

トレリスの枝削除による q 元ターボ復号の計算量低減

長谷川 裕[†] 細谷 剛[†] 八木 秀樹[†] 平澤 茂一[†]

[†] 早稲田大学理工学部経営システム工学科 〒169-8555 東京都新宿区大久保 3-4-1

E-mail: †{hasegawa,hosoya,yagi,hirasawa}@hirasa.mgmt.waseda.ac.jp

あらまし ターボ符号は複数個の比較的符号長の小さい要素符号を接続して構成される符号であり、ターボ復号は要素符号を復号した結果を繰り返し伝播することで全体の復号性能を向上させる。ターボ符号の研究は2元符号が主流であるが q 元符号の研究も行われており、2元ターボ符号に比べて復号アルゴリズムの収束性が良いことや、遅延が減少する等の利点が報告されている。 q 元ターボ符号に対する繰り返し復号法では、各繰り返しの各時点で $(q-1)$ 個の対数事後確率比・外部情報の計算が必要であり、その計算量が大きくなる。そこで、本研究では q 元トレリス符号を用いて、見込みのないシンボルに対応する枝を早期に削除することで、計算量を低減する復号法を提案する。また、提案したアルゴリズムによって、計算量を低減できること、復号性能の大幅な劣化は見られないことをシミュレーションにより示す。

キーワード q 元ターボ符号, ターボ復号法, 最大事後確率復号法, BCJR アルゴリズム, トレリス符号

A Decoding Algorithm for q -ary Turbo Codes by Deleting Trellis Branches

Yu HASEGAWA[†], Gou HOSOYA[†], Hideki YAGI[†], and Shigeichi HIRASAWA[†]

[†] Department of Industrial and Management Systems Engineering, School of Science and Engineering,
Waseda University Okubo 3-4-1, Shinjuku-ku, Tokyo, 169-8555 Japan

E-mail: †{hasegawa,hosoya,yagi,hirasawa}@hirasa.mgmt.waseda.ac.jp

Abstract The combination of turbo codes and turbo decoding achieves good error performance. Turbo codes are constructed by concatenating short enough component codes and turbo decoding is iteratively performed by exchanging decoding results of component codes. Not only binary turbo codes but q -ary turbo codes have been in some studies. The q -ary turbo codes have good convergence speed of their decoding and its decoding delay is small compared with the case of binary one. In the case of q -ary trellis turbo decoding scheme, the $q-1$ log-likelihoods and extrinsic values have been calculated and passed iteratively. In this paper, we propose an efficient decoding algorithm for q -ary trellis turbo codes by deleting trellis branches which need not be calculated at early stage of decoding. We show by simulation results that the proposed algorithm achieves low decoding complexity without degradation in error performance.

Key words q -ary turbo codes, turbo decoding, Maximum a-posteriori probability decoding, BCJR algorithm, trellis codes

1. ま え が き

1993年のC. Berrouらによるターボ符号[1]と、確率伝播に基づく反復復号法であるターボ復号[2]の発明は、現実的な計算量でShannon限界に迫る復号性能を実現させた。ターボ復号の各繰り返しでは、最大事後確率復号を忠実に実現するBCJRアルゴリズム[6]の他、その近似である準最適なMax-Log-MAPアルゴリズム、SOVAアルゴリズム[2]が用いられることが多

い。またターボ復号において、復号の停止規範の検討は重要であり、大幅な特性劣化を伴わず、効率的に反復回数を削減する方法が研究されている。さらに、並列接続ターボ符号の研究の他にも、直列接続、要素復号器を複数接続させるマルチターボ符号に対する研究もなされている。

2元ターボ符号の研究が主流ではあるが、 q 元ターボ符号の研究もなされており、2元ターボ符号に比べて復号アルゴリズムの収束性が良いことや、遅延が減少する等の利点が報告され

ている [4]. 復号法は、2 元ターボ符号の繰り返し復号法と基本的には同じであるが、要素復号器の各時点について $(q-1)$ 個の対数事後確率比の計算を行い、要素復号器間で $(q-1)$ 個の外部情報の伝播が必要になる [5].

各時点の q 個のシンボルの中には、解となる見込みのないシンボルが含まれている。そこで本研究では、復号アルゴリズムの各繰り返しにおいて、 $(q-1)$ 個のシンボルのうち、解となる見込みがないものを判定する方法を考える。解となる見込みのないシンボルであると予想される場合、該当するトレリスの枝を削除することで計算量を低減する復号アルゴリズムを提案する。また、提案したアルゴリズムによって計算量を低減できること、同時に復号性能の大幅な劣化は見られないことをシミュレーションにより示す。

2. 準備

2.1 通信路のモデル

情報源から発生した N 個の q 元シンボルから成る情報系列 $\mathbf{u} = (u_0, u_1, \dots, u_{N-1})$, $u_k \in \{0, 1, \dots, q-1\}$ は、符号器で符号系列 $\mathbf{x} = (x_0, x_1, \dots, x_{N-1})$ に変換され、加法的白色ガウス雑音 (AWGN) 通信路を介して送られる。ここで、 $\mathbf{x}_k = (x_{k,1}, x_{k,2}, \dots, x_{k,v})$ であり、 v はパリティシンボルの数を表す。AWGN 通信路では雑音 $\mathbf{e} = (e_0, e_1, \dots, e_{N-1})$ が多次元正規分布に従って発生し、受信側では受信系列 $\mathbf{y} = \mathbf{x} + \mathbf{e}$ を受け取る。復号器では受信系列から元の情報系列を推定し、復号系列 $\hat{\mathbf{u}} = (\hat{u}_0, \hat{u}_1, \dots, \hat{u}_{N-1})$ を出力する。

2.2 2 元ターボ符号

2 元ターボ符号器は、複数の要素符号器を接続して構成される [1], [2] (図 1)。インタリーブでは情報系列の順番の並び替えが行われ、インタリーブされた情報系列 $\hat{\mathbf{u}} = (\hat{u}_0, \hat{u}_1, \dots, \hat{u}_{N-1})$ が出力される。要素符号器 $m \in \{1, 2\}$ では、それぞれパリティシンボル $\mathbf{x}_{mp} = (x_{mp,0}, x_{mp,1}, \dots, x_{mp,N-1})$ が出力される。ある $k = 0, 1, \dots, N-1$, 番目の情報シンボル u_k の符号語シンボルは $\mathbf{x}_k = (u_k, x_{1p,k}, x_{2p,k})$ となる。

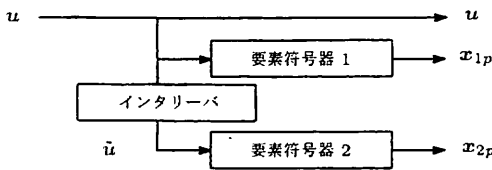


図 1 ターボ符号器

2.3 2 元ターボ復号法

ターボ復号 [1], [2] は、要素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。図 2 に復号器を示す。ここで、 $\mathbf{y}_l = (y_{l,0}, y_{l,1}, \dots, y_{l,N-1})$ は、情報系列 \mathbf{u} に対応する受信系列を表し、 \mathbf{y}_{mp} , $m \in \{1, 2\}$ は \mathbf{x}_{mp} に対応する受信系列を表す。

要素復号器 m では、シンボル毎の最大事後確率復号法によって、対数事後確率比 $L_m(\cdot)$ が計算される。このとき、対数事後確率比 $L(\hat{u}_k)$ は次式のように「事前情報 (第 1 項) + 尤度比 (第 2 項) + 外部情報 (第 3 項)」に分解される。

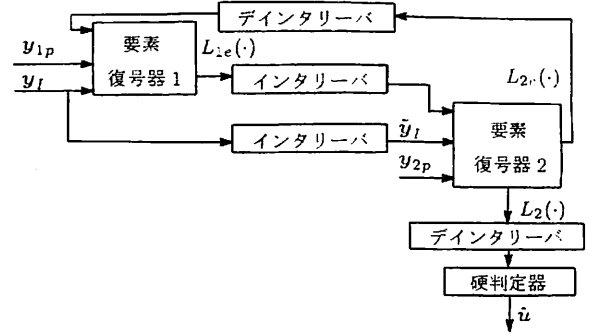


図 2 2 元ターボ復号器

$$\begin{aligned} L_m(\hat{u}_k) &= \log \frac{P(u_k = 1 | \mathbf{y})}{P(u_k = 0 | \mathbf{y})} \\ &= L_{ma}(\hat{u}_k) + \frac{2}{\sigma^2} \cdot y_{l,k} + L_{mc}(\hat{u}_k). \end{aligned} \quad (1)$$

ここで、 $L_{ma}(\cdot)$ は要素復号器 m に入力される事前情報を、 $L_{mc}(\cdot)$ は次の復号器に出力される外部情報を表す。

一方の要素復号器で計算された外部情報は他方の要素復号器に伝播され、他方の要素復号器の事前情報として利用される。要素復号器間では、受信系列と外部情報がインタリーブによって順番が並び換えられ、デインタリーブによって戻される。

ハード判定とは、対数事後確率比 $L(\hat{u}_k)$ からシンボル u_k が 0 か 1 かを定める操作である。この操作では要素復号器より得られた $L(\hat{u}_k)$ から、次式のように事後確率の大きい情報シンボルを送られた情報シンボルと推定する。

$$\hat{u}_k = \begin{cases} 1, & \text{if } L(\hat{u}_k) > 0; \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

2.4 BCJR アルゴリズム [6]

BCJR アルゴリズムは最大事後確率復号を達成するための符号のトレリスを用いたアルゴリズムである。ターボ復号における各要素復号器では、BCJR アルゴリズムを用いて事後確率が計算される。

要素符号の時点 k におけるトレリス線図を考える。まず、時点 k におけるトレリス線図のある状態を s_k で表す (図 3)。ここで、要素復号器へ入力される k 番目の受信系列を $\mathbf{y}_k = (y_{l,k}, y_{mp,k})$ とする。

このとき、時点 k において、 $L_m(\hat{u}_k)$ は次式のように表現される。

$$\begin{aligned} L_m(\hat{u}_k) &= \log \frac{P(u_k = 1 | \mathbf{y})}{P(u_k = 0 | \mathbf{y})} \\ &= \log \frac{\sum_{\substack{(s_{k-1}, s_k) \\ u_k=1}} p(s_{k-1}, s_k, \mathbf{y})}{\sum_{\substack{(s_{k-1}, s_k) \\ u_k=0}} p(s_{k-1}, s_k, \mathbf{y})}. \end{aligned} \quad (3)$$

ここで、 $p(s_{k-1}, s_k, \mathbf{y})$ は \mathbf{y} と状態 s_{k-1} から s_k への遷移に関するパスの同時生起確率を表す。無記憶通信路での伝送におい

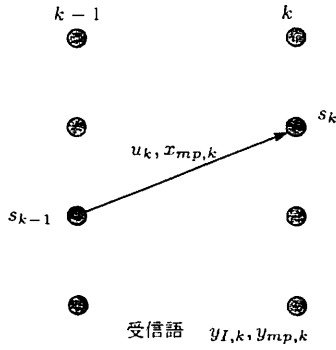


図3 トレリス線図のラベル

て $p(s_{k-1}, s_k, \mathbf{y})$ は次式のように独立な確率の積として与えられる。

$$p(s_{k-1}, s_k, \mathbf{y}) = p(s_{k-1}, \mathbf{y}_0^{k-1}) \cdot P(s_k | s_{k-1}) \cdot p(\mathbf{y}_k | s_{k-1}, s_k) \cdot p(\mathbf{y}_{k+1}^{N-1} | s_k). \quad (4)$$

ここで、 y_i^j は \mathbf{y} の $i \sim j$ 番目の成分からなる部分系列を表している。式(4)における各確率の項を次のように表す。

$$\alpha_k(s_k) = p(s_k, \mathbf{y}_0^k). \quad (5)$$

$$\beta_k(s_k) = p(\mathbf{y}_{k+1}^{N-1} | s_k). \quad (6)$$

$$\gamma_k(s_{k-1}, s_k) = p(\mathbf{y}_k | s_{k-1}, s_k) \cdot P(s_k | s_{k-1}). \quad (7)$$

式(7)は次式のように書き換えられる。

$$\gamma_k(s_{k-1}, s_k) = p(y_{l,k} | u_k) \cdot p(y_{mp,k} | u_k) \cdot P(u_k). \quad (8)$$

ここで、 $p(y_{l,k} | u_k)$ は情報の尤度、 $p(y_{mp,k} | u_k)$ はパリティシンボルの尤度、 $P(u_k)$ は事前確率を表す[3]。

式(5)、(6)について次の再起式を利用して、式(11)のように $L(\hat{u}_k)$ を求める。

$$\alpha_k(s_k) = \sum_{s_{k-1}} \alpha_{k-1}(s_{k-1}) \cdot \gamma_k(s_{k-1}, s_k). \quad (9)$$

$$\beta_{k-1}(s_{k-1}) = \sum_{s_k} \gamma_k(s_{k-1}, s_k) \cdot \beta_k(s_k). \quad (10)$$

$$L_m(\hat{u}_k) = \log \frac{\sum_{\substack{s_{k-1}, s_k \\ u_k=1}} \alpha_{k-1}(s_{k-1}) \cdot \beta_k(s_k) \cdot \gamma_k(s_{k-1}, s_k)}{\sum_{\substack{s_{k-1}, s_k \\ u_k=0}} \alpha_{k-1}(s_{k-1}) \cdot \beta_k(s_k) \cdot \gamma_k(s_{k-1}, s_k)}. \quad (11)$$

式(9)の計算は前向き計算、式(10)の計算は後向き計算と呼ばれる。得られた $L_m(\hat{u}_k)$ は式(1)のように分解されることから、外部情報は $L_m(\hat{u}_k)$ を BCJR アルゴリズムで求めてから事前情報と情報の尤度比を引くことで求まる。

2.5 q 元ターボ復号法 [5]

以降では、 q 元ターボ符号を考える。 q 元ターボ符号の復号法は、 $(q-1)$ 個のシンボルについて要素復号器間で外部情報の伝播及び、対数事後確率の計算を行う。シンボル $\mu \in \{0, 1, \dots, q-1\}$ の対数事後確率比を次式のように定義する。

$$L_{\mu,m}(\hat{u}_k) = \log \frac{P(u_k = \mu | \mathbf{y})}{P(u_k = 0 | \mathbf{y})}. \quad (12)$$

また、対数事後確率比と同様に、事前情報は次式のように定義される。

$$L_{\mu,ma}(\hat{u}_k) = \log \frac{P(u_k = \mu)}{P(u_k = 0)}. \quad (13)$$

図4に q 元ターボ復号器の振る舞いを示す。 $L_{\mu,m}(\cdot)$ は要素復号器 m から出力される μ に対する対数事後確率比を、 $L_{\mu,mc}(\cdot)$ は要素復号器 m から出力される μ に対する外部情報を、 $L_{\mu,ma}(\cdot)$ は要素復号器 m に入力される μ に対する事前情報を表す。 q 元ターボ復号法でも、 u_k の事後確率と外部情報は、BCJR アルゴリズムによって計算できる。

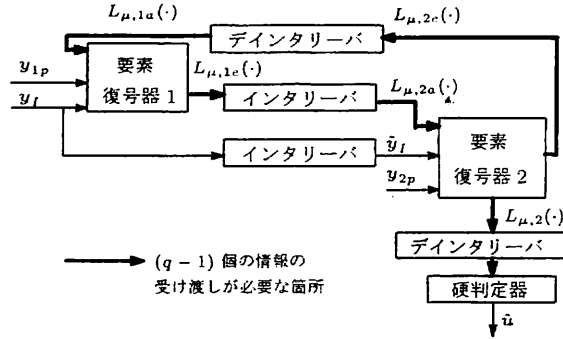


図4 q 元ターボ復号器

硬判定は次式のように行なわれる。

$$\hat{u}_k = \begin{cases} 0, & \text{if } \max_{\mu} L_{\mu,m}(\hat{u}_k) < 0; \\ \arg \max_{\mu} L_{\mu,m}(\hat{u}_k), & \text{otherwise.} \end{cases} \quad (14)$$

要素復号器 m より得られた $L_{\mu,m}(\hat{u}_k)$ から、事後確率を最大とする情報シンボルを送られた情報シンボルと推定する。もしすべての対数事後確率比が負ならば、 $\hat{u}_k = 0$ と推定する。

3. 提案手法

q 元ターボ復号では、 q 個のシンボルのうち最も事後確率の大きいシンボルに復号する。しかし、 q 個のシンボルの中には、復号アルゴリズムの繰り返し回数が少ない段階で、事後確率が最大になる見込みのないシンボルが含まれている場合もある。そこで本節では、このような見込みのないシンボルに対応する枝をトレリスから削除することで、計算量を低減する手法を提案する。

3.1 シンボル削除法

あるシンボル μ の事後確率が、最も大きい事後確率と比較して十分小さければ、解として判定される見込みのないシンボル μ と見なし、削除することを考える。事後確率最大のシンボルとの比較を容易にするため、対数事後確率比を次式のように定義する。

$$L_{\mu,m}^{(i)}(\hat{u}_k) = \log \frac{P(u_k = \mu | y)}{P(u_k = \mu_{\max} | y)}. \quad (15)$$

ここで、 $i = 1, 2, \dots$ は繰り返し回数を表し、 μ_{\max} は i 回目の繰り返しにおいて、最大の事後確率をもつシンボルを表す。

式 (15) を用いて、見込みのないシンボルを判定する。もし、各要素復号器の復号直後に次式を満たすならば、そのシンボル μ の事前確率を $P(u_k = \mu) = 0$ とする。

$$L_{\mu,m}^{(i)}(\hat{u}_k) < \epsilon \quad \text{if } i \geq 1. \quad (16)$$

ここで、 $\epsilon (< 0)$ はある閾値を表す。ただし、少なくとも各要素復号器で一度は復号させるように、1 回目の繰り返し終了時点からトレリスの枝削除を開始する。以下に、 i 回目の繰り返しにおける要素復号器 m で、復号される見込みのないシンボルの枝を削除するアルゴリズムを記す。

[トレリスの枝削除アルゴリズム]

1) $k := 0$ とする。

$\mu = 0, 1, \dots, q-1$ について、 $L_{\mu,m}^{(i)}(\hat{u}_k)$ を計算する。

もし、 $L_{\mu,m}^{(i)}(\hat{u}_k) < \epsilon$ ならば 2) へ行く。

そうでなければ、3) へ行く。

2) $L_{\mu,m}(\hat{u}_k) := -\infty$ とし、次の復号トレリスの k 時点目のシンボル μ の枝を削除する。

3) へ行く。

3) もし $k = N-1$ ならば、要素復号器 m での復号を終了する。

そうでなければ、 $k := k+1$ とし、1) へ行く。 □

この削除アルゴリズムによって見込みのないシンボル μ の枝が削除されるため、対数事後確率比 $L_{\mu,m}^{(i)}(\hat{u}_k)$ は計算される必要がなくなる。

3.2 閾値の設定法

本節では、提案した削除法に用いる閾値 ϵ の値を設定する方法を示す。復号器にとって通信路の SN 比は未知であると仮定する。従って、通信路の SN 比によらず、受信系列から閾値 ϵ_0 を設定する方法を提案する。

各時点 k のシンボル u_k には判定されたシンボルの信頼度が大きいものと、小さいものがある。判定しやすいシンボルというのは、復号を繰り返しても最大の事後確率 $P(u_k = \mu_{\max} | y)$ をもつシンボル μ_{\max} が変化しないシンボルである。このようなシンボルは、2 番目に大きい事後確率をもつシンボル μ_{sec} と比較して、 $P(u_k = \mu_{\max} | y)$ と $P(u_k = \mu_{\text{sec}} | y)$ の差は大きいと考えられる。

そこで、繰り返し $i = 1$ 回目を得られた $P(u_k = \mu_{\max} | y)$ と $P(u_k = \mu_{\text{sec}} | y)$ の差が最大となる時点における μ_{sec} の対数事後確率比を閾値 ϵ_0 として設定する。

$$\epsilon_0 = \min_k \{L_{\mu_{\text{sec}},m}^{(1)}(\hat{u}_k)\}. \quad (17)$$

式 (17) で与えられた閾値 ϵ_0 は、 $i = 1$ の各時点 k について、少なくとも 2 つのシンボル $\mu_{\max}, \mu_{\text{sec}}$ は残すような値をとる。

3.3 提案手法によって低減される計算量

本節では提案手法によって、低減される計算量について述べる。

BCJR アルゴリズムにおいて、トレリスの枝の削除による効果があるのは以下の 3 点である。

- (1) 前向き計算における計算量低減
- (2) 後向き計算における計算量低減
- (3) 対数事後確率の計算における計算量低減

(1) 前向き計算における計算量低減

枝の削除によって、各ノードに入る枝の本数は少なくなるため、式 (9) の計算量は低減される。削除されたシンボルひとつ当たり、状態数分の加算・乗算の演算が不要となる。

(2) 後向き計算における計算量低減

前向き計算と同様である。

(3) 対数事後確率の計算における計算量低減

従来、各繰り返し i において、 $q-1$ 個の対数事後確率比 $L_{\mu,m}^{(i)}$ を計算している。一方、提案手法では削除されたシンボルの対数事後確率比は計算しないため、対数事後確率比、外部情報の計算にかかる計算量が低減される。

4. シミュレーションによる評価

本節では、提案手法の有効性を評価するためシミュレーションについて記す。

4.1 評価条件

シミュレーションでは、情報 1 シンボルが 2 ビットで表現される 4 元符号を用い、変調方式は QPSK と BPSK を用いる。BPSK を用いる方式では、1 ビットずつ BPSK を用いて送信し、復号の際は 2 ビットをまとめて 1 シンボルとして復号する。符号化率 $2/3$ 、遅延素子数 3 (状態数 8) の同一の畳み込み符号を 2 つ並列に接続し、情報系列長 100 の 4 元ターボ符号を用いる。全体の符号化率は $1/2$ となる。ここで最大繰り返し回数 i_{\max} を 10 回とし、各 SN 比についてシミュレーションを行った。0.0~2.5 [dB] では、シミュレーション回数を 10^4 、3.0~4.0 [dB] では、シミュレーション回数を 10^5 とした。

評価指標には、次式で定義されるシンボルの残存率 SR を用いる。

$$SR = \frac{\sum_{i=1}^{i_{\max}} (q \times S \times N - \sum_{j=0}^{i-1} \text{count}[j])}{q \times S \times N \times i_{\max}}. \quad (18)$$

ここで、 i_{\max} は最大繰り返し回数、 S は状態数、 $\text{count}[j]$ は繰り返し j 回目終了後に削除した枝の数を表す。残存率は、提案手法の全ての繰り返しによって削除されなかった枝の割合を示しており、この値が小さいほど提案手法の計算量は低減される。式 (18) より、繰り返し回数が少ないうちに多くの枝が削られると残存率は向上することが分かる。

4.2 結果と考察

(1) 残存率と BER の関係

図 5, 6 に情報系列長 $N = 100$, SN 比 1.5, 2.5 [dB] において, 閾値 $\epsilon < 0$ の変化による BER と残存率の関係を示す. 縦軸をビット単位の復号誤り率 (BER), 横軸を残存率 (SR) としている. なお, 閾値 ϵ を大きくすれば, 残存率 SR は小さくなる. また, 少なくとも必ず一本の枝は残すので, 残存率 SR は $0.25 < SR \leq 1.0$ となる. 従来手法は, $\epsilon = -\infty (SR = 1.0)$ といった場合に相当する.

図 5, 6 より, 残存率が大きい範囲 ($\epsilon \ll 0$) では BER はほとんど変わらないことが分かる. これは, 枝を削除しても復号に影響を与えないほど事後確率が小さいシンボルのみを, 削除しているためであると思われる.

一方, 小さい残存率の範囲では BER の向上が見られる. これは, 復号に影響を与える遷移確率 $\gamma_k(s_{k-1}, s_k)$ をもつが, 送信シンボルと異なるシンボルの枝が削られたためであると思われる. 送信シンボルと異なるシンボルを正確に削ることができるならば, 誤ったシンボルの遷移確率の影響を受けることなく復号される. 従って $\alpha_k(s_k), \beta_k(k, s)$ をより正確に求めることができる. そのため, 対数事後確率比がより正確になり, 復号性

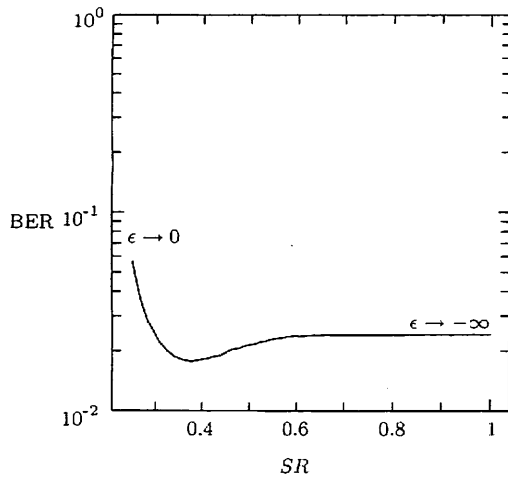


図 5 平均残存率と BER の関係 (1.5 [dB])

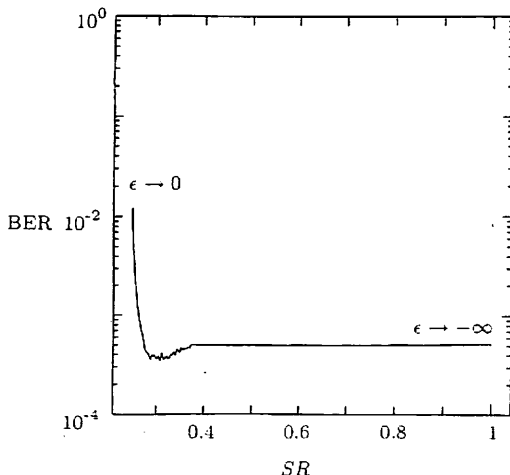


図 6 平均残存率と BER の関係 (2.5 [dB])

能が向上すると思われる.

さらに閾値 ϵ を大きくすると, 残存率は小さくなるが, BER が大きく劣化する残存率の範囲になる. これは, 送信シンボルを削除してしまう可能性が大きいからである.

以上の結果より, 閾値 ϵ を適切に設定することができるならば, BER を劣化させることなく, 多くの枝を削除することができると思われる.

(2) 従来手法と提案手法の BER の比較

従来のトレリスの枝を削除しない q 元ターボ復号法と提案したトレリスの枝を削除する手法における BER を, QPSK, BPSK についてそれぞれ図 7, 8 に示す.

図 7, 8 より, 提案手法の BER は僅かではあるが, 向上していることがわかる. ここで, 式 (17) に提案した閾値の設定法を用いている. その結果, BER が向上する範囲に閾値 ϵ_0 を設定することができていると言える.

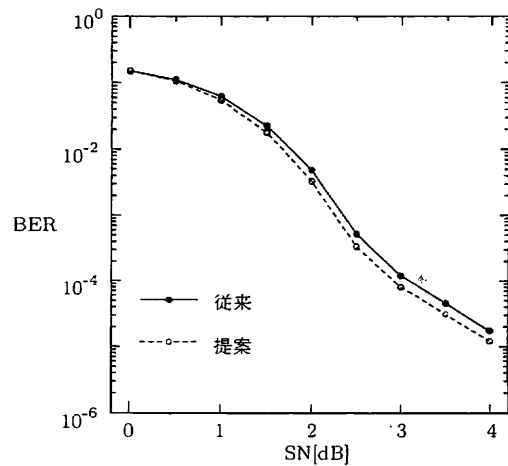


図 7 QPSK における従来手法と提案手法の BER の比較 (情報系列長 100)

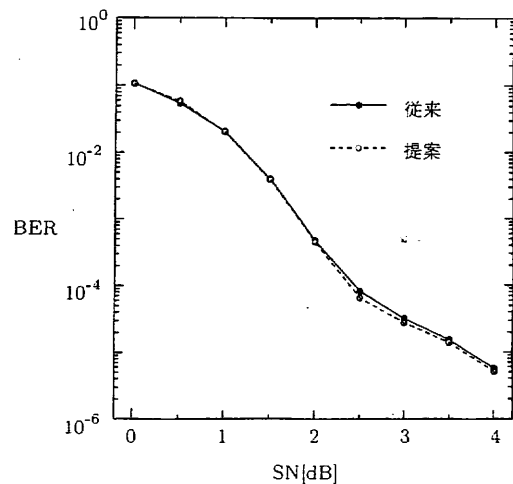


図 8 BPSK における従来手法と提案手法の BER の比較 (情報系列長 100)

(3) 残存率と SN 比の関係

提案手法における残存率 SR を QPSK, BPSK についてそれぞれ図 9, 10 に示す。

図 9, 10 より, 高 SN 比において残存率は小さくなり, 提案手法により計算量は大きく低減されていることがわかる。これは, 高 SN 比においては, 加わる雑音の分散が小さく, 判定しやすいシンボルが多いためであると思われる。

一方, SN 比が小さくなるにつれて, 残存率は大きくなり, 計算量はあまり低減できていないことが分かる。低 SN 比では加わる雑音の分散が大きいため, 事後確率が最大のシンボルとの差が小さく, 見込みのないシンボルであると判定することが困難であるためと思われる。

図 9, 10 より, 低 SN 比では残存率 SR は大きい値を示しているが, 高 SN 比では残存率 SR を約 35% にすることができる。これにより, 比較的 SN 比が大きいところで提案手法が有効であると言える。

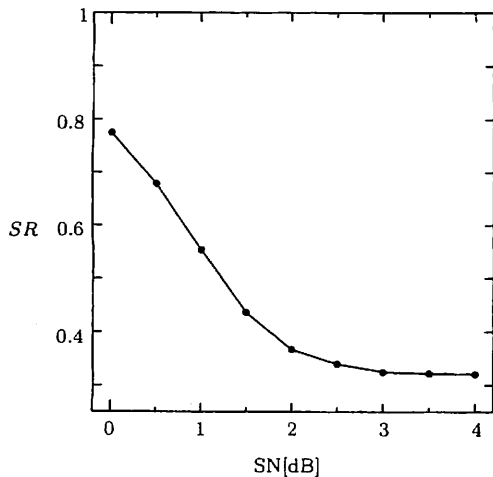


図 9 QPSK における平均残存率 (情報系列長 100)

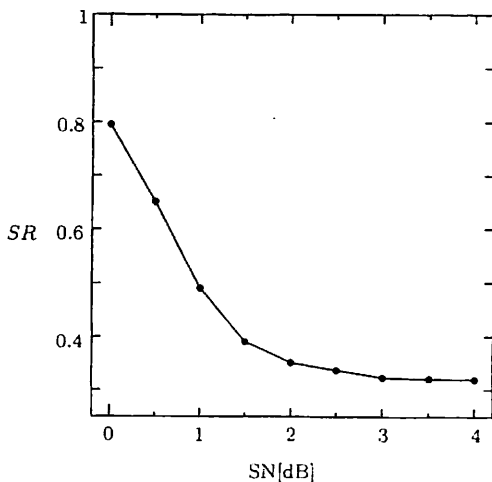


図 10 BPSK における平均残存率 (情報系列長 100)

5. まとめと今後の課題

本研究では, q 元ターボ符号に対し, 初期の段階で判定される見込みのないシンボルをトレリスの枝から削除することで, 計算量の低減を可能にする復号アルゴリズムを提案した。また, シミュレーションにより, 従来手法と比較して提案手法が計算量を低減できることを示した。また, 同時に復号誤り率の向上にも効果があることを示した。

今後の課題として, 最適な閾値 ϵ を設定する方法の検討, 提案手法の理論的な解析, $q > 4$ でのシミュレーションが挙げられる。また, 本提案手法を用いた場合の復号停止規範の検討も今後の課題である。

謝辞 著者の一人長谷川は, 本研究を進めるにあたり多くの有益なご助言, ご支援を賜りました早稲田大学平澤研究室の諸氏に深く感謝いたします。

文 献

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," *Proc. of IEEE International Conference on Commun.*, Genova, Italy, pp.1064-1071, May, 1993.
- [2] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol.42, pp.429-445, Mar, 1996.
- [3] 井坂元彦, 今井秀樹, "Shannon 限界への道標: "parallel concatenated (turbo) coding", "turbo (iterative) decoding" とその周辺," 信学技報, pp.1-18, 1998-12.
- [4] C. Berrou, M. Jezequel, C. Douillard, and S. Kerouedan, "The advantages of non-binary turbo codes," *Proc. of IWT2001*, pp.61-63, Sept. 2001.
- [5] J. Berkman, "On turbo decoding of nonbinary codes," *IEEE commn. letters*, vol.2, No.4, pp.94-96, April 1998.
- [6] L. R. Bahl, J. Coche, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol.5, pp.284-287, Mar. 1974.