

(19)日本国特許庁(JP)

(12)特許公報 ( B 2 )

(11)特許出願公告番号

特公平 8 - 8 5 0 2

(24)(44)公告日 平成8年(1996)1月29日

(51) Int. Cl. <sup>6</sup>	識別記号	庁内整理番号	F I	技術表示箇所
H 0 3 M 7/30		B 9382 - 5 K		
G 1 0 L 3/00	5 1 5	B		
	9/18	E		
H 0 4 B 14/04		Z		
請求項の数 2			H 0 4 N 7/13	Z
			(全 8 頁)	最終頁に続く

(21)出願番号 特願平2-160268  
 (22)出願日 平成2年(1990)6月18日  
 (65)公開番号 特開平4-49719  
 (43)公開日 平成4年(1992)2月19日

(71)出願人 999999999  
 日本電信電話株式会社  
 東京都新宿区西新宿3丁目19番2号  
 (71)出願人 999999999  
 エヌ・ティ・ティ 移動通信網株式会社  
 東京都港区虎ノ門2丁目10番1号  
 (72)発明者 守谷 健弘  
 東京都千代田区内幸町1丁目1番6号 日本  
 電信電話株式会社内  
 (72)発明者 三木 俊雄  
 東京都千代田区内幸町1丁目1番6号 日本  
 電信電話株式会社内  
 (74)代理人 弁理士 草野 卓  
  
 審査官 松尾 淳一

最終頁に続く

(54)【発明の名称】ベクトル量子化法

1

2

【特許請求の範囲】

【請求項 1】入力信号を複数サンプルまとめてベクトルとし、そのベクトルと、予め用意した符号帳の中から最も類似した再生ベクトルを選択し、その選択した再生ベクトルの番号を符号化することで量子化を行うベクトル量子化法において、  
 上記再生ベクトルの選択処理を、再生ベクトルの一部のサンプルに対して異なる処理をしたものとし、  
 上記処理を示す符号とを組み合わせることを特徴とするベクトル量子化法。

【請求項 2】上記再生ベクトルに対する処理は、偶数番目のサンプルに共通に符号の反転あるいは拡大縮小を行い、奇数番目のサンプルに共通に符号の反転あるいは拡大縮小を行うことを特徴とする請求項 1 記載

のベクトル量子化法。

【発明の詳細な説明】

「産業上の利用分野」

この発明は、入力信号を複数のサンプルにまとめてベクトルとし、このベクトルと、予め用意した符号帳の中から最も類似する再生ベクトルを選択し、その再生ベクトルの番号を符号として伝送あるいは記憶することにより、音声や画像の信号系列を少ない情報量で高効率にデジタル符号化するベクトル量子化法に関するものである。

「従来の技術」

音声や画像の信号系列を高効率で符号化するベクトル量子化法は、従来においては第 4 図に示すような構成で行われていた。入力端子 11 からの標準化されてデジタル信号とされた入力信号は波形切出し、またはスペクト

ル分析により、k 個のサンプルがベクトル化回路12で k 次元のベクトル u とされる。この入力ベクトル u と、予め用意した m 個の k 次元再生ベクトル x からなる符号帳 13 の中から距離が最小となる再生ベクトル、つまり最も類似した再生ベクトルを符号化部14で選択し、その選択した再生ベクトルの番号（インデックス）を伝送符号として伝送する（または記憶装置に蓄積する）。再生側では復号部15で再生された番号と対応する再生ベクトルを、符号帳13と同一構成の符号帳16から読み出して出力

$$d_j = \sum_{i=1}^k u(i)^2 - 2 \sum_{i=1}^k u(i)x_j(i) + \sum_{i=1}^k x_j(i)^2$$

となり、その第 1 項は x<sub>j</sub>（再生ベクトル）に無関係であるから

$$e_j = - 2 \sum_{i=1}^k u(i)x_j(i) + \sum_{i=1}^k x_j(i)^2$$

を最小化するような x<sub>j</sub> を符号化部14で捜せばよい。

量子化歪を小さくするためには、入力信号ベクトルの次元数 k を大きくすればよい。しかしながら、次元数 k を大きくすると入力ベクトルと再生ベクトルとの距離演算が 2 の k 次元数乗に比例して増大するため、実時間処

$$e_j = - 2 \sum_{i=1}^k u(i)x_j^*(i) + \sum_{i=1}^k x_j^*(i)^2$$

であるから、

$$e_j = - \left| 2 \sum_{i=1}^k u(i)x_j(i) \right| + \sum_{i=1}^k x_j(i)^2$$

を最小とする x<sub>j</sub> を求め、その x<sub>j</sub> についてのみ正の符号をつけるか負の符号をつけるかを求めれば、2m 個の再生ベクトルの中から距離最小のベクトルを選択できる。このときの伝送符号は b ビットの番号に 1 ビットの正負の符号を合わせた (b + 1) ビットである。このような量子化方法にも次のような難点がある。

「発明が解決しようとする課題」

伝送路符号に誤りがある場合、b ビットの番号に比較して、正負の符号は誤りに対して復号化したあとの歪が大きくなる。これは符号が反転すればすべての要素の値が反転するためである。従って、この正負の符号についてのみ特別な誤り保護手段が必要になるという欠点があった。

この発明の目的な演算量や符号帳のメモリ量増加を抑えつつ伝送ビット数を増やし、かつ伝送路誤りによる歪の増加を抑えるようなベクトル量子化法を提供することにある。

「課題を解決するための手段」

この発明によれば符号帳からの再生ベクトルの選択処理を、再生ベクトルの一部のサンプルに対して、極性反転や拡大縮小などの異なる処理をしたものと、しないも

する。

通常 b ビットの 2 進符号を、符号化出力、すなわち一つの伝送符号語とすると、m = 2<sup>b</sup> である。入力ベクトル u と再生ベクトル x<sub>j</sub> との距離（言い換えれば歪）d<sub>j</sub> をサンプルごとの 2 乗歪の和で定義すると、

$$d_j = \sum_{i=1}^k (u(i) - x_j(i))^2$$

である。これを展開すると

\* 理装置では限界がある。さらに、符号帳のメモリ量も 2 の k 次元数乗に比例して大きくなるという欠点がある。これを改善する方法として、従来において第 5 図に示すように第 4 図の場合と同じ符号帳13を用い、符号帳13から読み出した再生ベクトル x<sub>j</sub> のすべての要素の正負の符号を符号選択部17で逆転したものと、しないものを選択することにより、x<sub>j</sub> を対称化して用いることにより演算量を殆ど増やすことなくビット数を 1 だけ増加させていた。再生ベクトル x<sub>j</sub> のすべての要素の正負の符号を逆転したベクトル x<sub>j</sub><sup>\*</sup> と入力ベクトル u との距離は

のについて行い、その選択した再生ベクトルの番号と、前記処理を示す符号とを組み合わせると符号化出力する。前記処理は再生ベクトルの偶数番目のサンプルに共通に符号反転あるいは拡大縮小を行い、奇数番目のサンプルに共通に符号反転あるいは拡大縮小を行う。

従来のベクトル量子化では再生ベクトルのすべてのサンプル（要素）の正負の符号（極性）を反転して対称化するにとどまったが、この発明においては再生ベクトル中の一部のサンプルの極性の反転による対称化やサンプル値の拡大縮小により、演算量や符号帳のメモリ容量の増加を抑えつつ符号ビットを増加させることで量子化歪を小さくすると同時に、符号誤りの被害を軽減することが特徴である。

「実施例」

この発明の第 1 の実施例を第 1 図に示し、第 5 図と対応する部分に同一符号を付けてある。この例では k 次元、b ビット（簡単のため、ここでは k, b とともに偶数としておく）のベクトル量子化をもとに、再生ベクトルを奇数要素（サンプル）と偶数要素（サンプル）とにわけて、それぞれに極性を各 1 ビットもたせた場合である。すなわち、(b + 2) ビットの量子化を行う。符号帳13

5

6

には $2^b$ 個の再生ベクトルが用意されている。符号帳13からの再生ベクトルは奇数要素と偶数要素とに分離され、符号選択別17と18とにそれぞれ供給され、符号選択別17,18はそれぞれ入力された要素を極性反転するか、することなく出力し、これら出力は距離計算部21,22でそれぞれ入力ベクトル $u$ 中の奇数番目の要素(サンプ

\*ル)、偶数番目の要素(サンプル)との距離が計算され、これら計算された距離は加算回路23で加算され、制定部24で最も小さい距離の再生ベクトルが選択され、その再生ベクトルの番号がコードとして出力される。

このときの歪(距離)は奇数番目の成分(odd)と偶数番目の成分(even)とに分離して、

$$d_j = \sum_{\substack{i=1 \\ \text{odd}}}^k (u(i) - g_o x_j(i))^2 + \sum_{\substack{i=1 \\ \text{even}}}^k (u(i) - g_e x_j(i))^2$$

で表される。ただし、 $g_o$ と $g_e$ はそれぞれ独立に $+1$ 、 $-1$ のいずれかの値をとる定数である。この歪(距離)を書

き直すと

$$d_j = \sum_{\text{all}}^k u(i)^2 - 2 \sum_{\text{odd}}^k g_o u(i) x_j(i) - 2 \sum_{\text{even}}^k g_e u(i) x_j(i)$$

$$+ \sum_{\text{odd}}^k (g_o x_j(i))^2 + \sum_{\text{even}}^k (g_e x_j(i))^2$$

$$= \sum_{\text{all}}^k u(i)^2 - 2 \sum_{\text{odd}}^k g_o u(i) x_j(i) - 2 \sum_{\text{even}}^k g_e u(i) x_j(i)$$

$$+ \sum_{\text{all}}^k x_j(i)^2$$

ここで第1項は再生ベクトル $x_j$ の選択に無関係であり、第2項、3項、4項をそれぞれ

$$f_1 = 2 \sum_{\text{odd}}^k u(i) x_j(i)$$

$$f_2 = 2 \sum_{\text{even}}^k u(i) x_j(i)$$

$$f_3 = \sum_{\text{all}}^k x_j(i)^2$$

と計算しておく、以下の2種類についてスカラー量で歪を計算すればよい。

$$d_{j1} = -|f_1 + f_2| + f_3$$

$$d_{j2} = -|f_1 - f_2| + f_3$$

各 $j$ について $d_{j1}$ と $d_{j2}$ のうちから小さい方を選び、全ての再生ベクトル $x_j$ の中から最小値の $d_{min}$ を選ぶ。この

あとで、 $d_{j1}$ と $d_{j2}$ の絶対値の中が正となるように $g_o$ と $g_e$ を決め、これら $g_o, g_e$ の極性を示す各1ビットを出力する。

ここで $f_1, f_2, f_3$ の計算は、 $k$ 次元 $b$ ビットのベクトルの2乗距離の計算と同一演算量である。その他の計算は $2^b$ 個の数倍のスカラーの計算(加減算と絶対値)と $2^b$ 回の最小値探索ですむ。すなわち通常の $b$ ビットの場合のベクトル距離計算にわずかのスカラーの計算が追加されるだけである。このとき本来なら $b$ ビットの場合の4倍の計算量を要する $b+2$ ビットの量子化ができていから、大幅に演算量が削減できたことになる。さらに、 $g_o$ と $g_e$ の符号のうち一つが伝送路で誤っても選択された再生ベクトルの半分の要素の符号が反転するだけであるから、従来の技術で説明したようなすべての要素が反転する場合より、符号誤りの被害は小さい。

この発明の第2の実施例を第2図に示す。この例は $k$ 次元、各 $b/2$ ビット(簡単のため、ここでは $k, b$ ともに偶数とておく)の2チャンネルのベクトル量子化をもとに、再生ベクトルを奇数要素と偶数要素にわけて、それぞれ

に極性とゲイン（利得）を各 1 ビットもたせた例である。すなわち（b + 6）ビットの量子化を行う。Aチャネル用の符号帳13aは

$$m' \quad ( = \sqrt{m} )$$

個の再生ベクトル $X_1 \sim X_m$  を備え、各再生ベクトル $x_j$ の中の奇数要素、偶数要素はそれぞれ符号選択回路17a, 18aへ供給され、これら出力と、Aチャネル入力ベクトルの奇数要素、偶数要素とがそれぞれ距離計算部21a, 22aで距離計算される。同様にBチャネル用の符号帳13bは m 個の再生ベクトル $Y_1 \sim Y_m$  を備え、各再生ベクトル $Y_j$ の中の奇数要素、偶数要素はそれぞれ符号選択回路17b, 18bへ供給され、これら出力と、Bチャネル入力ベクトルの奇数要素、偶数要素とがそれぞれ距離計算部21b, 22bで距離計算される。距離計算部21a, 22bの各出力が拡大/縮小部25で係数が掛算されて拡大または縮小され、この出力は補正計算部26で補正計算されて、再生ベクトル $X_j, Y_j$ の各奇数要素をそれぞれ拡大/縮小したものと、A, Bチャネルの入力ベクトルの各奇数要素との距離を求める。同様に距離計算部22a, 22bの各出力は拡大/縮小部27で係数が掛算されて拡大または縮小され、この出力は補正計算部28で補正計算される。補正計算部26, 28の出力が加算され、そのうちの最小のものを示すコードが判定部24から出力される。

このときの入力ベクトルと再生ベクトルとの距離 $d_j$ は

$$d_j = \sum_{i=1}^k (u(i) - g_o(x_j(i) \pm y_n(i)))^2$$

o d d

$$+ \sum_{i=1}^k (u(i) - g_e(x_j(i) \pm y_n(i)))^2$$

e v e n

で表される。ただし、 $g_o$ と $g_e$ はそれぞれ独立に +G1, -G1, +G2, -G2のいずれかの値をとる定数である（奇数偶数各 2 ビット）。例えば、G1=0.7, G2=1.5の場合、G1が縮小操作を、G2が拡大操作を行うことになる。またxとyの符号の関係は2種類用意する（奇数偶数各 1 ビット）。この場合も先の例と同じく、式を展開していくと、bビットの2チャネルの量子化での距離計算と同じ演算量のベクトルの演算とそれに追加されるスカラの演算で（b + 6）ビットの量子化が実現できる。また符号の伝送路誤りがベクトルの一部の値にしか影響しないので、符号誤りの被害も小さくできる。

ここまでの例では簡単のために2乗距離を用いて説明したが、さらに歪の定義に固定的あるいは時变的な重みがつく場合にも同様に適用できることは明らかである。さらに、上記ではkおよびbを偶数としたが、k, bが奇数の場合にも同様に適用できることは明らかである。

上記2例の場合も含めて、基本となるベクトル量子化の符号帳は第3図に示されるように、通常の符号帳学習と同様に、

1) 最も歪（距離）が小さくなる再生ベクトルを捜す（量子化）。

2) 重心計算を行い、再生ベクトルを更新する。

という二つの操作を交互に収束するまで繰り返すことで作成できる。何故なら、1), 2)の両操作ともに、与えられた学習サンプルに対する量子化歪を減少させることができるからである。量子化の際に符号帳からの再生ベクトルに対し、符号反転や拡大縮小を行ったものと、行わないものについて距離計算を行う。また重心の計算は以下のように行う。ここでは入力学習ベクトルに依存した符号の反転や、要素の拡大縮小等の変形を行ったあとでの歪（距離）を最小化するような重心点の計算が行われることが通常のベクトル量子化と異なる。 $A_n$ をその変形を指定する対角要素のみからなる  $k \times k$  の行列として歪を行列形式で扱う。 $X_n$ 番目の学習ベクトル $U_n$ は $X_j$ と $A_n$ を用いると歪が最小化され、その値が $d_n$ であるとす

$$d_n = (U_n - A_n X_j)^T (U_n - A_n X_j)$$

となる。 $X_j$ を再生ベクトルとして採用する全ての学習ベクトルの歪の総和 $D_j$ は次式で表される。

$$D_j = \sum d_n$$

この歪の総和 $D_j$ を $X_j$ で偏微分した結果を0とおくことにより、 $X_j$ の重心は得られる。すなわち、

$$X_j = (A_n^2)^{-1} (A_n U_n)$$

となり、最も歪の少ない再生ベクトル $X_j$ を得ることができ、第2の実施例では極性反転と、拡大縮小とを併用したが、再生ベクトルの一部の要素（サンプル）について拡大縮小を行ってもよい。この場合も、再生ベクトルの全要素について拡大、縮小を行った場合に、そのことを示す符号に誤りが生じた場合よりも、符号誤りの被害が少ない。再生ベクトルの奇数番目と偶数番目とにわけて変形処理をしたが、要は再生ベクトルの一部について変形処理を行えばよく、また二つに分ける場合に限らず、三つ以上に分けて変形処理をしてもよい。

「発明の効果」

以上述べたように、この発明のベクトル量子化法を用いれば、基本となるベクトル量子化の距離計算の演算量は僅かのスカラ演算を追加するだけで、よりビット数の高い量子化が実現できる。また符号帳に必要なメモリの容量も基本となるベクトル量子化の場合と変わらない。さらに正負の符号ビットや拡大縮小の符号に伝送路で誤りが生じてベクトルの全ての要素が大幅に変わることはないので、符号誤りの被害が小さくできる。

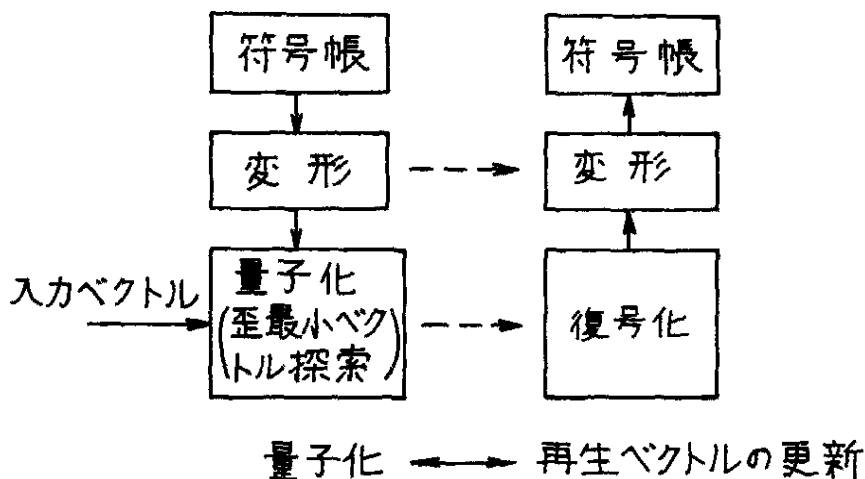
【図面の簡単な説明】

第1図はこの発明の第1の実施例で、奇数番目と偶数番目の要素で独立の符号で変形する場合の構成を示すブロック図、第2図はこの発明の第2の実施例で、奇数番目

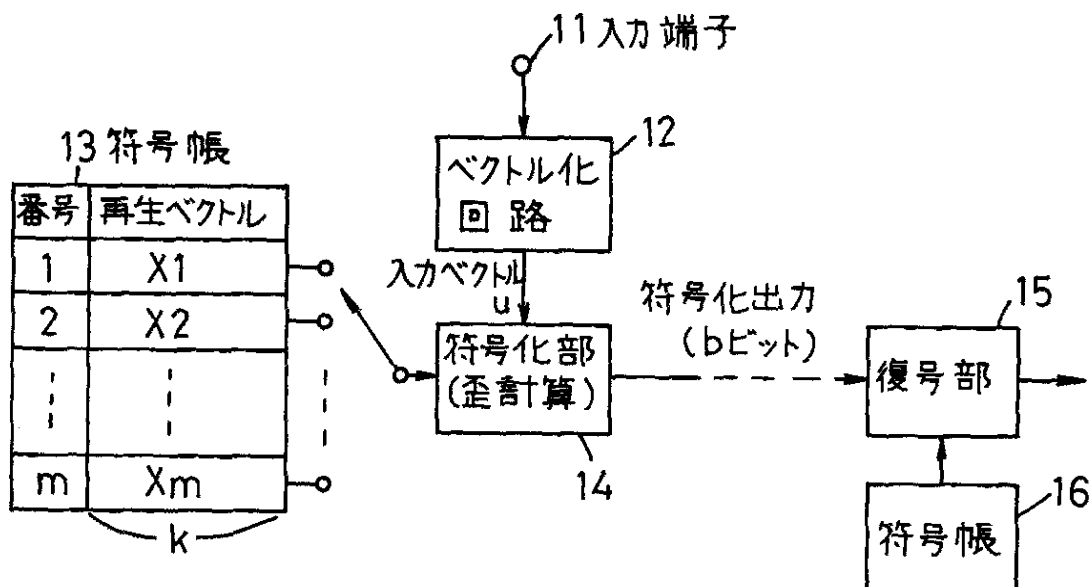
と偶数番目の要素で独立の符号と拡大縮小の符号で変形する2チャンネルのベクトル量子化の構成を示すブロック図、第3図はこの発明の量子化で用いる符号帳の作成法

を示すブロック図、第4図は従来のベクトル量子化法を示すブロック図、第5図は従来のベクトル量子化法で極性符号を併用する場合を示すブロック図である。

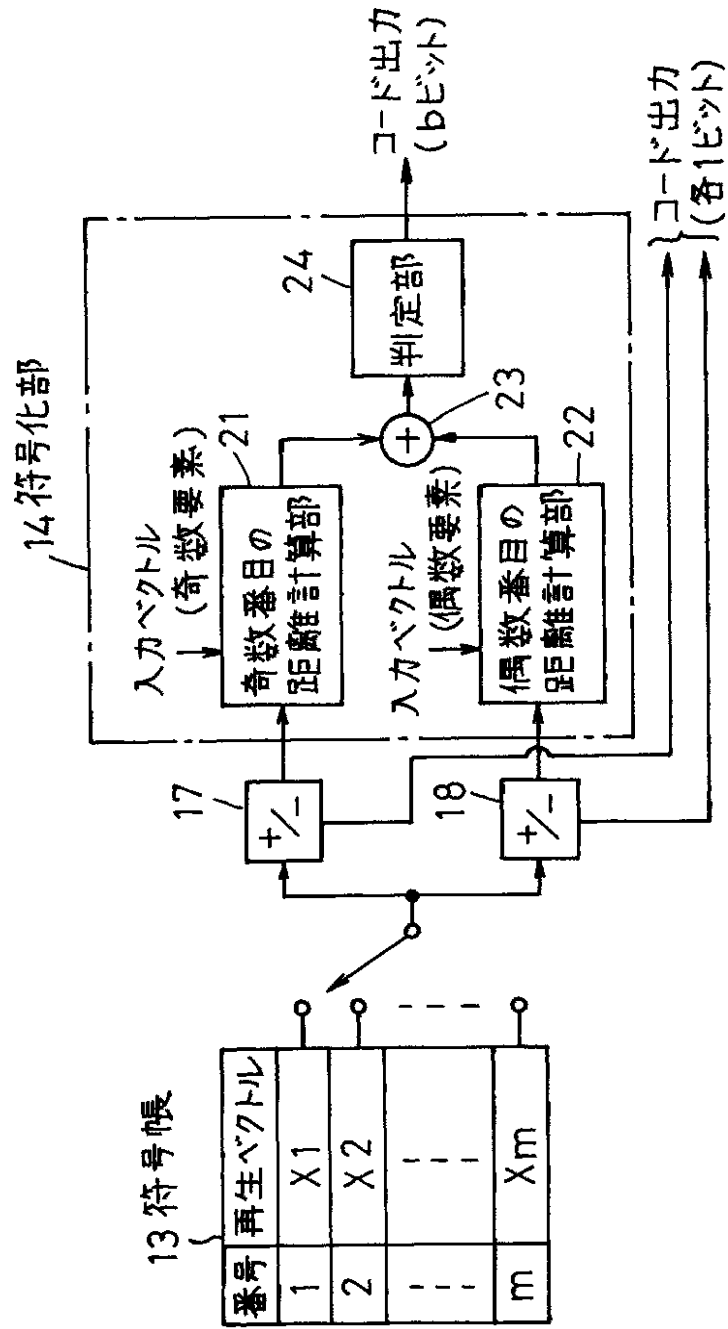
【第3図】



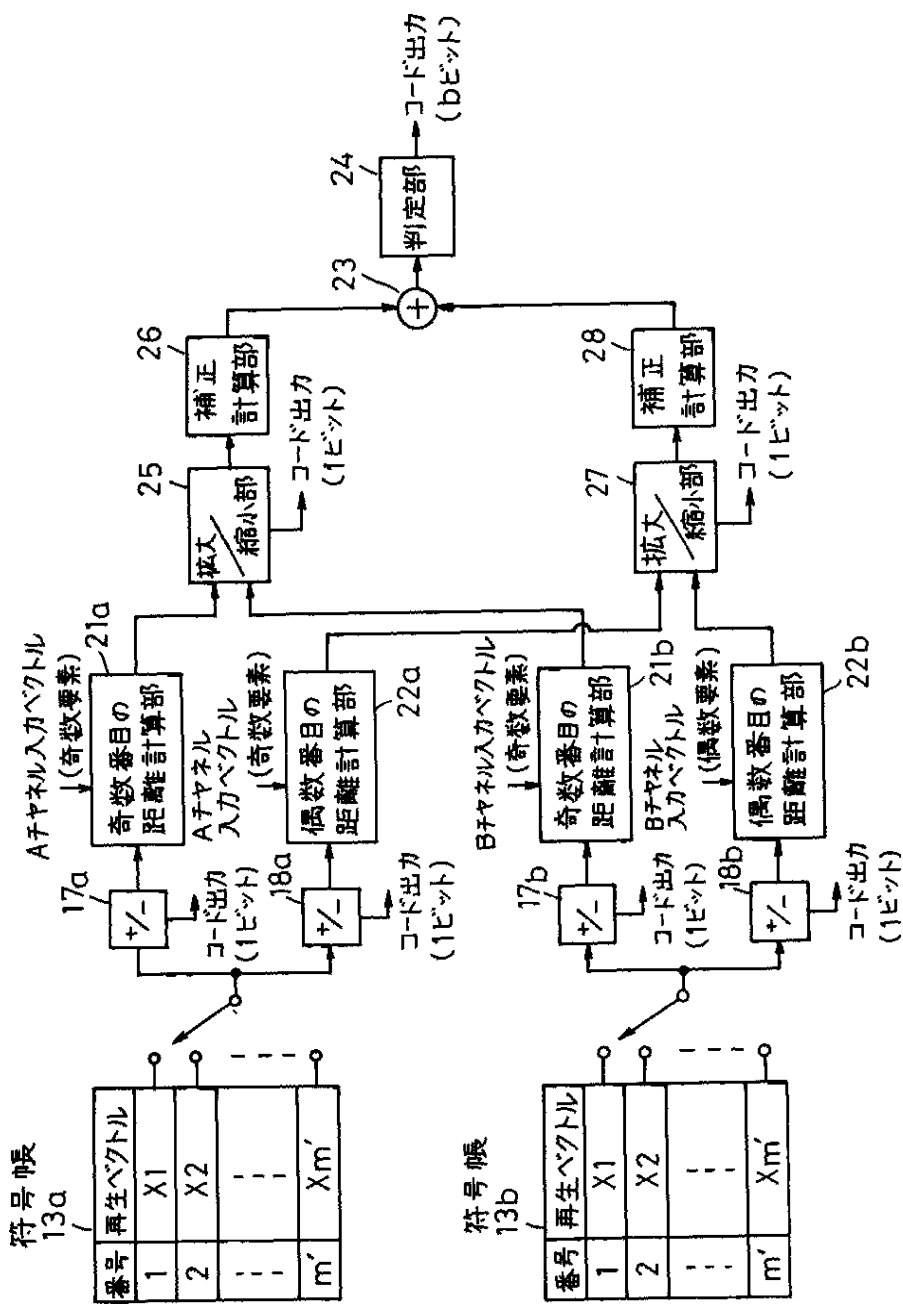
【第4図】



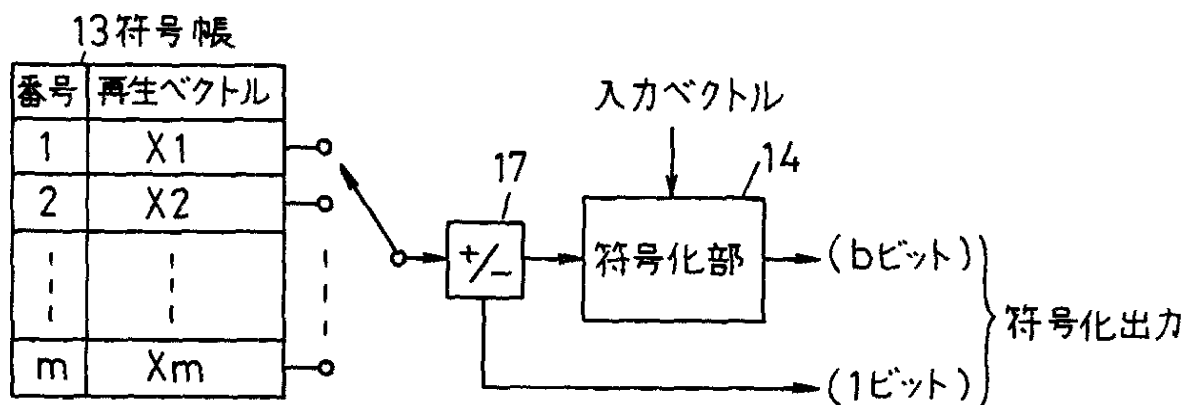
【第 1 図】



【第2図】



【第 5 図】



フロントページの続き

(51) Int. Cl.<sup>6</sup> 識別記号 庁内整理番号 F I 技術表示箇所  
H 0 4 N 7/24

(72) 発明者 須田 博人  
東京都千代田区内幸町 1 丁目 1 番 6 号 日  
本電信電話株式会社内

(72) 発明者 大矢 智之  
東京都千代田区内幸町 1 丁目 1 番 6 号 日  
本電信電話株式会社内