_{\rm row}$ and $\boldsymbol{E}_{\rm col}$.

まり，$\boldsymbol{E}_{\rm and}$ は正しく検出されている確率が高い．このことより，$\boldsymbol{E}_{\rm row}$ と $\boldsymbol{E}_{\rm col}$ で共通して検出された誤りビット位置 $\boldsymbol{E}_{\rm and}$ に対応する軟入力の硬判定値が変化するように，対応する軟入力の値を書き換えることで，高い復号性能が得られると考えられる．

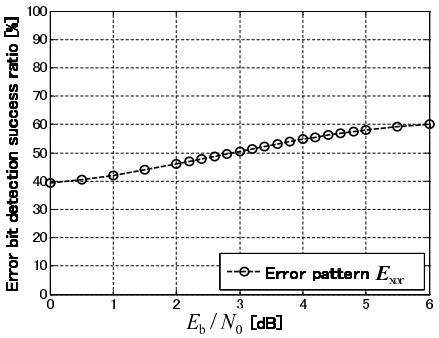
4.2.2 $\boldsymbol{E}_{\rm xor}$ の特性

$\boldsymbol{E}_{\rm and}$ の場合と同様に，$\boldsymbol{E}_{\rm row}$ と $\boldsymbol{E}_{\rm col}$ のどちらかのみが検出した誤りビット位置 $\boldsymbol{E}_{\rm xor}$ について，実際にどの程度誤りが生じているかを調べる．また，$\boldsymbol{E}_{\rm xor}$ で検出された誤りビット位置を，実際に誤りが生じている箇所（訂正すべき箇所）とそれ以外の箇所（訂正しなくてよい箇所）に分類し，それぞれに対応する受信系列の信頼度をそれぞれ求め，その特性を調べる．ただし，受信系列の信頼度は，受信系列の絶対値とする．試行回数を 10,000 回とし，その平均を求める．

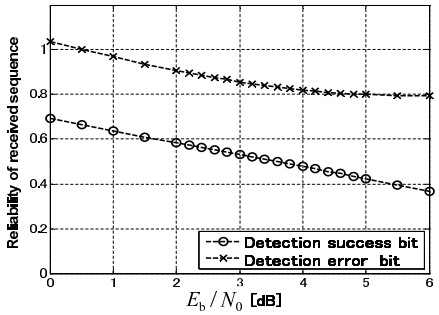
BCH$(32, 16, 8)^2$ の積符号で取得した $\boldsymbol{E}_{\rm xor}$ の特性を Fig. 7 に示す．Fig. 7(a) は検出された個数に対する実際に誤りが生じていた個数の割合を表しており，Fig. 7(b) は訂正すべき箇所と訂正しなくてよい箇所の受信系列の信頼度の平均値を表している．Fig. 7(a) より，$\boldsymbol{E}_{\rm xor}$ で検出された誤りビット位置には実際に誤りの生じている箇所が 40%以上含まれていることがわかる．また Fig. 7(b) より，訂正すべき箇所に対応する受信系列の信頼度は，訂正しなくてよい箇所の信頼度と比較して，小さい値となっている．したがって，$\boldsymbol{E}_{\rm xor}$ で検出された誤りビット位置の中で，受信系列の信頼度が低い箇所では，実際に誤りが生じている確率が高いと考えられる．実際にデータを調べてみると，受信系列の信頼度の低い約 1/3 に関しては，実際に誤りが生じている確率が 50%を超えていた．そこで，$\boldsymbol{E}_{\rm xor}$ で検出された誤りビット位置の中で，軟入力の信頼度が低い箇所について，その軟入力の信頼度がさらに低くなるように書き換えることで，次の復号器における復号精度を上げることができると考えられる．

4.3 提案方式の復号手順

誤りビット位置の特性評価をもとに決定した提案方式の復号手順を示す．提案方式では，大きく分けて 2 つの反復復号からなる．最初の反復復号では，誤りビット検出精度の高い $\boldsymbol{E}_{\rm and}$ のみを用いて軟入力を書き換えながら反復復号を行う．最初の反復復号で訂正できなかった（行方向と列方向のシンドロームが $\boldsymbol{0}$ にならなかった）場合は，2 つ目の反復復号として，$\boldsymbol{E}_{\rm and}$ と $\boldsymbol{E}_{\rm xor}$ を用いて反復復号を行う．



(a) Performance of error bit detection success ratio in error pattern $\boldsymbol{E}_{\mathrm{xor}}$



(b) Characteristics of reliability of received sequence corresponding to error bit detected in error pattern $\boldsymbol{E}_{\mathrm{xor}}$

Fig. 7. Performance of error bit detection success ratio and characteristics of reliability of received sequence corresponding to error bit detected in error pattern $\boldsymbol{E}_{\mathrm{xor}}$.

4.3.1 $\boldsymbol{E}_{\mathrm{and}}$ のみを用いた反復復号

まず，1 つ目の $\boldsymbol{E}_{\mathrm{and}}$ のみを用いた反復復号について述べる．

1. 復号器への軟入力を $\boldsymbol{R}_{\mathrm{in}}(q)$，軟入力の硬判定系列を $\boldsymbol{Y}_{\mathrm{in}}(q)$ とする．ただし，q は反復回数を表し，1 回目の反復における軟入力 $\boldsymbol{R}_{\mathrm{in}}(1)$ は受信系列 $\boldsymbol{R}$ とする．また，軟入力 $\boldsymbol{R}_{\mathrm{in}}(q)$ の i 行 j 列目を $r_{\mathrm{in}}(q)_{i,j}$ と表記する．

2. $\boldsymbol{Y}_{\mathrm{in}}(q)$ の行方向に対してテーブル参照軟判定復号を行い，誤りパターン $\boldsymbol{E}_{\mathrm{row}}(q)$ を求める．同様に $\boldsymbol{Y}_{\mathrm{in}}(q)$ の列方向に対してもテーブル参照軟判定復号を行い，誤りパターン $\boldsymbol{E}_{\mathrm{col}}(q)$ を求める．

3. $\boldsymbol{E}_{\mathrm{row}}(q)$ と $\boldsymbol{E}_{\mathrm{col}}(q)$ で共通して検出された誤りビット位置 $\boldsymbol{E}_{\mathrm{and}}(q) = \mathrm{AND}(\boldsymbol{E}_{\mathrm{row}}(q), \boldsymbol{E}_{\mathrm{col}}(q))$ を計算する．

4. $\boldsymbol{E}_{\mathrm{and}}(q)$ で検出された軟入力 $\boldsymbol{R}_{\mathrm{in}}(q)$ の箇所に対して，次のテーブル参照軟判定復号で訂正されやすくするため 1/4 倍する．これにより得られた新たな軟入力を $\boldsymbol{R}'_{\mathrm{in}}(q)$ とする．

5. 行方向にテーブル参照軟判定復号を行って得られた系列 $\boldsymbol{Y}_{\mathrm{in}}(q) \oplus \boldsymbol{E}_{\mathrm{row}}(q)$ に対して今度は列方向にテーブル参照軟判定復号を行い，誤りパターン $\boldsymbol{E}'_{\mathrm{col}}(q)$ を求める．同様に，列方向にテーブル参照軟判定復号を行って得られた系列 $\boldsymbol{Y}_{\mathrm{in}}(q) \oplus \boldsymbol{E}_{\mathrm{col}}(q)$ に対して行方向にテーブル参照軟判定復号を行い，誤りパターン $\boldsymbol{E}'_{\mathrm{row}}(q)$ を求める．ただし，ここでのテーブル参照軟判定復号では $\boldsymbol{R}'_{\mathrm{in}}(q)$ を基準としてユークリッド距離比較を行う．

6. $\boldsymbol{E}_{\mathrm{row}}(q) \oplus \boldsymbol{E}'_{\mathrm{col}}(q)$ と $\boldsymbol{E}_{\mathrm{col}}(q) \oplus \boldsymbol{E}'_{\mathrm{row}}(q)$ で共通して検出された箇所 $\boldsymbol{E}'_{\mathrm{and}}(q) = \mathrm{AND}(\boldsymbol{E}_{\mathrm{row}}(q) \oplus \boldsymbol{E}'_{\mathrm{col}}(q), \boldsymbol{E}_{\mathrm{col}}(q) \oplus \boldsymbol{E}'_{\mathrm{row}}(q))$ を計算する．

7. $\boldsymbol{E}'_{\mathrm{and}}(q)$ で検出された誤りビット位置について誤りレベル $\boldsymbol{E}_{\mathrm{level}}(q)$ を考える．Fig. 8 に誤りレベルの計算例を示す．まず，$\boldsymbol{E}_{\mathrm{level}}(q) = \boldsymbol{E}'_{\mathrm{and}}(q)$ として初期化する．次に，$\boldsymbol{E}_{\mathrm{row}}(q) \oplus \boldsymbol{E}'_{\mathrm{col}}(q)$ と $\boldsymbol{E}'_{\mathrm{and}}(q)$ の各行を比較し，完全に一致していれば，$\boldsymbol{E}'_{\mathrm{and}}(q)$ で検出された誤りビット位置に対応する $\boldsymbol{E}_{\mathrm{level}}(q)$ の箇所に 1 を足す．同様に，$\boldsymbol{E}_{\mathrm{row}}(q) \oplus \boldsymbol{E}'_{\mathrm{col}}(q)$ と $\boldsymbol{E}'_{\mathrm{and}}(q)$ の各列を比較し，完全に一致していれば，$\boldsymbol{E}'_{\mathrm{and}}(q)$ で検出された誤りビット位置に対応する $\boldsymbol{E}_{\mathrm{level}}(q)$ の箇所に 1 を足す．同様の処理を $\boldsymbol{E}_{\mathrm{col}}(q) \oplus \boldsymbol{E}'_{\mathrm{row}}(q)$ に対しても行う．最終的に得られた $\boldsymbol{E}_{\mathrm{level}}(q)$ では，値が大きいほど，誤っている可能性が高いと考えられる．なお，誤りレベル $\boldsymbol{E}_{\mathrm{level}}(q)$ の i 行 j 列目を $e_{\mathrm{level}}(q)_{i,j}$ と表記する．

8. $\boldsymbol{E}'_{\mathrm{and}}(q)$ で検出された誤りビット位置に対応する軟入力 $\boldsymbol{R}_{\mathrm{in}}(q)$ に対して，次式により軟出力

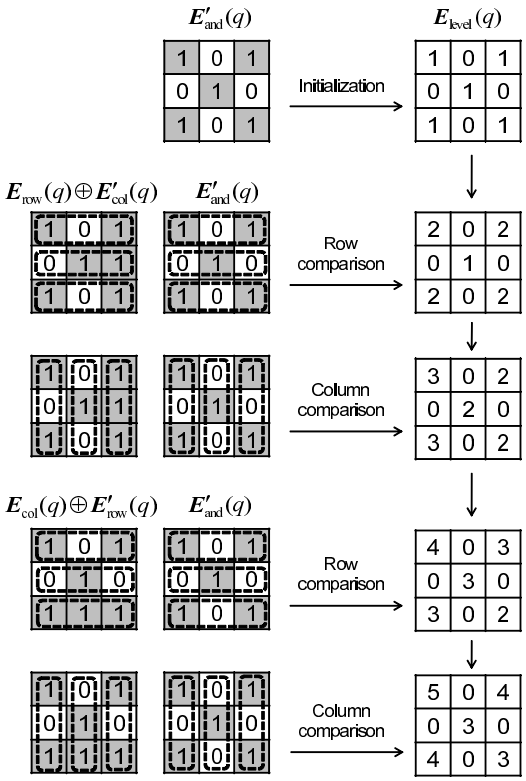


Fig. 8. Example of calculating error level.

$\boldsymbol{R}_{\rm out}(q)$ を計算する．ただし，軟出力 $\boldsymbol{R}_{\rm out}(q)$ の i 行 j 列目を $r_{\rm out}(q)_{i,j}$ と表記する．

$$r_{\rm out}(q)_{i,j} = r_{\rm in}(q)_{i,j} - {\rm sgn}(r_{\rm in}(q)_{i,j})\frac{r_{\rm av}(q)^2}{4}e_{\rm level}(q)_{i,j} \quad (8)$$

ただし，$r_{\rm av}(q)$ は，軟入力の信頼度の平均値であり，次式で計算される．

$$r_{\rm av}(q) = \frac{1}{N}\sum_{i,j}|r_{\rm in}(q)_{i,j}| \quad (9)$$

9. 更新されなかった軟入力の箇所は，書き換えずに軟出力とする．

10. 得られた軟出力 $\boldsymbol{R}_{\rm out}(q)$ を次回の復号器への軟入力 $\boldsymbol{R}_{\rm in}(q+1)$ として，1.〜9. の処理を繰り返す．

11. 適当な反復の後，軟出力 $\boldsymbol{R}_{\rm out}(q)$ の硬判定値を復号結果とする．

4.3.2 $\boldsymbol{E}_{\rm and}$ と $\boldsymbol{E}_{\rm xor}$ を用いた反復復号

$\boldsymbol{E}_{\rm and}$ のみを用いた反復復号で訂正できなかった場合，2 つ目の反復復号として $\boldsymbol{E}_{\rm and}$ と $\boldsymbol{E}_{\rm xor}$ を用いて

1. 4.3.1 の $\boldsymbol{E}_{\rm and}$ のみを用いた反復復号の 1.〜8. の処理を行う．

2. $\boldsymbol{E}_{\rm row}(q) \oplus \boldsymbol{E}'_{\rm col}(q)$ と $\boldsymbol{E}_{\rm col}(q) \oplus \boldsymbol{E}'_{\rm row}(q)$ のどちらか一方で検出された箇所 $\boldsymbol{E}'_{\rm xor}(q) = {\rm XOR}(\boldsymbol{E}_{\rm row}(q) \oplus \boldsymbol{E}'_{\rm col}(q), \boldsymbol{E}_{\rm col}(q) \oplus \boldsymbol{E}'_{\rm row}(q))$ を計算する．ただし，$\boldsymbol{E}'_{\rm xor}(q)$ の i 行 j 列目を $e'_{\rm xor}(q)_{i,j}$ と表記する．

3. $\boldsymbol{E}'_{\rm xor}(q)$ で検出された誤りビット位置に対応する軟入力の要素の集合 $\{r_{\rm in}(q)_{i,j}|e'_{\rm xor}(q)_{i,j} = 1\}$ において，Fig. 7(b) より軟入力の信頼度 $|r_{\rm in}(q)_{i,j}|$ が低い箇所ほど実際に誤りが生じている可能性が高い．そこで，$\{r_{\rm in}(q)_{i,j}|e'_{\rm xor}(q)_{i,j} = 1\}$ の中で信頼度 $|r_{\rm in}(q)_{i,j}|$ の低い 1/3 箇所に対して，軟入力の符号が反転するように軟入力 $\boldsymbol{R}_{\rm in}(q)$ を次式のように書き換え，軟出力 $\boldsymbol{R}_{\rm out}(q)$ を計算する．

$$r_{\rm out}(q)_{i,j} = r_{\rm in}(q)_{i,j} - \alpha(q)r_{\rm in}(q)_{i,j} \quad (10)$$

ただし，$\alpha(q)$ は反復ごとに異なる値を持つ重み付けのための係数であり，次式で与えられる．

$$\begin{aligned}\boldsymbol{\alpha} &= (\alpha(1), \alpha(2), \ldots, \alpha(q), \ldots) \\ &= (1.05, 1.10, 1.15, 1.20, 1.20, \ldots) \end{aligned} \quad (11)$$

反復復号が進むにつれて，誤りビット位置検出の精度が高くなると考えられるので，係数も大きくなるように設定されている．なお，$\boldsymbol{\alpha}$ の値は試行錯誤的に求められた値である．

4. $\{r_{\rm in}(q)_{i,j}|e'_{\rm xor}(q)_{i,j} = 1\}$ の中で，3. で軟入力を書き換えた箇所は除いて，信頼度 $|r_{\rm in}(q)_{i,j}|$ が低い 1/3 箇所に対して，軟入力の信頼度が 0 に近づくように軟入力 $\boldsymbol{R}_{\rm in}(q)$ を次式により書き換え，軟出力 $\boldsymbol{R}_{\rm out}(q)$ を計算する．

$$r_{\rm out}(q)_{i,j} = r_{\rm in}(q)_{i,j} - 0.8r_{\rm in}(q)_{i,j} \quad (12)$$

なお，係数の 0.8 は試行錯誤的に求められた値である．

5. 更新されなかった軟入力の箇所は，書き換えずに軟出力とする．

6. 得られた軟出力 $\boldsymbol{R}_{\rm out}(q)$ を次回の復号器への軟入力 $\boldsymbol{R}_{\rm in}(q+1)$ として，1.〜5. の処理を繰り返す．

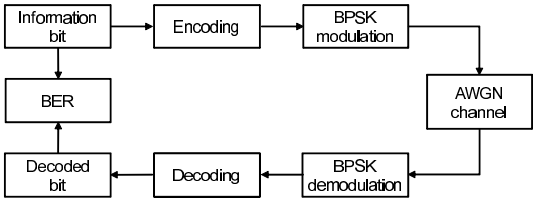


Fig. 9. Configuration of simulation system.

7. 適当な反復の後，軟出力 $\boldsymbol{R}_{\mathrm{out}}(q)$ の硬判定値を復号結果とする．

4.4 提案方式の計算量削減の効果

この提案方式では，従来のターボ復号に比べて，復号処理に複雑な計算を要しない．従来のターボ復号では，MAP 復号に基づいているため，すべてのビットに対して軟出力を計算しなければならないが，本稿の提案方式では，テーブル参照軟判定復号で検出された箇所のみに対して軟出力を計算するため，計算量が大幅に削減できる．また，行方向と列方向のシンドロームが $\mathbf{0}$ の場合は誤りがないと判断し，反復復号を終了することができる．

さらに，4.3.1, 4.3.2 での処理において，検出した誤りパターン $\boldsymbol{E}_{\mathrm{row}}$, $\boldsymbol{E}_{\mathrm{col}}$ が完全に一致した場合は，符号語に訂正されたと判断し，復号処理を終了する．この終了条件により，より計算量を削減することができる．

5. 復号特性シミュレーション

5.1 シミュレーションシステム

提案方式の軟入力軟出力反復復号法の復号性能を計算機シミュレーションにより検討する．シミュレーションシステムを Fig. 9 に，シミュレーション諸元を Table 1 に示す．まず情報データを要素符号が同一である積符号を用いて符号化し，符号語を生成する．次に符号語を BPSK 変調し，送信系列を作成する．通信路は AWGN 通信路を仮定し，送信系列に通信路からの雑音を付加して受信系列とする．その後，提案方式の反復復号を行うことで復号語を得る．復号語と送信した符号語を比較し，ビット誤り率を算出する．Table 2 に特性評価を行った積符号のパラメータを示す．ただし，$(n, k, d_{\min})$ は，それぞれ積符号の要素符号の符号長，情報長，最小ハミング距離であり，d_{D} はテーブルの復号半径である．

Table 1. Parameters of simulation system.

Channel	AWGN
Modulation	BPSK
Number of iterations	Up to 10 iterations

Table 2. Parameters of product codes.

$(n, k, d_{\min})^2$ product code	Code rate	d_{D}
$\mathrm{BCH}(32, 21, 6)^2$	0.431	4
$\mathrm{BCH}(32, 16, 8)^2$	0.250	5
$\mathrm{BCH}(64, 51, 6)^2$	0.635	4

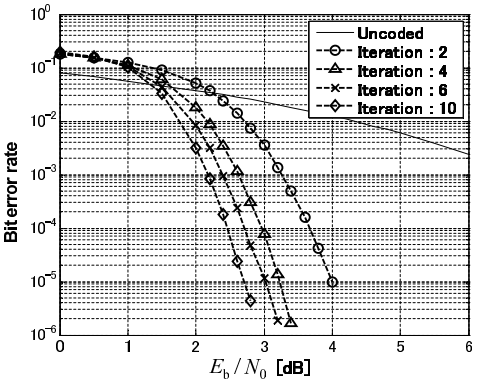


Fig. 10. BER performance of $\mathrm{BCH}(32, 21, 6)^2$ product code.

5.2 ビット誤り率特性

AWGN 通信路を仮定し，反復回数をパラメータとしてビット誤り率を求める．反復回数は最大 10 回とする．$\mathrm{BCH}(32, 21, 6)^2$ の BER 特性を Fig. 10 に，$\mathrm{BCH}(32, 16, 8)^2$ の場合を Fig. 11 に，$\mathrm{BCH}(64, 51, 6)^2$ の場合を Fig. 12 に示す．Fig. 10 より，反復回数 10 回で BER が 10^{-5} を達成する E_{b}/N_0 が $\mathrm{BCH}(32, 21, 6)^2$ では 2.7 dB となっている．同様に $\mathrm{BCH}(32, 16, 8)^2$ では 2.5 dB，$\mathrm{BCH}(64, 51, 6)^2$ では 2.8 dB となっている．Chase 復号を用いた従来の反復復号法 [3] と比較すると，$\mathrm{BCH}(64, 51, 6)^2$ ではほぼ同じ性能を示している．

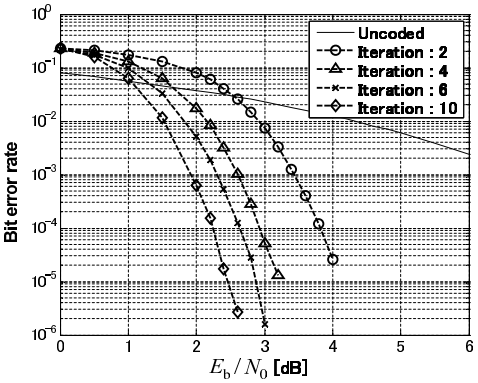


Fig. 11. BER performance of $\mathrm{BCH}(32,16,8)^2$ product code.

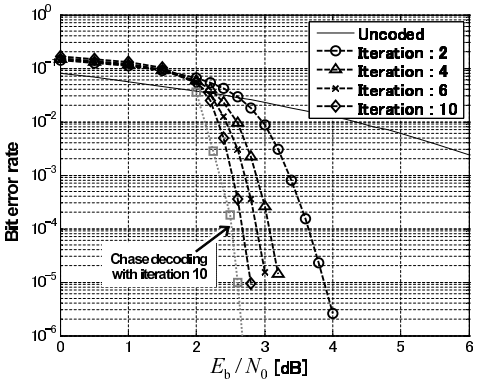


Fig. 12. BER performance of $\mathrm{BCH}(64,51,6)^2$ product code.

5.3 反復回数特性

符号語に訂正されたときの反復回数の特性を求める．前述したように，本方式では符号語に訂正されるとそれを検知し，復号を終了することができる．このときの反復回数を求め，その出現確率を調べる．反復回数は最大 10 回とし，10 回で訂正できない場合は訂正不可とする．$\mathrm{BCH}(32,21,6)^2$ の反復回数の特性を Fig. 13 に，$\mathrm{BCH}(32,16,8)^2$ の場合を Fig. 14 に，$\mathrm{BCH}(64,51,6)^2$ の場合を Fig. 15 に示す．Fig. 13 より，E_b/N_0 が大きくなるにつれて少ない反復回数で符号語に訂正される割合が増加している．$\mathrm{BCH}(32,21,6)^2$，$\mathrm{BCH}(64,51,6)^2$ についても同様の特性を示した．以上のことから，本方式の計算量削減の効果を確認することができる．

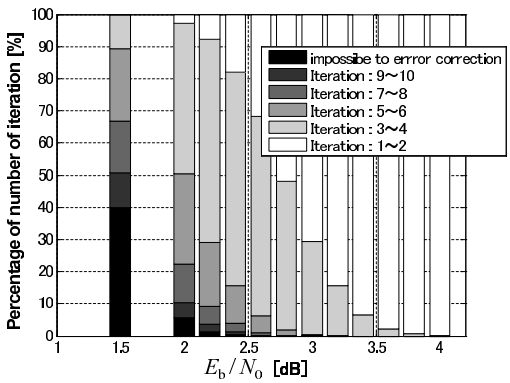


Fig. 13. Performance of the number of iterations with $\mathrm{BCH}(32,21,6)^2$ product code.

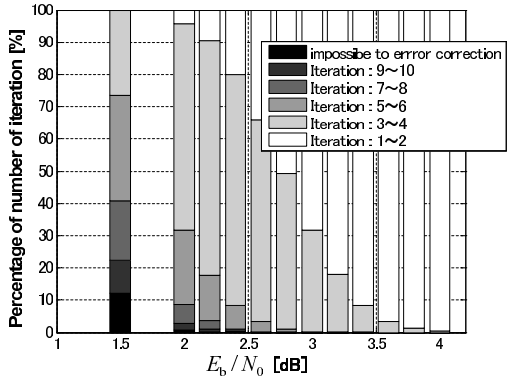


Fig. 14. Performance of the number of iterations with $\mathrm{BCH}(32,16,8)^2$ product code.

6. まとめ

本稿では，ブロック符号の積符号に対する復号アルゴリズムとして，テーブル参照軟判定復号法を用いた軟入力軟出力反復復号法を提案した．シンドロームとそのシンドロームに対応する誤りパターンの対応表を用いて誤りビット位置の候補を求め，その位置に対応する軟入力値を書き変えながら復号処理を繰り返す．また本方式では，複雑な処理を必要とせず，また符号語に訂正されたことを検知して復号処理を終了することが可能である．本方式の有効性を確認するために，計算機シミュレーションにより AWGN 通信路におけるビット誤り率特性を求めた．その結果，$\mathrm{BCH}(64,51,6)^2$ では，Chase 復号を用いた反復復号法とほぼ同等の性能

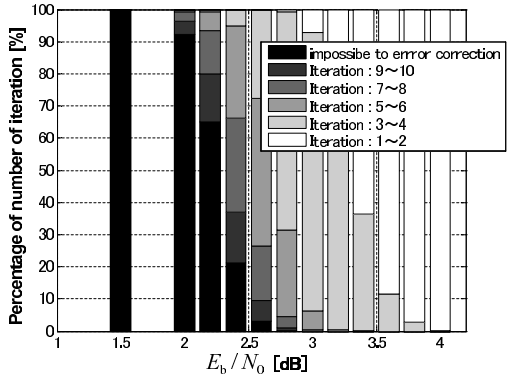


Fig. 15. Performance of the number of iterations with BCH$(64, 51, 6)^2$ product code.

を得ることができた．また，符号語に訂正されたときの反復回数の特性を求め，計算量削減の検討を行った．その結果，E_b/N_0 が大きくなるにつれて少ない反復回数で訂正される割合が増加し，計算量削減の効果を確認することができた．

参 考 文 献

1) C. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: turbo-codes. In *Proc. ICC'93*, Vol. 2, pp. 1064–1070, Geneva, Switzerland, May (1993).

2) J. Hagenauer, E. Offer, and L. Papke. Iterative decoding of binary block and convolutional codes. *IEEE Trans. Inform. Theory*, Vol. 42, No. 2, pp. 429–445, Mar. (1996).

3) R. M. Pyndiah. Near-optimum decoding of product codes: block turbo codes. *IEEE Trans. Commun.*, Vol. 46, No. 8, pp. 1003–1010, Aug. (1998).

4) M. P. C. Fossorier and S. Lin. Soft-input soft-output decoding of linear block codes based on ordered statistics. In *Proc. GLOBECOM'98*, Vol. 5, pp. 2828–2833, Sydney, Australia, (1998).

5) L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Inform. Theory*, Vol. 20, No. 2, pp. 284–287, Mar. (1974).

6) D. Chase. Class of algorithms for decoding block codes with channel measurement information. *IEEE Trans. Inform. Theory*, Vol. 18, No. 1, pp. 170–182, Jan. (1972).

7) M. P. C. Fossorier and S. Lin. Soft-decision decoding of linear block codes based on ordered statistics. *IEEE Trans. Inform. Theory*, Vol. 41, No. 5, pp. 1379–1396, Sept. (1995).

8) 清水隆史, 笹岡秀一. リードソロモン符号のテーブル参照軟判定復号法の検討. 信学技報, Vol. IT2004-67, pp. 101–106, Mar. (2005).