世界知的所有権機関
国 際 事 務 局

PCT

特許協力条約に基づいて公開された国際出願

(51) 国際特許分類 5 G10L 9/18	A1	(11) 国際公開番号	WO 92/05541
		(43) 国際公開日	1992年4月2日 (02.04.1992)

(21) 国際出願番号 PCT/JP91/01235
(22) 国際出願日 1991年9月17日(17. 09. 91)

(30) 優先権データ
特願平2/244174 1990年9月14日(14. 09. 90) JP
特願平3/127669 1991年5月30日(30. 05. 91) JP

(71) 出願人(米国を除くすべての指定国について)
富士通株式会社(FUJITSU LIMITED)〔JP/JP〕
〒211 神奈川県川崎市中原区上小田中1015番地
Kanagawa, (JP)
(72) 発明者;および
(75) 発明者/出願人(米国についてのみ)
谷口智彦(TANIGUCHI, Tomohiko)〔JP/JP〕
大田恭士(OHTA, Yasuji)〔JP/JP〕
栗原秀明(KURIHARA, Hideaki)〔JP/JP〕
田中良紀(TANAKA, Yoshinori)〔JP/JP〕
坂井良広(SAKAI, Yoshihiro)〔JP/JP〕
〒211 神奈川県川崎市中原区上小田中1015番地 富士通株式会社内
Kanagawa, (JP)
ジョンソン マータ (JOHNSON, Mark)〔US/US〕
マサチューセッツ02139, ケンブリッジ, ウィンザー ストリート
ナンバー2 161 Massachusetts, (US)

(74) 代理人
弁理士 青木 朗, 外(AOKI, Akira et al.)
〒105 東京都港区虎ノ門一丁目8番10号 静光虎ノ門ビル
青和特許法律事務所 Tokyo, (JP)

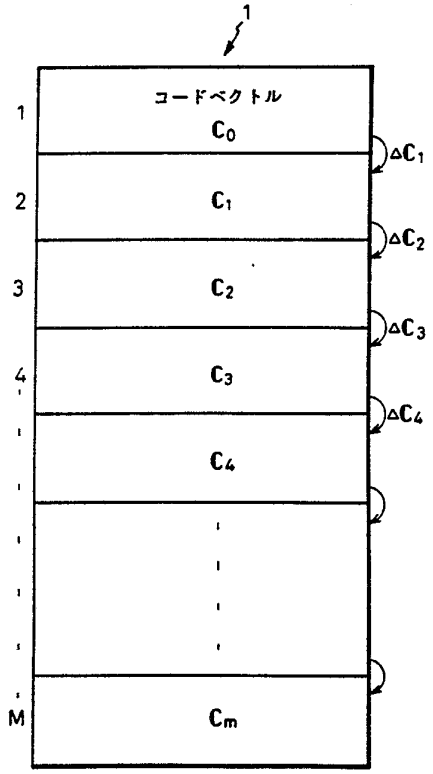
(81) 指定国
CA, DE(欧州特許), FR(欧州特許), GB(欧州特許), JP, US.

添付公開書類 国際調査報告書

(54) Title : VOICE CODING SYSTEM

(54) 発明の名称 音声符号化方式

1 ... code vector



(57) Abstract

A voice coding system which finds by evaluation operation a code vector that minimizes the error between an input voice signal and a reproduced signal obtained through a linear estimation synthesis filtering simulating the vocal tract characteristics for each of the code vectors successively read out from a code book that stores a plurality of noise sequences as code vectors (C_0, C_1, C_2 .....), and then encodes the input voice signal by using a code which specifies a code vector. The code book is constituted as a delta vector code book (11) that stores the initial vector (C_0) and a plurality of delta vectors (ΔCi) which consist of differential vectors among the neighboring code vectors. Operation means for the evaluation operation is provided with a cyclic adder means (20) which accumulates delta vectors for virtual reproduction of said code vectors (C_0, C_1, C_2, ...).

(57) 要約

複数の雑音列をそれぞれコードベクトル（C_0，C_1，C_2 …）として格納する符号帳から順次読み出した各コードベクトルに対し、声道特性を模擬した線形予測合成フィルタ処理を施して得られた再生信号と、入力音声信号との間の誤差が最小となるようなコードベクトルを評価演算によって求め、このコードベクトルを特定する符号を用いてその入力音声信号を符号化する音声符号化方式であって、ここに前記符号帳を、初期ベクトル（C_0）と、隣接する各コードベクトル間の差分ベクトルからなる複数のデルタベクトル（ΔC_i）とを格納するデルタベクトル符号帳（ｉｉ）として構成し、前記評価演算を行う演算手段の中に、デルタベクトルを累積して仮想的に前記コードベクトル（C_0，C_1，C_2 …）を再生する巡回加算手段（２０）を設ける。

情報としての用途のみ

PCTに基づいて公開される国際出願のパンフレット第1頁にPCT加盟国を同定するために使用されるコード

AT オーストリア
AU オーストラリア
BB バルバードス
BE ベルギー
BF ブルキナ・ファソ
BG ブルガリア
BJ ベナン
BR ブラジル
CA カナダ
CF 中央アフリカ共和国
CG コンゴー
CH スイス
CI コート・ジボアール
CM カメルーン
CS チェコスロバキア
DE ドイツ
DK デンマーク
ES スペイン
FI フィンランド
FR フランス
GA ガボン
GI ギニア
GB イギリス
GR ギリシャ
HU ハンガリー
IT イタリー
JP 日本
KP 朝鮮民主主義人民共和国
KR 大韓民国
LI リヒテンシュタイン
LK スリランカ
LU ルクセンブルグ
MC モナコ
MG マダガスカル
ML マリ
MN モンゴル
MR モーリタニア
MW マラウイ
NL オランダ
NO ノルウェー
PL ポーランド
RO ルーマニア
SD スーダン
SE スウェーデン
SN セネガル
SU+ソビエト連邦
TD チャード
TG トーゴ
US 米国

+SUの指定はロシア連邦の指定としての効力を有する。しかし、その指定が旧ソヴィエト連邦のロシア連邦以外の他の国で効力を有するかは不明である。

明　細　書

音声符号化方式

〔技術の分野〕

本発明は音声信号の情報圧縮を行うための音声符号化方式に関し、特に４～１６Ｋbps の伝送速度で符号化を行うためのＡｎａｌｙｓｉｓ－ｂｙ－Ｓｙｎｔｈｅｓｉｓ（Ａ－ｂ－Ｓ）形ベクトル量子化を用いた、すなわち合成による分析を行うベクトル量子化を用いた音声符号化方式に関する。

〔背景技術〕

Ａ－ｂ－Ｓ形ベクトル量子化を用いる音声符号器、例えばＣｏｄｅ－Ｅｘｃｉｔｅｄ　Ｌｉｎｅａｒ　Ｐｒｅｄｉｃｔｉｏｎ（ＣＥＬＰ）符号器は、近年、企業内通信システム、ディジタル移動無線システムなどにおいて、音声信号をその品質を保持しつつ情報圧縮する音声符号器として有望視されている。このベクトル量子化音声符号器（以下、単に符号器）においては、符号帳（コードブック）の各コードベクトルに予測重み付けを施して再生信号を作り、再生信号と入力音声信号との間の誤差電力を評価して最も誤差の少ないコードベクトルの番号（インデックス）を決定して受信側に送信するものである。

上記のＡ－ｂ－Ｓ形ベクトル量子化方式による符号器は、前記符号帳に格納された約１０００パターンの音源信号のベ

クトルの１つ１つに対して、線形予測合成フィルタ処理を施し、再生された各音声信号と、符号化すべき入力音声信号との間の誤差が最も小さくなる１つのパターンをその約１０００のパターンの中から探索するという処理を行う。

ところで符号器は通話の即時性が要求されるので、上記の探索処理をリアルタイムで行う必要がある。そうすると、その探索処理を行うのに、例えば５msという短い時間間隔で、通話の間連続して行わなければならない。

しかしながら後述する如く、この探索処理の中にフィルタ演算や相関演算という複雑な演算操作が含まれていて、これらの演算操作に要する演算量は、例えば数１００M乗算・加算／秒という膨大なものになる。これに対応するには、現在、最高速とされるＤｉｇｉｔａｌ Ｓｉｇｎａｌ Ｐｒｏｃｅｓｓｏｒ（ＤＳＰ）をもってしても、数チップを必要とし、例えば携帯電話に適用しようとする場合、その小形化ならびに低消費電力化が困難になるという問題がある。

〔発明の開示〕

本発明は上記問題点に鑑み、高品質・高能率というＡ－ｂ－Ｓ形ベクトル量子化符号器の特質をそのまま維持しつつ、上記演算量を飛躍的に減少させることのできる音声符号化方式を提供することを目的とするものである。

本発明は上記目的を達成するために、上記符号帳の各コードベクトルの間に、前のコードベクトルC_{n-1}に対し差分ベクトル（以下、デルタベクトルと称す）ΔC_nを加えること

で次のコードベクトルC_nが生成されるようなコードベクトル群を、該符号帳に格納する。ここにnはコードベクトル群の順番を表す。

〔図面の簡単な説明〕

本発明を添付の図面を参照しながら以下に説明する。

第１図は音声発生のメカニズムを説明するための図、

第２図はＡ－ｂ－Ｓ形ベクトル量子化音声符号器の一般的な構成を表すブロック図、

第３図は第２図の構成において符号帳探索処理の部分を具体的に示すブロック図、

第４図は本発明の基本的な考え方を表す図、

第５図は本発明に基づく第１実施例の概念を単純に表す図、

第６図は第１実施例のもとでの符号帳探索処理の部分を具体的に示すブロック図、

第７図は第１実施例のもとでの符号帳探索処理の部分を別の例をもって具体的に示すブロック図、

第８図は自己相関演算部の他の実現列を示す図、

第９図は第１実施例のもとでの符号帳探索処理の部分をさらに別の例をもって具体的に示すブロック図、

第１０図は自己相関演算部の他の実現例を示す図、

第１１図は本発明に基づく第２実施例の原理構成を表す図、

第１２図は第１１図の第２実施例を具体的に表す図、

第１３図は第２実施例の特徴をなす木構造状配列のデルタベクトルについて説明するための図、

第１４Ａ，１４Ｂおよび１４Ｃ図は符号帳内に仮想的に形成されるコードベクトルの分布（モードＡ、モードＢおよびモードＣ）をそれぞれ表す図、

第１５Ａ，１５Ｂおよび１５Ｃ図は変形第２実施例に基づくベクトルの並べ替えを説明するための図、

第１６図は変形第２実施例に基づく符号帳探索処理の部分の一例を示す図、

第１７図は逐次最適化ＣＥＬＰ形の符号器を示す図、

第１８図は同時最適化ＣＥＬＰ形の符号器を示す図、

第１９図は第１７図におけるアルゴリズムを表す図、

第２０図は第１８図におけるアルゴリズムを表す図、

第２１Ａ図は逐次最適化ＣＥＬＰ方式の場合におけるゲイン最適化動作を図解的に表すベクトル図、

第２１Ｂ図は同時最適化ＣＥＬＰ方式の場合におけるゲイン最適化動作を図解的に表すベクトル図、

第２１Ｃ図はピッチ直交変換最適化ＣＥＬＰ方式の場合におけるゲイン最適化動作を図解的に表すベクトル図、

第２２図はピッチ直交変換最適化ＣＥＬＰ形の符号器を示す図、

第２３図は第１実施例のもとでの符号帳探索処理の部分をさらに別の例をもって具体的に示す図、

第２４Ａおよび２４Ｂ図はハウス・ホルダー直交変換を説明するためのベクトル図、

第２５図は本発明に係る第１実施例によって演算量を減少させることができることを示す図、

第２６図は本発明に係る第２実施例によって演算量を減少させると共にメモリ量を削減できることを示す図である。

〔発明を実現する最良の形式〕

第１図は音声発生のメカニズムを説明するための図である。

音声には有声音と無声音があり、有声音は声帯の振動によるパルス音源が基となって発生し、個人個人のノドや口の声道特性が付加されて声になる。また、無声音は声帯を振るわせないで出す音で、単なるガウス性の雑音列が音源となって声道を通って声となる。したがって、音声発生メカニズムは第１図に示すように、有声音の元となるパルス音源ＰＳＧと無声音の元となる雑音源ＮＳＧと、各音源（ＰＳＧ，ＮＳＧ）から出力される信号に声道特性を付加する線形予測合成フィルタＬＰＣＦによりモデル化できる。なお、人の声は周期性を有し、該周期はパルス音源ＰＳＧから出力されるパルスの周期性に対応するから、人や話の内容によって異なる。

以上から、入力音声に対応するパルス音源のパルス周期と雑音源の雑音列を特定することができれば、これらパルス周期と雑音源の雑音列とを識別する符号（データ）により入力音声を符号化することができる。

そこで、適応符号帳を用いて入力音声信号の周期性に基づいてパルス音源のパルス周期を同定し、該周期を備えたパルス列を線形予測合成フィルタに入力してフィルタ演算処理を施し、得られたフィルタ演算結果を入力音声信号から減算して周期成分を除去する。ついで、予め複数の雑音列（各雑音

列はＮ次元の予め定めたコードベクトルで表現されている）を用意しておき、各コードベクトルに合成フィルタ処理を施した再生信号ベクトルと周期成分が除去された入力信号ベクトル（Ｎ次元ベクトル）との誤差が最小となるような１つのコードベクトルを求めれば、前記周期とコードベクトルを特定する符号（データ）により音声を符号化することができる。そしてこのデータが受信側に送信され、原音声（入力音声信号）が再生される。このデータは高い圧縮がかけられた情報である。

第２図はＡ－ｂ－Ｓ形ベクトル量子化音声符号器の一般的な構成を表すブロック図である。本図において、参照番号１はランダムに発生した複数の例えば１０２４種類の雑音列Ｃ（各雑音列はＮ次元のコードベクトルで表現されている）を記憶する雑音符号帳、２はゲインｇの増幅部、３は増幅部出力に声道特性を模擬した合成フィルタの演算処理を施す線形予測合成フィルタ、４は線形予測合成フィルタ３から出力される再生信号ベクトルと入力信号ベクトルの誤差を出力する誤差発生部、５は誤差を評価し、該誤差が最小となる雑音列（コードベクトル）を求める誤差電力評価部である。

Ａ－ｂ－Ｓによるベクトル量子化では通常のベクトル量子化と異なり、雑音符号帳１の各コードベクトル（Ｃ）に最適のゲインｇを掛けた後、線形予測合成フィルタ３でフィルタ処理を施し、フィルタ処理により得られる再生信号ベクトル（ｇＡＣ）と入力音声信号ベクトル（Ｘ）との間の誤差信号（Ｅ）を誤差発生部４で求め、誤差電力評価部５で誤差信号

の電力を評価関数（距離尺度）として雑音符号帳１の探索を行い、誤差電力が最小となる雑音列（コードベクトル）を求め、該雑音列（コードベクトル）を特定する符号により入力音声信号を符号化する。Ａは聴覚重み付けマトリクスである。

上記の誤差電力は次式

$$|E|^2 = |X - gAC|^2 \qquad (1)$$

により与えられる。最適なコードベクトルＣおよびゲインｇは、この（１）式に示す誤差電力を最小化するものとして決定される。なお、声の大きさによりパワーが異なるので、ゲインｇを最適化して再生信号ｇＡＣのパワーを入力音声信号Ｘのパワーに合わせる。最適ゲインは（１）式をｇで偏微分して０とおくことにより求めることができる。すなわち、

$$d|E|^2 / dg = 0$$

より、ｇは

$$g = (X^T AC) / ((AC)^T (AC)) \qquad (2)$$

で与えられる。このｇを（１）式に代入すると、

$$|E|^2 = |X|^2 - (X^T AC)^2 / ((AC)^T (AC)) \qquad (3)$$

となる。入力信号Ｘと合成フィルタ出力ＡＣの相互相関をR_{xc}、合成フィルタ出力ＡＣの自己相関をR_{cc}とすれば、相互相関、自己相関は次式

$$R_{xc} = X^T AC \qquad (4)$$

$$R_{cc} = (AC)^T (AC) \qquad (5)$$

により表現される。なお、Ｔは転置（ｔｒａｎｓｐｏｓｅ）マトリクスを表す。

（３）式の誤差電力Ｅを最小にするコードベクトルＣは同式の右辺第２項を最大にするものであるから、該コードベクトルＣは次式

$$C = \mathrm{argmax}\left(R_{xc}^2 / R_{cc}\right) \qquad (6)$$

（但し、ａｒｇｍａｘはａｒｇｕｍｅｎｔの最大）

と表現でき、最適のゲインは（６）式を満たす相互相関および自己相関を用いて（２）式より

$$g = R_{xc} / R_{cc} \qquad (7)$$

で与えられる。

第３図は第２図の構成において符号帳探索処理の部分を具体的に示すブロック図である。つまり、誤差電力が最小となる雑音列（コードベクトル）を求めて入力信号を符号化する雑音符号帳探索処理部分の構成図である。参照番号１はＭ種類（サイズＭ）の雑音列Ｃ（各雑音列はＮ次元のコードベクトルで表現されている）を記憶する雑音符号帳、３は声道特性を模擬したフィルタ演算処理を施す分析次数N_pの線形予測合成フィルタ（ＬＰＣフィルタ）である。なお、第２図の増幅部２は記載を省略した。

参照番号６は相互相関R_{xc}（$=\mathbf{X}^T\mathrm{AC}$）を演算する乗算部、７は相互相関R_{xc}の二乗を演算する二乗演算部、８は自己相関R_{cc}（$=(\mathrm{AC})^T(\mathrm{AC})$）を演算する自己相関演算部、９は$R_{xc}^2 / R_{cc}$を演算する除算部、１０は$R_{xc}^2 / R_{cc}$が最大となる、換言すれば誤差電力が最小となる雑音列（コードベクトル）を決定して、該コードベクトルを特定する符号を出力する誤差電力評価決定部である。これらの構成

要素６，７，８，９および１０が、第２図の誤差電力評価部５に対応する。

上述した従来の符号帳探索処理では、既に述べたような問題が生ずる。これについて説明する。

従来の符号帳探索処理のうちで主なものは、（１）コードベクトルCに対するフィルタ処理、（２）相互相関R_{xc}の算出処理および（３）自己相関R_{cc}の算出処理の３つである。ここでＬＰＣフィルタ３の次数をN_P、ベクトル量子化（コードベクトル）の次元をＮとすると、１つのコードベクトルに対して、上記（１）～（３）項のそれぞれに要する演算量はN_P・Ｎ，ＮおよびＮである。したがって、１つのコードベクトル当り、符号帳探索に要する演算量は（N_P＋２）・Ｎとなる。通常用いられている雑音符号帳１は、４０次元・符号帳サイズ１０２４（Ｎ＝４０，Ｍ＝１０２４）程度のものであり、ＬＰＣフィルタ３の分析次数N_Pが１０次程度であるため、１回の符号帳探索に

（１０＋２）・４０・１０２４＝４８０Ｋ

の積和算を要する。但し、Ｋ＝10^3である。

このような符号帳探索を、音声符号化のサブフレーム（５msec）毎に行うためには、９６Mops（メガオペレーション／秒）という膨大な処理能力が必要となり、現在最高速のデジタル・シグナル・プロセッサ（許容演算量２０～４０Mops）をもってしても、その実時間処理には数チップを要してしまう。これが問題点である。以下、この問題点を解決するいくつかの実施例を説明する。

第４図は本発明の基本的な考え方を表す図である。本図の雑音符号帳１には各々がＮ次元のＭ個の雑音列がコードベクトルC_0，C_1，C_2…C_3，C_4…C_mとして格納されている。通常これらのコードベクトル相互間には何の関連性もない。このため従来は、第３図の探索処理を行うのに、ｍ個のコードベクトルの１つ１つについて全く独立に誤差電力の評価のための演算を行っていた。

ところが、コードベクトルに対する見方を変えると、第４図に表すようにデルタベクトルΔＣによって相互間を関連付けることができる。これを数式で表すと次のとおりである。

$$C_0 = C_0$$

$$C_1 = C_0 + \Delta C_1$$

$$C_2 = C_1 + \Delta C_2 \quad (= C_0 + \Delta C_1 + \Delta C_2)$$

$$C_3 = C_2 + \Delta C_3 \quad (= C_0 + \Delta C_1 + \Delta C_2 + \Delta C_3)$$

$$\vdots$$

$$C_{1023} = C_{1022} + \Delta C_{1023} \quad (= C_0 + \Delta C_1 + \cdots + \Delta C_{1023}) \qquad (8)$$

上記式の中の例えば、コードベクトルC_2についてみると、この中にはコードベクトルC_1の要素が含まれている。そうするとコードベクトルC_2について演算を行うとき、コードベクトルC_1に関する部分については既に演算が終わっており、その結果を利用するならば、残りの演算はデルタベクトルΔC_2についてのみ更新を行えばよい、ということになる。

そうすると、各デルタベクトルΔＣができるだけシンプルであることが必要である。デルタベクトルΔＣが複雑である

と、上記の例の場合、コードベクトルC_2を単独で従来どおり演算するのに要した演算量と、デルタベクトルΔC_2について更新するための演算量との間に余り差がなくなってしまうからである。

第５図は本発明に基づく第１実施例の概念を単純に表す図であり、次のコードベクトル、例えばｉ番目のコードベクトルC_iは、前のコードベクトルすなわちコードベクトルC_{i-1}に対し、デルタベクトルΔC_iを加えたものになっている。このとき、デルタベクトルΔC_iは上述のようにできるだけシンプルなものであることが必要である。第５図のC_{i-1}，ΔC_i及びC_iの各欄の横軸に沿って描かれた黒丸の列は、Ｎ次元コードベクトルのときｎ個（Ｎサンプル）であり、雑音列の波形の各サンプル点に相当する。各コードベクトルが例えば４０サンプル（Ｎ＝４０）からなるとき、各欄の黒丸の数は４０個になる。第５図では、デルタベクトルΔC_iとしてわずか４個の有意義サンプルデータΔ１，Δ２，Δ３およびΔ４からなる例を示しており、きわめてシンプルである。

逆に言えば、雑音符号帳１の中に例えば１０２４（Ｍ＝１０２４）パターンのコードベクトルを格納してテーブルとするとき、どのようにこれらコードベクトルを配列するかは全く任意であるから、前後のコードベクトル（C_{i-1}，C_i）の間の差をとったときに、その差分ベクトル（Δ Ｃ）がなるべくシンプルになるように雑音符号帳１のコードベクトルを並べ換えればよい。つまり、隣接するどの２つのコードベクトル（C_{i-1}，C_i）をとっても、両者間のデルタベクトル

（ΔC_i）が第５図のように数サンプルデータのシンプルなベクトルとなるように、各コードベクトルを並べてオリジナルテーブルにする。

このようにすると、上記（８）式で示したように初期ベクトルC_0について演算した演算結果を記憶しておけば、その後は、コードベクトルC_1，C_2，C_3…に関し、それぞれ単純なデルタベクトルΔC_1，ΔC_2，ΔC_3…の部分についてのみ更新演算し、C_1の結果に対して巡回加算して行けばよいことになる。

なお、第５図のコードベクトルC_{i-1}，C_iとしては既にスパース化された、すなわちサンプル値が零であるコードを多く含むように予め処理されたコードベクトルを用いた例を示している。ただし、コードベクトルのスパース化技術は公知である。

具体的には、隣接する任意の２つのコードベクトルC_{i-1}およびC_iの間の差分が、単純なデルタベクトルΔC_iになるようなデルタベクトル群を順次、デルタベクトル符号帳１１（後述）に格納するようにする。

第６図は第１実施例のもとでの符号帳探索処理の部分を具体的に示すブロック図である。基本的には、前述した第３図の構成に対応するが、第６図は周知の逐次最適化ＣＥＬＰ形の音声符号器に適用した例を示す。したがって、入力音声信号Ｘ（第３図）に代えて、聴覚重み付けしたピッチ予測誤差信号ベクトルＡＹが示されているが、本発明を説明する上で何ら影響はない。また、演算手段１９が示されているが、こ

れは線形予測合成フィルタ３を第３図に示した位置から、第６図に示す位置に移し替えたことに伴う前処理手段であって、本発明を理解する上で重要な要素ではない。

第３図の相互相関R_{xc}を生成する部分に対応するのが、第６図の相互相関演算部１２であり、第３図の自己相関R_{cc}を生成する部分に対応するのが、第６図の自己相関演算部１３である。この相互相関演算部１２内に、本発明を実現する巡回加算手段２０が、加算部１４および遅延部１５として示されている。同じく自己相関演算部１３内に本発明を実現する上記の巡回加算手段２０が、加算部１６および遅延部１７として示されている。

最も注目すべき点は、第６図のデルタベクトル符号帳１１であり、第３図の雑音符号帳１のように各コードベクトルC_0，C_1，C_2…を格納せず、初期ベクトルC_0以降は、直前のベクトルとの差分であるデルタベクトルΔC_1，ΔC_2，ΔC_3…が格納される。

初めに初期ベクトルC_0について演算すると、その演算結果は遅延部１５（遅延部１７についても同じ）に保持され、フィードバックされて、次に来るデルタベクトルΔC_1に加算部１４（加算部１６についても同じ）で巡回的に加算される。以下、同様であり、結果的に、引き続くコードベクトルC_1，C_2，C_3…について個別に演算していた従来法と等価な処理をしたことになる。

以下、詳細に説明する。聴覚重み付けしたピッチ予測誤差信号ベクトルＡＹを演算手段２１でA^T ＡＹに変換しておき、

デルタベクトル符号帳１１のデルタベクトルΔＣをそのまま相互相関演算部１２に与えて乗算すると共に前回の相関値（ＡＣ$_{i-1}$）^{T} ＡＹを巡回的に加算することにより両者の相関値（ＡＣ）^{T} ＡＹを発生することができる。

すなわち、Ｃ$_{i-1}$ ＋Δ Ｃ$_{i}$ ＝Ｃ$_{i}$ であることから、

（ＡＣ$_{i}$）^{T} ＡＹ＝（Ｃ$_{i}$）^{T} Ａ^{T} ＡＹ

＝（ΔＣ$_{i}$）^{T} Ａ^{T} ＡＹ＋（ＡＣ$_{i-1}$）^{T} ＡＹ （９）

という演算により今回の相関値（ＡＣ）^{T} ＡＹが生成されて誤差電力評価部５に与えられることになる。

また第６図に示すように自己相関演算部１３においては、デルタベクトルΔＣを前回のコードベクトルＣ$_{i-1}$ と巡回的に加算することによりコードベクトルＣ$_{i}$ を生成すると共に、聴覚重み付け再生後のコードベクトルＡＣの自己相関値（ＡＣ）^{T} ＡＣを求めて、評価部５に与えている。

したがって、相互相関演算部１２および自己相関演算部１３においては、スパース化されたデルタベクトルの乗算で済むので演算量を削減することができる。

第７図は第１実施例のもとでの符号帳探索処理の部分を別の例をもって具体的に示すブロック図であり、周知の同時最適化ＣＥＬＰ形の音声符号器に適用した場合を示す。この場合も本発明は何の支障もなく適用される。本図でも、第１および第２の演算手段１９－１および１９－２は本発明に直接関係しない。なお、相互相関演算部（１２）は、入力音声系とピッチＰ（既述の周期）系とに分けて並行して処理されるので、第１および第２の相互相関演算部１２－１および１２

－２としている。

入力音声信号ベクトルＡＸを第１の演算手段１９－１でA^T ＡＸに変換し、ピッチ予測残差ベクトルＡＰを第２の演算手段１９－２でA^T ＡＰに変換しておき、それぞれデルタベクトルΔＣを第１および第２の相互相関演算部１２－１，１２－２で乗算すると共に巡回的に加算して$(AC)^T$ ＡＸおよび$(AC)^T$ ＡＰを生成し、また自己相関演算部１３では同様にして$(AC)^T$ ＡＣを生成して評価部５に与えるので、デルタベクトルの分だけの演算量で済むことになる。

第８図は自己相関演算部の他の実現例を示す図であり、第６図および第７図に示した自己相関演算部１３は他の構成でも実現することができる。ここに示す演算器２１は、第６図および第７図において、合成フィルタ３と自己相関演算部８においてそれぞれ必要であった乗算を１回の乗算で済ますようにしたものである。

この演算器２１において、前回のコードベクトルC_{i-1}および聴覚重み付けマトリクスＡの相関値A^T Ａを記憶しておき、デルタベクトルΔC_iとの演算を行うと共に、加算部１６と遅延部１７（巡回加算手段２０）で巡回的に加算することにより自己相関値$(AC)^T$ ＡＣを求めることができる。

すなわち、$C_{i-1} + \Delta C_i = C_i$であることから、

$$
\begin{aligned}
&(AC_i)^T AC_i \\
&= (AC_{i-1})^T (AC_{i-1}) \\
&- (\Delta C_i)^T (A^T A) C_{i-1} \\
&+ (\Delta C_i)^T (A^T A) \Delta C_i \qquad (10)
\end{aligned}
$$

という演算に従い、相関値$A^T A$および前回のコードベクトルC_{i-1}を記憶しておいて今回の自己相関値$(AC)^T AC$が生成されて評価部5に与えることができる。

こうすれば、$A^T A$とΔC_iとC_{i-1}との乗算だけになり、前述したように第6図や第7図のように2度の乗算を行う必要が無くなり、その分、演算量を削減することができる。

第9図は第1実施例のもとでの符号帳探索処理の部分をさらに別の例をもって具体的に示すブロック図であり、基本的には、前述した第3図の構成に対応するが、第9図はピッチ直交変換最適化CELP形の音声符号器に適用した例を示す。

第9図において、演算手段19′の後段に置かれたブロック22は、時間反転直交変換部であり、演算手段19′で、聴覚重み付けされた入力音声信号ベクトルAXから時間反転聴覚重み付け入力音声信号ベクトル$A^T AX$を算出した後、この時間反転直交変換部22で最適時の聴覚重み付けされたピッチ予測残差ベクトルAPに対して時間反転聴覚重み付け直交変換された入力音声信号ベクトル$(AH)^T AX$を算出するものである。ただし、これら演算手段19′や時間反転直交変換部22は、本発明の本質とは直接関係がない。

相互相関演算部12では、第6図および第7図の場合と同様、デルタベクトルΔCとの乗算および巡回的加算とを行って$(AHC)^T AX$なる相関値を評価部5に与える。Hは直交変換を表すマトリクスである。

このときの演算は、

$$(AHC_i)^T AX = C_i{}^T H^T A^T AX$$
$$= (\Delta C_i)^T (H^T A^T AX)$$
$$+ (AHC_{i-1})^T AX \qquad (11)$$

となる。

一方、自己相関演算部１３では、デルタベクトル符号帳１１の各デルタベクトルΔC_iを、加算部１６と遅延部１７で巡回的に加算してコードベクトルC_iを生成すると共に最適時の、聴覚重み付け（Ａ）されたピッチ予測残差ベクトルＡＰに対して聴覚重み付け直交変換されたコードベクトルＡＨＣ＝ＡＣ′を算出し、さらにこの聴覚重み付け直交変換されたコードベクトルＡＨＣの自己相関値$(AHC)^T AHC = (AC')^T AC'$を求める。

したがって、ピッチ直交変換最適化を行う場合にも同様にしてデルタベクトルによる演算量の削減が可能となる。

第１０図は自己相関演算部の他の実現例を示す図であり、第９図に示した自己相関演算部１３は他の構成でも実現することができる。これは前述した第８図の構成に対応する。

ここに示す演算器２３は、第９図において、合成フィルタ（ＡＨ）３′と自己相関演算部８においてそれぞれ必要であった乗算を１回の乗算で済ますようにしたものである。

この演算器２３において、前回のコードベクトルC_{i-1}および直交変換した聴覚重み付けマトリクスＡＨの相関値$(AH)^T AH$を記憶しておき、デルタベクトルΔC_iとの演算を行うと共に、加算部１６と遅延部１７で巡回的に加算することにより、

$$(AHC_i)^T AHC_i$$
$$= (AHC_{i-1})^T (AHC_{i-1})$$
$$+ (\Delta C_i)^T ((AH)^T AH) C_{i-1}$$
$$+ (\Delta C_i)^T ((AH)^T AH) \Delta C_i \quad (12)$$

なる自己相関値を求めることができ、演算量を削減することができる。ただし、Hは最適APに応じて変更される。

上述した第1実施例は、デルタベクトルΔC_1，ΔC_2，ΔC_3…を直線的に累積することによって、仮想的に、従来の雑音符号帳1に格納されるコードベクトルC_1，C_2，C_3…を得るものである。この場合、各デルタベクトルは例えば40サンプル中の任意の4サンプルを有意義データ（サンプル値が零でないサンプルデータ）とし、スパース化を図っている。しかしこの点を除いては各デルタベクトルの設定に当っては特別な規則付けはなされていない。

次に述べる第2実施例は、デルタベクトル群を特別の規則のもとで生成することにより、符号帳探索処理に要する演算量の大幅な削減を図るものである。また、この第2実施例は、デルタベクトル符号帳11内のメモリ容量を大幅に削減できるという利点も有する。以下、この第2実施例について説明する。

第11図は本発明に基づく第2実施例の原理構成を表す図である。第2実施例の概念は第11図の上半分に図解的に示されており、仮想的に形成される例えば1024個のパターンのコードベクトルを生成するためのデルタベクトルは、＋－の極性をもって一定の規則で木構造状に配列されている。

これにより、フィルタ演算や相関演算をわずか（Ｌ－１）個（Ｌは例えば１０）のデルタベクトルに対する演算に帰着させることができ、演算量を飛躍的に減少させることができる。

第１１図において、参照番号１１は、１つの基準雑音列である初期ベクトルC_0と（Ｌ－１）種類の差分雑音列であるデルタベクトル$\Delta C_1 \sim \Delta C_{L-1}$（Ｌは木構造の階層の数でＬ＝１０）を格納するデルタベクトル符号帳、３は声道特性を模擬したフィルタ演算処理を施す既述の線形予測合成フィルタ（ＬＰＣフィルタ）、３１は初期ベクトルC_0および（Ｌ－１）種類のデルタベクトル$\Delta C_1 \sim \Delta C_{L-1}$に対してそれぞれフィルタ３によるフィルタ演算処理を施して得られる初期ベクトルのフィルタ出力AC_0と（Ｌ－１）種類のデルタベクトルΔCのフィルタ出力$A\Delta C_1 \sim A\Delta C_{L-1}$を記憶する記憶部、１２は相互相関$R_{xc}$（$=X^T AC$）を演算する既述の相互相関演算部、１３は自己相関$R_{cc}$（$=(AC)^T (AC)$）を演算する既述の自己相関演算部、１０は$R_{xc}^2 / R_{cc}$が最大となる、換言すれば誤差電力が最小となる雑音列（コードベクトル）を決定する既述の誤差電力評価決定部、３０は誤差電力が最小となる雑音列（コードベクトル）を特定するデータ（符号）で入力音声信号を符号化する音声符号化部である。この符号器の動作は次のとおりである。

予め１つの基準雑音列である初期ベクトルC_0と（Ｌ－１）種類のデルタ雑音列であるデルタベクトル$\Delta C_1 \sim \Delta C_{L-1}$（Ｌ＝１０）をデルタベクトル符号帳１１に格納しておき、デルタベクトル$\Delta C_1 \sim \Delta C_{L-1}$をそれぞれ初期ベクトル

C_0 に階層毎に加え合わせ（＋）および差し引く（－）ことにより順次木構造状に（$2^{10}-1$）種類の雑音列のコードベクトルC_0 ～C_{1022}を表現し、またはこれらコードベクトルに零ベクトルまたは－C_0 ベクトルを加えて2^{10}のパターンのコードベクトルC_0 ～C_{1023}を表現する。このようにすれば、デルタベクトル符号帳１１に初期ベクトルC_0 と（Ｌ－１）種類のデルタベクトルΔC_1 ～ΔC_{L-1} （Ｌ＝１０）を格納しておくだけで、次々と2^L-1（$=2^{10}-1=M-1$）種類のコードベクトルまたは2^L （$=2^{10}=M$）種類のコードベクトルを生成することができ、デルタベクトル符号帳１１の記憶容量をＬ・Ｎ（＝１０・Ｎ）とすることができ、従来の雑音符号帳１の記憶容量Ｍ・Ｎ（＝１０２４・Ｎ）に比べて著しく減少することができる。

また、初期ベクトルC_0 および（Ｌ－１）種類のデルタベクトルΔC_1 ～ΔC_{L-1} （Ｌ＝１０）に対して、合成フィルタ３で合成フィルタ処理を施して初期ベクトルのフィルタ出力AC_0 と（Ｌ－１）種類のデルタベクトルのフィルタ出力$A\Delta C_1$ ～$A\Delta C_{L-1}$ （Ｌ＝１０）を求めて記憶部３１に記憶しておく。そして、初期ベクトルC_0 のフィルタ出力AC_0 に対して第１のデルタベクトルのフィルタ出力$A\Delta C_1$ を加え合わせおよび差し引くことにより２種類の雑音列のコードベクトルC_1 ，C_2 に対するフィルタ出力AC_1 ，AC_2 を演算し、新たに演算された各雑音列のコードベクトルに対するフィルタ出力AC_1 ，AC_2 に第２のデルタベクトルのフィルタ出力$A\Delta C_2$ を加え合わせおよび差し引くことにより、

それぞれ2種類の雑音列のコードベクトルC_3およびC_4とコードベクトルC_5およびC_6に対するフィルタ出力AC_3～AC_6を演算し、以下同様に第（i－1）番目のデルタベクトルのフィルタ出力AAC_{i-1}を作用させて演算したフィルタ出力AC_kに第i番目のデルタベクトルのフィルタ出力AAC_iを作用させてそれぞれ2種類の雑音列のコードベクトルに対するフィルタ出力AC_{2k+1}およびAC_{2k+2}を演算することにより全コードベクトルのフィルタ出力を発生するようにする。このようにすれば、各コードベクトルC_0～C_{1022}に対する合成フィルタ演算処理を、初期ベクトルC_0および（L－1）（L＝10）種類のデルタベクトルΔC_1～ΔC_{L-1}（L＝10）に対する合成フィルタ演算処理に帰着させることができ、従来、フィルタ処理に

$N_p \cdot N \cdot M$（＝$1024 \cdot N_p \cdot N$）

回の積和算が必要であったものを

$N_p \cdot N \cdot L$（＝$10 \cdot N_p \cdot N$）

回の積和算に減少できる。

さらに、誤差電力評価決定部10で誤差電力が最小となる雑音列（コードベクトル）を決定し、音声符号化部30で該コードベクトルを特定する符号を出力することで音声符号化すると共に、誤差電力が最小となるコードベクトルを求める処理を、合成フィルタ演算出力ACと入力音声信号ベクトルXの間の相互相関R_{xc}（＝X^TAC，Tは転置マトリクス）の二乗と合成フィルタ出力の自己相関R_{cc}（＝（AC）（AC））の比が最大となるコードベクトルを求めることに帰着

させる。そして、合成フィルタ演算出力AC_{2k+1}, AC_{2k+2}を1階層前の合成フィルタ演算出力AC_kと今回のデルタベクトルフィルタ出力ΔAC_iを用いて次式

$$AC_{2k+1} = AC_k + \Delta AC_i$$

$$AC_{2k+2} = AC_k - \Delta AC_i \qquad (12)$$

のように漸化式で表現することにより、相互相関$R_{xc}^{(2K+2)}$, $R_{xc}^{(2K+2)}$を次式

$$R_{xc}^{(2k+1)} = R_{xc}^{(k)} + \mathbf{X}^T (\Delta AC_i)$$

$$R_{xc}^{(2k+2)} = R_{xc}^{(k)} - \mathbf{X}^T (\Delta AC_i) \qquad (13)$$

に示すように漸化式で表現し、相互相関演算部12で1階層前の相互相関$R_{xc}^{(k)}$を用いて今回の相互相関$R_{xc}^{(2k+1)}$, $R_{xc}^{(2k+2)}$を演算するようにする。このようにすれば、右辺第2項の相互相関演算を行うだけで全コードベクトルのフィルタ出力と入力音声信号$\mathbf{X}$との間の相互相関を演算できる。すなわち、従来相互相関を求めるのにM・N（＝1024・N）回の積和算が必要であったものが、L・N（＝10・N）回の積和算で済ますことが可能となり、演算回数を著しく減少できる。

また、自己相関演算部13は、1階層前の自己相関$R_{cc}^{(k)}$を用いて今回の相互相関$R_{cc}^{(2k+1)}$, $R_{cc}^{(2k+2)}$を演算するようにする。このようにすれば、自己相関R_{cc}を初期ベクトルのフィルタ出力AC_0と（L－1）種類のデルタベクトルのフィルタ出力$\Delta AC_1 \sim \Delta AC_{L-1}$の合計L通りの自己相関$(AC_0)^2$, $(\Delta AC_1)^2 \sim (\Delta AC_{L-1})^2$と、各フィルタ出力$AC_0$, $\Delta AC_1 \sim \Delta AC_{L-1}$間の$(L^2-1)$

／２通りの相互相関を用いて演算できる。すなわち、従来自己相関を求めるのにＭ・Ｎ（＝１０２４・Ｎ）回の積和算が必要であったものが、Ｌ（Ｌ＋１）・Ｎ／２（＝５５・Ｎ）回の積和算で済ますことが可能となり、演算回数を著しく減少できる。

第１２図は第１１図の第２実施例を具体的に表す図であり、前述のように、１１は１つの基準雑音列を表現する初期ベクトルC_0と（Ｌ－１）種類の差分雑音列を表現するデルタベクトル$\Delta C_1 \sim \Delta C_{L-1}$（Ｌ＝１０）を記憶・保持するデルタベクトル符号帳であり、初期ベクトルC_0および各デルタベクトル$\Delta C_1 \sim \Delta C_{L-1}$（Ｌ＝１０）はそれぞれＮ次元で表現されている。すなわち、初期ベクトルおよびデルタベクトルは時系列的に発生するＮ個のサンプリングされた雑音の振幅をそれぞれコード化したＮ次元のベクトルである。３は声道特性を模擬したフィルタ演算処理を施す前述の線形予測合成フィルタ（ＬＰＣフィルタ）であり、次数N_pのＩＩＲ（Ｉｎｆｉｎｉｔｅ　Ｉｍｐｕｌｓｅ　Ｒｅｓｐｏｎｓｅ）型フィルタで構成され、Ｎ×Ｎの正方マトリクスＡとコードベクトルＣのマトリクス演算を行って、コードベクトルＣに合成フィルタ処理を施す。ＩＩＲ型フィルタのN_p個の係数は入力音声信号Ｘに基づいて変化し、その都度周知の方法で決定される。すなわち、入力音声信号の隣接サンプルには相関が存在するから、サンプル間の相関係数を求め、該相関係数からパーコール係数と称せられる偏自己相関係数を求め、該パーコール係数からＩＩＲフィルタのアルファ係数を決定

し、当該フィルタのインパルス応答列を用いてＮ×Ｎの正方マトリクスＡを作成してコードベクトルＣに合成フィルタ処理を施す。

３１は基準雑音列を表現する初期ベクトル$\mathbf{C}_0$および（Ｌ－１）種類のデルタ雑音列を表現するデルタベクトル$\Delta\mathbf{C}_1 \sim \Delta\mathbf{C}_{L-1}$に対してそれぞれフィルタ演算処理を施して得られるフィルタ出力$A\mathbf{C}_0$，$A\Delta\mathbf{C}_1 \sim A\Delta\mathbf{C}_{L-1}$を記憶する記憶部、１２は相互相関$R_{xc}$（$=\mathbf{X}^T A\mathbf{C}$）を演算する相互相関演算部、１３は自己相関$R_{cc}$（$=(A\mathbf{C})^T (A\mathbf{C})$）を演算する自己相関演算部、３８は相互相関を二乗したものと自己相関との比を演算する演算部である。

誤差電力$|\mathbf{E}|^2$は前記（３）式で表現されるから、誤差電力を最小にするコードベクトルＣは（３）式の右辺第２項を最大にするものである。したがって、演算部３８は二乗演算部７と除算部９を備え、次式

$$F(\mathbf{X}, \mathbf{C}) = R_{xc}^2 / R_{cc} \qquad (14)$$

を演算する。

１０は既述のように、R_{xc}^2 / R_{cc}が最大となる、換言すれば誤差電力が最小となる雑音列（コードベクトル）を決定する誤差電力評価決定部、３０は誤差電力が最小となる雑音列（コードベクトル）を特定する符号で入力音声信号を符号化する音声符号化部である。

第１３図は第２実施例の特徴をなす木構造状配列のデルタベクトルについて説明するための図である。デルタベクトル符号帳１１には、１つの初期ベクトル$\mathbf{C}_0$と（Ｌ－１）種類

のデルタベクトル$\Delta C_1 \sim \Delta C_{L-1}$（Ｌ＝１０）を格納しておき、各デルタベクトル$\Delta C_1 \sim \Delta C_{L-1}$をそれぞれ初期ベクトル$C_0$に対し階層毎に加え合わせるか（＋）、差し引くか（－）により順次木構造状に（$2^{10}-1$）種類のコードベクトル$C_0 \sim C_{1022}$を仮想的に表現し、これらコードベクトルに零ベクトル（Ｎ次元サンプルの全サンプル値がゼロ）を加えて2^{10}のコードベクトル$C_0 \sim C_{1023}$を表現する。このようにすれば、各コードベクトルの間の関係が次式

$$C_0 = C_0 \qquad \text{I}$$

$$\left.\begin{aligned} C_1 &= C_0 + \Delta C_1 \\ C_2 &= C_0 - \Delta C_1 \end{aligned}\right\} \text{II}$$

$$\left.\begin{aligned} C_3 &= C_1 + \Delta C_2 \quad (= C_0 + \Delta C_1 + \Delta C_2) \\ C_4 &= C_1 - \Delta C_2 \quad (= C_0 + \Delta C_1 - \Delta C_2) \\ C_5 &= C_2 + \Delta C_2 \quad (= C_0 - \Delta C_1 + \Delta C_2) \\ C_6 &= C_2 - \Delta C_2 \quad (= C_0 - \Delta C_1 - \Delta C_2) \end{aligned}\right\} \text{III}$$

$$\vdots$$

$$\left.\begin{aligned} C_{511} &= C_{255} + \Delta C_9 \quad (= C_0 + \Delta C_1 + \Delta C_2 + \cdots + \Delta C_9) \\ C_{512} &= C_{255} - \Delta C_9 \quad (= C_0 + \Delta C_1 + \Delta C_2 + \cdots - \Delta C_9) \\ &\vdots \\ C_{1021} &= C_{510} + \Delta C_9 \quad (= C_0 - \Delta C_1 - \Delta C_2 - \cdots + \Delta C_9) \\ C_{1022} &= C_{510} - \Delta C_9 \quad (= C_0 - \Delta C_1 - \Delta C_2 - \cdots - \Delta C_9) \end{aligned}\right\} \text{XX}$$

（１５）

（Ⅰは第１階層、Ⅱは第２階層、Ⅲは第３階層…XXは第１０階層）

により表現され、一般的に

$$C_{2k+1} = C_k + \Delta C_i \quad (16)$$

$$C_{2k+2} = C_k - \Delta C_i \quad (17)$$

のように漸化式で表現できる。すなわち、デルタベクトル符号帳１１に初期ベクトルC_0と（Ｌ－１）種類のデルタベクトル$\Delta C_1 \sim \Delta C_{L-1}$（Ｌ＝１０）を格納しておくだけで、次々と$2^L$（$=2^{10}$）種類の任意の雑音列のコードベクトルを仮想的に生成することができ、デルタベクトル符号帳１１の記憶容量をＬ・Ｎ（＝１０・Ｎ）とすることができ、従来の雑音符号帳１の記憶容量Ｍ・Ｎ（＝１０２４・Ｎ）に比べて著しく減少することができる。

上記（１６）式および（１７）式に一般的に表現されたコードベクトルC_{2k+1}およびC_{2k+2}に対する線形予測合成フィルタ（Ａ）（第１２図のフィルタ３）でのフィルタ処理について次に説明する。

上記コードベクトルC_{2k+1}，C_{2k+2}に対する合成フィルタ演算出力AC_{2k+1}，AC_{2k+2}は、１階層前のコードベクトルC_kに対する合成フィルタ演算出力AC_kと今回のデルタベクトルのフィルタ出力$A\Delta C_i$を用いてそれぞれ次式

$$AC_{2k+1} = A(C_k + \Delta C_i) = AC_k + A\Delta C_i \quad (18)$$

$$AC_{2k+2} = A(C_k - \Delta C_i) = AC_k - A\Delta C_i \quad (19)$$

但し、$i=1, 2, \cdots L-1$, $2^{i-1} \leqq k < 2^{i}-1$

のように漸化式で表現できる。したがって、初期ベクトルC_0および（Ｌ－１）種類のデルタベクトル$\Delta C_1 \sim \Delta C_{L-1}$（Ｌ＝１０）に対して、合成フィルタ３で合成フィルタ処理を施して初期ベクトルのフィルタ出力AC_0と（Ｌ－１）種類のデルタベクトルのフィルタ出力$A\Delta C_1 \sim A\Delta C_{L-1}$（Ｌ＝１０）とを求めて記憶部３１に記憶しておけば、以下のごとく全雑音列のコードベクトルに対するフィルタ処理を再帰的に行うことができる。

すなわち、（１）初期ベクトルのフィルタ出力AC_0に対して第１のデルタベクトルのフィルタ出力$A\Delta C_1$を次元毎に加え合わせおよび差し引くことにより２種類の雑音列のコードベクトルC_1，C_2に対するフィルタ出力AC_1，AC_2を演算でき、また、

（２）新たに演算された各フィルタ演算出力AC_1，AC_2に第２のデルタベクトルのフィルタ出力$A\Delta C_2$を加え合わせおよび差し引くことにより、それぞれ２種類、合計４種類のコードベクトルC_3，C_4，C_5，C_6に対するフィルタ出力$AC_3 \sim AC_6$を演算でき、以下同様に

（３）第（ｉ－１）番目のデルタベクトルのフィルタ出力$A\Delta C_{i-1}$を作用させて演算したフィルタ出力AC_kに第ｉ番目のデルタベクトルのフィルタ出力$A\Delta C_i$を作用させてそれぞれ２種類のフィルタ出力AC_{2k+1}，AC_{2k+2}を演算することにより2^L（$=2^{10}$）の全雑音列のコードベクトルに対するフィルタ出力を発生できる。

すなわち、本発明の木構造状のデルタベクトル符号帳１１を用いると、上記（１８），（１９）式より各コードベクトルに対するフィルタ処理を再帰的に行うことが可能となり、初期ベクトルC_0及び（Ｌ－１）種類のデルタベクトル$\Delta C_1 \sim \Delta C_{L-1}$（Ｌ＝１０）に対して合成フィルタ処理を施しておくだけで、それらを極性（＋，－）を切り替えながら加えあわせて行くことで全ての雑音列のコードベクトルに対するフィルタ処理が得られる。

実際には、後述するように第２実施例のデルタベクトル符号帳１１の場合、相互相関R_{xc}、自己相関R_{cc}の演算にあって、全コードベクトルに対するフィルタ演算出力は不要であり、初期ベクトルC_0および（Ｌ－１）種類のデルタベクトル$\Delta C_1 \sim \Delta C_{L-1}$（Ｌ＝１０）に対するフィルタ演算処理結果のみが得られればよいことになる。

したがって、従来における全雑音列のコードベクトル$C_0 \sim C_{1023}$（雑音符号帳１）に対する合成フィルタ演算処理を、初期ベクトルC_0および（Ｌ－１）種類のデルタベクトル$\Delta C_1 \sim \Delta C_{L-1}$（Ｌ＝１０）に対する合成フィルタ演算処理に帰着させることができる。このため、従来、フィルタ処理に

$N_p \cdot N \cdot M$（＝１０２４・N_p・Ｎ）

回の積和算が必要であったものを本実施例において

Ｎ・Ｎ・Ｌ（＝１０・N_p・Ｎ）

回の積和算に減少できる。

次に相互相関R_{xc}の算出について説明する。

合成フィルタ演算出力AC_{2k+1}，AC_{2k+2}を１つ前の合成フィルタ演算出力AC_kと今回のデルタベクトルのフィルタ出力$A\Delta C_i$を用いて（１８）および（１９）式に示すように漸化式で表現すると、相互相関$R_{xc}^{(2k+1)}$，$R_{xc}^{(2k+2)}$は次式

$$R_{xc}^{(2k+1)} = \mathbf{X}^T (AC_{2k+1})$$
$$= \mathbf{X}^T (AC_k) + \mathbf{X}^T (A\Delta C_i)$$
$$= R_{xc}^{(k)} + \mathbf{X}^T (A\Delta C_i) \quad (20)$$
$$R_{xc}^{(2k+2)} = \mathbf{X}^T (AC_{2k+2})$$
$$= \mathbf{X}^T (AC_k) - \mathbf{X}^T (A\Delta C_i)$$
$$= R_{xc}^{(k)} - \mathbf{X}^T (A\Delta C_i) \quad (21)$$

に示すように漸化式で表現できる。したがって、相互相関演算部１２で１階層前の相互相関$R_{xc}^{(k)}$を用いて今回の相互相関$R_{xc}^{(2k+1)}$，$R_{xc}^{(2k+2)}$を演算することができる。このようにすれば、上記（２０），（２１）式の右辺第２項の相互相関演算を行うだけで全雑音列のコードベクトルに対するフィルタ出力と入力音声信号Ｘとの間の相互相関を演算できる。すなわち、従来相互相関を演算するのに

Ｍ・Ｎ（＝１０２４・Ｎ）

回の積和算が必要であったものが、第２実施例によれば、

Ｌ・Ｎ（＝１０・Ｎ）

回の積和算で済ますことが可能となり、演算回数を著しく減少できる。

なお、第１２図において、参照番号６は上記（２０），（２１）式の右辺第２項$\mathbf{X}^T (A\Delta C_i)$を演算する乗算部、

３５は＋１，－１を発生する極性付与部、３６は極性±１を乗算してその右辺第２項に極性を付与する乗算部、１５は１つ前の相互相関$R_{xc}^{(k)}$（右辺第１項）を所定時間記憶遅延する既述の遅延部、１４は上記（２０），（２１）式の右辺第１項と第２項の加算を行って今回の相互相関$R_{xc}^{(2k+1)}$，$R_{xc}^{(2k+2)}$を出力する既述の加算部である。

次に自己相関R_{cc}の算出について説明する。

合成フィルタ演算出力AC_{2k+1}，AC_{2k+2}を、１階層前の合成フィルタ演算出力AC_kと今回のデルタベクトルのフィルタ出力$A\Delta C_i$を用いて上記（１８），（１９）式に示すように漸化式で表現すると、各雑音列のコードベクトルに対する自己相関R_{cc}は次式で表現される。すなわち、

$$R_{cc}^{(0)} = (AC_0)^T (AC_0)$$

$$AC_1 = AC_0 + A\Delta C_1$$

$$AC_2 = AC_0 - A\Delta C_1$$

$$R_{cc}^{(1)} = (AC_0)^T (AC_0) + (A\Delta C_1)^T (A\Delta C_1) + 2(AC_0)^T (A\Delta C_1)$$

$$R_{cc}^{(2)} = (AC_0)^T (AC_0) + (A\Delta C_1)^T (A\Delta C_1) - 2(AC_0)^T (A\Delta C_1)$$

$$AC_3 = AC_1 + A\Delta C_2 = AC_0 + A\Delta C_1 + A\Delta C_2$$

$$AC_4 = AC_1 - A\Delta C_2 = AC_0 + A\Delta C_1 - A\Delta C_2$$

$$AC_5 = AC_2 + A\Delta C_2 = AC_0 - A\Delta C_1 + A\Delta C_2$$

$$AC_6 = AC_1 - A\Delta C_2 = AC_0 - A\Delta C_1 - A\Delta C_2$$

$$R_{cc}^{(3)} = (AC_0)^T (AC_0) + (A\Delta C_1)^T (A\Delta C_1) + (A\Delta C_2)^T (A\Delta C_2) + 2(AC_0)^T (A\Delta C_1) + 2(A\Delta C_1)^T (A\Delta C_2) + 2(A\Delta C_2)^T (AC_0)$$

$$R_{cc}^{(4)} = (AC_0)^T (AC_0) + (A\Delta C_1)^T (A\Delta C_1) + (A\Delta C_2)^T (A\Delta C_2) + 2(AC_0)^T (A\Delta C_1) - 2(A\Delta C_1)^T (A\Delta C_2) - 2(A\Delta C_2)^T (AC_0)$$

$$R_{cc}^{(5)} = (AC_0)^T (AC_0) + (A\Delta C_1)^T (A\Delta C_1) + (A\Delta C_2)^T (A\Delta C_2) - 2(AC_0)^T (A\Delta C_1) - 2(A\Delta C_1)^T (A\Delta C_2) + 2(A\Delta C_2)^T (AC_0)$$

$$R_{cc}^{(6)} = (AC_0)^T (AC_0) + (A\Delta C_1)^T (A\Delta C_1) + (A\Delta C_2)^T (A\Delta C_2) - 2(AC_0)^T (A\Delta C_1) + 2(A\Delta C_1)^T (A\Delta C_2) - 2(A\Delta C_2)^T (AC_0) \quad (22)$$

と表現され、一般に

$$R_{cc}^{(2k+1)} = R_{cc}^{(k)} + (\Delta AC_i)^T (\Delta AC_i) + 2\Delta AC_i \cdot AC_k \quad (23)$$

$$R_{cc}^{(2k+2)} = R_{cc}^{(k)} + (\Delta AC_i)^T (\Delta AC_i) - 2\Delta AC_i \cdot AC_k \quad (24)$$

と表現できる。

すなわち、1階層前の自己相関$R_{cc}^{(k)}$に、今回のΔAC_iの自己相関$(\Delta AC_i)^T (\Delta AC_i)$を加え合わせると共に、$\Delta AC_i$と$AC_0$，$\Delta AC_i$～$\Delta AC_{i-1}$の相互相関を、極性（＋，－）を切り替えながら加え合わせることにより相互相関$R_{cc}^{(2k+1)}$，$R_{cc}^{(2k+2)}$を演算できる。このようにすれば、自己相関R_{cc}を初期ベクトルのフィルタ出力AC_0と（L－1）種類のデルタベクトルのフィルタ出力ΔAC_1～ΔAC_{L-1}の合計L通りの自己相関$(AC_0)^2$，$(\Delta AC_1)^2$～$(\Delta AC_{L-1})^2$と、各フィルタ出力AC_0，ΔAC_1～ΔAC_{L-1}間相互の$(L^2-1)/2$通りの相互相関を用いて演算できる。すなわち、従来自己相関を演算するのに

M・N（＝1024・N）

回の積和算が必要であったものが、

L（L＋1）・N／2（＝55・N）

回の積和算で済ますことが可能となり、演算回数を著しく減少できる。なお、第12図において、32は上記（23），（24）式の右辺第2項の自己相関

$(\Delta AC_i)^T (\Delta AC_i)$を演算する自己相関演算器、33は該（23），（24）式における各相互相関を演算する相互相関演算器、34は各相互相関を、所定の極性（＋，－）

を付して加算する相互相関合成部、１６は１階層前の自己相関$R_{cc}^{(k)}$と自己相関$(\Delta\Delta C_i)^T(\Delta\Delta C_i)$および相互相関を加算して（２３），（２４）式の演算を行う既述の加算部、１７は１階層前の自己相関$R_{cc}^{(k)}$を所定時間記憶して遅延する既述の遅延部である。

最後に第１２図の回路全体の動作について説明する。

予め１つの基準雑音列である初期ベクトルC_0と（Ｌ－１）種類の差分雑音列であるデルタベクトル$\Delta C_1 \sim \Delta C_{L-1}$（Ｌ＝１０）をデルタベクトル符号帳１１に格納しておき、線形予測合成（ＬＰＣ）フィルタ３において初期ベクトルC_0および（Ｌ－１）種類のデルタベクトル$\Delta C_1 \sim \Delta C_{L-1}$（Ｌ＝１０）に対して、合成フィルタ処理を施してフィルタ出力AC_0，$A\Delta C_1 \sim A\Delta C_{L-1}$（Ｌ＝１０）を求めて記憶部３１に記憶する。

かかる状態で、ｉ＝０とし、相互相関演算部１２で相互相関

$R_{xc}^{(0)}$（$=X^T AC_0$）を演算し、自己相関演算部１３で自己相関

$R_{cc}^{(0)}$（$=(AC_0)^T(AC_0)$）を演算し、これら相互相関と自己相関を用いて演算部３８で、上記（１４）式を用いてＦ（**Ｘ**，**Ｃ**）（$=R_{xc}^2 / R_{cc}$）を演算する。

誤差電力評価決定部１０は演算された演算値Ｆ（**Ｘ**，**Ｃ**）とそれ迄のＦ（**Ｘ**，**Ｃ**）の最大値F_{max}（初期値は０）を比較し、Ｆ（**Ｘ**，**Ｃ**）＞F_{max}であれば、Ｆ（**Ｘ**，**Ｃ**）→F_{max}としてF_{max}を更新すると共に、F_{max}を与える１つのコー

ドベクトルを特定する符号（インデックス）をもってそれ迄の符号を更新する。

2^i （$=2^0$）個のコードベクトルに対して上記処理を行えばi＝1とし、上記（２０）式（但し、k＝0，i＝1）に従って相互相関を演算し、上記（２３）式に従って自己相関を演算し、これら相互相関と自己相関を用いて演算部３８で上記（１４）式を演算する。

誤差電力評価決定部１０は演算された演算値F（**X**，**C**）とそれ迄のF（**X**，**C**）の最大値F_{max}（初期値は０）を比較し、F（**X**，**C**）＞F_{max}であれば、F（**X**，**C**）→F_{max}としてF_{max}を更新すると共に、該F_{max}を与える１つのコードベクトルを特定する符号（インデックス）をもってそれ迄の符号を更新する。

ついで、上記（２１）式（但し、k＝0，i＝1）に従って相互相関を演算し、（２４）式に従って自己相関を演算し、これら相互相関と自己相関を用いて演算部３８で上記（１４）式を演算する。

誤差電力評価決定部１０は同様に演算された演算値F（**X**，**C**）とそれ迄のF（**X**，**C**）の最大値F_{max}（初期値は０）を比較し、F（**X**，**C**）＞F_{max}であれば、F（**X**，**C**）→F_{max}としてF_{max}を更新すると共に、該F_{max}を与える１つのコードベクトルを特定する符号（インデックス）をもってそれ迄の符号を更新する。

2^i （$=2^1$）個のコードベクトルに対して上記処理を行えばi＝2とし、以上と同様の処理を繰返し、2^{10}個の全コー

ドベクトルに対して上記処理を行えば、音声符号化部３０は誤差電力評価決定部１０に記憶されている最新の符号（インデックス）を入力音声信号に対する音声符号化情報として出力する。

次に前述した第２実施例の変形に相当する変形第２実施例について説明する。前述した第２実施例では、初期ベクトルC_0と数少ない（Ｌ－１）個のデルタベクトル（ΔC_i）とを保持しておくだけで仮想的に全てのコードベクトルが再生されるので演算量の減少、さらには符号帳のメモリ容量の削減に有効である。

ところが、デルタベクトル符号帳１１の各ベクトルの成分について見てみると、上記（１５）式で示すように、初期ベクトルであるC_0の成分が全てのベクトルに含まれるのに対し、最下位の階層の成分である第９番目のデルタベクトルΔC_9の成分はその半分の５１２本のベクトルにしか含まれない（第１３図参照）。すなわち、各デルタベクトルが符号帳１１の構成に与える寄与は均等でなく、木構造の配列の上位の階層を構成するデルタベクトル、例えば、初期ベクトルC_0、第１デルタベクトルΔC_1等ほど、多くのコードベクトルにその成分として含まれ、符号帳の分布のモードを決定しているといえる。

第１４Ａ、１４Ｂおよび１４Ｃ図は符号帳内に仮想的に形成されるコードベクトルの分布（モードＡ、モードＢおよびモードＣ）をそれぞれ表す図である。例えば３つのベクトル、すなわちC_0，ΔC_1およびΔC_2について考察すると、こ

れらのベクトルの分布は6通り（モードA～モードF）存在する。このうちモードA～モードCについてそれぞれ第14A図～第14C図に表す。図中、e_x，e_yおよびe_zは、三次元を構成するx軸、y軸およびz軸の各方向の単位ベクトルをそれぞれ表す。残りのモードD，EおよびFは、各ベクトルに対し次のように単位ベクトルを割り当てたときに相当する

モードD：$C_0 = e_x$，$\Delta C_1 = e_z$，$\Delta C_2 = e_y$

モードE：$C_0 = e_y$，$\Delta C_1 = e_z$，$\Delta C_2 = e_x$

モードF：$C_0 = e_z$，$\Delta C_1 = e_x$，$\Delta C_2 = e_y$

かくして、デルタベクトルとして与えるベクトルの順番によって、異なるモードの分布を持ったデルタベクトル符号帳11が存在することが分かる。つまり、デルタベクトルの順番を、第13図に示すように常に固定的に割り付けたとすると、常に一定のモードに片寄ったコードベクトルしか再生できないことになり、ベクトル量子化の対象となる入力音声信号Xに対して必ずしも最適な音声符号化を行っているという保証はない。すなわち、量子化歪みが増大してしまうおそれがある。

そこで本発明の変形第2実施例では、初期ベクトルC_0およびデルタベクトルΔCとして与える合計L本のベクトルの順番を並べ替えることによって、符号帳11内に仮想的に形成されるコードベクトルの分布のモードを調整するようにする。つまり符号帳の性質を変化させる。

さらに上記のコードベクトルの分布のモードを、符号化す

べき入力音声信号の特性に合わせて調整する。これにより、再生音声の品質は一層改善される。

この場合、線形予測合成（ＬＰＣ）フィルタ３の特性に応じて、フレーム毎にベクトルの並べ替えを行う。このようにすると、音声符号化データを受信した側、すなわち復号器側では、符号器側から特別な調整情報を送ることなしに、符号器側で行ったのと全く同じ調整（ベクトルの並べ替え）をすることができる。

上記のベクトルの並べ替えを行うに当り、初期ベクトルおよび各デルタベクトルに対し線形予測合成フィルタ処理を施して得られるベクトルのフィルタ出力のパワーを評価し、このパワーがそのフィルタ処理前のパワーに比べて最も増大しているベクトルから順次、初期ベクトル、第１デルタベクトル、第２デルタベクトル…という順番に並べ替えをするのが具体的な一例である。

上記の並べ替えにおいて、初期ベクトルおよび各デルタベクトルが線形予測合成フィルタ処理した後に相互に直交するように、予めこれらベクトルを変換しておく。これにより符号帳１１内に仮想的に形成される各ベクトルを、超平面（Ｈｙｐｅｒ　Ｐｌａｎｅ）上に均一に分布させることができる。

また上記の並べ替えにおいて、初期ベクトルおよび各デルタベクトルのパワーを正規化（ｎｏｒｍａｌｉｚｅ）しておくのが好ましい。これにより各ベクトルのフィルタ出力が有するパワーを単純に比較するだけで、その並べ換えができる。

そして、受信側へ音声符号化情報を送出するに際し、木構

造状のベクトル配列のうち上位の階層に属するベクトル間の符号間距離（ベクトルのユークリッド距離）の方が、下位の階層に属するベクトル間の符号間距離の方よりも大きくなるように、該音声符号化情報の符号を割り付ける。これは、上位の階層に属するベクトル（初期ベクトルおよび第1デルタベクトル等）程、受信側で復号して得た再生音声の品質に大きな影響を与えることに着目したものである。これにより、受信側に至る伝送路上で伝送エラーが生じても、再生音声の品質の劣化を小さく抑えることができる。

第15A、15Bおよび15C図は変形第2実施例に基づくベクトルの並べ替えを説明するための図である。第15A図において、座標の原点を中心とする球体（ハッチング）は単位ベクトルe_x，e_yおよびe_zによって規定される全ベクトルの空間であり、仮に初期ベクトルC_0に単位ベクトルe_xが割り付けられ、また第1デルタベクトルΔC_1および第2ベクトルΔC_2に単位ベクトルe_yおよびe_zが割り付けられたとすると、これらによって規定される平面は、上記球体上のC_0点における法線を含む平面となる。これはモードA（第14A図）に相当する。

この第15A図に示すベクトル$C_0(=e_x)$，ΔC_1（$=e_y$）および$\Delta C_2(=e_z)$に対し、線形予測合成フィルタ（A）処理を施すと、通常、これらのフィルタ出力$A(e_x)$，$A(e_y)$および$A(e_z)$はx，yおよびz軸の各方向に一様ではなくなり、ある歪みを有する。この様子を示したのが第15B図であり、同図中の下側に示すような不等式が成

立する場合のベクトル分布を示している。つまり、線形予測合成フィルタ３を通すことによってある歪をもった増幅が行われる。

このように線形予測合成フィルタ３の特性Ａが、デルタベクトル符号帳１１の構成要素である各ベクトルに対して異なる振幅増幅特性を示すので、符号帳１１内に仮想的に形成される全ベクトルをベクトル空間全体に亘って均等に分布させるよりも不均一にした方がよい。そこで、どの方向のベクトル成分が最も大きく増幅されるかを調べ、その方向のベクトル成分の分布を増やしてやれば、符号帳１１に効率よくベクトルを格納することができ、結果として音声信号の量子化特性が改善されることになる。

既に述べたように、木構造状のデルタベクトルの分布には片寄りがあるが、デルタベクトルの順番を並べ換えることで、符号帳１１の性質が変えられる。

第１５Ｃ図を参照すると、第１５Ｂ図に示すようにフィルタ処理後のパワーの増幅率が片寄っていた場合、パワーの大きいデルタベクトル（ΔC_2）から順に並べ替えを行い、その後、改めて木構造状の配列に従って、符号帳ベクトルを生成する。このようにしてできたデルタベクトル符号帳１１を用いて符号化を行うことにより、前述した第２実施例のように固定的にデルタベクトルを割り付けて配列するのに比べて再生音声の品質改善が図れる。

第１６図は変形第２実施例に基づく符号帳探索処理の部分の一例を示す図であり、第１５Ａ、１５Ｂおよび１５Ｃ図に

表した並べ替えを行う一例を示す。これは前述の第１２図（第２変実施例）の構成に変形を加えたものに相当し、第１２図の構成に比べ、第１６図ではパワー評価部４１とソーティング部４２が記憶部３１に対して共働的に付加されている。パワー評価部４１は、線形予測合成フィルタ３でフィルタ処理を施した後の初期ベクトルおよび各デルタベクトルのパワーを評価する。この評価の結果得られた各ベクトルの振幅増幅率の大小に基づいて、ソーティング部４２において、各ベクトルの順番を並べ替える。これらパワー評価部４１とソーティング部４２を、前述の第１４Ａ～１４Ｃ図および第１５Ａ図～第１５Ｃ図に即して説明すると次のとおりである。

パワー評価部４１

デルタベクトル符号帳１１に格納されている各ベクトル（C_0，ΔC_1，ΔC_2）に対して線形予測合成フィルタ処理を施して得たベクトル（AC_0，$A\Delta C_1$，$A\Delta C_2$）のパワー（電力）を算出する。この際、前述したように、各ベクトルのパワーを予め正規化しておけば（下記（１）参照）、フィルタ処理後のパワーを直接比較することが、そのまま各ベクトルに対する振幅増幅率の比較をすることになる（下記（２）参照）。

（１）デルタベクトルの正規化：$e_x = C_0 / |C_0|$，

$e_y = \Delta C_1 / |\Delta C_1|$，

$e_z = \Delta C_2 / |\Delta C_2|$

$|e_x|^2 = |e_y|^2$

$= |e_z|^2$

（２）
- ベクトルC_0に対する振幅増幅率：$|AC_0|^2 / |C_0|^2 = |Ae_x|^2$
- ベクトルΔC_1に対する振幅増幅率：$|A\Delta C_1|^2 / |\Delta C_1|^2 = |Ae_y|^2$
- ベクトルΔC_2に対する振幅増幅率：$|A\Delta C_2|^2 / |\Delta C_2|^2 = |Ae_z|^2$

ソーティング部４２

各ベクトルの合成フィルタ（Ａ）による振幅増幅率をパワー評価部４１から受け、増幅率の大きい順に並べ替え（ソーティング）を行う。この並べ替えにより、増幅率の大きいものから順に初期ベクトル（C_0）、第１デルタベクトル（ΔC_1）、第２デルタベクトル（ΔC_2）…というように新たなデルタベクトルが設定される。以降の符号化処理は、得られたデルタベクトルによって構成される木構造状デルタ符号帳１１を用いて、第１２図の木構造状デルタ符号帳の場合と全く同様に行われる。以下に、第１５Ａ～１５Ｃ図で示した場合についてのソーティング処理を示す。

（ソーティング）

$|Ae_z|^2 > |Ae_x|^2 > |Ae_y|^2$

（並べ替え）

$C_0 = e_z$,　　$\Delta C_1 = e_x$,　　$\Delta C_2 = e_y$

上述した第２実施例および変形第２実施例は、既述の第１

実施例の場合と同様、逐次最適化ＣＥＬＰ形の音声符号器および同時最適化ＣＥＬＰ形の音声符号器あるいはピッチ直交変換最適化ＣＥＬＰ形の音声符号器等のいずれにも適用できる。適用の仕方は、第１実施例で詳しく説明した巡回加算手段２０（１４，１５；１６，１７；１４－１，１５－１；１４－２，１５－２）を用いるのと同じ手法である。

以下、参考のために上述した各種の形式の音声符号器について説明しておく。

第１７図は逐次最適化ＣＥＬＰ形の符号器を示す図、第１８図は同時最適化ＣＥＬＰ形の符号器を示す図である。なお、既に述べた構成要素と同等のものには同一の参照番号あるいは記号を付して示す。

第１７図において、適応符号帳１０１には１サンプルずつピッチ周期が遅延されたＮサンプルに対応するＮ次元のピッチ予測残差ベクトルが格納されており、また符号帳１には前述のとおり同様のＮサンプルに対応するＮ次元の雑音列を用いて生成したコードベクトルが2^mパターンだけ予め設定されている。好ましくは、各コードベクトルのサンプルデータのうち、その振幅が一定の閾値以下のサンプルデータ（例えばＮサンプルのうちのＮ／４サンプル）は０で置き換えられている。このような符号帳をスパース（間引き）符号帳と呼ぶ。

まず、適応符号帳１０１の各ピッチ予測残差ベクトルＰに$A=1/A'(z)$(但し、$A'(z)$は聴覚重み付け線形予測分析フィルタを示す）で示される聴覚重み付け線形予測合

成フィルタ１０３で聴覚重み付けして生成されたピッチ予測ベクトルＡＰに増幅器１０５でゲインｂを乗算してピッチ予測再生信号ベクトルｂＡＰを生成する。

そして、このピッチ予測再生信号ベクトルｂＡＰと、Ａ（ｚ）／Ａ′（ｚ）(但し、Ａ（ｚ）は線形予測分析フィルタを示す）で示される聴覚重み付けフィルタ１０７で聴覚重み付けされた入力音声信号ベクトルＡＸとの間の聴覚重み付けされたピッチ予測誤差信号ベクトルＡＹを減算部１０８で求め、このピッチ予測誤差信号ベクトルＡＹの電力が最小の値になるように評価部１１０がフレーム毎に下記式

$$|AY|^2 = |AX - bAP|^2 \qquad (25)$$

により、符号帳１０１の中から最適なピッチ予測残差ベクトルＰを選択すると共に最適なゲインｂを選択する。

さらに、前述したとおり、符号帳１の各コードベクトルＣにも同様にして線形予測合成フィルタ３で聴覚重み付けして生成された聴覚重み付け再生後のコードベクトルＡＣに増幅器２でゲインｇを乗算して線形予測再生信号ベクトルｇＡＣを生成する。なお、増幅器２の位置はフィルタ３の前段でも良い。

そして、この線形予測再生信号ベクトルｇＡＣと、上記のピッチ予測誤差信号ベクトルＡＹとの誤差信号ベクトルＥを誤差発生部４で求め、この誤差信号ベクトルＥの電力が

$$|E|^2 = |AY - gAC|^2 \qquad (26)$$

により最小の値になるように評価部５がフレーム毎に符号帳１の中から最適なコードベクトルＣを選択すると共に最適な

ゲインｇを選択する。

なお、適応符号帳１０１の適応化は、ｂＡＰ＋ｇＡＣを加算部１１２で求め、これを聴覚重み付け線形予測分析フィルタ（Ａ′（ｚ））１１３でｂＰ＋ｇＣに分析し、さらに遅延器１１４で１フレーム分遅延させたものを次のフレームの適応符号帳（ピッチ予測符号帳）として格納することにより行われる。

このように、第１７図に示した逐次最適化ＣＥＬＰ形符号器では、ゲインｂとｇが別々に制御されるのに対し、第１８図に示した同時最適化ＣＥＬＰ形符号器では、ｂＡＰとｇＡＣとを加算部１１５で加算してＡＸ′＝ｂＡＰ＋ｇＡＣを求め、さらに誤差発生部４でフィルタ１０７からの聴覚重み付けされた入力音声信号ベクトルＡＸとの間の誤差信号ベクトルＥを上記のようにして求め、評価部５がこのベクトルＥの電力を最小にするコードベクトルＣを符号帳１から選択すると共に最適なゲインｂとｇを同時に選択制御するものである。

この場合には、上記の（２５），（２６）式より、

$$|\mathbf{E}|^2 = |\mathbf{AX} - b\mathbf{AP} - g\mathbf{AC}|^2 \qquad (27)$$

となる。

なお、この場合の適応符号帳１０１の適応化は、第１７図の加算部１１２の出力に相当するＡＸ′に対して同様にして行われる。

以上の第１７図及び第１８図に概念的に示されたゲインｂ，ｇは実際には、第１９図および第２０図に示すようにそれぞ

れのＣＥＬＰ方式において符号帳１のコードベクトルＣについて最適化を行う。

すなわち、第１７図の場合には、上記の（２６）式において、ベクトルＥの電力を最小にするためのゲインｇを偏微分により求めると、

$$0 = \delta (|AY - gAC|^2) / \delta g$$

$$= 2(-AC)^T (AY - gAC)$$

より、

$$g = (AC)^T AY / (AC)^T AC \qquad (28)$$

となる。

そこで第１９図においては、ピッチ予測誤差信号ベクトルＡＹと、符号帳１の各コードベクトルＣを聴覚重み付け線形予測合成フィルタ３を通して得られるコードベクトルＡＣとを乗算部６で乗算して両者の相関値$(AC)^T AY$を発生し、聴覚重み付け再生後のコード・ベクトルＡＣの自己相関値$(AC)^T AC$を自己相関演算部８で求める。

そして、評価部５では、両相関値$(AC)^T AY$および$(AC)^T AC$に基づいて上記の（２８）式により該ピッチ予測誤差信号ベクトルＡＹに対する誤差信号ベクトルＥの電力を最小にする最適なコードベクトルＣおよびゲインｇを選択する。

なお、上記の（２６）式を最小とするように各コードベクトルＣに対してゲインｇが求められ、このゲインに対する量子化がオープン・ループで行われるなら、次式、

$$((AY)^T AC)^2 / (AC)^T AC$$

を最大にすることと等価になる。

また、第１８図の場合には、上記の（２７）式において、ベクトルＥの電力を最小にするためのゲインｂ，ｇを偏微分により求めると、

$$g = \{ (AP)^T AP (AC)^T AX - (AC)^T AP (AP)^T AX \} / \nabla$$

$$b = \{ (AC)^T AC (AP)^T AX - (AC)^T AP (AC)^T AX \} / \nabla \quad (29)$$

となる。但し、

$$\nabla = (AP)^T AP (AC)^T AC - ((AC)^T AP)^2$$

である。

そこで第２０図においては、聴覚重み付けされた入力音声信号ベクトルＡＸと、符号帳１の各コードベクトルＣを聴覚重み付け線形予測合成フィルタ３を通して得られるコードベクトルＡＣとを乗算部６－１で乗算して両者の相関値$(AC)^T AX$を発生し、聴覚重み付けされたピッチ予測ベクトルＡＰと、コードベクトルＡＣとを乗算部６－２で乗算して両者の相関値$(AC)^T AP$を発生すると共に、コードベクトルＡＣの自己相関値$(AC)^T AC$を自己相関演算部８で求める。

そして、評価部５では、これらの相関値$(AC)^T AX$，$(AC)^T AP$および$(AC)^T AC$に基づいて上記の（２９）式により聴覚重み付け入力音声信号ベクトルＡＸに対する誤差信号ベクトルＥの電力を最小にする最適なコードベクトルＣおよびゲインｂ，ｇを同時に選択する。

この場合も、ベクトルＥの電力を最小にすることは、相関値の比率、

$$2b(AP)^T AX - b^2(AP)^T AP + 2g(AC)^T AX - g^2(AC)^T AC - 2bg(AP)^T AC$$

を最大にすることと等価である。

このように、逐次最適化ＣＥＬＰ方式の場合には、同時最適化ＣＥＬＰ方式に比べて全体の演算量が少なくて済むが、符号化された音声の品質は劣化したものとなる。

第２１Ａ図は逐次最適化ＣＥＬＰ方式の場合におけるゲイン最適化動作を図解的に表すベクトル図、第２１Ｂ図は同時最適化ＣＥＬＰ方式の場合におけるゲイン最適化動作を図解的に表すベクトル図、第２１Ｃ図はピッチ直交変換最適化ＣＥＬＰ方式の場合におけるゲイン最適化動作を図解的に表すベクトル図である。

第２１Ａ図の逐次最適化方式の場合には、最適化されたベクトル$AX' = bAP + gAC$を得るには比較的少ない演算量で済むが、このベクトルAX'と入力ベクトルAXとは誤差が生じ易く音声の再生品質が悪くなる。

また、第２１Ｂ図の同時最適化方式は、二次元の場合には図示のように$AX' = AX$となることから、一般に同時最適化方式は逐次最適化方式に比べて音声の再生品質は良くなるが、上記の（２９）式に示すように演算量が多くなってしまうという問題がある。

そこで、本出願人は、このような逐次最適化／同時最適化

ＣＥＬＰ形の各音声符号化方式の長所を生かして音声の再生品質と演算量の両者において満足の行く符号化・復号化を実現するため、第２２図に示すような符号器を既に特許出願している（特願平２－161041号）。

すなわち、ピッチ周期については従来と同様にピッチ予測残差ベクトルＰとゲインｂの評価・選択を行うが、コードベクトルＣおよびゲインｇについては、重み付け直交変換部５０を設け、符号帳１の各コードベクトルＣを、該聴覚重み付けされた各ピッチ予測残差ベクトルの内の最適時のピッチ予測残差ベクトルＡＰに直交した聴覚重み付け再生コードベクトルＡＣ′に変換することによって行っている。

これをさらに第２１Ｃ図で説明すると、上記のように聴覚重み付けされたピッチ予測再生ベクトルｂＡＰに対して、符号帳１から取り出されて聴覚重み付けマトリクスＡが施されたコードベクトルＡＣが直交していないことが、第２１Ａ図のように逐次最適化方式において量子化誤差εを増大させる原因になっていることに鑑み、聴覚重み付けされたコードベクトルＡＣが聴覚重み付けされたピッチ予測残差ベクトルＡＰに直交するコードベクトルＡＣ′に、既知の手法により、直交変換すれば第２１Ａ図の逐次最適化ＣＥＬＰ方式においても量子化誤差を同時最適化方式と同程度まで減少させることが出来る。

そして、このようにして得られたコードベクトルＡＣ′にゲインｇを乗じて線形予測再生信号ｇＡＣ′を生成し、線形予測再生信号ｇＡＣ′と聴覚重み付けされた入力音声信号ベ

クトルＡＸとから線形予測誤差信号ベクトルＥを最小にするコードベクトルを評価部５が符号帳１から選択し、ゲインｇを選択している。

なお、符号帳探索時のフィルタ演算量を削減するために、符号帳として白色雑音の雑音列からなり、サンプル値としてゼロを多く挿入したスパース化雑音符号帳を用いるのが好ましい。この他にも、コードベクトル同士がオーバーラップしているオーバーラップ符号帳等を用いることができる。

第２３図は第１実施例のもとでの符号帳探索処理の部分をさらに別の例をもって具体的に示す図であり、上述したピッチ直交変換最適化ＣＥＬＰ形の音声符号器に適用した場合を示す。この場合にも本発明は何の支障もなく適用される。

この第２３図は、第９図に示した構成に第１０図の自己相関演算部１３を組み合わせた一例を示している。また第９図に示した演算手段１９′は第６図の演算手段１９と同様に転置マトリクスA^Tにより構成することができるが、この例では時間反転型のフィルタで構成されている。

本図の自己相関演算部６０は、演算部６０ａ～６０ｅで構成されており、演算部６０ａでは演算手段１９′と同様にして入力信号である最適時の聴覚重み付けされたピッチ予測残差ベクトルＡＰを時間反転聴覚重み付けして演算補助ベクトル$V = A^T AP$を生成する。

このベクトルＶは、デルタベクトル符号帳１１の全てのデルタベクトルΔＣに直交したベクトルＤを入力し聴覚重み付けフィルタ（Ａ）処理を施す演算部６０ｂにおいて３つのベ

クトルBとuBとABとに変換される。

このうちのベクトルBとuBは時間反転直交変換部71に送られて演算手段70から出力された$A^T AX$に対して時間反転したハウス・ホルダー直交変換を行って$H^T A^T AX = (AH)^T AX$を生成する。

ここで、変換部71における時間反転したハウス・ホルダー変換H^Tについて説明する。

まず、第24A図および第24B図を用いてハウス・ホルダー変換自体を説明すると、演算補助ベクトルVを、点線で示す折り返し線を境にしてベクトルDの平行成分として折り返したとき、$(|V|/|D|)D$なるベクトルが得られる。なお、$D/|D|$はD方向の単位ベクトルを示す。

このようにして得られたD方向ベクトルを−D方向、すなわち逆方向に$-(|V|/|D|)D$として図示のように取る。この結果、Vとの加算で得られるベクトル$B = V - (|V|/|D|)D$は折り返し線に直交することになる（第24B図参照）。

そして、このベクトルBにおけるベクトルCの成分を求めると、第24A図の場合と同様にして、ベクトル$\{(C^T B)/(B^T B)\}B$が得られる。

このベクトルと反対方向の2倍のベクトルをとってベクトルCに加えると、Vに直交したベクトルC′が得られることになる。即ち、

$$C' = C - 2B\{(C^T B)/(B^T B)\}B \quad (30)$$

となる。

この（３０）式において、ｕ＝２／$\mathbf{B}^T\mathbf{B}$とすると、

$$\mathbf{C}' = \mathbf{C} - \mathbf{B}(u\mathbf{B}^T\mathbf{C}) \qquad (31)$$

となる。

一方、$\mathbf{C}' = H\mathbf{C}$であるから、（３１）式は、

$$H = \mathbf{C}'\mathbf{C}^{-1}$$

$$= I - \mathbf{B}(u\mathbf{B}^T) \qquad (Iは単位ベクトル)$$

となる。したがって、

$$H^T = I - (u\mathbf{B})\mathbf{B}^T$$

$$= I - \mathbf{B}(u\mathbf{B}^T)$$

となり、これはＨと同じである。

したがって、変換部７１の入力ベクトル$A^T A\mathbf{X}$を例えば$\mathbf{W}$と置くと、

$$H^T\mathbf{W} = \mathbf{W} - (\mathbf{W}\mathbf{B})(u\mathbf{B}^T)$$

$$= (AH)^T A\mathbf{X}$$

となり、図示のような演算構成となる。なお、図中、○で示した箇所はベクトル演算を表し、△で示した箇所はスカラー演算を表している。

直交変換方法としては、この他にグラム・シュミット法等も既に知られている手法である。

そして、このベクトル$(AH)^T A\mathbf{X}$に符号帳１１からのデルタベクトルΔＣを乗算部６５で乗算すると、

$$R_{xc} = (\Delta C)^T (AH)^T A\mathbf{X}$$

$$= (AH\Delta C)^T A\mathbf{X}$$

なる相関値が得られ、これを巡回加算部６７（巡回加算手段２０）で巡回的に加算することにより$(AHC)^T A\mathbf{X}$が評

価部５に送られることになる。

これに対し、演算部６０ｃでは、入力ベクトルＡＢおよびｕＢから直交変換マトリクスＨおよび時間反転直交変換マトリクスH^Tを求め、これにさらにFinite Impulse Response (FIR) 聴覚重み付けフィルタマトリクスＡを組み込んで演算手段７０と変換部７１とによる時間反転聴覚重み付け直交変換マトリクスＡＨの自己相関マトリックスＧ＝（ＡＨ）^{T}ＡＨをフレーム毎に生成しておく。

そして、このようにして求めた自己相関マトリクスＧ＝（ＡＨ）^{T}ＡＨを第１０図に示したように演算部６０ｄに記憶しておき、この演算部６０ｄに符号帳１１からのデルタベクトルΔＣが与えられたときに、

$$(\Delta C_i)^T G C_{i-1} + (\Delta C_i)^T G \Delta C_i$$

が得られ、これを巡回加算部６０ｅ（巡回加算手段２０）で前回の自己相関値$(AHC_{i-1})^T AHC_{i-1}$と巡回的に加算することにより、$(AHC_i)^T AHC_i$なる今回の自己相関値を求めることができ、評価部５に送られる。

このように評価部５に送られる２つの相関値に基づいて最適なデルタベクトルとゲインとを選択することができる。

最後に、本発明の第１実施例および第２実施例によって得られる利益について数値例をもって説明する。

第２５図は本発明に係る第１実施例によって演算量を減少させることができることを示す図である。同図の（ａ）欄は逐次最適化ＣＥＬＰ形の符号器の場合であって、雑音符号帳として

（１）従来の４／５スパース符号帳、

（２）従来のオーバーラップ符号帳、

（３）本発明に係る第１実施例に基づくデルタベクトル符号帳、

をそれぞれ用いた場合の演算量を示している。

第２５図中のＮはサンプル数、Ｎｐはフィルタ３の次数である。また、演算量を算出する対象は種々あるが、符号器の中で極めて膨大な演算を必要とする、①フィルタ処理演算、②相互相関演算および③自己相関演算の３種のみを対象として示す。

具体的にサンプル数ＮをＮ＝１０とすると、合計演算量は同図の右端に示すように、従来例（１）で４３２Ｋ積和算、従来例（２）で８４Ｋ積和算である。これに対し、第１実施例によれば２８Ｋ積和算と大幅に減少している。

第２５図の（ｂ）欄および（ｃ）欄はそれぞれ同時最適化ＣＥＬＰ形の符号器およびピッチ直交変換最適化ＣＥＬＰ形の符号器の場合であり、（ａ）欄の場合と全く同様に３種の符号帳を用いた各ケースについて演算量を算出している。いずれにしても本発明に係る第１実施例を適用した場合には、３０Ｋ積和算あるいは２８Ｋ積和算と大幅に演算量が減少していることが分かる。

第２６図は本発明に係る第２実施例によって演算量を減少させると共にメモリ量を削減できることを示す図である。本図の（ａ）欄は演算量について、（ｂ）欄は雑音符号帳のメモリー量についてそれぞれ示す。

各コードベクトルのサンプル数Ｎは標準的なＮ＝４０とする。また符号帳のサイズＭとしては、従来方式において標準的なＭ＝１０２４とし、これに対して本発明の第２実施例ではサイズＭがＭ＝Ｌに減少し、具体的にはＬ＝１０とした。このＬは第１１図の上方に示した階層の数１，２，３…Ｌと同じである。

いずれにしても演算量の合計で見ると、従来方式が４８０Ｋ積和算（９６M_{ops}）を要するのに対し、本発明の第２実施例では６．６Ｋ積和算（１．３M_{ops}）と約１／７０に激減している。

さらに第２６図のメモリー量((ｂ）欄）について見ると、１／１００に激減している。

また前記の変形第２実施例においても、演算量の大半を占めるフィルタ処理演算、自己相関演量および相互相関演算の総演算量は第２６図に示した値と同様激減する。

かくのごとく、本発明の第１実施例によれば、雑音符号帳が格納すべきコードベクトルとして、隣接コードベクトル間の差分ベクトル（デルタベクトル）を採用する。この結果、演算量は従来よりも一層減少する。

そして本発明の第２実施例では、上記第１実施例にさらに改良を加えており、

（ｉ）従来、フィルタ処理にN_p・Ｎ・Ｍ（＝１０２４・N_p・Ｎ）回の積和算が必要であったものをＮ・Ｎ・Ｌ（＝１０・N_p・Ｎ）回の積和算に減少でき、

（ii）容易に誤差電力が最小となるコードベクトルを求め

ることができ、

（iii）従来相互相関を演算するのにM・N（＝１０２４・N）回の積和算が必要であったものが、L・N（＝１０・N）回の積和算で済ますことが可能となって、演算回数を著しく減少でき、

（iv）従来自己相関を演算するのにM・N（＝１０２４・N）回の積和算が必要であったものが、L（L＋１）・N／２（＝５５・N）回の積和算で済ますことが可能となって、演算回数を著しく減少でき、

（v）メモリー容量を大幅に削減できる。

また変形第２実施例によれば、さらに加えて再生音声の品質を向上できる。

〔産業上の利用分野〕

例えば携帯電話や自動車電話における送信系、特に入力音声をディジタルデータとして受信系に送信するため音声符号器に応用することができる。

請求の範囲

1．複数の雑音列をコードベクトル（C_0，C_1，C_2…）として格納する雑音符号帳から順次読み出した各該コードベクトルに対し声道特性を模擬した線形予測合成フィルタ処理を施して得られた再生信号と、入力音声信号との間の誤差が最小となるような1つのコードベクトルを評価演算によって求め、当該コードベクトルを特定する符号を用いてその入力音声信号を符号化する音声符号化方式において、

前記雑音符号帳を、初期ベクトル（C_0）と、隣接する前記コードベクトル間の差分ベクトルを全てのコードベクトルについて求めることによって得た複数のデルタベクトル（ΔC）と、を格納するデルタベクトル符号帳（11）によって構成し、

前記デルタベクトルを巡回的に加算することによって、仮想的に前記コードベクトル（C_0，C_1，C_2…）を再生することを特徴とする音声符号化方式。

2．各前記デルタベクトルが、N（Nは2以上の自然数）個の時系列的なサンプルデータからなるN次元ベクトルであって、N個の該サンプルデータのうち数個のサンプルデータが有意義データ（Δ1，Δ2，Δ3，Δ4）であって、残りは全てデータ零からなるスパース化ベクトルである請求項1に記載の音声符号化方式。

3．前記雑音符号帳内の前記コードベクトル（C_0，C_1，C_2…）を、隣接するコードベクトル間の差分ベクトルが小

さくなるように並べ替え、並べ替えた該コードベクトルについて、各隣接するコードベクトル間の差分ベクトルを求め、前記スパース化ベクトルを得る請求項２に記載の音声符号化方式。

４．前記の巡回的な加算を行うための巡回加算手段（２０）を、前記評価演算のための演算手段の一部として設ける請求項１に記載の音声符号化方式。

５．前記巡回加算手段（２０）が、演算データを加算する加算部（１４，１４－１，１４－２）と、該加算部の出力に遅延を与えて該加算部の１つの入力に戻す遅延部（１６，１６－１，１６－２）とからなり、該遅延部に前回の演算結果を保持しておいて、次に与えられる前記デルタベクトルを入力として、該演算結果を累積的に更新する請求項４に記載の音声符号化方式。

６．複数の前記デルタベクトル（ΔＣ）を、木構造状に配列された（Ｌ－１）種類のデルタベクトルによって表し、ここにＬは、前記初期ベクトル（C_0）を頂点とする該木構造を構成する階層の総数である請求項１に記載の音声符号化方式。

７．前記（Ｌ－１）種類のデルタベクトルを、前記階層毎に、前記初期ベクトル（C_0）から順次、加え合わせまたは差し引いて仮想的に（2^L-1）種類の前記コードベクトルを再生する請求項６に記載の音声符号化方式。

８．前記（2^L-1）種類のコードベクトルに零ベクトルを付加して、前記雑音符号帳に格納された2^L種類の前記コードベクトルと同数のコードベクトルを再現する請求項７に

記載の音声符号化方式。

9．前記（2^L －1）種類のコードベクトルに、前記初期ベクトル（C_0）に－1を乗算して得られるコードベクトル（$-C_0$）を付加して、前記雑音符号帳に格納された2^L種類の前記コードベクトルと同数のコードベクトルを再現する請求項7に記載の音声符号化方式。

10．前記の巡回的な加算を行うための巡回加算手段（20）を、前記評価演算のための演算手段の一部として設ける請求項6に記載の音声符号化方式。

11．前記評価演算に相互相関演算と、線形予測合成フィルタ演算を含み、その合成フィルタ演算出力（AC）を、1階層前の合成フィルタ演算出力と今回のデルタベクトルとを用いて漸化式で表現することにより、前記相互相関演算を漸化式で表現して演算する請求項10に記載の音声符号化方式。

12．前記評価演算に自己相関演算を含み、前記合成フィルタ演算出力（AC）を、1階層前の合成フィルタ演算出力と今回のデルタベクトルとを用いて漸化式で表現することにより、前記自己相関演算を、前記初期ベクトル（C_0）の合成フィルタ演算出力と、前記（L－1）種類のデルタベクトルのフィルタ演算出力の合計L通りの自己相関と、各前記合成フィルタ演算出力間の（L^2 －1）／2通りの相互相関を用いて表現して演算する請求項11に記載の音声符号化方式。

13．前記初期ベクトル（C_0）および前記（L－1）種類の各デルタベクトル（ΔC）の、前記木構造状の配列における順番を、前記入力音声信号の特性に合わせて変更し、該初

期ベクトルおよびデルタベクトルの並べ替えを行う請求項６に記載の音声符号化方式。

１４．前記評価演算の１つである線形予測合成フィルタ演算を行うためのフィルタ（３）の特性に応じて、フレーム毎に、前記初期ベクトルおよびデルタベクトルの並べ替えを行う請求項１３に記載の音声符号化方式。

１５．前記フィルタ（３）から得られる前記再生信号のパワーを前記評価演算において評価し、そのパワーが当該フィルタ処理の前におけるパワーに比べて最も増大しているベクトルから順次、新たな初期ベクトル（C_0）→第１デルタベクトル（ΔC_1）→第２デルタベクトル（ΔC_2）…という順番で、前記のベクトルの並べ替えを行う請求項１４に記載の音声符号化方式。

１６．前記初期ベクトル（C_0）およびデルタベクトル（ΔC）が、前記フィルタ処理の後に相互に直交するように予め変換し、前記デルタベクトル符号帳（１１）内の全てのベクトルを超平面上に均一に分布せしめる請求項１５に記載の音声符号化方式。

１７．前記パワーを正規化して得た正規化パワーによって、前記のパワーの大小比較を行う請求項１５に記載の音声符号化方式。

１８．前記コードベクトルを特定する前記符号を割り付けるに際し、前記木構造状のベクトル配列において上位の前記階層に属する符号間距離の方が、下位の前記階層に属する符号間距離の方よりも大きくなるように、該符号を割り付ける

請求項１３に記載の音声符号化方式。

Fig.1

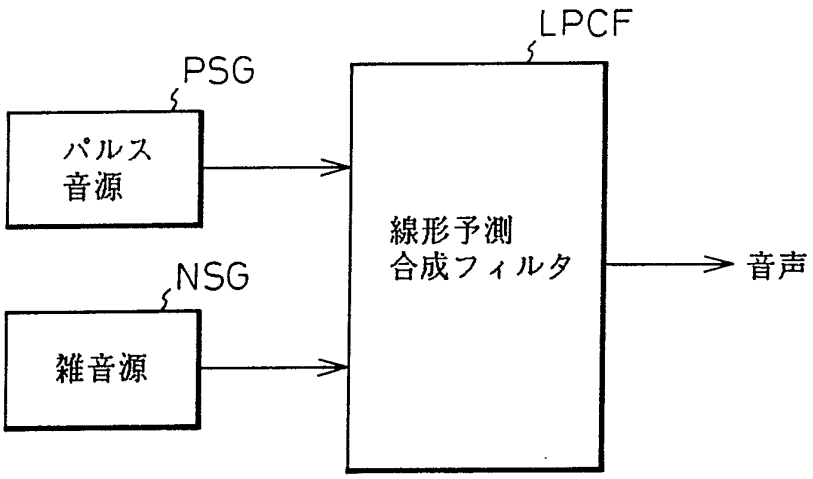


Fig.2

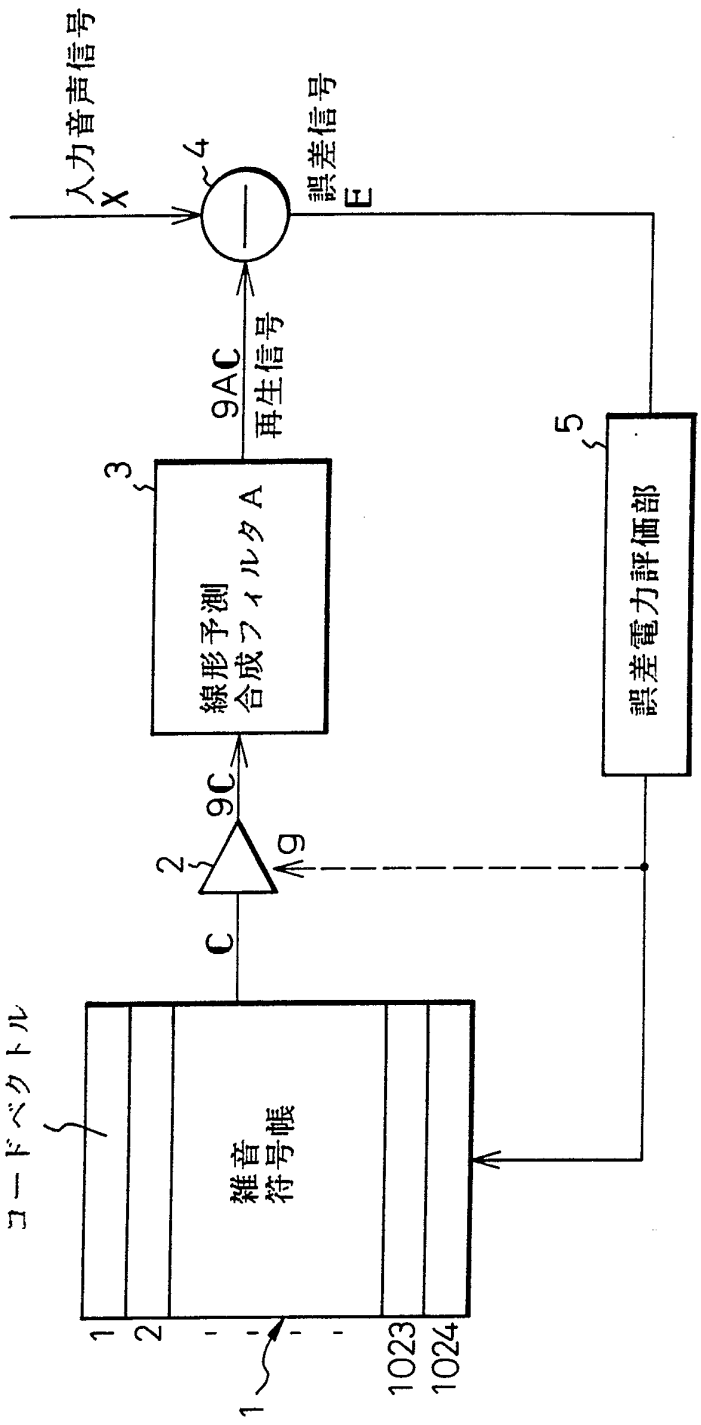

入力音声信号
X
4
誤差信号
E
9AC
再生信号
3
線形予測
合成フィルタA
5
誤差電力評価部
9C
2
9
C
コードベクトル
雑音
符号帳
1
2
1023
1024
1

Fig.3

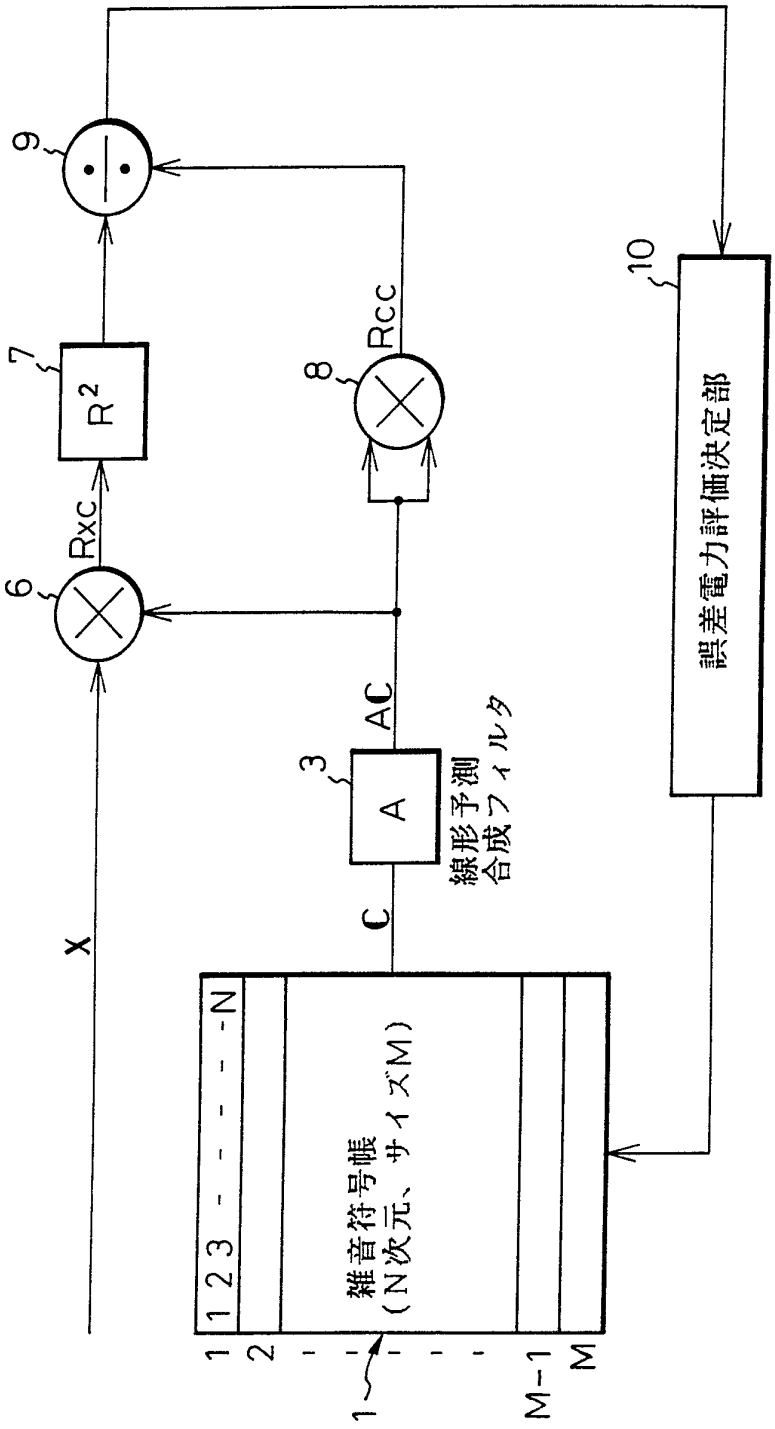

9
10
誤差電力評価決定部
7
R²
8
Rcc
Rxc
6
AC
3
A
線形予測
合成フィルタ
C
x
1 2 3 - - - - -N
雑音符号帳
(N次元、サイズM)
1
2
M-1
M

Fig.4

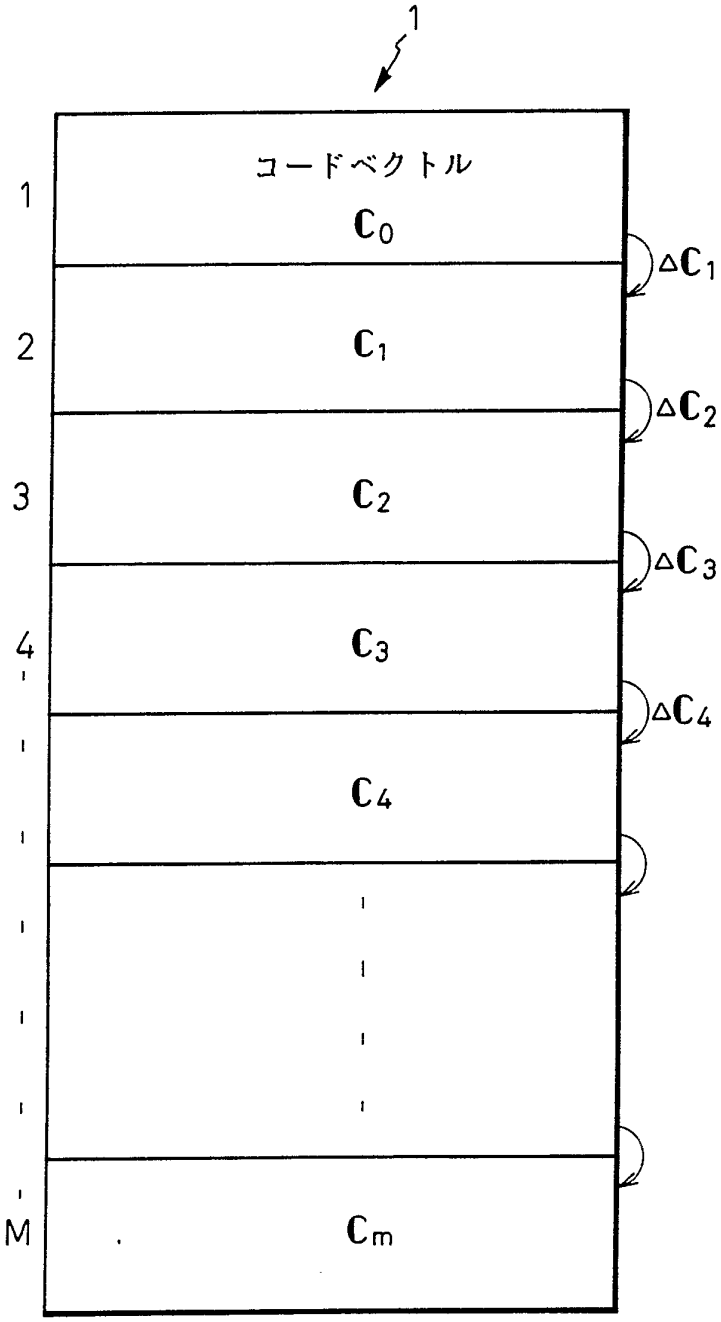

1
コードベクトル
1
C_0
ΔC_1
2
C_1
ΔC_2
3
C_2
ΔC_3
4
C_3
ΔC_4
C_4
M
C_m

Fig.5

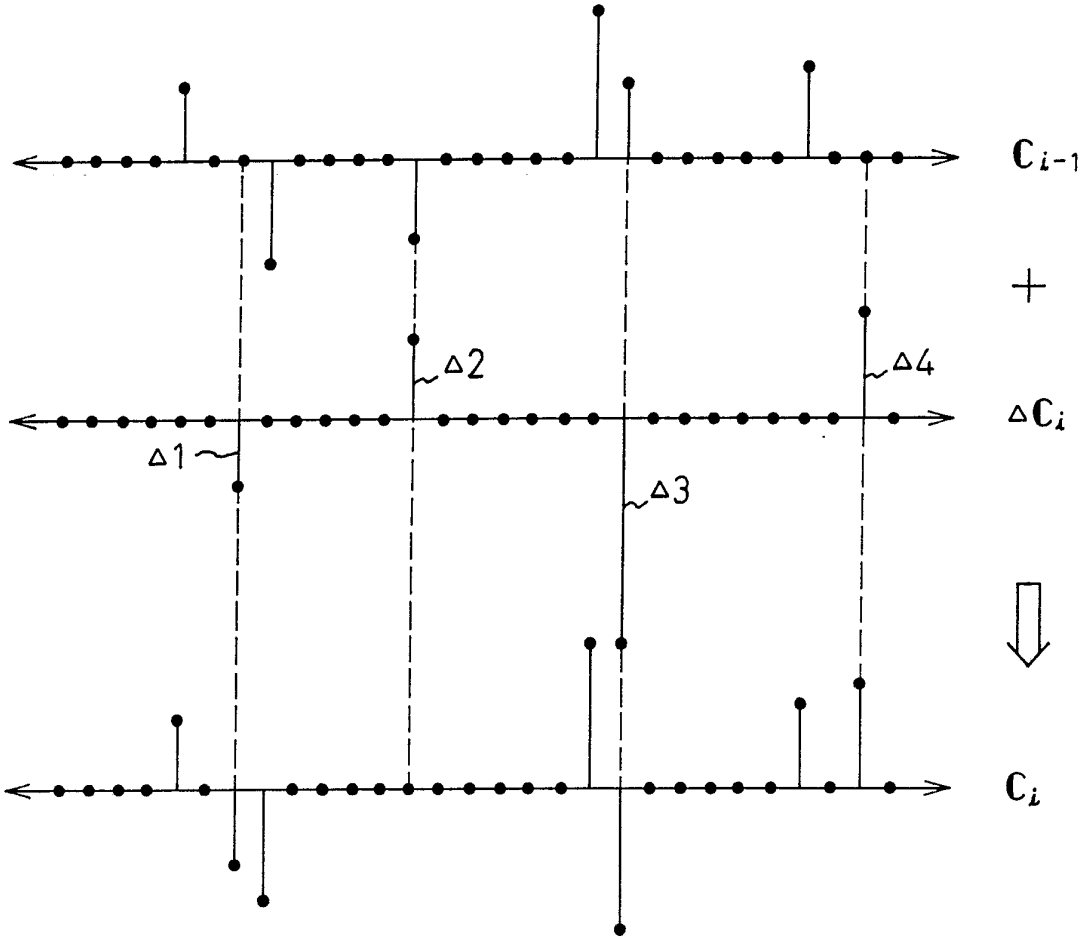


Fig.6

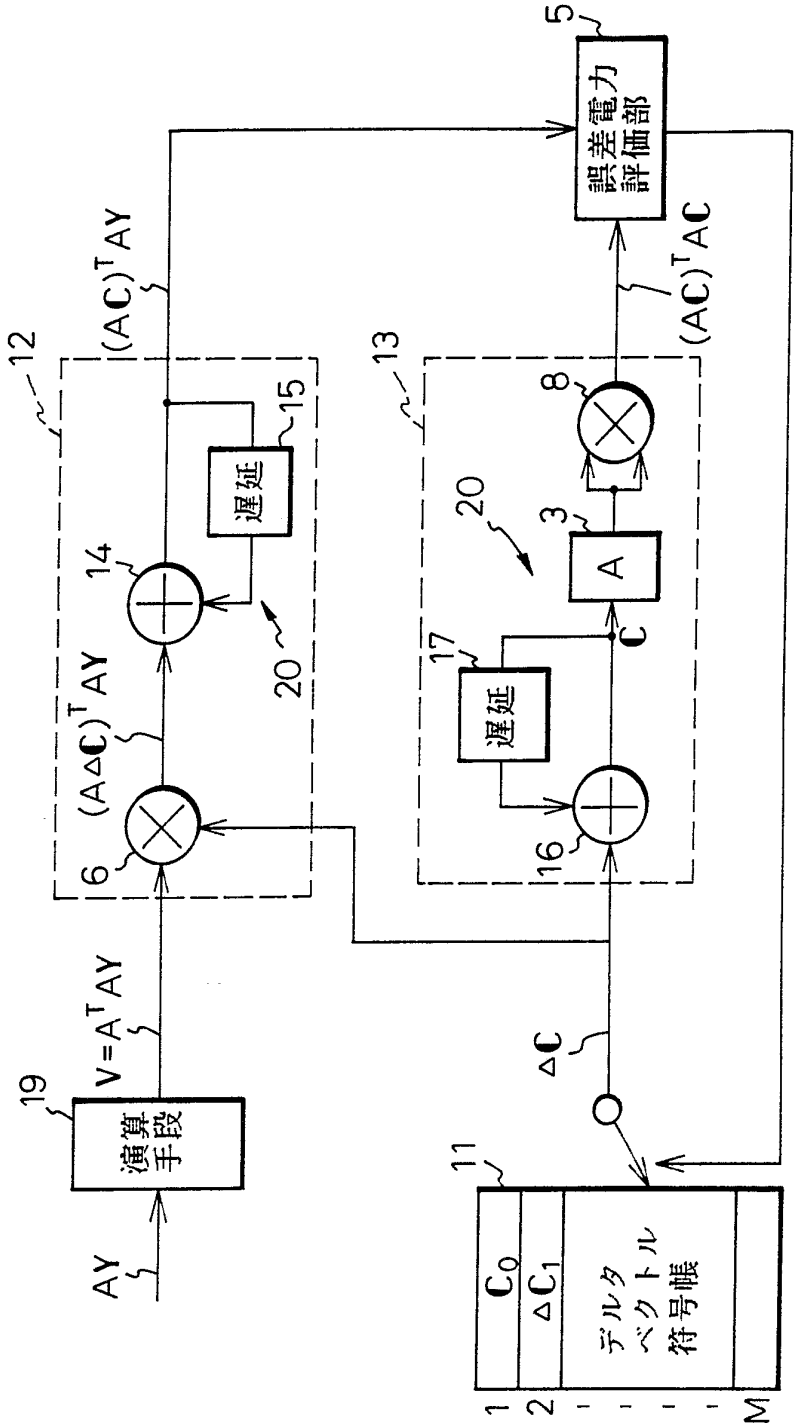

5
誤差電力評価部
$(A\mathbf{C})^T A\mathbf{C}$
$(A\mathbf{C})^T A\mathbf{Y}$
12
13
8
15
遅延
20
3
A
$\mathbf{C}$
14
$(A\Delta\mathbf{C})^T A\mathbf{Y}$
17
遅延
6
16
$V=A^T A\mathbf{Y}$
$\Delta\mathbf{C}$
19
演算手段
11
$\mathbf{C}_0$
$\Delta\mathbf{C}_1$
デルタベクトル符号帳
$A\mathbf{Y}$
1
2
M

Fig.7

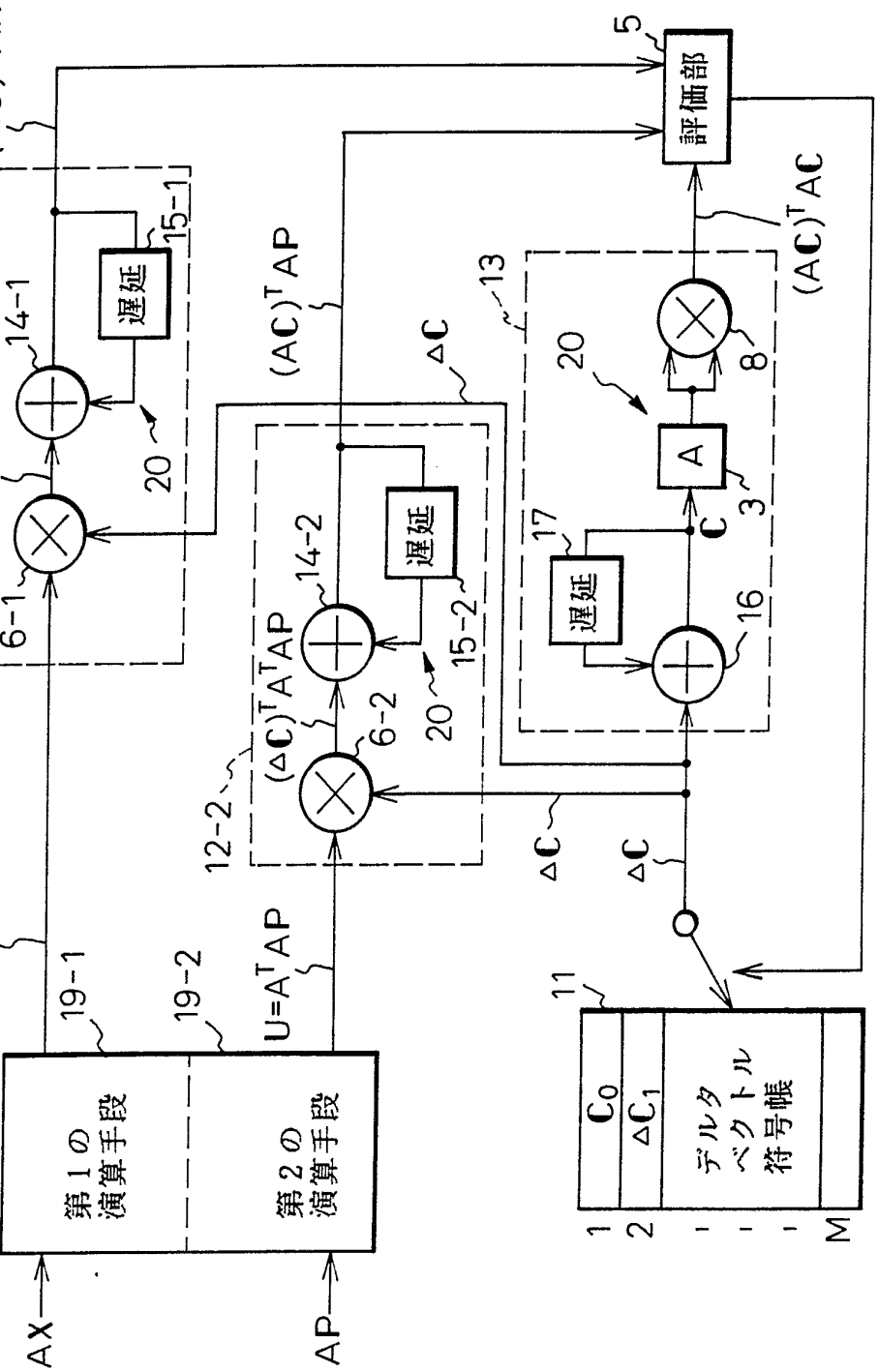

AX
AP
第1の演算手段
第2の演算手段
19-1
19-2
V=A^TAX
U=A^TAP
12-1
12-2
6-1
6-2
14-1
14-2
15-1
15-2
遅延
20
(ΔC)^TA^TAX
(ΔC)^TA^TAP
(AC)^TAX
(AC)^TAP
ΔC
13
17
16
A
C
3
8
(AC)^TAC
5
評価部
11
C0
ΔC1
デルタベクトル符号帳
1
2
M

Fig.8

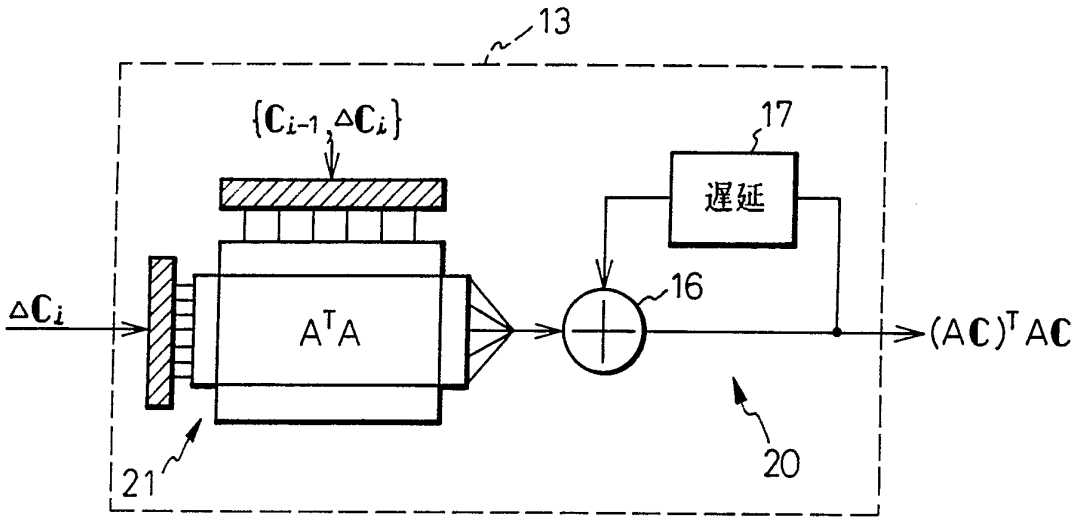


Fig.9

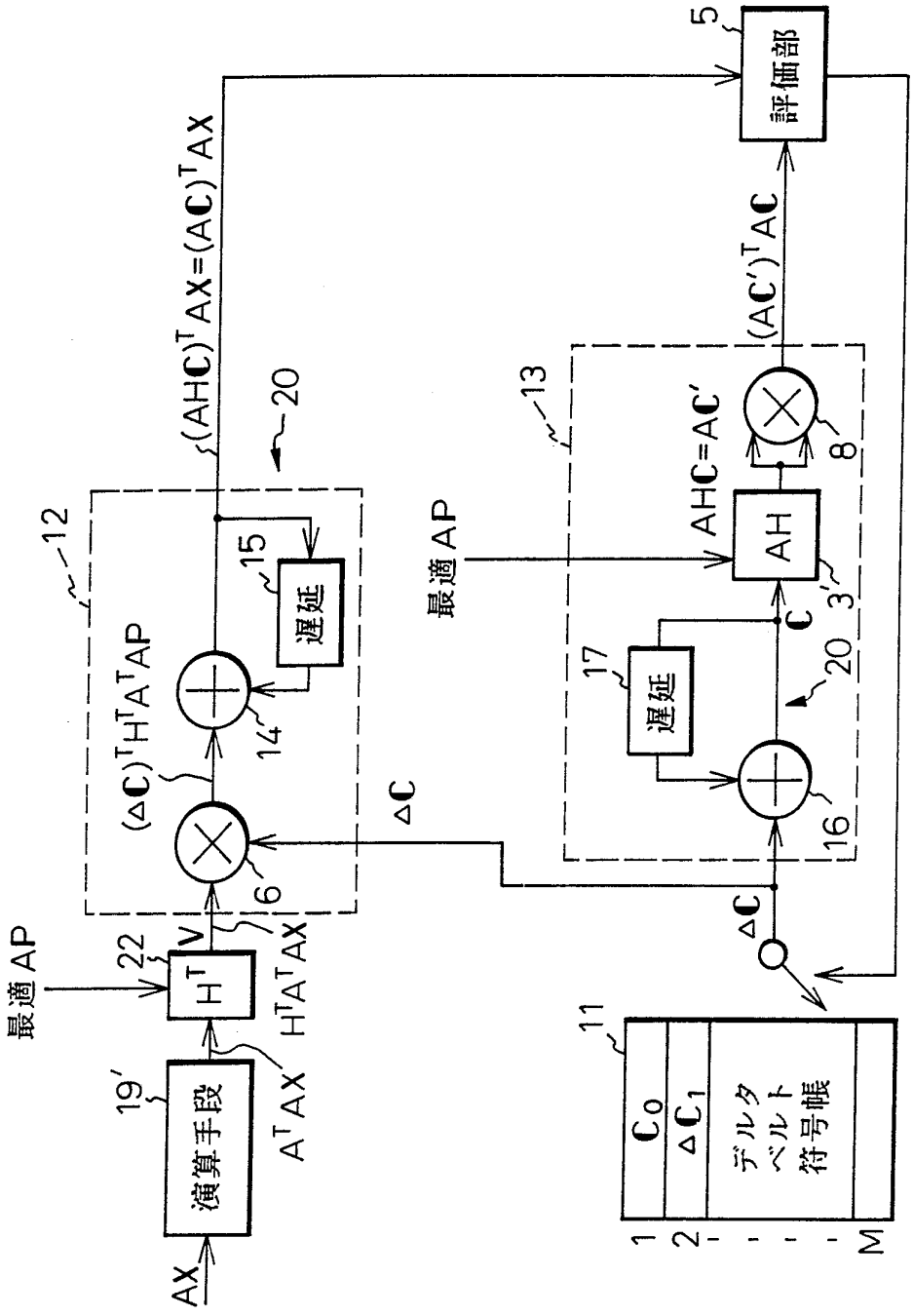

AX
演算手段
19'
A^TAX
最適 AP
22
H^T
H^TA^TAX
12
$(\Delta C)^TH^TA^TAP$
6
14
15
遅延
$(AHC)^TAX=(AC)^TAX$
20
ΔC
5
評価部
13
最適 AP
AHC=AC'
AH
3'
8
$(AC')^TAC$
C
17
遅延
20
16
ΔC
11
C_0
ΔC_1
デルタベルト符号帳
1
2
M

Fig.10

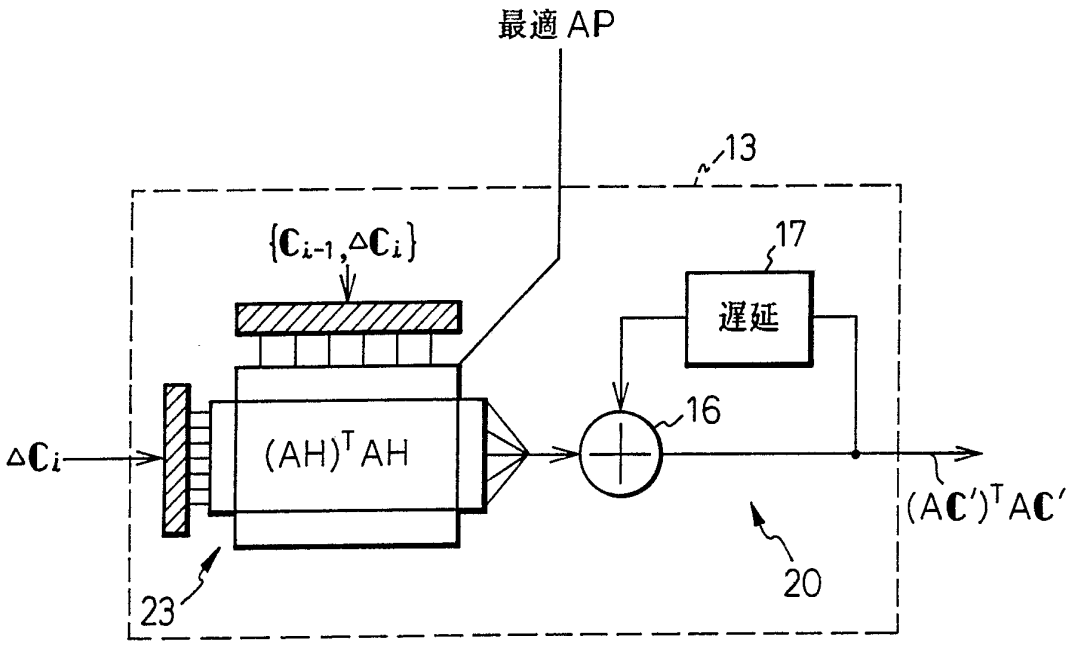


Fig.11

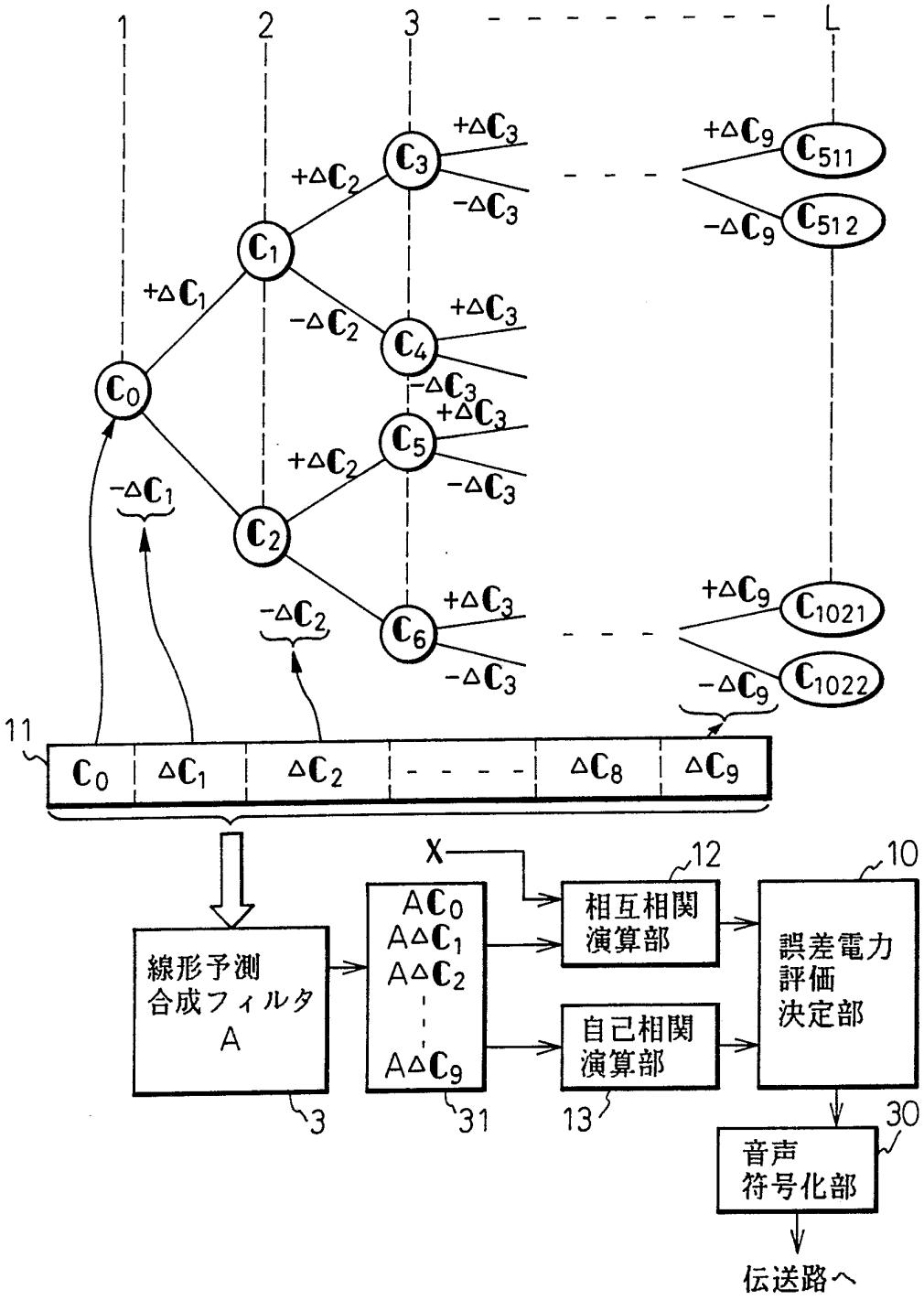

1
2
3
L
C0
C1
C2
C3
C4
C5
C6
C511
C512
C1021
C1022
+ΔC1
−ΔC1
+ΔC2
−ΔC2
+ΔC3
−ΔC3
+ΔC9
−ΔC9
11
C0
ΔC1
ΔC2
ΔC8
ΔC9
X
AC0
AΔC1
AΔC2
AΔC9
線形予測
合成フィルタ
A
3
31
相互相関
演算部
12
自己相関
演算部
13
誤差電力
評価
決定部
10
音声
符号化部
30
伝送路へ

Fig.12

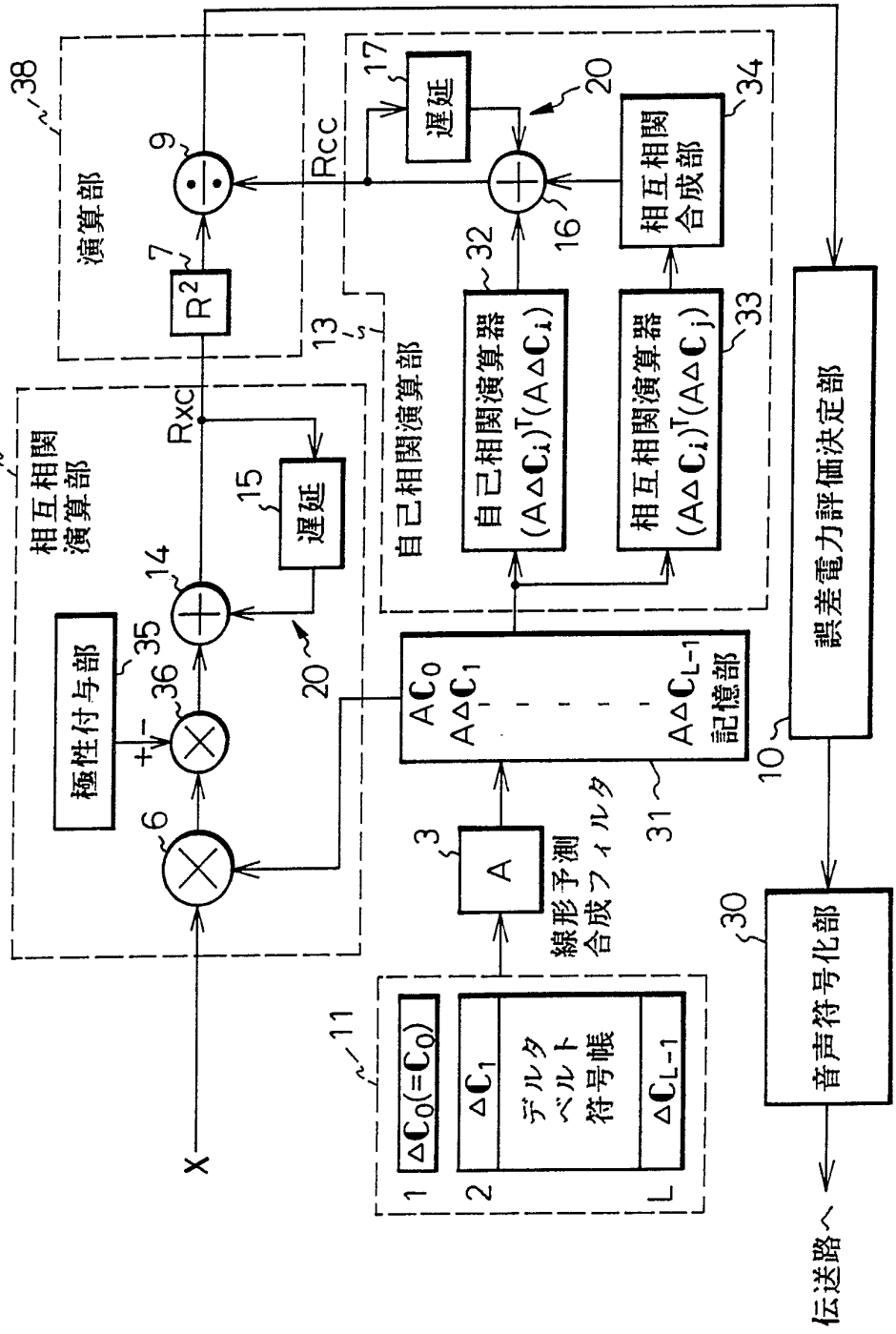

伝送路へ
音声符号化部
誤差電力評価決定部
相互相関合成部
遅延
演算部
自己相関演算部
相互相関演算部
極性付与部
遅延
デルタベクトル符号帳
線形予測合成フィルタ
記憶部
X
Rxc
Rcc
R²
A
ΔC₀(=C₀)
ΔC₁
ΔC_{L-1}
AC₀
AΔC₁
AΔC_{L-1}
自己相関演算器 (AΔCᵢ)ᵀ(AΔCᵢ)
相互相関演算器 (AΔCᵢ)ᵀ(AΔCⱼ)
1
2
L
3
6
7
9
10
11
12
13
14
15
16
17
20
30
31
32
33
34
35
36
38

Fig.13

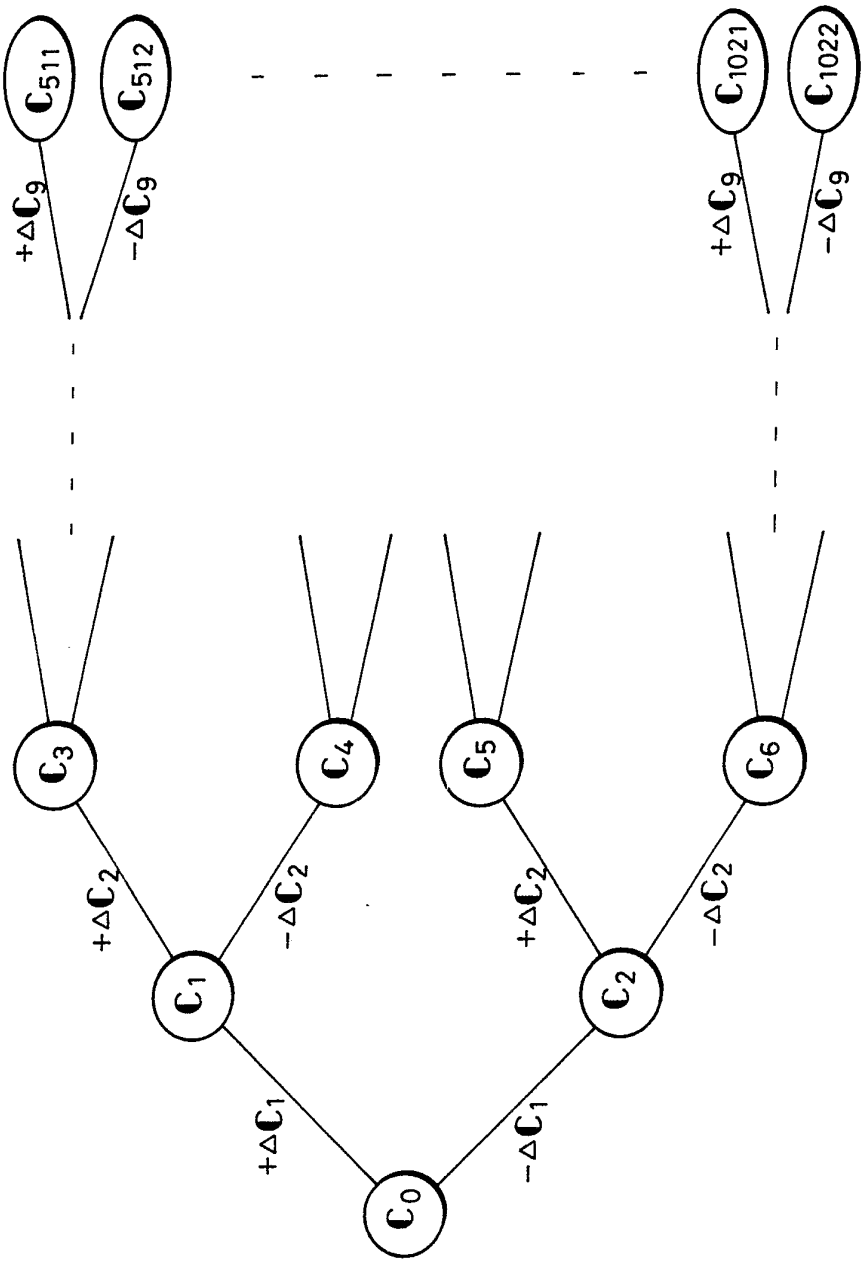
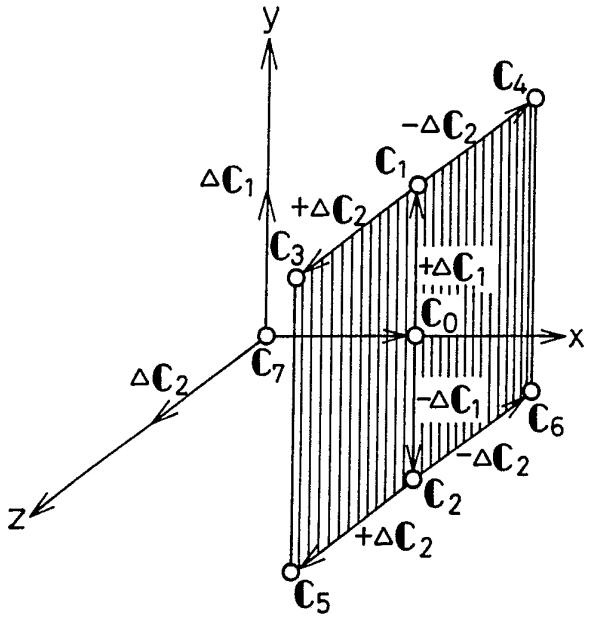
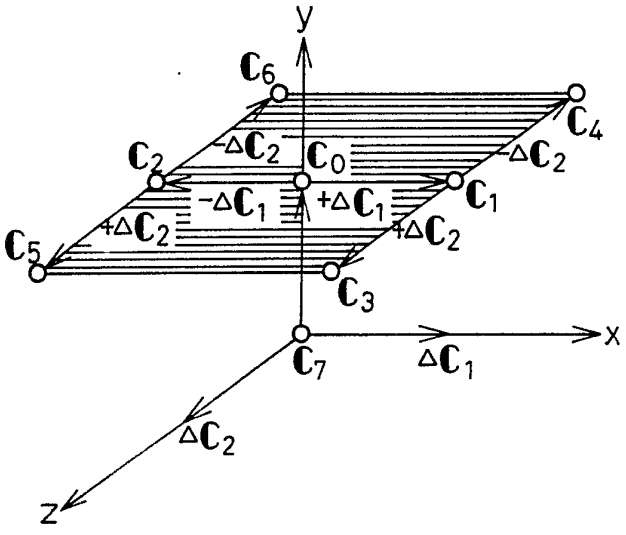

C511
C512
C1021
C1022
+ΔC9
-ΔC9
+ΔC9
-ΔC9
C3
C4
C5
C6
+ΔC2
-ΔC2
+ΔC2
-ΔC2
C1
C2
+ΔC1
-ΔC1
C0

Fig.14A



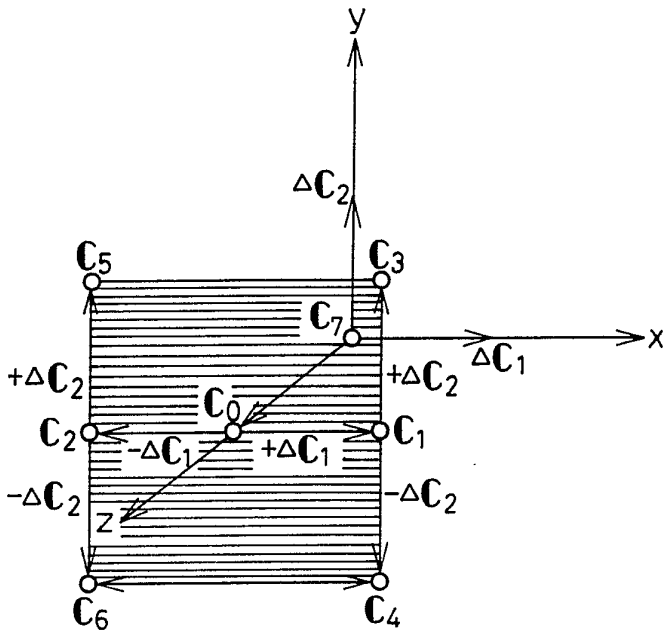
モードA：$C_0=e_x, \Delta C_1=e_y, \Delta C_2=e_z$

Fig.14B



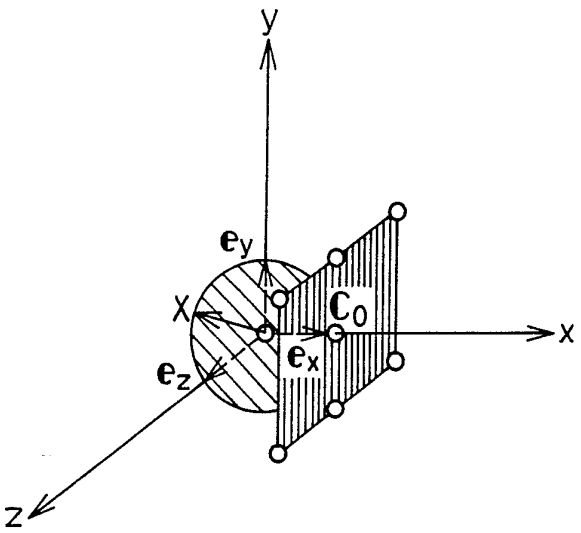
モードB：$C_0=e_y, \Delta C_1=e_x, \Delta C_2=e_z$

Fig.14C



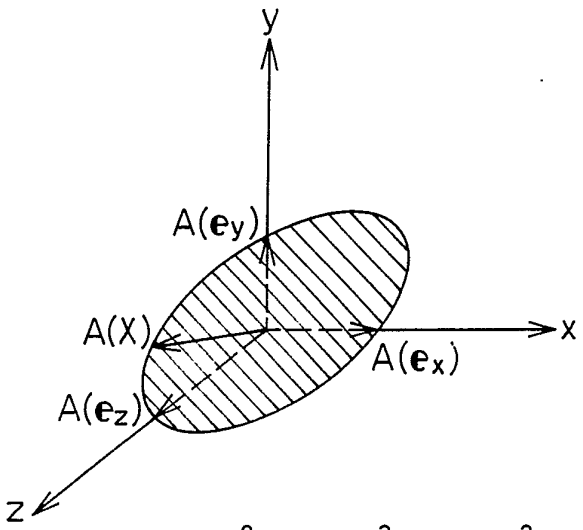
モードC: $C_0=e_z, \Delta C_1=e_x, \Delta C_2=e_y$

Fig.15A



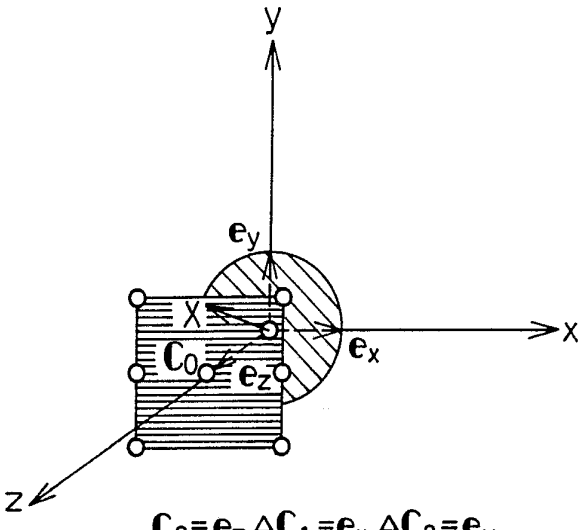
$C_0=e_x, \Delta C_1=e_y, \Delta C_2=e_z$

Fig.15B



$|A(e_z)|^2 > |A(e_x)|^2 > |A(e_y)|^2$

Fig.15C



$C_0=e_z, \Delta C_1=e_x, \Delta C_2=e_y$

Fig.16

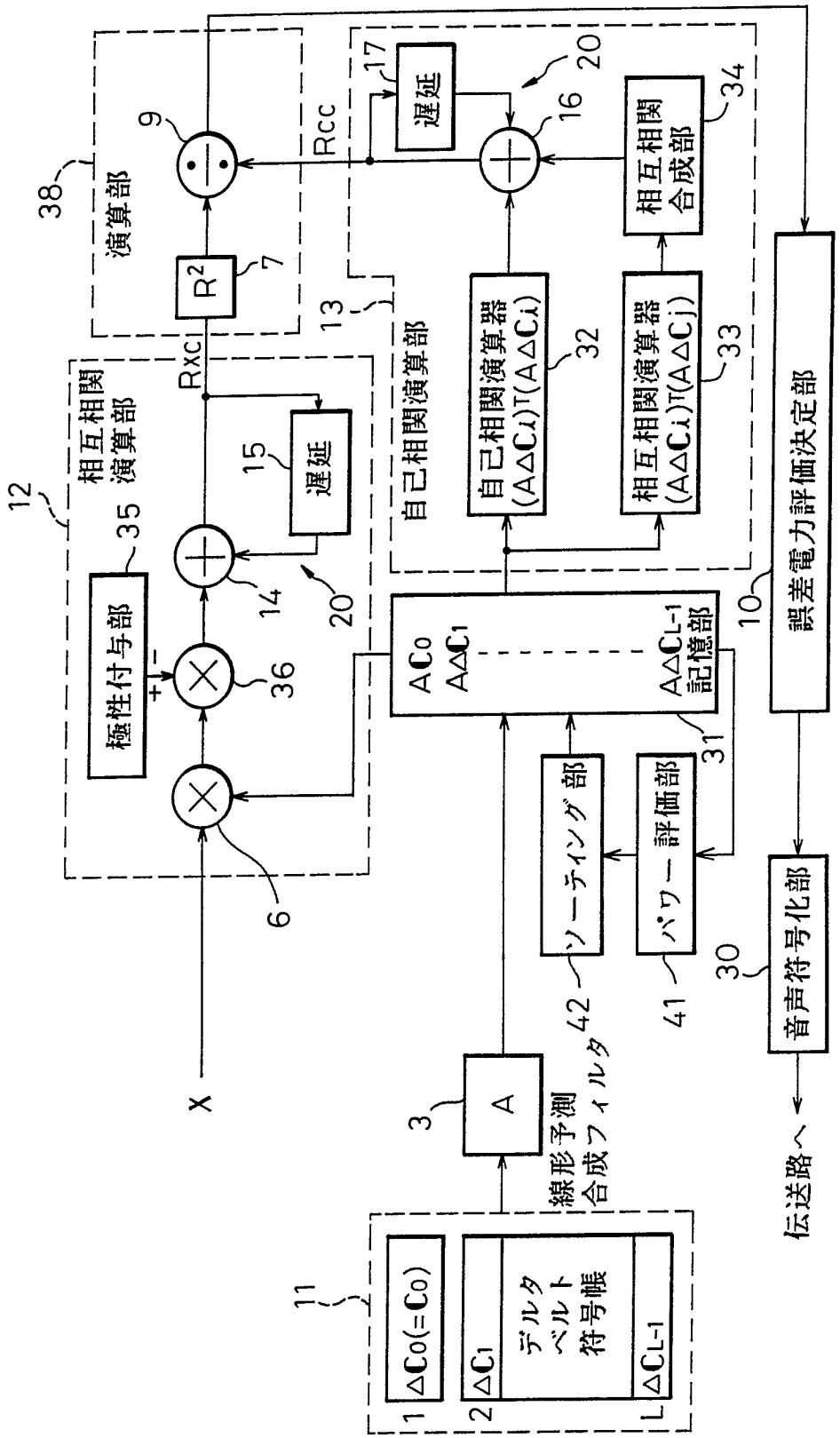

ΔC0(=C0)
ΔC1
デルタベクトル符号帳
ΔCL-1
1
2
L
11
線形予測合成フィルタ
A
3
42
ピーキング部
パワー評価部
41
AC0
AΔC1
AΔCL-1記憶部
31
30
音声符号化部
伝送路へ
誤差電力評価決定部
10
自己相関演算器 (AΔCi)T(AΔCi)
32
相互相関演算器 (AΔCi)T(AΔCj)
33
相互相関合成部
34
自己相関演算部
13
20
遅延
17
16
Rcc
演算部
38
9
R2
7
Rxc
相互相関演算部
12
極性付与部
35
36
遅延
15
14
20
6
X

Fig.17

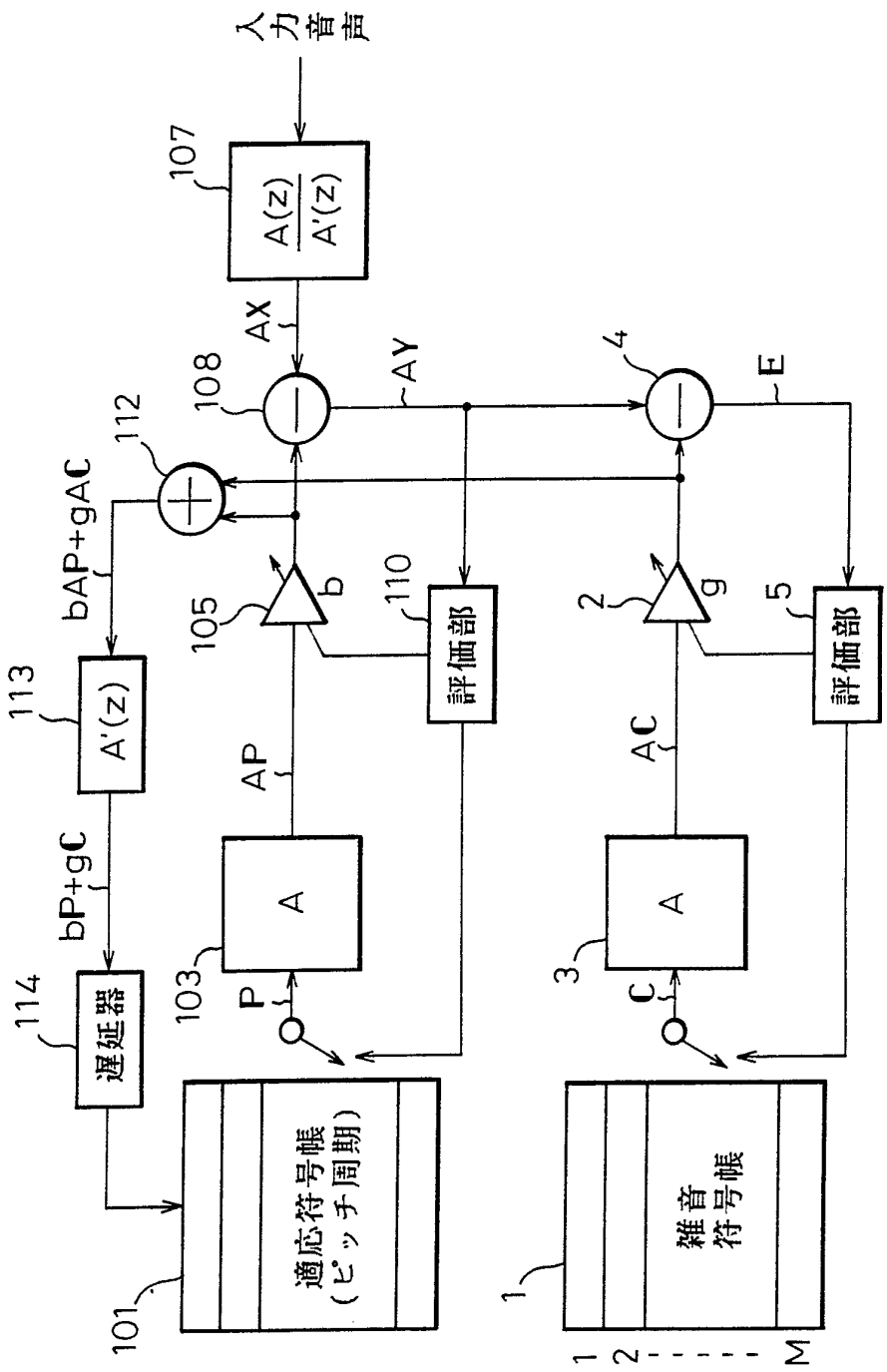

入力音声
107
A(z)/A'(z)
AX
108
AY
4
E
112
bAP+gAC
105
b
110
評価部
2
g
5
評価部
113
A'(z)
AP
AC
bP+gC
114
遅延器
103
A
3
A
P
C
101
適応符号帳
(ピッチ周期)
1
雑音
符号帳
1 2 - - - - M

Fig.18

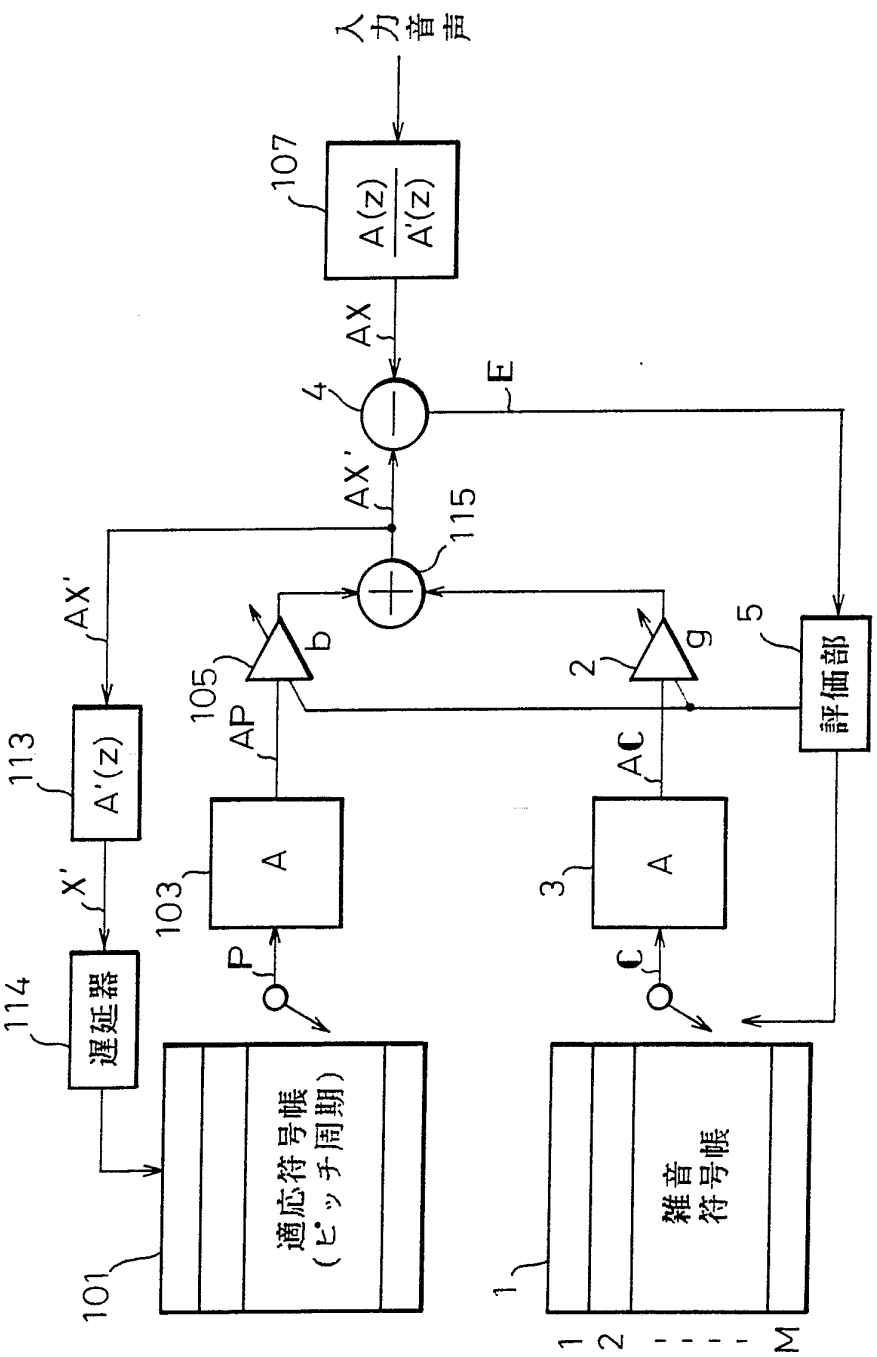

入力音声
107
A(z)/A'(z)
AX
4
E
AX'
115
AX'
b
105
AP
2
g
5
評価部
113
A'(z)
X'
114
遅延器
103
A
P
3
A
AC
C
101
適応符号帳
(ピッチ周期)
1
雑音
符号帳
1
2
M

Fig.19

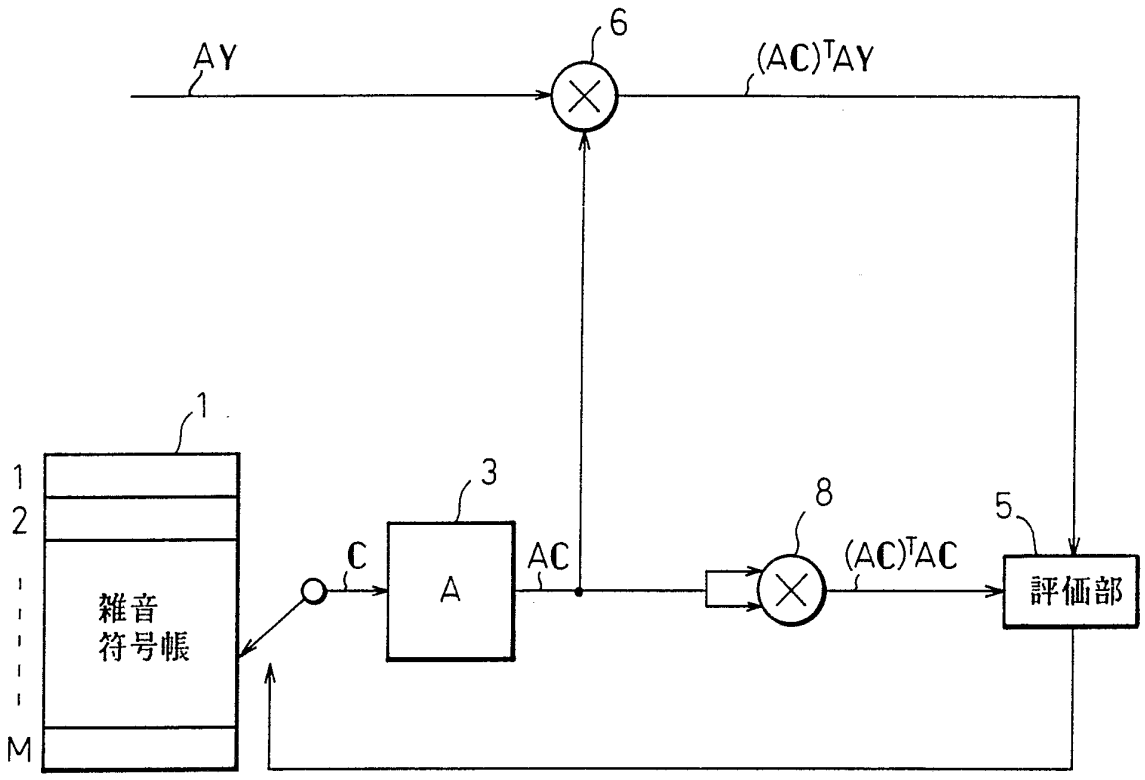


Fig.20

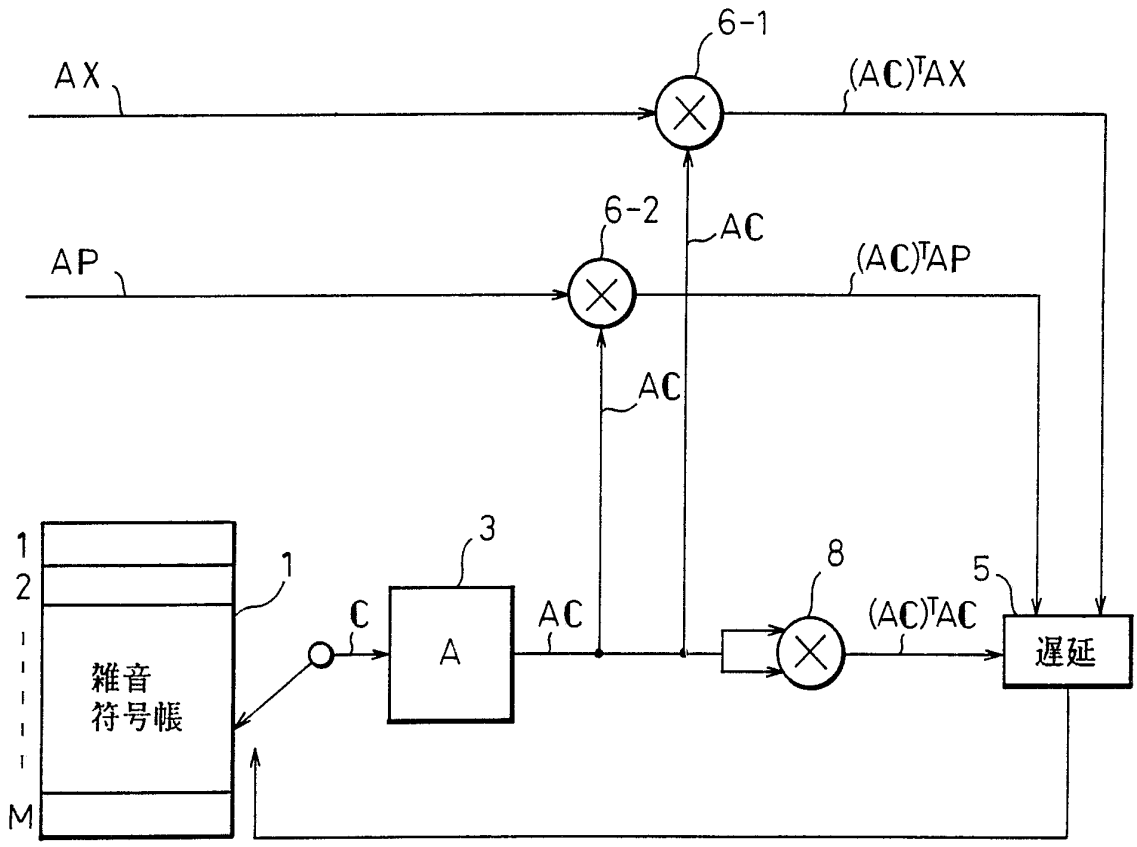


Fig. 21A

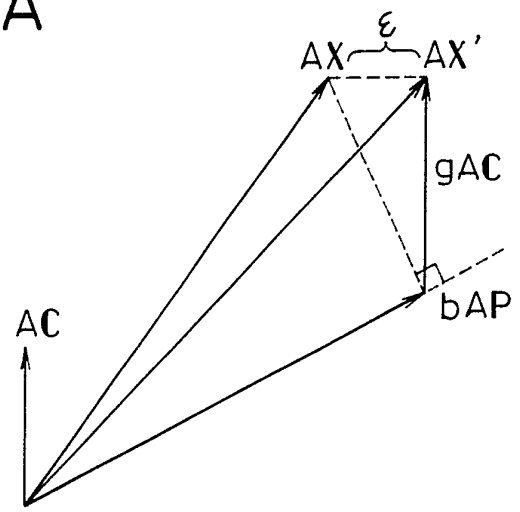


Fig. 21B

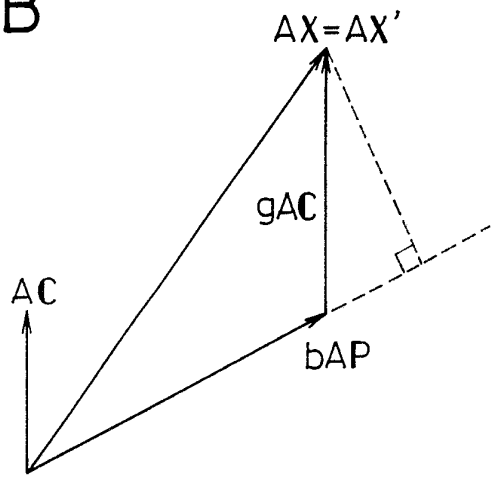


Fig. 21C

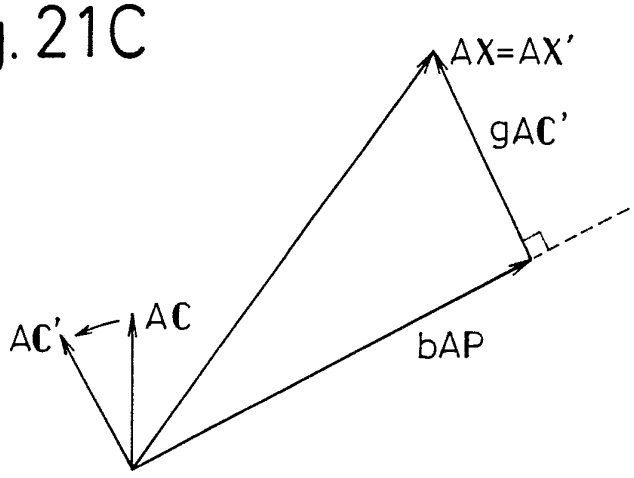


Fig. 22

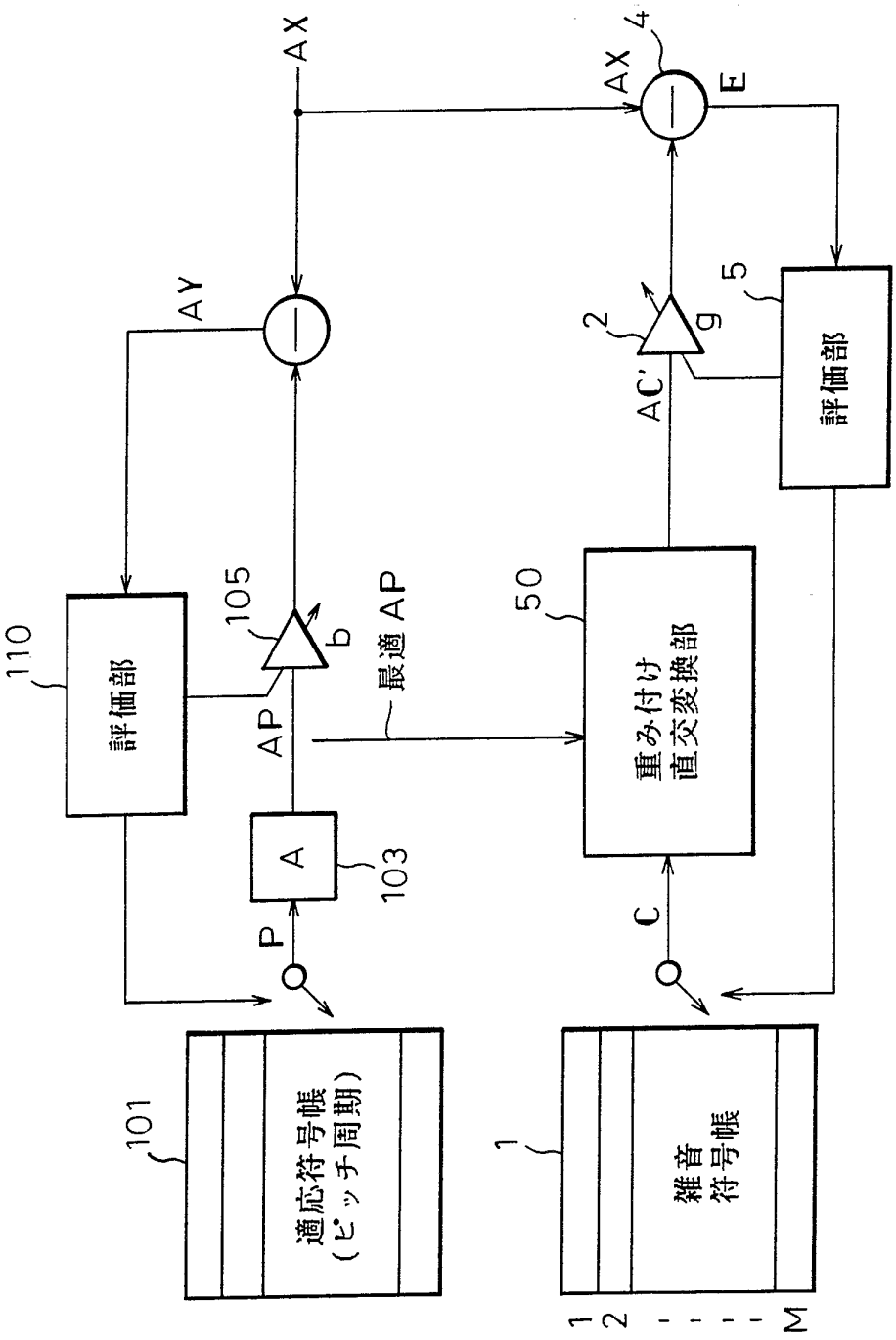

AX
AX
4
E
AY
2
g
5
AC'
評価部
105
b
110
評価部
最適 AP
50
重み付け
直交変換部
AP
A
103
P
C
101
適応符号帳
(ピッチ周期)
1
雑音
符号帳
1
2
M

Fig. 23

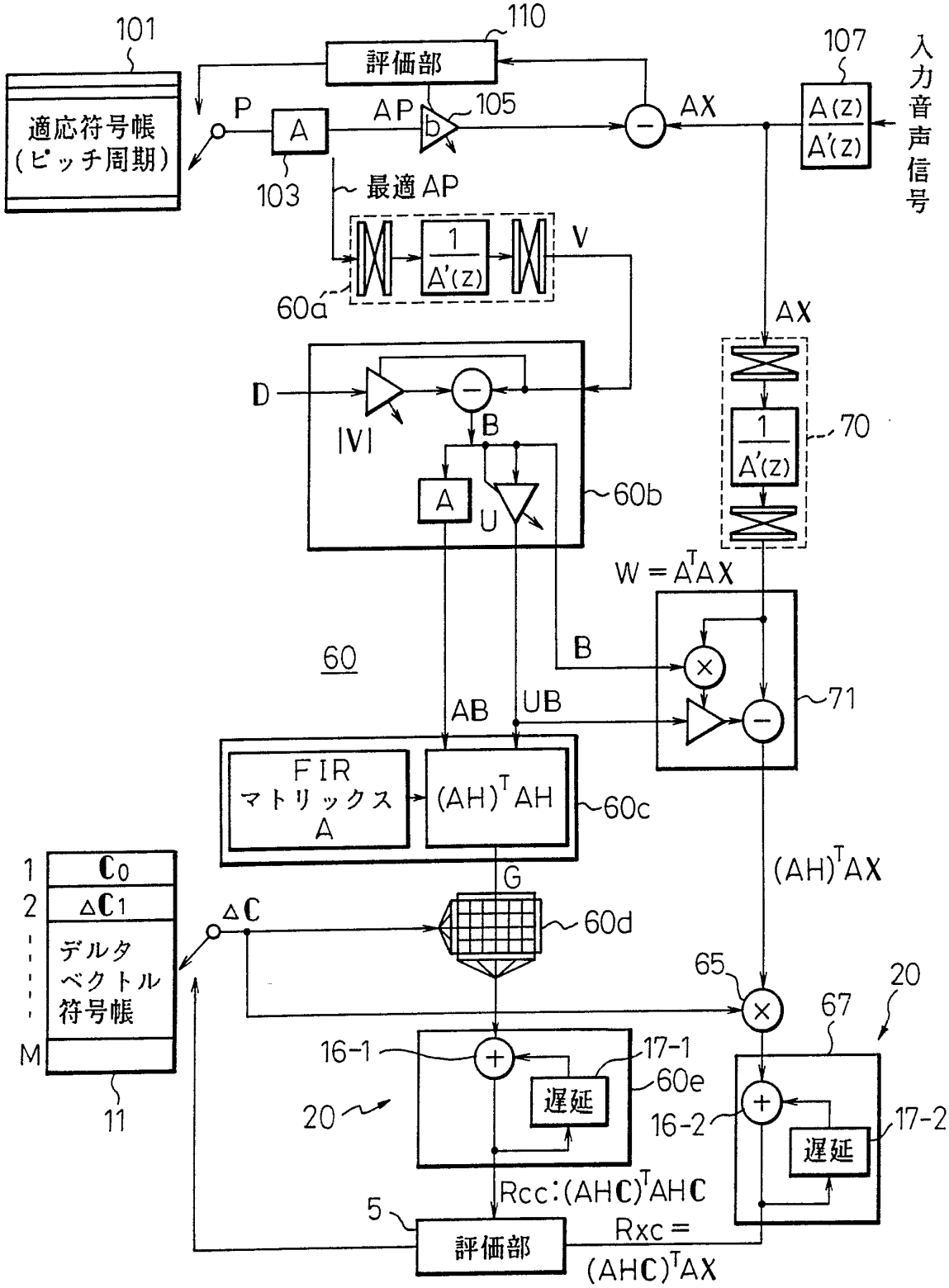

101
適応符号帳
(ピッチ周期)
110
評価部
P
A
AP
b
105
AX
107
A(z)/A'(z)
入力音声信号
103
最適AP
1/A'(z)
V
60a
AX
D
|V|
B
A
U
60b
1/A'(z)
70
W = A^T AX
60
B
AB
UB
71
FIR
マトリックス
A
(AH)^T AH
60c
(AH)^T AX
G
60d
1
2
M
C0
ΔC1
デルタ
ベクトル
符号帳
ΔC
11
65
20
67
16-1
17-1
60e
遅延
20
16-2
17-2
遅延
Rcc:(AHC)^T AHC
5
評価部
Rxc =
(AHC)^T AX

Fig. 24A

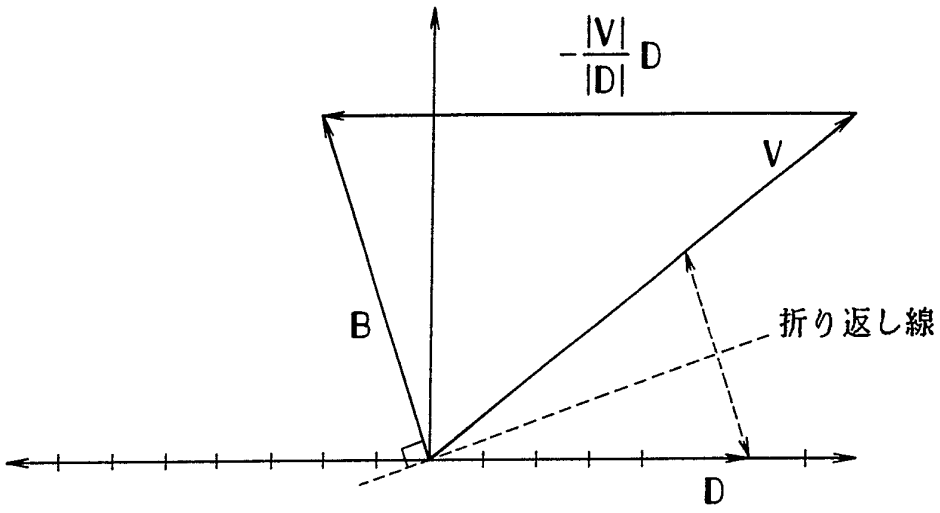


Fig. 24B

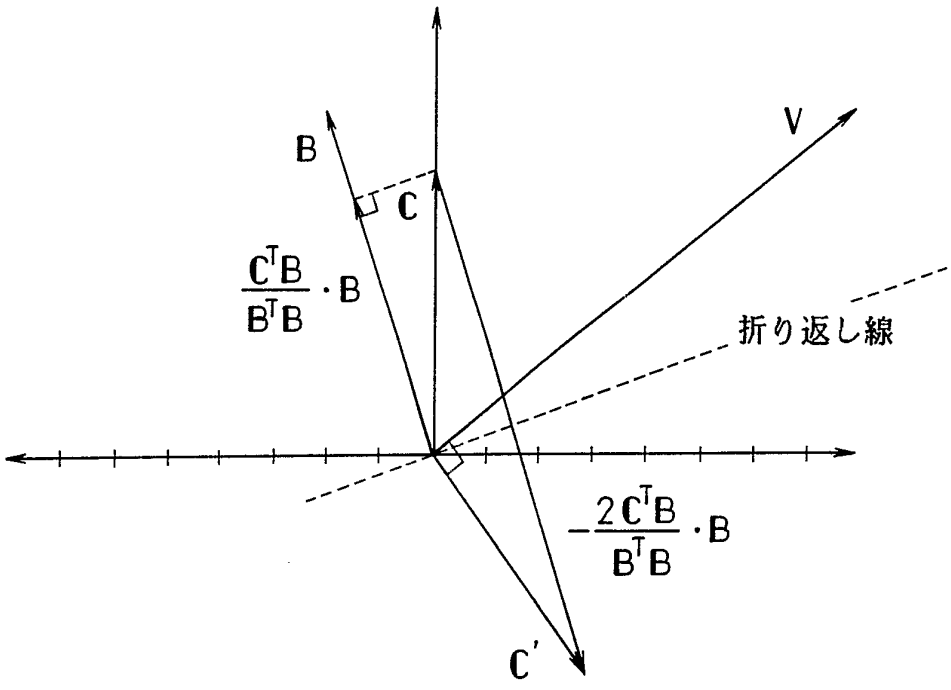


Fig.25

	符号帳		フィルタ	相互相関	自己相関	合計 (Np=10)	合計 N=10
逐次最適化方式 (a)	1	4／5スパース	$N^2/10$	N/5	N	$N^2/10$ +1.2N	432K
	2	オーバーラップ (1新サンプル／ベクトル +4／5スパース)	N/5	N/5	N	1.4N	84K
	3	2サンプル・デルタ ベクトル (4／5スパース)	0	2	2N/5 +2	0.4N +4	28K
同時最適化方式 (b)	1	4／5スパース	$N^2/10$	2N/5	N	$N^2/10$ +1.4N	444K
	2	オーバーラップ (1新サンプル／ベクトル +4／5スパース)	N/5	2N/5	N	1.6N	96K
	3	2サンプル・デルタ ベクトル (4／5スパース)	0	4	2N/5 +2	0.4N +6	30K
ピッチ直交変換最適化方式 (c)	1	4／5スパース	$N^2/5$ +1.2N	N/5	N	$N^2/10$ +2.4N	504K
	2	オーバーラップ (1新サンプル／ベクトル +4／5スパース)	1.4N	N/5	N	2.6N	156K
	3	2サンプル・デルタ ベクトル (4／5スパース)	0	2	2N/5 +2	0.4N +4	28K

Fig.26

			従来方式		本発明の第2実施例による方式	
(a)	演算量	フィルタ	Np×N×M	400K積和算 (N=40,M=1024)	Np×N×L	4K積和算 (N=40,L=10)
		相互相関	N×M	40K積和算 (N=40,M=1024)	N×L	0.4K積和算 (N=40,L=10)
		自己相関	N×M	40K積和算 (N=40,M=1024)	$\frac{N\times L(L+1)}{2}$	2.2K積和算 (N=40,L=10)
		合計		480K積和算 (96 Mops)		6.6K積和算 (1.3Mops)
(b)	メモリー量		N×M	40Kワード	N×L	0.4Kワード

参照符号の説明

1…雑音符号帳

3…線形予測合成フィルタ

5…誤差電力評価部

11…デルタベクトル符号帳

12…自己相関演算部

13…相互相関演算部

14，16…加算部

15，17…遅延部

20…巡回加算手段

30…音声符号化部

41…パワー評価部

42…ソーティング部

C_0…初期ベクトル

C_1，C_2，C_3…コードベクトル

ΔC…デルタベクトル

INTERNATIONAL SEARCH REPORT

International Application No PCT/JP91/01235

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) [6]

According to International Patent Classification (IPC) or to both National Classification and IPC

Int. Cl[5] G10L9/18

II. FIELDS SEARCHED

Minimum Documentation Searched [7]

Classification System	Classification Symbols
IPC	G10L9/18

Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched [8]

Jitsuyo Shinan Koho 1970 - 1991
Kokai Jitsuyo Shinan Koho 1970 - 1991

III. DOCUMENTS CONSIDERED TO BE RELEVANT [9]

Category *	Citation of Document, [11] with indication, where appropriate, of the relevant passages [12]	Relevant to Claim No. [13]
A	JP, A, 1-296300 (International Business Machines Corp.), November 29, 1989 (29. 11. 89) & (Family: none)	1
A	JP, A, 61-237519 (Mitsubishi Electric Corp.), October 22, 1986 (22. 10. 86), & EP, A1, 193185 & US, A, 4710812	1, 6, 7, 8, 17, 18
A	JP, A, 63-240600 (Matsushita Electric Ind. Co., Ltd.), October 6, 1988 (06. 10. 88), (Family: none)	13

* Special categories of cited documents: [10]

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

IV. CERTIFICATION

Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report
October 9, 1991 (09. 10. 91)	November 5, 1991 (05. 11. 91)
International Searching Authority	**Signature of Authorized Officer**
Japanese Patent Office	

国　際　調　査　報　告

国際出願番号PCT/JP91/01235

I. 発明の属する分野の分類

国際特許分類（IPC）

Int. Cl⁵
G10L9/18

II. 国際調査を行った分野

調査を行った最小限資料

分類体系	分類記号
IPC	G10L9/18

最小限資料以外の資料で調査を行ったもの

日本国実用新案公報　1970－1991年
日本国公開実用新案公報　1970－1991年

III. 関連する技術に関する文献

引用文献のカテゴリー※	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	請求の範囲の番号
A	JP, A, 1－296300（インターナショナル・ビジネス・マシーンズ・コーポレーション），29. 11月. 1989（29. 11. 89）&（データ未蓄積）	1
A	JP, A, 61－237519（三菱電機株式会社），22. 10月. 1986（22. 10. 86）&EP, A1, 193185&US, A, 4710812	1,6,7,8,17,18
A	JP, A, 63－240600（松下電器産業株式会社），6. 10月. 1988（06. 10. 88）（ファミリーなし）	13

※引用文献のカテゴリー
「A」特に関連のある文献ではなく、一般的技術水準を示すもの
「E」先行文献ではあるが、国際出願日以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献
「T」国際出願日又は優先日の後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」同一パテントファミリーの文献

IV. 認　証

国際調査を完了した日	国際調査報告の発送日
09. 10. 91	05.11.91

国際調査機関	権限のある職員
日本国特許庁（ISA/JP）	特許庁審査官　渡邊　聡　5H 8622