

研究速報

出力分布に応じてゆう度に上限を設けるターボ復号に関する一検討

松井 宗大^{†*}(学生員) 山里 敬也^{††a)}(正員)
片山 正昭^{††(正員)} 小川 明^{†††(正員)}

A Study on SOVA Based Turbo-Decoding with Threshold
Munehiro MATSUI^{†*}, Student Member,
Takaya YAMAZATO^{††a)}, Masaaki KATAYAMA^{††}, and
Akira OGAWA^{†††}, Regular Members

[†]名古屋大学電子情報学専攻, 名古屋市
Department of Information Electronics, Nagoya University,
Nagoya-shi, 464-8603 Japan

^{††}名古屋大学情報メディアセンター, 名古屋市
Center for Information Media Studies, Nagoya University,
Furo-cho, Chikusa-ku, Nagoya-shi, 464-8603 Japan

^{†††}名城大学理工学部, 名古屋市
School of Science and Technology, Meijo University, 1-501
Shiogamaguchi, Tenpaku-ku, Nagoya-shi, 468-8502 Japan

* 現在, 日本電信電話株式会社 NTT 未来ねっと研究所

a) E-mail: yamazato@nuee.nagoya-u.ac.jp

あらまし 本論文では, SOVA (Soft-Output Viterbi Algorithm) に基づくターボ復号において, 算出されたゆう度に対して上限 (スレッシュホールド) を設ける手法を提案し, 計算機シミュレーションによりビット誤り率特性が改善されることを示す。

キーワード ターボ復号, SOVA, ゆう度, 上限 (スレッシュホールド)

1. ま え が き

近年, 簡単な符号構成で高利得を得ることができる, ターボ符号 [1] が脚光を浴びている。

ターボ符号の復号は, ターボ復号 (反復復号) と呼ばれる独特の復号法を用い, 構成する復号器間で軟出力をやり取りすることで復号が行われる。軟出力を算出するのに用いられる復号アルゴリズムには, MAP (Maximum A Posteriori probability) アルゴリズム, SOVA (Soft-Output Viterbi Algorithm) がある。MAP アルゴリズムは最良のビット誤り率を与えるが, 複雑度は大きなものとなる。一方の SOVA は複雑度は低い反面, ビット誤り率特性は MAP に比べて劣る。こうした現象は, SOVA が算出するゆう度が過大であることに起因し, ゆう度を調整することで特性が改善されることが報告されている [2]。

SOVA から過大に算出されるゆう度を抑えるために, 算出されたゆう度に対して上限 (スレッシュホールド) を設ける手法が提案された [3], [4]。これにより, 初期のターボ復号において算出されたゆう度に固定の上限値を設けることで, 特性の改善を図ることができる。し

かしながら, 上記の手法では, ターボ復号を行うにつれて変動する軟出力分布を考慮に入れておらず, 効果が小さいものと推測される。そこで本論文では, ゆう度に対して設ける上限値を可変とする手法を提案する。この上限値を誤りを含む可能性に応じて設定する。誤りを含む可能性に応じて設定するために, 上限値を軟出力分布の平均値と分散値に依存するように設定する。本論文は以下のように構成される。まず, 次章において, 上限値を設けるターボ復号法について説明する。続いて 3. では, 提案するターボ復号法について述べ, 4. においてシステムモデルを示す。5. において, 計算機シミュレーションによる数値例を示し, 最後に 6. において結びとする。

2. 固定の上限値を設けるターボ復号法

SOVA に基づくターボ復号は, MAP アルゴリズムに基づくターボ復号に対して復号の複雑度は低い反面, 誤り率特性は劣化してしまう。これは, SOVA が過大にゆう度を算出してしまい, 結果的にターボ復号の効果が低下することに起因する。SOVA におけるゆう度算出に用いられるパスの数が, MAP アルゴリズムに比べて少ないことから上記の現象が発生する [2]。

図 1 に SOVA の出力 (LLR: Log Likelihood Ratio) 分布を示す。ただし, 縦軸は確率密度, 横軸はゆう度である。SOVA 復号器への入力信号は時刻 t における情報ビット $u_t(+1, -1)$ を平均値としてガウス分布を形成しているものとし, $\hat{u}_t = +1$ 側の分布を示して

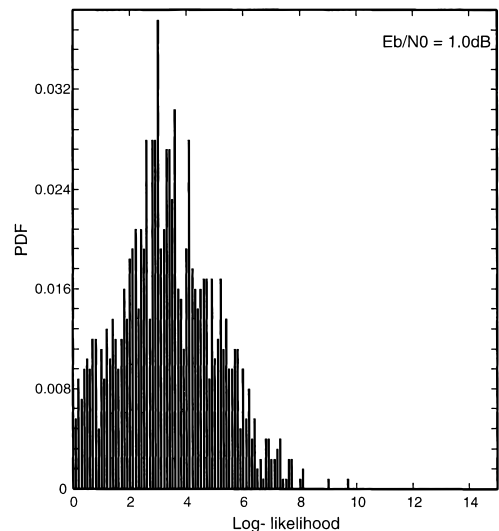


図 1 出力分布

Fig. 1 Distribution of output.

表 1 反復復号における平均値の変動

Table 1 Fluctuation of mean and standard deviation after iteration.

反復回数	1		2		3	
	DEC 1	DEC 2	DEC 1	DEC 2	DEC 1	DEC 2
平均値	3.378	10.266	23.021	46.401	88.726	165.979
標準偏差	1.320	2.623	4.834	7.767	12.608	19.335

いる．この図より，入力信号は+1を平均値としたガウス分布であったのに対し，軟出力分布は比較的高い値を平均値として分布しているのがわかる．ターボ復号を繰り返していくと，ゆう度の平均値が次第に大きくなっていき，反復復号の効率が低下する．表 1 に，1.5 dB における各反復回数での平均値と標準偏差を示す．

このゆう度の好ましくない増加を抑えるために，ゆう度上限（スレッシュホールド）を設ける手法が提案された [3]．これにより，算出されたゆう度に対して固定の上限値を設けることでゆう度の増加を抑えることができ，ターボ復号の誤り率特性を改善することができる．

しかしながら上記の手法では，上限値が固定であるために，ターボ復号によって変動していく軟出力分布を考慮に入れておらず，効果が小さいことが推測される．表 1 に示したように，分布の平均値，分散値は反復するにつれて変動していくため，上限値を固定とするのは適切ではないものと考えられる．したがって，本論文では，この上限値を可変とする手法を提案する．

3. 提案するターボ復号法

上限値を可変とするにあたって，誤りを含む可能性に応じて上限値を設定する手法を提案する．誤りを含む可能性に応じて上限値を設定するために，平均値と分散値に依存するように上限値を決定する．

提案する上限値決定の手法では，出力分布を分割する割合によって上限値を決定する．図 2 に上限値決定の概念を示す．SOVA の出力分布の正側を示しているものとする．また， m は分布の平均値を， TH は上限値を示している．この図では，分布を 20% と 80% に分割するように上限値を設定している．すなわち，上限を超えるゆう度が存在する割合が 80% となるように上限を設定し，この 80% のゆう度はすべて上限値をゆう度とする．

SOVA の出力分布はガウス分布に近い分布であるため，出力分布をガウス分布と近似することで，平均値，分散値を用いて累積分布関数より上限値を算出するこ

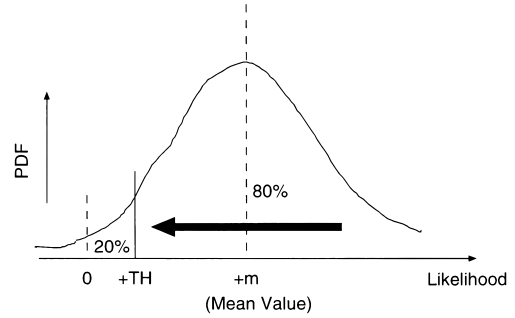


図 2 提案する上限値設定の手法

Fig. 2 Proposed method to set the threshold.

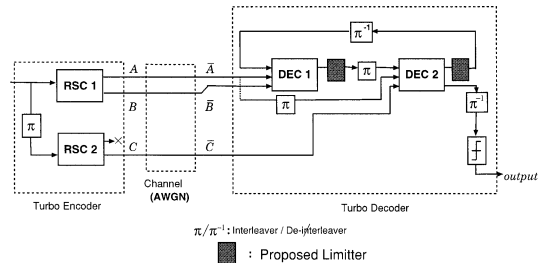


図 3 システムモデル

Fig. 3 System model.

とができる．したがって，提案する手法における計算量の増加は小さいものと考えられる．

上記の手法を用いることにより，上限値を出力分布の平均値と分散値に依存するように決定することができる．平均値が高いほど，あるいは分散値が低いほど上限値は高く設定され，ゆう度を高くすることができる．また，逆も成り立つ．したがって，誤りを含む可能性に応じてゆう度を調整することができ，反復復号の効率化が図られると期待できる．

4. システムモデル

提案するシステムモデルを図 3 に示す．ターボ符号器は，従来のターボ符号器のものと同じである．ターボ符号器に 2 値情報よりなる伝送するデータ系列が入力され，インターリーブによって並列に連結された再帰的組織畳込み符号器 (RSC1, RSC2) によって符号化されて， A, B, C の系列を得る．

符号化の後，符号化された系列は変調されて，通信路へ送られる．受信機側において，受信した信号を復調し， $\bar{A}, \bar{B}, \bar{C}$ の各系列を得る．

提案するターボ復号器は，従来のターボ復号器に，出力に対して上限を与えるリミッタを付加するものである．図 4 にリミッタを示す．リミッタでは，出力分

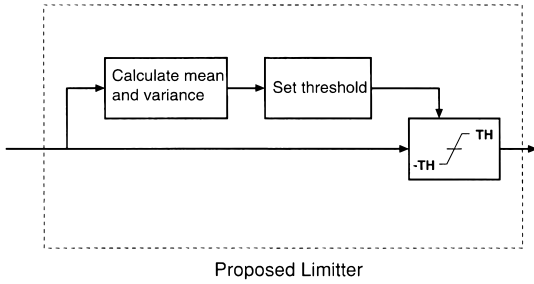


図4 リミッタ
Fig. 4 Limiter.

布の平均値と分散値を算出し、これらの値をもとに、ガウス分布の累積分布関数よりスレッシュヨルド値を導出し、出力に対して上限を設ける。

ターボ復号を行っている間、リミッタにおいて、やり取りされている軟出力に対して動的に上限を設ける。

5. 数値例

計算機シミュレーションによる数値例を示す。図5に3回反復後(3 iteration)の特性を、図6に6回反復後(6 iteration)の特性を示す。縦軸はビット誤り率を、横軸は信号と雑音のエネルギー比を表している。ゆう度が上限を超える割合が、98%、90%、80%となるように上限を設定したときの特性を示している。従来の復号法の特性も併せて示す。従来の出力に対して何も行わないターボ復号法は上記の割合で0%に相当する。

これらのシミュレーション結果より、ゆう度が上限を超える割合を約90%とした場合に良好な特性を得ることができると考えられる。図7に割合を変化させたときの特性を示す。縦軸はビット誤り率を、横軸は割合を表している。信号対雑音エネルギー比は1.0 dB、反復回数は6回としている。上記の98%、90%、80%に加えて、99%、70%、60%、40%、20%、0% (従来の復号法)の特性を示す。結果より、80%から98%とした場合に良好な特性を得ることができた。これは、割合が低いと上限を設ける効果が小さくなって特性が改善されなくなり、逆に割合が大きすぎると算出されたゆう度を小さくしすぎるので特性が劣化してしまうことに起因すると考えられる。

また、図5、図6に示すように、98%と設定した場合、6回反復後に高SNR領域において良好な特性を得ることができた。98%と設定することによって、ゆう度の上昇を緩やかにすることができる。ターボ復号では反復を行うにつれて誤り訂正がなされていくこと

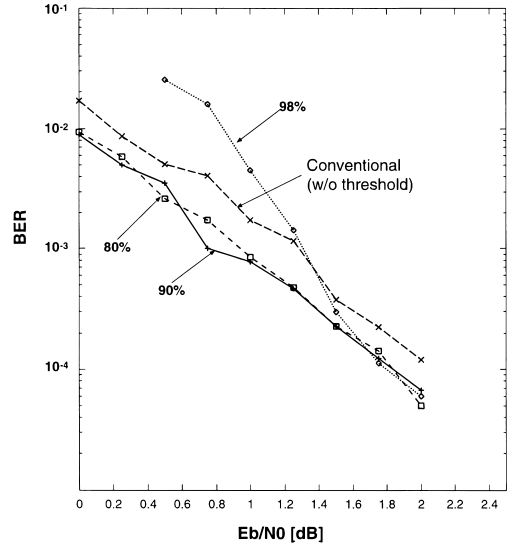


図5 3回反復後の特性
Fig. 5 BER performance after 3 iteration.

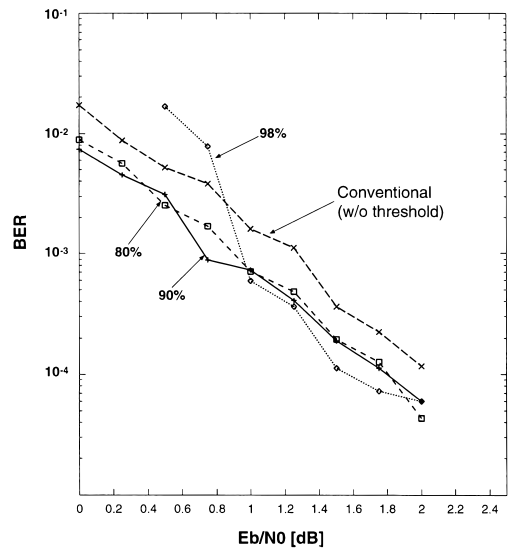


図6 6回反復後の特性
Fig. 6 BER performance after 6 iteration.

から、初期の復号ではゆう度を抑え、反復回数にしたがって徐々にゆう度を上げていくことが特性の改善につながると考えられる。ただし、良好なビット誤り率を達成するためには多くの反復回数を必要とする。

図8にビット誤り率と反復回数のグラフを示す。図よりわかるように98%と設定した場合、良好な特性を得るのに多くの反復回数を要することがわかる。一方、

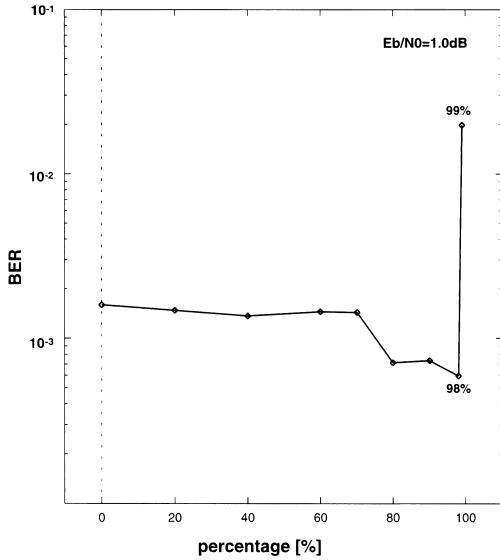


図7 割合に対する誤り率特性
Fig. 7 BER performance U.S. percentage.

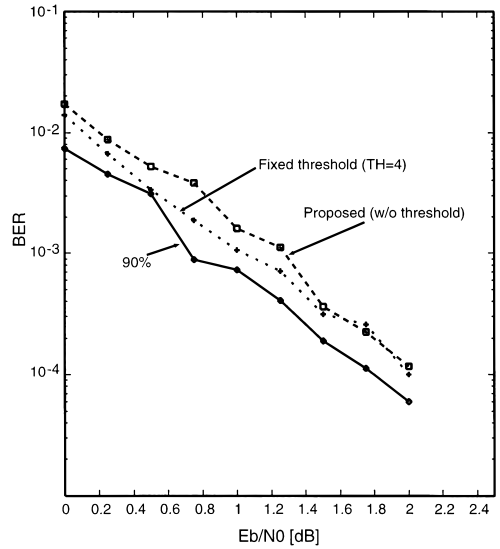


図9 固定上限法との比較
Fig. 9 Comparison with fixed threshold method.

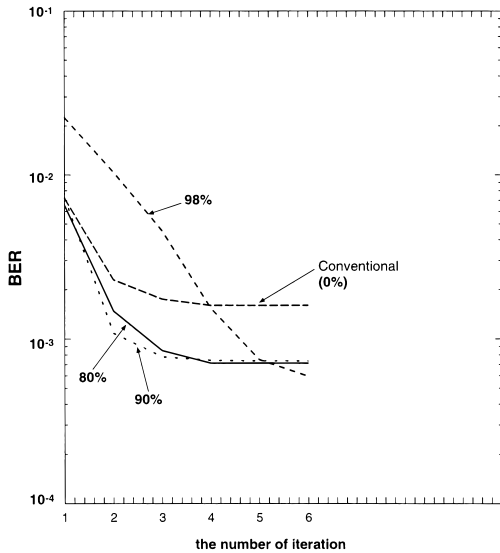


図8 反復回数に対する誤り率特性
Fig. 8 BER performance against iteration.

90%，80%と設定した場合，従来の復号法と同等の反復回数で良好な特性を得ることがわかる．上記を考慮すると，ゆう度が上限を超える割合を約90%と設定した方が良いと考えられる．

最後に図9に，上限値を固定としたターボ復号法，上限を設けない従来のターボ復号法との特性の比較を示す．固定とした場合の上限値は文献[3]に基づき，

4と設定している．上限を可変とした場合，誤り率 10^{-3} において，上限値固定のターボ復号法に対して約0.3dBの利得が，従来のターボ復号法に対して約0.6dBの利得を得ることができた．

6. むすび

本論文では，算出されたゆう度に対して，可変の上限値を設けるターボ復号法について議論した．誤りを含む可能性に応じて上限値を設定するために，出力の平均値と分散値に依存するように上限を設定する手法を提案し，計算機シミュレーションによってその有効性を示した．

文 献

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes(1)," IEEE Proc. ICC'93, pp.1064-1070, 1993.
- [2] 岡村利彦, "繰り返し復号における畳み込み符号の軟出力正規化について," SITA97, pp.641-644, 1997.
- [3] L. Lin and R.S. Cheng, "Improvement in SOVA-based decoding for turbo-codes," IEEE, Proc. ICC'97, pp.1473-1478, 1997.
- [4] C.H. Wang and C. Chao, "Modified SOVA decoding for turbo codes," IEEE Proc. ICC'98, pp.643-646, 1998.

(平成12年3月23日受付, 13年7月25日再受付, 14年1月11日最終原稿受付)