

スライディングウィンドウ復号化を使用する Braided Convolutional Codes

Kwame Ackah Bohulu

2017/11/7

1 Density Evolution Analysis 密度進化分析（みつどうしんかぶんせき）

Density Evolution Analysis を使用して、(バイナリ消去 (しょうきょ) チャンネル) BEC の上で送信するウィンドウ復号化を使用するブロック SBC 符号の漸近的な性能 (zenkintekina seinou) を分析する。
それで、分析の結果は、大きいブロックサイズの場合、復号化ウィンドウサイズと復号化ウィンドウスケジュールの選びのに使用する。

1.1 Erasure Probability of Component Decoders

ウィンドウ復号化を使用するブロック SBC 符号の解析式を導出する（かいせきしきをどうしゅつする）ために、それぞれの BCJR 復号器の外部出力消去確率 (gaibu shutsuryoku shoukyo kakuritsu) を計算する必要がある。

伝達関数 (dentatsu kansuu)

1.2 Density Evolution for SBC Codes with Window Decoding

復号化ウィンドウが、時間 t から始まって、 $t+w+1$ で終わりを想定（そうてい）する。

垂直（すいちよく）復号化のとき、 i 回の垂直（すいちよく）繰り返しを行った後、情報シンボル、入力パリティシンボルと出力パリティシンボルの外部出力消去確率 (gaibu shutsuryoku shoukyo kakuritsu) は式 8 で計算できる。

水平復号化のとき、左側から右側に移動するとき、 $D1$ ($D2$) が出力パリティシンボルの外部出力消去確率 (gaibu shutsuryoku shoukyo kakuritsu) を a-priori 確率として $D2$ ($D1$) に入力する。

最後に、 j 回の水平繰り返しを行った後、復号化しようとするブロックの復号化消去確率は式 10 で計算できる。

1.3 Results and Discussion

先行った Density Evolution Analysis で、ウィンドウサイズと復号性能の関係を確認する。そして、複雑さに対する最適なウィンドウ復号スケジュールも評価する。

ウィンドウサイズ、 w に対する BEC の閾値は表 1 にかいてあって、3 より大きい場合性能は良くならない。
大きいブロックサイズの場合、 $w=3$ にすれば性能はよくなるはず。

ウインドウ復号スケジュールに関しては、通信路の消去（しょうきょ）確率が与えたら、少ない繰り返し回数で期待している erasure 確率を得る方が最適である。

通信路の erasure 確率、期待している erasure 確率とウインドウサイズそれぞれが 0.65, 10^{-9} 、3 の場合、必要な繰り返し回数が表 2 で書かれている。

必要な垂直（すいちよく）繰り返し回数を 1 にしてもいい性能が得られるはずなので 1 にする。また、大きいブロックサイズの場合、性能たい複雑さのトレードオフのため 1 にする方がよい。

表 2 であきらかになったのは MU と LU 期待している erasure 確率をえるための繰り返し回数が一番少ない。また、繰り返し回数が同じである。

なので、大きいブロックサイズの場合、MU か LU が一番良い。

2 Performance of Blockwise SBC Codes with Sliding Window Decoding

図 5 では SU スライディングウインドウ復号化を使用するブロック SBC 符号のビット誤り率性能が書かれている。ブロックの大きさと性能の関係を示す。

あきらかになったのはブロックサイズが大きくなるほど復号化性能がよくなる。

BER= 10^{-5} を得るために必要な SNR 対復号化待ち時間が図 6 に描かれている。

あきらかになったのはブロックサイズが固定された場合、ウインドウサイズを大きくすると性能もよくなるけど、あるウインドウサイズまでよくなる。また、ある復号化待ち時間を超えたら、大きいブロックサイズの場合、小さいウインドウサイズを使用すると性能がよくなる。

繰り返し回数が固定された場合、スライディングウインドウ復号化と復号化スケジュールの性能を図 7 で比べられる。

最後に、[7] で使用する復号化方法よりいい性能が得られたことが明らかになったが、待ち時間が [7] より大きいです。

3 Computational Complexity

スライディングウインドウ復号化を使用する SBC 符号の計算量を調査し、計算量を減らすため停止（ていし）ルールを提案する。

3.1 Computational Complexity Analysis

スライディングウィンドウ復号器の計算は、大きさ T のブロックを復号化するのに必要な操作数で分析することができる。

1つのトレリス、1つの垂直（すいちょく）そして、1ビット当たり、log-MAP BCJR アルゴリズムを使用する。
復号器に必要な計算は表 3 に書かれている。

3.2 Stopping Rules

[1] クロスエントロピーに基づく停止ルール
連続の水平繰り返しをおこなった後、復号しようとする情報ブロックの APP 値の違いに基づくルールです。

経験で明らかになったことは、収束（しゅうそく）したら元のあたより 10^{-3} 下がる。

なので、 i 個の水平後に $T(i)$ が $\eta T(0)$ より小さかったら、復号化かを停止して、ウィンドウをシフトする。

[2] LLR 値の大きさに基づく停止ルール
復号しようとする情報ブロックの累積的（るいせきてき）LLR 値の収束（しゅうそく）に基づいている。

このルールは LLR の値の閾値（いきち）、 θ と depthfactor。M は LLR 値が本当に収束（しゅうそく）のかを確認する。

θ は連続の LLR 値のマグニチュードの違いが十部小さいかどうかを確認する。

なので、M 回の水平繰り返しをおこなった後値の違いが θ より小さかったら、復号化かを停止して、ウィンドウをシフトする。