

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4169738号
(P4169738)

(45) 発行日 平成20年10月22日(2008.10.22)

(24) 登録日 平成20年8月15日(2008.8.15)

(51) Int.Cl.		F I	
H03M 7/40	(2006.01)	H03M 7/40	
H04N 7/26	(2006.01)	H04N 7/13	Z

請求項の数 20 (全 33 頁)

(21) 出願番号	特願2004-523851 (P2004-523851)	(73) 特許権者	505026125
(86) (22) 出願日	平成15年7月16日(2003.7.16)		インリア・インスティテュート・ナショナル・ドゥ・ルシエルチェ・アン・インフォマティック・エ・アン・アートマティック
(65) 公表番号	特表2005-534241 (P2005-534241A)		INRIA INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE
(43) 公表日	平成17年11月10日(2005.11.10)		フランス国、F-78153 ル・シェズニー セデックス、ボアット・ポスタル 105、ロックエンコート、ドメイン・ドゥ・ボルソー
(86) 国際出願番号	PCT/FR2003/002246		
(87) 国際公開番号	W02004/012340		
(87) 国際公開日	平成16年2月5日(2004.2.5)	(74) 代理人	100080001
審査請求日	平成17年11月2日(2005.11.2)		弁理士 筒井 大和
(31) 優先権主張番号	02/09287		
(32) 優先日	平成14年7月22日(2002.7.22)		
(33) 優先権主張国	フランス (FR)		
(31) 優先権主張番号	03/03288		
(32) 優先日	平成15年3月18日(2003.3.18)		
(33) 優先権主張国	フランス (FR)		

最終頁に続く

(54) 【発明の名称】 算術符号をロバスト復号化する装置および方法

(57) 【特許請求の範囲】

【請求項 1】

復号化方法であって、

a. ソース状態図におけるソース状態 (C_k) 間の遷移に関連される x に関する記号 (S_k) の確率を定義するように設計されたソースモデルと、 x に関する記号と第 1 のバイナリ要素との間の対応を定義するように設計されたいわゆる「プロダクト」モデルとを定義し、「プロダクト」状態図のプロダクト状態に関してプロダクト状態がソース状態の関数であるようにし、

b. 第 1 のバイナリ要素の伝送時に、第 2 のバイナリ要素を受信する確率を定義するように設計された「チャンネル」特性を受け取り、

c. 第 2 のバイナリ要素の受信符号化されたフローについて、ソースモデルおよびプロダクトモデルのチャンネル特性から、第 2 のバイナリ要素のフローのビットを知って x に関する記号を復号化する確率、または第 2 のバイナリ要素のフローのビットを知ってプロダクト状態に到達する確率を計算し、

d. ステップ c で計算された確率から x に関する記号の最も起こりそうなフローを再構築することを特徴とする復号化方法。

【請求項 2】

請求項 1 に記載の復号化方法において、

前記プロダクトモデルが、前記ソースモデルに基づくとともに、 x に関する記号に関連される確率に関して x に関する記号と第 1 のバイナリ要素との間での対応を確立するよう

10

20

に設計された転送モデルに基づいて定義されることを特徴とする復号化方法。

【請求項 3】

請求項 2 に記載の復号化方法において、

前記ソースモデルが、ソースの m に関する記号 (A_k) と対象の x に関する記号 (S_k) との間での対応に関連される遷移を含み、特に $m = x$ かつ $x = 2$ を含むことを特徴とする復号化方法。

【請求項 4】

請求項 1 ~ 3 の何れか一項に記載の復号化方法において、

前記プロダクトモデルおよび転送モデルが算術型であり、前記転送モデルが符号化モデルまたは復号化モデルであることを特徴とする復号化方法。

10

【請求項 5】

請求項 1 ~ 4 の何れか一項に記載の復号化方法において、

ステップ c が、第 2 のバイナリ要素のフローのビットを知って、 x に関する記号を復号化する確率の計算を複数のステップにおいて行うことを特徴とする復号化方法。

【請求項 6】

請求項 1 ~ 5 の何れか一項に記載の復号化方法において、

ステップ c が、一定数のステップからステップごとに一定の選択基準に基づいて、第 2 のバイナリ要素のフローのビットを知って 1 つのステップの間に計算された x に関する記号を復号化する確率のうち特定の確率の選択を行うことを特徴とする復号化方法。

20

【請求項 7】

請求項 6 に記載の復号化方法において、

ステップ c で前記選択基準が、最大確率の中で保持される固定数の確率を含むことを特徴とする復号化方法。

【請求項 8】

請求項 6 または 7 に記載の復号化方法において、

ステップ c での前記選択基準が、第 1 のバイナリ要素の送信時に第 2 のバイナリ要素を受信する確率と比較される最小のしきい値を有し、前記最小のしきい値より高い第 1 のバイナリ要素の送信時に第 2 のバイナリ要素を受信する確率から計算された第 2 のバイナリ要素のフローのビットを知って x に関する記号を復号化する確率だけを各ステップで記憶するようにすることを特徴とする復号化方法。

30

【請求項 9】

請求項 5 ~ 8 の何れか一項に記載の復号化方法において、

ステップ c が、遷移によってリンクされた状態から構成されるツリーの構築を行い、各状態は確率の計算におけるステップに対応し、1 つの状態から開始する各遷移は、計算された確率の 1 つに対応することを特徴とする復号化方法。

【請求項 10】

請求項 1 または 9 に記載の復号化方法において、

ステップ d が、前記ツリーの中の連続遷移に対応する確率の積の計算を行うとともに、最も起こりそうな連続遷移と x に関する記号のうち最も起こりそうなフローとに対応する最大積の選択を行うことを特徴とする復号化方法。

40

【請求項 11】

請求項 5 または 10 に記載の復号化方法において、

ステップ c が、復号化される一定数の連続的な x に関する記号について、第 2 のバイナリ要素のフローのビットを知って x に関する記号のシーケンスを復号化する確率の計算を行うことを特徴とする復号化方法。

【請求項 12】

請求項 5 または 10 の復号化方法において、

ステップ c が、第 2 のバイナリ要素のフローにおける一定数の連続的なビットについて、第 2 のバイナリ要素のフローのビットを知って x に関する記号のシーケンスを復号化する確率の計算を行うことを特徴とする復号化方法。

50

【請求項 13】

請求項 1 ~ 4 の何れか一項に記載の復号化方法において、
前記対応を確立するプロダクトモデルに基づいてステップ c が、

c 1 . 初期状態から最終状態へプロダクト状態図の遷移によってリンクされた状態を連続的に構築し、各状態が前記初期状態からの第 2 のバイナリ要素のフローにおける一定数のビットについて予想数の復号化された x に関する記号を定義することを特徴とする復号化方法。

【請求項 14】

請求項 13 に記載の復号化方法において、
ステップ c の構成が、

c 2 . 前記最終状態から前記初期状態へ前記プロダクト状態図からの状態の連続的な抑制を行い、前記プロダクト状態図に対して第 2 のバイナリ要素のフローにおける一定数のビットについて x に関する記号の予想数が x に関する記号の既知の最大数と異なっていることを特徴とする復号化方法。

【請求項 15】

請求項 13 または 14 に記載の復号化方法において、
ステップ c が、

c 1 . 第 2 のバイナリ要素のフロー内の過去のビットを知って、各状態の構築後に一定状態に到達した第 1 の連続的に可能な計算を行い、

c 2 . 前記一定状態を知って、前記最終状態から前記初期状態への状態ごとに、第 2 のバイナリ要素のフローにおいて次のビットを観察する第 2 の連続的な確率の計算を行うことを特徴とする復号化方法。

【請求項 16】

請求項 13 または 14 に記載の復号化方法において、
ステップ c が、

c 3 . 前記初期状態から前記最終状態への図の状態ごとに、第 2 のバイナリ要素のフローにおいて過去のビットを知ってその状態に到達する第 1 の連続的な確率と、前記最終状態から前記初期状態までの状態ごとに、一定状態を知って第 2 のバイナリ要素のフローにおいて次のビットを観察する第 2 の連続的な確率との計算を行うことを特徴とする復号化方法。

【請求項 17】

請求項 13 ~ 16 の何れか一項に記載の復号化方法において、

ステップ c が、状態ごとに第 1 の確率と第 2 の確率の積を決定することにより第 2 のバイナリ要素のフローのビットを知って、前記一定状態に到達する確率の計算を行うことを特徴とする復号化方法。

【請求項 18】

請求項 1 または 17 に記載の復号化方法において、

ステップ d が、最大積の選択に続いて、さまざまな可能な連続的な状態の確率の積の計算を行うことを特徴とする復号化方法。

【請求項 19】

請求項 13 または 18 に記載の復号化方法において、

ステップ d が、プロダクトモデルの対応を使用して、前記連続的な状態に対応する x に関する記号のフローの確立を行うことを特徴とする復号化方法。

【請求項 20】

第 2 のバイナリ要素の符号化されたフローを受け取る第 1 の入力を有するデコーダであって、

第 1 のバイナリ要素の送信時に、第 2 のバイナリ要素を受信する確率を定義するように設計されチャンネル特性を受け取る第 2 の入力と、

第 1 のモジュールと、

処理モジュールとを備え、

前記第1のモジュールが、ソース状態図におけるソース状態間の遷移と関連される x に関する記号について確率を定義するように設計されたソースモデルと、「プロダクト」状態図においてソース状態の関数であるプロダクト状態に関して、 x に関する記号と第1のバイナリ要素との間の対応を定義するように設計されたいわゆる「プロダクト」モデルとを規定し、

前記処理モジュールが、チャンネル特性に基づいてソースモデルおよびプロダクトモデルの第2のバイナリ要素の符号化されたフローのビットを知って x に関する記号を復号化する確率、または第2のバイナリ要素のフローのビットを知ってプロダクト状態に到達する確率を計算するように適応されるとともに、計算された確率から x に関する記号の最も起こりそうなフローを再構築するように適応されたことを特徴とするデコーダ。

10

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、特にマルチメディア信号（オーディオ、音声、画像、ビデオ、音声）用のデジタルデータ圧縮／解凍に関し、および無線通信ネットワークと移動体通信ネットワークなどのノイズが多いネットワーク上でデータのロバスト伝送に関する。

【背景技術】

【0002】

デジタルデータの伝送速度を低減するために、デジタルデータが圧縮され、それと同時に専門家が「信号エントロピー」と呼ぶ理論上の最大値が達成されるようにする。これを行うためには、多くの場合、可変長符号とも呼ばれる統計学的な符号、たとえばハフマン符号が利用される。しかしながら、これらの符号には伝送エラーにきわめて敏感であるという欠点がある。ビットの逆転はデコーダの脱同期化につながり、その結果誤りのあるビット位置の後のすべてのデータの復号化が間違っただけとなることがある。

20

【発明の開示】

【発明が解決しようとする課題】

【0003】

ネットワーク上でのマルチメディア信号の圧縮、伝送、および解凍のための既存の解決策は、さらに言及されるが、データトランスポートサービスの一定品質が保証されているという仮説に基づいている。言い換えると、訂正符号の使用に依存することにより、トランスポート層とリンク層は（つまり、圧縮および解凍アプリケーションにより見られるように）擬似ヌルの残存誤り率を達成することを可能にしている。しかしながら、擬似ヌルの残存誤り率というこの仮定は、特に無線ネットワークと移動体ネットワークにおいておよびチャンネル符号の現実的な複雑さについて経時的にチャンネル特性が変わる（非静止チャンネル）時には、もはや当てはまらなくなる。さらに訂正符号による冗長性の付加は効果的な速度の低減につながる。

30

【0004】

したがって、新しい符号化／復号化の解決策に対するニーズがある。

【0005】

本発明はこの分野においてある程度の進展を提案するものである。

40

【課題を解決するための手段】

【0006】

本発明は、以下のステップを有する復号化方法に関するものである：

a．ソース状態図におけるソース状態間の遷移に関連される x に関する記号（ x 進法記号）の確率を定義するように設計されたソースモデルと、 x に関する記号と第1のバイナリ要素（2進法要素）との間の対応を定義するように設計されたいわゆる「プロダクト」モデルとを定義し、「プロダクト」状態図のプロダクト状態に関してプロダクト状態がソース状態の関数であるようにし、

b．第1のバイナリ要素の伝送時に、第2のバイナリ要素を受信する確率を定義するように設計された「チャンネル」特性を受け取り、

50

c. 第2のバイナリ要素の受信符号化されたフローについて、ソースモデルおよびプロダクトモデルのチャンネル特性から、第2のバイナリ要素のフローのビットを知ってxに関する記号を復号化する確率、および、第2のバイナリ要素のフローのビットを知ってプロダクト状態に到達する確率をそれぞれ計算し、

d. ステップcで計算された確率からxに関する記号の最も起こりそうなフローを再構築する。

【0007】

有利なことには、プロダクトモデルが、ソースモデルに基づくとともに、xに関する記号に関連される確率に関してxに関する記号と第1のバイナリ要素との間での対応を確立するように設計された転送モデルに基づいて定義される。ソースモデルが、ソースのmに関する記号(m進法記号)と対象のxに関する記号との間での対応に関連される遷移を含み、特にm $\times$ かつ $x = 2$ を含む。

【0008】

好適には、プロダクトモデルおよび転送モデルが算術型であり、転送モデルが符号化モデルまたは復号化モデルである。

【0009】

さらに好適には、ステップcが、第2のバイナリ要素のフローのビットを知って、xに関する記号を復号化する確率の計算を段階的に行う。

【0010】

本発明の第1の態様では、ステップcが、一定数のステップからステップごとに一定の選択基準に基づいて、第2のバイナリ要素のフローのビットを知って1つのステップの間に計算されたxに関する記号を復号化する確率のうち特定の確率の選択を行う。

【0011】

有利なことには、ステップcで選択基準が、最大確率の中で保持される固定数の確率を含んでいる。有利または付加的には、ステップcでの選択基準が、第1のバイナリ要素の送信時に第2のバイナリ要素を受信する確率と比較される最小のしきい値を有し、最小のしきい値より高い第1のバイナリ要素の送信時に第2のバイナリ要素を受信する確率から計算された第2のバイナリ要素のフローのビットを知ってxに関する記号を復号化する確率だけを各ステップで記憶するようにしている。

【0012】

好適には、ステップcが、遷移によってリンクされた状態から構成されるツリーの構築を行い、各状態は確率の計算におけるステップに対応し、1つの状態から開始する各遷移は、計算された確率の1つに対応する。このように、第2のバイナリ要素のフローのビットを知って、xに関する記号を復号化する確率のうち特定の確率を選択することが、その特定の確率の関数としてのツリーにおいて遷移の部分集合に対応する状態を構築することに対応している。

【0013】

加えて、ステップdが、ツリーの中の連続的な遷移に対応する確率の積の計算を行うとともに、最も起こりそうな連続的な遷移とxに関する記号のうち最も起こりそうなフローとに対応する最大積の選択を行う。

【0014】

第1の動作では、ステップcが、復号化される一定数の連続的なxに関する記号について、第2のバイナリ要素のフローのビットを知ってxに関する記号のシーケンスを復号化する確率の計算を行う。

【0015】

第2の動作では、ステップcが、第2のバイナリ要素のフローにおける一定数の連続的なビットについて、第2のバイナリ要素のフローのビットを知ってxに関する記号のシーケンスを復号化する確率の計算を行う

本発明の第2の態様では、対応を確立したプロダクトモデルに基づいてステップcが、
c1. 初期状態から最終状態へプロダクト状態図の遷移によってリンクされた状態を連

10

20

30

40

50

続的に構築し、各状態が前記初期状態からの第2のバイナリ要素のフローにおける一定数のビットについて予想数の復号化されたxに関する記号を定義する。

【0016】

有利なことには、ステップcの構成が、

c2. 前記最終状態から前記初期状態へ前記プロダクト状態図からの状態の連続的な抑制を行い、前記プロダクト状態図に対して第2のバイナリ要素のフローにおける一定数のビットについてxに関する記号の予想数がxに関する記号の既知の最大数と異なっている。

【0017】

第1の動作では、ステップcが、

c1. 第2のバイナリ要素のフロー内の過去のビットを知って、各状態の構築後に一定状態に到達した第1の連続的に可能な計算を行い、

c2. 最終状態から初期状態への状態ごとに、第2のバイナリ要素のフローにおいて次のビットを知って一定状態に到達する第2の連続的な確率の計算を行う。

【0018】

第2の動作では、ステップcが、

c3. 初期状態から最終状態への図の状態ごとに、第2のバイナリ要素のフローにおいて過去のビットを知ってその状態に到達する第1の連続的な確率と、最終状態から初期状態までの状態ごとに、第2のバイナリ要素のフローにおいて次のビットを知って一定状態に到達する第2の連続的な確率との計算を行う。

【0019】

好適には、ステップcが、状態ごとに第1の確率と第2の確率との積を決定することにより各状態の確率の計算を行い、そして、ステップdが、最大積の選択に続いて、さまざまな可能な連続的な状態の確率の積の計算を行う。

【0020】

有利なことには、ステップdが、プロダクトモデルの対応を使用して、連続的な状態に対応するxに関する記号のフローの確立を行う。

【0021】

また、本発明は、第2のバイナリ要素の符号化されたフローを受け取る第1の入力を有するデコーダに関する。そのデコーダは、第1のバイナリ要素の送信時に、第2のバイナリ要素を受信する確率を定義するように設計されチャンネル特性を受け取る第2の入力と、

第1のモジュールを備え、

第1のモジュールが、ソース状態図におけるソース状態間の遷移と関連されるxに関する記号について確率を定義するように設計されたソースモデルと、

「プロダクト」状態図においてソース状態の関数であるプロダクト状態に関して、xに関する記号と第1のバイナリ要素との間の対応を定義するように設計されたいわゆる「プロダクト」モデルと、

チャンネル特性に基づいてソースモデルおよびプロダクトモデルの第2のバイナリ要素の符号化されたフローのビットを知ってxに関する記号を復号化する確率、または第2のバイナリ要素のフローのビットを知ってプロダクト状態に到達する確率を計算するように適応されるとともに、計算された確率からxに関する記号の最も起こりそうなフローを再構築するように適応された処理モジュールとを規定する。

【0022】

本発明の他の特徴および利点は、添付図面とともに以下の詳細な説明を検討する際に明らかになる。

【発明を実施するための最良の形態】

【0023】

本明細書の記載に対しての図面および付属物には一定の文字の要素が本質的に含まれている。したがって、それらの要素は、明細書の記述の理解を助ける役割を果たすだけでは

10

20

30

40

50

なく、規定どおりに本発明の定義に貢献するものである。

【0024】

一般的には、マルチメディア信号（画像、ビデオ、オーディオ、音声）用の圧縮システムは、可変長符号とも呼ばれる統計符号を使用している。これらにより、専門化が「信号エントロピー」と呼ぶものに近い速度を得ることが可能になる。既存のシステムにおいて（特に規格において）最も広範囲に使用されている符号は、以下の論文、つまり、「最小冗長性符号の構築方法（A method for the construction of minimum redundancy codes）」、D.A.ハフマン、会報、IRE、40（1951年）、1098～1101ページの中で説明されたハフマン符号である。

10

【0025】

さらに最近では、圧縮に関して性能向上のために算術符号への関心が再び始まっている。実際には、これらの符号により、仮定されるソースモデルから符号化方法を切り離すことが可能になる。そのために、さらに高位の統計モデルの使用が容易になる。これらの算術符号は以下のような研究論文に記述され、すなわち、

[1]「一般化されたクラフト不等式および算術（Generalized kraft inequality and arithmetic）」、J.J.Rissanen、IBM J. Res. Develop.、20：198～203、1976年5月、

[2]「数表現としての算術符号化（Arithmetic coding as number representations）」、J.J.Rissanen、Acta Polytech. Scand. Math.、31：44～51、1979年12月、
および以下のような米国特許に記述されている：

20

米国特許第4,286,256号、第4,467,317号、第4,652,856号。

【0026】

最近まで、圧縮システムの設計は、トランスポートサービスの保証された品質を仮定して引き受けられてきている。実際に仮定されていることは、OSIモデルの下層が、本出願から理解されるように、擬似ヌルの残存誤り率を保証するエラー訂正符号を組み込んでいることである。

【0027】

したがって、可変長符号は、それらの伝送ノイズに対してかなりの感度にも関わらず、幅広く使用可能である。このような擬似ヌルの残存誤り率の仮定は、チャンネル特性が経時的に変化する（非静止チャンネル）無線ネットワークおよび移動体ネットワークではもはや当てはまらない。ソース信号デコーダにより見られるこの残存誤り率は多くの場合決して無視できない。バイナリストリームにおけるエラーによって、デコーダの脱同期化、それ故に復号化された情報ストリームへのエラーの伝播が引き起こされ得る。

30

【0028】

この伝播の問題を緩和するために、第1世代規格（H.261、H.263、MPEG-1、MPEG-2）は、送信されたバイナリストリームの構文中に同期マーカを組み込んでいる。これらは、他の符号語内で発生するエラーによってエミュレートされず、それ故に、「1」に近い確率でデコーダにより認識され得る長い符号語（「0」が後に続く「1」に設定された15ビットまたは21ビットの文字列から構成される16ビットまたは22ビット）である。

40

【0029】

これにより、この同期マーカによって区切られるパケット内にバイナリストリームが構築されるようになる。そのために、エラーの伝播がパケット内で限定されるようになる。

【0030】

しかしながら、エラーがパケットの開始時に発生した場合、パケットの残りが失われてしまう可能性がある。さらに、これらの同期マーカの頻度は、圧縮効率の不当な損失を回避するために制限されなければならない。

【0031】

50

そこで、新しい規格（H. 263+とMPEG-4）は可逆可変長符号（RVLC）を採用している。これらの符号の特別な特徴は、パケット内の最初のビットから最後のビットまで復号化可能であり、およびパケット内の最後のビットから最初のビットまで逆に復号化可能である。

【0032】

エラーがパケットの真中で発生した場合、この符号の対称性により、同期マーカによって区切られたパケットの最後まで符号を伝播する代わりにパケットの真中のセグメントにエラー伝播を限定することが可能になる。しかしながら、符号対称性は、ハフマン符号と比較されると、10%のオーダーで圧縮効率の損失を生じさせる結果となる。加えて、可逆可変長符号はエラー伝播の問題を完全に克服していない。つまり、エラーがパケットの開始時および終了時に発生すると、パケット全体が誤りとなる傾向がある。

10

【0033】

伝送ノイズに関連して圧縮ターム（つまり、ソースのエントロピーに接近可能な圧縮ターム）での極めて効率的な符号のファミリ（集合）に基づいて、符号化方法/復号化方法ロバストの設計は、特に将来のマルチメディア（画像、ビデオ、オーディオ、音声）の移動体通信システムにとって重要な目標である。新しい規格が、ITU（国際電気通信連合）およびISO（国際標準化機構）の双方の中でそのようなシステムのために開発中である。

【0034】

規格は電気通信業界部門において顕著な役割を果たしているけれども、そのような符号のファミリによっても独自仕様の解決策を求めるすき間産業において用途が見出され得る。

20

【0035】

本明細書中で記述されている発明は、特に、算術符号および擬似算術符号を含む算術型符号などのような圧縮システムに幅広く使用されている符号のファミリ用のエンコーダ/デコーダに関する。

【0036】

算術符号化および擬似算術符号化の原則は後述される。それらは、文献[2]と以下の論文との中でさらに詳細に説明される：

[3]「データ圧縮のための算術コード化（Arithmetic coding for data compression）」、I. H. Witten、R. M. Neal、およびJ. G. Cleary、ACMの通信（Communications of the ACM）、30（6）：520～540、1987年6月、

30

[4]「画像およびテキスト圧縮（Image and Text Compression）」、P. G. HowardおよびJ. S. Vitter、85～112ページ、クルワ・アカデミック・パブリッシャ（Kluwer Academic Publisher）、1992年。

【0037】

本明細書の記述の残りにおいては、用語「シーケンス」の使用は要素のフロー、つまり一連の要素を指す。たとえば、記号のそれぞれのビットのシーケンスは、記号のそれぞれのビットのフロー、つまり記号のそれぞれの一連のビットを示すものである。

40

【0038】

一般的には、符号化の原則には次のステップが含まれる：

$A = A_1 \dots A_I$ を定量化されたソース記号のシーケンスとし、定量化されたソース記号が、 $M = 2^q$ 記号から構成される有限アルファベット A すなわち $A = (A_1 \dots A_M)$ から当該ソース記号の値を取るものとする。このアルファベットは、記号ごとに関連付けられた定常確率を示す定常分布を有する。（ソース A とも呼ばれる） M に関する記号（ M 進法記号）のシーケンスは、（ x M を使用して） x に関する記号のシーケンスに変換される。 M に関する記号のシーケンスが（ソース S とも呼ばれる）バイナリ記号のシーケンス $S = S_1 \dots S_K$ に変換される場合には、 K は $K = q \times L$ によって定義される。

50

変数 M および x は 2 から始まる整数を定義する。したがって、この例の中では $x = 2$ および $M = 4$ である。

【 0 0 3 9 】

ほんの一例として本明細書中の記述の残りにおいては簡略のために、 $x = 2$ であると仮定する。この変換は、 M に関するソース A とバイナリソース S との間の対応を含めて、後述されるようなソースモデルを使用する。バイナリ記号 S のシーケンスは、精度 T の擬似算術エンコーダを介して第 1 のバイナリ情報要素のシーケンス $U = U_1 \dots U_N$ に順々に変換される。このバイナリストリートの長さ N は、 S のランダムな変数関数、すなわち、 A の関数自身であるバイナリ記号のシーケンスと M に関する記号のシーケンスとの関数である。第 1 のバイナリ要素 U のストリームは、ほんの一例として、メモリレス (memoryless) チャンネルで送信される。第 1 のバイナリ要素 U のこのストリームは、通信チャンネルによる伝送中に生じるエラーのために、第 2 のバイナリ要素 Y のストリームとして受け取られる。通信システムの下層によって、実際には、少なくとも通信チャンネルの現在の特性 I_c が決定可能になり、たとえばチャンネルにより通信されたビット中に生じた誤り率と受信されたビットの信頼測定値とが決定可能になる。このような後者の測定値により、チャンネル上で通信された第 1 のバイナリ要素を知る確率と第 2 のバイナリ要素を受け取る確率とが推定可能になる。これらの確率は、チャンネルの現在の特性 I_c の 1 つ、すなわちチャンネル上でのエラーの確率 p に基づいて付属物 (指数 i が整数である式 21) の中で示されるように推定される。

【 0 0 4 0 】

復号化の原則は記号 A のシーケンスの推定を伴い、以下のように「観察 y 」のシーケンスとも呼ばれる第 2 のバイナリ要素を知るものである。本明細書の記述の残りでは、大文字はランダム変数を示すために使用され、小文字は表現、つまりこれらの変数の観察を示すために使用される。一連の連続変数を示すために、表記 $X_u^v = (X_u, X_{u+1}, \dots, X_v)$ が使用される。

【 0 0 4 1 】

符号化 / 復号化の原則は、 $M > 2$ で M に関する記号 (A) に対応せずに $M = 2$ でバイナリ記号 (S) に対応するソースに直接的に適用することもできる。本明細書の記述の残りでは、それぞれ x に関する記号、 M に関する記号は、特に、バイナリ記号 (それぞれ $x = 2$ 、 $M = 2$) の場合とそれぞれ $x > 2$ 、 $M > 2$ の記号の場合とを含む。

【 0 0 4 2 】

図 1 は、組合されたエンコーダ 1 と、組合されたデコーダ 2 との例を示している。エンコーダおよびデコーダはそれぞれ入力 E_c 、 E_d と出力 W_c 、 W_d とを有する。符号化の間に、符号化される x に関する記号のフローは E_c においてエンコーダ入力で受け取られる。エンコーダは、たとえば圧縮形式で、 x に関する記号のフローを符号化することにより生成され第 1 のビットストリーム U_c と呼ばれる第 1 のバイナリ要素のフローを W_c で出力する。この第 1 のビットストリーム U_c は、デコーダ 2 の入力 E_d にチャンネル 11 を介して送信される。(たとえば第 1 のフローの中の特定のビットを反転させることによって) 第 1 のビットストリームの中にエラーを生じさせる可能性のあるチャンネルは、デコーダ 2 の入力に、第 2 のビットストリーム Y_d と呼ばれる第 2 のバイナリ要素のフローを送信する。本発明によれば、処理モジュール 6 は、デコーダの出力 W_d で記号 S_d のフローを得るために、チャンネルによって通信されるチャンネル特性 I_c とともに第 2 のビットストリームを受信し処理する。処理モジュール 6 は、モデルモジュール 12 と動作するように設計されている。このモデルモジュールは、

後述されるソース状態図の中のソース状態間での遷移に関連付けられる x に関する記号の確率を規定するように設計されたソースモデル 8 と、

ソース状態と転送モデル状態との関数である「プロダクト」状態の状態図に基づいて、 x に関する記号と第 1 のバイナリ要素との間の対応を規定するために使用されるいわゆる「プロダクト (積)」モデル 7 とを備えている。

【 0 0 4 3 】

プロダクトモデル 7 は、ソースモデル 8 と転送モデル 9 または 10 とに基づいて規定されている。プロダクトモデルはソースモデルと転送モデル 9 または 10 とから動的に構築できる。別の考えられる実施の形態では、プロダクトモデルは、動的に構築されずに、モデルモジュール内で実現できる。転送モデルは符号化モデル 9 または復号化モデル 10 を含む。この転送モデルは、バイナリ記号に関連付けられる確率の関数として、バイナリ記号と第 1 のバイナリ要素との間の対応を確立する働きをする。

【 0 0 4 4 】

ソースモデルを使用して圧縮される前に、x に関する記号のシーケンスに事前変換できる M に関する記号のシーケンスから開始することも可能である。

【 0 0 4 5 】

考えられる（可能な）実施の形態では、これらのモデルは、事前計算されて（p r e - c o m p u t e d）デコーダのモデルモジュールの外部のモジュール内のメモリに記憶可能である。また、モデルは、たとえばイーサネット型ローカルネットワークまたはインターネット型広域ネットワークを使用して遠隔的に使用することもできる。

【 0 0 4 6 】

処理モジュール 6 は、チャンネル、ソースモデル、およびプロダクトモデルの特性 I c に基づいて、第 2 のバイナリ要素のフローのビットを知る x に関する記号の確率を計算するように適応されている。処理モジュール 6 は、このように計算された確率から x に関する記号の最も起こりそうなフローを再構築するように、および、M に関する記号のフローが符号化された場合にソースモデルから M に関する記号のフローを再構築するように適応されている。

【 0 0 4 7 】

本発明による装置から導出される方法は、本明細書の記述の残りで説明され、特にそのような装置の動作のより良い理解を容易にする。

【 0 0 4 8 】

図 2 は、同期マーカの挿入を使用できる符号化方法の一例を示す。このプロセスは、ソース（M に関する）をその入力で受け取るエンコーダ、たとえば、データのシーケンスを受け取るエンコーダで実現される。

【 0 0 4 9 】

ステップ 100 で、データのシーケンスが M に関するマーカを必要としないとき、プロセスはステップ 104 で継続する。他方、M に関するマーカがデータのシーケンスに挿入される必要がない場合には、これらのマーカの周波数、サイズ、および値がエンコーダにより供給されてステップ 102 でそのマーカの挿入が容易にされる。

【 0 0 5 0 】

ステップ 104 では、（同期マーカを含んでも良いしまたは含まなくても良い）データのシーケンスが M に関する変換（M 進法変換）/ バイナリ変換（2 進法変換）を必要とするかどうかのチェックが行われる。これが当てはまらない場合には、それ故にデータのシーケンスはバイナリであり、ステップ 112 で算術符号化（図の例では擬似算術）を受ける（経る）。それ以外の場合、そのシーケンスの M に関する変換 / バイナリ変換はステップ 106 で発生する。ステップ 108 では、バイナリマーカが挿入されなければならない場合には、これらのマーカの周波数、サイズ、および値がエンコーダにより供給されてステップ 110 でこれらのマーカの挿入が容易にされる。

【 0 0 5 1 】

ステップ 112 では、M に関するデータまたはバイナリデータのシーケンスは擬似算術符号化を受け、その擬似算術符号化の複雑さはパラメータ T により制御される。T が大きい場合には、擬似算術符号化は算術符号化と同一になる。これにより、ビットのストリームへのバイナリデータのシーケンスの圧縮が最大になる。

【 0 0 5 2 】

図 3 は、挿入された同期マーカを考慮に入れた復号化方法の一例を示す。この方法は、その入力で、同期マーカの周波数、サイズ、および値とともに、M に関するソースまたは

10

20

30

40

50

xに関するソース（特にバイナリ）に関連するバイナリストリームおよび情報を受け取るデコーダで実現される。

【0053】

ステップ120では、以下のような復号化のさまざまなモード、すなわち

ステップ124での瞬間復号化と、

本明細書の記述の残りで詳述されるようにステップ122で本発明による連続復号化と

、
本明細書の記述の残りで詳述されるようにステップ126で本発明による最適復号化とが選択可能である。

【0054】

ステップ128では、復号化の後に得られたデータのシーケンスがバイナリ変換/Mに関する変換を必要とするかどうかを判断するためにチェックが行われる。

【0055】

そのようなことが当てはまる場合には、それ故にデータのシーケンスはバイナリであり、データのシーケンスがバイナリマーカを含むかどうかを判断するためのチェックがステップ130で行われる。そのようなことが当てはまる場合には、これらのマーカは、その周波数とサイズに関する情報を使用してステップ132で削除される。ステップ130またはステップ132の後に、データのシーケンスはステップ134でバイナリ変換/Mに関する変換を受ける。

【0056】

ステップ128またはステップ134の後、データのシーケンスがMに関するマーカを含むかどうかのチェックがステップ136で行われる。これが当てはまらない場合には、データシーケンス（またはソース）の復号化が完了される。それ以外の場合、マーカはステップ138で削除されてデータシーケンス（またはソース）が得られる。

【0057】

図4～図5は、データシーケンス（またはソース）のMに関する変換/バイナリ変換と、バイナリ変換/Mに関する変換のためのソースモデルを示している。

【0058】

図4Aは、4に関するソース、つまり、4に関するアルファベットからその値を取るソースの4個の記号のバイナリ表現を示す。このソースモデルは、 $q = 2$ および $M = 4$ について図4Aに図示されるように、奥行qのバイナリ系図である場合がある。変形例では、この系図（ツリー）は対応表によって表すことができる。黒のノード（節点）は系図の根と呼ばれ、白のノードは中間ノードであり、灰色のノードは系図の葉と呼ばれる。それぞれのMに関する記号の確率Pはソースの定常確率に対応する。このバイナリ系図の枝上の遷移（不図示）の確率は定常確率から計算される。

【0059】

図4Bは、独立したMに関する記号のシーケンスをバイナリ記号のシーケンスに変換することを可能にするオートマトンを示す。また、このソースモデルは3つの状態のオートマトンの形式で図4Bにおいて表現可能である。つまり、 $C = 0$ は初期状態であり、 $C = 1$ と $C = 2$ は中間状態である。変形例として、このソースモデルは同じ状態のトレリスの形式で図4Cにおいて表現可能である。つまり、左側で $C = 0$ 、右側で初期状態が中間状態 $C = 1$ と $C = 2$ に連結され、左側の中間状態は右側の状態 $C = 0$ に連結されている。また、この例では、それぞれのxに関する記号（このケースではバイナリ記号）の確率が、対応する遷移で示されている。

【0060】

幾つかの独立ではないMに関する記号を含むデータフローに対して、ソースの完全なモデルが、図5Aに図示されるような連続ローカルモデルを連結することにより得られる。ローカルモデルは図4Aに示されているような記号を含むフローのソースモデルに対応する。図5を参照して、2つの独立したMに関する記号のフローに対して、ソースの完全なモデルは、次に来る系図の根を使って、すなわち図5Aおよび図5Bの黒のノードにより

10

20

30

40

50

表現される根を使って第 1 のバイナリ系図の葉の特定を行っている。このように、図 5 A はバイナリ系図を互いに組合せている。変形例として、図 5 B は、第 1 のバイナリ系図を、その初期状態およびその最終状態として取る対応オートマトンに組合せている。このオートマトンは、依存する M に関する記号のシーケンスをバイナリ記号のシーケンスに変換する働きをする。

【 0 0 6 1 】

図 5 A に従って、状態 C_k は、オートマトンが k 個のバイナリ記号の生成の後になることがあると考えられるノードを規定する。考えられる（可能な）ノードの集合は、同じ状態 k のノードと連結する一点鎖線によって図中で記号により表される。シーケンス $C_0, \dots, C_k$ はマルコフ連鎖であり、バイナリ記号の対応フローはこの連鎖の遷移の関数である。そのような遷移は、バイナリ記号 S_k の伝送を引き起こすという点で S_k と呼ばれ、ソースモデルの状態 C_{k-1} の集合と状態 C_k の集合との間で考えられる（可能な）遷移の集合を表現する。

【 0 0 6 2 】

このような確率オートマトンから、バイナリ記号のフローは、隠れたマルコフモデルによってモデリングできる。

【 0 0 6 3 】

それぞれのバイナリ記号と関連付けられた各々の定常確率を知っているので、つまり単一間隔 $[0, 1[$ のセグメントの 1 つに対応する確率のそれぞれを知っているので、算術符号化によってバイナリ記号のシーケンスがこのシーケンスの間隔を決定できるようになる。後者は単一間隔のセグメントを表し、その下限は、下位サブ間隔に対応する記号シーケンスの確率の合計から引き出される。この間隔の幅は、その間隔に対応する記号シーケンスの確率により与えられる。用語「現在の間隔」は、間隔の計算の間に、すなわち状態 C_k に対応するサブシーケンスと関連付けられたサブセグメントの計算の間にシーケンスの一部と、それ故にシーケンスのその一部における現在の状態 C_k とに対応する間隔を言う。

【 0 0 6 4 】

算術符号化を実際的に実行することは、文献 [1] すなわち以下の論文、
[5] 「高速データ圧縮のためのソースコード化アルゴリズム (Source coding algorithms for fast data compression)。カリフォルニア、スタンフォード (Stanford Calif.)」、R F . P a s c o、カリフォルニア州スタンフォード、スタンフォード大学電気工学部博士号学位論文、1976 年、
で最初に紹介され、文献 [2] で再び取り上げられている。そのような実行で発生する問題は、非常に小さい実数で間隔を表現するために、高い数値精度に対するニーズである。その難題は、文献 [3] 中で説明されている間隔 $[0, 1[$ 内の実数のバイナリ表現に依存することにより解決できる。間隔 $[0, 0.5[$ に属する任意の数ではその第 1 のビットが 0 に等しいのに対し、間隔 $[0.5, 1[$ に属する任意の数ではその第 1 のビットが 1 に等しい。その結果、符号化プロセスの間、現在の間隔が $[0, 0.5[$ または $[0.5, 1[$ に完全に含まれている時に、対応するビットが送信され間隔のサイズが二倍になる。特定の処理が、 $1/2$ にまたがる間隔のため必要とされる。現在の間隔が $1/2$ にまたがり $[0.25, 0.75[$ に完全に含まれる場合には、その間隔は単一ビットで特定できない。そのサイズはビットを送信せずに依然として 2 倍にされ、ビットの送信前にそのような演算の実行される回数 (n) が記憶される。 n は、ビットの最後の伝送以降に実行されたりスケーリングの数である。ビット $U_i = u_i$ が送信される間隔に到達した時に、このビットの後には反対の値のビットを n 倍したシーケンスが続く。

【 0 0 6 5 】

この技法を使用することにより、現在の間隔が条件「low < 0.25 < 0.5 high」または「low < 0.5 < 0.75 high」を満たすことが確実にされる。ここに、「low (ロウ)」および「high (ハイ)」はそれぞれ現在の間隔の「下限」および「上限」である

10

20

30

40

50

。

【 0 0 6 6 】

算術符号化の原則は、状態 E が 3 個の変数、つまり low 、 up (アップ)、およびスケーリング数 $nsc1$ で定義される確率オートマトンによりモデリングできる。この符号化の欠点は、ソースの定常分布が先験的に既知ではない場合、またはフローが多数の x に関する記号を有する場合に、非常に多数の状態が存在できることである。精度低減 (reduced-precision) 算術符号化とも呼ばれる擬似算術符号化は、文献 [4] に説明されるように、圧縮性能を大幅に低下させずに考えられる (可能な) 状態の数を削減できる。擬似算術符号化を実行するためには、実際の間隔 $[0, 1 [$ が整数間隔 $[0, T [$ に置き換えられる。整数 T は、複雑さと圧縮効率との間のトレードオフを制御する働きをする。つまり、 T が大きい場合には、間隔再分割がソースの分布に密接に続く。それに対して、 T が小さい (1 に近い) 場合、すべての可能な間隔再分割は事前に計算でき、これらの再分割はソースの分布の近似を表現するにすぎない。

10

【 0 0 6 7 】

表 1 に示されるように、状態間の遷移とビットの対応放出とのすべては事前に計算できる。それから、算術演算は表索引に置き換えられる。

【 0 0 6 8 】

バイナリ記号のシーケンス S_1^K は、バイナリ系図によりビットストリーム U_1^N に変換される。この系図は、バイナリストリームの分布をモデリングするオートマトンまたはエンコーダモデルとして見なされ得る。記号の符号化は、ビットの放出が関連付けられる系図の頂点の選択枝を決定する。各ノードは算術エンコーダの状態 E に対応する。これらの状態間の連続遷移は、オーダー 1 のマルコフソース用のソース $P(S_k | S_{k-1})$ の分布に従う。 E_k をそれぞれの「記号インスタント」 k でのオートマトン状態であるとする。算術符号化の場合と同様に、擬似算術エンコーダの状態 E_k は 3 つの変数すなわち $low S_k$ 、 $up S_k$ 、および $nsc1_k$ によって規定される。用語「 $low S_k$ 」(ロウ S_k) および「 $up S_k$ 」(アップ S_k) は、シーケンス S_1^K の符号化に対応する $[0, T [$ の連続再分割の結果生じる間隔の限界を示す。ビットが放出されるたびに量 $nsc1_k$ はゼロにリセットされ、ビットを放出しないでリスケーリングが行われるたびにその量は増分される。したがって、その量は、最後に放出されたビット以降に実行されたスケーリングの数を示す。ビットが放出されると、その後には反対の値の $nsc1_k$ ビットが続く。

20

30

【 0 0 6 9 】

間隔 $[0, T [$ の考えられる再分割が有限数あるため、ソースの知識がなくても擬似算術エンコーダの状態のすべてを事前に計算することが可能でなければならない。

【 0 0 7 0 】

一例として、表 1 は、バイナリソースについて精度 $T = 4$ の擬似算術エンコーダの状態、出力、およびすべての可能な遷移を提示している。第 1 の例では、列グループ $C1$ 、 $C2$ が言及される。変数 $nsc1_k$ の値は状態モデルに含まれない。この変数の増分だけが文献 [4] と同様に表中の文字 f により信号で伝えられる。

【 0 0 7 1 】

エンコーダは、現在の間隔 $[low S_k, up S_k [$ により規定される間隔 $[0, 4 [$ の整数再分割に対応する 3 つの可能な状態を有する。これらの再分割の選択枝について以下の文献、すなわち、

40

[6] 「擬似算術符号化に基づく高速テキスト圧縮の設計および分析 (Design and Analysis of fast text compression based on quasi-arithmetic encoding)」、P. G. Howard および J. S. Vitter、データ圧縮会議 (Data Compression Conference) に於いて、98 ~ 107 ページ、ユタ州スノウバード、1993 年 3 月 ~ 4 月、

を参照する。入力バイナリ記号 0 の確率に応じて、現在の間隔 $[low S_k, up S_k [$ は整数間隔に再分割される。列 $C2$ では入力バイナリ記号が 0 または 1 であるのかに応じて、現

50

在の間隔 $[lowS_k, upS_k]$ の可能な整数間隔への再分割は、示されたビットの放出を生じさせる結果、またはオートマトンの次の状態を決定する前の適切なリスケーリングを生じさせる結果となる。

【0072】

入力記号がバイナリ記号である場合には、状態の数は、列 C 2 での 0 および 1 としてよりむしろ、列 C 3 でバイナリ記号を「起こりそうでない」(LPS) および「さらに起こりそうである」(MPS) として識別することによりさらに削減可能である。これは、バイナリ記号の値よりむしろ各バイナリ記号の確率を考慮することに対応しており、MPS と LPS は 0 ビットまたは 1 ビットのどちらかに対応している。バイナリソース確率の可能な組合せの数は削減される。これにより、特定の遷移を組合せること、およびオートマトンの状態を排除することが可能になる。

10

【0073】

擬似算術エンコーダの連続状態のシーケンス $E_0 \dots E_k$ はマルコフ連鎖を形成し、そのエンコーダの出力はこの連鎖における遷移の関数である。

【0074】

擬似算術符号化と同様に、復号化は、表 2 に示されているような有限数の状態 E_N を使用して確率オートマトンによりモデリングできる。

【0075】

算術デコーダの状態は 2 つの間隔、すなわち $[lowU_n, upU_n]$ および $[lowS_{k_n}, upS_{k_n}]$ によって定義される。間隔 $[lowU_n, upU_n]$ は、ビット U_1^n のシーケンスの受信により規定される間隔 $[0, 1]$ のセグメントに対応している。間隔 $[lowS_{k_n}, upS_{k_n}]$ は、記号 $S_1^{K_n}$ のシーケンスが復号化された時に得られる間隔 $[0, 1]$ のセグメントである。この間隔は、符号化された記号ごとの符号化と同様に、それぞれの復号化されたバイナリ記号の確率を累積することにより更新される。それはエンコーダの動作を再現する。

20

【0076】

算術エンコーダおよび算術デコーダは同期されていない。実際には、符号化の時に、記号の受信が必ずしもビットの放出という結果にならない。同様に、復号化の時に、ビットの読み取りが必ずしも記号の生成という結果にならない。したがって、符号化されるシーケンス内の第 1 の記号を読み取った結果がビットの放出にならない場合には、復号化の時に第 1 のビットの読み取りが第 1 の記号を復号化するのに十分となることもある。両方の間隔が、数値精度の問題を回避するために、符号化の時のようにスケールされる。

30

【0077】

表 2 の例では、確率オートマトンは $T = 4$ という値について定義されている。表は、 $T = 4$ のバイナリソースの擬似算術デコーダと MPS / LPS の簡略化とのために状態、遷移、および出力を与える。

【0078】

E_n を「ビットインスタント」 n の時に、つまり受信されたビット U_n の処理の瞬間の時にオートマトンの状態とする。この状態は間隔 $[lowU_n, upU_n]$ および $[lowS_{k_n}, upS_{k_n}]$ の値により定義される。(最も起こりそうな MPS 記号の確率により列 K 1 に示される) ソースの定常分布に応じて、現在の間隔 $[lowS_{k_n}, upS_{k_n}]$ は、整数間隔に再分割される。入力ビットが列 K 2 で $U_n = 0$ であるのかまたは列 K 3 で $U_n = 1$ であるのかに応じて、現在の間隔 $[lowS_{k_n}, upS_{k_n}]$ の整数間隔への可能な再分割の結果が、示された MPS 記号または LPS 記号の放出になり、またはオートマトンの次の状態を決定する前に適切なリスケーリングになる。

40

【0079】

表 1 および表 2 の確率オートマトンを使用して、エンコーダは状態 $E_k = 1$ および $E_n = 1$ でそれぞれ開始する。

【0080】

符号化方法 / 復号化方法で使用するために、および対応するエンコーダ / デコーダでの

50

実行のために、さまざまなモデルが予想される。このようにして、ソースモデルとエンコーダの積から生じるモデル、またはソースモデルとデコーダとの積から生じるモデルであることがある、いわゆる「プロダクト」モデルが予想される。

【 0 0 8 1 】

一般的には、プロダクトモデルの状態は、2つのモデル（ソースとエンコーダまたはソースとデコーダ）のそれぞれの状態に含まれる情報を収集しなければならない。一般的な場合には、「プロダクト」モデルの状態 - 空間のサイズは先験的に知り得ない。それは、バイナリソースモデルとパラメータ T から計算可能である。

【 0 0 8 2 】

ソース / エンコーダのプロダクトモデルに関して、プロダクトモデルの各遷移により生成される各バイナリ要素の中のビット数はランダムであるので、第2のバイナリ要素のシーケンスとプロダクトモデルの状態との間の依存性の構造はランダムである。この依存性構造を「捕捉（キャプチャ）する」ために、プロダクトモデルによるエンコーダの状態 X_k によって表されるマルコフプロセスがカウンタ N_k により増大されて、次の依存性構造 $(X, N) = (X_1, N_1) \dots (X_K, N_K)$ が得られる。プロダクトモデルの各状態 X_k は、エンコーダモデルの状態と、ソースモデルの状態と、第 k 番目の記号すなわち $(X_k, N_k) = (\text{low } S_k, \text{up } S_k, C_k, N_k)$ が符号化された時に放出されるビット数とによって定義される。これは、図8に図で描かれている依存性の構造をもたらす。この構造に従って、バイナリ記号 S_1^K のシーケンスはビット $U_1^{N_K}$ のシーケンスに変換される。ここに、 N_k は、 K 個の記号が符号化された時に放出されるビットの数である。状態 X_{k-1} および入力記号 S_k を考えれば、プロダクトモデルは、放出されなければならないビット $U_{N_{k-1}+1}$ から U_{N_k} を指定するとともに次の状態 X_k を指定する。ビットは特定の遷移により放出されなくてもよい。 $(X_k, N_k) = (\text{low } S_k, \text{up } S_k, C_k, N_k)$ と、トレリス $(X_{k+1}, N_{k+1}) = (\text{low } S_{k+1}, \text{up } S_{k+1}, C_{k+1}, N_{k+1})$ の次の状態との間での遷移の確率は、バイナリソースモデル $P(C_{k+1} | C_k) = P(S_{k+1} | C_k)$ により与えられる。第2のバイナリ要素 $Y_{N_{k-1}+1}$ から Y_{N_k} は、第1のバイナリ要素 $U_{N_{k-1}+1}$ から U_{N_k} が放出された時に伝送チャンネルの出力で得られる。そのような第2のバイナリ要素は第1のバイナリ要素についての観察である。

【 0 0 8 3 】

ソースモデルとエンコーダモデルとから導かれるプロダクトモデルの一例は図6および図7で与えられる。図6Aの表で規定されたプロダクトモデルの例は、図4Cのソースモデルと表1のエンコーダモデルとの積から導かれる。理論的には、この積から生じる状態 - 空間のサイズは、2つのモデル（ソースおよびエンコーダ）の状態 - 空間寸法の積により与えられる。しかしながら、より少ない数の状態を生じさせる簡略化が生じる。

【 0 0 8 4 】

列C3の簡略化された状態モデルを使用して表1 ($T = 4$) の算術エンコーダモデルは、図7Aおよび図7Bにトレリス形式で表現され、図7Aは確率 $0.63 \cdot P(MPS)$ に対応し、図7Bは確率 $0.37 \cdot P(MPS) \cdot 0.63$ に対応する。これらの図のどちらかはバイナリソースの確率に依存して使用され得る。Mに関するソースに基づく変換から生じたバイナリソースの確率は、このバイナリソースのモデルの状態 C_k に依存する。このようにして、ソース / エンコーダのプロダクトモデルの遷移はソース分布の関数である。図6Aの表は、図4Cのソースモデルの状態 C_k と、遷移 S_k の確率（図4Cの遷移 $S_k = 0$ および $S_k = 1$ は $S_k = LPS$ および $S_k = MPS$ として変更されている）と、ソース確率の関数として図7Aと図7Bに描かれているエンコーダモデルの遷移の確率との関数として構築されている。図6Aの列M1だけが、ソースモデルおよび対応エンコーダの状態変数により、列M2で規定されたプロダクトモデルの4つの状態を含む。ソースモデルは3つの状態を有しエンコーダモデルは2つの状態を有するので、プロダクトモデルは最大6つの状態を有する。実際には、図7Aと図7Bを用いて後述されるように簡略化が発生する。図7Aは、エンコーダの状態 $E_k = 0$ に関してソースモデルの確率 $0.$

63 < P (M P S) に対しエンコーダの状態 $E_k = 1$ で継続できることを示している。しかしながら、図 7 B は、エンコーダの状態 $E_k = 0$ から状態 $E_k = 1$ に達することは可能ではないことを示している。

【 0 0 8 5 】

したがって、図 6 A のプロダクトモデルでは、ソースモデルおよびエンコーダモデル ($E_k = 1$, $C_k = 1$) および ($E_k = 1$, $C_k = 2$) の状態のペア (対) に対応するプロダクトモデル $X_k = 4$ および $X_k = 5$ の状態は、ソースモデルの確率が $0.37 < P (M P S) < 0.63$ になるような確率状態に対して存在する。列 M 3 および列 M 4 は、入力記号 (M P S または L P S) に対して、放出されたビットとプロダクトモデルの次の状態とを示す。

10

【 0 0 8 6 】

図 9 A または図 9 B によるソース / デコーダモデルに対して、このプロダクトモデルの状態は、ソースモデルおよびデコーダモデルの状態情報を収集しなければならない。したがって、このプロダクトモデルの状態 X_n は、 $X_n = (\text{low} U_n , \text{up} U_n \text{ low} S_{K_n} , \text{up} S_{K_n} , C_{K_n})$ によって規定される。ここに、 K_n は、 n 個のビットが受信された時に復号化される記号の数である。プロダクトモデルの各遷移によってもたらされた記号の数はランダム (任意) である。その結果、第 2 のバイナリ要素のシーケンスと復号化された記号との間の依存性構造はランダムである。ソース / デコーダのプロダクトモデルのマルコフ連鎖 $X_1 \dots X_n$ は、カウンタ変数 K_n により増加され、図 10 に示されている依存性構造を生じさせるマルコフ連鎖 $(X , K) = (X_1 , K_1) \dots (X_n , K_n)$ を与えるようにしている。この構造に従って、ビット U_1^N のシーケンスが記号 $S_1^{K_N}$ のシーケンスに変換される。ここに、 K_n は、 N 個のビットが受信された時に復号化される記号の数である。状態 X_n および入力ビット U_{n+1} を考えれば、オートマトンは、バイナリ記号 $S_{K_{n+1}}^{K_{n+1}}$ のシーケンスをもたらし、次の状態 X_{n+1} を指定する。トレリスの (X_n , K_n) と (X_{n+1} , K_{n+1}) との間の遷移の確率はバイナリソースモデルに依存する。それらは添付の式 (17) により与えられる。ソース / エンコーダモデルの場合と同様に、このモデルの状態の簡略化は、使用されたデコーダモデルとソースモデルとに依存して可能である。図 9 A または図 9 B の例では、使用されたソースモデルは図 4 C のソースモデルであり、デコーダモデルは表 2 のデコーダモデルである。

20

【 0 0 8 7 】

算術符号用のロバスト復号化方法は、以前に説明された依存性モデルに基づいている。

30

【 0 0 8 8 】

ロバスト復号化方法の第 1 の実行が図 11 に描かれている。

【 0 0 8 9 】

関連される数学的展開については付録 1 を参照する。

【 0 0 9 0 】

デコーダは、その入力で復号化されるビットのストリームを受け取る。ソースモデルと、復号化される受信済みのビットの数を表現する変数 N と、観察された第 2 のビット (すなわち第 2 のバイナリ要素) Y_n を知っている第 1 のビット (つまり第 1 のバイナリ要素) U_n を放出した確率 $P (U_n / Y_n)$ とはデコーダ用の入力データであり、すなわち、ほんの一例としては、これらのデータは、復号化されるビットストリームとともに入力を受信可能であり、または、デコーダによりアクセス可能なメモリから読み取り可能である。確率 $P (U_n / Y_n)$ は、チャンネル特性から得ることができる。

40

【 0 0 9 1 】

M A P (事後的な最大値 A) 基準は、付録 1 の式 (3) によって示されるように、使用可能な観察 Y に基づいたプロセス X の最適ベイズ推定に対応している。最適化はすべての可能な「シーケンス」について実行される。この第 1 の実行では、この基準は、記号クロックプロセスと見なすことができるプロセス (X , N) の非表示状態と、観察されたビットシーケンスを考えればビットクロックプロセスと見なすことができるプロセス (X , K) の非表示状態との推定に適用される。推定は、ビットクロックプロセスまたは記号

50

クロックプロセスのどちらかとして実行され得る。その実行が単純であることの理由から、ビットクロック推定が好まれる。

【 0 0 9 2 】

その場合には、非表示状態 $(X, K) = (X_1, K_1) \dots (X_N, K_N)$ の集合の推定は、チャンネル出力での観察 Y_1^N を考えれば、復号化された記号 $S = S_1 \dots S_{K_N} \dots S_{K_N}$ の対応シーケンスの推定と同等である。最良のシーケンス (X, K) は、式 (4) を使用してペア (X_n, K_n) での局所的な確率から獲得可能である。

【 0 0 9 3 】

$P(X_n, K_n | Y)$ の計算は、解凍式 (5) を使用して実行でき、 α は正規化ファクタ (正規化係数) を示している。マルコフ連鎖の依存性によって、以下の文献、すなわち

10

[7] 「記号誤り率を最小にするための線形符号の最適復号化 (Optimal decoding of linear codes for minimizing symbol error rate)」、L. R. Bahl、J. Cocke、F. Jelinek、および J. Raviv、IEEE 変換理論に関する報告書 (IEEE trans. on transformation theory)、02:284~287、1974 年 3 月、

の中で説明された BCJR アルゴリズムに基づいて、上記の式の右辺における両方の項の帰納的な計算が可能となる。

【 0 0 9 4 】

20

したがって、ステップ 300 では、トレリスが、たとえば図 9A のソース/デコーダモデルを使用して最初に構築され、対応する確率がトレリスの遷移ごとに計算される。このように、プロダクトモデルに基づいて、順方向の掃引が、最初の状態から最後の状態へプロダクト状態図 (積状態図) の遷移によって連結される状態の連続的な構築を可能にする。それぞれの状態は、初期状態からの第 2 のバイナリ要素のフローの中で過去のビットの一定数に対して予想される数の復号化された記号を規定する。この同じステップの間に BCJR アルゴリズムの順方向の掃引によって、3 つの条件付き確率の積、すなわち、再帰的確率、(式 17 で示されソースモデル内で過去に遷移確率 $P(S_k | C_{k-1})$ の積を表すような) プロダクトモデルにより与えられた状態 (x_{n-1}, k_{n-1}) と状態 (x_n, k_n) との間の遷移の確率、および、第 2 のバイナリ要素 Y_n を与えられた状態 (x_{n-1}, k_{n-1}) と状態 (x_n, k_n) との間の遷移をトリガする (0 または 1 であってもよい) ビット U_n を放出した確率の積の x_{n-1}, k_{n-1} の可能な値の集合の合計として分解項 $P(X_n = x_n, K_n = k_n | Y_1^N)$ が計算可能になる (式 (6))。後者の確率はチャンネル特性から得られる。再帰的計算は、開始状態 (0, 0) で初期化され、ビット・クロックの $n = 1 \dots N$ を使用して任意の可能な状態 (x_n, k_n) について $P(X_n, K_n | Y_n^1)$ の計算を可能にする。このようにして、この順方向の掃引はプロダクトモデルのトレリスの構築と、連続的確率 $P(X_n, K_n | Y_n^1)$ とを容易にし、インスタント (瞬間) n ごとに第 2 のバイナリ要素ストリーム Y_1^N のビットを与えられた一定の状態 (X_n, K_n) で終了の計算を容易にする。

30

【 0 0 9 5 】

40

ステップ 302 では、逆方向の掃引は、 N から 1 まで連続的なビットインスタントのそれぞれでの可能な状態 (x_n, k_n) について分解項 $P(Y_{n+1}^N | X_n = x_n, K_n = k_n)$ を得る働きをする。その項は、3 つの条件付き確率の積、つまり、すべての可能な「最後の状態」 (X_N, K_N) について初期化された再帰的確率、プロダクトモデルにより与えられた状態 (X_{n+1}, K_{n+1}) と状態 (X_n, K_n) との間の遷移の確率、および、第 2 のバイナリ要素 Y_{n+1} を与えられた状態 (X_{n+1}, K_{n+1}) と状態 (X_n, K_n) との間の遷移をトリガするビット U_{n+1} を放出した確率の積の合計として計算される (式 (7))。前記のように、後者の確率はチャンネルモデルから得られる。したがって、この逆方向の掃引によって、連続確率 $P(Y_{n+1}^N | X_n = x_n, K_n = k_n)$ の計算と、 N から 1 までのインスタント n ごとに一定の状態 (X_n, K_n) を知っ

50

て第2のバイナリ要素ストリーム Y_1^n のビットを観測する計算とが可能になる。

【0096】

記号 K の数が既知であるとき、ステップ302で逆方向の掃引の間に終了制約が追加できる。したがって、記号 $K_N = K$ の正しい数につながらないトレリスのすべての経路が抑制される。このように、逆方向の掃引は、最終の状態から最初の状態へプロダクト状態図からの状態の連続的な抑制を含んでおり、そのプロダクト状態図では第2のバイナリ要素のストリームにおけるビットの一定数に対して記号の予想数が記号の既知の最大数と異なっている。それ故に、ステップ300での第1の掃引は、トレリスの最初の状態で開始し、トレリスのさまざまなセクションを連続的に構築する。この順方向の掃引の間に、どの経路が終了制約を満たさないのかを知ることはできない。いったん順方向の掃引が終了すると、制約を満たさないトレリスの最後のセクション ($n = N$) の状態が抑制される。次に、逆方向の掃引によって、終了しないすべての分岐の抑制が可能になる。

10

【0097】

ステップ304では、順方向の掃引および逆方向の掃引の得られた確率が1から N の範囲の n の値ごとに積の形式で結合されて、状態 n ごとに確率 $P(X_n, K_n | Y)$ が得られる。このように、構築されたトレリスの遷移に関連付けられた確率が推定される。

【0098】

ステップ306では、式(4)の積が1から N の範囲の n を使用して確率 $P(X_n, K_n | Y_1^n)$ で実行される。この積はこれらの確率のさまざまな組合せに対して実行される。したがって、最大の確率 $P(X, K | Y)$ が保持され、その積を構成する確率 $P(X_n, K_n | Y)$ はトレリス内で最も起こりそうな経路を決定するために使用される。トレリスとソース/デコーダのプロダクトモデルの対応表とに基づいて、その経路に対応する記号を決定すること、および記号 S_1^K のシーケンスを再構築することが可能である。

20

【0099】

別の可能な実行では、推定が行われたトレリスが、ソース/デコーダのプロダクトモデルと観察されたビットのシーケンスの長さ N とに基づいて事前に計算され記憶される。したがって、図12を参照して、ソース/デコーダモデルと K および N の値とに基づいて、トレリスが単一の順方向の掃引においてステップ400で構築され、ステップ402で逆方向の掃引によって簡略化される。このトレリスは、ソース/デコーダモデルと値 K および値 N とが同じであるビットストリームが出現するたびに、使用するメモリ内に記憶される。

30

【0100】

$N = 6$ ビットを含む受信ビットのストリームに対して、 $K = 7$ であることが判明した既知の数の記号と図9Aで描かれるようなソース/デコーダモデルとについて、格子構築が図13で図示される。格子は、それぞれの状態 (X_n, K_n) で複数のペアの値 (X_n, K_n) を含む。たとえば、ペア (X_1, K_1) は、放出されたビット U_1 が0または1の値を有するか否かに依存して、値 $(1, 1)$ または値 $(2, 1)$ を取ることができる。

【0101】

(トレリスと同じモデルを有し、トレリスと同じ K および N の値を有する) ビットのストリームが出現するたびに、順方向の掃引と逆方向の掃引とが、たとえば平行で、ステップ404およびステップ406でのビットストリームで実行される。確率 $P(X_n, K_n | Y_n^1)$ および $P(Y_{n+1}^N | X_n, K_n)$ は、式(6)および(7)で示されるように計算される。これらの確率の積は、 n の値ごとにステップ408で得られ、トレリスのそれぞれの遷移に対応する。次に、ステップ410では1から N の範囲の n に対して確率の積が(4)に示されるように計算されて、確率 $P(X, K | Y)$ が得られる。トレリス上で最も起こりそうな経路と記号の対応シーケンスとを決定するために、これらの確率 $P(X, K | Y)$ のうち最大のものが保持される。

40

【0102】

図13の例に従って、最大の確率 $P(X, K | Y)$ を得る最も起こりそうな経路は太字で記されている。状態 $(X_0, K_0) \dots (X_N, K_N)$ のシーケンスは、表8Aの

50

読み取りによって、この経路に対応する記号シーケンスの決定を可能にする。

【 0 1 0 3 】

擬似算術符号を復号化する最適な方法が与えられている。この方法は、たとえば図 1 3 のような推定トレリスの作成および操作を必要とする。このトレリスのサイズは、ソース / デコーダのプロダクトモデルの状態の数（たとえば図 9 A のソース / デコーダのプロダクトモデルに対して 4 つの状態が可能である）と、それぞれのビットインスタント n で変数 K_n が取ることができる値の振幅とに依存する。ソース / デコーダのプロダクトモデルの状態の数はそれ自体バイナリソースモデルと、擬似算術符号化精度パラメータ T とに依存する。したがって、擬似算術エンコーダの圧縮効率と推定方法の複雑さとの間でのトレードオフを制御するために、パラメータ T の値を制御することは十分である。4 と同じように小さな T の値は、たとえば、妥当な過剰率を生成する。しかしながら、最大圧縮が必要とされる場合には、 T ははるかに大きな値を有さなければならず、 T の最小必須値はソースの分散に依存する。このような場合の大部分では、 T の値があまりにも大きすぎて前述された最適な推定方法を使用することができない。また、ソース / デコーダのプロダクトモデルの状態の数はあまりにも大きすぎてそれらの状態を事前に計算することもできない。

10

【 0 1 0 4 】

算術符号化と擬似算術符号化とに適用可能な推定方法は、選択された精度 T に関わりなく、つまり選択された圧縮効率に関わりなく、ここで説明される。

【 0 1 0 5 】

この方法は、過去に見られたものと同じソースモデル、エンコーダモデル、およびデコーダモデルに依存する。エンコーダモデルおよびデコーダモデルの状態の数があまりにも大きすぎてそれらを事前に計算できないので、対応する推定トレリスもまた構築できない。それ故に、以下に提案される推定方法は、系図と見なされる推定トレリスの部分集合の調査に基づいている。この方法は逐次推定法と呼ばれる。それは、ソース / エンコーダのモデルに基づいて適用可能であり、「記号クロック」を使って、つまり検出される記号の数に従って増分されるカウンタを使って逐次計算を実行できる。また、それは、ソース / デコーダのモデルに基づいて適用可能であり、「ビットクロック」を使用して、つまり復号化されるビットの数に従って増分されるカウンタを使用して逐次計算を実行できる。この第 2 の推定方法は、ほんの一例として、バイナリソースに対して提示される。また、それは、 M に関するソース ($M > 2$) にも適用可能である。

20

30

【 0 1 0 6 】

図 1 4 は、いわゆる記号クロックの逐次推定法を示し、すなわち、ソースモデル / エンコーダに基づいている。したがって、ソース / エンコーダモデルの依存性構造を示す図 8 に従って、ビットのストリームの状態 (X, N) のシーケンスを推定することは、これらの状態の間の遷移のシーケンスを推定すること、つまり、チャンネル出力での観察 Y を与えられたシーケンス S を推定することと同等である。これは、(8) に提示される数学公式化によって表される。ここに、 α は正規化係数を示す。(T という大きな値に対して) 最適な算術符号化の場合では、放出されたビットが均一かつ独立に分布されるので、この正規化係数は $P(U_1^N) / P(Y_1^N) = 1$ である。

40

【 0 1 0 7 】

ステップ 5 0 0 では、記号インスタント k ごとに第 2 のバイナリ要素 $P(S_1^k / Y_1^{N_k})$ のフローのビットを知って記号のシーケンス (1 から k) を復号化する確率は、式 (9) の分解を使って単一順方向の掃引で計算可能である。ここに、 N_k は状態 X_k にあるときに放出されたビットの総数である。式 (1) で定義されたバイナリソースモデルを考慮に入れメモリレスチャンネルを考えると、そのモデルの状態に対応する第 2 のバイナリ要素を観察する確率は、この状態に対して第 1 のバイナリ要素だけに依存し、その確率は式 (1 0) に示されるように書き換え可能である。

【 0 1 0 8 】

計算は、バイナリソースの場合では 2 つの可能な値 0 または 1 を取ることができる S_1

50

$= s_1$ 、とMに関するソースの場合ではM個の可能な値を取ることができる $S_1 = s_1$ を使用して (11) に示されるように初期化される。 C_1 はソースモデルの初期状態である。考慮されるメモリレスチャンネルに対して、このチャンネルにより生じるノイズをモデリングする式の第2の項は、送信された第1のバイナリ要素 U_n について第2のバイナリ要素 Y_n を観察する確率の積として、ソース/エンコーダモデルの一定の状態 X_k について式 (12) により与えられる。この初期化に基づいて、式 (10) を使って連続的な記号インスタント k ごとに $P(S_1^k / Y_1^{N_k})$ を計算する。

【0109】

すべての可能な記号シーケンスが考慮されると、推定トレリスのサイズが記号 K の数とともに急激に増加する。このように、ステップ 502 によって、1 から x_1 までの範囲の記号 K の数について確率 $P(S_1^k / Y_1^{N_k})$ の計算が許容される。値 x_1 以上からの記号に対して、所望レベルの複雑さで推定トレリスを維持するために、ブルーニング（不要なものの削除）がステップ 504 で実行される。 x_1 は記号 K の数以下の整数である。

【0110】

ステップ 504 でのブルーニングにはさまざまな実行が含むことができる。一例として、2つの実行が後述される。その実行は組合せ可能であり、または別々に利用可能である。

【0111】

このように、ステップ 504 でのブルーニング動作の第1の実行では、状態 (X_k, N_k) の W 個の最も起こりそうなノードだけが数 $k = x_1$ から開始して選択され、計算された確率（尤度と呼ばれる）が顕著であることを確実にするようにしている。 W の値は、使用可能なメモリリソースに依存している。したがって、状態 (X_k, N_k) ごとに、第2のバイナリ要素 $P(S_1^k / Y_1^{N_k})$ のフローのビットを知って記号のシーケンス（1 から k まで）を復号化する計算済み確率のうちで W 個の最大確率だけが選択され記憶される。変形例では、この制約を精緻化することにより推定の正しい終了を確実にすることができる。それ故に、（記号クロックに対応する） N_k の可能な値ごとに w 個の最も起こりそうなノードが選択され保持される。 N_k は、符号化の間に、符号化される k 個の記号のために放出されたビットの数である。 W と w との関係は、もっとも起こりそうなシーケンスおよび最も起こりそうもないシーケンス S_1^k のそれぞれの確率に依存して、 $N_{k_{max}}$ および $N_{k_{min}}$ とともに式 (19) で与えられる。

【0112】

ステップ 504 でのブルーニングにおける第2の実行では、ブルーニング基準は経路の尤度に適用されるしきい値に基づいている。経路の尤度は式 (8)（記号クロック）で与えられる。この式では、項 $P(Y | U)$ は式 (12) に示されるようにチャンネルモデルにリンクされている。 $P(Y_n = y_n | U_n = u_n)$ は、ビット u_n が送信されたことを知ってビット y_n が観察される確率を表し、またはビット u_n が放出されたことを知って観察されたビットが誤りである確率と等価なビット y_n を受信する確率を表している。 p を、推定中にチャンネル上で許容されるように任意に決定されたバイナリエラーの平均確率とする。ブルーニングは、チャンネル出力時のバイナリエラーの平均確率が p より大きいすべての経路を抑制することを含んでいる。したがって、それぞれの記号インスタント N_k で、ブルーニング基準は、 $P(Y_{N_k} | U_{N_k}) < (1 - p)^{N_k}$ が確認された推定トレリス上のすべての経路が抑制されるように、確率 $P(Y_{N_k} | U_{N_k})$ と比較されたしきい値 p を含んでいる。第2のバイナリ要素 $P(S_1^k / Y_1^{N_k})$ のフローのビットを知って最小しきい値 p より高い確率 $P(Y_{N_k} | U_{N_k})$ から計算された記号のシーケンス（1 から k ）を復号化する確率だけが記憶される。

【0113】

構成単位の値 k を増分した後、記号 N_k （ $k > x$ を満たす）の数に対する確率 $P(S_1^k / Y_1^{N_k})$ がステップ 506 で計算される。ステップ 504 で実行されたブルーニングは、この新しい記号インスタント N_k について繰り返される。ステップ 504 およびステップ 506 は、値 K まで増分される値 k について実行される。ステップ 510 では、い

10

20

30

40

50

ったん値 K になると、限界に近い後天的な確率 $P(S_1^K / Y_1^{N^K})$ はすべての可能なシーケンス s_1^K について知らされる。最も起こりそうなシーケンス s_1^K 、つまり、得られた推定トレリスの可能なシーケンスの中で確率 $P(S_1^K / Y_1^{N^K})$ が最高であるシーケンスが選択される。

【0114】

ビット N の数が既知であるとき、それは推定トレリスの終了に対する制約を構成することができ、制約 N を満たさない経路は考慮されない。

【0115】

図15は、いわゆるビットクロックの連続推定方法を示し、つまりソース/デコーダモデルに基づいている。したがって、ソース/デコーダモデルの依存性構造を示す図10に従って、ビットのストリームを得るために状態 (X, K) のシーケンスの推定は、これらの状態の間の遷移のシーケンスの推定、つまりチャンネル出力での観察 Y を与えるシーケンス S の推定と等価である。これは、(13)で提示された数学公式化により表される。ここに、 α は正規化係数を示す。(Tの大きな値に対して)最適な算術符号化の場合では、放出されるビットが一様かつ独立に分布されるので、この正規化係数は $P(U_1^n) / P(Y_1^n) = 1$ である。

【0116】

ステップ600では、(後に定義される)ビット n ごとに第2のバイナリ要素 $P(S_1^K / Y_1^n)$ のフローのビットを知って K_n 個の記号のシーケンスを復号化する確率は、式(13)の分解を使用して単一の順方向の掃引で計算され得る。ここに、 K_n は状態 X_n におけるビット n の受信時に復号化された記号の総数である。状態 (X_n, K_n) と状態 (X_{n+1}, K_{n+1}) との間の遷移の確率を(2)で定義するバイナリソースモデルを考慮に入れると、記号シーケンス $S_1^{K_n}$ を獲得する確率 $P(S_1^{K_n})$ は式(15)を使用して式(14)に従って再帰的に表される。メモリレスチャンネルを考慮することは、この状態について送信された第1のバイナリ要素に依存し、確率 $P(Y_1^n / U_1^n)$ は、(16)に示されているような確率 $P(S_1^{K_n} / Y_1^n)$ を得るために書き換えて使用可能になる。

【0117】

図14と同様に、式(16)による確率 $P(S_1^{K_n} / Y_1^n)$ の計算は、バイナリソースの場合では2つの可能な値0または1を取り、および、 M に関するソースの場合では M 個の可能な値を取る $n = 1$ 、 $S_1 = s_1$ について初期化される。 C_1 はソースモデルの初期状態である。この初期化に基づいて、式(16)を使って連続的なビットインスタント n ごとに $P(S_1^K / Y_1^{N^K})$ を計算する。

【0118】

記号の全ての可能なシーケンスが考えられるので、推定トレリスのサイズは、推定される記号 K の数とともに急激に増加する。このように、ステップ602は、1から $\times 2$ の範囲のビットの数 N についてステップ600で確率 $P(S_1^{K_n} / Y_1^{N_n})$ の計算を容易にする。ここに、 $\times 2$ はビットのしきい値の数を示す整数である。値 $\times 2$ 以上からのビットに対して、所望なレベルで推定トレリスの複雑さを維持するために、ブルーニングがステップ604で実行される。

【0119】

ステップ604でのブルーニングは、図14のステップ504の実行に対応するさまざまな実行を含んでいる。そのような実行は組合せ可能であり、または別々に利用可能である。

【0120】

このように、ステップ604でのブルーニング動作の第1の実行では、状態 (X_n, K_n) の W 個の最も起こりそうなノードだけが、計算された確率(尤度と呼ばれる)が顕著であることを確実にするために、数 $n = \times 2$ から開始して保持される。したがって、それぞれの状態 (X_n, K_n) に対して、第2のバイナリ要素 $P(S_1^K / Y_1^{N^K})$ のフロ

10

20

30

40

50

一のビットを知って K_n 個の記号のシーケンスを復号化する計算済み確率のうち W 個の最大確率だけが記憶される。

【0121】

変形例では、(ビットクロックに対応する) K_n の可能な値ごとに w 個の最も起こりそうなノードだけを保持することが可能である。 K_n は、復号化の間に受信された n 個のビットについて放出された記号の数である。 W と w との関係は、最も起こりそうなシーケンスおよび最も起こりそうもないシーケンス S_1^k の確率にそれぞれ依存して、 K_{nmax} および K_{nmin} を使って (20) で与えられる。

【0122】

ステップ604でのブルーニングの第2の実行では、ブルーニング基準は経路の尤度に適用されたしきい値に基づいている。経路の尤度は式(13)(ビットクロック)により与えられる。この式では、用語 $P(Y|N)$ は、式(16)に示されるように、チャンネルモデルにリンクされている。図13について提案されるように、ブルーニングは、チャンネル出力でのバイナリエラーの平均確率が p より大きいすべての経路を抑制することを含んでいる。したがって、それぞれのビットインスタント n でのブルーニング基準は、 $P(Y_n|U_n) < (1-p)^n$ が確認された推定トレリス上のすべての経路が抑制されるように、確率 $P(Y_n|U_n)$ に比較されたしきい値 p を含んでいる。第2のバイナリ要素 $P(S_1^k/Y_1^n)$ のフローのビットを知って最小のしきい値 p より高い確率 $P(Y_n|U_n)$ から計算された次の記号を復号化する確率だけが記憶される。それ故に、それぞれのビットインスタント n_s で、ブルーニング基準 $P(Y_n|U_n) < (1-p)^n$ が満たされた推定トレリス上のすべての経路が抑制される。

【0123】

構成単位の値 n を増分した後に、ビット n ($n > x2$ を満たす) の数についての確率 $P(S_1^k/Y_1^n)$ がステップ606で計算される。ステップ604で実行されるブルーニングはこの新しいビットインスタント K_n について繰り返される。ステップ604およびステップ606は、値 N まで増分された値 n について実行される。ステップ610では、いったん値 N に達すると、最も起こりそうなシーケンス S_1^N が選択され、つまり、得られた推定トレリスの可能なシーケンスのうちで確率 $P(S_1^k/Y_1^N)$ が最大であるシーケンスが選択される。

【0124】

推定の最後には、限界に近い後天的な確率 $P(S_1^k/Y_1^{Nk})$ が、ブルーニング後に残っているすべてのシーケンス s_1^k について知らされる。

【0125】

前記の終了制約は、記号のシーケンスの最後で同期を強制するための手段と見なすことができる。実際には、それらは、復号化された記号の数と観察されたビットの数との間で正しい対応を得るためにデコーダを制限している。しかしながら、それらの制限にはシーケンスの真中で同期を確実にする効果はない。

【0126】

記号シーケンスに余分な情報を追加すると、シーケンスの全体での再同期が容易になる。このように、余分な記号つまりマーカは、記号シーケンスにおける既知の位置に導入される。これらの記号の選択は任意であり、(2、4、11、5、14)において説明される技術原則に基づいている。ほんの一例として、アルファベット内の記号の1つが選択可能であり、または最後に符号化された記号が反復可能である。これらのマーカは既知の記号-クロック位置で挿入される。その結果、バイナリストリーム中で対応する余分なビットの位置が、符号化された記号のシーケンスに依存しランダムになる。

【0127】

明らかに、前述された方法はこの余分な情報を考慮に入れなければならない。この知識は推定プロセスによって生かされ得る。変数 K_n (k 記号-クロック) は、マーカがいつ予想されるのかを示す。それに対応して、推定トレリスにおいて対応する遷移確率が更新される。ヌルの確率は、予想されるマーカを放出しないすべての遷移に割り当てられ、そ

の一方、1の確率は他に割り当てられる。それ故に、トレリス内の特定の経路が抑制可能になり、そのことが状態の数の低減と再同期能力の改善とにつながる。

【0128】

ロバスト復号化方法の例が実例として前述されている。他の変形例を想定することができる。さらに、他のプルーニング基準が提案可能であり、本発明の方法と組合せることが可能である。Mに関する変換/バイナリ変換は必須ではないが、Tの小さな値を使用可能にする。また、バイナリソースについて前記に提示された復号化方法は、Mに関するソースについて役に立つことがある。

【0129】

上記の説明で述べられているように、本発明は、擬似算術符号と呼ばれる低減精度の算術符号を含んだ算術型符号の使用に基づいて、算術符号を復号化するロバストな方法を提案している。これらの擬似算術符号の精度は、求められている効果に応じて調整可能なパラメータである。非常に高い精度により、算術符号の圧縮性能と同一の圧縮性能が達成可能になる。精度が低いと、圧縮での妥当な損失を犠牲にしてロバストネスを増大する効果がある。

【0130】

従来の「インスタントネス(瞬間的な)」復号化と比較すると、本発明はデコーダの非同期の現象を実質的に低減している。「逐次」復号化の他の従来の方法、またはツリー構造を使用する他の従来の方法と比較すると、本発明は、復号化の圧縮効率、ロバスト、および複雑さの間でのトレードオフを制御する際に、さらに大きな柔軟性を提供する。実際には、本発明は、チャンネル上で観察された誤り率に従って、端末により許容された複雑さに応じて選択可能な1セットの方法からなる。逐次復号化方法は、復号化プロセスの複雑さを制御することを可能にするプルーニング技術と結合される同期メカニズムを含んでいる。複雑さ/ロバストネス/圧縮トレードオフのパラメータ化を伴ったトレリスベースの復号化プロセスは、プロセスをチャンネルデコーダ(畳み込み符号デコーダ)と関連して使用できるようにし、従来の逐次プロセスを使ってそのような復号化プロセスを達成することは困難である。

【0131】

提案された発明において理解されるべきことは、数学的な性質のうち説明された要素がそれ自体重要ではないが、それらがハードウェア制約に被らされる情報(たとえばオーディオ信号またはビデオ信号)のフローに関係する程度までだけが重要であることである。

[表1] いわゆる「エンコーダ」モデルの表形式での表現を示す表である。

【0132】

10

20

30

【表 1】

C1		C2		C3	
		標準状態モデル		簡略化状態モデル	
状態 E_k	$\lfloor \text{low} S_k, \text{up} S_k \rfloor$	$P(0)$ 、対応する再分割間隔	$S_k=0$ 放出ビット 次の状態	$S_k=1$ 放出ビット 次の状態	$S_k=\text{MPS}$ 放出ビット 次の状態
0	$\lfloor 0, 4 \rfloor$	$0.63 \leq P(0)$	- 1	11 0	- 1
		$0.37 \leq P(0) < 0.63$	0 0	1 0	0 1
		$P(0) < 0.37$	00 0	- 2	- -
1	0, 3	$0.5 \leq P(0)$	0 0	10 0	0 0
		$P(0) < 0.5$	00 0	f 0	- -
2	$\lfloor 1, 4 \rfloor$	$0.5 \leq P(0)$	f 0	11 0	- -
		$P(0) < 0.5$	01 0	1 0	- -

Table 1

【表 2】いわゆる「デコーダ」モデルの表形式の表現を示す表である。

【 0 1 3 3 】

【表 2】

状態 E_n	状態変数	P(MPS) [low k n, up S k n [の対応する再分割]	K2		K3
			$U_h = 0$ 放出記号	$U_h = 1$ 放出記号	次の 状態
0	[low U_h , up U_h] : [0, 4]	$0.63 \leq P(\text{MPS})$ ([0, 3])	MPS, MPS	-	1
	[low S_{k_n} , up S_{k_n}] : [0, 4]	$0.37 < P(\text{MPS}) < 0.63$ ([0, 2])	MPS	LPS	0
1	[low U_h , up U_h] : [12, 4]	$0.63 \leq P(\text{MPS})$ ([0, 3])	MPS, LPS	LPS	0
	[low S_{k_n} , up S_{k_n}] : [0, 4]				

Table 2

【付録 1】 本明細書の記載において使用される数式を含む。

【付録 1】

【 0 1 3 4 】

$$P(C_{k+1}|C_k) = P(S_{k+1}|C_k) \quad (1)$$

$$\prod_{k=K_n+1}^{K_{n+1}} P(S_k|C_{k-1}) \quad (2)$$

$$\hat{X} = \arg \max_x P(X = x|Y). \quad (3)$$

10

$$P(X, K|Y) = \prod_{n=1}^N P(X_n, K_n|Y). \quad (4)$$

$$\begin{aligned} P(X_n, K_n|Y) &= \frac{P(X_n, K_n|Y_1^n) \cdot P(Y_{n+1}^N|X_n, K_n)}{P(Y_{n+1}^N)}, \\ &\propto P(X_n, K_n|Y_1^n) \cdot P(Y_{n+1}^N|X_n, K_n), \end{aligned} \quad (5)$$

20

$$\begin{aligned} P(X_n = x_n, K_n = k_n|Y_1^n) &= \sum_{(x_{n-1}, k_{n-1})} P(X_{n-1} = x_{n-1}, K_{n-1} = k_{n-1}|Y_1^{n-1}) \cdot \\ &\quad P(X_n = x_n, K_n = k_n|X_{n-1} = x_{n-1}, K_{n-1} = k_{n-1}) \cdot \\ &\quad P(U_n = u_{(x_{n-1}, k_{n-1})(x_n, k_n)}|Y_n). \end{aligned} \quad (6)$$

$$\begin{aligned} P(Y_{n+1}^N|X_n = x_n, K_n = k_n) &= \sum_{(x_{n+1}, k_{n+1})} P(Y_{n+2}^N|X_{n+1} = x_{n+1}, K_{n+1} = k_{n+1}) \cdot \\ &\quad P(X_{n+1} = x_{n+1}, K_{n+1} = k_{n+1}|X_n = x_n, K_n = k_n) \cdot \\ &\quad P(U_{n+1} = u_{(x_n, k_n)(x_{n+1}, k_{n+1})}|Y_{n+1}). \end{aligned} \quad (7)$$

30

$$\begin{aligned} P(S_1^K|Y_1^N) &= P(S_1^K) \cdot P(U_1^N|Y_1^N) \\ &\propto P(S_1^K) \cdot P(Y_1^N|U_1^N), \end{aligned} \quad (8)$$

$$\begin{aligned} P(S_1^k|Y_1^{N_k}) &\propto P(S_1^{k-1}|Y_1^{N_{k-1}}) \cdot P(S_k|S_{k-1}) \cdot \\ &\quad P(Y_{N_{k-1}+1}^{N_k}|Y_1^{N_{k-1}}, U_1^{N_k}), \end{aligned} \quad (9)$$

$$\mathbb{P}(S_1^k | Y_1^{N_k}) \propto \frac{\mathbb{P}(S_1^{k-1} | Y_1^{N_{k-1}}) \cdot \mathbb{P}(S_k | C_{k-1})}{\mathbb{P}(Y_{N_{k-1}+1}^{N_k} | U_{N_{k-1}+1}^{N_k})} \quad (10)$$

$$\mathbb{P}(S_1 = s_1 | Y_1^{N_1} = y_1^{N_1}) \propto \frac{\mathbb{P}(S_1 = s_1 | C_1) \cdot \mathbb{P}(Y_1^{N_1} = y_1^{N_1} | U_1^{N_1} = u_1^{N_1})}{\mathbb{P}(Y_1^{N_1} = y_1^{N_1} | U_1^{N_1} = u_1^{N_1})}, \quad (11)$$

$$\mathbb{P}(Y_{N_{k-1}+1}^{N_k} = y_{n_{k-1}+1}^{n_k} | U_{N_{k-1}+1}^{N_k} = u_{n_{k-1}+1}^{n_k}) = \begin{cases} \prod_{n=N_{k-1}+1}^{N_k} \mathbb{P}(Y_n = y_n | U_n = u_n) & \text{if } N_k > N_{k-1} \\ 1 & \text{sinon.} \end{cases} \quad (12)$$

10

$$\mathbb{P}(S_1^{K_n} | Y_1^n) \propto \mathbb{P}(S_1^{K_n}) \cdot \mathbb{P}(Y_1^n | U_1^n) \quad (13)$$

$$\mathbb{P}(S_1^{K_n}) = \mathbb{P}(S_1^{K_{n-1}}) \cdot \mathbb{P}(S_{K_{n-1}+1}^{K_n} | S_1^{K_{n-1}}), \quad (14)$$

$$\mathbb{P}(S_{K_{n-1}+1}^{K_n} | S_1^{K_{n-1}}) = \begin{cases} \prod_{k=K_{n-1}+1}^{K_n} \mathbb{P}(S_k | C_{k-1}) & \text{si } K_n > K_{n-1} \\ 1 & \text{sinon.} \end{cases} \quad (15)$$

20

$$\begin{aligned} \mathbb{P}(S_1^{K_n} | Y_1^n) &\propto \mathbb{P}(S_1^{K_{n-1}}) \cdot \mathbb{P}(Y_1^{n-1} | U_1^{n-1}) \cdot \mathbb{P}(Y_n | U_n) \cdot \prod_{k=K_{n-1}+1}^{K_n} \mathbb{P}(S_k | C_{k-1}) \\ &\propto \mathbb{P}(S_1^{K_{n-1}} | Y_1^{n-1}) \cdot \mathbb{P}(Y_n | U_n) \cdot \prod_{k=K_{n-1}+1}^{K_n} \mathbb{P}(S_k | C_{k-1}) \end{aligned} \quad (16)$$

30

$$\mathbb{P}(X_n = x_n, K_n = k_n | X_{n-1} = x_{n-1}, K_{n-1} = k_{n-1}) = \prod_{k=K_{n-1}+1}^{K_n} \mathbb{P}(S_k | C_{k-1}) \quad (17)$$

$$\mathbb{P}(Y_n | U_n) < (1-p)^n \quad (18)$$

$$W = (N_{kmax} - N_{kmin} + 1)w \quad (19)$$

$$W = (K_{nmax} - K_{nmin} + 1)w \quad (20)$$

40

$$\begin{aligned} \mathbb{P}(Y_i = 0 | U_i = 0) &= 1 - p \\ \mathbb{P}(Y_i = 1 | U_i = 0) &= p \\ \mathbb{P}(Y_i = 0 | U_i = 1) &= p \\ \mathbb{P}(Y_i = 1 | U_i = 1) &= 1 - p \end{aligned} \quad (21)$$

【図 1】本発明によるエンコーダおよびデコーダを備えた構成を示す図である。

【図 2】本発明によるエンコーダ内で実現可能な符号化方法の全体図である。

【図 3】本発明によるデコーダにおいて実現可能な復号方法の全体図を提供する図である。

。

【図 4 A】ソース記号とビットのシーケンスとの間の対応を確立するいわゆる同一の「ソース」モデルの 3 つの異なる表現を示す図である。

【図 4 B】ソース記号とビットのシーケンスとの間の対応を確立するいわゆる同一の「ソース」モデルの 3 つの異なる表現を示す図である。

【図 4 C】ソース記号とビットのシーケンスとの間の対応を確立するいわゆる同一の「ソース」モデルの 3 つの異なる表現を示す図である。

10

【図 5 A】図 4 A のいわゆる「ソース」モデルに基づいて記号のシーケンスとビットのシーケンスとの間での同じ対応の 2 つの異なる表現を示す図である。

【図 5 B】図 4 B のいわゆる「ソース」モデルに基づいて記号のシーケンスとビットのシーケンスとの間での同じ対応の 2 つの異なる表現を示す図である。

【図 6 A】本発明によるいわゆる「ソースエンコーダプロダクト」モデルの表形式の表現を示す図である。

【図 6 B】本発明による図 6 A のいわゆる「ソースエンコーダプロダクト」モデルのトレリス形式の表現を示す図である。

【図 7 A】表 1 のいわゆる「エンコーダ」モデルのトレリス形式の 2 つの部分的な表現を示す図である。

20

【図 7 B】表 1 のいわゆる「エンコーダ」モデルのトレリス形式の 2 つの部分的な表現を示す図である。

【図 8】本発明によるいわゆる「ソースエンコーダプロダクト」モデルの一般化された表現を示す図である。

【図 9 A】本発明によるいわゆる「ソースデコーダプロダクト」モデルの表形式の表現を示す図である。

【図 9 B】本発明による図 9 A のいわゆる「ソースデコーダプロダクト」モデルのトレリス形式の表現を示す図である。

【図 10】本発明のいわゆる「ソースデコーダプロダクト」モデルの一般化された表現を示す図である。

30

【図 11】本発明の第 1 の実施の形態における復号化方法を示す図である。

【図 12】本発明の第 2 の実施の形態における復号化方法を示す図である。

【図 13】図 11 または図 12 の復号化方法に基づくトレリス形式の表現を示す図である。

。

【図 14】本発明の第 3 の実施の形態における復号化方法を示す図である。

【図 15】本発明の第 4 の実施の形態における復号化方法を示す図である。

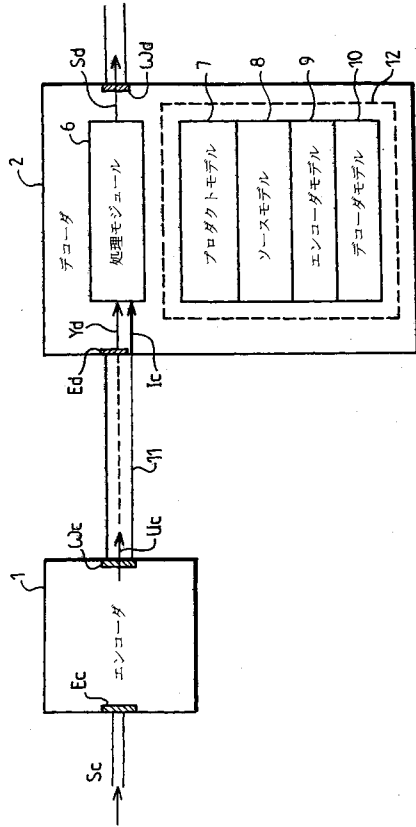
【符号の説明】

【0136】

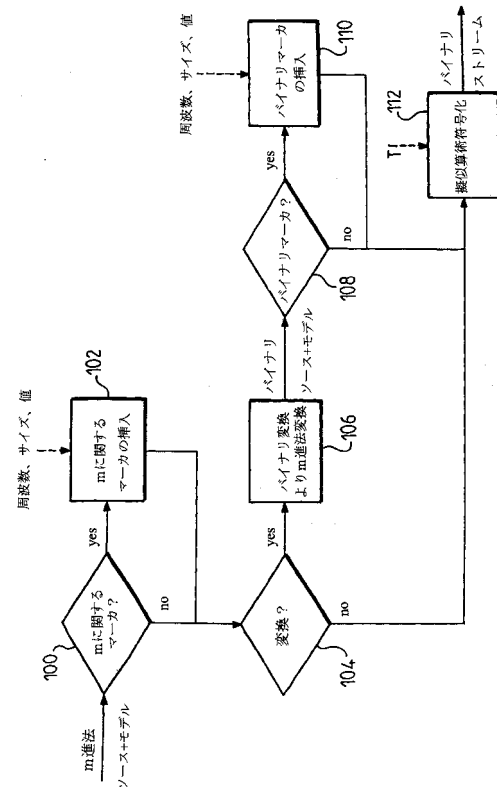
- 1 エンコーダ
- 2 デコーダ
- 6 処理モジュール
- 7 プロダクトモデル
- 8 ソースモジュール
- 9 エンコーダモデル
- 10 デコーダモデル
- 12 第 1 のモジュール (モデルモジュール)
- I c チャンネル特性

40

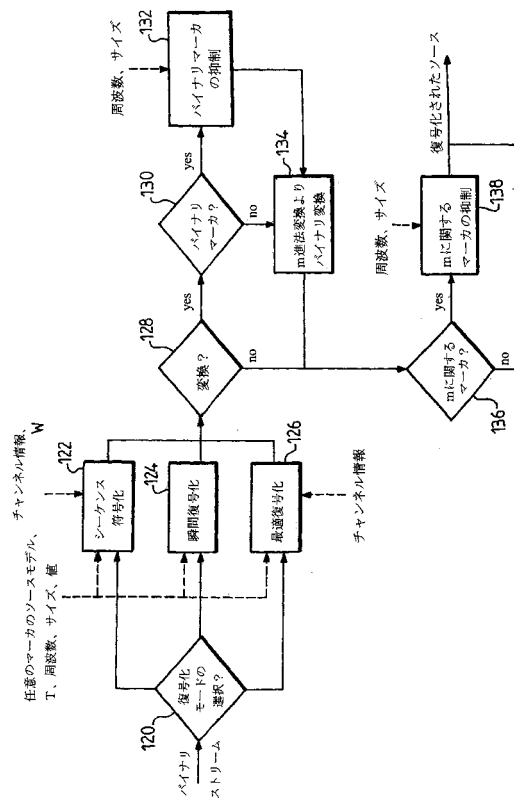
【図 1】



【図 2】



【図 3】



【図 4 A】

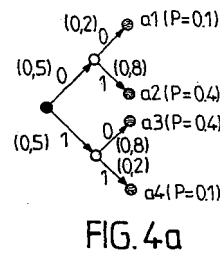


FIG. 4a

【図 4 B】

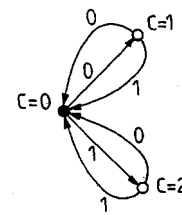


FIG. 4b

【図 4 C】

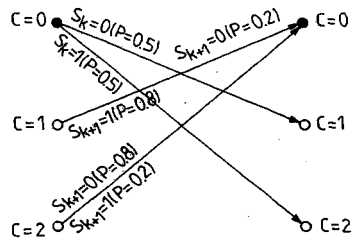


FIG. 4c

【図 5 B】

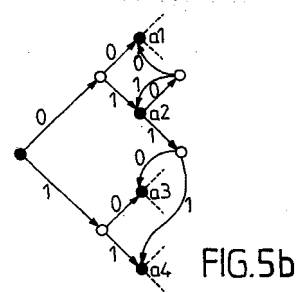


FIG. 5b

【図 5 A】

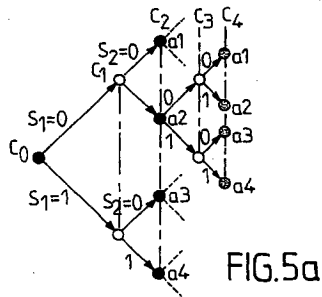


FIG. 5a

【図 6 A】

状態 X_k	状態変数	MPS	M4
0 初期 状態	{0,4] $C=0$	放出ビット: 0 次の状態: 1	放出ビット: 1 次の状態: 2
1	{0,4] $C=1$	放出ビット: 1 次の状態: 3	放出ビット: 1 次の状態: 0
2	{0,4] $C=2$	放出ビット: 1 次の状態: 3	放出ビット: 1 次の状態: 0
3	{0,3] $C=0$	放出ビット: 0 次の状態: 1	放出ビット: 0 次の状態: 2

FIG. 6A

【図 6 B】

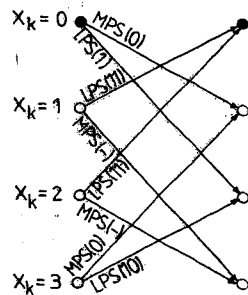


FIG. 6B

【図 7 B】

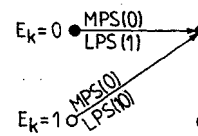


FIG. 7B

【図 7 A】

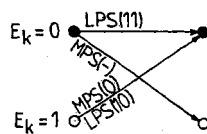


FIG. 7A

【図 8】

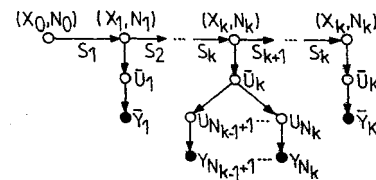


FIG. 8

【図 9 A】

状態 X_k	状態変数	$U_n = 0$	$U_n = 1$
0 初期 状態	$\begin{bmatrix} 0 & 4 \\ 1 & 0 & 4 \\ C=0 \end{bmatrix}$	放出記号: MPS 次の状態: 1	放出記号: LPS 次の状態: 2
1	$\begin{bmatrix} 0 & 4 \\ 1 & 0 & 4 \\ C=1 \end{bmatrix}$	放出記号: MPS, MPS 次の状態: 1	放出記号: - 次の状態: 3
2	$\begin{bmatrix} 0 & 4 \\ 1 & 0 & 4 \\ C=2 \end{bmatrix}$	放出記号: MPS, MPS 次の状態: 1	放出記号: - 次の状態: 4
3	$\begin{bmatrix} 2 & 4 \\ 1 & 0 & 4 \\ C=1 \end{bmatrix}$	放出記号: MPS, LPS 次の状態: 2	放出記号: LPS 次の状態: 0
4	$\begin{bmatrix} 2 & 4 \\ 1 & 0 & 4 \\ C=2 \end{bmatrix}$	放出記号: MPS, LPS 次の状態: 2	放出記号: LPS 次の状態: 0

FIG.9a

【図 10】

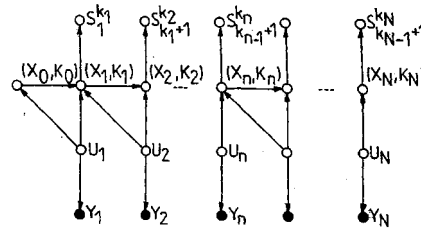


FIG.10

【図 9 B】

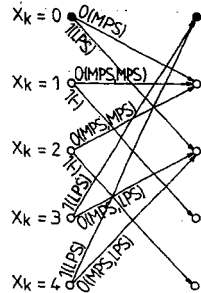


FIG.9b

【図 11】

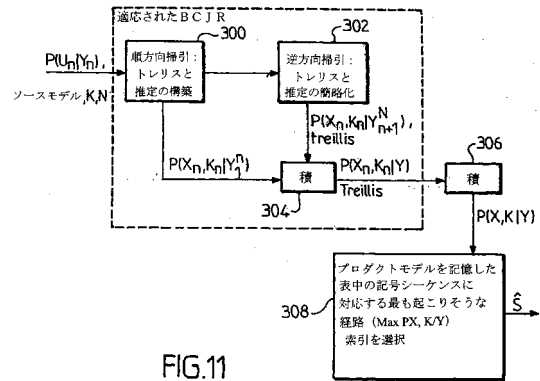


FIG.11

【図 12】

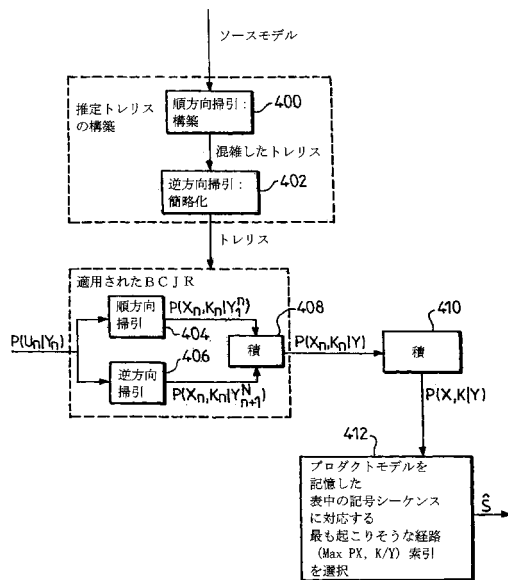


FIG.12

【図 13】

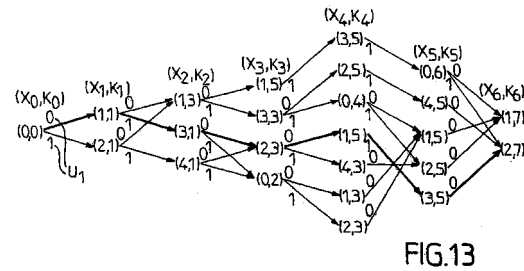


FIG.13

【図 14】

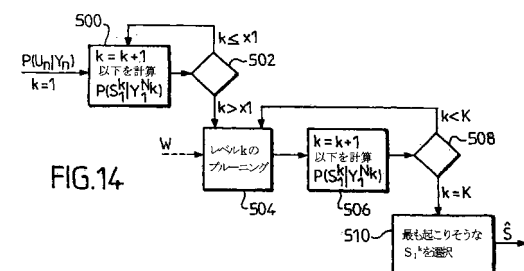
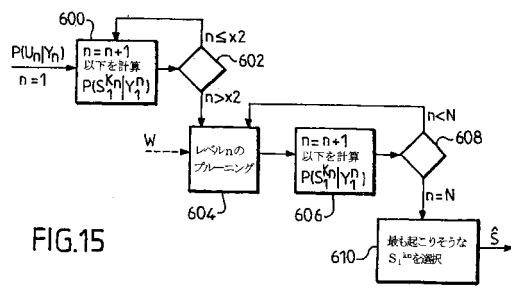


FIG.14

【図 15】



フロントページの続き

(74)代理人 100093023

弁理士 小塚 善高

(72)発明者 ギオネ・トマス

フランス国、F - 3 5 0 0 0 レンヌ、アヴェニュー・アリスティード・ブリエンド 8 1

(72)発明者 ギルモ・クリスティーヌ

フランス国、F - 3 5 1 3 5 シャンテピエ、アレ・フランソワーズ・ドルト 2

審査官 渡辺 未央子

(56)参考文献 特開 2 0 0 1 - 1 7 7 4 1 7 (J P , A)

国際公開第 0 1 / 3 5 5 3 5 (W O , A 1)

(58)調査した分野(Int.Cl. , D B 名)

H03M 7/40

H04N 7/26