

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5354979号
(P5354979)

(45) 発行日 平成25年11月27日(2013.11.27)

(24) 登録日 平成25年9月6日(2013.9.6)

(51) Int.Cl. F I
HO 3M 13/19 (2006.01) HO 3M 13/19
HO 3M 13/23 (2006.01) HO 3M 13/23

請求項の数 14 (全 61 頁)

(21) 出願番号	特願2008-178241 (P2008-178241)	(73) 特許権者	000005821
(22) 出願日	平成20年7月8日(2008.7.8)		パナソニック株式会社
(65) 公開番号	特開2009-177775 (P2009-177775A)		大阪府門真市大字門真1006番地
(43) 公開日	平成21年8月6日(2009.8.6)	(74) 代理人	100105050
審査請求日	平成22年11月29日(2010.11.29)		弁理士 鷺田 公一
(31) 優先権主張番号	特願2007-183492 (P2007-183492)	(72) 発明者	岡村 周太
(32) 優先日	平成19年7月12日(2007.7.12)		大阪府門真市大字門真1006番地 松下
(33) 優先権主張国	日本国(JP)		電器産業株式会社内
(31) 優先権主張番号	特願2007-339913 (P2007-339913)	(72) 発明者	村上 豊
(32) 優先日	平成19年12月28日(2007.12.28)		大阪府門真市大字門真1006番地 松下
(33) 優先権主張国	日本国(JP)		電器産業株式会社内
		(72) 発明者	折橋 雅之
			大阪府門真市大字門真1006番地 松下
			電器産業株式会社内

最終頁に続く

(54) 【発明の名称】 低密度パリティ検査畳み込み符号 (LDPC-CC) 符号化器及びLDPC-CC復号器

(57) 【特許請求の範囲】

【請求項1】

複数のシフトレジスタと、
 前記シフトレジスタの出力にウェイトを乗算する複数のウェイト乗算部と、
 前記複数のウェイト乗算部の出力を排他的論理和する排他的論理和演算器と、
 符号化する入力ビットのビット数をカウントするビット数カウンタと、
 前記ビット数に応じて前記複数のウェイト乗算部のウェイトを制御するウェイト制御部と、

を具備し、

前記ウェイト制御部は、LDPC-CC検査行列に準じた第1のウェイトパターンと、
 前記LDPC-CC検査行列を变形した検査行列に準じた第2のウェイトパターンと、を
 記憶し、前記入力ビットが情報系列の場合に、前記第1のウェイトパターンを用い、前記
 入力ビットがターミネーション系列の場合に、前記第2のウェイトパターンを用いる、
 低密度パリティ検査畳み込み符号 (LDPC-CC : Low-Density Parity-Check Convolutional Codes) 符号化器。

【請求項2】

前記第2のウェイトパターンは、前記LDPC-CC検査行列の各行の一番右の1を、
 行ごとに同じ数だけ左シフトして得られる検査行列に準じたウェイトパターンである

請求項1に記載のLDPC-CC符号化器。

【請求項3】

前記第2のウェイトパターンは、前記LDPC-CC検査行列の各行の一番右の1を0に変えた検査行列に準じたウェイトパターンである

請求項1に記載のLDPC-CC符号化器。

【請求項4】

前記第2のウェイトパターンは、前記LDPC-CC検査行列の各行の一番右の1を、ターミネーション系列のパリティ検査に用いられない列に左シフトして得られる検査行列に準じたウェイトパターンである

請求項1に記載のLDPC-CC符号化器。

【請求項5】

前記第2のウェイトパターンは、前記LDPC-CC検査行列のうち、検査ビットに対応する行の1を0に変えて得られる検査行列に準じたウェイトパターンである

請求項1に記載のLDPC-CC符号化器。

【請求項6】

前記入力ビットがターミネーション系列の場合に、前記ターミネーション系列のシステムティックビットをパンクチャするパンクチャ部、をさらに具備する

請求項1に記載のLDPC-CC符号化器。

【請求項7】

前記ウェイト制御部は、メモリ長が異なるLDPC-CC検査行列に準じたウェイトパターンを複数記憶し、

前記入力ビットが情報系列の場合に、前記情報系列の後部ほど、メモリ長が小さい前記ウェイトパターンを用いる

請求項1に記載のLDPC-CC符号化器。

【請求項8】

前記ウェイト制御部は、符号化率が異なるLDPC-CC検査行列に準じたウェイトパターンを複数記憶し、

前記入力ビットが情報系列の場合に、前記情報系列の終端ほど、符号化率が低い前記ウェイトパターンを用いる

請求項1に記載のLDPC-CC符号化器。

【請求項9】

前記ウェイト制御部は、メモリ長が異なるLDPC-CC検査行列に準じたウェイトパターンを複数記憶し、

前記入力ビットの先頭部および前記入力ビットの後部の符号化ほど、メモリ長が小さい前記ウェイトパターンを用いる

請求項1に記載のLDPC-CC符号化器。

【請求項10】

前記ウェイト制御部は、複数の多項式に準じたウェイトパターンを複数記憶し、

前記入力ビットの第1ビット目の符号化に、前記複数の多項式のうちメモリ長が最小の多項式に準じた前記ウェイトパターンを用い、前記入力ビットの最終ビットの符号化に、前記複数の多項式のうちメモリ長が最小の多項式に準じた前記ウェイトパターンを用いる

請求項1に記載のLDPC-CC符号化器。

【請求項11】

前記入力ビットの数及び前記多項式の数に応じた数の既知ビットを、前記入力ビットの後部に付加する調節ビット付加部、をさらに具備する

請求項10に記載のLDPC-CC符号化器。

【請求項12】

前記ウェイト制御部は、前記入力ビットの最終ビットの符号化に、前記複数の多項式のうちメモリ長が最小のLDPC-CC検査行列に準じたウェイトパターンを用いる

請求項10に記載のLDPC-CC符号化器。

【請求項13】

低密度パリティ検査畳み込み符号(LDPC-CC: Low-Density Parity-Check Convolutional Code)

10

20

30

40

50

lutional Codes) を復号する LDPC-CC 復号器であって、

LDPC-CC 検査行列に準じた複数のウェイトパターンを備える行列演算器と、
ウェイトパターン切り替えタイミングに基づいて前記複数のウェイトパターンを切り替える切替部と、
を具備する LDPC-CC 復号器。

【請求項 14】

前記切替部は、受信対数尤度比の数をカウントし、当該カウント値と送信情報系列長とを用いて、前記ウェイトパターン切り替えタイミングを算出するウェイト切り替えタイミング計算部、を具備する

請求項 13 に記載の LDPC-CC 復号器。

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、LDPC-CC (Low-Density Parity-Check Convolutional Code、低密度パリティ検査畳み込み符号) 符号化器、送信装置及び LDPC-CC 復号器に関し、LDPC-CC 符号化を用いて誤り訂正符号化を施す LDPC-CC 符号化器及び LDPC-CC 復号器に関する。

【背景技術】

【0002】

近年、実現可能な回路規模で高い誤り訂正能力を発揮する誤り訂正符号として、低密度パリティ検査 (LDPC: Low-Density Parity-Check) 符号に注目が集まっている。LDPC 符号は、その誤り訂正能力の高さと、実装の容易さから IEEE 802.11n の高速無線 LAN (Local Area Networks) システムやデジタル放送システムなどの誤り訂正符号化方式に採用されている。

20

【0003】

LDPC 符号は低密度な (行列中に含まれる 1 の要素数が 0 の要素数に比べて大幅に少ない) パリティ検査行列で定義される誤り訂正符号である。LDPC 符号は、検査行列の列数 N と等しいブロック長を持つブロック符号である。

【0004】

しかし、現在の通信システムの多くは、イーサネット (登録商標) のように可変長のパケットやフレームに基づいて通信を行うという特徴がある。このようなシステムにブロック符号である LDPC 符号を適用する場合、例えば、可変長なイーサネット (登録商標) のフレームに対して固定長の LDPC 符号のブロックをどのように対応させるかといった課題が発生する。LDPC 符号が採用されている無線 LAN の規格である IEEE 802.11n では、送信情報系列にパディングやパンクチャなどを適用して、送信情報系列の長さ、LDPC 符号のブロック長の調節を行っている。しかし、パディング、パンクチャにより符号化率の変化や冗長な系列の送信が必要になるという問題がある。

30

【0005】

このようなブロック符号の LDPC 符号 (以下「LDPC-BC: Low-Density Parity-Check Block Code」と表記する) に対して、任意の長さの情報系列に対しての符号化・復号化が可能な LDPC-CC の検討が行われている (非特許文献 1 参照)。

40

【0006】

LDPC-CC は、低密度なパリティ検査行列により定義される畳み込み符号である。

【0007】

図 43 に、一例として、符号化率 $R = 1/2 (= b/c)$ の LDPC-CC のパリティ検査行列 $H_{[0, n]}^T$ を示す。

【0008】

LDPC-CC において、検査行列 $H_{[0, n]}^T$ の要素 $h_1^{(m)}(t)$ 、 $h_2^{(m)}(t)$ は、0 または 1 をとる。また、検査行列 $H_{[0, n]}^T$ に含まれる $h_1^{(m)}(t)$ 、 $h_2^{(m)}(t)$ 以外の要素は全て 0 である。同図中、 M は、LDPC-CC におけるメモリ長を表

50

し、 n は、送信情報系列の長さを表す。図 4 3 に示されるように、LDPC-CC の検査行列は、行列の対角項とその近辺の要素にのみ 1 が配置されており、行列の左下及び右上の要素はゼロであり、平行四辺形の行列であるという特徴がある。

【0009】

ここで、符号化率 $R = 1/2 (= b/c)$ の例を示すと、 $h_1^{(0)}(t) = 1$ 、 $h_2^{(0)}(t) = 1$ であるとき、LDPC-CC の符号化は、図 4 3 の検査行列 $H_{[0,n]}^T$ に従って、式 (1) 及び式 (2) により行われる。

【0010】

【数 1】

$$v_{1,t} = u_t \quad \dots (1) \quad 10$$

【数 2】

$$v_{2,t} = \sum_{i=0}^M h_1^{(i)}(t)u_{t-i} + \sum_{i=1}^M h_2^{(i)}(t)v_{2,t-i} \quad \dots (2)$$

なお、 u_t は、送信情報系列を表し、 $v_{1,t}$ 、 $v_{2,t}$ は送信符号語系列を表す。

【0011】

図 4 4 に、式 (1) 及び式 (2) を実行する LDPC-CC 符号化器の要部構成の一例を示す。図 4 4 に示すように、LDPC-CC 符号化器 10 は、シフトレジスタ 11-1 ~ 11-M、14-1 ~ 14-M、ウェイト乗算器 12-0 ~ 12-M、13-0 ~ 13-M、mod 2 加算 (排他的論理和演算) 器 15、ビット数カウンタ 16、及びウェイト制御部 17 を備えて構成される。 20

【0012】

シフトレジスタ 11-1 ~ 11-M、14-1 ~ 14-M は、それぞれ $v_{1,t-i}$ 、 $v_{2,t-i}$ ($i = 0, \dots, M$) を保持するレジスタであり、次の入力が入ってくるタイミングで、保持している値を右隣のシフトレジスタに送出し、左隣のシフトレジスタから送われてきた値を新たに保持する。

【0013】

ウェイト乗算器 12-0 ~ 12-M、13-0 ~ 13-M は、ウェイト制御部 17 から出力される制御信号に従って、ウェイト値を 0 又は 1 に切り替える。ウェイト制御部 17 は、ビット数カウンタ 16 から出力されるカウント数と、ウェイト制御部 17 内に保持している検査行列に準じたウェイトパターンとに基づいて、そのタイミングにおける $h_1^{(m)}(t)$ 、 $h_2^{(m)}(t)$ の値をウェイト乗算器 12-0 ~ 12-M、13-0 ~ 13-M に送出する。mod 2 加算器 15 は、ウェイト乗算器 12-0 ~ 12-M、13-0 ~ 13-M の出力に対し mod 2 加算を行い、 $v_{2,t}$ を算出する。ビット数カウンタ 16 は、入力された送信情報系列のビット数をカウントする。 30

【0014】

このような構成を採ることで、LDPC-CC 符号化器 10 は、検査行列に従った LDPC-CC の符号化を行うことができる。 40

【0015】

LDPC-CC 符号化器は、生成行列の乗算を行う符号化器の回路や、後退代入法または前方代入法に基づく演算を行う LDPC-BC 符号化器に比べ、非常に簡易な回路で実現できるという特徴がある。また、LDPC-CC は畳み込み符号であるため、送信情報系列を固定長のブロックに区切って符号化する必要がなく、任意の長さの情報系列を符号化できる。

【0016】

LDPC-CC の復号には、LDPC-BC と同様に検査行列に基づいて Sum-product アルゴリズムを適用することができる。そのため、BCJR (Bahl, Cocke, Jeinek, Raviv) アルゴリズムや、ビタビアルゴリズムのような最尤系列推定に基づく復号アルゴリズム 50

ムを使用する必要がなく、低処理遅延で復号処理を完了できる。さらに、非特許文献 1 には、平行四辺形の形に 1 が配置されているという検査行列の形を活かしたパイプライン型の復号アルゴリズムが提案されている。

【 0 0 1 7 】

L D P C - C C と L D P C - B C の復号特性を復号器の回路規模が同等になるパラメータで比較した場合、L D P C - C C の復号特性の方が優れるということが示されている。(非特許文献 1 参照)

【 0 0 1 8 】

L D P C - C C において、任意の長さ n で符号化を終了した場合、受信側の復号器において受信符号語系列を復号する際に、Sum-product 復号における後部の $2M$ ビットの確率伝搬をその他のビットと同等にするためには、 n 以降の送信情報系列を符号化した符号語及び符号化終了時のシフトレジスタの状態が必要になる。

10

【 0 0 1 9 】

しかし、送信情報系列を単純に符号化しただけでは、符号化終了時点の符号化器のシフトレジスタの状態が送信情報系列に依存するため、受信側で復号する際にその状態を一意に決定することが難しい。

【 0 0 2 0 】

このような状況で、受信側で受信符号語に基づいて復号処理を行うと、復号後に得られる受信情報系列の終わりの方、特に後部 $2M$ ビットに誤りが増えるという現象が起こる。

【 0 0 2 1 】

このような誤りを避けるために、符号化の終了状態を一意に決める終端処理(ターミネーション)を送信情報系列に施すことが必要となる。

20

【 0 0 2 2 】

I E E E 8 0 2 . 1 1 a 準拠の畳み込み符号では、送信情報系列の後部にテイルビットと呼ばれる符号化器のシフトレジスタと同数(6個)の0ビットを付加して符号化することで終端処理を行う。このようにすれば、テイルビットが入力し終わった時点で、符号化器のシフトレジスタの状態をオールゼロにすることができる。なお、テイルビットを入力した際に出力される符号語は、受信側で復号処理を行う際に必要になるので、送信符号語とともに受信側に送信される。

【 0 0 2 3 】

L D P C - C C の場合、式(1)で示したように、符号語 $v_{2,t}$ を求めるには、過去の M 時刻における符号語 $v_{2,t-i}$ が必要になるため、L D P C - C C 符号化器には過去 M 時刻の符号語 $v_{2,t-i}$ を保持するシフトレジスタが備えられている。送信情報系列を保持するレジスタは、送信情報系列の終端を長さ M のオールゼロの系列にする(ターミネーション)ことでオールゼロ状態にすることができるが、この終端処理だけでは、符号語 $v_{2,t-i}$ を保持するシフトレジスタをオールゼロ状態にすることが困難であるという問題がある。

30

【 0 0 2 4 】

非特許文献 2 において、送信情報系列の後部にオールゼロではないターミネーション系列を付加した後に符号化することで、符号化終了時のシフトレジスタの状態をオールゼロにする終端処理が提案されている。

40

【 0 0 2 5 】

非特許文献 2 で提案されている終端処理では、送信符号語系列を式(3)の様に定義する。式(3)は、符号化率 $R = 1/2$ の場合の例である。式(3)において、 $v_{1 \times 2n}$ は、長さ n の情報系列を畳み込み符号化することによって得られる、長さ $2n$ の符号語系列、 $x_{1 \times 2L}$ は、長さ L のターミネーション系列を符号化することで得られるターミネーション符号語系列、 $0_{1 \times 2M}$ は長さ $2M$ の 0 系列である。

【数 3】

$$\begin{bmatrix} v_{1 \times 2n} & x_{1 \times 2L} & 0_{1 \times 2M} \end{bmatrix} H'_{2(n+L+M) \times (n+L+M)} = 0_{1 \times (n+L+M)} \quad \dots (3)$$

【0026】

ここで、ターミネーション系列 $x_{1 \times 2L}$ は、式(4)及び式(5)により決定される。

【数 4】

$$\begin{bmatrix} v_{1 \times 2n} & x_{1 \times 2L} & 0_{1 \times 2M} \end{bmatrix} \begin{bmatrix} A_{2n \times n} & B_{2n \times (L+M)} \\ 0_{2L \times n} & D_{2L \times (L+M)} \\ 0_{2M \times n} & F_{2M \times (L+M)} \end{bmatrix} = 0_{1 \times (n+L+M)} \quad \dots (4) \quad 10$$

【数 5】

$$x_{1 \times 2L} D_{2L \times (L+M)} = v_{1 \times 2n} B_{2n \times (L+M)} = \beta \quad \dots (5)$$

このようなターミネーション系列を付加した送信符号語系列を LDPC-CC 符号化器で符号化することで、シフトレジスタの状態をオールゼロ状態にすることができる。送信側の通信装置はこのようにして終端処理された送信符号語を受信装置に対して送信することで、受信側の復号器は、符号化終了時のシフトレジスタの状態を一意に決定することができ、所望の性能で誤り訂正復号を行う。 20

【非特許文献 1】Alberto Jimenez Felstorum, and Kamil Sh.Zigangirov, "Time-Varying Periodic Convolutional Codes With Low-Density Parity-Check Matrix.", IEEE Transactions on Information Theory, Vol.45, No.6, pp.2181-2191, September, 1999.

【非特許文献 2】Zhengang Chen, Stephen Bates, and Ziaodai Dong, "Low-Density Parity-Check Convolutional Codes Applied to Packet Based Communication Systems", Proceeding of IEEE Globecom 2005, pp.1250-1254.

【非特許文献 3】Stephen Bates, Duncan G. Elliott, and Ramkrishna Swamy, "Termination Sequence Generation Circuits for Low-Density Parity-Check Convolutional Codes," IEEE Transaction on Circuits and Systems-I:Regular Papers, vol.53, no.9, pp.1909-1917, September 2006 30

【非特許文献 4】S. Lin, D. J. Jr., Costello, "Error control coding : Fundamentals and applications," ,582-598, Prentice-Hall.

【非特許文献 5】R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. J. Costello Jr., "LDPC block and convolutional codes based on circulant matrices," IEEE Trans. Inform. Theory, vol.50, no.12, pp.2966-2984, Dec. 2004.

【非特許文献 6】G. Richter, M. Kaupper, and K. Sh. Zigangirov, "Irregular low-density parity-Check convolutional codes based on protographs," Proceeding of IEEE ISIT 2006, pp1633-1637.

【非特許文献 7】A. Pusane, R. Smarandache, P. Vontobel, and D. J. Costello Jr., "On deriving good LDPC convolutional codes from QC LDPC block codes," Proc. of IEEE ISIT 2007, pp.1221-1225, June 2007. 40

【非特許文献 8】Howard H.MAand Jack K.Wolf, On "Tail Bitin Convolutional Codes", IEEE Transactions on communications, vol.COM-34, No.2, pp.104-111, February 1986

【発明の開示】

【発明が解決しようとする課題】

【0027】

しかしながら、前記従来の構成では、送信するターミネーション系列長として $2L$ ビット ($L > M$) の系列が必要であり、冗長な信号系列を送出することによるオーバーヘッド量の増加や伝送効率の劣化が問題となる。例えば、メモリ長 $M = 200$ の LDPC-CC 50

を使って、8000ビットの送信情報系列を送信するとき、ターミネーションのために400ビット以上すなわち送信情報系列の5%以上もの冗長ビットを送信しなければならない。また、ターミネーション系列送信による伝送効率の劣化は、送信情報系列長が短いときや、符号化率が高いとき、メモリ長Mが大きいときにより顕著になってくる。

【0028】

本発明はかかる点に鑑みてなされたものであり、LDPC-CC符号化・復号化に必要なターミネーション系列の量を削減し、伝送効率の劣化を抑圧しつつ、誤り訂正符号化・復号化を行うことができるLDPC-CC符号化器及びLDPC-CC復号器を提供することを目的とする。

【課題を解決するための手段】

10

【0029】

本発明のLDPC-CC符号化器は、複数のシフトレジスタと、前記シフトレジスタの出力にウェイトを乗算する複数のウェイト乗算部と、前記複数のウェイト乗算部の出力を排他的論理和する排他的論理和演算器と、符号化する入力ビットのビット数をカウントするビット数カウンタと、前記ビット数に応じて前記複数のウェイト乗算部のウェイトを制御するウェイト制御部と、を具備する構成を採る。

【0030】

本発明のLDPC-CC符号化器の一つの態様は、前記ウェイト制御部は、LDPC-CC検査行列に準じた第1のウェイトパターンと、前記LDPC-CC検査行列を変形した検査行列に準じた第2のウェイトパターンと、を記憶し、前記入力ビットが情報系列の場合に、前記第1のウェイトパターンを用い、前記入力ビットがターミネーション系列の場合に、前記第2のウェイトパターンを用いる構成を採る。

20

【0031】

この構成によれば、LDPC-CC検査行列を用いたLDPC-CC符号化を行うことができる。また、入力ビットが情報系列とターミネーション系列との場合で、シフトレジスタのウェイト値を切り替えることができるので、入力ビットがターミネーション系列の場合に、送信符号語系列の値に関わらず、送信符号語系列と乗算されるウェイト値を0にして、終端処理を完了することができ、送信するターミネーション系列を削減することができる。

【発明の効果】

30

【0032】

本発明によれば、LDPC-CC符号化・復号化に必要なターミネーション系列の量を削減し、伝送効率の劣化を抑圧しつつ、誤り訂正符号化・復号化を行うことができる。

【発明を実施するための最良の形態】

【0033】

以下、本発明の実施の形態について、図面を参照して詳細に説明する。

【0034】

(実施の形態1)

本実施の形態では、LDPC-CCにおけるターミネーション系列の数を少なくするため、LDPC-CC検査行列のうち、送信情報系列の後部Mビットに対応する行列要素を変形させた検査行列、及び当該検査行列を用いてLDPC-CC符号化する符号化器の構成について説明する。

40

【0035】

図1に、LDPC-CCの検査行列の一例を示す。図1の検査行列100は、メモリ長M=5、符号化率1/2、送信情報系列長nの場合の例である。なお、図1では、簡単のため、検査行列100のうち、送信符号語系列の終端部分及びその後続くターミネーション系列に対応する部分のみを抽出して示している。

【0036】

検査行列100において、各行は、送信符号語系列、ターミネーション系列、及びゼロ系列の各ビットに対応する。また、各列 p_1 、 p_2 、...、 p_{18} は、パリティ検査式に相

50

当する。なお、説明を簡単にするため右側の列から順にインデックスを付している。LDPC-CCでは、各列において、1が配置されている行に対応する送信符号語ビットのmod 2加算結果(排他的論理和演算結果)がゼロになるような符号化が行われる。図1の検査行列100は、送信符号語系列 $v_{1,t}, v_{2,t}$ に送信情報系列 u_t がそのままの形で含まれる組織化符号の例であり、送信符号語ビット $v_{1,t}, v_{2,t}$ は、式(1)で表される。

【0037】

ここで、ターミネーションが必要な理由について説明する。ターミネーションを行わない場合、すなわち、LDPC-CC検査行列を $v_{2,n}$ の行で終了した場合、検査行列は、図2に示すようになる。図2の検査行列200を用いて、復号側でSum-product復号を行う場合を考える。Sum-product復号では、検査行列200の行方向には対数尤度比の加算処理による繰り返し符号の復号を行い、列方向には対数尤度比の $\tanh(\cdot)$ を取った値の乗算処理による単一パリティ検査符号の復号を行う。このとき、検査行列200の $v_{2,n}$ の行に着目すると、 $v_{2,n}$ の行には1が配置されている列が1列(p_1)しかない。 $v_{1,n}, v_{1,n-1}, v_{2,n-1}, v_{2,n-2}, v_{2,n-3}, v_{2,n-4}$ の行についても同様である。このような行では、Sum-product復号における繰り返し符号の復号処理時に十分な符号化ゲインが得られないため、これらの符号語ビットが影響を与える他の符号語ビットの復号特性に対しても悪影響を及ぼしてしまう。この結果、復号後の受信情報系列の後部にビット誤りが多く発生してしまう。

【0038】

これに対し、送信符号語系列 $v_{1,n}, v_{2,n}$ の後に、ターミネーション系列及びゼロ系列を追加する場合には、図1に示すように、検査行列100において、 $v_{2,n}$ の行には、他の $v_{2,x}$ の行と同数だけ1が配置されている列が存在するので、送信符号語系列 $v_{1,n}, v_{2,n}$ をSum-product復号を用いて繰り返し復号する際に、十分な符号化ゲインを得ることができる。

【0039】

また、符号語ビット $v_{2,n}$ を含むパリティ検査方程式(p_6, p_{11})に含まれる $v_{2,n}$ 以外の符号語ビットに対しても、ターミネーション系列及びゼロ系列の付加により、十分な符号化ゲインが得られるようになるため、復号後の受信情報系列の後部にビット誤りが発生することがなくなる。しかし、この場合、 $2n$ ビットの送信符号語系列に加え、 $2L$ ビット(この例では $2M=10$ ビット)のターミネーション系列を送信する必要があり、伝送効率の劣化が問題となる。

【0040】

そこで、本実施の形態では、検査行列100のうち、送信情報系列の後部 M ビットに対応する行列要素を変形させた検査行列を用いて、LDPC-CC符号化を行う。以下、詳細に説明する。

【0041】

図3に、本実施の形態における検査行列300を示す。図3の検査行列300は、図1の検査行列100に対し、 $x_{2,1}, x_{2,2}, x_{2,3}, x_{2,4}, x_{2,5}$ の符号化に用いるパリティ検査式($p_6, p_7, p_8, p_9, p_{10}$)の列において、 $v_{2,t}$ に対応する行に配置されている1を0に変更したものである。具体的には、検査行列300は、検査行列100の $v_{2,n}, v_{2,n-1}, v_{2,n-2}, v_{2,n-3}, v_{2,n-4}$ に対応する行における一番右側にある1を0に変更して得られた行列である。なお、検査行列300の $v_{2,n-5}$ より前の送信符号語ビットに関する行は、検査行列100と同じである。

【0042】

検査行列300では、 $v_{2,n}, v_{2,n-1}, v_{2,n-2}, v_{2,n-3}, v_{2,n-4}$ に対応する行における一番右側が0となるため、この検査行列300を用いた場合、 $x_{2,1}, x_{2,2}, x_{2,3}, x_{2,4}, x_{2,5}$ の符号化に、 $v_{2,t}$ の値が不要となり、送信情報系列 $v_{1,t}(=u_t)$ 及び $x_{1,x}$ のみが必要となる。そのため、従来のように、 $v_{1,t}, v_{2,t}$ 双方のシフトレジスタをオールゼロ状態にする必要はなく、 $v_{1,t}$ のシフトレジスタのみをオールゼロ

10

20

30

40

50

状態にするだけでよい。 $v_{1,t}$ のシフトレジスタをオールゼロ状態にするためには、従来通り、 $x_{1,1}, x_{1,2}, x_{1,3}, x_{1,4}, x_{1,5}$ は全て 0 として符号化器に入力すればよい。

【0043】

さらに、 $x_{1,1} \sim x_{1,5}$ を全て 0 にすることを送信側及び受信側で予め取り決めておけば、 $x_{1,1} \sim x_{1,5}$ を実際に受信機に送信する必要はなく、符号化によって得られた $x_{2,m}$ のみを送信すればよい。このとき、受信機では、 $x_{1,m}$ に対応するビットの対数尤度比 (LLR: Log-Likelihood Ratio) を (無限大) として復号する。

【0044】

このようにすることで、送信するターミネーション系列を従来の 2L ビットから少なくとも M ビットにまで削減することができる。

10

【0045】

例えば、検査行列 300 を用いる場合には、 $x_{2,1}, x_{2,2}, x_{2,3}$ の 3 ビットだけを送信すれば、 $v_{1,t}, v_{2,t}$ に関する全てのパリティ検査式に必要な符号語ビットを受信側で得ることができる。これに対し、検査行列 100 を用いる場合には、10 ビットのターミネーション系列 $x_{1,1} \sim x_{1,5}, x_{2,1} \sim x_{2,5}$ を送信する必要がある。つまり、検査行列 300 を用いる場合には、ターミネーション系列の送信量を、検査行列 100 を用いる場合の 3 割にまで削減できるようになる。

【0046】

次に、図 3 の検査行列 300 を用いて LDPC-CC 符号化を行う LDPC-CC 符号化器の構成を図面を参照しながら説明する。

20

【0047】

図 4 は、本実施の形態における LDPC-CC 符号化器の要部構成を示すブロック図である。

【0048】

図 4 において、LDPC-CC 符号化器 400 は、シフトレジスタ 410-1 ~ 410-M, 440-1 ~ 440-M、ウェイト乗算器 420-0 ~ 420-M, 430-0 ~ 430-M、mod 2 加算 (排他的論理和演算) 器 450、ビット数カウンタ 460、ウェイト制御部 470、パンクチャ部 480 を備えて構成される。

【0049】

シフトレジスタ 410-1 ~ 410-M, 440-1 ~ 440-M は、それぞれ $v_{1,t-i}, v_{2,t-i}$ ($i = 0, \dots, M$) を保持するレジスタであり、次の入力が入ってくるタイミングで、保持している値を右隣のシフトレジスタに送出し、左隣のシフトレジスタから送出されてきた値を新たに保持する。なお、n は、送信情報系列 u_t の送信情報系列長を表す。

30

【0050】

ウェイト乗算器 420-0 ~ 420-M, 430-0 ~ 430-M は、ウェイト制御部 470 から出力される制御信号に従って、ウェイト値を 0 又は 1 に切り替える。

【0051】

mod 2 加算器 450 は、ウェイト乗算器 420-0 ~ 420-M, 430-0 ~ 430-M の出力に対し mod 2 加算を行い、 $v_{2,t}$ を算出する。

40

【0052】

ビット数カウンタ 460 は、入力された送信情報系列及びターミネーション系列のビット数をカウントし、カウントビット数をウェイト制御部 470 に出力する。

【0053】

ウェイト制御部 470 は、ビット数カウンタ 460 から出力されるカウントビット数と、送信情報系列長 n と、ウェイト制御部 470 内に保持している検査行列 300 に準じたウェイトパターンに基づいて、そのタイミングにおける検査行列要素 $h_1^{(m)}(t), h_2^{(m)}(t)$ の値をウェイト乗算器 420-0 ~ 420-M, 430-0 ~ 430-M に送出する。

【0054】

50

図5にウェイト制御部470の構成図を示す。ウェイト制御部470は、セクタ471、474、ウェイトパターン記憶部472、及びウェイトパターン記憶部473から構成される。

【0055】

セクタ471は、カウントビット数及び送信情報系列長 n を入力し、カウントビット数 送信情報系列長 n の場合、カウントビット数をウェイトパターン記憶部472に送出する。一方、カウントビット数 $>$ 送信情報系列長 n の場合、セクタ471は、カウントビット数をウェイトパターン記憶部473に送出する。

【0056】

ウェイトパターン記憶部472は、ウェイトパターン475で示されるウェイトパターンを保持しており、カウントビット数の増加に伴い周期的に h_{a1} 、 h_{a2} 、 h_{a3} 、 h_{a4} をセクタ474に出力する。なお、ウェイトパターン475は、検査行列100の行列要素 $h_1^{(m)}(t)$ 、 $h_2^{(m)}(t)$ ($m=0, \dots, M$)である。

10

【0057】

ウェイトパターン記憶部473は、ウェイトパターン476で示されるウェイトパターンを保持しており、カウントビット数の増加に伴い周期的に h_{b1} 、 h_{b2} 、 h_{b3} 、 h_{b4} をセクタ474に出力する。なお、ウェイトパターン476は、検査行列100の要素を変形した検査行列300の行列要素 $h_1^{(m)}(t)$ 、 $h_2^{(m)}(t)$ ($m=0, \dots, M$)である。

【0058】

セクタ474は、ウェイトパターン記憶部472から入力される行列要素 h_{a1} 、 h_{a2} 、 h_{a3} 、 h_{a4} 、又は、ウェイトパターン記憶部473から入力される行列要素 h_{b1} 、 h_{b2} 、 h_{b3} 、 h_{b4} をウェイト乗算器420-0~420-M、430-0~430-Mに出力する。

20

【0059】

すなわち、セクタ474は、セクタ471と連動して、ビット数カウントと送信情報系列長との比較結果に応じて、入力ビットが送信情報系列の場合、ウェイトパターン記憶部472に記憶される検査行列100に準じたウェイトパターン475の行列要素を、ウェイト乗算器420-0~420-M、430-0~430-Mに出力する。一方、入力ビットがターミネーション系列の場合、セクタ474は、ウェイトパターン記憶部473に記憶される検査行列100を変形した検査行列300に準じたウェイトパターン476の行列要素を、ウェイト乗算器420-0~420-M、430-0~430-Mに出力する。

30

【0060】

パンクチャ部480は、送信符号語系列 $v_{1,1} \sim v_{1,n}$ 、 $x_{1,1} \sim x_{1,L}$ から、ターミネーション系列 $x_{1,1} \sim x_{1,L}$ をパンクチャする。

【0061】

以下、上述のように構成されたLDPC-CC符号化器400の動作について説明する。

【0062】

送信情報系列 $u_1 \sim u_n$ 及びターミネーション系列 $x_{1,1} \sim x_{1,L}$ は、シフトレジスタ410-1、ウェイト乗算器420-0、及びビット数カウンタ460に順次出力されるとともに、送信情報系列 u_t は、送信符号語系列 $v_{1,t}$ として、パンクチャ部480に出力される。

40

【0063】

ビット数カウンタ460では、入力される送信情報系列 $u_1 \sim u_n$ 及びターミネーション系列 $x_{1,1} \sim x_{1,L}$ のビット数がカウントされ、得られたカウントビット数は、ウェイト制御部470に出力される。

【0064】

ウェイト制御部470では、カウントビット数と送信情報系列長 n との比較結果に応じ

50

て、ウェイトパターン475又はウェイトパターン476のいずれか一方が選択されて、選択されたウェイトパターンの行列要素が、ウェイト乗算器420-0~420-M, 430-0~430-Mに出力される。

【0065】

具体的には、カウントビット数<送信情報系列長、つまり、入力ビットが送信情報系列の場合、ウェイト乗算器420-0~420-M, 430-0~430-Mには、検査行列100に準じたウェイトパターン475の行列要素が出力される。ウェイトパターン475は、 $v_{2,n}, v_{2,n-1}, v_{2,n-2}, v_{2,n-3}, v_{2,n-4}$ に対応する行の一番右側に1が配置されている従来のLDPC-CC符号化に用いられるパターンと同一のパターンである。

10

【0066】

一方、カウントビット数>送信情報系列長、つまり、入力ビットがターミネーション系列の場合、ウェイト乗算器420-0~420-M, 430-0~430-Mには、検査行列100を変形した検査行列300に準じたウェイトパターン476の検査行列が出力される。ウェイトパターン476は、ウェイトパターン475に対し、 $v_{2,n}, v_{2,n-1}, v_{2,n-2}, v_{2,n-3}, v_{2,n-4}$ に対応する行における一番右側の1を0に代えたパターンである。すなわち、図5において、検査行列の行の一番右の $h_2^{(5)}$ が0のパターンである。検査行列100の例では、 $h_2^{(1)} \sim h_2^{(4)}$ が0であるので、ウェイトパターン476を用いることにより、ターミネーション系列が入力される場合に、 $v_{2,n}, v_{2,n-1}, v_{2,n-2}, v_{2,n-3}, v_{2,n-4}$ と乗算されるウェイト値が0になり、 $x_{2,1}, x_{2,2}, x_{2,3}, x_{2,4}, x_{2,5}$ の符号化に、 $v_{2,n}, v_{2,n-1}, v_{2,n-2}, v_{2,n-3}, v_{2,n-4}$ が用いられなくなる。この結果、ターミネーション時に $v_{2,t}$ に関するシフトレジスタ440-0~440-Mを0にする必要がなくなり、 $v_{2,t}$ を0にするための冗長ビットが不要となる。

20

【0067】

さらに、パンクチャ部480では、送信符号語系列 $v_{1,1} \sim v_{1,n}, x_{1,1} \sim x_{1,L}$ から、ターミネーション系列 $x_{1,1} \sim x_{1,L}$ がパンクチャされる。このようにすることで、ターミネーション系列を送信することによる伝送効率の劣化を従来方式に比べて削減することができる。

【0068】

以上のように、本実施の形態では、LDPC-CC符号化器400は、複数のシフトレジスタ410-1~410-M, 430-1~430-Mと、シフトレジスタ410-1~410-M, 430-1~430-Mの出力にウェイトを乗算する複数のウェイト乗算器420-0~420-Mと、ウェイト乗算器420-0~420-Mの出力をmod2加算するmod2加算器450と、符号化する入力ビットのビット数をカウントするビット数カウンタ460と、入力ビットのビット数に応じてウェイト乗算器420-0~420-Mのウェイトを制御するウェイト制御部470と、を備えるようにした。これにより、LDPC-CC検査行列を用いたLDPC-CC符号化を行うことができる。

30

【0069】

また、ウェイト制御部470は、LDPC-CC検査行列100に準じたウェイトパターン475と、LDPC-CC検査行列100を変形した検査行列300に準じたウェイトパターン476と、を記憶し、入力ビットが情報系列の場合に、ウェイトパターン475を用い、入力ビットがターミネーション系列の場合に、ウェイトパターン476を用いるようにした。このようにすることで、送信情報系列 u_t が入力される場合は、ウェイトパターン475を使って送信符号語系列 $v_{1,t}, v_{2,t}$ を取得し、ターミネーション系列 $x_{1,m}$ が入力される場合は、 $v_{2,t}$ と乗算されるウェイト値を0にするウェイトパターン476を使って送信符号語系列 $v_{1,t}, v_{2,t}$ を取得することができるので、送信するターミネーション系列を削減することができる。

40

【0070】

また、終端処理のために送信される送信側と受信側とで既知であるゼロ系列($x_{1,1} \sim$

50

$x_{1,L}$)をパンクチャするパンクチャ部480を設けることで、ターミネーション系列を送信することによる伝送効率の低下を抑圧することができる。

【0071】

なお、非特許文献2に開示されるターミネーション方法では、ターミネーション系列 $x_{1 \times 2 L}$ を求めるために別の回路が必要となるのに対し(非特許文献3参照)、本実施の形態では、このような特別な回路を必要とせず終端処理を含んだ符号化処理を完了することができる。

【0072】

(変形例)

図6に、本実施の形態における検査行列の別の例を示す。図6の検査行列500は、 $x_{2,1}, x_{2,2}, x_{2,3}, x_{2,4}, x_{2,5}$ の符号化に用いるパリティ検査式($p_6, p_7, p_8, p_9, p_{10}$)に対応する列において、 $v_{2,t}$ に対応する行の一番右に配置されている1を0に変更したものであり、さらに、 $v_{2,n}, v_{2,n-1}, v_{2,n-2}, v_{2,n-3}, v_{2,n-4}$ に対応する行において、 $x_{2,1}, x_{2,2}, x_{2,3}, x_{2,4}, x_{2,5}$ の符号化に用いるパリティ検査式($p_6, p_7, p_8, p_9, p_{10}$)以外の列に新たに1を配置したものである。図6の矢印で示すように、1が配置されている位置を、 $x_{2,1}, x_{2,2}, x_{2,3}, x_{2,4}, x_{2,5}$ の符号化に用いるパリティ検査式($p_6, p_7, p_8, p_9, p_{10}$)以外の列に移動した形になる。

【0073】

このようにすることで、送信するターミネーション系列を従来の $2L$ ビットから少なくとも L ビットにまで半減することができるという検査行列300と同様の効果に加え、検査行列500では、1が配置されている位置を移動した行では、行方向の1の数(行重み)が変わらないので、Sum-product復号における繰り返し符号の復号処理による符号化ゲインを維持することができる。

【0074】

また、図6の検査行列500では、1を左に移動する数が行毎に同数でない場合の例を示しているが、この限りではなく、1を行毎に同じ数だけ左に移動するようにしても良い。行毎に左に移動する数が等しい場合には、行毎に左に移動する数が同数でない場合に比べ、ウェイト制御部470が保持するウェイトパターンの種類が少なくて済む。

【0075】

また、 $x_{2,1}, x_{2,2}, x_{2,3}, x_{2,4}, x_{2,5}$ の符号化に関わる全ての1を移動させたり、一部の行の1のみ移動させて、その他の行の1は0に変更するのみとするなどしたりしても、ターミネーション系列量の削減による伝送効率の劣化の抑圧という本発明の効果を達成することができる。

【0076】

(実施の形態2)

本実施の形態では、LDPC-CCにおけるターミネーション系列の数を少なくするため、送信情報系列の後部ほど、LDPC-CC符号化のメモリ長 M が小さくなるように設計した検査行列、及び当該検査行列に基づいたLDPC-CC符号化器の構成について説明する。

【0077】

図7に、本実施の形態における検査行列の一例を示す。図7の検査行列600は、符号化率 $R = b/c = 1/2$ 、送信情報系列長 n の場合の例であり、検査行列600は、送信情報系列 u_t のインデックスが n に近づくにつれて、符号化のメモリ長 M が $M = 5, 4, 3$ の順に小さくなっていく点が、図1の検査行列100と異なっている。

【0078】

つまり、図7に示すように、検査行列600を用いる場合には、送信情報系列 $u_1 \sim u_{n-4}$ に対しては、メモリ長 $M = 5$ で符号化し、送信符号語系列 $v_{1,1} \sim v_{1,n-4}, v_{2,1} \sim v_{2,n-4}$ が取得される。送信情報系列 $u_{n-3} \sim u_{n-1}$ に対しては、メモリ長 $M = 4$ で符号化し、送信符号語系列 $v_{1,n-3} \sim v_{1,n-1}, v_{2,n-3} \sim v_{2,n-1}$ が取得される。送信情報

10

20

30

40

50

系列 u_n 及びターミネーション系列 $x_{1,1} \sim x_{1,3}$ に対しては、メモリ長 $M = 3$ で符号化し、 $v_{1,n}$ 、 $v_{2,n}$ 及び $x_{1,1} \sim x_{1,3}$ 、 $x_{2,1} \sim x_{2,3}$ が取得される。

【0079】

上述したように、ターミネーション系列の長さ L は、 $L > M$ を満たす必要がある。したがって、ターミネーションを行う時点で符号化器のメモリ長 M が小さければ、その分ターミネーション系列長 L を短くすることができる。

【0080】

このように、図7に示す検査行列600を用いてLDPC-CC符号化を行うことで、ターミネーション系列の長さを従来に比べより短くすることができ、その結果、ターミネーション系列を送信することによる伝送効率の劣化を抑えることができる。

10

【0081】

ところで、LDPC-CCでは、メモリ長 M が大きければ大きいほど符号化ゲインが得られ、良好な誤り率特性が得られるという特徴がある。そのため、検査行列600のように、送信符号語系列の後部に行くに従ってメモリ長 M を小さくしていくと、後部のビット誤り率が劣化することが予想される。しかし、LDPC-CCでは、適切な終端処理を行うことにより、符号化終了時点のシフトレジスタの値をオールゼロ状態に決定し、ターミネーション系列の後に、復号時に $LLR = 0$ とすることができるゼロ系列を仮定できるので、復号後の受信情報系列後部のビット誤り率特性が、その他の部分より良くなるという特徴がある。

【0082】

20

そのため、検査行列600を用いて、送信符号語系列の後部に行くに従ってメモリ長 M を小さくするLDPC-CC符号化を行う場合に、復号後の受信情報系列の後部のビット誤り率が低下するという問題が生じない。

【0083】

次に、図7の検査行列600を用いてLDPC-CC符号化を行うLDPC-CC符号化器の構成を図面を参照しながら説明する。

【0084】

図8は、本実施の形態におけるLDPC-CC符号化器の要部構成を示すブロック図である。本実施の形態の説明にあたり、図4と同一構成部分には同一符号を付して説明を省略する。図8のLDPC-CC符号化器700は、図4のLDPC-CC符号化器400に対し、バンクチャ部480を削除し、ウェイト制御部470に代え、ウェイト制御部710を備えて構成される。

30

【0085】

ウェイト制御部710は、ビット数カウンタ460から出力されるカウントビット数と、メモリ長切り替えタイミング情報と、ウェイト制御部710内に保持している検査行列600に準じたウェイトパターンとに基づいて、そのタイミングにおける行列要素 $h_1^{(m)}(t)$ 、 $h_2^{(m)}(t)$ の値をウェイト乗算器420-0 ~ 420-M、430-0 ~ 430-Mに送出する。

【0086】

ここで、メモリ長切り替えタイミング情報とは、検査行列600のメモリ長 M を切り替える送信情報系列のインデックスを表す。例えば、3種類のメモリ長 M を使用する場合、メモリ長切り替えタイミング情報は2つの値を取る。つまり、メモリ長として、メモリ長 $M = 5, 4, 3$ の3種類を用いる場合、メモリ長切り替えタイミング情報は、 $M = 5$ から $M = 4$ に切り替えるタイミング情報のインデックスと、 $M = 4$ から $M = 3$ に切り替えるタイミング情報のインデックスとを持っている。

40

【0087】

図9に、メモリ長 M が $M = 5, 4, 3$ の3種類を用いる場合のウェイト制御部710の構成例を示す。図9のウェイト制御部710は、セクタ711、715、ウェイトパターン記憶部712 ~ 714から構成される。以下では、メモリ長 $M = 5$ から $M = 4$ に切り替えるタイミングを示すインデックスをメモリ長切り替えタイミング情報1とし、メモリ

50

長 $M = 4$ から $M = 3$ に切り替えるタイミングを示すインデックスをメモリ長切り替えタイミング情報 2 とする。

【 0 0 8 8 】

セレクタ 7 1 1 は、カウントビット数及びメモリ長切り替えタイミング情報 1, 2 を入力し、カウントビット数 > メモリ長切り替えタイミング情報 1 の場合、ビット数カウントをウェイトパターン記憶部 7 1 2 に送出する。

【 0 0 8 9 】

一方、カウントビット数 > メモリ長切り替えタイミング情報 1、かつ、カウントビット数 > メモリ長切り替えタイミング情報 2 の場合、セレクタ 7 1 1 は、カウントビット数をウェイトパターン記憶部 7 1 3 に送出する。

10

【 0 0 9 0 】

また、セレクタ 7 1 1 は、カウントビット数 > メモリ長切り替えタイミング情報 2 の場合、カウントビット数をウェイトパターン記憶部 7 1 4 に送出する。

【 0 0 9 1 】

ウェイトパターン記憶部 7 1 2 は、ウェイトパターン 7 1 6 で示されるウェイトパターンを保持しており、カウントビット数の増加に伴い周期的に $h_{a1}, h_{a2}, h_{a3}, h_{a4}$ をセレクタ 7 1 5 に出力する。なお、ウェイトパターン 7 1 6 は、メモリ長 $M = 5$ の場合の LDPC - CC 検査行列の行列要素 $h_1^{(m)}(t), h_2^{(m)}(t)$ ($m = 0, \dots, 5$) である。

【 0 0 9 2 】

20

ウェイトパターン記憶部 7 1 3 は、ウェイトパターン 7 1 7 で示されるウェイトパターンを保持しており、ビット数カウントの増加に伴い周期的に h_{b1}, h_{b2}, h_{b3} をセレクタ 7 1 5 に出力する。なお、ウェイトパターン 7 1 6 は、メモリ長 $M = 4$ の場合の LDPC - CC 検査行列の行列要素 $h_1^{(m)}(t), h_2^{(m)}(t)$ ($m = 0, \dots, 4$) と、 $h_1^{(5)} = 0, h_2^{(5)} = 0$ とである。メモリ長 $M = 4$ の場合、 h_{b1}, h_{b2}, h_{b3} のウェイトパターンの要素数は 1 0 であるが、符号化器 7 0 0 には、メモリ長 $M = 5$ に対応できるように 1 2 個のウェイト乗算器 $4 2 0 - 0 \sim 4 2 0 - M, 4 3 0 - 0 \sim 4 3 0 - M$ が備えられている。そのため、ウェイトパターン 7 1 7 では、 $h_1^{(5)}, h_2^{(5)}$ のウェイト要素が、どのパターンにおいても 0 となっている。

【 0 0 9 3 】

30

ウェイトパターン記憶部 7 1 4 は、ウェイトパターン 7 1 8 で示されるウェイトパターンを保持しており、カウントビット数の増加に伴い周期的に $h_{c1}, h_{c2}, h_{c3}, h_{c4}, h_{c5}, h_{c6}, h_{c7}$ をセレクタ 7 1 5 に出力する。なお、ウェイトパターン 7 1 6 は、メモリ長 $M = 3$ の場合の LDPC - CC 検査行列の行列要素 $h_1^{(m)}(t), h_2^{(m)}(t)$ ($m = 0, \dots, 3$) と、 $h_1^{(4)} = 0, h_2^{(4)} = 0, h_1^{(5)} = 0, h_2^{(5)} = 0$ とである。 $M = 4$ の場合と同様に、ウェイトパターン 7 1 8 では、 $h_1^{(4)}, h_2^{(4)}, h_1^{(5)}, h_2^{(5)}$ のウェイト要素が、どのパターンにおいても 0 となっている。

【 0 0 9 4 】

セレクタ 7 1 5 は、ウェイトパターン記憶部 7 1 2 から入力される行列要素 $h_{a1}, h_{a2}, h_{a3}, h_{a4}$ 、ウェイトパターン記憶部 7 1 3 から入力される行列要素 h_{b1}, h_{b2}, h_{b3} 、又は、ウェイトパターン記憶部 7 1 4 から行列要素 $h_{c1}, h_{c2}, h_{c3}, h_{c4}, h_{c5}, h_{c6}, h_{c7}$ をウェイト乗算器 $4 2 0 - 0 \sim 4 2 0 - M, 4 3 0 - 0 \sim 4 3 0 - M$ に出力する。

40

【 0 0 9 5 】

すなわち、セレクタ 7 1 5 は、セレクタ 7 1 1 と連動して、ビット数カウントとメモリ長切り替えタイミング情報 1, 2 との比較結果に応じて、ビット数カウント > メモリ長切り替えタイミング情報 1 の場合、ウェイトパターン記憶部 7 1 2 に記憶されるメモリ長 $M = 5$ の検査行列に準じたウェイトパターン 7 1 6 の行列要素を、ウェイト乗算器 $4 2 0 - 0 \sim 4 2 0 - M, 4 3 0 - 0 \sim 4 3 0 - M$ に出力する。

【 0 0 9 6 】

50

一方、ビット数カウント > メモリ長切り替えタイミング情報 1、かつ、ビット数カウント < メモリ長切り替えタイミング情報 2 の場合、セクタ 715 は、ウェイトパターン記憶部 713 に記憶されるメモリ長 $M = 4$ の検査行列に準じたウェイトパターン 717 の行列要素を、ウェイト乗算器 $420 - 0 \sim 420 - M$ 、 $430 - 0 \sim 430 - M$ に出力する。

【0097】

また、ビット数カウント > メモリ長切り替えタイミング情報 2 の場合、セクタ 715 は、ウェイトパターン記憶部 714 に記憶されるメモリ長 $M = 3$ の検査行列に準じたウェイトパターン 718 の行列要素を、ウェイト乗算器 $420 - 0 \sim 420 - M$ 、 $430 - 0 \sim 430 - M$ に出力する。

10

【0098】

以下、上述のように構成された LDPC - CC 符号化器 700 の動作について説明する。

【0099】

送信情報系列 $u_1 \sim u_n$ 及びターミネーション系列 $x_{1,1} \sim x_{1,L}$ は、シフトレジスタ 410 - 1、ウェイト乗算器 $420 - 0$ 、及びビット数カウンタ 460 に順次出力される。

【0100】

ビット数カウンタ 460 では、入力される送信情報系列 $u_1 \sim u_n$ 及びターミネーション系列 $x_{1,1} \sim x_{1,L}$ のビット数がカウントされ、得られたカウントビット数は、ウェイト制御部 710 に出力される。

20

【0101】

ウェイト制御部 710 では、カウントビット数とメモリ長切り替えタイミング情報との比較結果に応じて、ウェイトパターン 716、ウェイトパターン 717、又はウェイトパターン 718 のいずれかが選択されて、選択されたウェイトパターンの行列要素が、ウェイト乗算器 $420 - 0 \sim 420 - M$ 、 $430 - 0 \sim 430 - M$ に出力される。

【0102】

具体的には、カウントビット数 < メモリ長切り替えタイミング情報 1 の場合、ウェイト乗算器 $420 - 0 \sim 420 - M$ 、 $430 - 0 \sim 430 - M$ には、メモリ長 $M = 5$ の検査行列に準じたウェイトパターン 716 の行列要素が出力される。

【0103】

カウントビット数 > メモリ長切り替えタイミング情報 1、かつ、カウントビット数 < メモリ長切り替えタイミング情報 2 の場合、ウェイト乗算器 $420 - 0 \sim 420 - M$ 、 $430 - 0 \sim 430 - M$ には、メモリ長 $M = 4$ の検査行列に準じたウェイトパターン 717 の行列要素が出力される。

30

【0104】

また、カウントビット数 > メモリ長切り替えタイミング情報 2 の場合、ウェイト乗算器 $420 - 0 \sim 420 - M$ 、 $430 - 0 \sim 430 - M$ には、メモリ長 $M = 3$ の検査行列に準じたウェイトパターン 718 の行列要素が出力される。

【0105】

このようにすることで、送信符号語系列が終端に近づくに従い、送信符号語系列を生成するために必要な過去の送信符号語系列のビット数を少なくすることができる。

40

【0106】

以上のように、本実施の形態によれば、ウェイト制御部 710 は、メモリ長が異なる LDPC - CC 検査行列に準じたウェイトパターン 716、717、718 を記憶し、入力ビットが情報系列の場合に、情報系列の終部ほど、メモリ長が小さいウェイトパターンを用いるようにした。ターミネーション系列長 L は、メモリ長 M が小さければ小さいほど、短くすることができるので、情報系列の後部ほど、メモリ長が小さいウェイトパターンを用いるようすることで、ターミネーション系列長 L をより短くすることができ、伝送効率の劣化を抑えることができる。

【0107】

50

なお、本実施の形態では、送信情報系列の後部に近づくに従って、メモリ長を $M = 5, 4, 3$ の順にメモリ長を 1 つずつ小さくしていく場合を例に説明したが、これに限るものではなく、任意のメモリ長に減少したり、メモリ長の減少量を任意にした検査行列を用いても、本発明による、ターミネーション系列送信量削減による伝送効率劣化の抑圧という効果を得ることができる。

【0108】

(実施の形態3)

本実施の形態では、ターミネーション系列の数を少なくしたことによって受信情報系列の後部に発生する伝送誤りを軽減するため、送信情報系列の後部ほど、LDPC-CC符号化の符号化率が低くなるように設計した検査行列、及び当該検査行列に基づいたLDPC-CC符号化器の構成について説明する。

10

【0109】

図10に、本実施の形態における検査行列の一例を示す。図10の検査行列800は、メモリ長 $M = 5$ 、送信情報系列長 n の場合の例であり、検査行列800は、送信符号語系列 u_t のインデックスが大きくなるにつれて、符号化率 R が $R = 1/2, 1/3, 1/4$ の順に低くなっていく点が、従来のLDPC-CC検査行列と異なっている。

【0110】

つまり、図10に示すように、検査行列800を用いる場合には、送信情報 $u_i \sim u_{i+7}$ に対しては、符号化率 $1/2$ で符号化し、送信符号語系列 $v_{1,i} \sim v_{1,i+7}, v_{2,i} \sim v_{2,i+7}$ が取得される。送信情報 $u_{i+8} \sim u_{i+11}$ に対しては、符号化率 $1/3$ で符号化し、送信符号語系列 $v_{1,i+8} \sim v_{1,i+11}, v_{2,i+8} \sim v_{2,i+11}, v_{3,i+8} \sim v_{3,i+11}$ が取得される。送信情報 $u_{i+12} \sim u_{i+15}$ に対しては、符号化率 $1/4$ で符号化し、送信符号語系列 $v_{1,i+12} \sim v_{1,i+15}, v_{2,i+12} \sim v_{2,i+15}, v_{3,i+12} \sim v_{3,i+15}, v_{4,i+12} \sim v_{4,i+15}$ が取得される。

20

【0111】

このように、図10に示す検査行列800を用いてLDPC-CC符号化を行うことで、送信符号語系列の後部に行くにしたがい、符号化率が低いLDPC-CC符号語を生成することができる。符号化率が低いほど、誤り訂正能力が高いため、このようにすることで、全ターミネーション系列を送信しない場合においても、受信情報系列後部に発生するビット誤りを訂正することができるようになる。

30

【0112】

次に、図10の検査行列800を用いてLDPC-CC符号化を行うLDPC-CC符号化器の構成を図面を参照しながら説明する。

【0113】

図11は、本実施の形態におけるLDPC-CC符号化器の要部構成を示すブロック図である。

【0114】

図11のLDPC-CC符号化器900は、シフトレジスタ910-1-1~910-c-M、ウェイト乗算器920-1-1-0~920-c-c-M、mod2加算器930-1~930-c-1、ビット数カウンタ940、ウェイト制御部950を備えて構成される。

40

【0115】

LDPC-CC符号化器900において、シフトレジスタ、ウェイト乗算器、及びビット数カウンタ、及びウェイト制御部は、従来及び上述の実施の形態1~2のシフトレジスタ、ウェイト乗算器、及びビット数カウンタ、及びウェイト制御部と同様であるため説明を省略する。

【0116】

ウェイト制御部950は、ビット数カウンタ940から出力されるカウントビット数と、符号化率切り替えタイミング情報と、ウェイト制御部950内に保持している検査行列に準じたウェイトパターンとに基づいて、そのタイミングにおける検査行列要素 $h_{1,2}^{(m)}$

50

(t) , $h_{2,2}^{(m)}(t)$, ..., $h_{c,c}^{(m)}(t)$ の値をウェイト乗算器 920-1-1-0 ~ 920-c-c-M に送出する。

【0117】

ここで、符号化率切り替えタイミング情報とは、検査行列 800 の符号化率 R を切り替える送信情報系列のインデックスを表す。例えば、3種類の符号長 R を使用する場合、符号長切り替えタイミング情報は2つの値を持つ。つまり、符号長 $R = 1/2, 1/3, 1/4$ を用いる場合、 $R = 1/2$ から $R = 1/3$ に切り替えるタイミング情報のインデックスと、及び、 $R = 1/3$ から $R = 1/4$ に切り替えるタイミング情報のインデックスを持っている。

【0118】

ウェイト制御部 950 の構成及びウェイトパターン切り替え処理は、ウェイト制御部 710 と同様であるので、説明を省略する。

【0119】

以上のように、本実施の形態では、ウェイト制御部 950 は、符号化率が異なる LDPC-CC 検査行列に準じたウェイトパターンを複数記憶し、入力ビットが情報系列の場合に、情報系列の終端ほど、符号化率が低いウェイトパターンを用いるようにした。符号化率が低いほど、誤り訂正能力が高いので、このようにすることで、ターミネーション系列を削減した場合においても、受信情報系列の後部に発生するビット誤りを訂正しつつ、伝送効率の劣化を抑えることができる。

【0120】

(実施の形態 4)

本実施の形態では、LDPC-CC におけるターミネーション系列の数を少なくするため、LDPC-CC 符号化後にターミネーション系列の一部のビットをパンクチャしてターミネーション系列の量を削減するターミネーション系列パンクチャ部を備える送信装置について説明する。

【0121】

図12は、本実施の形態における送信装置の要部構成を示すブロック図である。図12の送信装置 1000 は、LDPC-CC 符号化部 1010、ターミネーション系列パンクチャ部 1020、インタリーブ部 1030、変調部 1040、制御情報生成部 1050、無線部 1060、及び送信アンテナ 1070 を備えて構成される。

【0122】

LDPC-CC 符号化部 1010 は、送信情報系列にターミネーション系列が付加された入力系列に対し、LDPC-CC 符号化処理を行い、符号化後の送信符号語系列をターミネーション系列パンクチャ部 1020 に出力する。

【0123】

ターミネーション系列パンクチャ部 1020 は、送信符号語系列のうち、ターミネーション系列に対しパンクチャを行い、パンクチャ後のターミネーション系列をインタリーブ部 1030 に出力する。パンクチャ処理については、後述する。

【0124】

インタリーブ部 1030 は、送信符号語系列に対して、系列の順序の並び替え処理(インタリーブ)を行い、インタリーブ後の送信符号語系列を変調部 1040 に出力する。

【0125】

変調部 1040 は、インタリーブ後の送信符号語系列を、PSK (Phase Shift Keying), QAM (Quadrature Amplitude Modulation) などの変調方式で変調し、変調後の送信変調シンボル系列を無線部 1060 に出力する。

【0126】

制御情報生成部 1050 は、送信側と受信側との間で信号を送受信するために必要な制御情報を生成し、変調部 1040 に送出する。制御情報としては、変調方式や送信情報系列長、時間・周波数同期のためのプリアンブル信号などがある。

【0127】

10

20

30

40

50

無線部 1060 は、送信変調シンボル系列に対して、D/A (Digital to Analog) 変換、周波数変換、RF (Radio Frequency) フィルタ処理などの無線変調処理を行い、送信 RF 信号を生成し、送信アンテナ 1070 を介して送信する。

【0128】

以下、上述のように構成された送信装置 1000 の動作について主にターミネーション系列パンクチャ部 1020 のパンクチャ処理を中心に説明する。なお、以下では、LDPC-CC 符号化部 1010 における LDPC-CC 符号化率を $R = b/c$ として説明する。

【0129】

LDPC-CC 符号化部 1010 において、送信情報系列 u_i ($i = 1, \dots, n$) にターミネーション系列が付加された入力系列に対し、LDPC-CC 符号化処理が施され、送信符号語系列 $[v_{k,i}, x_{k,j}]$ が取得される。ここで、 $k = 1, \dots, c$ 、 $j = 1, \dots, L$ であり、 L は、ターミネーション系列の系列長を表す。LDPC-CC の符号化方法については、非特許文献 1 及び非特許文献 2 に記述されているので、説明を省略する。

10

【0130】

ターミネーション系列パンクチャ部 1020 では、LDPC-CC 符号化部 1010 から出力される送信符号語系列 $[v_{k,i}, x_{k,j}]$ のうち、ターミネーション系列に相当する $x_{k,j}$ に対し、パンクチャ処理が施される。

【0131】

図 13 に、ターミネーション系列パンクチャ部 1020 が実施するパンクチャの一例を示す。図 13 は、符号化率 $R = 1/2$ ($b = 1, c = 2$) の場合の例を示し、上段は $x_{1,1} \sim x_{1,L}$ を示し、下段は $x_{2,1} \sim x_{2,L}$ を示し、左から右の順に各系列がターミネーション系列パンクチャ部 1020 に入力される。図 13 において、斜線が記されているビット $x_{k,j}$ が、ターミネーション系列パンクチャ部 1020 においてパンクチャされるビットを表している。図 13 に示すように、本実施の形態では、ターミネーション系列パンクチャ部 1020 は、ターミネーション系列の前方部のビットに対してはパンクチャの頻度を少なくし、ターミネーション系列の後方部のビットに対してはパンクチャの頻度を多くする。

20

【0132】

LDPC-CC 符号化では、ターミネーション系列の前方部ほど送信情報系列に近いので、ターミネーション系列の前方部に比べ後方部は、送信符号語系列 $v_{k,i}$ へ与える影響が小さい。したがって、ターミネーション系列パンクチャ部 1020 が、ターミネーション系列の後部ほどパンクチャするビットの割合を多くすることによって、パンクチャによる受信情報系列の誤り率特性の劣化を抑えつつ、ターミネーション系列送信量を削減することができる。

30

【0133】

パンクチャ後の送信符号語系列は、インタリーブ部 1030 によってインタリーブされ、変調部 1040 によって、インタリーブ後の送信符号語系列、制御情報に対し変調が施される。変調後の変調シンボル系列は、無線部 1060 によって、無線変調処理が施され、送信 RF 信号は、送信アンテナ 1070 を介して送信される。

40

【0134】

以上のように、本実施の形態では、送信装置 1000 は、LDPC-CC 符号化後の系列に含まれるターミネーション系列をパンクチャするターミネーション系列パンクチャ部 1020 を備え、ターミネーション系列パンクチャ部 1020 は、ターミネーション系列の後部ほどパンクチャするビットの割合を多くするようにした。

【0135】

ターミネーション系列の前方部のビットに対してはパンクチャの頻度が少ないパンクチャパターンを用い、ターミネーション系列の後方部のビットに対しては、パンクチャの頻度が多いパンクチャパターンを用いるようにすることで、復号時に送信符号語系列 $v_{k,i}$

50

i への影響が比較的小さいビット、つまり、ターミネーション系列の後方部のビットが優先的にパンクチャされるようになるので、パンクチャによる受信情報系列の誤り率特性の劣化を抑えつつ、ターミネーション系列送信量の削減を行うことができる。

【0136】

なお、ターミネーション系列パンクチャ部1020におけるパンクチャパターンは、図13に示すパターンに限られるものではなく、ターミネーション系列の前方部のビットに比べ、後方部のビットに対するパンクチャの頻度が多い他のパンクチャパターンを用いても、本発明の効果を楽しむことができる。

【0137】

(実施の形態5)

本実施の形態では、送信するターミネーション系列の数を少なくしたことによって発生する伝送誤りを、一部または全部のターミネーション系列を再送することによって補償する機能を備えた送信装置及び受信装置について説明する。

【0138】

図14は、本実施の形態における送信装置の要部構成を示すブロック図である。本実施の形態の説明にあたり、図12と同一構成部分には同一符号を付して説明を省略する。図14の送信装置1100は、図12の送信装置1000に対し、バッファ1110、1120、受信アンテナ1130、応答信号検出部1140、再送制御部1150、及び送信系列選択部1160を追加した構成を採る。

【0139】

バッファ1110は、ターミネーション系列パンクチャ部1020によってパンクチャされたパンクチャ後の送信符号語系列を蓄積する。一方、バッファ1120は、ターミネーション系列パンクチャ部1020でパンクチャしたパンクチャビット系列を蓄積する。

【0140】

応答信号検出部1140は、受信アンテナ1130を介して受信した受信信号から、後述する通信相手の受信装置1200から通知される応答信号を検出し、検出した応答信号を再送制御部1150に送出する。

【0141】

再送制御部1150は、応答信号に基づいて、再送制御情報を作成する。具体的には、再送制御部1150は、応答信号に基づいて、以下の4つのカテゴリの再送制御情報を作成する。

【0142】

- (0)再送制御情報「0」：再送無し
- (1)再送制御情報「1」：全送信符号語系列再送
- (2)再送制御情報「2」：パンクチャ後送信符号語系列再送
- (3)再送制御情報「3」：パンクチャビット系列再送

【0143】

なお、応答信号と再送制御情報との対応については、後述する。再送制御部1150は、再送制御情報を、送信系列選択部1160及び制御情報生成部1050に出力する。

【0144】

送信系列選択部1160は、再送制御部1150から出力される再送制御情報に応じて、インタリーブ部1030に出力する送信符号語系列を選択する。具体的には、送信系列選択部1160は、再送制御情報「0」の場合、ターミネーション系列パンクチャ部1020から出力される新たな送信符号語系列をインタリーブ部1030に送出する。

【0145】

また、送信系列選択部1160は、再送制御情報「1」の場合、バッファ1110及びバッファ1120に蓄積されている系列から、ターミネーション系列パンクチャ部1020においてパンクチャされる前の送信符号語系列を再生し、再生した送信符号語系列をインタリーブ部1030に送出する。

【0146】

10

20

30

40

50

また、送信系列選択部 1 1 6 0 は、再送制御情報「2」の場合、バッファ 1 1 1 0 に蓄積されているパンクチャ後の送信符号語系列をインタリーブ部 1 0 3 0 に送出する。

【0 1 4 7】

また、送信系列選択部 1 1 6 0 は、再送制御情報「3」の場合、バッファ 1 1 2 0 に蓄積されているパンクチャビット系列を送信符号語系列としてインタリーブ部 1 0 3 0 に送出する。

【0 1 4 8】

図 1 5 は、本実施の形態における受信装置の要部構成を示すブロック図である。図 1 5 の受信装置 1 2 0 0 は、受信アンテナ 1 2 0 1、無線部 1 2 0 2、直交復調部 1 2 0 3、チャンネル変動推定部 1 2 0 4、制御情報検波部 1 2 0 5、対数尤度演算部 1 2 0 6、デ
10
インタリーブ部 1 2 0 7、ターミネーション系列デパンクチャ部 1 2 0 8、Sum-product復号部 1 2 0 9、バッファ 1 2 1 0、誤り検出部 1 2 1 1、応答信号生成部 1 2 1 2、及び送信アンテナ 1 2 1 3 を備えて構成される。

【0 1 4 9】

受信アンテナ 1 2 0 1 は、送信装置 1 1 0 0 から送信される送信 R F 信号を受信し、無線部 1 2 0 2 に送出する。

【0 1 5 0】

無線部 1 2 0 2 は、R F フィルタ処理、周波数変換、A / D (Analog to Digital) 変換などの無線復調処理を行い、無線復調処理後のベースバンド信号を直交復調部 1 2 0 3
20
に送出する。

【0 1 5 1】

直交復調部 1 2 0 3 は、I チャンネル、Q チャンネルそれぞれのベースバンド信号を検出し、チャンネル変動推定部 1 2 0 4、制御情報検波部 1 2 0 5、及び対数尤度演算部 1 2 0 6 に送出する。

【0 1 5 2】

チャンネル変動推定部 1 2 0 4 は、ベースバンド信号に含まれる既知信号を用いて、送信装置 1 1 0 0 と受信装置 1 2 0 0 との間の無線伝搬路におけるチャンネル変動を推定する。

【0 1 5 3】

制御情報検波部 1 2 0 5 は、ベースバンド信号に含まれる制御情報を検出し、検出した制御情報を対数尤度演算部 1 2 0 6 に送出する。また、制御情報検波部 1 2 0 5 は、制御
30
情報に含まれる再送制御情報を検出し、検出した再送制御情報をターミネーション系列デパンクチャ部 1 2 0 8 及びバッファ 1 2 1 0 に送出する。

【0 1 5 4】

対数尤度演算部 1 2 0 6 は、ベースバンド信号を用いて、各符号語ビットの対数尤度比を求め、得られた対数尤度比をデインタリーブ部 1 2 0 7 に送出する。

【0 1 5 5】

デインタリーブ部 1 2 0 7 は、送信装置 1 0 0 0 のインタリーブ部 1 0 3 0 で行った並び替え処理の逆の処理を用いて、対数尤度比の系列の順序を元の並び順に並び替え、並び替え後の対数尤度比をターミネーション系列デパンクチャ部 1 2 0 8 に送出する。

【0 1 5 6】

ターミネーション系列デパンクチャ部 1 2 0 8 は、制御情報検波部 1 2 0 5 から出力される再送制御情報に応じて、デインタリーブ部 1 2 0 7 から出力される対数尤度比に対しデパンクチャを行う。デパンクチャ処理については、後述する。

【0 1 5 7】

Sum-product復号部 1 2 0 9 は、ターミネーション系列デパンクチャ部 1 2 0 8 から出力される対数尤度比系列を用いてSum-product復号を行い、Sum-product復号終了時の対数尤度比系列をバッファ 1 2 1 0 に送出する。また、Sum-product復号部 1 2 0 9 は、Sum-product復号終了時の対数尤度比系列を用いて硬判定により受信符号語系列を取得し、得られた受信符号語系列を誤り検出部 1 2 1 1 に送出する。

【0 1 5 8】

10

20

30

40

50

誤り検出部 1 2 1 1 は、Sum-product復号部 1 2 0 9 から出力される受信符号語系列に対して、LDPC-CC検査行列を用いてパリティ検査を行い、誤りを検出する。なお、誤り検出部 1 2 1 1 は、送信情報系列長 n をメモリ長 M 毎に分割したグループ毎に誤り検出する。グループ毎の誤り検出処理については、後述する。

【 0 1 5 9 】

誤り検出部 1 2 1 1 は、誤り検出の結果、誤りが検出されなかった場合、受信符号語系列の中の受信情報系列のみを受信系列として出力する。

【 0 1 6 0 】

応答信号生成部 1 2 1 2 は、誤り検出部 1 2 1 1 から出力される誤り検出情報に応じて応答信号を生成する。例えば、誤り検出情報が「誤りなし」を示す場合、応答信号生成部 1 2 1 2 は、正しく受信できたことを送信装置 1 1 0 0 に知らせるため、ACK信号を生成する。

10

【 0 1 6 1 】

一方、誤り検出情報が「誤り有り」を示す場合、応答信号生成部 1 2 1 2 は、正しく受信していないことを送信装置 1 1 0 0 に知らせるため、NACK信号を生成する。

【 0 1 6 2 】

なお、応答信号生成部 1 2 1 2 は、グループ毎の誤り検出の結果から、受信符号語系列全体にわたって誤りが発生している場合や、受信符号語系列の前方や中央付近のグループにのみ誤りが発生している場合には、送信符号語系列全体の再送を要求するNACK: type-I、若しくは、ターミネーション系列パンクチャ後の送信符号語系列の再送を要求するNACK: type-IIを生成する。また、応答信号生成部 1 2 1 2 は、受信符号語系列の後方のグループにのみ誤りが発生している場合には、ターミネーション系列パンクチャ部 1 0 2 0 によりパンクチャされたビット系列のみの再送を要求するNACK: type-IIIを生成する。

20

【 0 1 6 3 】

送信アンテナ 1 2 1 3 は、応答信号生成部 1 2 1 2 から出力されるACK、又はNACK信号を送信装置 1 1 0 0 に送信する。

【 0 1 6 4 】

以下、上述のように構成された送信装置 1 1 0 0 及び受信装置 1 2 0 0 の動作について、主に再送及び復号処理を中心に説明する。

30

【 0 1 6 5 】

送信装置 1 1 0 0 から送信された送信RF信号は、受信装置 1 2 0 0 の受信アンテナ 1 2 0 1 を介して受信され、無線部 1 2 0 2 によって、無線復調処理が施される。無線復調処理後の信号は、直交復調部 1 2 0 3 でベースバンド信号に復調される。

【 0 1 6 6 】

制御情報検波部 1 2 0 5 では、ベースバンド信号に含まれる制御情報が検出される。また、制御情報検波部 1 2 0 5 では、制御情報に含まれる再送制御情報が検出される。

【 0 1 6 7 】

対数尤度演算部 1 2 0 6 では、ベースバンド信号から、送信された各符号語ビットの対数尤度比が求められ、デインタリーブ部 1 2 0 7 において、送信装置 1 1 0 0 でのインタリーブ部 1 0 3 0 で行った並び替え処理の逆の処理を用いて、対数尤度比の系列の順序が並び替えられる。

40

【 0 1 6 8 】

ターミネーション系列デパンクチャ部 1 2 0 8 では、制御情報検波部 1 2 0 5 から出力される再送制御情報の種類に応じて対数尤度比がデパンクチャされる。

【 0 1 6 9 】

(0) 再送制御情報「 0 」の場合

ターミネーション系列デパンクチャ部 1 2 0 8 は、ターミネーション系列パンクチャ部 1 0 2 0 でパンクチャされたビットの位置に相当する位置にLLR = 0を挿入して対数尤度比系列を生成(デパンクチャ)する。ターミネーション系列デパンクチャ部 1 2 0 8 は

50

、デパンクチャ後の対数尤度比系列をSum-product復号部1209に送出する。再送制御情報「0」の場合、後述の再送制御情報「1」～「3」の場合と異なり、ターミネーション系列デパンクチャ部1208は、バッファ1210に蓄積されている対数尤度比系列を用いず、対数尤度比系列をデパンクチャする。

【0170】

(1)再送制御情報「1」の場合

バッファ1210は、蓄積している過去の対数尤度比系列をターミネーション系列デパンクチャ部1208に送出する。ターミネーション系列デパンクチャ部1208は、デインタリーブ部1207から出力される対数尤度比系列とバッファ1210から出力される対数尤度比系列とを合成した後、合成後の対数尤度比系列をSum-product復号部1209

10

【0171】

(2)再送制御情報「2」の場合

バッファ1210は、蓄積している過去の対数尤度比系列をターミネーション系列デパンクチャ部1208に送出する。ターミネーション系列デパンクチャ部1208は、デインタリーブ部1207から出力される対数尤度比系列を、再送制御情報「0」の場合と同様にデパンクチャし、デパンクチャ後の対数尤度比系列とバッファ1210から出力される対数尤度比系列とを合成した後、合成後の対数尤度比系列をSum-product復号部1209に送出する。

【0172】

20

(3)再送制御情報が「3」の場合

バッファ1210は、蓄積している過去の対数尤度比系列をターミネーション系列デパンクチャ部1208に送出する。ターミネーション系列デパンクチャ部1208は、バッファ1210から出力される対数尤度比系列のうち、ターミネーション系列デパンクチャ部1020でパンクチャされたビット位置に相当する位置のLLRを、デインタリーブ部1207から出力される対数尤度比系列に挿入して対数尤度比系列を生成(デパンクチャ)する。ターミネーション系列デパンクチャ部1208は、デパンクチャ後の対数尤度比系列をSum-product復号部1209に送出する。

【0173】

Sum-product復号部1209は、ターミネーション系列デパンクチャ部1208から出力される対数尤度比系列を用いてSum-product復号を行う。Sum-product復号部1209は、Sum-product復号終了時の対数尤度比系列をバッファ1210に送出する。また、Sum-product復号部1209は、Sum-product復号終了時の対数尤度比系列を硬判定した受信符号語系列を誤り検出部1211に送出する。

30

【0174】

誤り検出部1211は、Sum-product復号部1209から出力される、受信符号語系列に対して、LDPC-CC検査行列を用いてパリティ検査を行い、誤り検出する。なお、LDPC-CCパリティ検査は、受信符号語系列が式(6)を満たすかどうかで行う。

【数6】

$$v_t H_0^T(t) + v_{t-1} H_1^T(t) + \dots + v_{t-m_s} H_{m_s}^T(t) = 0 \quad \dots (6)$$

40

式(6)を用いてLDPC-CCパリティ検査を行うことで、誤り検出部1211は、送信情報系列長nをメモリ長M毎に分割したグループ毎に誤り検出することができる。したがって、グループ毎の誤り検出結果を比較することにより、受信符号語系列のうち、どの箇所に誤りが多いか検出することができる。

【0175】

このように、LDPC-CCパリティ検査では、誤りが検出された位置がわかるので、誤りが検出された位置から、その誤りが、ターミネーション系列をパンクチャしたことによるものかどうかを識別することができる。したがって、誤りが、ターミネーション系列をパンクチャしたことによるものである場合には、後述する応答信号生成部1212にお

50

いて、パンクチャされたビットのみの再送要求を示す応答信号 (N A C K : T y p e - I I I) を生成することにより、再送による伝送効率の低下を抑えることができる。

【 0 1 7 6 】

誤り検出部 1 2 1 1 は、グループ毎のパリティ検査結果を誤り検出情報として応答信号生成部 1 2 1 2 に出力する。なお、誤り検出情報の例として、ベクトル $E = [e_1, e_2, \dots, e_{J-1}, e_J]$ を使うことができる。ここで、 J は誤り検出を行ったグループ数であり、 i 番目のグループ ($i = 1, \dots, J$) でパリティ検査を満たし誤りが検出されない場合、 $e_i = 0$ とし、誤りが検出された場合、 $e_i = 1$ とする。

【 0 1 7 7 】

応答信号生成部 1 2 1 2 は、誤り検出部 1 2 1 1 から出力される誤り検出情報に応じて応答信号を生成する。具体的には、ベクトル E がオールゼロベクトルの場合、誤りが検出されなかったとして、応答信号生成部 1 2 1 2 は、A C K 信号を生成する。また、ベクトル E に 1 つ以上の 1 の値をとる要素が含まれる場合、誤りが検出されたとして、応答信号生成部 1 2 1 2 は、N A C K 信号を生成する。

【 0 1 7 8 】

なお、応答信号生成部 1 2 1 2 は、ベクトル E の要素が 1 となる位置より、受信符号語系列のどのあたりに誤りが発生しているかをグループ単位で推定することができる。受信符号語系列全体にわたって誤りが発生している場合や、受信符号語系列の前方や中央付近のグループにのみ誤りが発生している場合、応答信号生成部 1 2 1 2 は、全送信符号語系列の再送を要求する N A C K : t y p e - I、若しくは、ターミネーション系列パンクチャ後の送信符号語系列の再送を要求する N A C K : t y p e - I I を生成する。また、応答信号生成部 1 2 1 2 は、受信符号語系列の後方のグループにのみ誤りが発生している場合、ターミネーション系列パンクチャ部 1 0 2 0 によりパンクチャされたビットのみの再送を要求する N A C K : t y p e - I I I を生成する。

【 0 1 7 9 】

応答信号生成部 1 2 1 2 から出力される A C K、又は N A C K 信号は、送信アンテナ 1 2 1 3 を介して送信装置 1 1 0 0 に送信される。

【 0 1 8 0 】

送信装置 1 1 0 0 の L D P C - C C 符号化部 1 0 1 0 では、送信情報系列 u_i ($i = 1, \dots, n$) に対して、ターミネーションを含む L D P C - C C 符号化処理が施され、送信符号語系列 $[v_{k,i}, x_{k,j}]$ が取得される。ターミネーション系列パンクチャ部 1 0 2 0 では、L D P C - C C 符号化部 1 0 1 0 から出力された送信符号語系列 $[v_{k,i}, x_{k,j}]$ のうち、ターミネーション系列に相当する $x_{k,j}$ に対し、パンクチャ処理が施される。

【 0 1 8 1 】

パンクチャ後の送信符号語系列は、バッファ 1 1 1 0 及び送信系列選択部 1 1 6 0 に出力される。また、パンクチャされたパンクチャビット系列は、バッファ 1 1 2 0 に出力される。

【 0 1 8 2 】

応答信号検出部 1 1 4 0 では、受信アンテナ 1 1 3 0 を介して受信された受信信号から、受信装置 1 2 0 0 から送信される応答信号が検出され、再送制御部 1 1 5 0 では、応答信号に応じて、再送制御情報が次のように生成される。

【 0 1 8 3 】

- (1) 応答信号が A C K の場合、再送制御情報「 0 」が作成される。
- (2) 応答信号が N A C K : t y p e - I の場合、再送制御情報「 1 」が作成される。
- (3) 応答信号が N A C K : t y p e - I I の場合、再送制御情報「 2 」が作成される。

(4) 応答信号が N A C K : t y p e - I I I の場合、再送制御情報「 3 」が作成される。

【 0 1 8 4 】

10

20

30

40

50

送信系列選択部 1160 では、再送制御部 1150 から出力される再送制御情報に基づいて、送信符号語系列が選択される。

【0185】

具体的には、再送制御情報「0」の場合、ターミネーション系列パンクチャ部 1020 から出力される新たな送信符号語系列が選択される。また、再送制御情報「1」の場合、ターミネーション系列をパンクチャする前の送信符号語系列が選択される。また、再送制御情報「2」の場合、バッファ 1110 に蓄積されているパンクチャ後の送信符号語系列が選択される。また、再送制御情報「3」の場合、バッファ 1120 に蓄積されているパンクチャビット系列が送信符号語系列として選択される。

【0186】

このようにすることで、受信装置 1200 の誤り検出部 1211 で誤りが検出された場合において、誤りに影響を与える系列のみが再送されるようになるので、再送による伝送効率の低下を抑圧することができる。

【0187】

選択された送信符号語系列は、インタリーブ部 1030 によってインタリーブされ、変調部 1040 によって、インタリーブ後の送信符号語系列、制御情報に対し変調が施される。なお、制御情報は、受信装置 1200 が、どのような信号が送信されたかを判別できるようにするため、再送制御情報を含んでいる。

【0188】

以上のように、本実施の形態によれば、送信装置 1100 は、ターミネーション系列パンクチャ部 1020 がパンクチャしたターミネーション系列を蓄積するバッファ 1120 を備え、通信相手の受信装置 1200 から再送要求が通知された場合に、バッファ 1120 に蓄積されたターミネーション系列を送信するようにした。LDPC-CC パリティ検査では、誤りが検出された位置がわかるので、誤りが検出された位置から、その誤りが、ターミネーション系列をパンクチャしたことによるものかどうかを識別することができる。したがって、誤りがターミネーション系列をパンクチャしたことによるものである場合には、ターミネーション系列パンクチャ部 1020 がパンクチャしたビットのみを再送することにより、再送による伝送効率の低下を抑えることができる。

【0189】

なお、本実施の形態では、図 13 で示されるパンクチャパターンを用いた送信装置を例に挙げて説明したが、これに限るものではなく、その他の任意のパンクチャパターンを用いた場合においても、本実施の形態で説明した効果を得ることができる。

【0190】

また、ターミネーション系列パンクチャ部 1020 が、ターミネーション系列を全てパンクチャするパンクチャパターンを用い、送信装置 1100 は、初回送信時には、ターミネーション系列を全く送信しないようにしても良い。このようにすることで、ターミネーション系列を送信することによる伝送効率の低下を回避することができる。その場合に、受信装置 1200 から NACK: type - III が送信されてきたら、初回送信時に送信しなかったターミネーション系列を送信するようにすることで、本実施の形態で説明した効果を得ることができる。

【0191】

また、本実施の形態では、再送制御情報「0」～「3」、また、応答信号を ACK、NACK: type - I ~ type - III とした例を用いて説明したが、この限りではなく、ビット誤りが、ターミネーション系列をパンクチャしたことによって発生したものかどうか判別でき、そのことを送信装置 1100 に通知できる応答信号を送出する他の方法を用いても、本発明の効果を得ることができる。

【0192】

また、本実施の形態では、送信装置 1100 が、再送制御情報「1」又は「2」の場合、バッファ 1110 及びバッファ 1120 に蓄積されている送信符号語系列を再送する場合について説明したが、再送系列を再度 LDPC-CC 符号化部 1010 により符号化し

10

20

30

40

50

て送信するようにしても良い。図16にこの場合の送信装置の構成図を示す。なお、図16の送信装置の説明にあたり、図14と同一構成部分には同一符号を付して説明を省略する。

【0193】

図16の送信装置1300は、図14の送信装置1100に対し、バッファ1310及びLDPC-CC符号化部1010に代え、バッファ1310及びLDPC-CC符号化部1320を追加した構成を採る。

【0194】

バッファ1310は、送信情報系列を蓄積する。また、バッファ1310は、再送制御部1150から再送制御情報「0」が出力される場合、蓄積されている送信情報系列をクリアし、新規の送信情報系列を蓄積するとともに、新規の送信情報系列をLDPC-CC符号化部1320に送出する。また、バッファ1310は、再送制御情報「1」又は「2」の場合、蓄積されている送信情報系列をLDPC-CC符号化部1320に送出する。なお、再送制御情報が「3」の場合、バッファ1310は、蓄積している送信情報系列をそのまま蓄積する。

【0195】

LDPC-CC符号化部1320は、送信情報系列に対し任意の符号化率でLDPC-CC符号化を行う。ここで、LDPC-CC符号化部1320は、初回の送信時と再送時とで異なる符号化率を用いてLDPC-CC符号化を行うようにしてもよい。これにより、受信装置1200から再送制御情報「1」又は「2」の再送要求があった場合、再送時に、同一の送信情報系列に対し、誤り訂正能力が高い異なる符号化率でLDPC-CC符号化を施すことができる。

【0196】

なお、バッファ1310は、再送制御情報「0」の場合、蓄積する送信情報系列を新規な送信情報系列に更新する際、新規の送信情報系列の順序を並び替えて蓄積するようにしても良い。LDPC-CCでは、ターミネーション系列をパンクチャした場合に発生する誤りが、送信情報系列の後方のビットに集中する。したがって、バッファ1310が、送信情報系列の順序を並び替えて蓄積することにより、再送時に、初回送信時と異なる送信情報ビットが送信情報系列の後方に位置するようになるので、再送時においてビット誤りが発生する確率を減少させることができる。送信情報系列の順序を並び替える方法としては、例えば、送信情報系列の後から順に並び替えるなどの方法がある。このようにすることで、初回送信時に送信情報系列の後方でLDPC-CC符号化され送信されたビットが、再送時には前方でLDPC-CC符号化され送信されるようになるので、初回送信時に誤りの影響を受けやすい後方のビットが正しく受信される割合が高くなる。

【0197】

(実施の形態6)

本実施の形態では、送信するターミネーション系列の数を少なくしたことによって受信情報系列の後部に発生する伝送誤りを、送信情報系列の後部のビットをあらかじめ別に符号化しておき、その符号化利得によって軽減する機能を備えた送信装置及び受信装置の構成について説明する。

【0198】

図17は、本実施の形態における送信装置の要部構成を示すブロック図である。本実施の形態の説明にあたり、図12と同一構成部分には同一符号を付して説明を省略する。図17の送信装置1400は、図12の送信装置1000に対し、情報系列分割部1410、外符号化部1420、及び並び替え部1430をさらに追加した構成を採る。

【0199】

情報系列分割部1410は、送信情報系列を前方と後方とに分割することにより2つの系列を取得する。例えば、情報系列分割部1410は、 n ビットの送信情報系列を、送信情報系列の前方から K ビットと残りの $n - K$ ビットとに分割する。情報系列分割部1410は、分割後の K ビットの送信情報系列を並び替え部1430に出力するとともに、分割

10

20

30

40

50

後の $n - K$ ビットの送信情報系列を外符号化部 1 4 2 0 に出力する。

【 0 2 0 0 】

外符号化部 1 4 2 0 は、情報系列分割部 1 4 1 0 で分割された $n - K$ ビットの送信情報系列に対し外符号化を施す。このようにすることで、送信情報系列の後方の $n - K$ ビットに対して外符号化が施されるので、ターミネーション系列の数を少なくすることにより、受信情報系列の後部に発生する伝送誤りを、外符号化の符号化利得により軽減することができるようになる。

【 0 2 0 1 】

外符号化の符号化方式としては、ターミネーションの必要がないブロック符号が好ましく、例えば、情報系列長 $n - K$ ビットの LDPC - BC を適用する。外符号化部 1 4 2 0 は、外符号化後の送信情報系列を並び替え部 1 4 3 0 に出力する。

10

【 0 2 0 2 】

並び替え部 1 4 3 0 は、情報系列分割部 1 4 1 0 からの送信情報系列と、外符号化部 1 4 2 0 からの外符号化語の送信情報系列とを入力し、これら送信情報系列の順序を並び替える。並び替えの順序としては、外符号化部 1 4 2 0 により生成されたパリティ系列を、外符号化部 1 4 2 0 に入力した送信情報系列より前方に優先的に配置する。

【 0 2 0 3 】

上述したように、LDPC - CC では、ターミネーション系列をパンクチャした場合に発生する誤りが、受信符号語系列の後方のビットに集中する。したがって、並び替え部 1 4 3 0 が、外符号化部 1 4 2 0 により生成したパリティ系列を、前方に優先的に配置するようにすることで、当該パリティ系列が誤る割合が軽減されるようになり、外符号化利得を向上させることができるようになる。

20

【 0 2 0 4 】

並び替え部 1 4 3 0 は、インタリーブした送信情報系列を LDPC - CC 符号化部 1 0 1 0 に送出する。

【 0 2 0 5 】

図 1 8 は、本実施の形態における受信装置の要部構成を示すブロック図である。本実施の形態の説明にあたり、図 1 5 と同一構成部分には同一符号を付して説明を省略する。図 1 8 の受信装置 1 5 0 0 は、図 1 5 の受信装置 1 2 0 0 に対し、応答信号生成部 1 2 1 2 及び送信アンテナ 1 2 1 3 を削除し、並び替え部 1 5 0 1、受信情報系列分割部 1 5 0 2、及び外符号復号部 1 5 0 3 をさらに追加した構成を採る。

30

【 0 2 0 6 】

並び替え部 1 5 0 1 は、Sum-product 復号部 1 2 0 9 によって復号された受信情報系列を、送信装置 1 4 0 0 における並び替え部 1 4 3 0 と逆の規則で並び替え、並び替え後の受信情報系列を受信情報系列分割部 1 5 0 2 に送出する。

【 0 2 0 7 】

受信情報系列分割部 1 5 0 2 は、送信装置 1 4 0 0 における情報系列分割部 1 4 1 0 と同じ規則で、並び替え部 1 5 0 1 から出力される受信情報系列を 2 つの受信情報系列に分割する。受信情報系列分割部 1 5 0 2 は、分割後の K ビットの情報系列を誤り検出部 1 2 1 1 に送出し、分割後の残りの受信情報系列を外符号復号部 1 5 0 3 に送出する。

40

【 0 2 0 8 】

外符号復号部 1 5 0 3 は、分割後の受信情報系列に対し外符号の復号処理を行う。例えば、外符号化部 1 4 2 0 において外符号として情報系列長 $n - K$ ビットの LDPC - BC が適用された場合、Sum-product 復号やビリーフ伝搬 (Belief Propagation: BP) 復号を用いて LDPC - BC 復号を行う。外符号復号部 1 5 0 3 は、復号後の受信情報系列を誤り検出部 1 2 1 1 に送出する。

【 0 2 0 9 】

以上のように、本実施の形態では、情報系列分割部 1 4 1 0 は、送信情報系列を前方と後方とに分割することにより 2 つの系列を取得し、外符号化部 1 4 2 0 は、送信情報系列の後方のビットに対し、外符号化を施すようにした。これにより、LDPC - CC 符号化

50

部 1 0 1 0 がターミネーション系列を削減したことにより受信情報系列の後部に誤りが発生した場合に、外符号によって当該誤りを訂正することができるので、受信品質の劣化を抑圧することができる。さらに、外符号化は、送信情報系列全体に行うのではなく、ターミネーション系列削減により誤りが発生しやすい送信情報系列の後部にのみ施すので、連接符号化による符号化率の低下を抑えることができる。さらに、外符号化を施した後に、並び替え部 1 4 3 0 を設け、外符号化によって得られた外符号語系列の一部を送信情報系列の前方に優先的に配置することで、外符号化の符号化利得を向上することができる。

【 0 2 1 0 】

なお、以上の説明では、情報系列分割部 1 4 1 0 が、送信情報系列を、送信情報系列の先頭から連続して K ビット抽出する場合について説明したが、送信情報系列の任意の位置から連続して K ビット抽出する、又は、ランダムに K ビットを抽出し、残りの $n - K$ ビットを外符号化部 1 4 2 0 に出力するようにしても良い。このようにしても、送信情報系列のうち連続した $n - K$ ビット、又は、ランダムな $n - K$ ビットに対し外符号化が施されるので、パースト的に発生する誤りを訂正することができる。

10

【 0 2 1 1 】

また、以上の説明では、並び替え部 1 4 3 0 は、外符号化部 1 4 2 0 により生成されパリティ系列を、外符号化部 1 4 2 0 に入力した送信情報系列より前に優先的に配置する場合について説明したが、これに限るものではない。例えば、並び替え部 1 4 3 0 は、情報系列分割部 1 4 1 0 から出力される外符号化されていない送信情報系列と、外符号化部 1 2 2 0 により外符号化された送信情報系列とが混在するように並び替えるようにしても良い。このようにすることで、外符号化が施されたビットが、送信符号語系列の後部以外にも配置されるようになるので、外符号化利得をさらに向上させることができる。

20

【 0 2 1 2 】

なお、本実施の形態における受信装置の別の要部構成のブロック図を図 1 9 に示す。図 1 9 の受信装置 1 6 0 0 は、図 1 8 の受信装置 1 5 0 0 に対し、受信情報系列結合部 1 6 0 1 をさらに備える構成を採る。

【 0 2 1 3 】

受信情報系列結合部 1 6 0 1 は、受信情報系列分割部 1 5 0 2 から出力される受信情報系列と、外符号復号部 1 5 0 3 から出力される外符号化復号後の受信情報系列とを結合して、受信情報系列を取得し、得られた受信情報系列を誤り検出部 1 2 1 1 及び Sum-product 復号部 1 2 0 9 に出力する。

30

【 0 2 1 4 】

このようにすることで、Sum-product 復号部 1 2 0 9 は、外符号の復号結果を反映させた受信情報系列を用いながら、再度内符号である LDPC - CC の復号を行うことができるので、内符号の復号と外符号の復号とを反復して実行することにより、復号利得が向上し、受信情報系列に含まれる誤りの数を減らすことができる。

【 0 2 1 5 】

さらに、誤り検出部 1 2 1 1 の誤り検出結果を用いて、受信情報系列のうち、誤りが検出された部分にのみ誤り訂正復号を行う場合には、反復復号による演算回数を削減することができる。また、誤り検出部 1 2 1 1 の誤り検出結果から、誤りなく受信できた情報系列が確定されるので、そのビットの受信尤度を + または - にすることで内符号及び外符号の復号利得を向上することができる。

40

【 0 2 1 6 】

(実施の形態 7)

実施の形態 4 から実施の形態 6 では、ターミネーション系列に対しパンクチャを施す場合について説明した。本実施の形態では、LDPC - CC 符号化により得られた送信符号語系列に適するパンクチャを施す送信装置及びパンクチャ方法について説明する。

【 0 2 1 7 】

図 2 0 は、本実施の形態において用いられる LDPC - CC 検査行列の構成を示す図である。図 2 0 は、図 4 3 と異なり、 H^T でなく、検査行列 H の構成を示している。送信符

50

号語ベクトルを v で表すと、 $Hv = 0$ の関係式が成立する。

【0218】

本実施の形態におけるパンクチャ方法の説明にあたり、先ず、一般的なパンクチャ方法を、上記送信符号語系列 v に適用した場合の課題について説明する。一般的なパンクチャ方法については、例えば、非特許文献4に記載されている。なお、以下では、LDPC-CCが、符号化率 $R = 1/2$ 、 $(177, 131)$ の畳み込み符号を用いて構成される場合を例に説明する。

【0219】

図21は、一般的なパンクチャ方法を説明するための図である。同図において、 $v_{1,t}$ 、 $v_{2,t}$ ($t = 1, 2, \dots$) は、送信符号語系列 v を示す。一般的なパンクチャ方法では、送信符号語系列 v は、複数のブロックに分けられ、各ブロックに対し同一のパンクチャパターンが用いられ、送信符号語ビットが間引かれる。

10

【0220】

図21は、送信符号語系列 v が、6ビットごとにブロックに分けられ、すべてのブロックに対し、同一のパンクチャパターンが用いられ、一定の割合で送信符号語ビットが間引かれる様子を示している。同図において、丸印で囲まれたビットがパンクチャされるビット（送信しないビット）を示し、すべてのブロック1～ブロック5に対し、パンクチャ後の符号化率が $3/4$ となるように、 $v_{2,1}$ 、 $v_{2,3}$ 、 $v_{2,4}$ 、 $v_{2,6}$ 、 $v_{2,7}$ 、 $v_{2,9}$ 、 $v_{2,10}$ 、 $v_{2,12}$ 、 $v_{2,13}$ 、 $v_{2,15}$ を選択し、パンクチャする（送信しないビットとする）。

【0221】

20

次に、LDPC-CCを用いた符号化により得られた送信符号語系列に、図21に示すような一般的なパンクチャを施した場合の受信側（復号側）の影響を考える。なお、以下では、受信側（復号側）においてBP復号を用いる場合について検討する。BP復号では、LDPC-CCの検査行列に基づいて復号処理を行う。図22に、送信符号語系列 v とLDPC-CC検査行列 H との対応を示す。図22において、丸印で囲まれたビットは、パンクチャにより間引かれる送信符号語ビットである。この結果、検査行列 H において、四角枠で囲まれた1に対応するビットが、送信符号語系列に含まれなくなる。この結果、BP復号を行う際、四角枠で囲まれた1に対応するビットに対しては、初期の対数尤度比が存在しないので、対数尤度比が0に設定されることになる。

【0222】

30

BP復号では行演算と列演算とを反復して行う。したがって、初期の対数尤度比が存在しない（対数尤度比が0の）ビット（図22において四角枠で囲まれた1に対応するビット）が、同一行に2つ以上含まれると、当該行では、列演算により初期の対数尤度比が存在しない（対数尤度比が0の）ビットの対数尤度比が更新されるまで、当該行の行演算単独では、対数尤度比が更新されないことになる。すなわち、行演算単独では信頼度が伝搬されず、信頼度を伝搬させるためには、行演算と列演算とを反復する必要がある。したがって、このような行が多数存在すると、BP復号において反復処理数に制限があるような場合には、信頼度が伝搬されず、受信品質の劣化を招く原因となる。図22に示す例では、行1710は、行演算単独では信頼度が伝搬されない行、つまり、受信品質の劣化を招く原因となる行となる。

40

【0223】

これに対し、本実施の形態におけるパンクチャ方法を用いる場合、行演算単独で信頼度が伝搬されない行数を削減することができる。本実施の形態では、受信側（復号側）における、送信符号語ビットの処理単位ごとに、第1のパンクチャパターンと、第1のパンクチャパターンに比べより多くのビットを間引く第2のパンクチャパターンとを用いて、送信符号語ビットをパンクチャする。以下、図23及び図24を用いて説明する。

【0224】

図23は、本実施の形態におけるパンクチャ方法を説明するための図である。図21と同様に、 $v_{1,t}$ 、 $v_{2,t}$ ($t = 1, 2, \dots$) は、送信符号語系列 v を示す。なお、以下では、図21と同様に、1ブロックが6ビットから構成される場合について説明する。また

50

、受信側（復号側）における送信符号語ビットの処理単位が、ブロック1～ブロック5であるとする。図23に示す例では、先頭のブロック1に対しては、パンクチャを行わない第1のパンクチャパターンが用いられ、ブロック2～ブロック5に対しては、パンクチャを行う第2のパンクチャパターンが用いられ、この結果、 $v_{2,1}$ 、 $v_{2,3}$ 、 $v_{2,4}$ 、 $v_{2,6}$ 、 $v_{2,7}$ 、 $v_{2,9}$ 、 $v_{2,10}$ 、 $v_{2,12}$ 、 $v_{2,13}$ 、 $v_{2,15}$ がパンクチャされる様子が示されている。このように、本実施の形態では、符号化率が異なるパンクチャパターンを用いて、送信符号語ビットの処理単位内で、間引かれるビット数が少ない範囲を設けるようにする。

【0225】

図24に、この場合の送信符号語系列 v とLDPC-CC検査行列 H との対応を示す。図24では、同一行に四角枠で囲まれた1を2つ以上含む行が3行発生しているものの、図22の場合に比べ、その行数が削減されたことがわかる。これは、ブロック1に対してパンクチャを施さないようにしたことによる。

10

【0226】

このように、パンクチャを行わないブロックを設けることにより、BP復号時の受信品質の劣化を招く原因となる行数を削減することができる。この結果、行1720までの行では、初期に対数尤度が存在し、BP復号において、信頼度が確実に更新され、更新後の信頼度が、行1720に伝搬していくので、受信品質の劣化を抑えることができるようになる。このように、畳み込み符号（LDPC-CC）の検査行列の構造の特徴から、反復復号を複数回行うことにより、行演算単独で得られる行の信頼度が、順次、伝播し、パンクチャによる受信品質の劣化を抑えることができる。また、行演算単独では信頼度が伝搬

20

【0227】

ところで、図23に示す例では、パンクチャされないブロックが設けられることにより、送信される送信符号語ビットは増加し、伝送速度が低下する。しかし、第1のパンクチャパターンが用いられるビット数 N と、第2のパンクチャパターンが用いられるビット数 M との間に、 $N \ll M$ の関係が成り立つようにすれば、伝送速度の低下を抑えつつ、受信品質を向上することができる。図23は、 $N = 6$ 、 $M = 24$ の例であり、追加送信符号語ビット数は2ビットと少ないにもかかわらず、行演算単独では対数尤度を伝搬されない行数を6行から3行に減らすことができる。

30

【0228】

以下、本実施の形態における送信装置の構成について説明する。図25は、本実施の形態における送信装置の要部構成を示すブロック図である。本実施の形態の説明にあたり、図12と同一構成部分には同一符号を付して説明を省略する。図20の送信装置1800は、図12の送信装置1000に対し、ターミネーション系列パンクチャ部1020に代え、パンクチャ部1810を備えて構成される。なお、パンクチャ部1810は、第1パンクチャ部1811、第2パンクチャ部1812、及び、切り替え部1813を備えて構成される。

【0229】

パンクチャ部1810は、送信情報系列及びターミネーション系列からなる送信符号語系列に対しパンクチャを行い、パンクチャ後の送信符号語系列をインタリーブ部1030に出力する。

40

【0230】

具体的には、パンクチャ部1810は、第1のパンクチャパターンと、第1のパンクチャパターンより多くのビットを間引く第2のパンクチャパターンとを用いて、送信符号語系列をパンクチャする。第1のパンクチャパターンと第2のパンクチャパターンとは、パンクチャするビットの割合が異なる。パンクチャ部1810は、例えば、図26に示すようなパンクチャパターンを用いて、送信符号語系列をパンクチャする。図26において、 $(N + M)$ ビットは、受信側（復号側）における処理単位である。

【0231】

50

第1パンクチャ部1811は、第1のパンクチャパターンを用いて、送信符号語系列に対しパンクチャを行う。第2パンクチャ部1812は、第2のパンクチャパターンを用いて、送信符号語系列に対しパンクチャを行う。

【0232】

図26のパンクチャパターンを用いる場合、第1パンクチャ部1811は、受信側（復号側）の処理単位の先頭からNビットの送信符号語系列に対してはパンクチャを行わず、第1パンクチャ部1811に入力される送信符号語系列を切り替え部1813に出力する。第2パンクチャ部1812は、 $(N+1) \sim (N+M)$ ビットの送信符号語系列に対してパンクチャを行い、パンクチャ後の送信符号語系列を切り替え部1813に出力する。

【0233】

なお、第1パンクチャ部1811及び第2パンクチャ部1812は、制御情報生成部1050からの制御情報に基づいて、送信符号語系列にパンクチャを施すか否か決定するようにしてもよい。切り替え部1813は、制御情報生成部1050からの制御情報に応じて、第1パンクチャ部1811から出力される送信符号語系列、又は、第2パンクチャ部1812から出力される送信符号語系列の一方をインタリーブ部1030に出力する。

【0234】

以下、上述のように構成された送信装置1800の動作について主にパンクチャ部1810のパンクチャ処理を中心に説明する。なお、以下では、LDPC-CC符号化部1010が、符号化率 $R = 1/2$ 、 $(177, 131)$ の畳み込み符号を用いて、LDPC-CC符号化を施す場合を例に説明する。

【0235】

LDPC-CC符号化部1010において、送信情報系列 u_t ($t = 1, \dots, n$)に対し、LDPC-CC符号化処理が施され、 $v = (v_{1,t}, v_{2,t})$ が取得される。組織化符号の場合、 $v_{1,t}$ は送信情報系列 u_t であり、 $v_{2,t}$ はパリティを示す。パリティ $v_{2,t}$ は、送信情報系列 $v_{1,t}$ 及び図24の各行の検査式に基づいて求められる。

【0236】

パンクチャ部1810によって、符号化率 $R = 1/2$ の送信符号語系列 v に対し、パンクチャ処理が施される。例えば、パンクチャ部1810によって、図23に示すパンクチャが用いられる場合、ブロック1に対しては、パンクチャが施されず、ブロック2～ブロック5に対しては、所定の間隔で規則的にビットが間引かれる。つまり、ブロック2に対しては、 $v_{2,4}, v_{2,6}$ のビットが間引かれ、ブロック3に対しては、 $v_{2,7}, v_{2,9}$ のビットが間引かれ、ブロック4に対しては、 $v_{2,10}, v_{2,12}$ が間引かれ、ブロック5に対しては、 $v_{2,13}, v_{2,15}$ が間引かれる。このようにして、ブロック2～ブロック5に対して、符号化率 $R = 3/4$ の送信符号語系列が取得される。

【0237】

パンクチャ後の送信符号語系列は、インタリーブ部1030、変調部1040、無線部1060、及び、送信アンテナ1070を介して、受信側（復号側）に送信される。このとき、図23に示すパンクチャパターンが用いられる場合には、 $v_{2,4}, v_{2,6}, v_{2,7}, v_{2,9}, v_{2,10}, v_{2,12}, v_{2,13}, v_{2,15}$ は、送信されないことになる。

【0238】

このように、図23に示すパンクチャパターンが用いられる場合には、所定の周期ごとに、パンクチャを行わないブロックが発生する。図23に示すように、ブロック1に対してはパンクチャを行わないようにすることにより、図21の一般的なパンクチャ方法を用いた場合には送信されなかった $v_{2,1}, v_{2,3}$ が、送信されることになる。このようにすることで、BP復号を用いたとき行演算単独では信頼度が伝搬されない行は、図24の行1720に示される3行となる。図21と図23との比較から分かるように、送信ビットを2ビット追加することにより、行演算単独では信頼度が伝搬されない行数が6行から3行に削減される。この結果、対数尤度が初期に存在する行数が増加し、BP復号により、初期の信頼度が確実に更新されるようになり、さらに、この信頼度が、図24の行1720に伝搬するようになる。

10

20

30

40

50

【0239】

以後、畳み込み符号(LDPC-CC)の検査行列の構造の特徴から、複数回反復復号を行うことで、検査行列の先頭に多く存在する信頼度が、順次、伝播するようになり、パンクチャによる受信品質の劣化を抑えることができる。

【0240】

図23の例では、送信されることになった増加ビット数は2ビットと少ないので、伝送速度の低下は小さく、かつ、受信品質の劣化を抑えることができる。なお、このような効果が得られるのは、LDPC-CCが図31のように、検査行列において、1の存在する場所が平行四辺形の範囲に集中する型を採るという特徴によるものである。したがって、LDPC-BCの場合に適用しても、同様の効果を得ることができる可能性は低い。

10

【0241】

このように、パンクチャしないブロックを設けることにより、BP復号時に悪影響を与える行数を削減することができる。このとき、伝送効率を考慮すると、パンクチャしないブロックを構成するビットMと、パンクチャの対象となるブロックを構成するビットNとの間に、 $N \ll M$ の関係が成り立つことが重要である。 $N \ll M$ とすることにより、伝送効率の劣化を抑えつつ、受信品質の劣化を抑圧することができる。

【0242】

なお、第2のパンクチャパターンが適用されるブロック2~ブロック5に対し、パンクチャ部1810は、ランダムにパンクチャするのではなく、所定の規則に従って、パンクチャするようにするとよい。ランダムにパンクチャする場合に比べ、所定の規則に従ってパンクチャする場合には、パンクチャ演算処理が簡易になる。

20

【0243】

(他のパンクチャパターン)

パンクチャ部1810が用いるパンクチャパターンは図26に限られない。例えば、パンクチャ部1810が、図27に示すように、パンクチャ部1810は、第1のパンクチャパターンとして符号化率 $R_1 = 2/3$ のパンクチャパターンを用い、第2のパンクチャパターンとして符号化率 $R_2 = 5/6$ のパンクチャパターンを用いるようにしてもよい。

【0244】

また、図28A、図28Bに示すように、n個のフレームを受信側(復号側)における処理単位として、パンクチャを施すようにしてもよい。図28Aに示すように、nフレーム(nは、1以上の整数)の先頭からNビットに対しては、パンクチャを行わない第1のパンクチャパターンを用い、 $(N+1) \sim (N+M)$ ビットに対しては、パンクチャを行う第2のパンクチャパターンを用いるようにしてもよい。

30

【0245】

また、図28Bに示すように、nフレームの先頭からNビットに対しては、符号化率 $R_1 = 2/3$ の第1のパンクチャパターンを用い、 $(N+1) \sim (N+M)$ ビットに対しては、符号化率 $R_2 = 5/6$ の第2のパンクチャパターンを用いるようにしてもよい。

【0246】

また、図29A、図29Bに示すように、受信側(復号側)における処理単位の後部ほどパンクチャにより間引かれるビットが少なくなるようなパターンを用いるようにしてもよい。受信側(復号側)における処理単位の後部ほどパンクチャにより間引かれるビット数が少なくなるようにすることにより、BP復号において、受信品質の向上が図れる。

40

【0247】

なお、図26の場合と同様に、第1のパンクチャパターンが用いられるビット数Nと、第2のパンクチャパターンが用いられるビット数Mとの間に、 $N \ll M$ の関係が成り立つようにすれば、伝送速度の低下を抑えつつ、受信品質を向上することができる。

【0248】

また、図30Aに示すように、受信側(復号側)における処理単位であるn個のフレーム(nは、1以上の整数)の先頭からN1ビットに対しては、パンクチャを行わない第1のパンクチャパターンを用い、 $(N1+1) \sim (N1+M)$ ビットに対しては、パンクチャ

50

ヤを行う第2のパンクチャパターンを用い、 $(N1 + M + 1) \sim (N1 + M + N2)$ ビットに対しては、パンクチャを行わない第1のパンクチャパターンを用いるようにしてもよい。

【0249】

また、図30Bに示すように、受信側（復号側）における処理単位である n 個のフレーム（ n は、1以上の整数）の先頭から $N1$ ビットに対しては符号化率 $R1 = 2/3$ の第1のパンクチャパターンを用い、 $(N1 + 1) \sim (N1 + M)$ ビットに対しては、符号化率 $R2 = 5/6$ の第2のパンクチャパターンを用い、 $(N1 + M + 1) \sim (N1 + M + N2)$ ビットに対しては、符号化率 $R1 = 2/3$ の第1のパンクチャパターンを用いるようにしてもよい。

10

【0250】

パンクチャにより間引かれるビット数が少ない第1のパンクチャパターンが、受信側（復号側）における処理単位に1箇所用いられる場合に比べ（図28及び図29参照）、2箇所用いられる場合には（図30参照）、信頼度が高い検査行が増えるため、BP復号の際の収束速度が速く、少ない反復回数で復号結果を得ることができる。

【0251】

なお、上記処理単位にパンクチャにより間引かれるビット数が少ない第1のパンクチャパターンが用いられる箇所は、2箇所に限られず、3箇所以上でもよい。

【0252】

また、上記処理単位にパンクチャにより間引かれるビット数が少ない第1のパンクチャパターンが用いられる箇所が2箇所以上のときも、第1のパンクチャパターンが用いられるビット数の総数 N と、第2のパンクチャパターンが用いられるビット数の総数 M との間に、 $N < M$ の関係が成り立つようにすれば、伝送速度の低下を抑えつつ、受信品質を向上することができる。

20

【0253】

なお、図28、図29、及び図30には、 n フレームに対し、第1のパンクチャパターン及び第2のパンクチャパターンを用いる場合について説明したが、 n は、1以上の整数であればよく、1フレームの場合にも適用できる。

【0254】

以下では、復号処理タイミングとの関係を考慮して、LDPC-CC符号化により得られた送信符号語系列に対し適したパンクチャパターンについて検討する。

30

【0255】

図31は、復号処理タイミングを説明するための図である。図31において、受信データ系列は、それぞれ、 n フレーム（例えば、 n 個のOFDM（Orthogonal Frequency Division Multiplexing）シンボル：OFDMシンボルとは、OFDM方式が、32サブキャリアで構成されており、各サブキャリアごとに変調信号を構成する場合、全キャリア（32サブキャリア）で構成されるシンボルをいう。）から構成される。この受信データ系列長は、受信側（復号側）における処理単位であり、当該 n フレーム（又は、 n 個のOFDMシンボル）が、1つのまとまりとして、上位層のレイヤに受け渡される。一般的に、上位層のレイヤが次の n フレームのデータを取り込むまでにタイムラグが生じるため、図31の $t3$ 、 $t6$ 、 $t9$ のタイミング、つまり、 n フレームの最後部を受信したタイミングを、BP復号を行う期間の最後とするのが現実的である。

40

【0256】

LDPC-CCは、畳み込み符号の性質を有するため、 $t2$ のタイミングから、BP復号により推定されたデータが有効なデータ（正しい可能性が高いデータ）とするためには、 $t2$ のタイミングより前にBP復号を開始する必要がある。例えば、図31に示す例では、 $t2 \sim t5$ の間にBP復号により得られた推定データを有効なデータとするために、 $t1 \sim t6$ の間、BP復号を行う必要がある。同様に、 $t5 \sim t8$ の間に得られた推定データを有効なデータとするために、 $t4 \sim t9$ の間、BP復号を行う必要がある。

【0257】

50

このような復号処理タイミングを考慮した場合、例えば、 n フレームから成る受信データ系列の後部に、パンクチャにより間引かれるビット数が少なくなるようなパンクチャを施すようにする、と、 $B P$ 復号処理期間において、前方と後方の双方に、信頼度が伝搬される行が含まれるようになるので、効率的に、信頼度を伝搬させることができるようになる。

【0258】

以上のように、本実施の形態によれば、パンクチャ部1810は、送信符号語ビットの処理単位ごとに、第1のパンクチャパターンと、第1のパンクチャパターンに比べより多くのビットを間引く第2のパンクチャパターンとを用いて、送信符号語ビットをパンクチャするようにした。

10

【0259】

送信符号語系列に対し、一定の割合でパンクチャを施すのではなく、パンクチャ後の符号化率が異なる第1及び第2のパンクチャパターンを用いることにより、 $B P$ 復号による復号特性の劣化を抑えることができるようになる。

【0260】

パンクチャを行う限り、受信品質の劣化の要因となる行が発生してしまうものの、本実施の形態におけるパンクチャ方法のように、伝送速度の低下を抑圧しつつ、受信品質の劣化を抑える方法は、パフォーマンスのよいシステムを構築する上で非常に重要となる。

【0261】

なお、第1及び第2のパンクチャパターンは、各々、同一の複数のサブパターンから構成されるようにしてもよい。すなわち、図23に示すように、ブロック2～ブロック5それぞれに対し、同一のサブパンクチャパターンを用いるようにして、規則的に送信符号語ビットを間引くようにしてもよい。これにより、パンクチャ演算処理をより簡易することができる。

20

【0262】

また、符号化率が小さい第1のパンクチャパターンは、 n フレームの最後部に必ずしも配置する必要はなく、図31から分かるように、 $t_1 \sim t_3$ 、 $t_4 \sim t_6$ 、 $t_7 \sim t_9$ の間に設けるようにすればよい。また、 $t_1 \sim t_3$ 、 $t_4 \sim t_6$ 、 $t_7 \sim t_9$ の期間は、 $B P$ 復号処理期間と、有効データが得られる期間との関係によって特定されるため、 $B P$ 復号処理期間が変わる場合には、第1のパンクチャパターンを配置するのに適する位置も変動する。

30

【0263】

なお、以上の説明では、一例として、畳み込み符号に対し $B P$ 復号を行う場合のパンクチャ方法について説明したが、これに限ったものではなく、本発明のパンクチャ方法は、非特許文献1、非特許文献5～非特許文献7で記述されているような、時不変LDPC-CC、時変LDPC-CCの場合に対しても、同様に実施することができる。

【0264】

(実施の形態8)

本実施の形態では、送信情報系列の先頭部及び後部ほど、LDPC-CC符号化のメモリ長 M が小さくなるように設計した検査行列、及び当該検査行列に基づいたLDPC-CC符号化器の構成について説明する。このような構成にすることで、符号化の開始時及び終了時の符号化器の状態数を少なくすることができる。そのため、例えば、非特許文献8に記載のtail-bittingによるターミネーションを用いる場合に、符号化・復号時において考慮すべき状態数を減少させることができるため、符号化器・復号器の構成が簡易になる。

40

【0265】

図32に、本実施の形態における検査行列の一例を示す。図32の検査行列1900は、符号化率 $R = b / c = 1 / 2$ 、送信情報系列長 n の場合の例である。検査行列1900は、送信情報系列 u_t のインデックスが1に近い部分及び n に近い部分において、符号化のメモリ長 M_1 が、その他の部分のメモリ長 M_2 に比べ小さい点が、図1の検査行列10

50

0と異なっている。

【0266】

なお、検査行列1900は、 $M1 = 3$ 、 $M2 = 5$ の例を示しているが、 $M1$ 及び $M2$ の値はこれに限るものではなく、 $M1 < M2$ の関係を満たしていればよい。特に、 $M1 < M2$ である場合は、tail-bittingの状態数を少なくし、かつ、ターミネーションに関わる部分以外のビットにおけるLDPC-CCの拘束長を長くすることができるので、好ましい。また、メモリ長の数を3つ以上とし、送信情報系列 u_t のインデックスが1に近い部分及び n に近い部分ほど、メモリ長が段階的に小さくなるようにしても良い。

【0267】

図32に示すように、検査行列1900を用いる場合には、送信情報系列 $u_1 \sim u_5$ に対しては、メモリ長 $M1 = 3$ で符号化し、送信符号語系列 $v_{1,1} \sim v_{1,5}$ 、 $v_{2,1} \sim v_{2,5}$ が取得される。送信情報系列 $u_6 \sim u_{n-5}$ に対しては、メモリ長 $M2 = 5$ で符号化し、送信符号語系列 $v_{1,6} \sim v_{1,n-5}$ 、 $v_{2,6} \sim v_{2,n-5}$ が取得される。また、送信情報系列 $u_{n-4} \sim u_n$ に対しては、メモリ長 $M1 = 3$ で符号化し、送信符号語系列、 $v_{1,n-4} \sim v_{1,n}$ 、 $v_{2,n-4} \sim v_{2,n}$ が取得される。

【0268】

非特許文献8に記載されているように、符号化の初期状態及び終了状態の数は、符号化器のメモリ長の2乗となる。したがって、初期状態及び終了状態の符号化に関わる符号化器のメモリ長 $M1$ を小さくすれば、tail-bittingに関わる状態数を少なくすることができる。

【0269】

このように、図32に示す検査行列1900を用いてLDPC-CC符号化を行うことにより、符号化器の初期状態及び終了状態の数を少なくすることができ、その結果、符号化・復号に関わる演算量及び処理遅延を少なくすることができる。

【0270】

ところで、LDPC-CCでは、メモリ長 M が大きければ大きいほど、符号化ゲインが得られ、良好な誤り率特性が得られるという特徴がある。そのため、検査行列1900では、ターミネーションに関わる部分以外のメモリ長を大きくすることで、符号化ゲインを獲得することができる。

【0271】

次に、図32の検査行列1900を用いてLDPC-CC符号化を行うLDPC-CC符号化器の構成を、図面を参照しながら説明する。

【0272】

図33は、本実施の形態におけるLDPC-CC符号化器の要部構成を示すブロック図である。本実施の形態の説明にあたり、図4と同一構成部分には同一符号を付して説明を省略する。図33のLDPC-CC符号化器2000は、図4のLDPC-CC符号化器400に対し、パンクチャ部480を削除し、ウェイト制御部470に代え、ウェイト制御部2010を備えて構成される。

【0273】

ウェイト制御部2010は、ビット数カウンタ460から出力されるカウントビット数と、メモリ長切り替えタイミング情報と、ウェイト制御部2010内に保持している検査行列1900に準じたウェイトパターンとに基づいて、メモリ長切り替えタイミングにおける行列要素 $h_1^{(m)}(t)$ 、 $h_2^{(m)}(t)$ の値をウェイト乗算器420-0～420-M、430-0～430-Mに送出する。ここで、 $M > M1$ である。

【0274】

ここで、メモリ長切り替えタイミング情報とは、検査行列1900のメモリ長 M を切り替える送信情報系列のインデックスを表す。例えば、図32に示される検査行列1900の場合、送信情報系列の先頭部及び後部に対しては $M1 = 3$ を用いて符号化し、それ以外の部分に対しては $M2 = 5$ を用いて符号化するので、メモリ長切り替えタイミング情報は2つの値を取る。つまり、メモリ長として、メモリ長 $M1 = 3$ 、 $M2 = 5$ の2種類を用い

10

20

30

40

50

る場合、メモリ長切り替えタイミング情報は、 $M1 = 3$ から $M2 = 5$ に切り替えるタイミング情報のインデックスと、 $M2 = 5$ から $M1 = 3$ に切り替えるタイミング情報のインデックスとを持っている。

【0275】

図34に、メモリ長 M が $M1 = 3$ 、 $M2 = 5$ の2種類を用いる場合のウェイト制御部2010の構成例を示す。図34のウェイト制御部2010は、セクタ2011、2015、ウェイトパターン記憶部2012、2013を備える。以下では、メモリ長 $M1 = 3$ から $M2 = 5$ に切り替えるタイミングを示すインデックスをメモリ長切り替えタイミング情報1とし、メモリ長 $M2 = 5$ から $M1 = 3$ に切り替えるタイミングを示すインデックスをメモリ長切り替えタイミング情報2とする。

10

【0276】

セクタ2011は、カウントビット数及びメモリ長切り替えタイミング情報1、2を入力し、カウントビット数 > メモリ長切り替えタイミング情報1の場合、ビット数カウントをウェイトパターン記憶部2012に送出する。

【0277】

一方、カウントビット数 > メモリ長切り替えタイミング情報1、かつ、カウントビット数 > メモリ長切り替えタイミング情報2の場合、セクタ2011は、カウントビット数をウェイトパターン記憶部2013に送出する。

【0278】

また、セクタ2011は、カウントビット数 > メモリ長切り替えタイミング情報2の場合、カウントビット数をウェイトパターン記憶部2012に送出する。

20

【0279】

ウェイトパターン記憶部2012は、ウェイトパターン2016で示されるウェイトパターンを保持しており、カウントビット数の増加に伴い周期的に h_{a1} 、 h_{a2} 、 h_{a3} 、 h_{a4} をセクタ2015に出力する。なお、ウェイトパターン2016は、メモリ長 $M1 = 3$ の場合のLDPC-CC検査行列の行列要素 $h_1^{(m)}(t)$ 、 $h_2^{(m)}(t)$ ($m = 0, \dots, 3$)である。メモリ長 $M1 = 3$ の場合、 h_{b1} 、 h_{b2} 、 h_{b3} のウェイトパターンの要素数は8であるが、LDPC-CC符号化器2000は、メモリ長 $M2 = 5$ に対応できるように12個のウェイト乗算器420-0 ~ 420-M ($M = 5$)、430-0 ~ 430-M ($M = 5$)を備える。そのため、ウェイトパターン2016では、 $h_1^{(4)}$ 、 $h_2^{(4)}$ 、 $h_1^{(5)}$ 、 $h_2^{(5)}$ のウェイト要素が、どのパターンにおいても0となっている。

30

【0280】

ウェイトパターン記憶部2013は、ウェイトパターン2017で示されるウェイトパターンを保持しており、ビット数カウントの増加に伴い周期的に h_{b1} 、 h_{b2} 、 h_{b3} をセクタ2015に出力する。なお、ウェイトパターン2017は、メモリ長 $M2 = 5$ の場合のLDPC-CC検査行列の行列要素 $h_1^{(m)}(t)$ 、 $h_2^{(m)}(t)$ ($m = 0, \dots, 5$)である。

【0281】

セクタ2015は、ウェイトパターン記憶部2012から入力される行列要素 h_{a1} 、 h_{a2} 、 h_{a3} 、 h_{a4} 、及び、ウェイトパターン記憶部2013から入力される行列要素 h_{b1} 、 h_{b2} 、 h_{b3} をウェイト乗算器420-0 ~ 420-M、430-0 ~ 430-Mに出力する。

40

【0282】

すなわち、セクタ2015は、セクタ2011と連動して、ビット数カウントとメモリ切り替えタイミング情報1、2との比較結果に応じて、ビット数カウント > メモリ長切り替えタイミング情報1の場合、ウェイトパターン記憶部2012に記憶されるメモリ長 $M1 = 3$ の検査行列に準じたウェイトパターン2016の行列要素を、ウェイト乗算器420-0 ~ 420-M、430-0 ~ 430-Mに出力する。

【0283】

一方、ビット数カウント > メモリ長切り替えタイミング情報1、かつ、ビット数カウ

50

ト メモリ長切り替えタイミング情報 2 の場合、セクタ 2 0 1 5 は、ウェイトパターン記憶部 2 0 1 3 に記憶されるメモリ長 $M 2 = 5$ の検査行列に準じたウェイトパターン 2 0 1 7 の行列要素を、ウェイト乗算器 4 2 0 - 0 ~ 4 2 0 - M , 4 3 0 - 0 ~ 4 3 0 - M に出力する。

【 0 2 8 4 】

また、ビット数カウンタ > メモリ長切り替えタイミング情報 2 の場合、セクタ 2 0 1 5 は、ウェイトパターン記憶部 2 0 1 2 に記憶されるメモリ長 $M 1 = 3$ の検査行列に準じたウェイトパターン 2 0 1 6 の行列要素を、ウェイト乗算器 4 2 0 - 0 ~ 4 2 0 - M , 4 3 0 - 0 ~ 4 3 0 - M に出力する。

【 0 2 8 5 】

以下、上述のように構成された LDPC - CC 符号化器 2 0 0 0 の動作について説明する。

【 0 2 8 6 】

シフトレジスタ 4 1 0 - 1 ~ 4 1 0 - M の状態が $S 1$ にセットされ、シフトレジスタ 4 4 0 - 1 ~ 4 4 0 - M の状態が $S 2$ にセットされている。状態 $S 1$, $S 2$ は、送信系列によって決まる。送信情報系列 $u_1 \sim u_n$ は、シフトレジスタ 4 1 0 - 1、ウェイト乗算器 4 2 0 - 0、及びビット数カウンタ 4 6 0 に順次出力される。

【 0 2 8 7 】

ビット数カウンタ 4 6 0 では、入力される送信情報系列 $u_1 \sim u_n$ 及びターミネーション系列 $x_1, 1 \sim x_1, L$ のビット数がカウントされ、得られたカウントビット数は、ウェイト制御部 2 0 1 0 に出力される。

【 0 2 8 8 】

ウェイト制御部 2 0 1 0 では、カウントビット数とメモリ長切り替えタイミング情報との比較結果に応じて、ウェイトパターン 2 0 1 6、ウェイトパターン 2 0 1 7 のいずれかが選択されて、選択されたウェイトパラメータの行列要素がウェイト乗算器 4 2 0 - 0 ~ 4 2 0 - M , 4 3 0 - 0 ~ 4 3 0 - M に出力される。

【 0 2 8 9 】

具体的には、カウントビット数 < メモリ長切り替えタイミング情報 1 の場合、ウェイト乗算器 4 2 0 - 0 ~ 4 2 0 - M , 4 3 0 - 0 ~ 4 3 0 - M には、メモリ長 $M 1 = 3$ の検査行列に準じたウェイトパターン 2 0 1 6 の行列要素が出力される。

【 0 2 9 0 】

カウントビット数 > メモリ長切り替えタイミング情報 1、かつ、カウントビット数 < メモリ長切り替えタイミング情報 2 の場合、ウェイト乗算器 4 2 0 - 0 ~ 4 2 0 - M , 4 3 0 - 0 ~ 4 3 0 - M には、メモリ長 $M 2 = 5$ の検査行列に準じたウェイトパターン 2 0 1 7 の行列要素が出力される。

【 0 2 9 1 】

また、カウントビット数 > メモリ長切り替えタイミング情報 2 の場合、ウェイト乗算器 4 2 0 - 0 ~ 4 2 0 - M , 4 3 0 - 0 ~ 4 3 0 - M には、メモリ長 $M 1 = 3$ の検査行列に準じたウェイトパターン 2 0 1 6 の行列要素が出力される。

【 0 2 9 2 】

このようにすることで、送信情報系列の先頭部及び後部の符号化に関わるメモリ長を小さくすることができ、結果的に符号化器の初期状態及び終了状態の状態数を削減することができる。

【 0 2 9 3 】

以上のように、本実施の形態によれば、ウェイト制御部 2 0 1 0 は、メモリ長が異なる LDPC - CC 検査行列に準じたウェイトパターン 2 0 1 6 , 2 0 1 7 を記憶し、入力ビットが情報系列の場合に、情報系列の先頭部及び後部では、メモリ長が小さいウェイトパターンを用いるようにした。符号化器の初期状態及び終了状態の状態数は、メモリ長 M が小さければ小さいほど、少なくすることができるので、符号化・復号にかかる演算量及び処理遅延時間を小さくすることができる。

10

20

30

40

50

【 0 2 9 4 】

なお、本実施の形態では、送信情報系列の先頭部及び後部では、メモリ長 $M_1 = 3$ を用いて符号化し、その他の部分では、メモリ長 $M_2 = 5$ を用いて符号化する場合を例に説明したが、これに限るものではなく、送信情報系列の先頭部及び後部において用いるメモリ長 M_1 を 3 未満の任意のメモリ長に設定したり、メモリ長 M_2 からメモリ長 M_1 への減少量を任意とする検査行列を用いても、本発明による、符号化器の初期状態・終了状態の数を少なくするという効果を得ることができる。

【 0 2 9 5 】

(実施の形態 9)

本実施の形態では、複数の多項式から構成される LDPC-CC の符号化において、複数の多項式のうち、メモリ長が最小となる多項式を用いて符号化処理を開始し、メモリ長が最小となる多項式を用いて符号化処理を終了するように設計した LDPC-CC 符号化器の構成について説明する。このような構成にすることで、符号化の開始時及び終了時の符号化器の状態数を少なくすることができるので、符号化・復号にかかる演算量及び処理遅延時間を小さくすることができる。

10

【 0 2 9 6 】

図 3 5 に、本実施の形態における検査行列 2 1 0 0 を示す。検査行列 2 1 0 0 は、多項式の数が 2、すなわち、時変周期 2 の LDPC-CC を定義する検査行列である。検査行列 2 1 0 0 の各行は、検査多項式に対応し、各列は、符号語ビットに対応する。2 つの多項式はそれぞれ、式 (7-1)、式 (7-2) で与えられる。

20

【数 7】

$$(D^{16} + D^{10} + D^6 + 1)X(D) + (D^{17} + D^8 + D^4 + 1)P(D) = 0 \quad \dots (7-1)$$

$$(D^{17} + D^8 + D^4 + 1)X(D) + (D^{19} + D^{12} + D^5 + 1)P(D) = 0 \quad \dots (7-2)$$

【 0 2 9 7 】

以下、式 (7-1) で与えられる多項式を第 1 の多項式 p_1 、式 (7-2) で与えられる多項式を第 2 の多項式 p_2 と呼ぶ。

【 0 2 9 8 】

第 1 の多項式 p_1 のメモリ長は、 $M_{d1} = 16$ 、 $M_{p1} = 17$ である。ここで、添字の d 、 p は、情報ビット、パリティビットを示す。また、第 2 の多項式 p_2 のメモリ長は、 $M_{d2} = 17$ 、 $M_{p2} = 19$ である。すなわち、第 1 の多項式 p_1 と第 2 の多項式 p_2 とでは、第 1 の多項式 p_1 の方が、いずれのメモリ長も小さい。

30

【 0 2 9 9 】

実施の形態 8 で述べたように、メモリ長が小さいほど、符号化器の状態数が少なくなるので、状態数を少なくするためには、メモリ長が小さい多項式を使って符号化を開始し、メモリ長が小さい多項式を使って符号化を終了することが望ましい。

【 0 3 0 0 】

このとき、第 1 の多項式 p_1 で符号化を開始することにより、符号化の開始時にメモリ長の小さい多項式を使って符号化することができる。一方、時変周期 2 であって、2 つの多項式が交互に用いられる場合には、符号化する情報ビットの数によって、符号化の終了時の多項式が変わる。そのため、必ずしもメモリ長の小さい多項式を使って符号化処理を終了することができるとは限らない。

40

【 0 3 0 1 】

そこで、本実施の形態では、多項式調節部 2 2 1 0 を設け、メモリ長の小さい多項式で符号化が必ず終了するようにした。

【 0 3 0 2 】

図 3 6 に、本実施の形態における送信装置の要部構成を示す。図 3 6 の送信装置 2 2 0 0 は、図 1 2 の送信装置 1 0 0 0 に、多項式調節部 2 2 1 0 を更に追加した構成を採る。

【 0 3 0 3 】

50

多項式調節部 2 2 1 0 は、送信情報ビット数 N_{data} と、LDPC-CC の多項式の数 N_{poly} とを用いて、調節ビット数 N_{adj} を求める。

【0304】

図 3 7 に、多項式調節部 2 2 1 0 の構成を示す。多項式調節部 2 2 1 0 は、調節ビット数計算部 2 2 1 1、既知ビット生成部 2 2 1 2 及び既知ビット追加部 2 2 1 3 を備える。

【0305】

ここで、LDPC-CC 符号化部 1 0 1 0 は、メモリ長が最小の多項式 p_1 を用いて LDPC-CC の符号化を開始する。

【0306】

したがって、符号化する情報ビットのインデックスを N_i ($= 1, 2, \dots, N_{data}$) とすると、 $N_i \% N_{poly} = 1$ を満たすインデックスを持つ情報ビットは、多項式 p_1 が用いられて符号化される。ここで、「 $\%$ 」は、除算の余りを求める演算子を表す。

【0307】

調節ビット数計算部 2 2 1 1 は、送信情報ビット数 N_{data} と多項式の数 N_{poly} とから、 $(N_{data} + N_{adj}) \% N_{poly} = 1$ となる最小の調節ビット数 N_{adj} を求め、求めた調節ビット数 N_{adj} を既知ビット生成部 2 2 1 2 に送出する。

【0308】

既知ビット生成部 2 2 1 2 は、調節ビット数 N_{adj} 個の既知ビットを生成する。既知ビットとしては、送信側と受信側とで既知のビット系列であれば任意のビット系列を使うことができる。例えば、既知ビット生成部 2 2 1 2 は、 N_{adj} 個の既知ビットとして、オールゼロ系列などを使うことができる。既知ビット生成部 2 2 1 2 は、生成した既知ビットを既知ビット追加部 2 2 1 3 に送出する。

【0309】

既知ビット追加部 2 2 1 3 は、 N_{adj} 個の既知ビットを送信情報ビット系列の後部に追加し、LDPC-CC 符号化部 1 0 1 0 に送出する。

【0310】

このように、本実施の形態では、多項式調節部 2 2 1 0 が、既知の調節ビットを挿入することにより、メモリ長が最小の多項式 p_1 を用いて確実に符号化の開始・終了することができる。

【0311】

なお、調節ビット数計算部 2 2 1 1 は、調節ビット数 N_{adj} を制御情報生成部 1 0 5 0 に送出する。このようにすることで、 N_{adj} が、制御情報に含まれて送信されるので、 N_{adj} を受信側に伝えることができる。なお、受信側に伝える情報は、 N_{adj} でなくても、受信側で N_{adj} を計算するために必要な情報だけでも良く、例えば、 $N_{data} \% N_{poly}$ 、又は、送信情報ビット数及び多項式の数でもよい。

【0312】

なお、本実施の形態では、多項式の数 2 、すなわち、時変周期が 2 の場合について説明したが、これに限るものではなく、多項式の数 3 以上の場合にも、同様に適用することが可能である。

【0313】

また、本実施の形態では、メモリ長が最小となる多項式が 1 つだけの場合を例に挙げて説明したが、メモリ長が最小となる多項式が複数であっても良い。例えば、全多項式数が 5 で、各メモリ長 $M_{p1} \sim M_{p5}$ が、 $M_{p1} = 16$ 、 $M_{p2} = 18$ 、 $M_{p3} = 16$ 、 $M_{p4} = 20$ 及び $M_{p5} = 19$ の場合、 $N_i \% N_{poly} = 1$ の場合には、多項式 p_1 が使われ、 $N_i \% N_{poly} = 3$ の場合には、多項式 P_3 が使われることになるので、多項式調節部 2 2 1 0 が、 $(N_{data} + N_{adj}) \% N_{poly} = 1$ 又は 3 となる最小の N_{adj} を求めるようにしても良い。

【0314】

また、本実施の形態における多項式調節部 2 2 1 0 は、送信情報系列の後部に既知ビッ

10

20

30

40

50

トを追加することで、符号化の開始時・終了時に用いられる多項式をメモリ長の小さい多項式とする場合について説明したが、多項式調節部 2 2 1 0 の構成及び動作は、これに限るものではない。

【 0 3 1 5 】

図 3 8 に、多項式調節部 2 2 1 0 の別の構成例を示す。図 3 8 の多項式調節部 2 2 1 0 は、剰余計算部 2 2 1 4 及び多項式指示信号生成部 2 2 1 5 を備える。

【 0 3 1 6 】

剰余計算部 2 2 1 4 は、送信情報ビット数 N_{data} 及び多項式の数 N_{poly} から、 $N_{rem} = (N_{data}) \% N_{poly}$ で与えられる剰余 N_{rem} を計算する。剰余計算部 2 2 1 4 は、計算した剰余 N_{rem} を多項式指示信号生成部 2 2 1 5 に送出する。

10

【 0 3 1 7 】

多項式指示信号生成部 2 2 1 5 は、剰余 N_{rem} から多項式指示信号を生成する。多項式指示信号は、以下のように生成する。例えば、時変周期が 2 であり、 $N_{poly} = 2$ の場合を考える。2 つの多項式 p_1 及び多項式 p_2 が、式 (7 - 1)、式 (7 - 2) の場合、多項式 p_1 の方がメモリ長が小さいので、LDPC - CC 符号化器 1 0 1 0 は、多項式 p_1 を用いて符号化を開始する。

【 0 3 1 8 】

したがって、剰余 $N_{rem} = 1$ の場合、 N_{data} 番目の送信情報ビットは、多項式 p_1 が用いられ符号化される。一方、剰余 $N_{rem} = 0$ の場合、 N_{data} 番目の送信情報ビットは、多項式 p_2 が用いられ符号化される。

20

【 0 3 1 9 】

符号化開始時に加え、終了時においても、メモリ長の小さい多項式を用いて符号化すると、状態数を少なくすることができる。そのため、 $N_{rem} = 1$ の場合、多項式指示信号生成部 2 2 1 5 は、最終ビットを多項式 p_1 を用いて符号化するという指示信号を LDPC - CC 符号化器 1 0 1 0 に出力する。

【 0 3 2 0 】

なお、LDPC - CC 符号化器 1 0 1 0 は、多項式 p_1 を用いて符号化を開始し、以降、多項式を切り替えながら符号化を行うので、最終ビットにおいて $N_{rem} = 1$ の場合、特に指示信号がなくても、最終ビットは、多項式 p_1 が用いられ符号化される。そのため、多項式指示信号生成部 2 2 1 5 が、指示信号を出力しないようにしてもよい。

30

【 0 3 2 1 】

一方、最終ビットにおいて $N_{rem} = 0$ の場合、メモリ長の大きい多項式 p_2 が用いられ符号化されることになる。そのため、これを避けるために、多項式指示信号生成部 2 2 1 5 は、最終ビットを多項式 p_1 を用いて符号化するという指示信号を LDPC - CC 符号化器 1 0 1 0 に出力する。この結果、LDPC - CC 符号化に使用される多項式は、「 $p_1, p_2, p_1, p_2, \dots, p_1, p_2, p_1, p_1$ 」の順序になる。

【 0 3 2 2 】

このようにすることで、多項式調節部 2 2 1 0 は、LDPC - CC 符号化器 1 0 1 0 が、最終ビットの符号化に必ずメモリ長の短い多項式を用いるように指示することができるので、符号化の開始時及び終了時の状態数を少なくすることができ、符号化・復号に要する演算量及び処理遅延時間を小さくすることができる。

40

【 0 3 2 3 】

なお、以上の説明では、時変周期 2、すなわち多項式の数が 2 の場合を例に挙げたが、これに限るものではなく、多項式の数が 3 以上の場合であっても本実施の形態を適用することができる。

【 0 3 2 4 】

また、以上の説明では、最小のメモリ長を与える多項式が 1 つの場合を例に挙げたが、これに限るものではなく、最小のメモリ長を与える多項式の数が 2 つ以上の場合であっても良い。この場合、多項式指示信号生成部 2 2 1 5 は、最小のメモリ長の多項式のいずれかで符号化が終了するように多項式指示信号を生成し、LDPC - CC 符号化器 1 0 1 0

50

に送出する。

【0325】

(実施の形態10)

本実施の形態では、実施の形態1等で説明した、複数のウェイトパターンを切り替えて符号化されたLDPC-CC符号語を復号する復号器について説明する。なお、以下では、図3に示す検査行列に準じた複数のウェイトパターンを切り替えて符号化されたLDPC-CC符号語を復号する場合を例に説明する。

【0326】

図39に、本実施の形態における受信装置の要部構成を示す。なお、図39の受信装置2300において、図15の受信装置1200と同一の構成については、同一の符号を付してその説明を省略する。図39の受信装置2300は、受信アンテナ1201、無線部1202、直交復調部1203、チャネル変動推定部1204、制御情報検波部1205、対数尤度演算部1206、デインタリーブ部1207及びsum-product復号部2310を主に備える。

10

【0327】

sum-product復号部2310は、デインタリーブ部1207から送出される受信対数尤度比と制御情報検波部1205から送出される送信情報系列長とを入力とし、sum-product復号を行い、復号結果を得る。

【0328】

図40に、sum-product復号部2310の構成を示す。sum-product復号部2310は、記憶部2311、行処理演算部2312、列処理演算部2313及び切替部2314を備える。なお、記憶部2311、行処理演算部2312及び列処理演算部2313は、行列処理演算部2315を構成する。

20

【0329】

記憶部2311は、受信対数尤度比、行処理によって得られる外部値 α_{mn} 、及び、列処理によって得られる事前値 β_{mn} を保持する。

【0330】

行処理演算部2312は、必要な事前値 β_{mn} を記憶部2311から読み込み、LDPC-CCの検査行列Hの行方向のウェイトパターンにしたがって行処理演算を行う。ウェイトパターンは、切り替えタイミング情報に基づいて切り替えられる。行処理演算部2312は、行処理演算において、事前値 β_{mn} を用いて単一パリティ検査符号の復号を行い、外部値 α_{mn} を求める。

30

【0331】

第m番目の行の処理を説明する。

【0332】

$H_{mn} = 1$ を満たす全ての組(m, n)に対して、次の更新式(8)を利用して外部値 α_{mn} を更新する。

【数8】

$$\alpha_{mn} = \left(\prod_{n \in A(m) \setminus n} \text{sign}(\beta_{mn'}) \right) \Phi \left(\sum_{n \in A(m) \setminus n} \Phi(|\beta_{mn'}|) \right) \quad \dots (8)$$

40

ここで、 $\Phi(x)$ は、Gallagerのf関数と呼ばれ、次式で定義される。

【数9】

$$\Phi(x) = \ln \frac{\exp(x) + 1}{\exp(x) - 1} \quad \dots (9)$$

【0333】

列処理演算部2313は、必要な外部値 α_{mn} を記憶部2311から読み込み、LDPC-CCの検査行列Hの列方向のウェイトパターンにしたがって列演算処理を行う。ウェイトパターンは、切り替えタイミング情報に基づいて切り替えられる。

50

【 0 3 3 4 】

列処理演算部 2 3 1 3 は、列処理演算において、入力対数尤度比 β_n と外部値 β_{mn} とを用いて繰り返し符号の復号を行い、事前値 β_{mn} を求める。

【 0 3 3 5 】

第 m 番目の列の処理を説明する。

【 0 3 3 6 】

$H_{m,n} = 1$ を満たす全ての組 (m, n) に対して、次の更新式 (1 0) を利用して β_{mn} を更新する。ただし、 $q = 1$ の場合のみ、 $\beta_{m,n} = 0$ として計算する。

【 数 1 0 】

$$\beta_{mn} = \lambda_n + \sum_{m \in B(n)/m} \alpha_{m'n} \quad \cdots (10)$$

10

【 0 3 3 7 】

切替部 2 3 1 4 は、ウェイトパターン切り替えタイミングに基づいて、行処理演算部 2 3 1 2 及び列処理演算部 2 3 1 3 のウェイトパターンを切り替える。なお、ウェイトパターン切り替えタイミングは、外部より入力される構成としても良いし、切替部 2 3 1 4 の内部で生成する構成としても良い。図 4 0 は、切替部 2 3 1 4 が、受信対数尤度比及び送信情報系列長 n を入力し、受信対数尤度比をカウントし、当該カウント値と送信情報系列長 n を用いて、ウェイトパターン切り替えタイミングを計算する構成を示している。

【 0 3 3 8 】

図 4 1 に、行処理演算部 2 3 1 2 の構成を示す。行処理演算部 2 3 1 2 は、セレクタ 2 3 1 2 - 1, 2 3 1 2 - 2、第 1 演算部 2 3 1 2 - 3、及び第 2 演算部 2 3 1 2 - 4 を備える。

20

【 0 3 3 9 】

セレクタ 2 3 1 2 - 1, 2 3 1 2 - 2 は、切り替えタイミング情報に基づいて、事前値の送信先及び外部値の受信先を切り替える。

【 0 3 4 0 】

第 1 演算部 2 3 1 2 - 3 は、ウェイトパターン 4 7 5 に基づいた行処理演算を行う。ウェイトパターン 4 7 5 は、符号化側のウェイト制御部 4 7 0 のウェイトパターン記憶部 4 7 2 が備えるパターンである。また、第 2 演算部 2 3 1 2 - 4 は、ウェイトパターン 4 7 6 に基づいた行処理演算を行う。ウェイトパターン 4 7 6 は、符号化側のウェイト制御部 4 7 0 のウェイトパターン記憶部 4 7 3 が備えるパターンである。

30

【 0 3 4 1 】

図 4 2 に、列処理演算部 2 3 1 3 の構成を示す。列処理演算部 2 3 1 3 は、セレクタ 2 3 1 3 - 1, 2 3 1 3 - 2、第 1 演算部 2 3 1 3 - 3 及び第 2 演算部 2 3 1 3 - 4 を備える。セレクタ 2 3 1 3 - 1, 2 3 1 3 - 2 は、切り替えタイミング情報に基づいて、事前値の送信先及び外部値の受信先を切り替える。

【 0 3 4 2 】

第 1 演算部 2 3 1 3 - 3 は、ウェイトパターン 2 3 1 3 - 5 に基づいた列処理演算を行う。また、第 2 演算部 2 3 1 3 - 4 は、ウェイトパターン 2 3 1 3 - 6 に基づいた列処理演算を行う。ウェイトパターン 2 3 1 3 - 5 及びウェイトパターン 2 3 1 3 - 6 は、符号化側のウェイト制御部 4 7 0 が保持する図 3 の検査行列に準じたウェイトパターンである。

40

【 0 3 4 3 】

切替部 2 3 1 4 は、受信対数尤度比のカウント値 C が、 $C < 2n$ では、ウェイトパターン 4 7 5 を用いることを指示する切り替えタイミング情報を生成し、 $2n + 1 - C < 2(n + 5)$ では、ウェイトパターン 4 7 6 を用いることを指示するウェイト切り替えタイミング情報を生成し、 $2n + 1 - C$ 以上では、ウェイトパターン 4 7 5 を用いることを指示するウェイト切り替えタイミング情報を生成する。

【 0 3 4 4 】

また、切替部 2 3 1 4 は、受信対数尤度比のカウント値 C が $C < 2(n - 5)$ では、ウ

50

エイトパターン 2 3 1 3 - 5 を用いることを指示する切り替えタイミング情報を生成し、 $2(n-5) < 2n$ では、ウェイトパターン 2 3 1 3 - 6 を用いることを指示する切り替えタイミング情報を生成し、 $2n < 2n$ では、ウェイトパターン 2 3 1 3 - 5 を用いることを指示する切り替えタイミング情報を生成する。

【0345】

このようにして生成した切り替えタイミング情報を、切替部 2 3 1 4 は、行処理演算部 2 3 1 2 及び列処理演算部 2 3 1 3 に出力する。これにより、行処理演算部 2 3 1 2 及び列処理演算部 2 3 1 3 は、符号化側において用いられた LDPC - CC 検査行列に準じて復号処理を施すことができるようになる。

【0346】

sum-product 復号部 2 3 1 0 は、行処理及び列処理を所定の回数だけ繰り返した後、事後対数尤度比を得る。

【0347】

以上のように、本実施の形態における受信装置は、sum-product 復号部 2 3 1 0 を備え、sum-product 復号部 2 3 1 0 は、切り替えタイミング情報により、行処理演算、列処理演算に使用するウェイトパターンを切り替える構成とした。このようにすることで、実施の形態 1 で説明した、ウェイトパターンを切り替えて符号化した符号語の sum-product 復号を実現することができる。

【0348】

なお、本実施の形態では、図 3 で示した検査行列で定義される LDPC - CC について説明を行ったが、これに限るものではなく、例えば他のメモリ長、ウェイトパターンの LDPC - CC であってもよい。

【0349】

また、本実施の形態では、実施の形態 1 における符号化器に対応した受信装置及び復号器について説明したが、これに限るものではなく、例えば実施の形態 2 における符号化器に対しても、切り替えタイミング情報及びウェイトパターンをそれに応じたものに変更することにより、実施の形態 2 で説明した符号化器に対応する受信装置及び復号器を構成することができる。

【0350】

本発明は上記すべての実施の形態に限定されず、種々変更して実施することが可能である。例えば、上記実施の形態では、主に、符号化器及び送信装置で実現する場合について説明しているが、これに限られるものではなく、電灯線通信装置で実現する場合においても適用可能である。

【0351】

また、この符号化方法及び送信方法をソフトウェアとして行うことも可能である。例えば、上記符号化方法及び通信方法を実行するプログラムを予め ROM (Read Only Memory) に格納しておき、そのプログラムを CPU (Central Processor Unit) によって動作させるようにしても良い。

【0352】

また、上記符号化方法及び送信方法を実行するプログラムをコンピュータで読み取り可能な記憶媒体に格納し、記憶媒体に格納されたプログラムをコンピュータの RAM (Random Access Memory) に記録して、コンピュータをそのプログラムにしたがって動作させるようにしても良い。

【0353】

また、本発明は、無線通信に限らず、電灯線通信 (PLC: Power Line Communication)、可視光通信、光通信においても有用であることは言うまでもない。

【0354】

本発明の LDPC - CC 符号化器の一つの態様は、複数のシフトレジスタと、前記シフトレジスタの出力にウェイトを乗算する複数のウェイト乗算部と、前記複数のウェイト乗算部の出力を mod 2 加算する mod 2 加算器と、符号化する入力ビットのビット数をカ

10

20

30

40

50

ウントするビット数カウンタと、前記ビット数に応じて前記複数のウェイト乗算部のウェイトを制御するウェイト制御部と、を具備する構成を採る。

【0355】

この構成によれば、LDPC-CC検査行列を用いたLDPC-CC符号化を行うことができる。

【0356】

本発明のLDPC-CC符号化器の一つの態様は、前記ウェイト制御部は、LDPC-CC検査行列に準じた第1のウェイトパターンと、前記LDPC-CC検査行列を変形した検査行列に準じた第2のウェイトパターンと、を記憶し、前記入力ビットが情報系列の場合に、前記第1のウェイトパターンを用い、前記入力ビットがターミネーション系列の場合に、前記第2のウェイトパターンを用いる構成を採る。

10

【0357】

この構成によれば、入力ビットが情報系列とターミネーション系列との場合で、シフトレジスタのウェイト値を切り替えることができるので、入力ビットがターミネーション系列の場合に、送信符号語系列の値に関わらず、送信符号語系列と乗算されるウェイト値を0にして、終端処理を完了することができ、送信するターミネーション系列を削減することができる。

【0358】

本発明のLDPC-CC符号化器の一つの態様は、前記第2のウェイトパターンは、前記LDPC-CC検査行列の各行の一番右の1を0に変えた検査行列に準じたウェイトパターンである構成を採る。

20

【0359】

この構成によれば、入力ビットがターミネーション系列の場合に、送信符号語系列の値に関わらず、送信符号語系列と乗算されるウェイト値を0にすることができ、送信するターミネーション系列を削減することができる。

【0360】

本発明のLDPC-CC符号化器の一つの態様は、前記第2のウェイトパターンは、前記LDPC-CC検査行列の各行の一番右の1を、ターミネーション系列のパリティ検査に用いられない列に左シフトして得られる検査行列に準じたウェイトパターンである構成を採る。

30

【0361】

この構成によれば、ターミネーション系列復号時に、行方向の1の数(行重み)が変わらないので、Sum-product復号における繰り返し符号の復号処理による符号化ゲインを維持することができる。

【0362】

本発明のLDPC-CC符号化器の一つの態様は、前記第2のウェイトパターンは、前記LDPC-CC検査行列の各行の一番右の1を、行ごとに同じ数だけ左にシフトして得られる検査行列に準じたウェイトパターンである構成を採る。

【0363】

この構成によれば、記憶するウェイトパターンの種類が少なくなる。

40

【0364】

本発明のLDPC-CC符号化器の一つの態様は、前記第2のウェイトパターンは、前記LDPC-CC検査行列のうち、検査ビットに対応する行の1を0に変えて得られる検査行列に準じたウェイトパターンである構成を採る。

【0365】

この構成によれば、送信符号語系列の値に関わらず、送信符号語系列と乗算されるウェイト値を0にすることができ、送信するターミネーション系列を大幅に削減することができる。

【0366】

本発明のLDPC-CC符号化器の一つの態様は、前記入力ビットがターミネーション

50

系列の場合に、前記ターミネーション系列のシステマティックビットをパンクチャするパンクチャ部、をさらに具備する構成を採る。

【0367】

この構成によれば、ターミネーション系列を、例えば、受信側とで既知であるゼロ系列とした場合に、当該ゼロ系列をパンクチャすることで、伝送効率の低下を抑圧することができる。

【0368】

本発明のLDPC-CC符号化器の一つの態様は、前記ウェイト制御部は、メモリ長が異なるLDPC-CC検査行列に準じたウェイトパターンを複数記憶し、前記入力ビットが情報系列の場合に、前記情報系列の後部ほど、メモリ長が小さい前記ウェイトパターンを用いる構成を採る。

10

【0369】

この構成によれば、情報系列の終部ほど、メモリ長を小さくして、ターミネーション系列長をより短くすることができ、伝送効率の劣化を抑えることができる。

【0370】

本発明のLDPC-CC符号化器の一つの態様は、前記ウェイト制御部は、符号化率が異なるLDPC-CC検査行列に準じたウェイトパターンを複数記憶し、前記入力ビットが情報系列の場合に、前記情報系列の終端ほど、符号化率が低い前記ウェイトパターンを用いる構成を採る。

【0371】

20

この構成によれば、情報系列の終部ほど、符号化率を低くすることができるので、ターミネーション系列を削除した場合においても、終端処理における誤り耐性を強化することができる。

【0372】

本発明の送信装置の一つの態様は、入力ビットに対しLDPC-CC符号化を施すLDPC-CC符号化手段と、LDPC-CC符号化後の系列に含まれるターミネーション系列をパンクチャするターミネーション系列パンクチャ手段と、を具備する構成を採る。

【0373】

この構成によれば、ターミネーション系列送信量を削減することができ、伝送効率の低下を抑圧することができる。

30

【0374】

本発明の送信装置の一つの態様は、前記ターミネーション系列パンクチャ手段は、前記ターミネーション系列の後部ほど、パンクチャするビットの割合を多くする構成を採る。

【0375】

この構成によれば、LDPC-CC符号化の終端処理において、送信符号語系列への影響度が少ないターミネーション系列の後部を優先的にパンクチャすることができるので、受信情報系列の誤り率特性の劣化を抑えつつ、ターミネーション系列送信量を削減することができる。

【0376】

本発明の送信装置の一つの態様は、前記ターミネーション系列パンクチャ手段は、前記ターミネーション系列を全てパンクチャする構成を採る。

40

【0377】

この構成によれば、送信側と受信側との間でターミネーション系列を既知とした場合に、ターミネーション系列送信量を大幅に削減することができる。

【0378】

本発明の送信装置の一つの態様は、前記ターミネーション系列パンクチャ手段がパンクチャしたターミネーション系列を蓄積する第1のバッファ、をさらに具備し、通信相手から再送要求が通知された場合に、前記第1のバッファに蓄積されたターミネーション系列を送信する構成を採る。

【0379】

50

この構成によれば、LDPC-CCパリティ検査により、誤りが検出された位置がターミネーション系列の位置であると識別された場合に、パンクチャしたターミネーション系列のみを再送することができるので、再送による伝送効率の低下を抑えることができる。

【0380】

本発明の送信装置の一つの態様は、前記LDPC-CC符号化手段の前段に前記入力ビットを蓄積する第2のバッファ、をさらに具備する構成を採る。

【0381】

この構成によれば、再送要求時に、同一の送信情報系列に対し、例えば誤り訂正能力が高い異なる符号化率でLDPC-CC符号化を施し送信することができ、通信相手の受信装置で正しく受信される割合を高くすることができる。

10

【0382】

本発明の送信装置の一つの態様は、前記第2のバッファは、前記入力ビットの順序を後ろから順に並び替える構成を採る。

【0383】

この構成によれば、入力ビットの順序を並び替える蓄積するようにすることで、再送時に、初回送信時とは異なる系列順序で送信されるようになるので、初回送信時に誤りの影響を受けたビットが再度誤る割合を減らすことができる。

【0384】

本発明の送信装置の一つの態様は、前記第2のバッファは、前記入力ビットの順序を後ろから順に並び替える構成を採る。

20

【0385】

この構成によれば、初回送信時に誤りの影響を受けやすい後方部のビットが、再送時には前方部でLDPC-CC符号化されるようになるので、再送時に、同一のビットが誤る確率を減少させることができる。

【0386】

本発明の送信装置の一つの態様は、情報系列を2つに分割することにより第1及び第2の情報系列を取得する情報系列分割手段と、前記第2の情報系列を符号化する外符号化手段と、前記第1の情報系列と、外符号化後の前記第2の情報系列の順序を並び替える並び替え手段と、をさらに具備し、前記LDPC-CC符号化手段は、前記並び替え手段からの出力ビットに対しLDPC-CC符号化を施す構成を採る。

30

【0387】

この構成によれば、情報系列を前方と後方とに分割することにより2つの系列を取得した場合に、終端処理において誤りの影響を受けやすい後方の情報系列に外符号化を施すことができるので、受信情報系列の後部に誤りが発生した場合において、当該誤りを訂正することができる、受信品質の劣化を抑圧することができる。

【0388】

本発明の送信装置の一つの態様は、前記外符号化手段は、低密度パリティ検査ブロック符号(LDPC-BC)化を行う構成を採る。

【0389】

この構成によれば、外符号化に用いる回路規模を小さくしつつ、高い誤り訂正能力を得ることができる。

40

【0390】

本発明の送信装置の一つの態様は、前記並び替え手段は、前記第1の情報系列および符号化後の前記第2の情報系列を、前記外符号化手段によって得られるパリティ系列を優先的に前方に配置するように並び替える構成を採る。

【0391】

この構成によれば、外符号化によって得られた外符号語系列の一部を送信情報系列の前方に優先的に配置することができるので、外符号化の符号化利得を向上することができる。

【0392】

50

本発明の送信装置の一つの態様は、入力ビットに対しLDPC-CC符号化を施し送信符号語ビットを生成するLDPC-CC符号化手段と、受信側における、前記送信符号語ビットの処理単位ごとに、第1のパンクチャパターンと、前記第1のパンクチャパターンに比べより多くのビットを間引く第2のパンクチャパターンとを用いて、前記送信符号語ビットをパンクチャするパンクチャ手段と、を具備する構成を採る。

【0393】

本発明の送信装置の一つの態様は、前記パンクチャ手段は、前記送信符号語ビットの処理単位の先頭に、前記第1のパンクチャパターンを用いる構成を採る。

【0394】

これらの構成によれば、受信側（復号側）においてBP復号時に用いる検査行列において、悪影響を与える行数を削減することができるので、受信品質の劣化を抑圧することができる。

10

【0395】

本発明の送信装置の一つの態様は、前記パンクチャ手段は、前記送信符号語ビットの処理単位の後部に、前記第1のパンクチャパターンを用いる構成を採る。

【0396】

この構成によれば、BP復号処理期間の前方と後方の双方に、信頼度が伝搬される検査行列の行が含まれるようになるので、効率的に、信頼度を伝搬させることができるようになる。

【0397】

20

本発明の送信装置の一つの態様は、前記パンクチャ手段は、受信側におけるBP（Belief Propagation）復号処理単位時間と、前記送信符号語ビットの処理単位とのタイミングに基づいて、前記第1のパンクチャパターンが用いられる前記送信符号語ビットが、前記BP復号処理単位時間内に、複数含まれるように、前記送信符号語ビットの処理単位に、前記第1のパンクチャパターンと前記第2のパンクチャパターンとを割り当てる構成を採る。

【0398】

この構成によれば、BP復号処理期間に、信頼度が伝搬される検査行列の行により復号することができる範囲が多くなるので、効率的に、信頼度を伝搬させることができるようになる。

30

【0399】

本発明のLDPC-CC符号化器の一つの態様は、前記ウェイト制御部は、メモリ長が異なるLDPC-CC検査行列に準じたウェイトパターンを複数記憶し、前記入力ビットの先頭部および前記入力ビットの後部の符号化ほど、メモリ長が小さい前記ウェイトパターンを用いる構成を採る。

【0400】

本発明のLDPC-CC符号化器の一つの態様は、前記ウェイト制御部は、複数の多項式に準じたウェイトパターンを複数記憶し、前記入力ビットの第1ビット目の符号化に、前記複数の多項式のうちメモリ長が最小の多項式に準じた前記ウェイトパターンを用い、前記入力ビットの最終ビットの符号化に、前記複数の多項式のうちメモリ長が最小の多項式に準じた前記ウェイトパターンを用いる構成を採る。

40

【0401】

本発明のLDPC-CC符号化器の一つの態様は、前記入力ビットの数及び前記多項式の数に応じた数の既知ビットを、前記入力ビットの後部に付加する調節ビット付加部、をさらに具備する構成を採る。

【0402】

本発明のLDPC-CC符号化器の一つの態様は、前記ウェイト制御部は、前記入力ビットの最終ビットの符号化に、前記複数の多項式のうちメモリ長が最小のLDPC-CC検査行列に準じたウェイトパターンを用いる構成を採る。

【0403】

50

これらの構成によれば、LDPC-CCの符号化の開始時及び終了時の状態数を少なくすることができ、符号化・復号に要する演算量及び処理遅延時間を小さくすることができる。

【0404】

本発明のLDPC-CC復号器の一つの態様は、低密度パリティ検査畳み込み符号(LDPC-CC: Low-Density Parity-Check Convolutional Codes)を復号するLDPC-CC復号器であって、LDPC-CC検査行列に準じた複数のウェイトパターンを備える行列演算器と、ウェイトパターン切り替えタイミングに基づいて前記複数のウェイトパターンを切り替える切替部と、を具備する構成を採る。

【0405】

この構成によれば、LDPC-CC検査行列を用いた符号化されたLDPC-CC符号語を当該LDPC-CC検査行列に準じて復号することができる。

【0406】

本発明のLDPC-CC復号器の一つの態様は、前記切替部は、受信対数尤度比の数をカウントし、当該カウント値と送信情報系列長とを用いて、前記ウェイトパターン切り替えタイミングを算出するウェイト切り替えタイミング計算部、を具備する構成を採る。

【0407】

この構成によれば、LDPC-CC検査行列のウェイトパターンが切り替わるような場合においても、当該LDPC-CC検査行列に準じてLDPC-CC符号語を復号することができる。

【産業上の利用可能性】

【0408】

本発明にかかる低密度パリティ検査畳み込み符号(LDPC-CC)符号化器及び送信装置は、LDPC-CC符号化・復号化に必要なターミネーション系列の量を削減し、伝送効率の劣化を抑えつつ、誤り訂正符号化・復号化を行うことができ、LDPC-CC符号化を用いて誤り訂正符号化を施すLDPC-CC符号化器及びLDPC-CC復号器などに有用である。

【図面の簡単な説明】

【0409】

【図1】従来のターミネーション系列を付加した場合のLDPC-CC符号の検査行列の一例を示す図

【図2】従来のターミネーション系列を付加しない場合のLDPC-CC符号の検査行列の一例を示す図

【図3】本発明の実施の形態1における検査行列の一例を示す図

【図4】実施の形態1におけるLDPC-CC符号化器の構成を示すブロック図

【図5】実施の形態1におけるウェイト制御部の構成を示すブロック図

【図6】実施の形態1における検査行列の別の例を示す図

【図7】本発明の実施の形態2における検査行列の例を示す図

【図8】実施の形態2におけるLDPC-CC符号化器の構成を示すブロック図

【図9】実施の形態2におけるウェイト制御部の構成を示すブロック図

【図10】本発明の実施の形態3における検査行列の例を示す図

【図11】実施の形態3におけるLDPC-CC符号化器の構成を示すブロック図

【図12】本発明の実施の形態4における送信装置の要部構成を示すブロック図

【図13】実施の形態4におけるパンクチャの一例を説明するための図

【図14】本発明の実施の形態5における送信装置の要部構成を示すブロック図

【図15】実施の形態5における受信装置の要部構成を示すブロック図

【図16】実施の形態5における送信装置の別の要部構成を示すブロック図

【図17】本発明の実施の形態6における送信装置の要部構成を示すブロック図

【図18】実施の形態6における受信装置の要部構成を示すブロック図

【図19】実施の形態6における受信装置の別の要部構成を示すブロック図

10

20

30

40

50

【図20】本実施の形態7におけるLDPC-CC検査行列の構成を示す図

【図21】一般的なパンクチャ方法を説明するための図

【図22】一般的なパンクチャ方法による送信符号語系列vとLDPC-CC検査行列Hとの対応を示す図

【図23】実施の形態7におけるパンクチャ方法を説明するための図

【図24】実施の形態7におけるパンクチャ方法による送信符号語系列vとLDPC-CC検査行列Hとの対応を示す図

【図25】実施の形態7における送信装置の別の要部構成を示すブロック図

【図26】実施の形態7におけるパンクチャパターンの一例を示す図

【図27】実施の形態7における別のパンクチャパターンを示す図 10

【図28】実施の形態7における別のパンクチャパターンを示す図

【図29】実施の形態7における別のパンクチャパターンを示す図

【図30】実施の形態7における別のパンクチャパターンを示す図

【図31】復号処理タイミングを説明するための図

【図32】本発明の実施の形態8における検査行列の一例を示す図

【図33】実施の形態8におけるLDPC-CC符号化器の構成を示すブロック図

【図34】実施の形態8におけるウェイト制御部の構成を示すブロック図

【図35】本発明の実施の形態9における検査行列の一例を示す図

【図36】実施の形態9における送信装置の構成を示すブロック図

【図37】実施の形態9における多項式調節部の構成を示すブロック図 20

【図38】実施の形態9における多項式調節部の別の構成を示すブロック図

【図39】本発明の実施の形態10における受信装置の構成を示すブロック図

【図40】実施の形態10におけるsum-product復号部の構成を示すブロック図

【図41】実施の形態10における行処理演算部の構成を示すブロック図

【図42】実施の形態10における列処理演算部の構成を示すブロック図

【図43】従来のLDPC-CC符号の検査行列の一例を示す図

【図44】従来のLDPC-CC符号化器の要部構成の一例を示す図

【符号の説明】

【0410】

100, 200, 300, 500, 600, 800 検査行列 30

400, 700, 900, 1010, 1320 LDPC-CC符号化器

410-1~410-M, 440-1~440-M, 910-1-1~910-c-M
シフトレジスタ

420-0~420-M, 430-0~430-M, 920-1-1-0~920-c
-c-M ウェイト乗算器

450, 930-1~930-c-1 mod2加算器

460, 940 ビット数カウンタ

470, 710, 950, 2010 ウェイト制御部

471, 474, 711, 715, 2011, 2015, 2312-1, 2312-2
, 2313-1, 2313-2 セレクタ 40

472, 473, 712, 713, 714, 2012, 2013 ウェイトパターン記憶部

480 パンクチャ部

1000, 1100, 1300, 1400, 1800, 2200 送信装置

1020 ターミネーション系列パンクチャ部

1030 インタリーブ部

1040 変調部

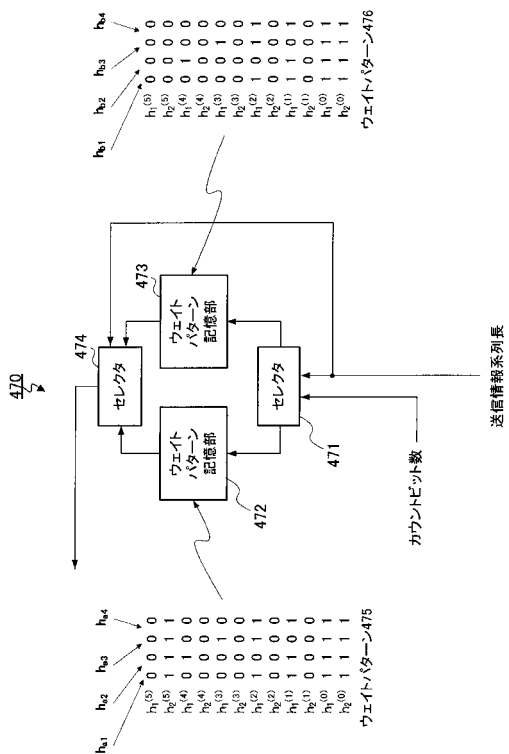
1050 制御情報生成部

1060 無線部

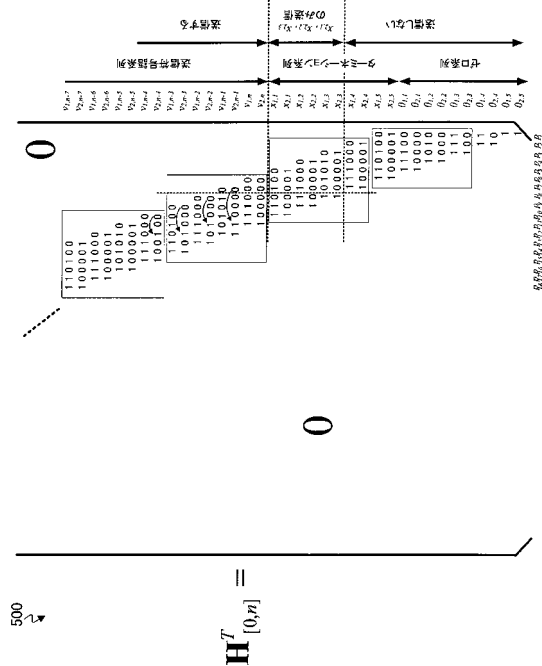
1070, 1213 送信アンテナ 50

1 1 1 0 , 1 1 2 0 , 1 2 1 0 , 1 3 1 0	バッファ	
1 1 3 0 , 1 2 0 1	受信アンテナ	
1 1 4 0	応答信号検出部	
1 1 5 0	再送制御部	
1 1 6 0	送信系列選択部	
1 2 0 0 , 1 5 0 0 , 1 6 0 0 , 2 3 0 0	受信装置	
1 2 0 2	無線部	
1 2 0 3	直交復調部	
1 2 0 4	チャネル変動推定部	
1 2 0 5	制御情報検波部	10
1 2 0 6	対数尤度演算部	
1 2 0 7	デインタリーブ部	
1 2 0 8	ターミネーション系列デパンクチャ部	
1 2 0 9	Sum-product復号部	
1 2 1 1	誤り検出部	
1 2 1 2	応答信号生成部	
1 4 1 0	情報系列分割部	
1 4 2 0	外符号化部	
1 4 3 0 , 1 5 0 1	並び替え部	
1 5 0 2	受信情報系列分割部	20
1 5 0 3	外符号復号部	
1 6 0 1	受信情報系列結合部	
1 8 1 0	パンクチャ部	
1 8 1 1	第1パンクチャ部	
1 8 1 2	第2パンクチャ部	
1 8 1 3	切り替え部	
2 2 1 0	多項式調節部	
2 2 1 1	調節ビット数計算部	
2 2 1 2	既知ビット生成部	
2 2 1 3	既知ビット追加部	30
2 2 1 4	剰余計算部	
2 2 1 5	多項式指示信号生成部	
2 3 1 0	sum-product復号部	
2 3 1 1	記憶部	
2 3 1 2	行処理演算部	
2 3 1 3	列処理演算部	
2 3 1 4	切替部	
2 3 1 2 - 3 , 2 3 1 3 - 3	第1演算部	
2 3 1 2 - 4 , 2 3 1 3 - 4	第2演算部	
2 3 1 5	行列処理演算部	40

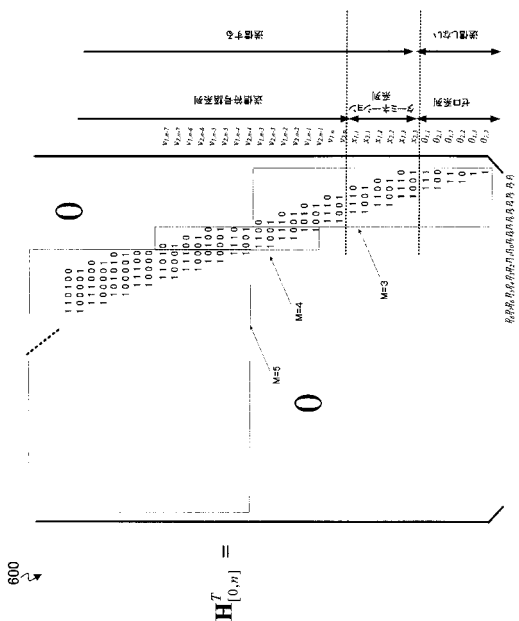
【図5】



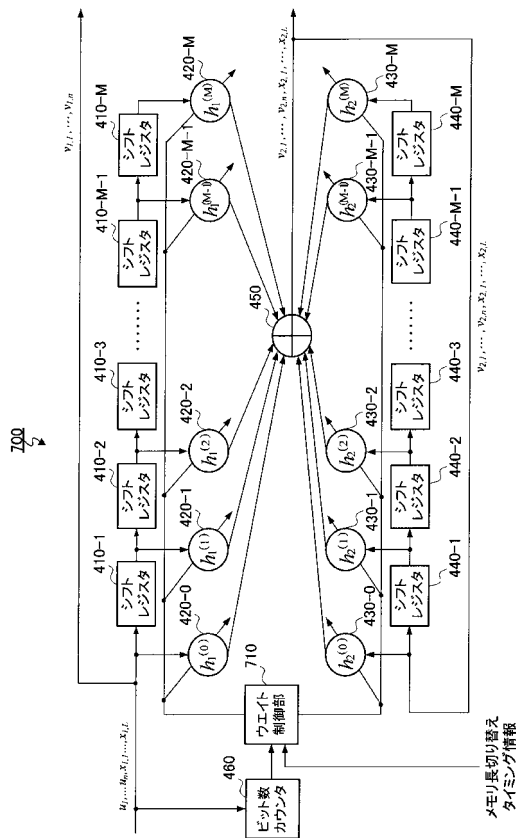
【図6】



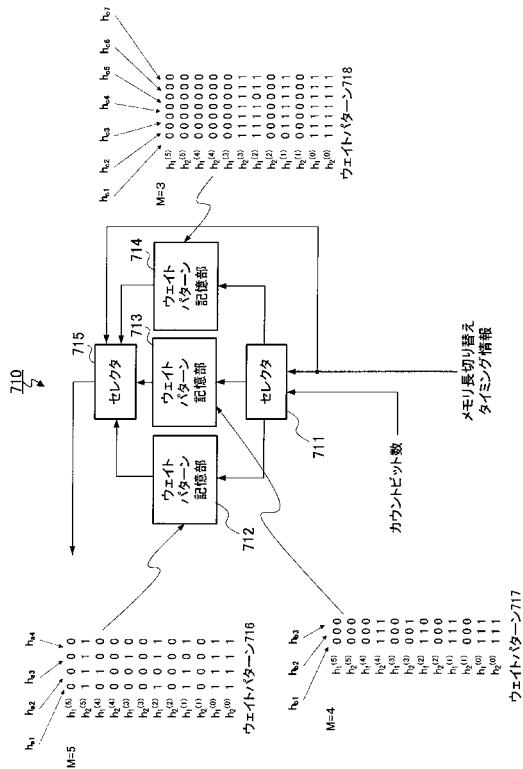
【図7】



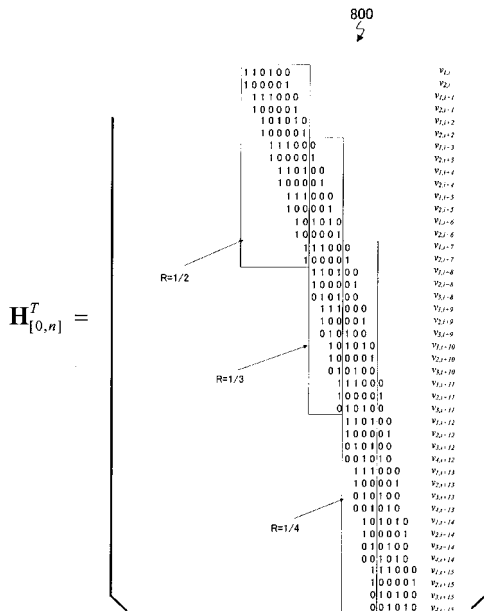
【図8】



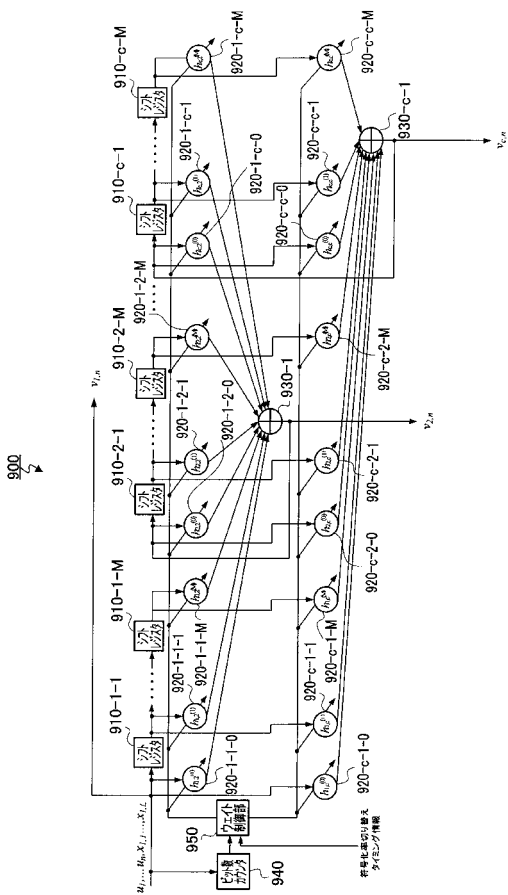
【図9】



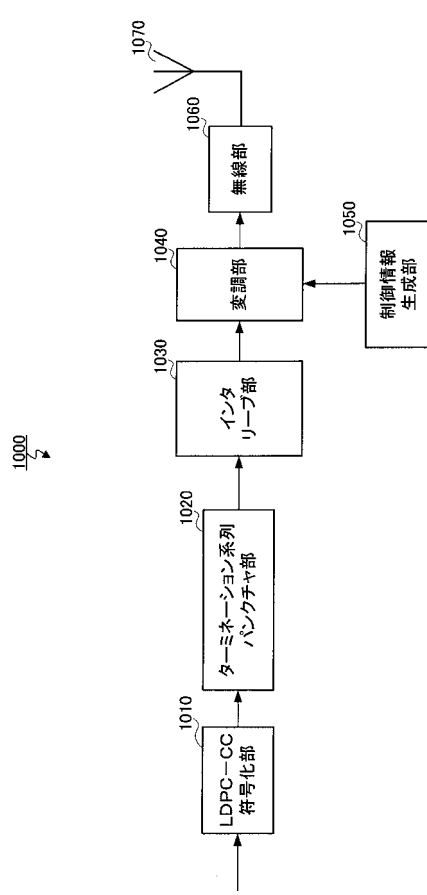
【図10】



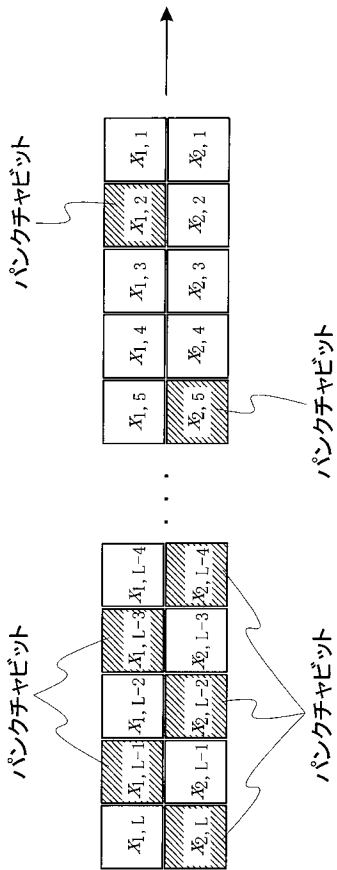
【図11】



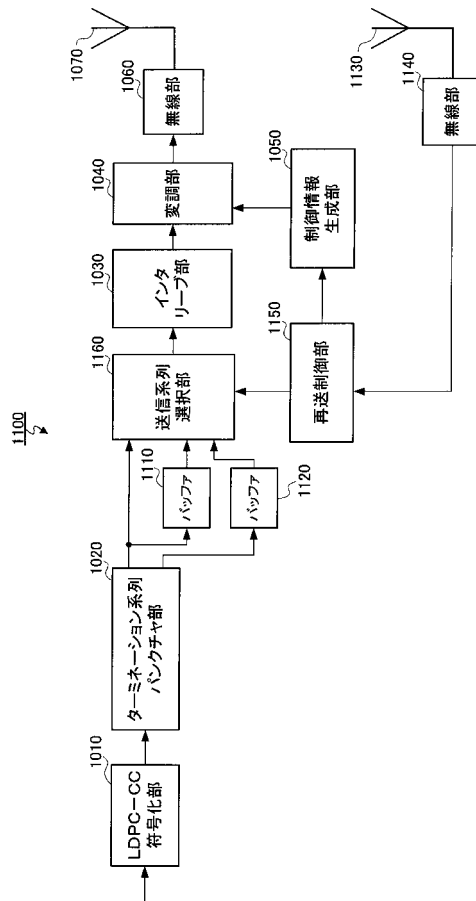
【図12】



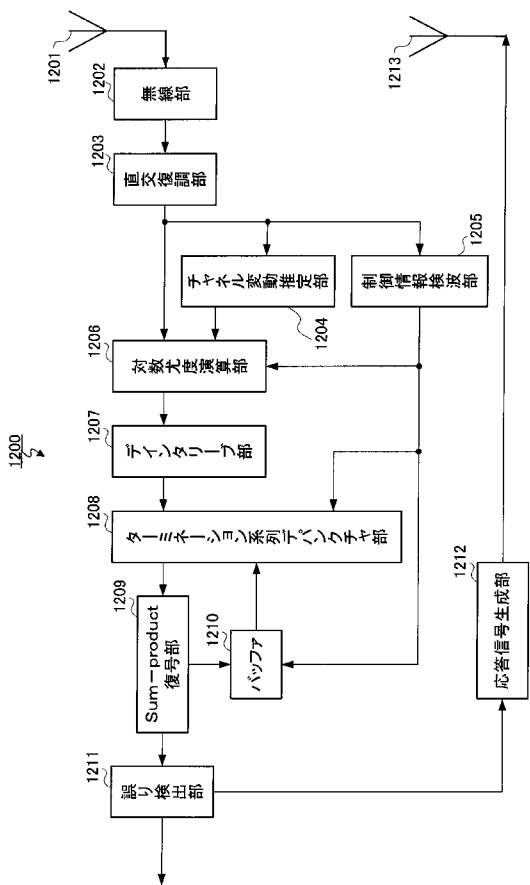
【図13】



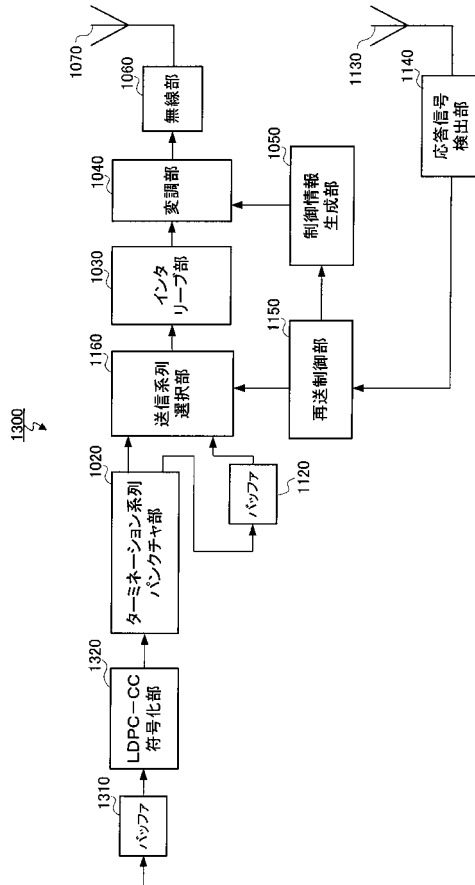
【図14】



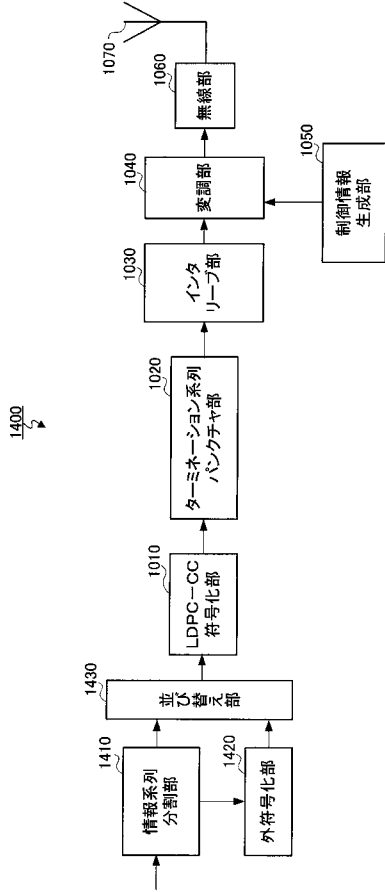
【図15】



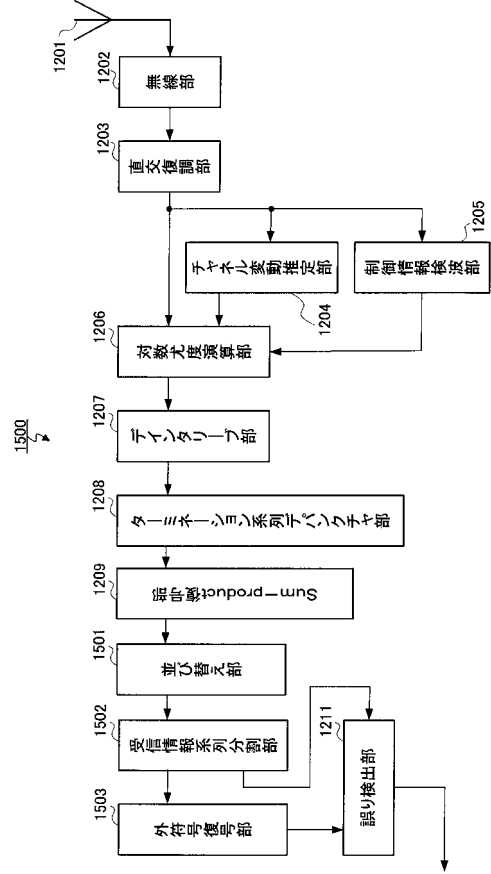
【図16】



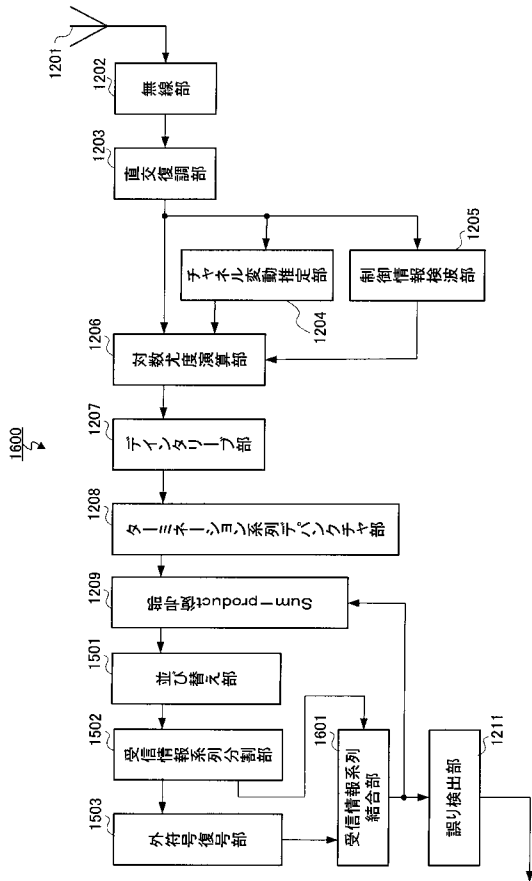
【 図 17 】



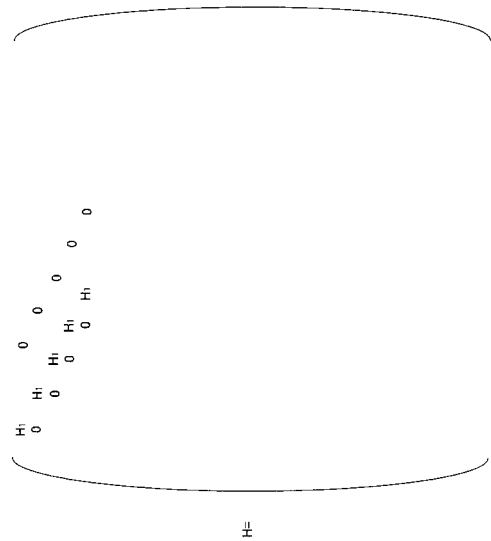
【 図 18 】



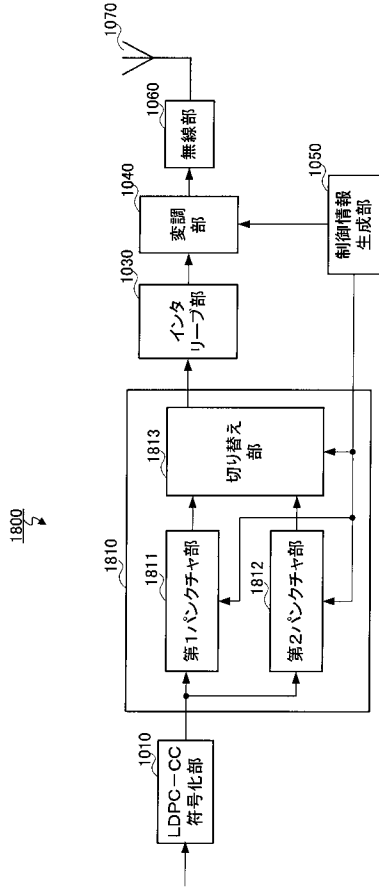
【 図 19 】



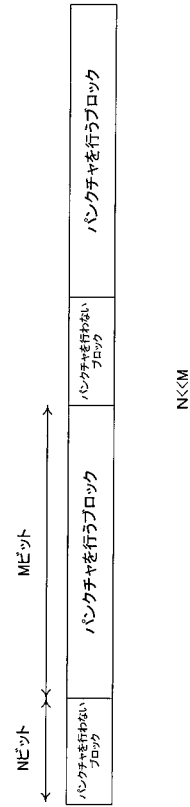
【 図 20 】



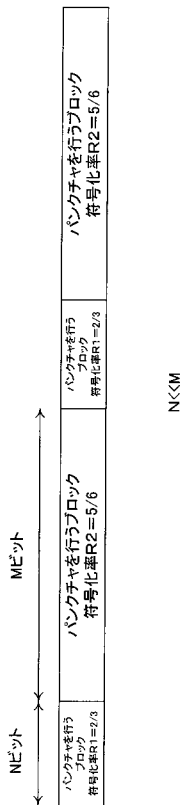
【図25】



【図26】



【図27】



【図28】

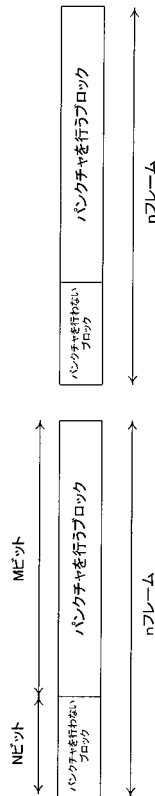


図28A

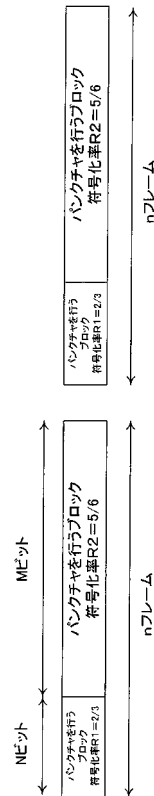
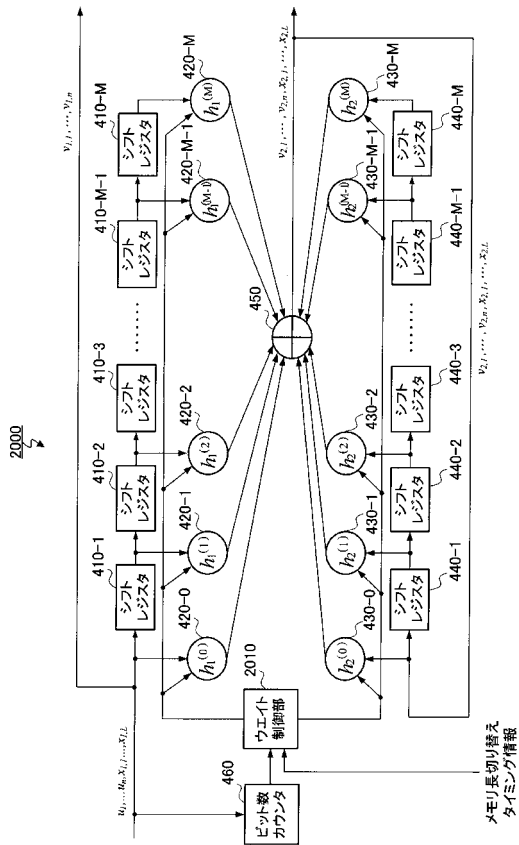
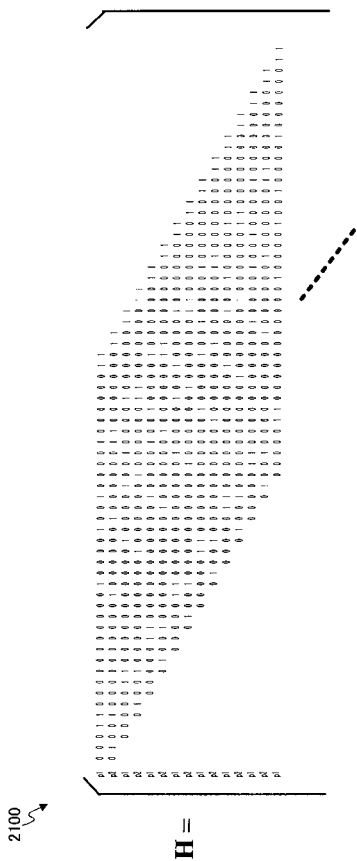


図28B

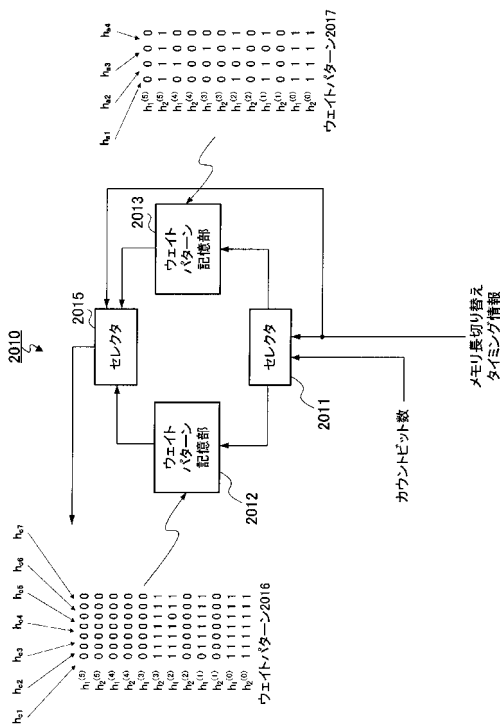
【図 3 3】



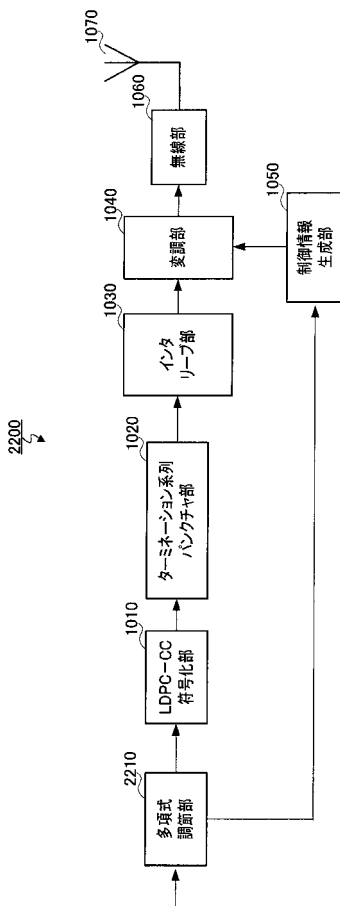
【図 3 5】



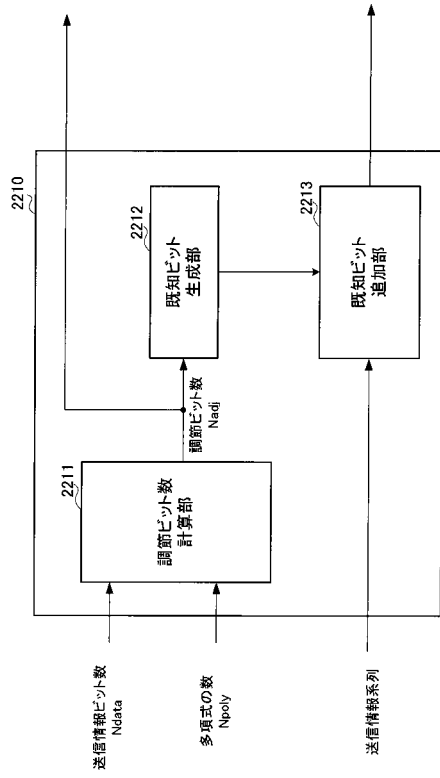
【図 3 4】



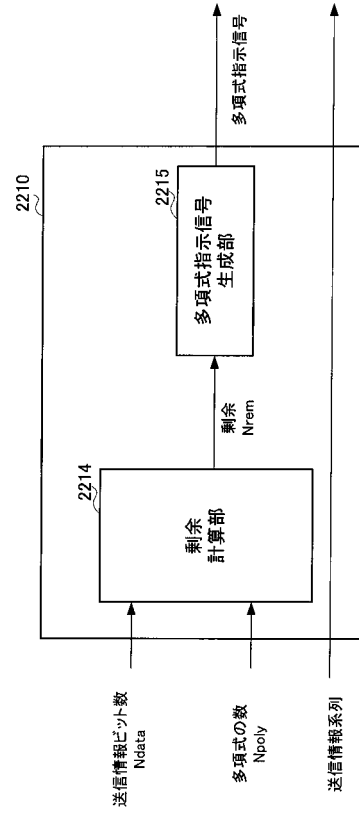
【図 3 6】



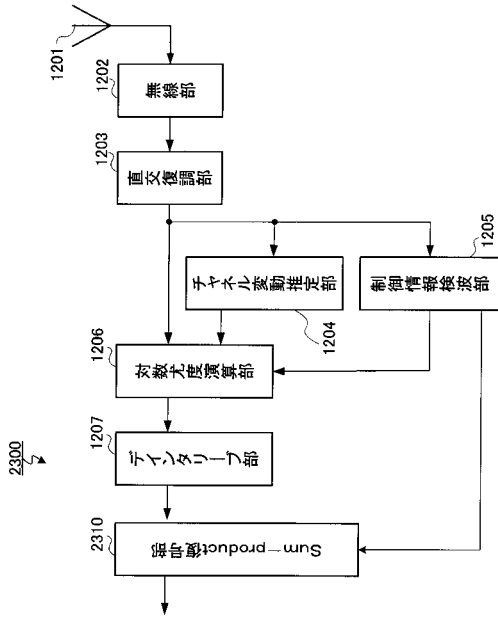
【図 37】



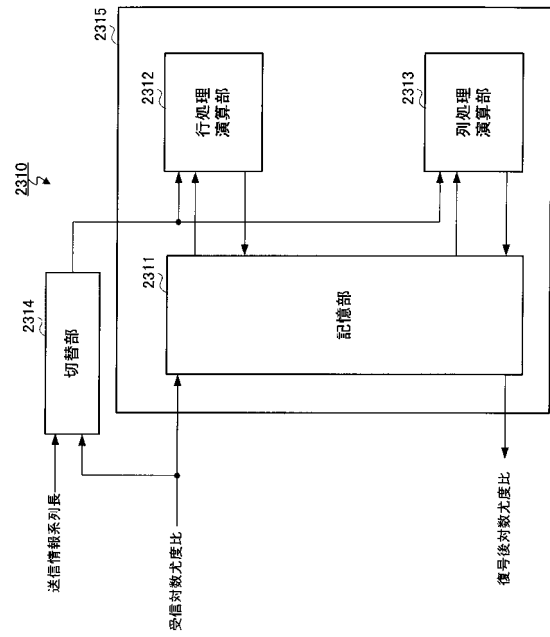
【図 38】



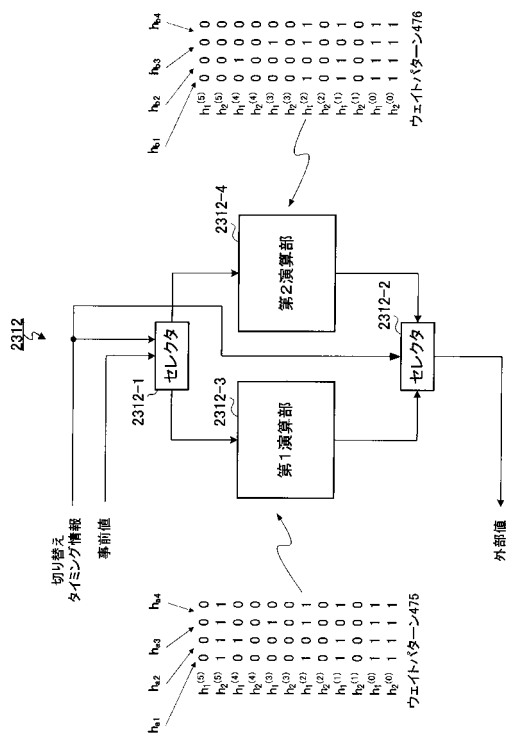
【図 39】



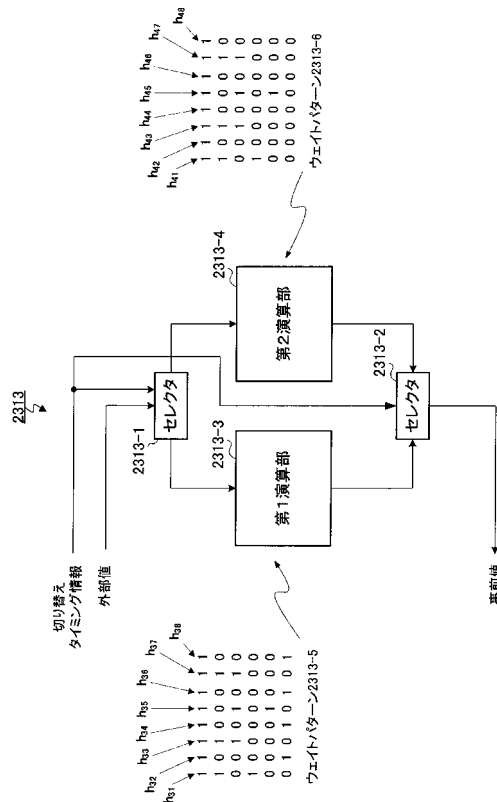
【図 40】



【 図 4 1 】



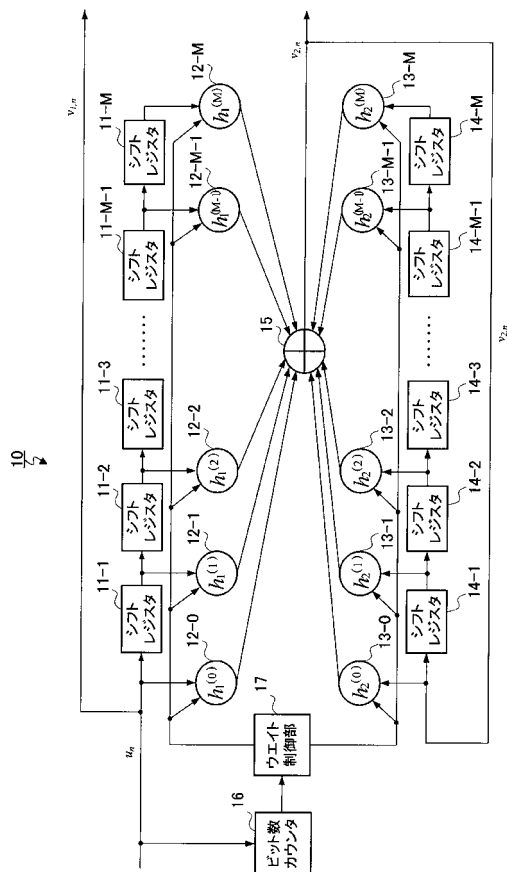
【 図 4 2 】



【 図 4 3 】

$$\mathbf{H}_{[0,n]}^T = \begin{bmatrix}
 h_1^{(0)}(0) & h_1^{(1)}(1) & \dots & h_1^{(M)}(M) & 0 & \dots & 0 \\
 h_2^{(0)}(0) & h_2^{(1)}(1) & \dots & h_2^{(M)}(M) & 0 & \dots & 0 \\
 0 & h_1^{(0)}(1) & \dots & h_1^{(M)}(M+1) & \dots & \dots & \dots \\
 \vdots & h_2^{(0)}(1) & \dots & h_2^{(M)}(M+1) & \dots & \dots & \dots \\
 0 & 0 & \dots & \dots & \dots & \dots & 0 \\
 \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots \\
 h_1^{(M)}(n) & \vdots & \dots & \vdots & \vdots & \vdots & \vdots \\
 h_2^{(M)}(n) & \vdots & \dots & \vdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots \\
 0 & h_2^{(0)}(n) & \dots & \vdots & \vdots & \vdots & \vdots
 \end{bmatrix}$$

【 図 4 4 】



フロントページの続き

審査官 矢頭 尚之

(56)参考文献 国際公開第2007/037124(WO, A1)

Alberto Jimenez Felstrom, Kamil Sh. Zigangirov, Time-varying periodic convolutional codes with low-density parity-check matrix, Information Theory IEEE Transactions, 米国, IEEE, 1999年9月, VOL.45, NO.6, pp.2181-2191

(58)調査した分野(Int.Cl., DB名)

H03M 13/19

H03M 13/23