

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5199255号
(P5199255)

(45) 発行日 平成25年5月15日(2013.5.15)

(24) 登録日 平成25年2月15日(2013.2.15)

(51) Int.Cl. F I
HO3M 13/19 (2006.01) HO3M 13/19

請求項の数 17 (全 32 頁)

(21) 出願番号	特願2009-521250 (P2009-521250)	(73) 特許権者	508367728
(86) (22) 出願日	平成19年7月25日 (2007.7.25)		コミッサリヤ ア レネルジ アトミック
(65) 公表番号	特表2009-545220 (P2009-545220A)		フランス国, 75015 パリ, “ル・ポ
(43) 公表日	平成21年12月17日 (2009.12.17)		ナン・デー” パティモン リュ・ルブラ
(86) 国際出願番号	PCT/EP2007/057644		ン 25
(87) 国際公開番号	W02008/012318	(74) 代理人	100070150
(87) 国際公開日	平成20年1月31日 (2008.1.31)		弁理士 伊東 忠彦
審査請求日	平成22年7月5日 (2010.7.5)	(74) 代理人	100091214
(31) 優先権主張番号	06/53148		弁理士 大貫 進介
(32) 優先日	平成18年7月27日 (2006.7.27)	(74) 代理人	100107766
(33) 優先権主張国	フランス (FR)		弁理士 伊東 忠重
		(72) 発明者	サヴィン, ヴァレンティン
			フランス国, 38000 グルノーブル
			アヴェニュー・サン・ロッシュ 9

最終頁に続く

(54) 【発明の名称】 近傍信頼性に依存するスケジューリングを用いたメッセージ・パッシングによる復号方法

(57) 【特許請求の範囲】

【請求項 1】

複数の変数ノードおよび複数のチェック・ノードを有する二部グラフによって表現できる誤り訂正符号の復号のためのメッセージ・パッシングによる逐次反復式方法であって、複数の復号反復工程の各反復工程について：

・変数ノードまたはチェック・ノードが、これらのノードの近傍 ($V_n^{(d)}$ 、 $V_m^{(d)}$) において利用可能な復号情報の対応する信頼度に応じて分類され (720)、ノードの近傍において利用可能な復号情報の前記信頼度は、前記二部グラフにおいてこのノードから1より大きな所定の距離 (d) 以内にある全ノードに存在する、またはそのようなノードによって送信もしくは受信された情報の信頼性の指標

【数 5 1】

$$(f_a(n), \tilde{f}_a(m))$$

から得られ、信頼度の高いノードが信頼度の低いノードより前に分類され；

・こうして分類された各ノードが (725)、前記分類によって定義された順序で、少なくとも一つのメッセージ (m_n 、 m_n) を隣接ノードに渡す、方法。

【請求項 2】

請求項 1 記載の復号方法であって、前記ノードの分類はさらに、こうして得られた該ノ

ードのそれぞれの信頼性の指標の値をソートすることによって行われる、復号方法。

【請求項 3】

請求項 2 記載の復号方法であって、前記複数の反復工程の各反復工程について、分類されたノードは、前記分類によって定義された順序で逐次的に処理され、分類された各ノードについて (8 2 5)、それに隣接するノードに宛てられたメッセージが計算され (8 3 5)、前記各隣接ノードについて、該隣接ノードに隣接するノードへのメッセージが計算される (8 3 0)、復号方法。

【請求項 4】

請求項 1 記載の復号方法であって、前記ノードの分類はさらに、前記指標の値の諸区間において該ノードをグループ化することによって行われる、復号方法。

10

【請求項 5】

前記信頼性指標が整数値を使い、前記各整数値について、信頼性指標がこの値に等しくなるノードのインデックスが、それに関連付けられたメモリ・ゾーンに記憶される、請求項 4 記載の復号方法。

【請求項 6】

請求項 4 または 5 記載の復号方法であって、前記複数の反復工程の各反復工程について、ノード・グループは前記分類によって定義された順序で逐次的に処理され、各ノード・グループ (G_j) について、そのグループ内のノードに隣接するノードへのメッセージが計算され (9 3 5)、前記各隣接ノードについて、それ自身が該隣接ノードに隣接するノードへのメッセージも計算される、復号方法。

20

【請求項 7】

請求項 1 ないし 6 のうちいずれか一項記載の復号方法であって、前記二部グラフにおける各変数ノードについて、前記複数の反復工程の各反復工程は、このノード中にすでに存在する事前情報 (α_n) および隣接チェック・ノードからこのノードによって受信されるメッセージ (β_{mn}) の関数として事後情報

【数 5 3】

$$(\hat{\alpha}_n)$$

を計算するステップ (7 4 0、8 4 0、9 4 0) をも有する、復号方法。

30

【請求項 8】

前記の事後情報計算ステップに続いて、前記変数のハードな値

【数 5 4】

$$(\bar{\alpha}_n)$$

についての決定ステップ (7 5 0、8 5 0、9 5 0) がある、請求項 7 記載の復号方法。

【請求項 9】

請求項 8 記載の復号方法であって、次のステップ (7 6 0、8 6 0、9 6 0) が、前記のようにして得られた変数のハードな値が、グラフ中のすべてのチェック・ノードに関連付けられたパリティ・チェックを満たすかどうかを試験し、もし満たせば、前記ハードな値によって構成される語が復号された語として提供される、復号方法。

40

【請求項 10】

請求項 1 ないし 9 のうちいずれか一項記載の復号方法であって、前記分類は、所定数の復号反復工程 ($Iter_f$) の後に中断され、当該復号方法は次いでその復号反復工程を、前記変数ノードまたは前記チェック・ノードの前記分類なしに続ける、復号方法。

【請求項 11】

請求項 7 記載の復号方法であって、前記分類は、前記変数の事後の値と事前の値との間の差の絶対値の最小値が所定の閾値 (T_f) より大きい場合、中断され、その際、当該復号方法は、その復号反復工程を、前記変数ノードまたは前記チェック・ノードの前記分類な

50

しに続ける、復号方法。

【請求項 1 2】

前記誤り訂正符号がターボ符号である、請求項 1 ないし 1 1 のうちいずれか一項記載の復号方法。

【請求項 1 3】

前記誤り訂正符号が、N個の変数ノードおよびM=N-K個のチェック・ノードをもつ二部グラフによって表現されるLDPC符号(K,N)である、請求項 1 ないし 1 1 のうちいずれか一項記載の復号方法。

【請求項 1 4】

請求項 1 3 記載の復号方法であって、インデックスm {1, ..., M}をもつチェック・ノードからインデックスn {1, ..., N}をもつ変数ノードへの α_{mn} メッセージが次のように計算され：

【数 5 5】

$$\beta_{mn} = \left(\prod_{n' \in H(m) - \{n\}} \text{sgn}(\alpha_{mn'}) \right) \Phi \left(\sum_{n' \in H(m) - \{n\}} \Phi(|\alpha_{mn'}|) \right)$$

ここで、 α_{mn} はインデックスn'をもつ変数ノードからインデックスmをもつチェック・ノードへのメッセージを表し、H(m)はインデックスmをもつチェック・ノードに隣接するすべての変数ノードを表し、xが正であれば $\text{sgn}(x) = 1$ 、そうでなければ $\text{sgn}(x) = -1$ であり、

【数 5 6】

$$\Phi(x) = \ln \left(\frac{e^x + 1}{e^x - 1} \right)$$

である、復号方法。

【請求項 1 5】

請求項 1 3 記載の復号方法であって、インデックスm {1, ..., M}をもつチェック・ノードからインデックスn {1, ..., N}をもつ変数ノードへのメッセージ α_{mn} は次のように計算され：

【数 5 7】

$$\beta_{mn} = \left(\prod_{n' \in H(m) - \{n\}} \text{sgn}(\alpha_{mn'}) \right) \cdot \min_{n' \in H(m) - \{n\}} (|\alpha_{mn'}|)$$

ここで、 α_{mn} はインデックスn'をもつ変数ノードからインデックスmをもつチェック・ノードへのメッセージを表し、H(m)はインデックスmをもつチェック・ノードに隣接するすべての変数ノードを表し、xが正であれば $\text{sgn}(x) = 1$ 、そうでなければ $\text{sgn}(x) = -1$ である、復号方法。

【請求項 1 6】

請求項 5 および 1 5 に記載の復号方法であって、前記分類はチェック・ノードに適用され、前記所定の距離は2に等しく、インデックスmをもつチェック・ノードの信頼性指標が次のように計算され：

【数 5 8】

$$\tilde{f}_2(m) = c + \sum_{n \in H(m)} \min(1, \text{Card}\{m' \in H(n) - \{m\} | c_{m'} = -1\})$$

10

20

30

40

50

ここで、 $H(m)$ はインデックス m をもつチェック・ノードに隣接する変数ノードの集合を表し、 $H(n)$ はインデックス n をもつ変数ノードに隣接するチェック・ノードの集合を表し、 $\text{Card}(\cdot)$ はある集合のカーディナル数を表し、 $c_m = +1$ であれば $c = 0$ 、 $c_m = -1$ であれば $c = m_{\text{ax}} + 1$ であり、ここで m_{ax} は二分グラフ中のチェック・ノードの最大次数であり、 $c_m = +1 / c_m = -1$ は、インデックス m をもつチェック・ノードについてパリティ・チェックがそれぞれ満たされている / 満たされていないことを意味する、復号方法。

【請求項 17】

コンピュータによって実行されたときに、請求項 1 ないし 16 のうちいずれか一項記載の復号方法のステップを実装するよう適応されたソフトウェア手段を有するコンピュータ・プログラム。

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、遠隔通信またはデータ記録の分野における誤り訂正符号の復号に関する。より詳細には、本発明は、LDPC (Low Density Parity Check [低密度パリティ・チェック]) 符号またはターボ符号 (turbo codes) のような二部グラフ (bipartite graph) において表示できる誤り訂正符号の復号のためのメッセージ・パッシング (message passing) による逐次反復方法に関する。

【背景技術】

【0002】

20

二部グラフによって表示できる誤り訂正符号は幅広い多様な符号をカバーしており、具体的には、非特許文献 1 の論文において最初に記載され、最近興味深い属性が再発見された LDPC 符号や画期的な論文である非特許文献 2 によって導入されたターボ符号がある。

【0003】

二部グラフは、あらゆるノードが二つの別個の部分集合をなし、単一の部分集合内のどの二つのノードもグラフのエッジによって接続されていない非有向グラフである。

【0004】

いくつかの誤り訂正符号が二部グラフとして表示できる。グラフは、符号語を形成するシンボルに関連付けられたノードからなる第一の部分集合と、符号制約条件、典型的にはパリティ・チェックに関連付けられたノードからなる第二の部分集合とに分割される。制約条件のグループに関連付けられた二部グラフはタナー・グラフ (Tanner graph) とも呼ばれる。

30

【0005】

符号語中のシンボルは通例、ガロア体の元 $F_2 = \{0, 1\}$ 、つまりビットであるが、より一般には、任意の標数 2 をもつ体

【0006】

【数 1】

$$F_{2^p}$$

40

の、よって 2^p 進アルファベットの元であってもよい。以下では、一般性を失うことなく、 $p=1$ の場合、つまり二進符号に限定する。

【0007】

二部グラフによる追加的な表現符号は、逐次反復的なメッセージ・パッシング (MP) または BP (Belief Propagation [信頼伝搬]) 復号を使って復号できる。この復号方法についての一般的な記述は非特許文献 3 によって与えられている。MP 型逐次反復式復号は実際には、復号の分野でよく知られたアルゴリズム、すなわちターボ符号に使われる「前方 - 後方 (forward-backward)」アルゴリズムおよび LDPC 符号のためのギャラガー・アルゴリズムの一般化である。

50

【 0 0 0 8 】

簡単のため、以下には、LDPC符号の枠組みにおけるメッセージ・パッシングによる逐次反復復号の原理の記載が含まれる。線形符号(K,N)を考える。ここで、Kはいくつかの情報ビットを表す符号の大きさ(dimension)であり、Nは前記いくつかの符号化されたビットを表す符号の長さである。M=N-Kはパリティ・ビットの数、すなわちパリティ制約条件の数に等しい。

【 0 0 0 9 】

図1は、線形符号(K,N)のタナー・グラフを示している。グラフの左側は、符号中のビットに対応する、「変数(variable)」型ノードまたは単に「変数」とも呼ばれるノードを示し、右側は、「チェック(check)」型ノードまたはより単純に「チェック」とも呼ばれるパリティ・チェックに対応するノードを示している。このグラフの結合行列は、大きさM×Nをもつ前記符号のパリティ行列に対応する。よって、この二部グラフは、N個の「変数」型ノードおよびM個の「チェック」型ノードを有し、変数ノードnがチェックノードmに接続されるのは $h_{mn}=1$ である場合でありかつその場合に限られる。たとえば、図1におけるグラフは符号(10,5)に対応し、次のパリティ行列

【 0 0 1 0 】

【 数 2 】

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \quad (1)$$

をもつ。

【 0 0 1 1 】

一般に、線形符号は二進値の要素をもつ生成行列Gによって定義され、符号語 $x = (x_1, x_2, \dots, x_N)$ は、情報ビット $a = (a_1, a_2, \dots, a_k)$ をもつ語から

$$x = aG \quad (2)$$

によって得られることを想起されたい。すべての符号語はパリティ・チェックを満たすので、関係

$$H \cdot G^T = 0 \quad (3)$$

が得られる。ここで、 G^T は行列Gの転置を表す。

【 0 0 1 2 】

符号語xは通信チャネル上で伝送されたり、データ担体に記録されたりする。xのノイズ入りバージョン、すなわち $y = (y_1, y_2, \dots, y_N)$ が受信に際して、あるいは担体を読むときに回復される。復号動作は、観察値yから出発してxを、したがってaを見出すことである。

【 0 0 1 3 】

逐次反復的メッセージ・パッシング復号の原理を記述する前に、以下の記号法について取り決めておく。

【 0 0 1 4 】

$H(n)$ は、二部グラフ中の変数nに関係したすべての試験、つまりノードnに隣接するすべてのノードを表す。

【 0 0 1 5 】

$H(m)$ は、二部グラフ中のチェックmに接続された変数の集合、つまりノードmに隣接する

すべてのノードを表す。

【 0 0 1 6 】

p_n は二部グラフ中の変数 n に関する事前 (a priori) 情報、つまり符号語中の n 番目のビットに関する事前情報を表す。この情報が、受信される信号および伝送チャネルの特性を取り入れるものである。これはデコーダへの入力をなし、通例、ソフトな値の形で、すなわち確率を用いて復調器によって与えられる：

$$p_n = (p_n^0, p_n^1) \quad (4)$$

ここで、 $p_n^a = \Pr(x_n = a | y_n)$, $a \in \{0, 1\}$

あるいはより便利には、対数尤度比 (LLR: logarithmic likelihood ratio) の形

【 0 0 1 7 】

【 数 3 】

$$\alpha_n = \ln \left(\frac{\Pr(x_n = 0 | y_n)}{\Pr(x_n = 1 | y_n)} \right) \quad (5)$$

で与えられる。こうして、中央化された (centred) ガウス型ホワイトノイズおよびBPSK 20
変調については、復調器は単に

$$z_n = (2/\sigma^2) y_n \quad (6)$$

を計算する。ここで、 σ^2 はノイズ分散である。

【 0 0 1 8 】

m_n は、変数 n によってチェック $m \in H(n)$ に伝送されるメッセージを表す。ターボ符号を参照すると、 m_n は外来情報 (extrinsic information) とも呼ばれる。

【 0 0 1 9 】

n_m は対称的にチェック m によって変数 $n \in H(m)$ に伝送されるメッセージを表す。これも外来情報と形容される。

【 0 0 2 0 】

$\hat{\alpha}_n$ 付きの α_n は、変数 n に関係した事後 (a posteriori) 情報を表す。これは、事前情報 α_n と、復号の間に隣接するチェックから変数 n が受信したメッセージ m_n との両方を取り入れている。

【 0 0 2 1 】

α_n 付きの α_n は、ソフト値である $\hat{\alpha}_n$ 付きの α_n に対応するハードな値、つまりビット x_n についてなされた決定である。

【 0 0 2 2 】

図 2 は、メッセージ・パッシングによる逐次反復復号の原理を示している。

【 0 0 2 3 】

ステップ 2 1 0 では、メッセージ m_n は、変数 n およびチェック $m \in H(n)$ の対のそれぞれ 40
について初期化される。メッセージ n_m は、通例、前記の事前情報によって初期化される。すなわち、 $m_n = \alpha_n$, $n_m = \alpha_m$, $m \in H(n)$ 。逐次反復カウンタ (iteration counter) l iter も 0 に初期化される。

【 0 0 2 4 】

この初期化ステップには以下のステップを含む逐次反復ループが続く。

【 0 0 2 5 】

2 2 0 では、チェックが初期化される。より正確には、各チェック m について、チェック m から対応する変数 $n \in H(m)$ へのメッセージ n_m が計算される。すなわち、

$$n_m = F_C(\{\alpha_n \mid n \in H(m) - \{n\}\}) \quad (7)$$

ここで、 F_C はチェック処理関数である。任意の与えられたノードの対 $m, n \in H(m)$ について 50

10

20

30

40

50

、メッセージ m_n は、チェック m 自身が変数 $n \in H(m) - \{n\}$ から受信したメッセージの関数として計算される。結果として、変数ノードから自分自身に回送される外来情報はないことがわかる。チェック処理ステップは水平ステップとも呼ばれる。

【0026】

230では、変数是对称的に処理される。より正確には、各変数 n について、対応するチェック $m \in H(n)$ に向けられたメッセージ m_n が計算される。すなわち、

$$m_n = F_V(\{m_n | m \in H(n) - \{n\}\}) \quad (7')$$

ここで、変数処理関数は F_V と記されている。与えられたノードの対 $n, m \in H(n)$ について、メッセージ m_n は、変数 n 自身がチェック $m \in H(n) - \{n\}$ から受信したメッセージの関数として計算される。よって、上記のように、ノードから自分自身に回送される外来情報はない。変数処理ステップは垂直ステップとも呼ばれる。

10

【0027】

240では、事後情報 (\wedge 付きの α_n) が、事前情報 β_n と、変数 n によってその隣接チェック・ノード $m \in H(n)$ から受信されたメッセージ m_n とから推定される。記号的には

【0028】

【数4】

$$\hat{\alpha}_n = F_{AP}(\{\alpha_n\} \cup \{\beta_{mn} | m \in H(n)\}) \quad (8)$$

20

と表される。ここで、事後推定関数が F_{AP} と表されている。

【0029】

250では、ソフトな値 (\wedge 付きの α_n) からハードな値 (\wedge 付きの $\bar{\alpha}_n$) に対する決定がなされる。すなわち、

【0030】

【数5】

30

$$\bar{\alpha}_n = F_D(\hat{\alpha}_n) \quad (8')$$

ここで、決定関数が F_D で表されている。典型的には、BPSK変調について、ソフト値の符号について決定がされる。すなわち、

【0031】

【数6】

40

$$\bar{\alpha}_n = \text{sgn}(\hat{\alpha}_n)$$

以下では便宜上、ビットの値はその変調された値と同定される。通常、ビット「0」が値「+1」によって表現され、ビット「1」が値「-1」によって表現される。

【0032】

260では、ベクトル

【0033】

50

【数 7】

$$\bar{\alpha} = (\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_N)$$

が符号語かどうか検査される。つまり、前記ベクトルがパリティ・チェックを満たすかどうか検査される。もし満たせば、ステップ 265 でループは終了され、復号された語は

【0034】

【数 8】

$\bar{\alpha}$

となる。もし満たさなければ、反復数がステップ 267 でインクリメントされ、ステップ 270 における比較が、行われた反復数 iter が閾値 iter_max に達したかどうかを判定する。もし達していなければ、逐次反復式復号はステップ 220 にループで戻って継続される。デフォルトでは、復号動作が失敗したと結論され、ループはステップ 275 で終了される。

【0035】

逐次反復ループ中のステップの順序は、図 2 に示された順序とは異なるものであってもよい。特に、変数およびチェックの処理順は逆にでき、メッセージ m_n を $m_n = 0$ 、 $n \in \{1, \dots, N\}$ および $m = H(n)$ と初期化することによって開始することが可能である。

【0036】

図 2 に示される逐次反復符号の原理によれば、すべてのチェックがまず処理され、続いてすべての変数が処理されてもよいし、あるいは上述したように、すべての変数がまず処理されて、続いてすべてのチェックが処理されてもよい。これは、「並列スケジューリング (parallel scheduling)」または「洪水スケジューリング (flooding scheduling)」と称される。文献では他の型のスケジューリングも提案されており、それらは二つの範疇に分類されうる。

【0037】

- シリアル型スケジューリング。「シリアル・スケジューリング (serial scheduling)」、「シャッフルド BP (shuffled-BP)」、「水平シャッフルド (horizontal shuffled)」または「垂直シャッフルド (vertical shuffled)」スケジューリングを含みうる範疇である。シリアル型スケジューリングは、チェックおよび変数に等しく適用可能である。チェックへの適用については、復号は以下の戦略を使う：

- ・まず、単一のチェック m が変数 $n \in H(m)$ に送られるべきメッセージ m_n を計算することによって処理される；

- ・チェック $m' \in H(n) - \{m\}$ に送られるべき各変数 $n \in H(m)$ からのメッセージ m'_n が更新され、送信される。これらの変数に関係した事後情報 (\wedge 付きの m_n) も更新される；

- ・それ以上チェックがなくなるまで、次のチェックが処理され、上記 2 つのステップが反復される。

【0038】

同様に、チェックごとの処理の代わりに、変数ごとの処理を採用することができる。想定される場合に依存して、用語「水平シャッフルド」または「垂直シャッフルド」スケジューリングが使用される。

【0039】

上述した二つのスケジューリング型は、「混合 (mixed)」または「グループ・シャッフルド (group shuffled)」スケジューリングの形で混成されることもできる。混合ス

10

20

30

40

50

ケジューリングに対応するデコード戦略の記述は、非特許文献4の論文に与えられている。戦略は、ノードのグループによる分割に基づき、処理はグループ内では並列、あるグループから次のグループへは直列である。より正確には、チェック・グループの分布について：

- ・チェック $\{m_1, m_2, \dots, m_g\}$ からなる第一のグループGは、 $i = 1, \dots, g$ について変数 n $H(m_i)$ に送られるべきメッセージ $m_{i,n}$ を計算することによって処理される；

- ・ $i = 1, \dots, g$ について対応するチェック $m'_{i,n} = H(n) - \{m_i\}$ に送られるべき各変数 n $H(m_i)$ についてのメッセージ $m'_{i,n}$ が更新され、送信される。これらの変数についての事後情報 (\wedge 付きの n) も更新される；

- ・それ以上チェック・グループがなくなるまで、次のチェック・グループが処理され、上記2つのステップが反復される。

10

【0040】

同様に、処理は、チェックのグループごとの分割ではなく、変数のグループごとの分割に基づくこともできる。

【0041】

図3、図4および図5は、それぞれ並列スケジューリング、シリアル・スケジューリングおよび混合スケジューリングについてのメッセージ・パッシング機構を図的に示している。変数が処理される前にチェックが処理されると想定されている。図3は、チェックの処理における第一のステップ(a)および変数の処理における第一のステップ(b)を示している。図4および図5は、チェックの処理における第一(a)および第二(c)のステップならびに変数の処理における第一のステップ(b)および第二のステップ(d)を示している。図5では、すべてのチェック・ノードは、逐次的に処理される G_1 および G_2 で表される二つのグループに分割されている。

20

【0042】

なお、シリアル・スケジューリングおよび並列スケジューリングは混合スケジューリングの特殊な場合のように考えることもできる。前者は諸グループが単一元に縮小された場合に対応し、後者は一つのグループがすべてのチェック・ノード(および変数ノード)を含む場合に対応する。

【0043】

LDPC符号について、二つの主要な逐次反復メッセージ・パッシング復号アルゴリズムが知られている。すなわち、「log-BP[対数BP]」とも呼ばれるSPA(Sum Product Algorithm[和積アルゴリズム])および「based BP[基礎付きBP]」とも呼ばれる「Min-Sum[最小和]」アルゴリズムである。これら二つのアルゴリズムの詳細な記述は、非特許文献5の論文によって与えられている。

30

【0044】

SPAとMin-Sumアルゴリズムの唯一の相違は、チェック処理ステップにおいてであり、これについては後述する。他のステップは同一である。すなわち：

変数処理ステップ230は次のようにメッセージ $m_{n,n}$ を計算することからなる：

【0045】

【数9】

40

$$\alpha_{mn} = \ln \left(\frac{\Pr(x_n = 0 | y_n, B_{mn}^*, C_{mn})}{\Pr(x_n = 1 | y_n, B_{mn}^*, C_{mn})} \right) \quad (10)$$

ここで、 B_{mn}^* は、コントロール $m'_{i,n} = H(n) - \{m_i\}$ から変数 n によって受信されたすべてのメッセージ $m'_{i,n}$ を表し、 C_{mn} は、これらのチェックのそれぞれについて検証されたパリティ・チェックに対応するイベントを表す。 y_n 値どうしが独立であるとする、 α_{mn} はLLRの形

50

で

【 0 0 4 6 】

【 数 1 0 】

$$\alpha_{mn} = \alpha_n + \sum_{m' \in H(n) - \{m\}} \beta_{m'n} \quad (11)$$

10

のように表せることが示される。

【 0 0 4 7 】

事後情報を推定するステップ 2 4 0 は

【 0 0 4 8 】

【 数 1 1 】

$$\hat{\alpha}_n = \ln \left(\frac{\Pr(x_n = 0 | y_n, B_n, C_n)}{\Pr(x_n = 1 | y_n, B_n, C_n)} \right) \quad (12)$$

20

を計算することからなる。ここで、 B_n はすべての $H(n)$ チェックから変数 n によって受信されるメッセージを表し、 C_n は、これらのチェックのそれぞれについて検証されたパリティ・チェックに対応するイベントを表す。上と同じ想定に基づき、 $\hat{\alpha}_n$ が LLR の形で

【 0 0 4 9 】

【 数 1 2 】

$$\hat{\alpha}_n = \alpha_n + \sum_{m \in H(n)} \beta_{mn} \quad (13)$$

30

のように表せることが示せる。

式 (11) および (13) により、

【 0 0 5 0 】

【 数 1 3 】

40

$$\alpha_{mn} = \hat{\alpha}_n - \beta_{mn} \quad (14)$$

であることが見出される。その結果、変数処理ステップ 2 3 0 は、事後情報の推定後の、反復工程の終わりに置かれてもよい。表式 (14) は、あるノード (m) が外来情報 (この場合 α_{mn}) を自分自身に返さないことを翻訳したものである。

【 0 0 5 1 】

ハード値決定ステップ 2 5 0 は単に：

【 0 0 5 2 】

【数 1 4】

$$\bar{\alpha}_n = \text{sgn}(\hat{\alpha}_n) \quad (15)$$

によって行われる。ここで、 x が正であれば $\text{sgn}(x) = 1$ 、そうでなければ $\text{sgn}(x) = -1$ である。

【0 0 5 3】

ステップ 2 6 0 におけるハード値に対するパリティ・チェックの検証は、パリティ・チェックの計算： 10

【0 0 5 4】

【数 1 5】

$$c_m = \prod_{n \in H(m)} \bar{\alpha}_n, \quad m = 1, \dots, M \quad (16)$$

20

を使う。すべてのパリティ・チェックが満たされるのは：

【0 0 5 5】

【数 1 6】

$$\sum_{m=1}^M c_m = M \quad (17)$$

の場合であって、かつこの場合のみである。

30

【0 0 5 6】

SPAアルゴリズムについては、チェック処理ステップ 2 2 0 は次を計算することからなる。

【0 0 5 7】

【数 1 7】

$$\beta_{mn} = \ln \left(\frac{\Pr(c_m = 1 | x_n = 0, A_{mn}^*)}{\Pr(c_m = 1 | x_n = 1, A_{mn}^*)} \right) \quad (18)$$

40

ここで、 $c_m = 1$ は、あるパリティ条件がチェック m について満足されることを意味し、 A_{mn}^* は、変数 $n' \in H(m) - \{n\}$ からチェック m によって受信されたすべてのメッセージ $x_{n'}$ を表す。 x_n は

【0 0 5 8】

【数 1 8】

$$\beta_{mn} = \left(\prod_{n' \in H(m) - \{n\}} \text{sgn}(\alpha_{mn'}) \right) \cdot \Phi \left(\sum_{n' \in H(m) - \{n\}} \Phi(|\alpha_{mn'}|) \right) \quad (19)$$

によって表せることが示される。ここで、

【0059】

10

【数 1 9】

$$\Phi(x) = \ln \left(\frac{e^x + 1}{e^x - 1} \right) = -\ln \left(\tanh \left(\frac{x}{2} \right) \right)$$

である。

【0060】

Min-Sumアルゴリズムによるチェックの処理は、表式(19)の単純化に対応する。関数 $\Phi(x)$ の速い減衰および $\Phi(x)$ がその逆に等しい、すなわち $\Phi(x) = x$ という事実のため、次のような近似：

【0061】

【数 2 0】

$$\beta_{mn} = \left(\prod_{n' \in H(m) - \{n\}} \text{sgn}(\alpha_{mn'}) \right) \cdot \min_{n' \in H(m) - \{n\}} (|\alpha_{mn'}|) \quad (20)$$

30

をすることが正当化できる。

【0062】

このMin-Sum復号アルゴリズムは、加算、比較および符号変化を実行するだけなので、SPA復号アルゴリズムより著しく単純である。さらに、Min-Sumアルゴリズムの性能はノイズ分散²の推定とは独立である。

【0063】

SPA復号アルゴリズムの性能のほうがMin-Sumアルゴリズムの性能より上ではあるが、SPA復号アルゴリズムの性能は、ノイズ・パワーの推定が悪いとひどく劣化することがありうる。

【先行技術文献】

40

【非特許文献】

【0064】

【非特許文献1】R・ギャラガー (Gallager)、「低密度パリティ・チェック符号 (Low density parity check codes)」、IEEE Trans. Inform. Theory、vol. IT-8、第21-28頁、1962年

【非特許文献2】C・ベルー (Berrou)ら、「最適に近い誤り訂正符号化および復号：ターボ符号 (Near optimum error correcting coding and decoding: turbo-codes)」、IEEE Trans. Inform. Theory、vol. 44、No.10、第1261-1271頁、1996年

【非特許文献3】N・ウィベリ (Wiberg)、「一般グラフに対する符号および復号 (Codes and decoding on general graphs)」、学位論文、1996年

50

【非特許文献4】J・チャン (Zhang) ら、「シャッフルド逐次反復復号 (Shuffled iterative decoding)」、IEEE Trans. on Comm., Vol.53, No.2, 2005年2月、第209-213頁

【非特許文献5】W・E・ライアン (Ryan)、「LDPC符号入門 (An introduction to LDPC codes)」、『記録システムのための符号化および信号処理のCRCハンドブック (CRC Handbook for coding and signal processing for recording systems)』所収、リンクwww.csee.wvu.edu/wcrl/ldpc.htm

【発明の概要】

【発明が解決しようとする課題】

【0065】

本発明の一般的な目的は、二部グラフによって表現できる誤り訂正符号を復号するための逐次反復式メッセージ・パッシング型復号アルゴリズムであって、現状技術で知られている同じ型のアルゴリズムよりも誤り率 (error rate) および収束率 (convergence rate) の意味で性能がよいものを提案することである。

【0066】

本発明の第一の目的は、LDPC符号を復号するための逐次反復式メッセージ・パッシング型復号アルゴリズムであって、SPAアルゴリズムより著しく低い複雑さを持ち、その一方で所与の信号対雑音比に対してSPAアルゴリズムに匹敵するまたさらにはよりよい誤り率性能を持ち、またノイズ・パワーの推定を必要としないものを提案することである。

【0067】

本発明のもう一つの個別目的は、LDPC符号を復号するためのメッセージ・パッシング型復号アルゴリズムであって、SPAまたはMin-Sumアルゴリズムより高い収束率をもつものを提案することである。

【課題を解決するための手段】

【0068】

本発明は、複数の変数ノードおよび複数のチェック・ノードを有する二部グラフにおいて表示できる誤り訂正符号の復号のためのメッセージ・パッシングによる逐次反復的方法であって、複数の復号反復工程の各反復工程について：

- ・変数ノードまたはチェック・ノードが、これらのノードの近傍において利用可能な復号情報の対応する信頼度 (degree of reliability) に応じて分類され、信頼度の高いノードが信頼度の低いノードより前に分類され；

- ・こうして分類された各ノードが、前記分類によって定義された順序で、少なくとも一つのメッセージを隣接ノードに渡す (pass)、方法によって定義される。

【0069】

第一の実施形態によれば、分類されるべき各ノードについて、前記分類は、二部グラフにおいてこのノードから所定の距離以内にあるノードに存在する、またはそのようなノードによって送信もしくは受信された情報の信頼性の指標 (measurement) の計算と、こうして得られた指標の値のソート (sorting) を含む。

【0070】

前記複数の反復工程の各反復工程について、分類されたノードは次いで、前記分類によって定義された順序で逐次的に処理される。分類された各ノードについて、それに隣接するノードに宛てられたメッセージが計算され、前記各隣接ノードについて、該隣接ノードに隣接するノードへのメッセージが計算される。

【0071】

第二の実施形態によれば、分類されるべき各ノードについて、前記分類は、二部グラフにおいてこのノードから所定の距離以内に位置するノードに存在する、またはそのようなノードによって送信もしくは受信された情報の信頼性の指標の計算を含む。ノードは、前記指標の値の諸区間においてグループ化される。

【0072】

10

20

30

40

50

信頼性指標が整数値を使う場合、前記各整数値について、信頼性指標がこの値に等しくなるノードのインデックスが、それに関連付けられたメモリ・ゾーンに記憶される。

【0073】

前記複数の反復工程の各反復工程について、ノード・グループは前記分類によって定義された順序で逐次的に処理され、各ノード・グループについて、そのグループ内のノードに隣接するノードへのメッセージが計算され、前記各隣接ノードについて、それ自身が該隣接ノードに隣接するノードへのメッセージが計算される。

【0074】

ある変形によれば、前記二部グラフにおける各変数ノードについて、前記複数の反復工程の各反復工程は、このノード中にすでに存在する事前情報および隣接チェック・ノードからこのノードによって受信されるメッセージの関数として事後情報を計算するステップをも有する。

10

【0075】

この事後情報計算ステップに続いて、前記変数のハードな値についての決定ステップがあってもよい。

【0076】

次のステップは、こうして得られた変数のハードな値が、グラフ中のすべてのチェック・ノードに関連付けられたパリティ・チェックを満たすかどうかを試験することである。もし満たせば、前記ハードな値によって構成される語が復号された語として与えられる。

【0077】

20

有利には、前記分類は、所定数の復号反復工程の後に中断され、本復号方法は次いでその復号反復工程を、前記変数ノードまたは前記チェック・ノードの前記分類なしに続ける。

【0078】

前記分類は、前記変数の事後の値と事前の値との間の差の絶対値の最小値が所定の閾値より大きい場合、中断される。その際、本復号方法は、その復号反復工程を、前記変数ノードまたは前記チェック・ノードの前記分類なしに続ける。

【0079】

第一の応用では、前記誤り訂正符号はターボ符号である。

【0080】

30

第二の応用では、前記誤り訂正符号は、N個の変数ノードおよびM=N-K個のチェック・ノードをもつ二部グラフによって表されるLDPC符号(K,N)である。

【0081】

後者の場合、インデックスm {1,...,M}をもつチェック・ノードからインデックスn {1,...,N}をもつ変数ノードへの α_{mn} メッセージは次のように計算できる：

【0082】

【数21】

$$\beta_{mn} = \left(\prod_{n' \in H(m) - \{n\}} \text{sgn}(\alpha_{mn'}) \right) \cdot \Phi \left(\sum_{n' \in H(m) - \{n\}} \Phi(|\alpha_{mn'}|) \right)$$

40

ここで、 α_{mn} はインデックスn'をもつ変数ノードからインデックスmをもつチェック・ノードへのメッセージを表し、H(m)はインデックスmをもつチェック・ノードに隣接するすべての変数ノードを表し、xが正であれば $\text{sgn}(x) = 1$ 、そうでなければ $\text{sgn}(x) = -1$ であり、

【0083】

【数 2 2】

$$\Phi(x) = \ln\left(\frac{e^x + 1}{e^x - 1}\right)$$

である。

【0084】

あるいはまた、インデックス $m \in \{1, \dots, M\}$ をもつチェック・ノードからインデックス $n \in \{1, \dots, N\}$ をもつ変数ノードへの α_{mn} メッセージは次のように計算できる：

10

【0085】

【数 2 3】

$$\beta_{mn} = \left(\prod_{n' \in H(m) - \{n\}} \text{sgn}(\alpha_{mn'}) \right) \cdot \min_{n' \in H(m) - \{n\}} (|\alpha_{mn'}|)$$

20

ここで、 α_{mn} はインデックス n をもつ変数ノードからインデックス m をもつチェック・ノードへのメッセージを表し、 $H(m)$ はインデックス m をもつチェック・ノードに隣接するすべての変数ノードを表し、 x が正であれば $\text{sgn}(x) = 1$ 、そうでなければ $\text{sgn}(x) = -1$ である。

【0086】

ある例示的な実施形態によれば、前記分類はチェック・ノードに適用され、前記所定の距離は2に等しい。インデックス m をもつチェック・ノードの信頼性指標は次いで次のように計算される：

【0087】

30

【数 2 4】

$$\tilde{f}_2(m) = c + \sum_{n \in H(m)} \min(1, \text{Card}\{m' \in H(n) - \{m\} | c_{m'} = -1\})$$

ここで、 $H(m)$ はインデックス m をもつチェック・ノードに隣接する変数ノードの集合を表し、 $H(n)$ はインデックス n をもつ変数ノードに隣接するチェック・ノードの集合を表し、 $\text{Card}(\cdot)$ はある集合のカージナル数 (cardinal) を表す。 $c_m = +1$ であれば $c = 0$ 、 $c_m = -1$ であれば $c = \max + 1$ であり、ここで \max は二分グラフ中のチェック・ノードの最大次数 (degree) である。 $c_m = +1 / c_m = -1$ は、インデックス m をもつチェック・ノードについてパリティ・チェックがそれぞれ満たされている / 満たされていないことを意味する。

40

【0088】

最後に、本発明は、コンピュータによって実行されたときに上記で定義された復号方法の諸ステップを実装するよう適応されたソフトウェア手段を有するコンピュータ・プログラムに係する。

【図面の簡単な説明】

【0089】

50

【図1】(K,N)符号についてのタナー・グラフの例を示す図である。

【図2】現状技術に基づいて知られている逐次反復式メッセージ・パッシング復号の原理を図的に示す図である。

【図3】並列型スケジューリングに基づくメッセージ・パッシングによる逐次反復式復号の最初のいくつかの反復工程を示す図である。

【図4】シリアル型スケジューリングに基づくメッセージ・パッシングによる逐次反復式復号の最初のいくつかの反復工程を示す図である。

【図5】混合型スケジューリングに基づくメッセージ・パッシングによる逐次反復式復号の最初のいくつかの反復工程を示す図である。

【図6A】あるノードの1次の近傍を示す図である。

10

【図6B】あるノードの2次の近傍を示す図である。

【図7】本発明に基づく、メッセージ・パッシングによる逐次反復式復号の原理を示す図である。

【図8】本発明の第一の実施形態に基づく、メッセージ・パッシングによる復号のための逐次反復式方法を示す図である。

【図9】本発明の第二の実施形態に基づく、メッセージ・パッシングによる復号のための逐次反復式方法を示す図である。

【図10】LDPC符号を復号するための、本発明に基づく復号方法の例示的な適用を示す図である。

【図11】特定のLDPC符号について、本発明に基づく復号方法によって得られる誤り率を、現状技術に基づく他の諸方法との比較で示す図である。

20

【発明を実施するための形態】

【0090】

いま一度、N個の変数およびM個のチェックをもつ二部グラフによって表現できる誤り訂正符号を考える。

【0091】

グラフ上の二つのノード μ 、 ν をつなぐ、グラフを通る最短経路の長さは距離と呼ばれ、その長さはエッジ数で表されて $D(\mu, \nu)$ と記される。二部グラフの定義を考えると、同じ型の二つのノードの間の距離は偶数であり、異なる型の二つのノードの間の距離は奇数であることが演繹される。

30

【0092】

グラフ G の任意のノード μ の d 次 (order d) の近傍は、 μ から d 以下の距離に位置するノードの集合 $V^{(d)}$ として定義される。すなわち：

$$V^{(d)} = \{ \mu \mid D(\mu, \nu) \leq d \} \quad (21)$$

よって、あるノードの0次の近傍はそのノード自身からなり、1次の近傍は0次の近傍とこのノードに隣接する全ノードとの組み合わせであり、2次の近傍は1次の近傍と1次の近傍中のノードに隣接する全ノードとの組み合わせである、などとなる。n個の変数およびm個のチェックからなる二部グラフについては、これらのノードの0次から2次の近傍は：

【0093】

【数 2 5】

$$V_n^{(0)} = \{n\} ; V_n^{(1)} = \{n\} \cup H(n) ; V_n^{(2)} = H(n) \bigcup_{m \in H(n)} H(m) \quad (22)$$

$$V_m^{(0)} = \{m\} ; V_m^{(1)} = \{m\} \cup H(m) ; V_m^{(2)} = H(m) \bigcup_{n \in H(m)} H(n) \quad (23)$$

10

によって与えられる。

【0094】

単集合 $\{n\}$ (または $\{m\}$) が $V_n^{(2)}$ (または $V_m^{(2)}$) の表式に現れていないことに気づかれるであろう。これは、該単集合は、隣接関係の対称性から、組み合わせの第二項にすでに含まれているからである。

【0095】

一般に、二部グラフのノード の近傍は、再帰関係

【0096】

20

【数 2 6】

$$V_v^{(d+1)} = \bigcup_{\mu \in V_v^{(d)}} H(\mu) \quad (24)$$

によって得ることができる。

【0097】

図 6 A および図 6 B は、近傍の概念を説明するために、図 1 のグラフ上のチェック・ノード m の 1 次および 2 次の近傍を示している。前記近傍に属するノードはグレーで示されている。

30

【0098】

本記載の残りの部分では、 のノード の d 次の信頼性指標 (order d reliability measurement) という用語が、近傍 $V_v^{(d)}$ 中の利用可能な復号情報、換言すれば前記近傍に属するノードに存在するまたはそのようなノードによって送信もしくは受信される情報に依存し、復号動作の局所的な信頼度を示す変数 $f_d(\)$ を指すために使われることになる。信頼度が高まるとともに $f_d(\)$ が増加するときその指標を「正」と形容し、そうでない場合に「負」と形容することにする。

【0099】

40

チェック・ノード m において利用可能な復号情報は、第一に、パリティ・チェックが検証されているかどうかを示す値 c_m と、第二に、隣接変数ノード $n \in H(m)$ に送信される m_n メッセージとを含む。値 c_m およびメッセージ m_n が復号についての信頼性情報を担う。

【0100】

変数ノード n において利用可能な情報は、事前情報 (\hat{c}_n)、事後情報 (\hat{c}_n 付きの \hat{c}_n)、隣接チェック・ノード $m \in H(n)$ に送られるべき外来情報メッセージ (m_n) およびハードな値 (\hat{c}_n 付きの \hat{c}_n) を含む。

【0101】

上記の情報は、ハードな値を除いて、復号についての信頼性情報を担持している。

【0102】

50

一般に、LLRの形で表されたソフトな値

【 0 1 0 3 】

【 数 2 7 】

$$\beta_{mn}, \alpha_{mn}, \alpha_n, \hat{\alpha}_n$$

を考えると、その絶対値が大きいくほど復号は信頼できることになる。

【 0 1 0 4 】

以下は、0次ないし2次について信頼性指標の若干の例示的な、限定するものではない例を含む。

10

【 0 1 0 5 】

・0次の信頼性指標：

この場合、近傍はノード自身に還元される。変数ノードnの0次の信頼性を測定するために、次の関数を選ぶことができる：

【 0 1 0 6 】

【 数 2 8 】

$$f_0(n) = |\hat{\alpha}_n| \quad (25)$$

20

これは、ハードな値

【 0 1 0 7 】

【 数 2 9 】

$$\bar{\alpha}_n$$

についての信頼の度合い (degree of confidence) を示す。

【 0 1 0 8 】

30

チェック・ノードmについては、単に次の関数を選ぶことができる：

【 0 1 0 9 】

【 数 3 0 】

$$\tilde{f}_0(m) = c_m \quad (26)$$

パリティ・チェックが満たされていたとすると ($c_m = 1$)、満たされていない場合よりも復号が信頼できることは理解できる。

40

【 0 1 1 0 】

・1次の信頼性指標：

ここでもまた、変数ノードとチェック・ノードについて処理が異なる。変数ノードについては、次の関数

【 0 1 1 1 】

【数 3 1】

$$f_1(n) = \text{Card}\{m \in H(n) | c_m = +1\} \quad (27)$$

$$f'_1(n) = \text{Card}\{m \in H(n) | c_m = -1\} \quad (27')$$

$$f_1(n) = \text{Card}\{m \in H(n) | c_m = +1\} - \text{Card}\{m \in H(n) | c_m = -1\} \quad (28)$$

$$f_1(n) = \text{Card}\{m \in H(n) | c_m = -1\} - \text{Card}\{m \in H(n) | c_m = +1\} \quad (28')$$

$$f_1(n) = \min_{m \in H(n)} |\beta_{mn}| \quad (29)$$

10

20

の一つが1次の信頼性指標として選べる。

【0 1 1 2】

表式(27)は、パリティ・チェックが検証されている、変数ノードnに接続されたチェック・ノードmの数を表す。

【0 1 1 3】

表式(28)は、符号の不規則性（換言すれば、複数の変数ノードが異なる次数（degree）をもつ可能性）取り入れ、検証されたおよび検証されていないパリティ・チェックのサマリーを生成する。

【0 1 1 4】

表式(27')および(28')は、正の指標(27)および(28)に対応する負の指標である。

30

【0 1 1 5】

(29)に表されている信頼性指標は、隣接チェック・ノードから受信される信頼性情報の縮小（reduction）に基づいているので、保守的なものである。

【0 1 1 6】

チェック・ノードmについては、1次の信頼性指標について次の関数

【0 1 1 7】

【数 3 2】

$$\tilde{f}_1(m) = \min_{n \in H(m)} (|\hat{\alpha}_n|) \quad (30)$$

$$\tilde{f}_1(m) = \min_{n \in H(m)} (|\alpha_{mn}|) \quad (31)$$

$$\tilde{f}_1(m) = \left(c_m, \min_{n \in H(m)} (|\hat{\alpha}_n|) \right) \quad (32)$$

10

の一つを選ぶことができる。

【0 1 1 8】

表式(30)は、チェックmが接続されている変数の信頼性（事後情報）の最小値を示す。

【0 1 1 9】

表式(31)は同様の基準を表しているが、外来情報に適用される。表式(29)と対応していることに気づかれるであろう。

【0 1 2 0】

表式(32)は二つの情報項目のデカルト積を表し、結果としてそれらを統合して取り入れている。第一のものはパリティ・チェックの検証についてであり、第二のものは(30)と同様に事後情報についてである。

【0 1 2 1】

・2次の信頼性指標：

変数ノードnについては次の関数

【0 1 2 2】

【数 3 3】

$$f_2(n) = \text{Card} \left\{ m \in H(n) \mid c_m = -1 \wedge \hat{\alpha}_n = \min_{n' \in H(m)} \hat{\alpha}_{n'} \right\} \quad (33)$$

40

を選ぶことができる。

【0 1 2 3】

表式(33)は負の復号信頼性指標を表す。これは、変数nに関係したチェックの数であって、パリティ条件が満たされておらず、変数nがこれらのチェックに隣接するすべての変数の間で最も信頼できないものであるようなものを示す。実は、測定値(33)は、指標(27')と(30)の合成と考えることができる。

【0 1 2 4】

チェック・ノードmについては次の関数のうちの一つを選ぶことができる。

【0 1 2 5】

50

【数 3 4】

$$\tilde{f}_2(m) = \text{Card}\{m' \in V_m^{(2)} - \{m\} | c_{m'} = -1\} = \sum_{n \in H(m)} \text{Card}\{m' \in H(n) - \{m\} | c_{m'} = -1\} \quad (34)$$

$$\tilde{f}_2(m) = \sum_{n \in H(m)} \min(1, \text{Card}\{m' \in H(n) - \{m\} | c_{m'} = -1\}) \quad (35) \quad 10$$

$$\tilde{f}_2(m) = c + \sum_{n \in H(m)} \min(1, \text{Card}\{m' \in H(n) - \{m\} | c_{m'} = -1\}) \quad (36)$$

ここで、 $c_m = +1$ であれば $c = 0$ 、 $c_m = -1$ であれば $c = \delta_{\max} + 1$ であり、ここで δ_{\max} はチェック・ノードの最大の次数 (degree)、すなわち

【0 1 2 6】

【数 3 5】

$$\delta_{\max} = \max_m (\text{Card}(H(m)))$$

である。

【0 1 2 7】

表式(34)は、 m の m 自身を除いた次数2の近傍中において、満たされていないパリティ・チェックの数を与え、復号信頼性の負の指標を表している。表式(34)の二番目の等号は、当該近傍に長さ4のサイクルがない場合にのみ満たされることを注意しておく。實際上、タナー・グラフが可能な最長のサイクルをもつような符号が使われるので、この等号は通例検証される。

【0 1 2 8】

表式(35)は m に隣接する、 m 自身を除いた、少なくとも一つの満たされていないチェックに接続されている変数の数を示し、結果として、復号信頼性の負の指標を表している。和の記号の中にある項は0または1に等しいので、この測定値の値は0から δ_{\max} の間である。

【0 1 2 9】

表式(35)は、ノード m についてパリティ・チェックが検証されているかどうかには依存した区別をしない。この情報を取り入れるために、チェック m の検証に依存する値をもつバイアス c が表式(36)に導入された。上述したバイアス値 ($c = 0$ 、 $\delta_{\max} + 1$) について、信頼性指標は検証されたチェックについては0から δ_{\max} の間の値に等しく、検証されたチェックについては $\delta_{\max} + 1$ から $2 \delta_{\max} + 1$ までの値に等しい。これは、上で定義した意味において負の信頼性指標であることを注意しておく。すなわち、チルダ付き $f_2(m)$ の値が小さいほどチェック m の値は信頼できる。よって、(35)の値に関わりなく、検証されたチェックの信頼度は、検証されていないチェックの信頼度よりも常に大きい。

【0 1 3 0】

上で定義された関数のいくつか、特に関数(34)ないし(36)は整数値であることを注意しておく。したがって、これらの関数はマイクロプロセッサ実装に特に好適である。

【0 1 3 1】

重要なことに、上で定義された関数のいくつか、特に関数(34)ないし(36)は、ハードな値 c_m しか使っていない。

【 0 1 3 2 】

【 数 3 6 】

$$c_m = \prod_{n \in H(m)} \bar{\alpha}_n$$

であるから、ノイズ・パワー(2)の値はこれらの関数の計算に現れていないことが見て取れる。

【 0 1 3 3 】

本発明の基本的な概念は、グラフのノードをその信頼性に応じて分類し、最も信頼できるノードまたはノードのグループから始めて、シリアルまたは混合型スキームを使ったメッセージ・パッシングによって復号を行うということである。これは復号性能を、収束率およびビット誤り率(BER: bit error rate)の両方の面で著しく改善する。

【 0 1 3 4 】

最も信頼できるノードによって伝送されるメッセージが、隣接ノードを復号するのに最も効率的に寄与できることは理解されるであろう。よって、ある反復工程から次の反復工程へ、そしてノードからノードへの信頼性の高速の伝搬があり、これに相関して諸反復工程が進行するにつれて外来情報の絶対値の著しい増加がある。

【 0 1 3 5 】

図7は、本発明に基づく復号の原理を図的に示している。のちに詳述する初期化段階710ののち、反復工程ループが始まる。グラフのチェック・ノードまたは変数ノードが720において信頼度に依存して分類される。ここでは、チェック・ノードが分類されると想定する。変数ノードの分類は対称的な諸ステップにつながる。

【 0 1 3 6 】

ステップ730ではシリアル処理または混合処理が行われる。シリアル型処理について、チェック・ノードが逐次的に処理され、各チェック・ノードに隣接する変数ノードも、最も信頼できるチェック・ノードから始まって信頼度が低下していく順序で、逐次的に処理される。混合型の処理については、チェック・ノードは信頼性グループに分類されている。第一のグループは最も信頼できるノードからなり、第二のグループはそれよりは信頼できないノードからなる、などである。変数ノードおよび第一のグループ中のチェック・ノードに対して並列処理がなされ、次いで次のグループの変数ノードおよびチェック・ノードに対して並列処理がなされ、それ以上グループがなくなるまで続けられる。ステップ740、750、760、765、767、770、775はそれぞれ図2のステップ240、250、260、265、267、270、275と同一であり、よってこれ以上詳細に述べることはしない。

【 0 1 3 7 】

図8は、シリアル・スケジューリングを使った本発明のある実施形態のさらなる詳細を示している。

【 0 1 3 8 】

ステップ810でメッセージ m_n および c_m が初期化される。たとえば、変数 n およびチェック m $H(n)$ の対のそれぞれについて、 c_m は $c_m = 1$ と初期化される。同様に、チェック m および変数 n $H(m)$ の対のそれぞれについて、 m_n は $m_n = 0$ と初期化される。

【 0 1 3 9 】

次いでステップ820で、前記チェックは、信頼性指標

【 0 1 4 0 】

10

20

30

40

【数 3 7】

$$\tilde{f}_d(m)$$

を使って信頼度の降順にソートされる。還元すれば、正の指標については、チェックは
【 0 1 4 1】

【数 3 8】

$$\tilde{f}_d(m_0) \geq \tilde{f}_d(m_1) \geq \dots \geq \tilde{f}_d(m_{M-1})$$

10

となるようソートされ、負の指標については逆順にソートされる。デカルト積指標（たとえば(32)参照）については、その指標が正か負かに依存して辞書順関係または逆辞書順関係が使われる。また、チェック・ノードのカウンタを $j=0$ と初期化する。

【 0 1 4 2】

次いで 8 2 5 で現在のチェック $m=m_j$ が選択され、8 3 0 で変数 $n \in H(m)$ が処理される。換言すれば、 $m_n = F_V(\{m'_n | m' \in H(n) - \{m\}\})$ が計算される。

【 0 1 4 3】

次いでステップ 8 3 5 は現在チェック処理 m を行う。換言すれば、 $n \in H(m)$ について $m_n = F_C(\{m'_n | n' \in H(m) - \{n\}\})$ が計算される。 20

【 0 1 4 4】

ある変形実施形態によれば、ステップ 8 3 0 と 9 3 5 は逆にされる。

【 0 1 4 5】

ステップ 8 3 7 でチェック・ノード・カウンタがインクリメントされ、ステップ 8 3 9 で全部処理されたかどうかを見出す試験がなされる。もしまだである場合には、処理はループでステップ 8 2 5 に戻り、それ以外の場合には、処理は、それぞれ事後情報の計算、ハード値の決定およびパリティ・チェックの検証を表すステップ 8 4 0 ないし 8 6 0 に続く。ステップ 8 4 0、8 5 0、8 6 0、8 6 5、8 6 7、8 7 0、8 7 5 はそれぞれ図 2 のステップ 2 4 0、2 5 0、2 6 0、2 6 5、2 6 7、2 7 0、2 7 5 と同一である。 30

【 0 1 4 6】

ある代替的な実施形態によれば、チェックではなく変数がソートされる。その場合、内側のループ 8 2 5 ないし 8 3 9 は変数一つずつに適用されていき、前と同じようにして、現在の変数が、それに隣接するチェックの処理の前または後に、処理されうる。

【 0 1 4 7】

図 9 は、混合スケジューリングを使った本発明のある実施形態を示している。

【 0 1 4 8】

ステップ 9 1 0 では、図 8 のステップ 8 1 0 のように、メッセージ m_n および m'_n が初期化される。

【 0 1 4 9】

40

ステップ 9 2 0 では、すべてのチェック・ノードまたは変数ノードが、それぞれ異なる信頼性クラスに対応するグループに分割される。たとえば、信頼性クラス (reliability class) は信頼性指標の値のある区間によって定義される。この実施形態は、指標が整数値を使うときに特に有利である。その場合、ある整数値に、あるいは複数の連続する整数値にある範囲を割り当て、ノードのインデックスを対応するクラスの関数としてテーブルに記憶することができる。以後、異なる信頼性クラスを G_0, G_1, \dots, G_{G-1} と記すことにする。ここで、 G_0 は信頼度が最も高いクラス、 G_{G-1} は信頼度が最も低いクラスである。 G_0, G_1, \dots, G_{G-1} は対応するノード・グループである。以下では前記の分割はチェック・ノードに対してなされたと想定する。

【 0 1 5 0】

50

また、信頼性クラスのカウンタを $j = 0$ に初期化する。

【0151】

ステップ925では、現在の信頼性クラス j に対応するノードのグループ、つまり G_j を選択する。

【0152】

ステップ930では、変数 $n \in H(m)$ 、 $m \in G_j$ が並列に処理される。換言すれば、メッセージ $m_n = F_V(\{m'_n | m' \in H(n) - \{m\}\})$ 、 $m \in G_j$ 、 $n \in H(m)$ が計算される。

【0153】

ステップ935では、チェック $m \in G_j$ が並列に処理される。換言すれば、メッセージ $m_n = F_C(\{m'_n | n' \in H(m) - \{n\}\})$ 、 $m \in G_j$ 、 $n \in H(m)$ が計算される。

10

【0154】

ある変形実施形態によれば、ステップ930と935は逆にされる。

【0155】

ステップ937ではクラス・カウンタがインクリメントされ、ステップ939で全部処理されたかどうかを判定するべく試験される。もしまだである場合には、処理はループでステップ925に戻り、それ以外の場合には、処理は、それぞれ事後情報の計算、ハード値の決定およびパリティ・チェックの検証を表すステップ940ないし960に続く。ステップ940、950、960、965、967、970、975はそれぞれ図2のステップ240、250、260、265、267、270、275と同一である。

【0156】

20

図10は、整数値をもつ信頼性指標に基づく、混合スケジューリングによるLDPC符号の復号への、本発明の個別的な適用を示している。この場合、(36)で定義されたチェック・ノードに対する負の指標を選んだ。

【0157】

ステップ1010では、ソフトな事後の値 ($\hat{\alpha}_n$ 付きの α_n) が対応する観察

【0158】

【数39】

$$\hat{\alpha}_n = \alpha_n$$

30

によって初期化され、 m_n メッセージは0によって初期化される。

【0159】

また、反復カウンタも0に初期化する。

【0160】

ステップ1015では、ハードな値についての決定がなされる。すなわち、

【0161】

【数40】

$$\bar{\alpha}_n = \text{sgn}(\hat{\alpha}_n)$$

40

とする。

【0162】

ステップ1020は

【0163】

【数 4 1】

$$\chi = \sum_{m=1}^M c_m$$

の値を計算する。ここで、 c_m はパリティ・チェック

【0 1 6 4】

【数 4 2】

10

$$c_m = \prod_{n \in H(m)} \bar{\alpha}_n, \quad m=1, \dots, M$$

である。

【0 1 6 5】

ステップ 1 0 2 3 は、 M かどうかを、つまりすべてのパリティ・チェックが検証されているかどうかを試験する。もしそうであれば、ステップ 1 0 2 5 で、符号語を提供することによって復号アルゴリズムは終了される。もしそうでなければ、処理は、指標 (36) を使ってチェック・ノードの信頼性指標を計算することによってステップ 1 0 2 7 に続く。この指標は $0, 1, \dots, 2^{\max} + 1$ のうちの整数値に等しいことを想起されたい。ここで、 \max はチェック・ノードの最大の次数である。この場合、信頼性クラスを整数値、つまり $0, 1, \dots, 2^{\max} + 1$ によって割り当てる。ここで、 $2^{\max} + 1$ であり、 j は指標の値に対応し、 $j \in \{0, 1, \dots, 2^{\max} + 1\}$ である。

20

【0 1 6 6】

ステップ 1 0 2 9 では、現在の信頼性クラス j に対応するノードのグループ、つまり G_j が選ばれる。

【0 1 6 7】

ステップ 1 0 3 0 では、変数 $n \in H(m)$ 、 $m \in G_j$ の処理が並列になされる。換言すれば、

30

【0 1 6 8】

【数 4 3】

$$\alpha_{mn} = \hat{\alpha}_n - \beta_{mn} \quad \forall m \in G_j$$

が計算される。

【0 1 6 9】

ステップ 1 0 3 5 ではチェック $m \in G_j$ が並列に処理される。換言すれば、変数 $n \in H(m)$ 、 $m \in G_j$ に送られるべきメッセージ、具体的には、SPA 処理については

40

【0 1 7 0】

【数 4 4】

$$\beta_{mn} = \left(\prod_{n' \in H(m) - \{n\}} \text{sgn}(\alpha_{mn'}) \right) \cdot \Phi \left(\sum_{n' \in H(m) - \{n\}} \Phi(|\alpha_{mn'}|) \right)$$

簡略化された Min-Sum 型処理については

【0 1 7 1】

50

【数 4 5】

$$\beta_{mn} = \left(\prod_{n' \in H(m) - \{n\}} \text{sgn}(\alpha_{mn'}) \right) \cdot \min_{n' \in H(m) - \{n\}} (|\alpha_{mn'}|)$$

が計算される。図にはこの後者の場合が示されている。次いでステップ 1 0 4 0 で事後情報が

【0 1 7 2】

【数 4 6】

10

$$\hat{\alpha}_n = \alpha_n + \sum_{m \in H(n)} \beta_{mn}$$

を使って計算される。

【0 1 7 3】

和の中で使われているチェック $m \in H(n)$ の一部はグループ G_j の一部を形成しない可能性があり、よって、これらのチェックを起源とするメッセージ β_{mn} はステップ 1 0 3 5 で計算され直さなかったことを注意しておく。さらに、変数 n はいくつかのチェック $m \in H(n)$ に隣接するので、事後情報の前記計算は、 $H(n)$ のチェックが処理されるにつれ複数回なされることになる。ステップ 1 0 4 0 における和が複数回計算されるのを防ぎ、変更されなかった一部のメッセージ β_{mn} が毎回再計算されるのを防ぐため、ステップ 1 0 3 5 および 1 0 4 0 は有利には次の演算を含む単一のステップによって置き換えられる。

20

【0 1 7 4】

$m \in G_j$ 、 $n \in H(m)$

・古いメッセージを引く：

【0 1 7 5】

【数 4 7】

30

$$\hat{\alpha}_n = \hat{\alpha}_n - \beta_{mn}$$

・新しいメッセージを計算する：

【0 1 7 6】

【数 4 8】

$$\beta_{mn} = \left(\prod_{n' \in H(m) - \{n\}} \text{sgn}(\alpha_{mn'}) \right) \cdot \min_{n' \in H(m) - \{n\}} (|\alpha_{mn'}|)$$

40

・前記新しいメッセージを足すことによって事後情報を更新する：

【0 1 7 7】

【数 4 9】

$$\hat{\alpha}_n = \hat{\alpha}_n + \beta_{mn}$$

このようにして、チェック $m \in H(n)$ によって変数 n に送られた新しいメッセージが計算される際に事後情報が更新される。これは、事後情報の計算のために実行される演算の数を

50

著しく減らすことができる。

【0178】

次のステップ1043は信頼性クラス・カウンタをインクリメントし、次いでステップ1045ですべてのクラスが処理されたかどうかを試験する。もしまだである場合には、ステップ1029が繰り返されて次の信頼性クラスが処理される。もしそうである場合には、ステップ1050で反復カウンタがインクリメントされる。ステップ1053は、停止基準が満たされているかどうか、換言すれば最大反復回数に達しているかどうかを試験する。もしそうであれば、ステップ1055でループは終了され、復号は失敗したと結論される。もしそうでなければ、処理は新しい反復工程のためにステップ1015にループで戻る。

10

【0179】

図10において、ハードな値についての決定ステップおよびパリティ・チェックの検証のためのステップが、信頼性計算およびノード処理のステップに先行していることが見て取れる。これは、ここで使われる信頼性指標(表式(36))がハード決定 c_m を必要とし、結果としてハード決定が前もって計算されることを要求するという事実のためである。

【0180】

図10に示されるアルゴリズムは、現状技術に基づくSPAまたはMin-Sumアルゴリズムよりやや複雑なだけである。他方、近傍信頼性によるスケジューリングは、同じ信号対雑音比(SNR)についてより低いビット誤り率(BER)に達するための収束を加速する。あるいはまた、同じ設定された値の誤り率について、本発明に基づく復号方法は、より低い符号化利得をもつ符号を使うことができる。

20

【0181】

図11は、従来技術のMin-SumおよびSPAアルゴリズムのビット誤り率(BER)の面での性能を、広い範囲の信号対雑音比について、Min-Sum-FVと記される近傍信頼性によるスケジューリングを使ったMin-Sumアルゴリズムと比較している。使用された符号は不規則なLDPC符号[1008,2016]および[4032,8064]であり、したがって1/2の効率をもつ。反復数は、上記三つのアルゴリズムについて200に等しい。

【0182】

本発明に基づくMin-Sum-FVアルゴリズムが、複雑さは著しく低くなったにもかかわらず、従来技術のSPAアルゴリズムより優れていることが観察される。

30

【0183】

図7ないし図10に示された本発明に基づく復号方法では、近傍信頼性によるスケジューリングが復号期間を通じて使用されることが想定されている。ある変形によれば、このスケジューリングは初期の反復工程の間のみ、観察値からノイズを取り除くために適用され、次いで通常の処理に移行することができる。移行基準は、反復回数 l_{iter_f} であつてもよいし、あるいは有利には外来情報の絶対値とある基準閾値との比較に基づいていてもよい。たとえば、逐次反復式復号において、最小信頼性閾値を T_f で表すものとして、

【0184】

【数50】

40

$$\min_{n=1,\dots,N} |\hat{\alpha}_n - \alpha_n| > T_f \quad (37)$$

となるやいなや、Min-Sumまたは通常のSPA処理に復帰することができる。この変形は、復号された値が十分信頼できるようになり、収束が確実となるとすぐ、ノードのソートまたは分類を防止する。

50

【 0 1 8 5 】

本発明は、二部グラフによって表現できる誤り訂正符号、特にLDPC符号またはターボ符号の復号に適用可能である。本発明は、データ記録または遠隔通信の分野において、特に、すでにLDPC符号を使っている遠隔通信システム、たとえばIEEE 802.3a (10ギガビット毎秒イーサネット (登録商標))、DVB-S2 (衛星ビデオ放送)、IEEE 802.16 (WiMAX) といった規格に準拠する遠隔通信システムのために、あるいはこれらの符号を使うことができる遠隔通信システム、たとえばIEEE 802.11 (WLAN) および IEEE 802.20 (移動体ブロードバンド無線アクセス) といった規格を満たすシステムのために使用できる。

【 図 1 】

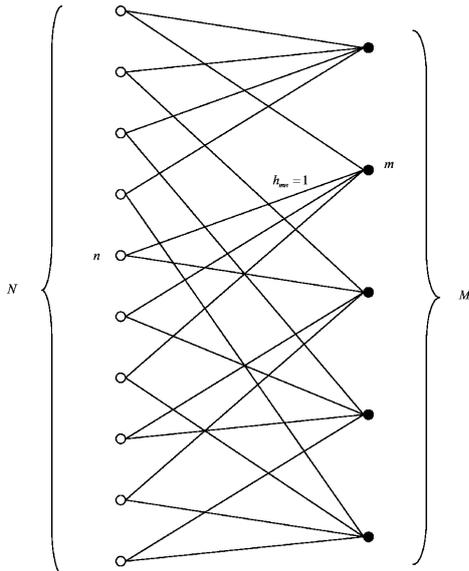


FIG. 1

【 図 2 】

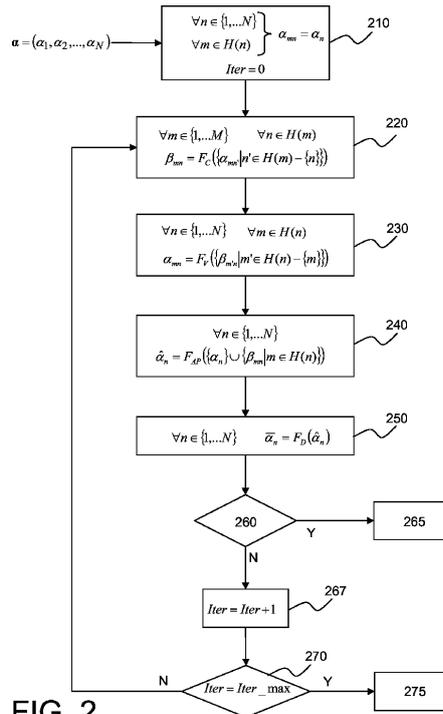


FIG. 2

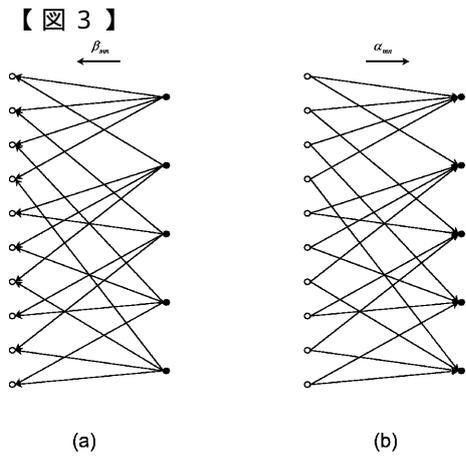


FIG. 3

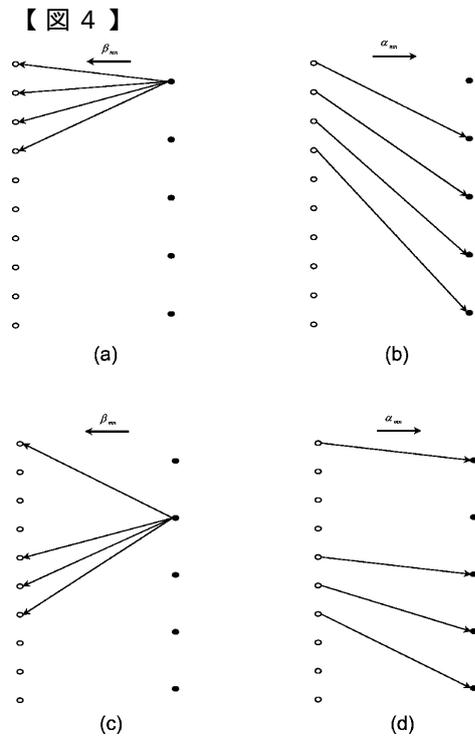


FIG. 4

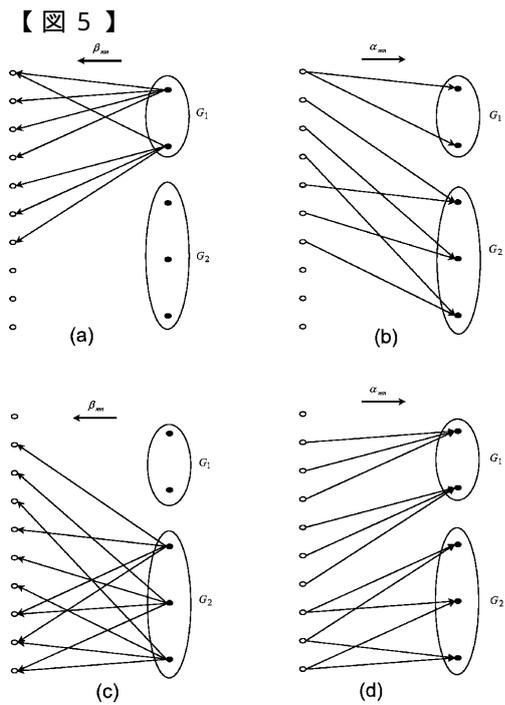


FIG. 5

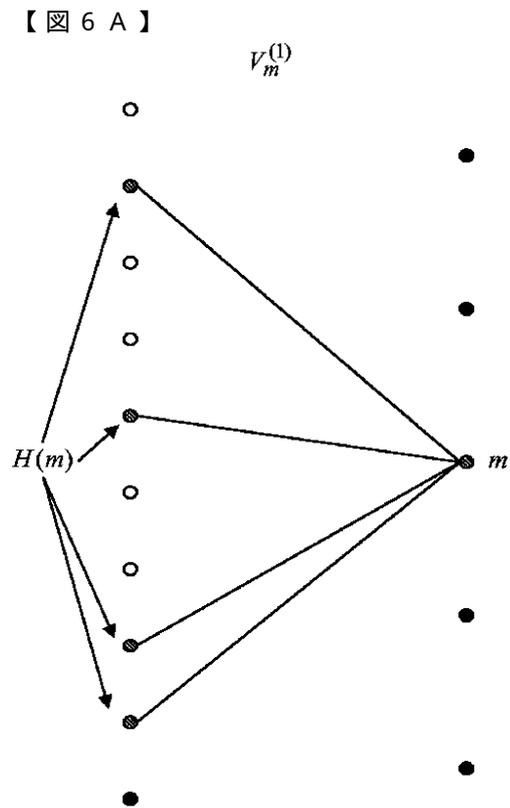


FIG. 6A

【 図 6 B 】

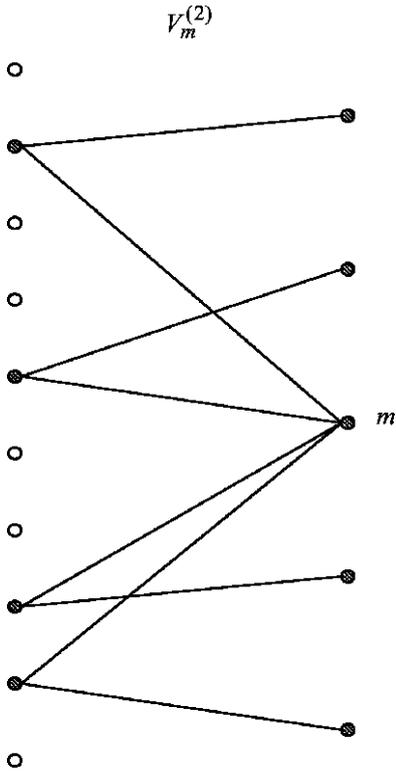


FIG. 6B

【 図 7 】

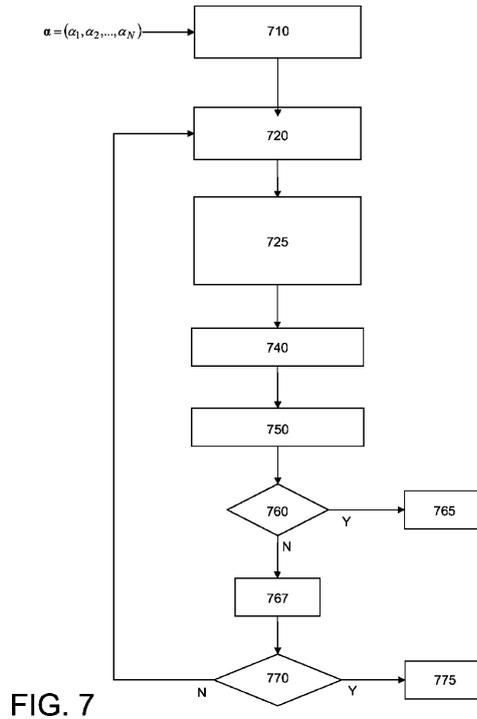


FIG. 7

【 図 8 】

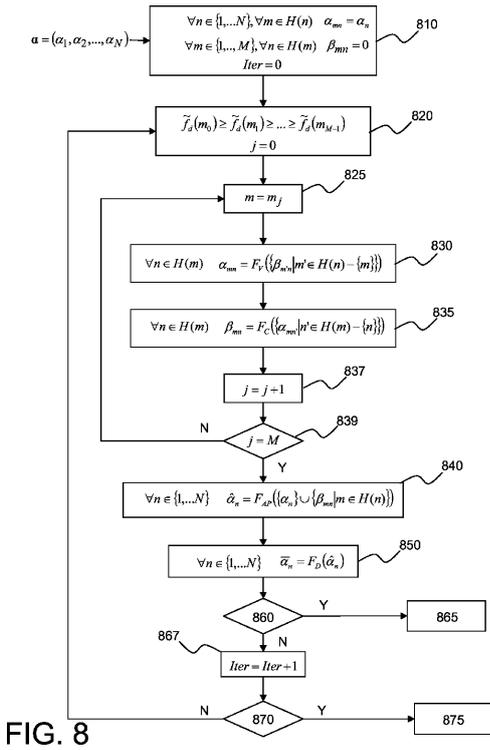


FIG. 8

【 図 9 】

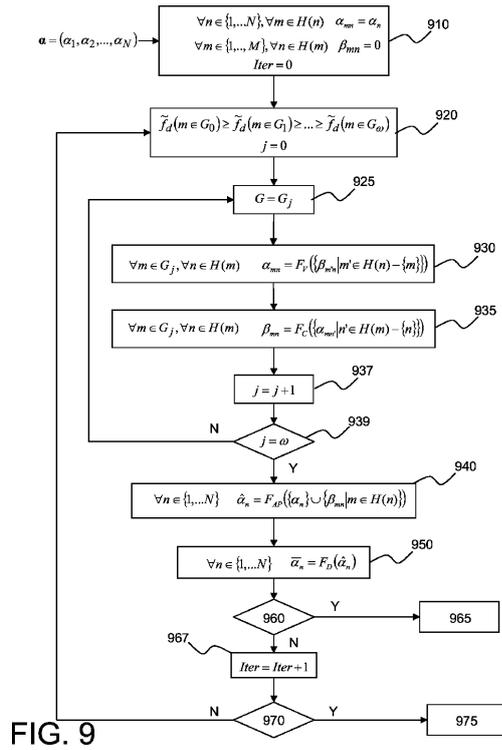


FIG. 9

【 10 】

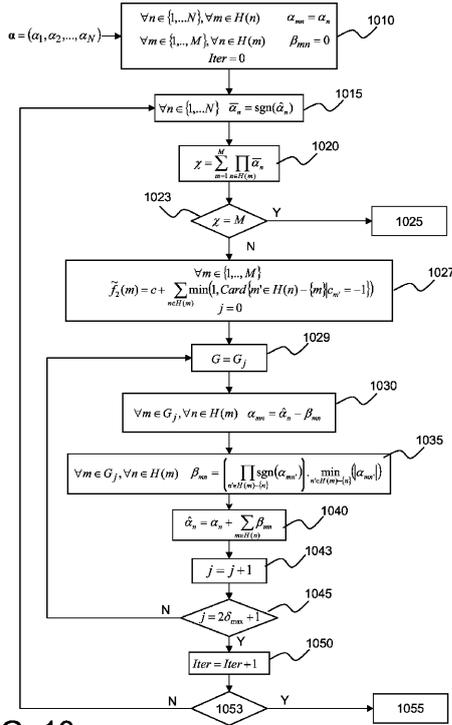
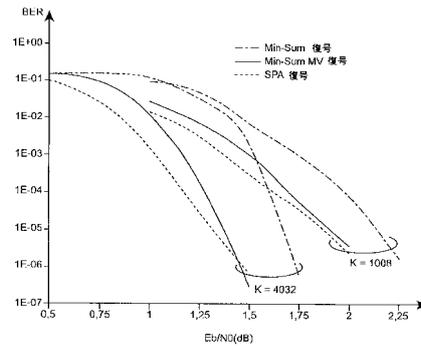


FIG. 10

【 11 】



フロントページの続き

(72)発明者 テナ, デイミトリ
フランス国, 3 8 6 0 0 フォンテーヌ テル・リュ・デュ・ラ・リベルテ 2 3

審査官 上田 翔太

(56)参考文献 特開2 0 0 6 - 2 7 9 3 9 6 (J P , A)
特開2 0 0 5 - 2 7 7 7 8 4 (J P , A)
Robert J. McEliece, 他2名, Turbo decoding as an instance of Pearl's "belief propagation" algorithm, Selected Areas in Communications, IEEE Journal on, 1 9 9 8年 2月
, Volume:16, Issue:2, pp.140-152

(58)調査した分野(Int.Cl., D B名)
H03M 13/19