

【特許請求の範囲】**【請求項 1】**

符号化器の初期状態と終了状態が同一の状態となるように情報ビット列を入力して畳み込み符号語を生成する巡回畳み込み符号化によって生成された符号について、ビタビ復号を行って前記情報ビット列を復号するビタビ復号方法であって、

加算比較選択演算において、トレリス図上の各状態についての生き残りパスが決まるところ、それらの生き残りパスの各々に關し、パスの尤度であるパスメトリックを記憶し、かつ、各生き残りパスの加算比較選択演算開始時点における状態を記憶し、この処理を加算比較選択演算の終了時点まで繰り返し行うステップと、

加算比較選択演算が終了すると、前記生き残りパスの各々について、前記加算比較選択演算の終了時点における状態と開始時点における状態の一致／不一致を判定するステップと、

前記判定するステップにおいて一致が確認された生き残りパスのみをトレースバック処理の候補とし、その候補の生き残りパスの少なくとも一つについて、トレースバック処理を実行するステップと、

を有するビタビ復号方法。

【請求項 2】

前記トレースバック処理の候補となる生き残りパスが複数ある場合には、まず、前記パスメトリックが最も小さいパスについてトレースバック処理を行い、そのトレーバック処理の結果として復号された符号ビット列について誤りの有無のチェックを行い、誤りがない場合には、その復号された符号ビット列を復号結果とし、誤りが確認された場合は、前記パスメトリックが2番目に小さい生き残りパスについてトレースバック処理を行い、誤りの無いビット列が得られるまで、同様の処理を繰り返す請求項1記載のビタビ復号方法。

【請求項 3】

トレースバック処理の候補となる生き残りパスが存在しない場合には、トレースバック処理を行わない請求項1記載のビタビ復号方法。

【請求項 4】

ビタビ復号を実行する復号化装置であって、

受信データ列からトレリス図における各ブランチの尤度を計算するブランチメトリック計算手段と、

各時点の各状態に至る複数のパスの尤度を比較し、尤度の最も高いパスを生き残りパスとして選択する加算比較選択演算手段と、

各生き残りパスのパスメトリックを記憶するパスメトリック記憶手段と、

各生き残りパスにおける、加算比較選択演算開始時点の状態を記憶する開始状態記憶手段と、

加算比較選択演算が終了すると、前記開始状態記憶手段に記憶されている各生き残りパスの開始状態を参照し、その生き残りパスの各々について、前記加算比較選択演算の終了時点における状態と開始時点における状態の一致／不一致を判定し、その一致が確認された生き残りパスのみをトレースバック処理の候補とし、その候補の生き残りパスの少なくとも一つについてトレースバック処理を実行し、復号データを得るトレースバック手段と、

前記復号データの誤りを検出する誤り検出手段と、
を備える復号化装置。

【請求項 5】

請求項4記載の復号化装置を搭載した移動局無線装置。

【請求項 6】

請求項4記載の復号化装置を搭載した移動通信の基地局無線装置。

【請求項 7】

請求項5記載の移動局無線装置または請求項6記載の基地局無線装置の少なくとも一方

10

20

30

40

50

を含んで構成される移動通信システム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、デジタル携帯電話等の分野に使用される、巡回畳み込み符号についてのビタビ復号方法、復号化装置、移動局無線装置、基地局無線装置および移動通信システムに関する。

【背景技術】

【0002】

ビタビ復号法は、畳み込み符号の復号方法の1つであり、特に移動通信システムでは多く採用されている方法である。 10

【0003】

ビタビ復号法を説明する前に、まず、畳み込み符号化について、図9を参照して説明する。畳み込み符号化には、通常の畳み込み符号化（線形畳み込み符号化）と、巡回畳み込み符号化がある。

【0004】

ここで、「巡回畳み込み符号化」は、例えば、移動通信分野における特定の制御信号の伝送に用いられている。本発明は、この「巡回畳み込み符号」の復号化技術に関する。 20

【0005】

ここで、「巡回畳み込み符号化」とは、初期状態と終了状態が同一の状態となるように情報ビット列Uを入力して符号語を生成する符号化をいい、巡回入力畳み込み符号化と呼ばれることがある。

【0006】

以下、順を追って説明する。

【0007】

図9は畳み込み符号化器の例であり、この符号化器の入力側に情報ビット列Uを入力することで、出力側に符号化率 $r = 1/2$ 、拘束長 $k = 3$ 、2個の生成多項式がそれぞれ $G_1(X) = X^2 + 1$ 及び $G_2(X) = X^2 + X + 1$ の畳み込み符号化が行われた符号化ビット列Xが得られる。 30

【0008】

畳み込み符号化器には、シフトレジスタSR1とSR2が従属接続され、SR2には情報ビット列Uが順次入力され、出力側では1ビットの情報ビット入力に対し、2ビットの符号化ビット出力(X1、X2)が得られる。

【0009】

このとき、シフトレジスタSR1、及びSR2のビット内容(SR1、SR2)を状態と呼び、図9の畳み込み符号化器では次の4つ(S0～S3)のいずれかの状態をとりうる。

$$S_0 = (0, 0)$$

$$S_1 = (0, 1)$$

$$S_2 = (1, 0)$$

$$S_3 = (1, 1)$$

【0010】

一般に、1ビット入力かつ符号化率 $r = 1/2$ の畳み込み符号化器の状態数は、拘束長kを用いて 2^{k-1} 状態となる。すなわち、状態数は拘束長に従って指数関数的に増大する。

【0011】

図9の符号化器の初期状態と終了状態はそれぞれ4状態考えられるが、初期状態と終了状態が同一の状態となるように情報ビット列Uを入力して符号語を生成するとき、これを「巡回畳み込み符号化」と呼ぶ。

【0012】

10

20

30

40

50

例えば、情報ビット列 U が n ビット ($U[0] \sim U[n-1]$) である場合、図 9 の符号化器の初期状態と終了状態を同一とするためには、開始時点 S において $S R_1$ に $U[n-2]$ 、 $S R_2$ に $U[n-1]$ を予めセットしておき、その後 $U[0] \sim U[n-1]$ を順次入力していくことにより巡回置み込み符号が得られる。

【0013】

また、現時点の符号化器の状態に対して、情報ビット (“0” または “1”) を 1 ビット入力すると、符号化ビット出力及び次時点の符号化器の状態が一意に定まる。この置み込み符号化器の生成規則を図 10 に示す。

【0014】

例えば、現時点の符号化器の状態が S_3 で、ビット “0” が入力された場合には、符号化ビット出力は “10” であり、次時点の符号化器の状態は S_2 となる。

【0015】

図 11 は図 9 の置み込み符号化器の生成規則の状態遷移をトレリス図によって示したものである。図 11 のトレリス図において、実線のブランチは入力の情報ビットが “0” であることを示し、点線のブランチは入力の情報ビットが “1” であることを示す。さらに、ブランチ部分には符号化ビット出力 (X_1 、 X_2) を示す。ただし、 $a = (0, 0)$ 、 $b = (0, 1)$ 、 $c = (1, 0)$ 、 $d = (1, 1)$ である。

【0016】

いま、例として、入力の情報ビット列 U が 6 ビットであり、 $U[0 : 5] = (1, 1, 0, 0, 1, 0)$ である場合の巡回置み込み符号化のトレリス図を図 12 に示す。

【0017】

図 12において、横方向は符号化器への入力タイミングの時点であり（時点 S は初期時点、時点 E は終了時点）、縦方向は符号化器の状態である。このとき、符号化器の時点 S での初期状態は ($U[4]$ 、 $U[5]$) であるから S_2 となっており、巡回置み込み符号化を行っていることから、時点 E での終了状態も S_2 となり、時点 S と時点 E は同一状態になる。また、図 12 の最下段が置み込み符号化結果の符号化ビット列 X であり、 $X[0 : 11] = (0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1)$ となる。

【0018】

このように置み込み符号化されたビット列を復号する方法のひとつにビタビ復号がある。ビタビ復号は、トレリス図上で出力されうるパターンと受信パターンとを時系列順に逐次比較し、より尤もらしいパターンを候補として残しつつ、最終的に誤りが最も少ないパターンを復号データとするアルゴリズムである。

【0019】

以下、巡回置み込み符号に対する従来のビタビ復号方法を図 13、図 14 および図 15 を参照しながら簡単に説明する。図 13 は、時点 t から時点 $t+1$ にかけて状態 S_i が状態 S_i および状態 S_j に状態遷移するときのトレリス図、図 14 は、従来のビタビ復号方法における ACS 处理の詳細説明図、図 15 は、従来のビタビ復号方法を適用するトレリス図の例である。

【0020】

ビタビ復号は、ACS (Add - Compare - Select : 加算 - 比較 - 選択) 処理とトレースバック処理の組み合わせによって行われる。

【0021】

ACS 处理では、パスの尤度（確からしさ）の指標としてパスメトリックが用いられる。パスメトリックは、各時点の各状態に至るまでのパスが経由するブランチのブランチメトリックの累積加算値であり、その値が小さいほどパスが確からしいと判定する。ブランチメトリックは、ノード間の状態遷移で期待される出力ビットと受信ビットとの誤りビット数（ハミング距離：硬判定）や、信号空間上の距離（軟判定）などが用いられる。

【0022】

まず、ACS 处理の説明の前に、以下の説明で用いる記号について図 13 を用いて定義する。図 13 は、時点 t から時点 $t+1$ にかけて状態 S_i が状態 S_i 及び状態 S_j に状態

10

20

30

40

50

遷移するときのトレリス図を示したものである。いま、時点 t における状態 S_i を状態 $S_i(t)$ と記述する。図 13 中の記号 P_M はパスメトリックであり、 $P_{Mi}(t)$ は、状態 $S_i(t)$ における生き残りパスのパスメトリックである。また、図 13 中の記号 B_M はプランチメトリックであり、 $B_{Mij}(t)$ は、状態 $S_i(t)$ が状態 $S_j(t+1)$ に状態遷移するときのプランチメトリックである。

【0023】

なお、パスメトリックとプランチメトリックには、 $P_{Mj}(t+1) = P_{Mi}(t) + B_{Mij}(t)$ なる関係が成り立つ。

【0024】

次に、ACS 处理について、図 14 及び図 15 を用いて説明する。図 14 及び図 15 の各ノード内の数値は各時点各状態におけるパスメトリックである。 10

【0025】

ここでは、受信データは巡回畳み込み符号化されており、且つ符号化器の初期状態は不明であることを前捏とし、プランチメトリックにはハミング距離を用いて説明する。また、受信ビット列 Y は、 $Y[0:11] = (0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1)$ とする。

【0026】

まず、時点 0 (= 時点 S) から時点 1 にかけて行われる ACS 处理について、図 14 を用いて説明する。時点 0 では、符号化器の初期状態が不明であるので、各状態のパスメトリック（尤度）は全て同一であり、 $P_{M0}(0) = P_{M1}(0) = P_{M2}(0) = P_{M3}(0) = 0$ である。 20

【0027】

はじめに、状態 $S_0(1)$ におけるパスメトリック $P_{M0}(1)$ を算出する。状態 $S_0(1)$ に遷移してくるパスは、状態 $S_0(0)$ 及び状態 $S_2(0)$ からのパスである。

【0028】

状態 $S_0(0)$ からのプランチメトリックは、符号化器からの期待出力が “00” であるのに対し、受信ビットが $Y[0:1] = “00”$ であるから、ハミング距離を用いて、 $B_{M00}(0) = 0$ となり、1 つ目の仮のパスメトリック P_{Ma} は、 $P_{Ma} = P_{M0}(0) + B_{M00}(0) = 0 + 0 = 0$ と記憶される。

【0029】

一方、状態 $S_2(0)$ からのプランチメトリックは、符号化器からの期待出力が “11” であるのに対し、受信ビットが “00” であるから、ハミング距離を用いて、 $B_{M20}(0) = 2$ となり、2 つ目の仮のパスメトリック P_{Mb} は、 $P_{Mb} = P_{M2}(0) + B_{M20}(0) = 0 + 2 = 2$ と記憶される。以上が ACS 处理の加算 (Add) である。 30

【0030】

次に、2 つの仮のパスメトリックの大きさを比較 (Compare) する。 P_{Ma} 及び P_{Mb} を比較すると、 P_{Ma} の方が小さいため、状態 $S_0(0)$ からのパスの尤度は状態 $S_2(0)$ からのパスの尤度より大きいため、状態 $S_0(1)$ では、状態 $S_0(0)$ からのパスを選択 (Select) し、 $P_{M0}(1) = P_{Ma} = 0$ となる。このとき、選択されたパスを生き残りパスと呼び、パス選択情報は各時点各状態の全てにおいて記憶しておく。図 14 において × 印が打たれたパスは破棄されたパスである。 40

【0031】

次に、状態 $S_1(1)$ におけるパスメトリック $P_{M1}(1)$ を算出する。上記と同様に考えることで、状態 $S_0(0)$ から至るパスのパスメトリック P_{Ma} 、及び状態 $S_2(0)$ から至るパスのパスメトリック P_{Mb} はそれぞれ、

$$P_{Ma} = P_{M0}(0) + B_{M01}(0) = 0 + 2 = 2 \quad (\text{符号化器の期待出力 “11”})$$

$P_{Mb} = P_{M2}(0) + B_{M21}(0) = 0 + 0 = 0 \quad (\text{符号化器の期待出力 “00”})$ となり、 P_{Mb} の方が小さいため、生き残りパスは $S_2(0)$ からのパスとなり、 $P_{M1}(1) = P_{Mb} = 0$ となる。

【0032】

同様に、状態 S₂(1)におけるパスメトリック PM₂(1)を算出する。上記と同様に考えることで、状態 S₁(0)から至るパスのパスメトリック PM_a、及び状態 S₃(0)から至るパスのパスメトリック PM_bはそれぞれ、

$$PM_a = PM_1(0) + BM_{12}(0) = 0 + 1 = 1 \text{ (符号化器の期待出力 "01")}$$

となる。このとき PM_a = PM_bとなるが、ここでは状態番号が小さい状態 S₁(0)から至るパスを生き残りパスとし、PM₂(1) = 1となる。なお、ここで生き残りパス選択方法は、ランダムに選択する方法も考えられる。

【0033】

同様に、状態 S₃(1)におけるパスメトリック PM₃(1)を算出する。上記と同様に考えることで、状態 S₁(0)から至るパスのパスメトリック PM_a、及び状態 S₃(0)から至るパスのパスメトリック PM_bはそれぞれ、

$$PM_a = PM_1(0) + BM_{13}(0) = 0 + 1 = 1 \text{ (符号化器の期待出力 "10")}$$

となる。このとき PM_a = PM_bとなるが、ここでは状態番号が小さい状態 S₁(0)から至るパスを生き残りパスとし、PM₃(1) = 1となる。なお、ここで生き残りパスの選択方法は、ランダムに選択する方法も考えられる。

【0034】

以上が、時点 0 (= 時点 S) から時点 1 にかけて行われる ACS 处理である。

【0035】

以下、時点 1 から時点 2 にかけての ACS 处理、時点 2 から時点 3 にかけての ACS 处理を順次実行していき、時点 5 から時点 6 (= 時点 E) にかけての ACS 处理までを終了すると、図 15 のようなトレリス図が得られる。図 15において太線が生き残りパスである。なお、パス選択情報は S_i(t) (i = 0 ~ 3, t = 1 ~ 6) の各時点各状態の全てにおいて記憶しておく。

【0036】

最終時点である時点 E まで ACS 处理が終了した後、トレースバックによって復号ビット列を得る。一般の畳み込み符号化では、送信側でテールビットと呼ばれる既知ビットを ((拘束長 k) - 1) ビットだけ入力することで、ACS 处理の最終時点における状態が 1 つに収束する。したがって、トレースバックの開始状態はテールビットパターンによって一意に決定されうる。

【0037】

一方、巡回畳み込み符号化では、最終時点 E における状態は不明であるが、初期時点 S と最終時点 E の状態番号が同一になるようにしている。すなわち、トレースバックの開始時点 (時点 E) とトレースバックの終了時点 (時点 S) における状態が同一でなければ、復号ビット列には誤りビットが必ず含まれることになる。

【0038】

(第 1 の従来例)

以下、第 1 の従来例について、巡回畳み込み符号の従来のトレースバックについて図 15 を参照しながら説明する。なお、説明においては、初期時点 (時点 S) は “ACS 演算開始時点” 及び “トレースバック終了時点” と同意であり、最終時点 (時点 E) は “ACS 演算終了時点” 及び “トレースバック開始時点” と同意である。

【0039】

巡回畳み込み符号のトレースバックで最終時点 E にて最もパスメトリックが小さい (尤度が高い) 状態を選択し、この状態から処理を開始する。図 15 では時点 E において状態 S₃ のパスメトリックが “1” であり、最も尤度が高いため、状態 S₃ からトレースバックを開始する。

【0040】

状態 S₃ から時点の逆順に (時点 E から時点 S に向かって) 生き残りパス (図 15 中の太線) を辿り、実線のブランチについては “0” 、破線のブランチについては “1” を復

10

20

30

40

50

号ビットとすることによって、復号ビット列 D [0 : 5] が得られる。しかしながら、状態 S 3 からトレースバックを行った場合には、トレースバックの終了時点（時点 S ）において状態 S 1 に到達することから、状態 S 3 状態 S 1 により、得られた復号ビット列には誤りが含まれる。

【 0 0 4 1 】

このため、トレースバックの開始状態を再度選択して、処理をやり直す。時点 E において状態 S 3 の次に尤度が高い状態は、状態 S 2 (パスマトリック “ 2 ”) であるので、2 回目のトレースバックは、状態 S 2 から開始する。状態 S 2 からトレースバックを行うと、トレースバックの終了時点（時点 S ）において状態 S 2 に到達することから、復号ビット列が正しいための必要条件が得られる。さらに、この復号ビット列に対して誤りの有無があるかどうかチェックを行い、誤りがなければこれを復号ビット列として採用する。復号ビット列の誤り判定は、送信側で冗長に付加されたパリティビットや C R C (Cyclic Redundancy Check) 等のチェックビットによって行えるようになっているのが一般的である。

【 0 0 4 2 】

上記で誤りがないと判定された場合、このビタビ復号方法では、トレースバック回数は 2 回となる。このケースではトレースバックは 2 回で済んだが、この第 1 の従来例のように尤度だけを頼りにトレースバックをリトライする方法では、トレースバックを何回も行わなければ正しい復号ビットが得られないことが多くなる。これは、巡回畳み込み符号のビタビ復号では、トレースバックを開始する状態の決め方が難しいという点に起因している。

【 0 0 4 3 】

いま、もしトレースバックの開始時点（時点 E ）と終了時点（時点 S ）において同一の状態になるパスが最初から存在しない場合には、状態数分のトレースバックを実行しなければならず、結果として正しいパスが存在しないということになってしまう。

【 0 0 4 4 】

また、トレースバックの回数の増加を回避するために、トレースバック回数を制限する方法をとると、正しいパス（復号ビット列）があるにも関わらず、トレースバックを行わないがために、正しい結果が破棄されてしまうという問題点がある。

【 0 0 4 5 】

（第 2 の従来例）

これに対し、特許文献 1 に開示されたビタビ復号方法は、1 回目のトレースバックを行い、復号ビット列に誤りがあった場合には、そのトレースバックの終了時点の状態を 2 回目のトレースバックの開始状態にし、再度トレースバックを行うことで、少ないトレースバック回数で正しい復号ビット列が得られるというものであった。

【 0 0 4 6 】

以下、第 2 の従来例について、図 15 のトレリス図を用いて説明する。時点 E において状態 S 3 のパスメトリックが “ 1 ” で最も小さい（尤度が高い）ため、状態 S 3 から 1 回目のトレースバックを開始する。この 1 回目のトレースバックでは、終了時点（時点 S ）において状態 S 1 に到達することから、状態 S 3 状態 S 1 により、得られた復号ビット列には誤りが含まれる。このため 2 回目のトレースバックを行う。

【 0 0 4 7 】

2 回目のトレースバックでは、1 回目のトレースバックの終了時点（時点 S ）の状態 S 1 を新しい開始状態としてトレースバックを行う。この 2 回目のトレースバックでは、終了時点（時点 S ）において状態 S 2 に到達することから、状態 S 1 状態 S 2 により、得られた復号ビット列には誤りが含まれる。このため 3 回目のトレースバックを行う。

【 0 0 4 8 】

3 回目のトレースバックでは、2 回目のトレースバックの終了時点（時点 S ）の状態 S 2 を新しい開始状態としてトレースバックを行う。この 3 回目のトレースバックでは、終了時点（時点 S ）において開始状態と同一の状態 S 2 に到達することから、復号ビット列

10

20

30

40

50

が正しいための必要条件が得られる。さらに、この復号ビット列に対して誤りの有無があるかどうかチェックを行い、誤りがなければこれを復号ビット列として採用する。

【0049】

上記のように、条件によっては、特許文献1に開示されたビタビ復号方法によっても、トレースバック回数が増えてしまうことがある。また、トレースバック回数の制限を行えば、正しいパス（復号ビット列）があるにも関わらず、トレースバックを行わないために、正しい結果が破棄されてしまうという場合がある。

【0050】

また、第1及び第2の従来例のビタビ復号方法では、トレースバック回数の増大によって、ハードウェア実現の場合には回路規模の増大してしまい、ソフトウェア実現の場合には演算量が増大してしまうため、結果として消費電流の増加を招いてしまう。

【0051】

また、巡回畳み込み符号化では、トレースバックの開始時点（時点E）と終了時点（時点S）における状態が異なる場合には、復号ビット列が正しいことはないが、従来例のビタビ復号方法では、そのような場合にもトレースバックを行ってしまうため、処理そのものが無駄になることが多い。

【特許文献1】特開平9-232971号公報

【発明の開示】

【発明が解決しようとする課題】

【0052】

上述のとおり、通常の畳み込み符号化（線形畳み込み符号化）では、テールビットの付加によって符号化開始時点および終了時点の状態を所望の状態（通常は、状態S0）に揃えることができ、よって、トレースバック処理の対象となる生き残りパスはただ一つである（すなわち、終了時点の状態S0からトレースバックを行うことで、符号を復元できる）。

【0053】

これに対し、巡回畳み込み符号化の場合は、入力符号列の周期性を確保する観点から、テールビットを付加することはできないため、符号化開始時点および終了時点の状態を一意に決めることができない。したがって、状態数がn個ある巡回符号化の場合、符号化が終了した時点では、各状態に対応したn個の生き残りパスが存在しており、そのn個の生き残りパスのうちのどのパスについてトレースバックをすればよいかが不明である。従来の復号化法は、結局のところ、試行錯誤的に一つの生き残りパスを選び、正しい復号結果が得られるまでトレースバックを繰り返すという無駄の多い処理となっている。

【0054】

すなわち、従来の復号化方法では、少ないトレースバック回数で正しい復号ビット列を得ることが難しい。

【0055】

また、トレースバック処理の回数の増加によってLSIの消費電流量が増大する。この処理を高速化しようとすると、回路規模や演算量が増え、さらに消費電流量が増大してしまう。

【0056】

移動通信分野では、通信規格において、特定の制御信号は巡回畳み込み符号化しなければならないことが決められている場合があり、このような特殊な制御信号の復号化のために回路規模や消費電力が増大することは、小型化、バッテリーの長寿命化、低コスト化が厳しく求められる移動通信機器においては、好ましいものではない。

【0057】

本発明は、このような考察に基づいてなされたものであり、巡回畳み込み符号について、少ないトレースバック回数で正しい復号ビット列を得ることを可能とすることを目的とする。

【課題を解決するための手段】

10

20

30

40

50

【 0 0 5 8 】

本発明のビタビ復号方法は、符号化器の初期状態と終了状態が同一の状態となるように情報ビット列を入力して畳み込み符号語を生成する巡回畳み込み符号化によって生成された符号について、ビタビ復号を行って前記情報ビット列を復号するビタビ復号方法であって、加算比較選択演算において、トレリス図上の各状態についての生き残りパスが決まるとき、それらの生き残りパスの各々に關し、パスの尤度であるパスメトリックを記憶し、かつ、各生き残りパスの加算比較選択演算開始時点における状態を記憶し、この処理を加算比較選択演算の終了時点まで繰り返し行うステップと、加算比較選択演算が終了すると、前記生き残りパスの各々について、前記加算比較選択演算の終了時点における状態と開始時点における状態の一致／不一致を判定するステップと、前記判定するステップにおいて一致が確認された生き残りパスのみをトレースバック処理の候補とし、その候補の生き残りパスの少なくとも一つについて、トレースバック処理を実行するステップと、を有する。
10

【 0 0 5 9 】

巡回畳み込み符号の復号化では、トレースバック開始時点では、複数の生き残りパスが存在し、どの状態からトレースバックを行えばよいかは決まっていないが、しかし、巡回畳み込み符号化の特徴である、「ACS演算の開始時点と終了時点の状態が同じである」という条件を適用すれば、それらの複数の生き残りパスの中から、その条件を満たさないものを排除し、トレースバックの候補となる生き残りパスを絞り込むことが可能となる。ただし、従来のように、トレースバックをしないとACS開始時点の状態が特定できないのでは、結局のところ、試行錯誤的に無駄にトレースバックを行う回数が増えるから、本発明では、トレースバックをしなくてもACS開始時点の状態がわかるようとする。すなわち、トレリス図上で生き残りパスが更新される毎に、そのパスの起点の状態(ACS開始時点の状態)を示す情報を、そのパスに対応付けて記憶する。そして、トレースバック開始時点において、ACS終了時点の各状態に対応した生き残りパスについて、記憶されている初期状態を参照し、終了時点の状態との一致／不一致を調べ、不一致のパスを、トレースバック対象から除外する。これにより、トレースバック対象のパスを絞り込むことができ、復号化処理を効率化することができる。
20

【 0 0 6 0 】

また、本発明のビタビ復号方法の一態様では、前記トレースバック処理の候補となる生き残りパスが複数ある場合には、まず、前記パスメトリックが最も小さいパスについてトレースバック処理を行い、そのトレーバック処理の結果として復号された符号ビット列について誤りの有無のチェックを行い、誤りが無い場合には、その復号された符号ビット列を復号結果とし、誤りが確認された場合は、前記パスメトリックが2番目に小さい生き残りパスについてトレースバック処理を行い、誤りの無いビット列が得られるまで、同様の処理を繰り返す。
30

【 0 0 6 1 】

トレースバック候補の生き残りパスが複数存在する場合には、パスメトリックを基準として、より確からしいパスを優先してトレースバックを行うものである。これにより、正しい復号データが早期に得られる確率を高めることができる。
40

【 0 0 6 2 】

また、本発明のビタビ復号方法の他の態様では、トレースバック処理の候補となる生き残りパスが存在しない場合には、トレースバック処理を行わない。

【 0 0 6 3 】

本発明では、トレースバックを行う前に、そのパスのACS演算開始時点の状態がわかる。そこで、終了時点の状態との一致判定の結果、一致が確認できた生き残りパスが一つもない場合には、いずれのパスも誤りを含むことは明らかであることから、トレースバックを行わず、復号処理を中止することにしたものである。これにより、無駄なトレースバック処理が繰り返される事態を回避できる。

【 0 0 6 4 】

また、本発明の復号化装置は、本発明のビタビ復号方法を実行する復号化装置であって、受信データ列からトレリス図における各ブランチの尤度を計算するブランチメトリック計算手段と、各時点の各状態に至る複数のパスの尤度を比較し、尤度の最も高いパスを生き残りパスとして選択するACS演算手段と、各生き残りパスのパスメトリックを記憶するパスメトリック記憶手段と、各生き残りパスにおける、ACS演算開始時点の状態を記憶する開始状態記憶手段と、ACS演算が終了すると、前記開始状態記憶手段に記憶されている各生き残りパスの開始状態を参照し、その生き残りパスの各々について、前記ACS演算の終了時点における状態と開始時点における状態の一致／不一致を判定し、その一致が確認された生き残りパスのみをトレースバック処理の候補とし、その候補の生き残りパスの少なくとも一つについてトレースバック処理を実行し、復号データを得るトレースバック手段と、前記復号データの誤りを検出する誤り検出手段と、を有する。

10

【0065】

トレースバック手段が、トレースバック処理を行う際に、開始状態記憶手段に記憶された、各生き残りパスのACS開始時点の状態を参照し、終了時点の一致／不一致を判定し、一致が確認されたパスのみをトレースバック候補とすることで、効率的な復号化を行うことができる。

【0066】

また、本発明の移動局無線装置、本発明の基地局無線装置は、本発明の復号化装置を搭載している。また、本発明の移動通信システムは、本発明の移動局無線装置または基地局無線装置の少なくとも一つを含むシステムである。

20

【0067】

本発明の復号化装置を利用することで、巡回畳み込み符号を効率的に復号化できる、小型、低消費電力の通信機器が実現される。

【発明の効果】

【0068】

本発明によれば、少ないトレースバック回数で正しい復号ビット列を得ることが可能なビタビ復号方法、復号化装置、移動局無線装置、基地局無線装置および移動通信システムを提供することができる。

【発明を実施するための最良の形態】

【0069】

以下、本発明の実施の形態を、図16を参照して説明する。まず、本発明の実施形態における復号化方法の特徴を、図16および図17を用いて説明する。図16は、対比例としての従来の復号方法の問題点を示す図であり、図17は、本発明の実施形態の復号方法の特徴を説明するための図である。図16および図17では、説明の便宜を考慮して、トレリス図における一つの生き残りパスを簡略化して描いてある。なお、図16および図17のトレリス図では、縦軸として2つの状態S0、S1が示され、横軸として時点1～時点4が示されている。

30

【0070】

図16に示すように、従来の復号方法によって巡回畳み込み符号の復号化を行う場合、トレースバックを行わないと、その生き残りパスの、時刻t1における状態が判明しない。

40

【0071】

図16の場合、時刻t1(ACS開始時点)の状態がS1であり、時刻t4(ACS終了時点)の状態はS0であり、不一致であるから、巡回畳み込み符号の条件を満たさない。すなわち、このパスは必ず誤りを含んでいる。したがって、トレースバックで得られた復号データは廃棄されることになり、結果的に、無駄なトレースバック処理が行われたことになる。

【0072】

一方、図17の本発明の実施形態の復号方法では、トレリス図上で、生き残りパスが更新される毎に、そのパスの時刻t1における状態を示す情報を、そのパスと関連付けして

50

保存していく。図17において、時刻t4の状態(S0)の下に示される記号[S1]は、保存されている時刻t1における状態を示している。

【0073】

図17では、トレースバックをしなくても、時刻t1における状態(S1)と時刻t4における状態(S0)が不一致であることが判明する。これにより、このパスには誤りが含まれることがわかり、無駄なトレースバックを回避することができる。つまり、このようなACS開始時点および終了時点における状態が不一致である生き残りパスは、トレースバックの対象から外し、一致が確認されたパスのみをトレースバックの候補として、トレースバックの対象となるパスの数を絞り込むことができる。そして、その候補のパスが複数ある場合には、パスメトリックがより小さなパス(より尤度が高いパス)を優先してトレースバックを行うことで、効率的に復号化処理を行うことができる。10

【0074】

このように、従来例では、「トレースバック処理が、ACS開始時点の状態を特定する手段としても使われていた」ため、無駄なトレースバック処理が発生する確率が高い。これに対し、本発明では、「ACS開始時点の状態を各生き残りパスと関連づけて保存する構成を採用した」ため、ACS開始時点の状態をトレースバック前に特定することができるようになり、これにより、無駄なトレースバックを回避できるようになり、復号化処理の大幅な効率化が達成される。20

【0075】

(第1の実施形態)

まず、図1～図6を参照しながら、本発明の第1の実施形態の説明をする。図1(a)は誤り訂正符号化装置のブロック図、図1(b)は本発明の実施形態の誤り訂正復号化装置のブロック図、図2は本発明の実施形態のビタビ復号方法におけるACS処理を詳細に説明するための図、図3は本発明の実施形態のビタビ復号方法におけるトレースバック処理の処理手順を示すフローチャートである。

【0076】

なお、トレースバック処理は、ハードウェアによる回路やソフトウェアによって実現することが可能である。

【0077】

また、図4は本実施形態のビタビ復号方法及び誤り訂正復号化回路を適用するトレリス図の例1、図5は本実施形態を適用するトレリス図の例2、図6は本実施形態を適用するトレリス図の例3をそれぞれ示す。30

【0078】

図1(a)において、参照符号101は誤り訂正符号化装置、102は誤り訂正復号化装置である。誤り訂正符号化装置101は、情報ビットにCRC符号等の誤り検出符号を付加する誤り検出符号付加回路103と、誤り検出符号を付加したビット列を巡回置み込み符号化して送信する誤り訂正符号化回路104を備える。

【0079】

また、図1(b)において、誤り訂正復号化装置102は、プランチメトリック計算回路106、パスメトリックメモリ107、加算器108、比較器109、開始状態メモリ110、選択器111、パスメモリ112およびトレースバック制御回路113(誤り検出回路114を内蔵する)を備える。なお、ACS回路105は加算器108、比較器109及び選択器111を有して構成されている。40

【0080】

開始状態メモリ110が設けられたことによって、ACS開始時点の状態を、各生き残りパスと関連づけて保存できるようになっている点が、この誤り訂正符号化装置102の大きな特徴である。

【0081】

プランチメトリック計算手段の機能を有するプランチメトリック計算回路106には、例えばプランチメトリックにハミング距離を用いた硬判定を採用する場合には、処理時点50

毎に受信データの1時点分のビットが入力される。

【0082】

いま、プランチメトリック計算回路106が上述の図9の符号化器に対応するものであれば、各処理時点毎に2ビット(X1、X2に対応)が入力される。プランチメトリック計算回路106では、現時点の受信ビットの内容に基づいてプランチメトリックが計算され、加算器108に入力される。

【0083】

なお、本実施形態ではハミング距離を用いた硬判定によって説明するが、プランチメトリック計算回路に軟判定情報を用いる場合には、プランチメトリック計算回路106には、処理時点毎に受信データ1時点分の軟判定情報が入力される。

10

【0084】

パスメトリックメモリ107には、各状態について、現時点までの生き残りパスのパスメトリックが格納されている。

【0085】

加算器108では、パスメトリックメモリ107から読み出した各状態の現時点までの生き残りパスのパスメトリックに、プランチメトリック計算回路106で計算されたプランチメトリックが加算され、次時点の各状態に至るパスメトリックが算出され、比較器109及び選択器111に入力される。

【0086】

なお、次時点のある状態に至るパスは複数存在するため、加算器108から出力されるパスメトリックは1時点1状態の演算ごとに複数存在する。

20

【0087】

比較器109では、各状態について、加算器108から入力された複数のパスメトリックが比較され、最も尤度の高いパスの選択信号が選択器111に出力されると共に、そのパスの選択信号がパスメモリ112に格納される。

【0088】

開始状態記憶手段の機能を有する開始状態メモリ110には、各状態について、現時点までの生き残りパスのACS演算開始時点(時点S)における状態番号が格納されている。

【0089】

生き残りパス選択手段の機能を有する選択器111では、各状態について、比較器109から入力されたパス選択信号に基づいて、加算器108から入力されたパスメトリックの中から最も尤度の高いパスが選択され、選択されたパスメトリックはパスメトリックメモリ107に出力され、パスメトリックメモリ107の内容は更新される。

30

【0090】

さらに、選択器111では、各状態について、比較器109から入力されたパス選択信号に基づいて、選択されたパスの現時点のACS演算開始時点(時点S)の状態番号が、開始状態メモリ110から読み出され、この状態番号は、次時点のACS演算開始時点(時点S)の状態番号とされ、開始状態メモリ110の内容が更新される。

【0091】

パスメモリ112には、上述のように、各時点でのパス選択信号が格納されている。

40

【0092】

ACS演算が終了し、全状態・全時点のパス選択信号がパスメモリ112に格納され、全状態のACS演算開始時点(時点S)の状態番号が開始状態メモリ110に格納された後、トレースバック手段であるトレースバック制御回路113の動作が開始され、復号ビット列が得られる。トレースバックは、開始状態メモリ110に格納された内容と、パスメモリ112に格納された内容に基づき行われる。

【0093】

次に、以上のような各機能部106～113からなる誤り訂正復号化装置の動作、すなわち本実施形態のビタビ復号方法について、ACS処理とトレースバック処理に分けて説

50

明する。

【0094】

ここでは、受信データは巡回置み込み符号化されており、且つ符号化器の初期状態は不明であることを前提とし、プランチメトリックにはハミング距離を用いて説明する。

【0095】

また、送信側の入力の情報ビット列Uは、 $U[0:5] = (1, 1, 0, 0, 1, 0)$ の6ビットとし（誤り訂正復号化装置102側では未知）、巡回置み込み符号化の方法（生成多項式等）は背景技術に示したものと同-とする。さらに受信ビット列Yは硬判定されており、 $Y[0:11] = (0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1)$ の12ビットとする。

10

【0096】

まず、時点0（=時点S）から時点1にかけて行われるACS処理について、図2のトレリス図を参照しながら説明する。

【0097】

図2の各ノード内の数値は各時点各状態におけるパスメトリックである。また、各ノード下の括弧内の数値*は、各時点各状態における開始状態メモリ110の内容であり、選択されたパスのACS演算開始時点（時点S）の状態番号を示している。

【0098】

時点0では、符号化器の初期状態が不明であるので、各状態の尤度は全て同-であり、 $PM_0(0) = PM_1(0) = PM_2(0) = PM_3(0) = 0$ である。また、状態iにおける開始状態メモリ110の内容をS、 $S[i]$ と表すと、時点0の開始状態番号は状態番号そのものであるから、 $SS[0] = 0, SS[1] = 1, SS[2] = 2, SS[3] = 3$ が開始状態メモリ110の初期値となる。

20

【0099】

以下、ACSの説明では、プランチメトリックの計算、パスメトリックの計算及び生き残りパスの選択の方法については、背景技術の説明で図14を用いて行っており全く同一であるため省略する。

【0100】

本実施形態が図14の背景技術と異なるのは、選択器111によって、具備された開始状態メモリ110の内容が更新される点である。

30

【0101】

時点1の状態S0においては、時点0（=時点S）で状態S0を開始状態とするパスを選択していることから、時点1における新たな開始状態メモリ110の内容は、 $SSR[0] = SS[0] = 0$ となる。

【0102】

時点1の状態S1においては、時点0（=時点S）で状態S2を開始状態とするパスを選択していることから、時点1における新たな開始状態メモリ110の内容は、 $SSR[1] = SS[2] = 2$ となる。

【0103】

時点1の状態S2においては、時点0（=時点S）で状態S1を開始状態とするパスを選択していることから、時点1における新たな開始状態メモリ110の内容は、 $SSR[2] = SS[1] = 1$ となる。

40

【0104】

時点1の状態S3においては、時点0（=時点S）で状態S1を開始状態とするパスを選択していることから、時点1における新たな開始状態メモリ110の内容は、 $SSR[3] = SS[1] = 1$ となる。

【0105】

このようにして、時点1の全ての状態S0～S3についての新たな開始状態メモリの内容を抽出した後に、開始状態メモリ110の内容SS[0:3]を更新する。すなわち、 $SS[0] = SSR[0] = 0, SS[1] = SSR[1] = 2, SS[2] = SSR[2] = 1, SS[3] = SSR[3] = 1$

50

$S[2] = 1$ 、 $SS[3] = SSR[3] = 1$ とする。

【0106】

以上が、時点0(=時点S)から時点1にかけて行われるACS処理である。

【0107】

以下、時点1から時点2にかけてのACS処理、時点2から時点3にかけてのACS処理を順次実行していき、時点5から時点6(=時点E)にかけてのACS処理までを終了すると、図4のようなトレリス図が得られる(ACS処理の比較(Compare)において、到達する2つのパスのパスメトリックが同じ値であるときには、状態番号がより小さい状態から至るパスを生き残りパスとした場合)。

【0108】

図4における太線の矢印は生き残りパスである。なお、バス選択情報は $Si(t)$ ($i = 0 \sim 3$ 、 $t = 1 \sim 6$)の各時点各状態の全てにおいて記憶しておく。また、ACS処理終了後の開始状態メモリ内容は、時点6(=時点E)を参照することで、 $SS[0] = 2$ 、 $SS[1] = 2$ 、 $SS[2] = 2$ 、 $SS[3] = 1$ となっている。

【0109】

最終時点である時点EまでACS処理が終了した後、トレースバック処理が開始され、トレースバック処理の結果として復号ビット列が得られる。巡回畳み込み符号化では、最終時点Eにおける状態は不明であるが、初期時点Sと最終時点Eの状態番号が同一になるようにしている。

【0110】

すなわち、少なくともトレースバック開始時点(時点E)とトレースバックの終了時点(時点S)における状態が同一でなければ、復号ビット列には誤りビットが必ず含まれることになる。

【0111】

従って、ACS処理が終了した後の各状態*i*に対する開始状態メモリ110の内容 $SS[i]$ について、 $SS[i] = i$ ($i = 0 \sim 3$)が満たされていない生き残りパスについては、トレースバックをするまでもなく誤ったパスであることがわかるため、トレースバックの必要はない。

【0112】

以下、トレースバック制御回路113におけるトレースバック処理について、図3のトレースバック処理の処理手順を示すフローチャートを用いて説明する。

【0113】

まず、各状態*i*について、開始状態 $SS[i]$ を調べる。 $SS[i] = i$ であれば、当該状態に至るバスの初期時点Sと最終時点Eの状態が同じであり、バスが正しいための必要条件を満たしている。そこで、状態*i*をトレースバック開始状態の候補とする(ステップS301)。

【0114】

次に、Nに上記で候補としたトレースバック開始状態の候補数をセットする(ステップS302)。

【0115】

もし、Nが0であれば、正しいバスが存在しないため、当該フレームはエラーとし、トレースバックを行わずに処理を終了する(ステップS303、ステップS309)。

【0116】

ステップS303でNが0以外であれば、ステップS304～ステップS307のkのループを実行する。

【0117】

まず、トレースバック開始状態の候補のN状態の中から、パスメトリックが最も小さい状態からトレースバックを開始する(ステップS304でk=1、ステップS305)。

【0118】

上記トレースバックによって得られたビット列に付加されている誤り検出符号(例えば

10

20

30

40

50

C R C 符号) を用いて、該ビット列が正しいかどうか調べる(ステップ S 3 0 6)。

【0 1 1 9】

ステップ S 3 0 6 で、該ビット列に誤りがないと判定された場合には、このトレースバックで得られたビット列を復号ビット列 D とし、処理を終了する(ステップ S 3 0 8)。

【0 1 2 0】

ステップ S 3 0 6 で、該ビット列に誤りがあると判定され、かつ $k < N$ の場合には、k のループの先頭に戻り、トレースバック開始状態の候補の N 状態の中から、パスメトリックが次に小さい状態からトレースバックを開始する(ステップ S 3 0 4 で $k = 2$ 、ステップ S 3 0 5)。

【0 1 2 1】

ステップ S 3 0 6 で誤りが存在する限りは、最大で N 回、ステップ S 3 0 4 ~ ステップ S 3 0 7 の k のループを実行する。

【0 1 2 2】

ステップ S 3 0 6 で、該ビット列に誤りがあると判定され、かつ $k = N$ の場合には、正しいパスが存在しないため、当該フレームはエラーとし(ステップ S 3 0 9)、処理を終了する。

【0 1 2 3】

以上が、トレースバック制御回路 1 1 3 におけるトレースバック処理であるが、これを図 4 のトレリス図(本発明の実施形態のビタピ復号方法及び誤り訂正復号化回路を適用するトレリス図の例 1)に当てはめ、図 3 のフローチャートを実行すると次のようになる。

【0 1 2 4】

まず、ステップ S 3 0 1 の処理では、最終時点 E の各状態 S 0 ~ S 3 について、ACS 一演算開始時点(時点 S)の開始状態 SS を調べる。時点 E で状態 S 0 に到達したパスの ACS 開始状態は、開始状態メモリ SS [0] = 2 を参照することで状態 S 2 であるが、S 0 ~ S 2 であり状態番号が異なるため、状態 S 0 はトレースバック開始の候補状態とはしない。

【0 1 2 5】

時点 E で状態 S 1 に到達したパスの ACS 開始状態は、開始状態メモリ SS [1] = 2 を参照することで状態 S 2 であるが、S 1 ~ S 2 であり状態番号が異なるため、状態 S 1 はトレースバック開始の候補状態とはしない。

【0 1 2 6】

時点 E で状態 S 2 に到達したパスの ACS 開始状態は、開始状態メモリ SS [2] = 2 を参照することで状態 S 2 であり、状態番号がともに S 2 で一致しているため、状態 S 2 をトレースバック開始の候補状態とする。

【0 1 2 7】

時点 E で状態 S 3 に到達したパスの ACS 開始状態は、開始状態メモリ SS [3] = 1 を参照することで状態 S 1 であるが、S 3 ~ S 1 であり状態番号が異なるため、状態 S 3 はトレースバック開始の候補状態とはしない。

【0 1 2 8】

上記よりトレースバック開始状態の候補は、状態 S 2 の 1 つであるから、トレースバック開始候補数は N = 1 となる(ステップ S 3 0 2)。

【0 1 2 9】

以下、ステップ S 3 0 3 以降を実行すると、トレースバック開始状態の候補は状態 S 2 のみであるため、状態 S 2 からトレースバックを行うと、6 ビットのビット列(1、1、0、0、1、0)が得られる。このビット列を、図 1 のトレースバック制御回路 1 1 3 に内蔵される誤り検出回路 1 1 4 が、誤り検出符号を用いてチェックし、誤りがなければ、この 6 ビットを復号ビット列 D とする。

【0 1 3 0】

実際、入力の情報ビット列 U は、U [0 : 5] = (1、1、0、0、1、0) であったので、この復号ビット列は正しいものとなっている。

10

20

30

40

50

【 0 1 3 1 】

以上の説明のように、図4のトレリス図に本実施形態を適用すると、トレースバックを行う回数は1回である。これに対し、背景技術で説明した第1の従来例（時点Eでパスメトリックが小さい状態から優先的にトレースバックする）ではトレースバック回数が2回、第2の従来例ではトレースバック回数が3回であったため、本発明の実施形態の方法ではトレースバック回数を減らすことができ、短時間で正しい復号ビット列を得ることが可能になることがわかる。

【 0 1 3 2 】

さて、図4のトレリス図は、ACS処理の比較（Compare）において、到達する2つのパスのパスメトリックが同じ値であるときには、状態番号がより小さい状態から至るパスを生き残りパスとした場合に得られるものであった。このように複数のパスの尤度（パスメトリック）が同じときの生き残りパスの選択方法としては、ランダムに選択する方法も考えられる。

【 0 1 3 3 】

図5のトレリス図（本実施形態のビタビ復号方法及び誤り訂正復号化回路を適用するトレリス図の例2）は、複数のパスの尤度が同じときの生き残りパスがランダムに選択された場合の例であり、図4と比較して時点5から時点6（時点E）に至る過程のみが異なっている。

【 0 1 3 4 】

すなわち、図5において、時点6状態S0に至る2つのパスにおいて、時点5状態S0から至るパスのパスメトリックは3、時点5状態S2から至るパスのパスメトリックも3であり同じ値であることから、生き残りパスがランダムに選択され、結果として時点5状態S2から至るパスが生き残りパスとなったものである（図4では、時点5状態S0から至るパスを生き残りパスとしている）。

【 0 1 3 5 】

この図5についても本発明の実施形態のトレースバック処理に当てはめ、図3のフローチャートを実行すると次のようになる。

【 0 1 3 6 】

まず、ステップS301の処理では、最終時点Eの各状態S0～S3について、ACS演算開始時点（時点S）の開始状態SSを調べる。時点Eで状態S0に到達したパスのACS開始状態は、開始状態メモリSS[0]=0を参照することで状態S0であり、状態番号がともにS0で一致しているため、状態S0をトレースバック開始の候補状態とする。

【 0 1 3 7 】

時点Eで状態S1に到達したパスのACS開始状態は、開始状態メモリSS[1]=2を参照することで状態S2であるが、S1 S2であり状態番号が異なるため、状態S1はトレースバック開始の候補状態とはしない。

【 0 1 3 8 】

時点Eで状態S2に到達したパスのACS開始状態は、開始状態メモリSS[2]=2を参照することで状態S2であり、状態番号がともにS2で一致しているため、状態S2をトレースバック開始の候補状態とする。

【 0 1 3 9 】

時点Eで状態S3に到達したパスのACS開始状態は、開始状態メモリSS[3]=1を参照することで状態S1であるが、S3 S1であり状態番号が異なるため、状態S3はトレースバック開始の候補状態とはしない。

【 0 1 4 0 】

上記よりトレースバック開始状態の候補は、状態S0と状態S2の2つであるから、トレースバック開始候補数はN=2となる（ステップS302）。

【 0 1 4 1 】

以下、ステップS303以降を実行する。

【0142】

ステップS305を初めて実行するとき、 $k = 1$ である。このときステップS305は、『トレースバック開始状態の候補の($N =$)2状態の中で、パスメトリックが($k =$)1番目小さい状態からトレースバックを開始する』となる。トレースバック開始状態の候補は状態S0と状態S2であるが、状態S0に至るパスのパスメトリックは“3”であるのに対し、状態S2に至るパスのパスメトリックは“2”であるから、よりパスメトリックの小さい状態S2からトレースバックを開始すればよい。

【0143】

状態S2からトレースバックを行うと、6ビットのビット列(1、1、0、0、1、0)が得られる。このビット列を誤り検出符号によってチェックし誤りがなければ、この6ビットを復号ビット列Dとする。実際、入力情報ビット列Uは、 $U[0:5] = (1, 1, 0, 0, 1, 0)$ であったので、この復号ビット列は正しいものとなっている。10

【0144】

以上の説明では、トレースバック回数は1回であったが、もし状態S0(状態S0に至るパスのパスメトリックは“3”)と状態S2(状態S2に至るパスのパスメトリックは“2”)のトレースバック開始候補のうち、パスメトリックが大きい状態S0からトレースバックを開始すると、トレースバック回数は2回となってしまい、回数が増えてしまう(状態S0に至るパスから得られるビット列は、ステップS306の誤り検出で誤りありと判定され、 $k = 2$ のステップS305で状態S2から2回目のトレースバックが行われる)。20

【0145】

このように、トレリス図上の生き残りパスについて、ACSの開始時点Sと終了時点Eの状態が同一であるパスが複数存在する場合には、ビタビ復号においてはパスメトリックが小さい(尤度が高い)パスの方が正しい確率が高いという性質を利用し、その状態に至るパスを優先的にトレースバックすることで、より短時間で正しい復号ビット列が得ることが可能となる。

【0146】

さらに、別の例として、受信ビット列Yが硬判定されており、 $Y[0:11] = (0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0)$ の12ビットが誤り訂正復号化装置102に入力され、トレリス図が図6(本発明の実施形態のビタビ復号方法及び誤り訂正復号回路を適用するトレリス図の例3)に示すものになった場合を考える(入力の情報ビット列Uや畳み込み符号の生成多項式等はこれまでの説明と同一である)。なお、図6はACS処理の比較(Compare)において、到達する2つのパスのパスメトリックが同じ値であるときには、状態番号がより小さい状態から至るパスを生き残りパスとした場合に得られるものである(図6は図4と比較して $Y[10:11]$ のみが違うため、時点5から時点6(時点E)に至る過程のみが異なっている)。30

【0147】

この図6についても本発明の実施形態のトレースバック処理に当てはめ、図3のフローチャートを実行すると次のようになる。

【0148】

まず、ステップS301の処理では、最終時点Eの各状態S0～S3について、ACS演算開始時点(時点S)の開始状態SSを調べる。時点Eで状態S0に到達したパスのACS開始状態は、開始状態メモリSS[0] = 2を参照することで状態S2であるが、S0 S2であり状態番号が異なるため、状態S0はトレースバック開始の候補状態とはしない。40

【0149】

時点Eで状態S1に到達したパスのACS開始状態は、開始状態メモリSS[1] = 2を参照することで状態S2であるが、S1 S2であり状態番号が異なるため、状態S1はトレースバック開始の候補状態とはしない。

【0150】

時点 E で状態 S₂ に到達したパスの ACS 開始状態は、開始状態メモリ SS[2] = 1 を参照することで状態 S₁ であるが、S₂ S₁ であり状態番号が異なるため、状態 S₂ はトレースバック開始の候補状態とはしない。

【0151】

時点 E で状態 S₃ に到達したパスの ACS 開始状態は、開始状態メモリ SS[3] = 2 を参照することで状態 S₂ であるが、S₃ S₂ であり状態番号が異なるため、状態 S₃ はトレースバック開始の候補状態とはしない。

【0152】

上記よりトレースバック開始状態の候補は、ひとつも存在しないため、トレースバック開始候補数は N = 0 となる（ステップ S₃ 0 2）。

10

【0153】

ステップ S₃ 0 3 では、N = 0 のため、ステップ S₃ 0 9 に移行する。ステップ S₃ 0 9 では、正しい生き残りパスが存在しないとして当該フレームをフレームエラーと判定し、トレースバックを終了する。

【0154】

以上の説明のとおり、図 6 のトレリス図に、本実施形態を適用すると、トレースバックを行う回数は 0 回であり、トレースバックを行わなくとも当該受信ビット列は誤り訂正復号が不可能であることが判定できるといった効果がある。

【0155】

これに対し、背景技術で説明した第 1 の従来例を図 6 のトレリス図に適用した場合、時点 E において状態 S₂ のパスメトリックが “1” で最も小さい（尤度が高い）ため、状態 S₂ からトレースバックをするが、この結果得られるビット列は誤りである。

20

【0156】

以下、他のどの状態からトレースバックを開始しても、得られるビット列は全て誤りとなり、処理が無駄になってしまう。

【0157】

また、背景技術で説明した第 2 の従来例を図 6 のトレリス図に適用した場合を考える。

【0158】

図 6 の時点 E において状態 S₂ のパスメトリックが “1” で最も小さい（尤度が高い）ため、状態 S₂ から 1 回目のトレースバックを開始する。この 1 回目のトレースバックでは、終了時点（時点 S）において状態 S₁ に到達することから、状態 S₂ 状態 S₁ により、得られた復号ビット列には誤りが含まれる。このため 2 回目のトレースバックを行う。

30

【0159】

2 回目のトレースバックでは、1 回目のトレースバックの終了時点（時点 S）の状態 S₁ を新しい開始状態としてトレースバックを行う。この 2 回目のトレースバックでは、終了時点（時点 S）において状態 S₂ に到達することから、状態 S₁ 状態 S₂ により、得られた復号ビット列には誤りが含まれる。

【0160】

以後、同様にトレースバックを続けても正しいパスは見付からず、トレースバックの開始状態が、状態 S₂ 状態 S₁ 状態 S₂ 状態 S₁ … を繰り返すだけとなってしまい、処理は無駄となってしまう。

40

【0161】

これを回避するために従来の技術 2 では、トレースバック回数の制限を加えているが、もし正しいパスが存在した場合においては、正しいパス（復号ビット列）があるにも関わらず、トレースバックを行わないがために、正しい結果が破棄されてしまう場合がある。

【0162】

巡回置み込み符号化では、トレースバックの開始時点（時点 E）と終了時点（時点 S）における状態が異なる場合には、復号ビット列が正しいことはないが、上述の通り従来例のビタビ復号方法では、そのような場合にもトレースバックを行ってしまうため、処理その

50

ものが無駄になることが多い。

【0163】

以上に説明したように、本発明の実施形態のビタビ復号方法及び誤り訂正復号化装置では、巡回畳み込み符号化の性質を利用し、ACS演算終了時点（時点E）において、全ての状態の生き残りパスについてACS演算開始時点（時点S）における開始状態を記憶していることから、巡回畳み込み符号によって得られるパスの性質である（符号器の初期時点の状態）=（符号器の最終時点の状態）を利用することができ、トレースバックを行わずにパス（復号ビット列）の誤りを検出可能となるため、トレースバック回数を大きく減らすことができるという利点がある。また少ないトレースバック回数で正しい復号ビット列が得られる可能性が高くなるという効果がある。

10

【0164】

さらに、本発明の実施形態のビタビ復号方法及び誤り訂正復号化装置では、トレースバックを行う回数を減らすことができるため、ハードウェア実現の場合には回路規模を小さく実現でき、またソフトウェア実現の場合には演算量の低減が図られるため、消費電流を小さくできるという効果を得ることができる。

【0165】

（第2の実施の形態）

次に、本発明の実施形態の移動局無線装置、基地局無線装置および移動通信システムについて、図7及び図8を参照しながら説明する。

【0166】

無線通信システムでは、基地局無線装置と移動局無線装置を結ぶ電波の伝搬路状況が劣悪となり、受信側において、送信ビット列を正しく復号することが困難になる状況が多々ある。

20

【0167】

このような劣悪な伝搬路状況においては、基地局無線装置と移動局無線装置との同期がとれているにも関わらず、正しい復号結果（復号ビット列）を得にくくなる。

【0168】

上記状況下においては、誤り訂正符号化方式に巡回畳み込み符号化を採用する無線通信システムでは、誤り訂正復号のビタビ復号処理にて、トレースバックを開始する状態を確定することが困難となり、従来の技術においては、正しいパスを抽出するのに時間がかかるため、或いは正しいパスが存在しないにも関わらずトレースバックを行ってしまうため、基地局無線装置、移動局無線装置および移動通信システム全体としての消費電流が増大してしまう。

30

【0169】

第2の実施形態ではこのような問題点を解決するような移動局無線装置、基地局無線装置および移動通信システムについて説明するものである。

【0170】

図7は第2の実施形態の移動局無線装置および基地局無線装置のブロック図である。

【0171】

まず、図7における送信系について説明する。送信系は図7における符号701～708に相当する。

40

【0172】

送信系では、情報ビット列Uは誤り訂正符号化装置701に入力され、巡回畳み込み符号化が行われ、符号化ビット列が出力される。誤り訂正符号化装置701は、図1における誤り訂正符号化装置101に相当するものである。なお、畳み込み符号化は、情報ビット列Uのうち重要度の高いビットにのみ行われることが多い。

【0173】

誤り訂正符号化装置701から出力された符号化ビット列は、波形生成器702に入力され、制御データが付加され、変調データが生成される。

【0174】

50

D / A 変換器 703 では、上記変調データがデジタルからアナログに変換され、さらに直交変調器 704 では、アナログに変換されたデータが直交変調される。

【0175】

直交変調器 704 で直交変調された信号は、所定の送信周波数に変換するミキサ 705 によってアップコンバートされ、アップコンバートされた信号は増幅器 706 によって増幅される。

【0176】

増幅された信号は、送受信信号を分岐する共用器 707 に入力され、送受信共用アンテナ 708 から無線信号として送信される。

【0177】

次に、図 7 における受信系について説明する。受信系は図 7 における符号 707 ~ 714 に相当する。

【0178】

受信系では、送受信共用アンテナ 708 及び共用器 707 を通じて受信された受信信号は、ミキサ 709 によってダウンコンバートされ、さらに直交復調器 710 によって直交復調される。

【0179】

直交復調されたアナログ信号は、A / D 変換器 711 によってデジタル信号に変換され、同期部 712 及びデジタル検波器 713 に入力される。

【0180】

同期部 712 では、上記デジタル信号を用いて同期獲得が行われ、シンボルタイミング信号がデジタル検波器 713 に入力される。

【0181】

デジタル検波器 713 では、上記デジタル信号とシンボルタイミング信号を用いてデジタル検波（例えば変調方式が 1/4 シフト QPSK 方式であれば遅延検波）が行われ、受信データが出力される。

【0182】

受信データは、例えば、誤り訂正復号化装置 714 で硬判定によるビタビ復号を行う場合にはビット列となり、誤り訂正復号化装置 714 で軟判定によるビタビ復号を行う場合には信号空間上の I、Q 値列となる。

【0183】

上記受信データはさらに誤り訂正復号化装置 714 に入力され、誤り訂正復号され、復号ビット列 D が得られる。

【0184】

誤り訂正復号化装置 714 は、図 1 における誤り訂正復号化装置 102 に相当するものであり、本発明の実施形態のビタビ復号方法や誤り訂正復号化装置が含まれたものである。

【0185】

誤り訂正復号化装置 714 からは、同期部 712 及びデジタル検波器 713 へフレーム誤り情報が提供される。これによって、フレームが誤った場合には、いち早く次処理に移行することが可能となる。

【0186】

例えば、移動局無線装置でフレーム誤りが一定回数続いた場合には、デジタル検波器 713 をすぐにアイドル状態にして消費電流を低減させ、また同期部 712 にて即座に初期同期獲得を開始することが可能となる。

【0187】

このように構成された本発明の実施形態の移動局無線装置および基地局無線装置によれば、誤り訂正復号化装置 714 において、第 1 の実施形態に記載した通りの効果があり、すなわちビタビ復号処理におけるトレースバック回数を大きく減らすことができ、また少ないトレースバック回数で正しい復号ビット列が得られることから、受信性能の良好な移

10

20

30

40

50

動局無線装置および基地局無線装置が得られるという効果がある。

【0188】

さらに、本発明の実施形態の移動局無線装置および基地局無線装置では、誤り訂正復号化装置714において、トレースバックを行う回数を減らすことができるため、ハードウェア実現の場合には回路規模を小さく実現でき、またソフトウェア実現の場合には演算量の低減が図られるため、消費電流の小さい装置が実現できるという効果がある。また、少ないトレースバック回数でフレーム誤りが判定可能であるため、いち早く次処理に移行でき、消費電流が小さい装置を構成できるという効果がある。

【0189】

また、上記の移動局無線装置もしくは基地局無線装置の少なくとも一方を含むような、本発明の実施形態の移動通信システムの構成図を図8に示す。図8では、移動局無線装置MS1、MS2、…、MSmおよび基地局無線装置BSのうち、少なくともひとつに図7で構成された装置が含まれているものとする。

【0190】

図8の移動通信システムは、図7に示した基地局無線装置および移動局無線装置が複数含まれるように構成されているため、システム全体として受信性能の良好なシステムを構築できるという効果がある。また同様に、消費電流が低減されたシステムを構築できるという効果がある。

【0191】

以上説明したように、本発明の実施形態によれば、ACS演算終了時点において、全ての状態の生き残りパスについてACS演算開始時点における開始状態を記憶し、巡回畳み込み符号のパスの性質である（符号器の初期時点の状態）＝（符号器の最終時点の状態）を利用して一致判定を行うことで、トレースバックを行わずして、そのパス（復号ビット列）が誤りを含むパスであることを検出することができ、したがって、そのようなパスをトレースバックの対象から除外することができる。したがって、トレースバック対象のパスを絞り込むことができ、トレースバック回数を大きく減らすことができる。

【0192】

また、本発明の実施形態によれば、少ないトレースバック回数で正しい復号ビット列が得られる可能性が非常に高くなり、効率的、かつ迅速な復号化を行うことができる。

また、本発明によれば、無駄なトレースバック処理（演算処理）がなくなり、したがって、移動体通信機器における回路の消費電力や回路規模の増大を抑制することができる。このことは、良好な移動通信システムの構築に役立つものである。

【産業上の利用可能性】

【0193】

本発明は、巡回畳み込み符号の復号化処理を効率化するという効果を奏し、したがって、デジタル移動無線通信システムをはじめとして、誤り訂正符号化に巡回畳み込み符号化を採用する通信システム全般に適用可能である。また、各種記憶装置、記憶媒体においても、誤り訂正符号に巡回畳み込み符号化が採用されていれば、全て適用可能である。

【図面の簡単な説明】

【0194】

【図1】(a)は、誤り訂正符号化装置の構成を示すブロック図、(b)本発明の実施形態の誤り訂正復号化装置の構成を示すブロック図

【図2】本発明の実施形態のビタビ復号方法におけるACS処理について詳細に説明するための図

【図3】本発明の実施形態のビタビ復号方法におけるトレースバック処理の処理手順を示すフローチャート

【図4】本発明の実施形態のビタビ復号方法及び誤り訂正復号化回路を適用するトレリス図の例1を示す図

【図5】本発明の実施形態のビタビ復号方法及び誤り訂正復号化回路を適用するトレリス図の例2を示す図

10

20

30

40

50

【図6】本発明の実施形態のビタビ復号方法及び誤り訂正復号化回路を適用するトレリス図の例3を示す図

【図7】本発明の実施形態の移動局無線装置および基地局無線装置の構成を示すブロック図

【図8】本発明の実施形態の移動通信システムの構成を示す図

【図9】畳み込み符号化器の構成例を示す図

【図10】図9の畳み込み符号化器の生成規則を示す図

【図11】図9の畳み込み符号化器の生成規則の状態遷移のトレリス図を示す図

【図12】図9の畳み込み符号化器を用いた巡回畳み込み符号化のトレリス図の例を示す図

【図13】時点 t から時点 $t+1$ にかけて状態 S_i が状態 S_i 及び状態 S_j に状態遷移するときのトレリス図

【図14】従来の技術のビタビ復号方法におけるACS処理について詳細に説明するための図

【図15】従来の技術のビタビ復号方法を適用するトレリス図の例を示す図

【図16】対比例としての従来の復号方法の問題点を示す図

【図17】本発明の実施形態の復号方法の特徴を説明するための図

【符号の説明】

【0195】

101 誤り訂正符号化装置

20

102 誤り訂正復号化装置

103 誤り検出符号付加回路

104 誤り訂正符号化回路

105 ACS回路

106 ブランチメトリック計算回路

107 パスマトリックメモリ

108 加算器

109 比較器

110 開始状態メモリ

111 選択器

30

112 パスマモリ

113 トレースバック制御回路

114 誤り検出回路

701 誤り訂正符号化装置

702 波形整形器

703 D/A変換器

704 直交変調器

705 ミキサ

706 増幅器

707 共用器

40

708 送受信共用アンテナ

709 ミキサ

710 直交復調器

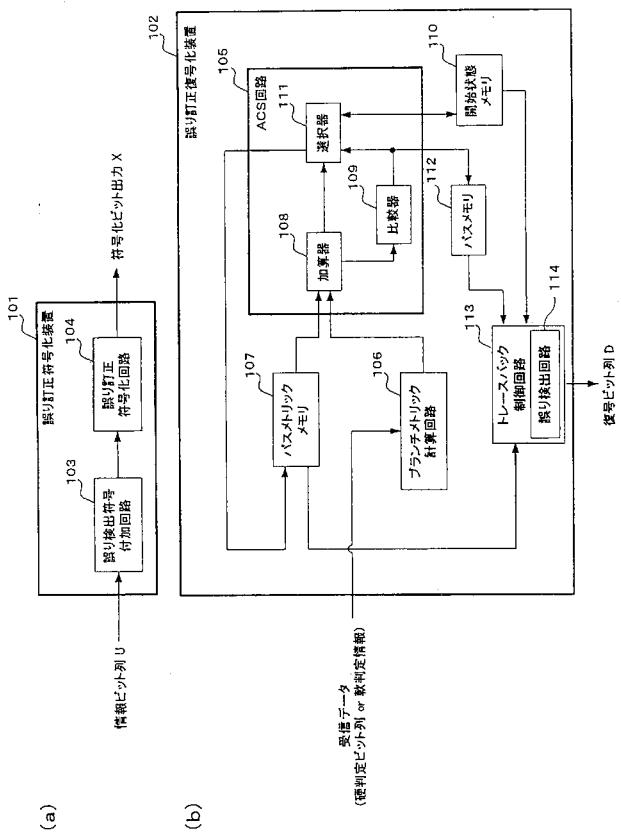
711 A/D変換器

712 同期部

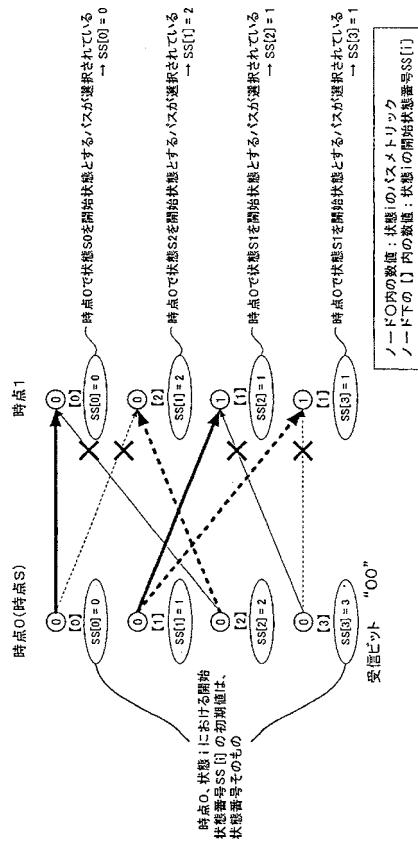
713 デジタル検波器

714 誤り訂正復号化装置

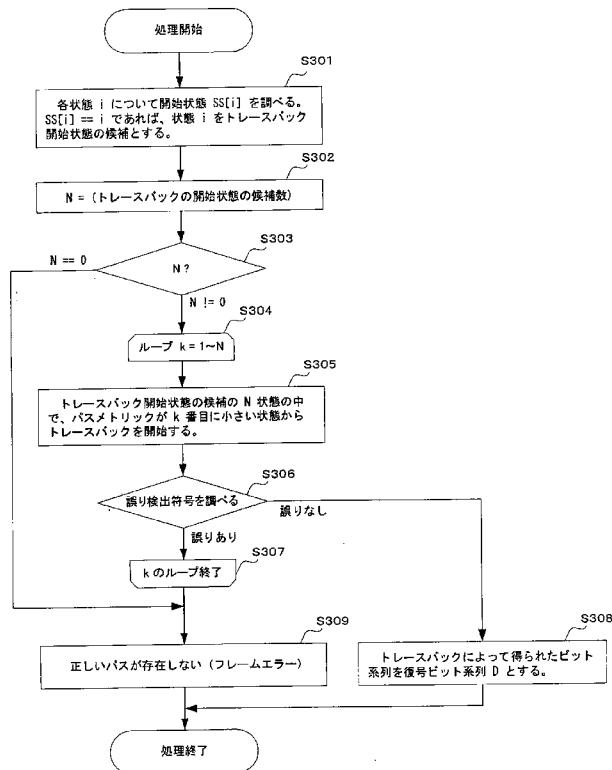
【図1】



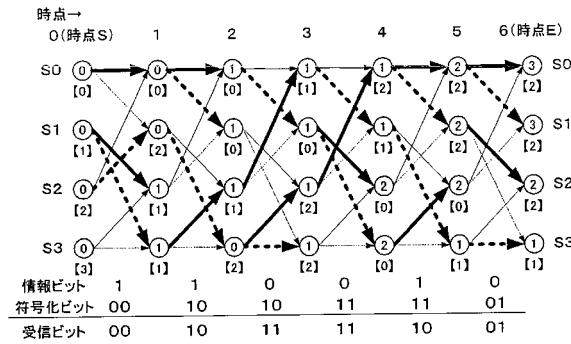
【図2】



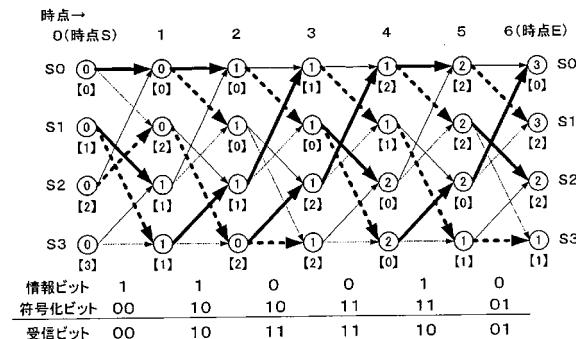
【図3】



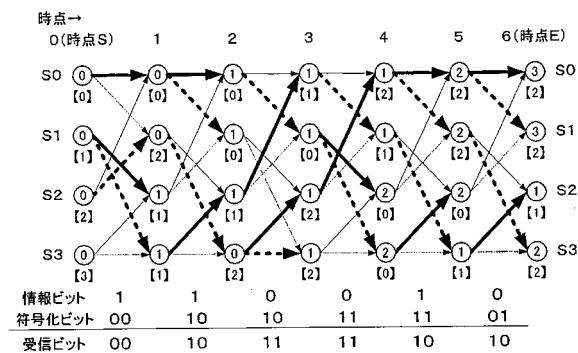
【図4】



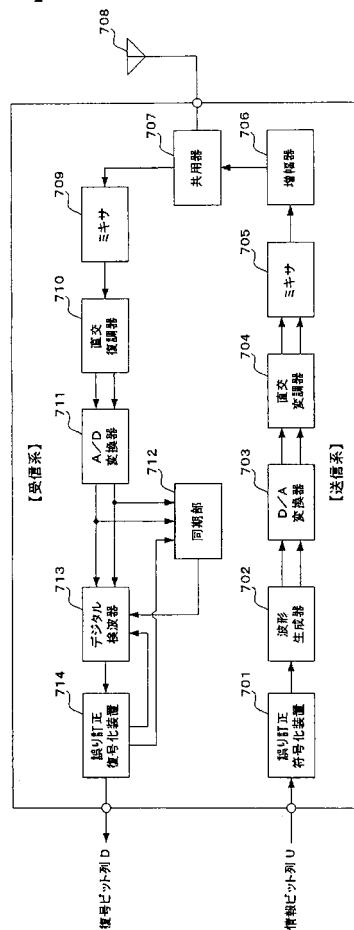
【図5】



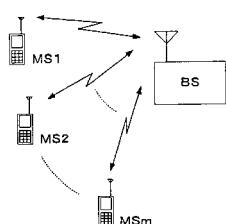
【図6】



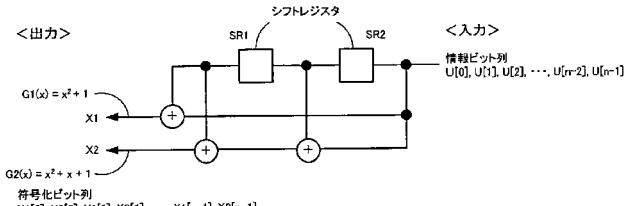
【図7】



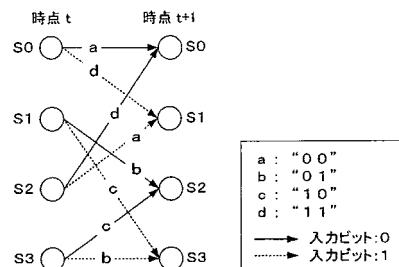
【図8】



【図9】



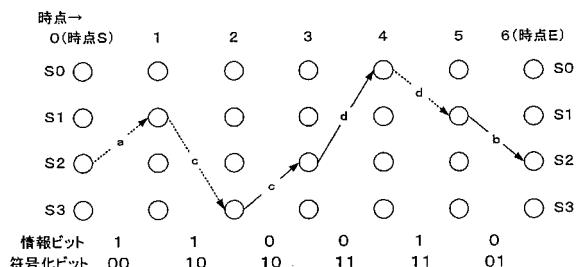
【図11】



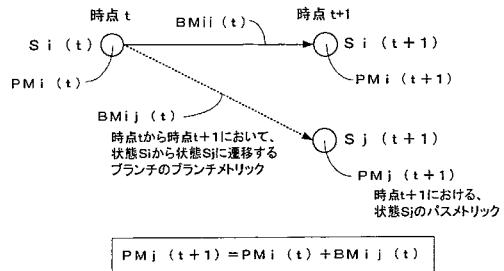
【図10】

現時点の状態	入力ビット	出力ビット (X1, X2)	次時点の状態
S 0	0	(0, 0) = a	S 0
	1	(1, 1) = d	S 1
S 1	0	(0, 1) = b	S 2
	1	(1, 0) = c	S 3
S 2	0	(1, 1) = d	S 0
	1	(0, 0) = a	S 1
S 3	0	(1, 0) = c	S 2
	1	(0, 1) = b	S 3

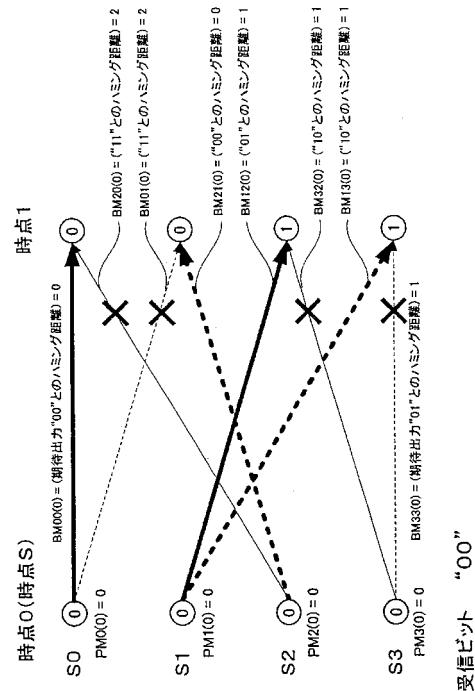
【図12】



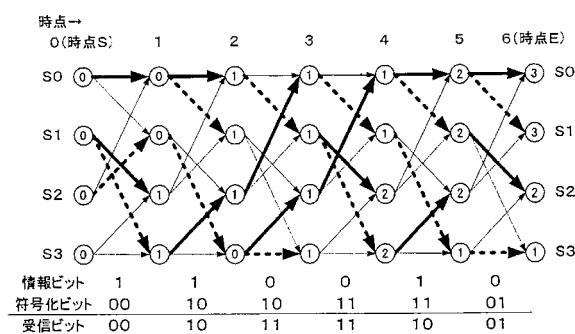
【図13】



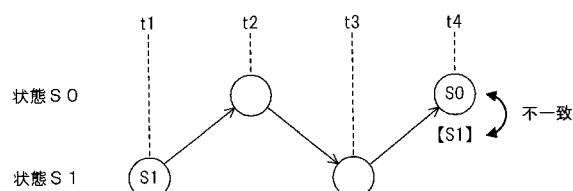
【図14】



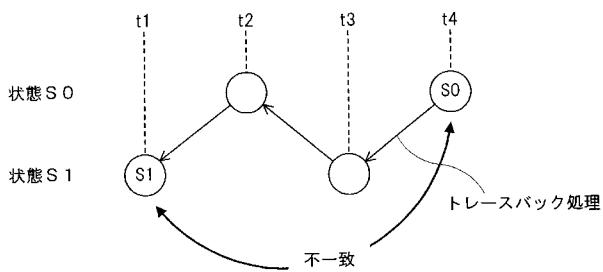
【図15】



【図17】



【図16】



フロントページの続き

(72)発明者 高梨 和司

静岡県浜松市元城町216番18号 株式会社パナソニックモバイル静岡研究所内

(72)発明者 秋山 健

大阪府門真市大字門真1006番地 松下電器産業株式会社内

F ターム(参考) 5B001 AA10

5J065 AC02 AE06 AF03 AG05 AH17 AH21 AH23

5K014 AA01 BA10