様式Ｃ－１９

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成２４年５月２１日現在

機関番号：１７２０１
研究種目：若手研究（B）
研究期間：2010～2011
課題番号：22760276
研究課題名（和文）　LDPC 畳込み符号の能力評価手法の開発とそれを応用した符号設計に関する研究
研究課題名（英文）　A Method for Evaluating the Decoding Performance of LDPC Convolutional Codes and its Application to Code Constructions
研究代表者
廣友　雅徳（HIROTOMO　MASANORI）
佐賀大学・総合情報基盤センター・特任助教
研究者番号：00529795

研究成果の概要（和文）：LDPC 畳込み符号は，低密度のパリティ検査行列を畳込み符号の構造を持たせて設計する誤り訂正符号である．そのような符号の設計方法によって高い誤り訂正能力が得られ，良い復号特性を示すことが実験結果によって報告されている．本研究では，LDPC 畳込み符号の復号能力を理論的に評価することを目的に，能力評価パラメータの一つである重み分布を効率的に計算する方法を開発した．開発した方法では，低密度の検査行列から構成される符号木の特徴と，畳込み符号の構造的性質に着目して，重み分布を計算するための計算量を削減している．開発した方法をいくつかの LDPC 畳込み符号に適用し，重み分布を求めることによって，その符号の能力を評価した．

研究成果の概要（英文）：Low-density parity-check (LDPC) convolutional codes are error correcting codes constructed by sparse parity-check matrices of convolutional codes. The codes show good performance when they are decoded by iterative algorithms. In this research, we propose an efficient method for computing the low part of weight spectrum of LDPC convolutional codes. The minimum free distance and the weight spectrum are important factors to evaluate the error performance of LDPC convolutional codes. We focus on the structure of sparse parity-check matrices and the property of convolutional codes in order to reduce the computational cost for enumerate the low-weight codewords. Furthermore, we apply the proposed method to several LDPC convolutional codes, and evaluate the performance of their codes.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
2010 年度	1,600,000	480,000	2,080,000
2011 年度	1,100,000	330,000	1,430,000
年度			
年度			
年度			
総　計	2,700,000	810,000	3,510,000

研究分野：　情報通信工学
科研費の分科・細目：電気電子工学　通信・ネットワーク工学
キーワード：誤り訂正符号，LDPC 畳込み符号，復号特性，重み分布，最小自由距離

１．研究開始当初の背景

今日の情報化社会ではサービスのクラウド化が進み，携帯端末による 3G データ通信や高速 Ethernet を介してそのサービスを利用する．ネットワークを介して通信されるデータは伝送中に生じる誤りや消失に耐性を

持たなければならず，ネットワークの高速化によって通信データが増加するため，通信システムではこれまで以上に通信品質の向上（誤り率の低減）が要求されている．

誤り訂正符号は送信する側でデータに冗長を付加することによって，通信路で生じた誤りを受信側で訂正可能にする符号化技術である．その一つである低密度パリティ検査（Low Density Parity Check：LDPC）符号は Shannon 限界（Shannon の第二定理で示されている通信路容量の限界）に近い特性を示す符号として注目されている．LDPC 符号はブロック符号に分類され，疎な（行列内の 1 の密度が低い）パリティ検査行列によって設計される線形ブロック符号である．LDPC 符号の設計では，検査行列内に 0，1 をランダムに配置し，その配置を最適化することで，Shannon 限界に近い特性を持つ LDPC 符号が得られる．

一方，衛星通信や宇宙通信などの高信頼性が要求される通信システムでは，畳込み符号と呼ばれる種類の誤り訂正符号が古くから利用されている．畳込み符号の符号器はシフトレジスタとモジュロ 2 加算器で構成され，レジスタに保持される過去 m ビットの入力系列が現在の符号語生成に影響する点がブロック符号と大きく異なる（レジスタ数 m をメモリサイズと呼ぶ）．このような符号構造によって回路規模と符号化演算コストを抑えて良い符号が設計できる．LDPC 符号と同様に，畳込み符号のパリティ検査行列を低密度化して設計する LDPC 畳込み符号が提案され，復号シミュレーションによって LDPC 符号と同等の能力を持つことが示されている．しかしながら，LDPC 畳込み符号の理論的な性能解析に関しては十分な研究がなされておらず，性能の良い符号の設計方法も具体的には示されていない．LDPC 畳込み符号の性能解析は符号器にレジスタを有している点で LDPC 符号より難しくなる．

２．研究の目的

LDPC 畳込み符号の能力は一般に復号シミュレーションによって評価されている．シミュレーションに頼ることなく理論的な能力評価を行う場合，LDPC 畳込み符号の最小自由距離 d_{free} とその周辺の重み分布（あるハミング重みを有する符号語の数の分布）から計算できる復号誤り確率が利用可能である．しかしながら，LDPC 畳込み符号が能力を十分に発揮するためにはメモリサイズを非常に大きくしなければならず，重み分布を求める際の計算量が増加し，評価が困難になる．本研究では，能力評価における上記の問題を解決するために，LDPC 畳込み符号の重み分布を高速に計算する手法を開発する．開発する方法では，LDPC 畳込み符号のパリティ検査行列の特徴と符号構造を利用して計算量を削減する．本手法は復号シミュレーションによって評価されているようなメモリサイズが大きい LDPC 畳込み符号を想定しており，現実的な計算時間で結果が得られるようにする．さらに，この結果から性能の良い LDPC 畳込み符号の設計法の指針を与える．

３．研究の方法

(1) LDPC 畳込み符号の重み分布計算法の開発

①重み分布を求める木探索アルゴリズム

LDPC 畳込み符号の重み分布計算は，符号語を表現する符号木を構成し，それを探査する木探索アルゴリズムにより実現できる．よって，LDPC 畳込み符号を定義する検査行列から符号木を構成し，計算対象とする最小自由距離 d_{free} の周辺の重み分布に含まれる符号語を数え上げる木探索アルゴリズムを開発する．畳込み符号の重み分布計算法に関しては従来までに種々提案されており，その方法では生成行列から符号木を構成して符号語を探索する．LDPC 畳込み符号の検査行列から生成行列に変換し適用することも可能であるが，メモリサイズが大きい場合計算量が増加し，計算が困難になる．本研究では，検査行列に対する木探索アルゴリズムを考え，検査行列の低密度性と畳込み符号の符号構造の両面から，重み分布計算の効率化を検討する．

②疎な検査行列の構造に着目した木探索アルゴリズムの高速化

符号化率$R = k/n$の LDPC 畳込み符号の検査行列から符号木を構成した場合，木の分岐数が 2^nになるため，生成行列から符号木を構成した場合の分岐数 2^kに比べて多く，手続きが複雑になる．そこで，疎な検査行列の特徴に着目し，分岐数を削減する方法を与える．

③前方探索と後方探索を組み合わせた重み分布計算法の高速化

畳込み符号の重み分布計算では，符号 C と逆符号$\tilde{C}$の関係に基づき，前方探索と後方探索を組み合わせることによって計算量を削減できる．検査行列から構成する符号木を用いて実行する木探索アルゴリズムにおいて，前方探索と後方探索を組み合わせた重み分布計算法を具体的に与え，計算量を削減する．

(2) 重み分布計算法による LDPC 畳込み符号の能力評価

開発した重み分布計算法を，いくつかの LDPC 畳込み符号に適用し，重み分布を求めることによって，その符号の復号能力を評価する．また，その結果より，LDPC 畳込み符号の設計方法の指針を与える．

４．研究成果

(1) LDPC 畳込み符号の重み分布計算法の開発

①重み分布を求める木探索アルゴリズム

次の多項式表現のパリティ検査行列で定義される LDPC 畳込み符号を例にして説明する．

$$\mathbf{H}_1(D) = \begin{bmatrix} 1 & D & D^3 \\ D^3 & D^2 & 1 \end{bmatrix}$$

これをスカラー表現の検査行列で表すと次のようになる．

$$\mathbf{H}_1 = \begin{bmatrix} 100 & & & \\ 001 & & & \\ 010 & 100 & & \\ 000 & 001 & & \\ 000 & 010 & 100 & \\ 010 & 000 & 001 & \\ 001 & 000 & 010 & \ddots \\ 100 & 010 & 000 & \\ & 001 & 000 & \ddots \\ & 100 & 010 & \\ & & 001 & \ddots \\ & & 100 & \\ & & & \ddots \end{bmatrix}$$

この検査行列は符号化率$R = k/n = 1/3$，メモリサイズ3のLDPC畳込み符号を構成する．一般に，LDPC 畳込み符号の符号語は木で表現され，重み分布計算は木探索として扱うことができる．時刻 t までの符号語の部分系列

$$\mathbf{c}_{[1,t]} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_t]$$

$$\mathbf{c}_i = (c_{i,0}\ c_{i,1}\ \dots\ c_{i,n})$$

に対してシンドロームは次のように計算できる．

$$\mathbf{s}_{[0,t+m-1]} = [\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{t+m-1}] \triangleq \mathbf{c}_{[1,t]}\mathbf{H}_{[1,t]}^T$$

$$\mathbf{s}_i = (s_{i,0}\ s_{i,1}\ \dots\ s_{i,n-k})$$

時刻 t において内部状態としてメモリに持つシンドロームの部分系列は次のようになる．

$$\mathbf{s}^{(t)} = [\mathbf{s}_0^{(t)}, \mathbf{s}_1^{(t)}, \dots, \mathbf{s}_m^{(t)}] \triangleq [\mathbf{s}_{t-1}, \mathbf{s}_t, \dots, \mathbf{s}_{t+m-1}]$$

$$\mathbf{s}_i^{(t)} = (s_{i,1}^{(t)}, s_{i,2}^{(t)}, \dots, s_{i,n-k}^{(t)})$$

LDPC 畳込み符号の符号木では，深さ t のノードは内部状態として$\mathbf{s}^{(t)}$と符号語の部分系列$\mathbf{c}_{[1,t]}$の重み w_t を持つ．深さ t - 1 のノードから $\mathbf{c}_t$の枝を通って深さ tのノードへ進む場合，内部状態は次のように更新される．

$$\mathbf{s}^{(t)} := [\mathbf{s}_1^{(t-1)}, \dots, \mathbf{s}_m^{(t-1)}, \mathbf{0}] + \mathbf{c}_t \begin{bmatrix} \mathbf{H}_0 \\ \vdots \\ \mathbf{H}_m \end{bmatrix}^T \quad (1)$$

$$w_t := w_{t-1} + W_h(\mathbf{c}_t) \quad (2)$$

ただし，$W_h(\mathbf{c}_t)$は$\mathbf{c}_t$のハミング重みである．

LDPC 畳込み符号の重み分布 A_w を $w = d_{\text{free}}$ から $w = d_{\text{free}}+N$ まで求めるには，符号木を $w_t \le d_{\text{free}} + N$の範囲で探査し，符号語を数え上げることで実行できる．その木探索アルゴリズムを以下に示す．

<u>木探索アルゴリズム</u>

入力： 検査行列 $\mathbf{H}$，重み分布の計算範囲 N

出力： 重み分布$A_{d_{\text{free}}}, A_{d_{\text{free}}+1}, \dots, A_{d_{\text{free}}+N}$

初期化：$t := 1, d_{\text{free}} := \infty, A_w := 0 (w = 0,1, \dots)$

Step1. 全ての $\mathbf{c}_t$ に対して，式(1)，(2)を用いて$\mathbf{s}^{(t)}$と w_t を計算する．

Step2. それぞれの $\mathbf{c}_t$ から得られたノード $(\mathbf{s}^{(t)}, w_t)$に対して以下を実行する．

Step2.1 $\mathbf{s}^{(t)} = 0$ならば$\mathbf{c}_{[1,t]}$は符号語であり，$A_{w_t} := A_{w_t} + 1$として符号語数をカウントする．さらに，$w_t < d_{\text{free}}$ならば $d_{\text{free}} := w_t$として最小自由距離を更新する．

Step2.2 $\mathbf{s}^{(t)} \neq 0$ または$w_t > d_{\text{free}} + N$ ならば，現ノードへの枝を打ち切り，現ノード以降の探索を行わない．$\mathbf{s}_i^{(t)}$のいずれかが非零であり，かつ$w_t < d_{\text{free}} + N$ならば，次の深さのノードに進むため，$t := t + 1$として Step1 へ戻る．

②疎な検査行列の構造に着目した木探索アルゴリズムの高速化

LDPC 畳込み符号の木探索では，検査行列から符号木を構成するため，$\mathbf{c}_t$ の選び方によって符号語にならないパスを生成する．すなわち，$\mathbf{s}_0^{(t)} \neq 0$となる $\mathbf{c}_t$の選ぶ場合がある．符号化率$R = k/n$の場合 $\mathbf{c}_t$の選び方は 2^n 通りあり，木探索のコストが大きくなる．以下に，符号語にならないパスの生成を省き，符号木を 2^k 分木にする方法を説明する．

時刻 m まで生成した符号語の部分系列 $\mathbf{c}_{[1,m]}$に対して，検査行列から計算したシンドローム$\mathbf{s}_{[0,2m-1]}$を考える．検査行列内の非零要素の配置から，符号語の要素とシンドロームの要素の間には次の関係式が成り立つ．

$$s_{0,1} = \mathrm{c}_{1,1}, \quad s_{0,2} = \mathrm{c}_{1,3}, \quad s_{1,2} = \mathrm{c}_{2,3}$$

$$s_{1,1} = \mathrm{c}_{1,2} + \mathrm{c}_{2,1}, \quad s_{2,2} = \mathrm{c}_{1,2} + \mathrm{c}_{3,3}$$

ここで$\mathbf{s}^{(0)} = 0$であることから，次の関係式が成り立つ．

$$\mathrm{c}_{1,1} = 0, \quad \mathrm{c}_{1,3} = 0, \quad \mathrm{c}_{2,3} = 0$$

$$\mathrm{c}_{2,1} = \mathrm{c}_{1,2}, \quad \mathrm{c}_{3,3} = \mathrm{c}_{1,2}$$

上式は時刻 1 の場合を考え，$\mathbf{c}_{[1,m]}$と$\mathbf{s}^{(0)}$の関係を示しているが，同様の関係が任意の時刻 t においても$\mathbf{c}_{[1,t+m]}$と内部状態$\mathbf{s}^{(t-1)}$の間に成り立つ．

$$\mathrm{c}_{t,1} = s_{0,1}^{(t-1)},\ \mathrm{c}_{t,3} = s_{0,2}^{(t-1)},\ \mathrm{c}_{t+1,3} = s_{1,1}^{(t-1)} \quad (3)$$

$$\mathrm{c}_{t+1,1} = s_{1,1}^{(t-1)} + \mathrm{c}_{t,2},\ \mathrm{c}_{t+2,3} = s_{2,2}^{(t-1)} + \mathrm{c}_{t,2} \quad (4)$$

式(3)はシンドロームのみによって定まることから，時刻 t の $\mathbf{c}_t$の選び方に依存しない．

また, 式(4)は時刻 t の $\mathbf{c}_t$ の選び方によって $\mathbf{c}_{t+1}$ と $\mathbf{c}_{t+2}$ の要素を決定する.

符号化率$R = k/n$, メモリサイズ m の LDPC 畳込み符号の場合, m が大きくなると式(3)のようなシンドロームのみによって符号語の要素を決定する関係式の数は多くなる. ただし, 初期値が$\mathbf{s}^{(0)} = 0$であることから, これらの関係式はその符号語の要素を 0 に定めるのみである. 一方, 式(4)のような時刻 t の符号語 $\mathbf{c}_t$の選び方によって時刻$t+1$以降の符号語の要素を決定する. そのような関係式の数は n - k である. この n - k 個の関係式を用いて時刻 t + 1 以降の符号語の要素を決定しながら木探索を進めることによって, 符号語にならないパスの生成を省くことができる. 図 1 に, そのようにして構成した符号木を示す. 図 1 では, 式(3), (4)を用いて符号語の要素を決定しながら, $w_t > 6$ を枝刈り条件として木探索アルゴリズムを実行した結果である.

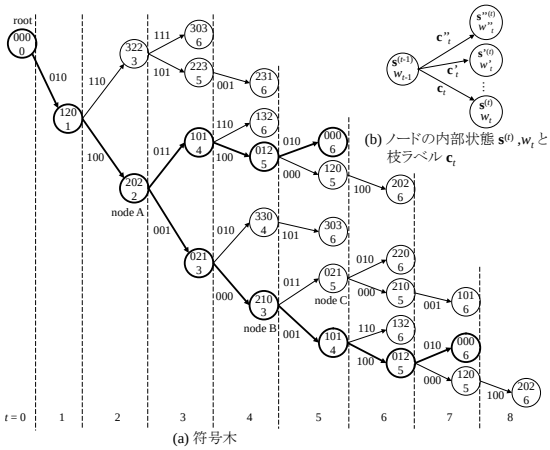

図 1　LDPC 畳込み符号の符号語の木表現

③前方探索と後方探索を組み合わせた重み分布計算法の高速化

$\mathbf{H}$ の要素を上下左右逆順に並べて得られる検査行列$\tilde{\mathbf{H}}$を考える. この検査行列で定義される符号の符号語は, 元符号 C の符号語を逆順に並べた系列になる. よって, $\tilde{\mathbf{H}}$ から構成される$\tilde{C}$の符号木では, 符号語は元符号 C の符号語を終端から生成していることに相当する. このことから, LDPC 畳込み符号の重み分布を求めるためには, $\mathbf{H}$ から構成させる C の符号木, $\tilde{\mathbf{H}}$から構成される$\tilde{C}$の符号木, いずれによっても探索可能である. C の符号木を用いた木探索を前方探索, $\tilde{C}$の符号木を用いた木探索を後方探索と呼ぶ.

<u>前方探索と後方探索を組み合わせた重み分布計算法</u>

入力：検査行列 $\mathbf{H}$, 重み分布の計算範囲 N, 後方探索を行うノードの最小重み$\tilde{w}_{\min}$

初期化： $d_{\text{free}} := \infty, A_w := 0 (w = 0,1, \dots)$

Step1.　後方探索によって$\tilde{C}$の符号木を$\tilde{w}_t \geq \tilde{w}_{\min}$になるまで構成し, その葉ノードの内部状態$\tilde{\mathbf{s}}^{(t)}$とパス$\tilde{\mathbf{c}}_{[1,t]}$を記憶する.

Step2.　前方探索によってCの符号木を構成していき, 各ノードにおいて次の処理を行う.

Step2.1　現ノードの内部状態$\mathbf{s}^{(t)}$が後方探索で記憶したノードの内部状態$\tilde{\mathbf{s}}^{(t)}$と対応した場合, 前方探索のパス$\mathbf{c}_{[1,t]}$と後方探索で記憶したパス$\tilde{\mathbf{c}}_{[1,t]}$の連結が符号語になるため, $A_{w_t} := A_{w_t} + 1$として符号語をカウントする. さらに, $w_t < d_{\text{free}}$ならば$d_{\text{free}} := w_t$として最小自由距離を更新する.

Step2.2　$w_t > d_{\text{free}} + N - \tilde{w}_{\min}$ならば, 現ノードへつながる枝を打ち切り, 現ノード以降の探索を行わない.

図 2 は後方探索で構成させる$\tilde{C}$の符号木を表している. 図 2 では, $\tilde{w}_{\min} = 3$の場合を示しており, $\tilde{w}_t \geq \tilde{w}_{\min}$となるまで構成した符号木の範囲を灰色で表している. Step1 の後方探索では二重丸で表しているノードの内部状態とパスを記憶する. この記憶した内部状態を前方探索に用いることで, Step2 では枝刈り条件を$w_t > d_{\text{free}} + N - \tilde{w}_{\min}$として木探索アルゴリズムを実行できる.

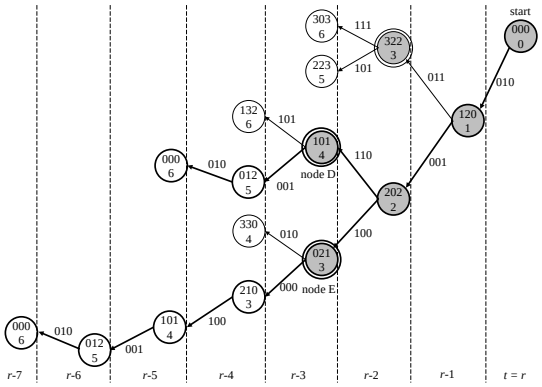

図 2　後方探索で構成される$\tilde{\mathbf{C}}$の符号木

(2) 重み分布計算法による LDPC 畳込み符号の能力評価

$\mathbf{H}_1(D)$および以下の検査行列で定義される LDPC 畳込み符号の重み分布を求めた結果を表 1 に示す.

・符号化率 2/5, メモリサイズ 21

$$\mathbf{H}_2(D) = \begin{bmatrix} D & D^2 & D^4 & D^8 & D^{16} \\ D^5 & D^{10} & D^{20} & D^9 & D^{18} \\ D^{25} & D^{19} & D^7 & D^{14} & D^{28} \end{bmatrix}$$

・符号化率 2/5, メモリサイズ 57

$$\mathbf{H}_2(D) = \begin{bmatrix} D & D^9 & D^{20} & D^{58} & D^{34} \\ D^{13} & D^{56} & D^{16} & D^{22} & D^{15} \\ D^{47} & D^{57} & D^{25} & D^{42} & D^{12} \end{bmatrix}$$

これらの重み分布を求める際，提案手法のパラメータは次のようにした．
・$\mathbf{H}_1(D)$: $l = 10,\ \tilde{w}_{\min} = 3$
・$\mathbf{H}_2(D)$: $l = 6,\ \tilde{w}_{\min} = 14$
・$\mathbf{H}_3(D)$: $l = 2,\ \tilde{w}_{\min} = 6$
表 2 に木探索アルゴリズムと前方探索と後方探索を組み合わせた重み分布計算法で探索したノード数を示す．表 2 から，提案手法によって計算コストを削減することに成功している．

表 1　LDPC 畳込み符号の重み分布

	d_{free}	A_w ($w = d_{\text{free}}, d_{\text{free}}+1,\ldots,d_{\text{free}}+N$)
$\mathbf{H}_1(D)$	6	2, 0, 1, 0, 5, 0, 15, 0, 27, 0, 98
$\mathbf{H}_2(D)$	24	6, 0, 6, 0, 8, 0, 45
$\mathbf{H}_3(D)$	24	5, 0, 0

表 2　探索ノード数

	木探索アルゴリズム	前方探索と後方探索を組み合わせた重み分布計算法
$\mathbf{H}_1(D)$	3,082	620
$\mathbf{H}_2(D)$	19,707,326,406	67,349,483
$\mathbf{H}_3(D)$	−	90,865,559,756

次に，求めた重み分布から復号誤り確率を評価する．ここでは，符号化率が等しい $\mathbf{H}_2(D)$ と $\mathbf{H}_3(D)$を比較する．AWGN 通信路における LDPC 畳込み符号の復号誤り確率は，ブロック符号と同様に次式で評価できる．

$$P_E \le \frac{1}{2}\sum_{w=d_{\text{free}}}^{\infty} A_w Q\left(\sqrt{2wR \cdot E_b/N_0}\right)$$

$$Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^{\infty} \mathrm{e}^{-\frac{z^2}{2}} dz$$

ただし，E_b/N_0は AWGN 通信路の SN 比である．LDPC 畳込み符号に適用する反復復号法の特性は高 SN 比領域において上式の誤り確率に近づく．$E_b/N_0 = 3.0$ [dB]において，$\mathbf{H}_2(D)$の LDPC 畳込み符号の復号誤り確率は 1.414×10^{-11}であり，$\mathbf{H}_3(D)$の LDPC 畳込み符号の復号誤り確率は1.129×10^{-11}である．よって，$\mathbf{H}_3(D)$で定義される LDPC 畳込み符号は $\mathbf{H}_2(D)$より復号能力が高いことが分かる．

５．主な発表論文等
（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計 2 件）

① M. Hirotomo, M. Mohri, and M. Morii, "A reliability-based computation method for the weight distribution of LDPC codes using the probabilistic algorithm," Proc. 2011 IEEE International Symposium on Information Theory (ISIT2011), pp.361–365, 2011.（査読有）
② M. Hirotomo and M. Morii, "Detailed evaluation of error floors of LDPC codes using the probabilistic algorithm," Proc. 2010 International Symposium on Information Theory and its Applications (ISITA2010), pp.513-518, 2010.（査読有）

〔学会発表〕（計 5 件）

① 廣友雅徳, “LDPC 符号の性能評価とその手法，” 防衛省情報本部 解析技法研究会招待講演，2012 年 2 月，東京都.
② 廣友雅徳，森井昌克，“LDPC 畳込み符号の重み分布計算のための木探索アルゴリズムについて，” 第 34 回情報理論とその応用シンポジウム（SITA2011），pp.69-74, 2011 年 12 月，岩手県.
③ 廣友雅徳，森井昌克，“LDPC 畳込み符号の重み分布計算法の高速化，” 電子情報通信学会情報理論研究会，IT2011-26, pp.15-20，2011 年 9 月，東京都.
④ 廣友雅徳，森井昌克，“LDPC 畳込み符号の重み分布計算法について，”第 33 回情報理論とその応用シンポジウム（SITA2010），pp.444-449，2010 年 12 月，山口県.
⑤ 廣友雅徳，森井昌克，“LDPC 畳込み符号の重み分布計算について，” 電子情報通信学会情報理論研究会，IT2010-35，pp.7-11，2010 年 9 月，宮城県.

〔図書〕（計 0 件）

〔産業財産権〕
○出願状況（計 0 件）
○取得状況（計 0 件）

〔その他〕
なし

６．研究組織
(1)研究代表者

廣友　雅徳（HIROTOMO MASANORI）
佐賀大学・総合情報基盤センター・特任助教
研究者番号：00529795

(2)研究分担者
なし

(3)連携研究者
なし