

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2011-107602

(P2011-107602A)

(43) 公開日 平成23年6月2日(2011.6.2)

(51) Int.Cl.
G10L 21/02 (2006.01)

F I
G10L 21/02 201D

テーマコード (参考)

審査請求 未請求 請求項の数 10 O L (全 76 頁)

(21) 出願番号	特願2009-265075 (P2009-265075)	(71) 出願人	000002185 ソニー株式会社 東京都港区港南1丁目7番1号
(22) 出願日	平成21年11月20日(2009.11.20)	(74) 代理人	100093241 弁理士 官田 正昭
		(74) 代理人	100101801 弁理士 山田 英治
		(74) 代理人	100086531 弁理士 澤田 俊夫
		(74) 代理人	100095496 弁理士 佐々木 榮二
		(74) 代理人	110000763 特許業務法人大同特許事務所
		(72) 発明者	廣江 厚夫 東京都港区港南1丁目7番1号 ソニー株式会社内

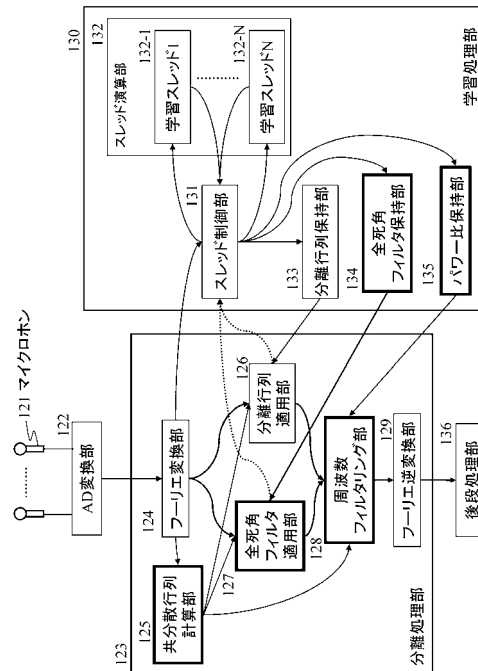
(54) 【発明の名称】 信号処理装置、および信号処理方法、並びにプログラム

(57) 【要約】

【課題】突発音等が含まれる混合信号に対する高精度な音源分離を可能とする装置、方法を提供する。

【解決手段】複数音源からの出力を混合した混合信号からなる観測信号に対して、独立成分分析(ICA)を適用した学習処理により分離行列を求めて分離信号を生成するとともに、観測信号として検出された音源に対する死角を持つ全死角空間フィルタを適用して検出音を除去した全死角空間フィルタ適用信号を生成する。さらに、分離信号に含まれる全死角空間フィルタ適用信号に対応する信号成分を除去するフィルタリング処理を実行し、周波数フィルタリング処理結果から音源分離結果を生成する。

【選択図】 図12



【特許請求の範囲】**【請求項 1】**

複数のセンサが取得した複数音源の出力の混合信号に対する短時間フーリエ変換 (S T F T) によって時間周波数領域の観測信号を生成し、該観測信号に対する線形フィルタリング処理により各音源対応の音源分離結果を生成する分離処理部を有し、

前記分離処理部は、

前記観測信号に対する線形フィルタリング処理を実行して各音源対応の分離信号を生成する線形フィルタリング処理部と、

前記複数のセンサによって取得された観測信号に含まれる全ての音源方向に死角を形成した全死角空間フィルタを適用して死角方向の音を除去した全死角空間フィルタ適用信号を生成する全死角空間フィルタ適用部と、

10

前記分離信号と前記全死角空間フィルタ適用信号を入力し、前記分離信号に含まれる前記全死角空間フィルタ適用信号に対応する信号成分を除去するフィルタリング処理を実行する周波数フィルタリング部を有し、前記周波数フィルタリング部の処理結果を音源分離結果として生成する信号処理装置。

【請求項 2】

前記信号処理装置は、

複数音源からの出力を混合した混合信号からなる観測信号に対して、独立成分分析 (I C A : I n d e p e n d e n t C o m p o n e n t A n a l y s i s) を適用した学習処理により、前記混合信号を分離する分離行列を求め、さらに前記観測信号から取得された全ての音源方向に死角を形成した全死角空間フィルタを生成する学習処理部を有し、

20

前記線形フィルタリング処理部は、

前記観測信号に対して、前記学習処理部の生成した分離行列を適用して前記混合信号を分離して各音源対応の分離信号を生成し、

前記全死角空間フィルタ適用部は、

前記観測信号に対して、前記学習処理部の生成した全死角空間フィルタを適用して死角方向の音を除去した全死角空間フィルタ適用信号を生成する請求項 1 に記載の信号処理装置。

【請求項 3】

前記周波数フィルタリング部は、

30

前記分離信号から前記全死角空間フィルタ適用信号を減算する処理により、前記分離信号に含まれる前記全死角空間フィルタ適用信号に対応する信号成分を除去するフィルタリング処理を実行する請求項 1 または 2 に記載の信号処理装置。

【請求項 4】

前記周波数フィルタリング部は、

前記全死角空間フィルタ適用信号を雑音成分としたスペクトル減算による周波数フィルタリング処理により、前記分離信号に含まれる前記全死角空間フィルタ適用信号に対応する信号成分を除去するフィルタリング処理を実行する請求項 1 または 2 に記載の信号処理装置。

【請求項 5】

40

前記学習処理部は、

前記観測信号を区分したブロック単位での学習処理を実行してブロック単位の学習結果に基づく分離行列および全死角空間フィルタを生成する処理を実行し、

前記分離処理部は、

前記学習処理部の生成した最新の分離行列および全死角空間フィルタを適用した処理を実行する請求項 2 に記載の信号処理装置。

【請求項 6】

前記周波数フィルタリング部は、

前記分離信号からの前記全死角空間フィルタ適用信号対応成分の除去レベルを分離信号チャンネルに応じて変更する処理を行う請求項 1 または 2 に記載の信号処理装置。

50

【請求項 7】

前記周波数フィルタリング部は、

前記分離信号からの前記全死角空間フィルタ適用信号対応成分の除去レベルを分離信号チャンネルのパワー比に応じて変更する処理を行う請求項 6 に記載の信号処理装置。

【請求項 8】

前記分離処理部は、

観測信号からの切り出しデータ単位であるフレーム中、現在の観測信号を含むフレームを適用したスケール調整としてのリスケーリング処理を実行した分離行列と、全死角空間フィルタを生成して、リスケーリング処理後の分離行列と全死角空間フィルタを適用した処理を行う請求項 2 に記載の信号処理装置。

10

【請求項 9】

信号処理装置において音源分離処理を実行する信号処理方法であり、

分離処理部において、複数のセンサが取得した複数音源の出力の混合信号に対する短時間フーリエ変換 (S T F T) によって時間周波数領域の観測信号を生成し、該観測信号に対する線形フィルタリング処理により各音源対応の音源分離結果を生成する分離処理ステップを有し、

前記分離処理ステップは、

前記観測信号に対する線形フィルタリング処理を実行して各音源対応の分離信号を生成する線形フィルタリング処理ステップと、

前記複数のセンサによって取得された観測信号に含まれる全ての音源方向に死角を形成した全死角空間フィルタを適用して死角方向の音を除去した全死角空間フィルタ適用信号を生成する全死角空間フィルタ適用ステップと、

20

前記分離信号と前記全死角空間フィルタ適用信号を入力し、前記分離信号に含まれる前記全死角空間フィルタ適用信号に対応する信号成分を除去するフィルタリング処理を実行する周波数フィルタリングステップを有し、前記周波数フィルタリングステップの処理結果を音源分離結果として生成する信号処理方法。

【請求項 10】

信号処理装置において音源分離処理を実行させるプログラムであり、

分離処理部において、複数のセンサが取得した複数音源の出力の混合信号に対する短時間フーリエ変換 (S T F T) によって時間周波数領域の観測信号を生成し、該観測信号に対する線形フィルタリング処理により各音源対応の音源分離結果を生成する分離処理ステップを実行させ、

30

前記分離処理ステップにおいては、さらに、

前記観測信号に対する線形フィルタリング処理を実行して各音源対応の分離信号を生成する線形フィルタリング処理ステップと、

前記複数のセンサによって取得された観測信号に含まれる全ての音源方向に死角を形成した全死角空間フィルタを適用して死角方向の音を除去した全死角空間フィルタ適用信号を生成する全死角空間フィルタ適用ステップと、

前記分離信号と前記全死角空間フィルタ適用信号を入力し、前記分離信号に含まれる前記全死角空間フィルタ適用信号に対応する信号成分を除去するフィルタリング処理を実行する周波数フィルタリングステップを実行させて、前記周波数フィルタリングステップの処理結果を音源分離結果として生成させるプログラム。

40

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、信号処理装置、および信号処理方法、並びにプログラムに関する。さらに、詳細には複数の信号が混合された信号を独立成分分析 (I C A : I n d e p e n d e n t C o m p o n e n t A n a l y s i s) を用いて分離する処理に関し、特にリアルタイム処理、すなわち連続的に入力される観測信号を少ない遅延で独立な成分へと分解し、それを連続的に出力する信号処理装置、および信号処理方法、並びにプログラムに関する

50

。【背景技術】

【0002】

まず、本発明の背景技術として、独立成分分析（ICA：Independent Component Analysis）について説明し、さらに、独立成分分析（ICA）のリアルタイム化の方法について説明する。

【0003】

[A1. ICAの説明]

ICAとは、多変量分析の一種であり、信号の統計的な性質を利用して多次元信号を分離する手法のことである。ICA自体の詳細については、例えば非特許文献1 [『入門・独立成分分析』（村田昇著、東京電機大学出版局）]などを参照されたい。

10

【0004】

以下では、音信号のICA、特に時間周波数領域のICAについて説明する。

図1に示すように、N個の音源から異なる音が鳴っていて、それらをn個のマイクで観測するという状況を考える。音源が発した音（原信号）がマイクに届くまでには、時間遅れや反射などがある。従って、マイクロホンkで観測される信号（観測信号）は式[1.1]のように、原信号と伝達関数（transfer function）との畳み込み演算（convolution）を全音源について総和した式して表わすことができる。この混合を以下では「畳み込み混合」（convolutive mixtures）と呼ぶ。

なお、マイクnの観測信号を $x_n(t)$ とする。マイク1、マイク2の観測信号はそれぞれ $x_1(t)$ 、 $x_2(t)$ となる。

20

全てのマイクについての観測信号を一つの式で表わすと、以下に示す式[1.2]のように表わせる。

【0005】

【数1】

$$x_k(t) = \sum_{j=1}^N \sum_{l=0}^L a_{kj}(l) s_j(t-l) = \sum_{j=1}^N \{ a_{kj} * s_j \} \quad \dots\dots[1.1]$$

$$\mathbf{x}(t) = \mathbf{A}^{[0]} \mathbf{s}(t) + \dots + \mathbf{A}^{[L]} \mathbf{s}(t-L) \quad \dots\dots[1.2]$$

ただし、

$$\mathbf{s}(t) = \begin{bmatrix} s_1(t) \\ \vdots \\ s_N(t) \end{bmatrix}, \quad \mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{bmatrix}, \quad \mathbf{A}^{[l]} = \begin{bmatrix} a_{11}(l) & \dots & a_{1N}(l) \\ \vdots & \ddots & \vdots \\ a_{n1}(l) & \dots & a_{nN}(l) \end{bmatrix} \quad \dots\dots[1.3]$$

30

【0006】

ただし、 $\mathbf{x}(t)$ 、 $\mathbf{s}(t)$ はそれぞれ $x_k(t)$ 、 $s_k(t)$ を要素とする列ベクトルであり、 $\mathbf{A}^{[l]}$ は $a_{kj}^{[l]}$ を要素とする $n \times N$ の行列である。以降では、 $n = N$ とする。

40

【0007】

時間領域の畳み込み混合は、時間周波数領域では瞬時混合で表わされることが知られており、その特徴を利用したのが時間周波数領域のICAである。

【0008】

時間周波数領域ICA自体については、非特許文献2 [『詳解 独立成分分析』の「19.2.4. フーリエ変換法」]や、特許文献1（特開2006-238409号公報『音声信号分離装置・雑音除去装置および方法』）などを参照されたい。

【0009】

以下では、主に本発明と関係ある点を説明する。

上記の式[1.2]の両辺を短時間フーリエ変換すると、以下に示す式[2.1]が得

50

られる。

【 0 0 1 0 】

【 数 2 】

$$X(\omega, t) = A(\omega)S(\omega, t) \quad \dots\dots[2.1]$$

$$X(\omega, t) = \begin{bmatrix} X_1(\omega, t) \\ \vdots \\ X_n(\omega, t) \end{bmatrix} \quad \dots\dots[2.2]$$

$$A(\omega) = \begin{bmatrix} A_{11}(\omega) & \dots & A_{1N}(\omega) \\ \vdots & \ddots & \vdots \\ A_{n1}(\omega) & \dots & A_{nN}(\omega) \end{bmatrix} \quad \dots\dots[2.3]$$

$$S(\omega, t) = \begin{bmatrix} S_1(\omega, t) \\ \vdots \\ S_N(\omega, t) \end{bmatrix} \quad \dots\dots[2.4]$$

$$Y(\omega, t) = W(\omega)X(\omega, t) \quad \dots\dots[2.5]$$

$$Y(\omega, t) = \begin{bmatrix} Y_1(\omega, t) \\ \vdots \\ Y_n(\omega, t) \end{bmatrix} \quad \dots\dots[2.6]$$

$$W(\omega) = \begin{bmatrix} W_{11}(\omega) & \dots & W_{1n}(\omega) \\ \vdots & \ddots & \vdots \\ W_{m1}(\omega) & \dots & W_{mn}(\omega) \end{bmatrix} \quad \dots\dots[2.7]$$

10

20

30

【 0 0 1 1 】

上記式 [2 . 1] において、

は周波数ピンの番号、

t はフレームの番号、

である。

【 0 0 1 2 】

を固定すると、この式は瞬時混合（時間遅れのない混合）と見なせる。そこで、観測信号を分離するには、分離結果 [Y] の算出式 [2 . 5] を用意した上で、分離結果：Y (, t) の各成分が最も独立になるように分離行列 W () を決める。

40

【 0 0 1 3 】

従来、時間周波数領域 I C A では、パーミュテーション問題と呼ばれる、「どの成分がどのチャンネルに分離されるか」が周波数ピンごとに異なるという問題が発生していたが、本願と同一発明者による前の特許出願である特許文献 1（特開 2 0 0 6 - 2 3 8 4 0 9 号公報）『音声信号分離装置・雑音除去装置および方法』に示した構成によって、このパーミュテーション問題は、ほぼ解決することができた。本発明でもこの方法を用いるため、特許文献 1（特開 2 0 0 6 - 2 3 8 4 0 9 号公報）に開示したパーミュテーション問題の解決手法について簡単に説明する。

【 0 0 1 4 】

特許文献 1（特開 2 0 0 6 - 2 3 8 4 0 9 号公報）では、分離行列 W () を求めるた

50

めに、以下に示す式 [3 . 1] から式 [3 . 3] までを分離行列 W () が収束するまで (または一定回数) 繰り返し実行する。

【 0 0 1 5 】

【 数 3 】

$$Y(\omega, t) = W(\omega)X(\omega, t) \quad (t = 1, \dots, T \quad \omega = 1, \dots, M) \quad \dots[3.1]$$

$$\Delta W(\omega) = \left\{ I + \left\langle \varphi_{\omega}(Y(t))Y(\omega, t)^H \right\rangle_t \right\} W(\omega) \quad \dots[3.2]$$

$$W(\omega) \leftarrow W(\omega) + \eta \Delta W(\omega) \quad \dots[3.3]$$

$$Y(t) = \begin{bmatrix} Y_1(1, t) \\ \vdots \\ Y_1(M, t) \\ \vdots \\ Y_n(1, t) \\ \vdots \\ Y_n(M, t) \end{bmatrix} = \begin{bmatrix} Y_1(t) \\ \vdots \\ Y_n(t) \end{bmatrix} \quad \dots[3.4]$$

$$\varphi_{\omega}(Y(t)) = \begin{bmatrix} \varphi_{\omega}(Y_1(t)) \\ \vdots \\ \varphi_{\omega}(Y_n(t)) \end{bmatrix} \quad \dots[3.5]$$

$$\varphi_{\omega}(Y_k(t)) = \frac{\partial}{\partial Y_k(\omega, t)} \log P(Y_k(t)) \quad \dots[3.6]$$

$P(Y_k(t))$: $Y_k(t)$ の確率密度関数 (PDF)

$$P(Y_k(t)) \propto \exp(-\gamma \|Y_k(t)\|_2) \quad \dots[3.7]$$

$$\|Y_k(t)\|_m = \left\{ \sum_{\omega=1}^M |Y_k(\omega, t)|^m \right\}^{1/m} \quad \dots[3.8]$$

$$\varphi_{\omega}(Y_k(t)) = -\gamma \frac{Y_k(\omega, t)}{\|Y_k(t)\|_2} \quad \dots[3.9]$$

10

20

30

$$W = \begin{bmatrix} W_{11}(1) & & \mathbf{0} & & W_{1n}(1) & & \mathbf{0} \\ & \ddots & & \dots & & \ddots & \\ \mathbf{0} & & W_{11}(M) & & \mathbf{0} & & W_{1n}(M) \\ \hline & \vdots & & \ddots & & \vdots & \\ W_{n1}(1) & & \mathbf{0} & & W_{nn}(1) & & \mathbf{0} \\ & \ddots & & \dots & & \ddots & \\ \mathbf{0} & & W_{n1}(M) & & \mathbf{0} & & W_{nn}(M) \end{bmatrix} \quad \dots\dots[3.10]$$

$$X(t) = \begin{bmatrix} X_1(1,t) \\ \vdots \\ X_1(M,t) \\ \vdots \\ X_n(1,t) \\ \vdots \\ X_n(M,t) \end{bmatrix} \quad \dots\dots[3.11]$$

$$Y(t) = WX(t) \quad \dots\dots[3.12]$$

10

20

【0016】

この繰り返し実行を以降では「学習」と呼ぶ。ただし、式[3.1]～式[3.3]は、全ての周波数ビンに対して行ない、さらに式[3.1]は、蓄積された観測信号の全てのフレームに対しても行なう。また、式[3.2]において、 $\langle \cdot \rangle_t$ は全フレームについての平均を表わす。 $Y(\cdot, t)$ の右上についている上付きのHはエルミート転置（ベクトルや行列の転置を取ると共に、要素を共役複素数に変換する）である。

【0017】

分離結果 $Y(t)$ は式[3.4]で表わされる、分離結果の全チャンネル・全周波数ビンの要素を並べたベクトルである。 $Y(t)$ は、式[3.5]で表わされるベクトルである。各要素 $Y_k(t)$ はスコア関数と呼ばれ、 $Y_k(t)$ の多次元（多変量）確率密度関数（PDF）の対数微分である（式[3.6]）。多次元PDFとして、例えば式[3.7]で表わされる関数を用いることができ、その場合、スコア関数 $Y_k(t)$ は式[3.9]のように表わせる。ただし、 $Y_k(t)_2$ はベクトル $Y_k(t)$ のL-2ノルム（全要素の2乗和を求め、さらに平方根をとったもの）である。L-2ノルムを一般化したL-mノルムは式[3.8]で定義される。式[3.7]および式[3.9]の $sqr t(M)$ は、 $Y_k(\cdot, t)$ のスケールを調整するための項であり、例えば $sqr t(M)$ （周波数ビン数の平方根）といった適切な正の定数を代入しておく。式[3.3]の α は学習率や学習係数と呼ばれる正の小さな値（例えば0.1程度）である。これは、式[3.2]で計算された $W(\cdot)$ を分離行列 $W(\cdot)$ に少しずつ反映させるために用いられる。

30

40

【0018】

なお、式[3.1]は一つの周波数ビンにおける分離（図2（A）参照）を表わしているが、全周波数ビンの分離を一つの式で表わす（図2（B）参照）ことも可能である。

【0019】

そのためには、上述した式[3.4]で表わされる全周波数ビンの分離結果 $Y(t)$ および、式[3.11]で表わされる観測信号 $X(t)$ 、さらに式[3.10]で表わされる全周波数ビン分の分離行列を用いればよく、それらのベクトルと行列を用いることで、

50

分離は式 [3 . 1 2] のように表わすことができる。本発明は、必要に応じて式 [3 . 1] と式 [3 . 1 1] とを使い分ける。

【 0 0 2 0 】

なお、図 2 に示した $X_1 \sim X_n$ および $Y_1 \sim Y_n$ の図はスペクトログラムと呼ばれ、短時間フーリエ変換 (S T F T) の結果を周波数ビン方向とフレーム方向とに並べたものである。縦方向が周波数ビン、横方向がフレームである。式 [3 . 4] や式 [3 . 1 1] では低い周波数を上に書いてあるが、スペクトログラムでは低い周波数を下に描いてある。

【 0 0 2 1 】

ここまでの説明では、音源数 N はマイクロホン数 n と等しいとしていたが、 $N < n$ であっても分離は可能である。その場合、 n 個の出力チャンネルの内の N 個には音源に対応した信号がそれぞれ出力されるが、残りの $n - N$ チャンネルには、どの音源にも対応しない、無音に近い信号が出力される。

10

【 0 0 2 2 】

[A 2 . I C A のリアルタイム化]

「 A 1 . I C A の説明」で説明した式 [3 . 1] から式 [3 . 3] までを分離行列 W () が収束するまで (または一定回数) 繰り返し実行する学習処理は、例えばバッチ処理によって行われる。すなわち、観測信号全体を蓄積した後で、式 [3 . 1] ~ 式 [3 . 3] を繰り返す方式である (式 [3 . 1] ~ 式 [3 . 3] の繰り返し実行は、前述したように、学習と呼ばれる。

【 0 0 2 3 】

このバッチ処理は、ある工夫を行なうことで、リアルタイム (低遅延) の音源分離に適用することができる。リアルタイム処理方式を実現した音源分離処理例として、本願と同一出願人の先の特許出願である特許文献 2 (特開 2 0 0 8 - 1 4 7 9 2 0 号公報「リアルタイム音源分離装置および方法」) に開示した構成について説明する。

20

【 0 0 2 4 】

特許文献 2 (特開 2 0 0 8 - 1 4 7 9 2 0 号公報) で開示した処理方式は、図 3 に示すように、観測信号のスペクトログラムを重複のある複数のブロック 1 ~ N に分割し、ブロックごとに学習を行なって分離行列を求めている。ブロックに重複を持たせる理由は、分離行列の精度と更新頻度とを両立させるためである。

【 0 0 2 5 】

なお、特許文献 2 (特開 2 0 0 8 - 1 4 7 9 2 0 号公報) より以前に開示されていたリアルタイム I C A (ブロック単位の I C A) ではブロック間に重複がなかった。従って、分離行列の更新間隔を短くするにはブロック長 (= 観測信号を蓄積する時間) を短くする必要があったが、ブロック長を短くすると分離精度は低下するという問題があった。

30

【 0 0 2 6 】

このように、観測信号の各ブロックに対してバッチ処理を適用する方法を、以降では「ブロック単位 (b l o c k w i s e) バッチ処理」と呼ぶ。

【 0 0 2 7 】

各ブロックから求めた分離行列は、それより以降の観測信号へ適用する (同一ブロックへは適用しない) ことで分離結果を生成する。ここでは、その方式を「ずらし適用」と呼ぶ。

40

【 0 0 2 8 】

図 4 は、「ずらし適用」の説明である。現時点において、 t 番目のフレームの観測信号 $X(t)$ 4 2 が入力されたとする。この時点では、観測信号 $X(t)$ を含むブロック (例えば現在時刻を含む観測信号ブロック 4 6) に対応した分離行列はまだ求まっていない。そこで、ブロック 4 6 の代わりに、それより前のブロックである学習データブロック 4 1 から学習された分離行列を観測信号 $X(t)$ に乗じることで、 $X(t)$ に対応した分離結果、すなわち現在時刻の分離結果 $Y(t)$ 4 4 を生成する。なお、学習データブロック 4 1 から学習された分離行列は、フレーム t の時点で既に求まっているものとする。

【 0 0 2 9 】

50

前述の通り、分離行列は混合過程の逆の過程を表わしていると考えられる。

そのため、学習データのブロック設定区間 4 1 の観測信号と現在時刻の観測信号 4 2 とで混合過程が同一であれば（例えば、音源とマイクとの位置関係が変化していなければ）、異なる区間で学習された分離行列を適用しても信号を分離することができ、そうすることで遅延の少ない分離を実現することができる。

【 0 0 3 0 】

特許文献 2（特開 2 0 0 8 - 1 4 7 9 2 0 号公報）において開示した構成では、重複のあるブロックから分離行列を求めるためスレッドと呼ばれる処理の単位を、時刻をずらして複数並列に起動する方式を提案している。図 5 を参照してこの並列処理方式について説明する。

10

【 0 0 3 1 】

図 5 には、処理単位としてのスレッド各々の時間経過に従った処理推移を示している。図 5 には 6 つのスレッド 1 ~ 6 を示している。各スレッドは A) 蓄積、B) 学習、C) 待機、という 3 つの状態を繰り返す。すなわち、スレッド長は、A) 蓄積、B) 学習、C) 待機の 3 つの処理のトータル時間長に対応する。図 5 の左から右に時間が推移する。

【 0 0 3 2 】

「A) 蓄積」は図 5 の濃い灰色の区間であり、スレッドがこの状態にあるときは、観測信号を蓄積する。蓄積を開始する時刻をスレッドごとにずらすことで、図 5 の重複つきブロックが表現できる。図 5 では蓄積時間の 1 / 4 だけずらしているため、1 つのスレッドの蓄積時間を例えば 4 秒とすると、各スレッド間のずらし時間は 1 秒となる。

20

【 0 0 3 3 】

観測信号を一定時間（例えば 4 秒）蓄積すると、各スレッドは「B) 学習」に状態を遷移させる。「B) 学習」は図 5 の薄い灰色の区間であり、この状態にあるときは、蓄積された観測信号に対して先に説明した式 [3 . 1] ~ 式 [3 . 3] を繰り返し実行する。

【 0 0 3 4 】

学習（式 [3 . 1] ~ 式 [3 . 3] の繰り返し）によって分離行列 W が十分に収束したら（または単に一定回数繰り返したら）、学習を終了し、スレッドは「C) 待機」状態に遷移する（図 5 の白い区間）。「待機」は、蓄積開始時刻および学習開始時刻をスレッド間で一定の間隔に保つためであり、その結果、学習終了時刻（= 分離行列が更新される時刻）もほぼ一定の間隔に保たれる。

30

【 0 0 3 5 】

学習で求めた分離行列 W は、次のスレッドの学習が終了するまでの間、分離に使用される。すなわち、図 4 の分離行列 4 3 として使用される。これを説明するのが図 5 の最下段に示す分離行列の時間推移に示す適用分離行列規定区間 5 1 ~ 5 3 に使用される分離行列である。

【 0 0 3 6 】

システムが起動してから、最初の分離行列が学習されるまでの適用分離行列規定区間 5 1 では、図 4 の分離行列 4 3 として初期値（例えば単位行列）が使用される。図 5 に示すスレッド 1 の学習が終了してからスレッド 2 の学習が終了するまでの区間 5 2 では、スレッド 1 の観測信号蓄積区間 5 4 に由来する分離行列が図 4 の分離行列 4 3 として使用される。図 5 の区間 5 2 に示す数字「1」は、この期間に使用される分離行列 W がスレッド 1 の処理によって求められたことを表わす。適用分離行列規定区間 5 2 よりも右側の数字も同様に、分離行列が何番目のスレッドに由来するかを表わす。

40

【 0 0 3 7 】

なお、学習を開始する時点において、他のスレッドで求めた分離行列が存在する場合は、それを学習の初期値として使う。これを「分離行列の引継ぎ」と呼ぶことにする。図 5 に示す例では、スレッド 3 の最初の学習が始まるタイミングである学習開始タイミング 5 5 では、スレッド 1 に由来する分離行列 5 2 が既に求まっているので、それを学習の初期値として使う。

【 0 0 3 8 】

50

このような処理を行うことで、スレッド間でパーミュテーション (p e r m u t a t i o n) の発生を防止または低減することができる。スレッド間のパーミュテーションとは、例えば、1番目のスレッドで求めた分離行列では1番目のチャンネルに音声、2番目のチャンネルに音楽が出力されるのに対し、3番目のスレッドで求めた分離行列ではそれが逆転しているなどの問題である。

【0039】

図5を参照して説明したように、他のスレッドで求めた分離行列が存在する場合は、それを学習の初期値として使う「分離行列の引継ぎ」を行うことで、スレッド間のパーミュテーションを低減することが可能となる。また、スレッド1の学習では分離行列が十分には収束していなくても、それを次のスレッドが引き継ぐことで、収束の度合いを向上させることができる。

10

【0040】

このように時間をずらしながら複数のスレッドを起動することで、分離行列は、スレッド間のずれ、すなわちブロックシフト量56とほぼ同じ間隔で更新されていくのである。

【0041】

[B . 従来技術の問題点]

次に、上述の「A2 . I C A のリアルタイム化」の問題点について検証する。上述の「A2 . I C A のリアルタイム化」において説明した「ブロック単位バッチ処理」と「ずらし適用」との組み合わせでは、音源分離が正確に行われない場合が発生する。それは以下の2つの要因に分けて考えることができる。

20

B1 . 追従遅れ

B2 . 消し残り

以下、この2つの要因によって音源分離が正確に行われない理由について各々説明する。

【0042】

[B1 . 追従遅れ]

「ずらし適用」を用いると、分離行列の学習に用いた区間（例えば図4に示す学習データブロック41）と現在時刻の観測信号42との間で音源が変化した場合（音源が移動したり、急に鳴り始めたりした場合）に、一時的に不整合が生じる。

【0043】

30

その後、変化した音源を観測した学習処理によって新たな分離行列が求まるため、やがて不整合は消える。しかし、その新たな分離行列が生成されるまでの間は、不整合が発生することになる。ここではその現象を「追従遅れ」と呼ぶ。追従遅れは、たとえ音源が移動しなくても、音が急に鳴り始めた場合や、音がいったん鳴り止んだ後に再び鳴り始めた場合などにも発生する。以降では、これらの音を「突発音」と呼ぶ。

【0044】

図6は突発音と観測信号との対応について説明する図である。この図6の例は音源が2つあると想定している。

(a) 音源 1

(b) 音源 2

これらの2つの音源である。

40

左から右に時間が経過している。(a) 音源 1、(b) 音源 2、(c) 観測信号に示すブロックの高さは音量を示すものとする。

【0045】

(a) 音源 1 は無音区間 6 7 を挟んで 2 回鳴っている。音源出力区間をそれぞれ音源 1 出力区間 6 1 , 6 2 とする。現在時刻の観測信号 6 6 を観測している現在時刻においても出力されている。

(b) 音源 2 は、連続的に鳴っているとする。すなわち音源 2 出力区間 6 3 を有する。

(c) 観測信号は、これらの音源 1、音源 2 からマイクロホンへ到達した信号の和として表せる。

50

【0046】

(c) 観測信号中に点線枠で示す学習データのブロック64は、図4に示す学習データのブロック41と同じ区間であり、学習データのブロック64の区間の観測信号から学習された分離行列を現在時刻(t_1)の観測信号66に適用することで分離を行なう。学習データのブロック64と現在時刻(t_1)の観測信号66との間には区間65(ブロック終端から現在時刻までの区間65)が存在する。

現在時刻(t_1)の観測信号66は、現在時刻の音源出力69に基づく観測信号である。

【0047】

しかし、音源1の無音区間67と学習データのブロック長64(図4に示す学習データのブロック41と同一)との長さによっては、学習データと現在の観測信号との間で不整合が発生する場合がある。

【0048】

例えば(c)観測信号中、現在時刻(t_1)の観測信号66には音源1由来の音源1出力区間62と、音源2由来の音源2出力区間63との両方が観測信号として含まれるのに対し、学習データのブロック64は音源2由来の音源2出力区間63しか観測されていない。

【0049】

現在時刻(t_1)の観測信号66のように、学習データのブロックには含まれない音が現在鳴っていることを、「突発音が発生した」と表現する。言い換えると、学習データのブロック64には音源1の観測信号が含まれていないため、たとえそのブロックよりも前に音源1が鳴っていた(音源1出力区間61に相当)としても、学習データのブロック64において学習された分離行列にとって音源1(音源1出力区間62の区間)は突発音である。

【0050】

図7は突発音の発生が分離結果に及ぼす影響、特に追従遅れについて説明する図である。図7には、

(a) 観測信号

(b1) 分離結果1

(b2) 分離結果2

(b3) 分離結果3

これらのデータを示している。

図の左から右に時間が経過している。

【0051】

図7に示す例において、ICA(独立成分分析)システムは3個以上のマイクロホンを有し、出力チャンネルも3以上であると想定している。

(a) 観測信号には、時間 $t_0 \sim t_5$ の間、継続して鳴り続けている継続音71と、時間 $t_1 \sim t_4$ の間のみ出力される突発音72が含まれている。

図7の(a)観測信号は、図6(c)観測信号と同様の観測信号であり、継続音71は、例えば図6の(b)音源2に、突発音72は図6の(a)音源1に対応する。

【0052】

突発音72の出力開始前、継続音71のみが鳴っている $t_0 \sim t_1$ 区間73において分離行列が十分に収束した後では、継続音71に対応した信号は一つのチャンネルのみ出力される。これが(b1)分離結果1である。他のチャンネル、すなわち(b2)分離結果2と、(b3)分離結果3にはほぼ無音が出力される。

【0053】

ここで、突発音72が発生したとする。例えば、それまで黙っていた人がしゃべり始めた場合などである。この時点で観測信号に適用可能な分離行列は、突発音72が発生する以前のデータ、すなわち、時間 t_1 以前の継続音71のデータのみを観測データとして学習して生成された分離行列となる。

10

20

30

40

50

【 0 0 5 4 】

この結果、時間 t_1 以前の観測信号に基づいて生成した分離行列を適用して、時間 t_1 以降の突発音 7 2 が観測された観測信号を分離することになり、観測信号に対応する正しい分離結果は得られない、すなわち、時間 t_1 以前の観測信号に基づいて生成した分離行列は、時間 t_1 以降の観測信号に含まれる突発音 7 2 を考慮しない分離行列であるからである。この結果、その分離行列を適用した分離結果、例えば、時間 $t_1 \sim t_3$ の分離結果と、実際の観測信号、すなわち継続音 7 1 と突発音 7 2 との混合である観測信号との間に不整合が発生する。

【 0 0 5 5 】

突発音が鳴り始めてから、その突発音を反映した分離行列が学習されるまでの間（時間 $t_1 \sim t_2$ の区間 7 4）は、全部のチャンネル（（b 1）分離結果 1、（b 2）分離結果 2、（b 3）分離結果 3）に突発音が出力されるという現象が発生する。すなわち、突発音については全く音源分離がなされない。この時間は、最小で学習時間より若干大きい値、最大で学習時間とブロックシフト幅との和である。例えば、学習時間が 0.3 秒、ブロックシフトが 0.2 秒というシステムでは、最小で 0.3 秒強、最大で 0.5 秒は突発音が分離せずに全チャンネルに出力される。

10

【 0 0 5 6 】

その後、新たな学習ブロックにおける学習処理により逐次、新たな分離行列が生成され更新される。この分離行列更新処理によって、突発音が分離行列に反映されるにつれて一つのチャンネル（図 7 では（b 2）分離結果 2）を除いて突発音の出力が小さくなる（時間 $t_2 \sim t_3$ の区間 7 5）。やがて一つのチャンネル（（b 2）分離結果 2）のみに出力されるようになる（ $t_3 \sim$ の区間 7 6）。

20

【 0 0 5 7 】

この図 7 に示す例において、追従遅れの発生区間は、時間 $t_1 \sim t_2$ の区間 7 4 と時間 $t_2 \sim t_3$ の区間 7 5 を合わせた区間、すなわち時間 $t_1 \sim t_3$ の区間 7 7 である。

【 0 0 5 8 】

突発音発生時に生ずる追従遅れの問題点がどこにあるかについては、突発音が目的音であるか妨害音であるかによって異なる。以降、それぞれの場合について説明する。目的音とは、解析対象とする音である。

【 0 0 5 9 】

突発音が妨害音である場合、言い換えると鳴りっぱなし継続音 7 1 が目的音である場合、突発音は除去されるのが望ましい。従って、問題は、図 7 に示す（b 1）分離結果 1 において妨害音が除去されずに残っていることである。

30

【 0 0 6 0 】

一方、突発音が目的音である場合、突発音は残すが、妨害音である鳴りっぱなしの継続音 7 1 は除去されるのが望ましい。一看すると、図 7 に示す（b 2）分離結果 2 はそのような出力に見える。しかし、追従遅れが発生している時間 $t_1 \sim t_3$ の区間 7 7 では入力と分離行列とで不整合が発生しているため、出力の音が歪む（周波数間のバランスが原信号とは異なったものとなっている）可能性がある。すなわち、突発音が目的音である場合、問題は出力音が歪む場合があることである。

40

【 0 0 6 1 】

このように、突発音の性質によって、除去するか残すかという相反する処理を行なう必要があるため、単一の方法で解決するのは困難である。

【 0 0 6 2 】

[B 2 . 消し残り]

次に、前述の「A 2 . I C A のリアルタイム化」において説明した「ブロック単位バッチ処理」と「ずらし適用」との組み合わせにおいて、音源分離が正確に行われぬもう一つの要因である「消し残り」について説明する。

【 0 0 6 3 】

例えば、図 7 の時間 $t_0 \sim t_1$ の区間 7 3、あるいは時間 $t_3 \sim t_4$ の区間 7 6 等では

50

分離行列が十分収束し、先行する学習データに基づく分離行列を適用して観測データの分離を行えば、正確な分離が行えているはずである。しかし、このような区間でも一つのチャンネルに完全に一つの音源が出力されるとは限らず、他の音源がある程度残る。これを「消し残り」と呼ぶ。例えば、図7に示す消し残り78は、(b2)分離結果には残るべきでない音である。同様に、消し残り79も(b3)分離結果3には出現すべきでない音である。

【0064】

このような消し残りが発生する主な要因として、以下の点が考えられる。

a) その空間の残響の長さが、短時間フーリエ変換(STFT)のフレーム長よりも長い。

b) 音源の数が、マイクロホンの数よりも多い。

c) マイク間隔が狭いため低い周波数において妨害音が消しきれない。

【0065】

リアルタイムICAを用いた音源分離システムにおいては、追従遅れの短縮と消し残りの減少とがトレードオフになり得る。なぜなら、追従遅れの短縮には学習時間の短縮が有効だが、その方法によっては消し残りが増加してしまうからである。

【0066】

ICAの学習の計算量(computational cost)は、短時間フーリエ変換(STFT)のフレーム長に比例し、また、チャンネル数(マイクロホンの個数)の2乗に比例する。従って、それらの値を小さくすると、ループ回数が同じでも学習時間を短縮できるため、追従遅れも短縮できる。

【0067】

しかし、フレーム長の短縮は、上記の消し残り発生の要因の1つ、すなわち、要因a)をさらに悪化させることになる。

また、マイクロホン数の減少は上記の消し残り発生の要因の1つ、すなわち要因b)をさらに悪化させることになる。

【0068】

従って、短時間フーリエ変換(STFT)のフレーム長を短くする処理や、チャンネル数(マイクロホンの数)を減少させる処理は、追従遅れの短縮には貢献するが、その一方で、消し残りが発生しやすくなるという問題を生じさせる。

このように、追従遅れの短縮と消し残りは一方を解消しようとするとも一方が悪化するという関係にある。

【0069】

図7に示す消し残り78は、鳴りっぱなしの継続音、すなわち、(b1)分離結果1に対応する音として分離されるべきであり、消し残りが発生すると、そのチャンネルで支配的に出力されている成分((b1)分離結果1においては突発音72)に対する分離性能が低下することになる。

一方、前述の「追従遅れ」が大きいと、突発音の正確な分離結果を得る時間が遅延することになる。具体的には、図7に示す突発音の発生時間 t_1 から、突発音に対応するチャンネル、すなわち(b2)分離結果2にのみ、突発音に対応する音が分離される時間である時間 t_3 に至るまでの時間が延びてしまうことになる。

【0070】

複数の音源からどの音源の音を取得したいか、これは目的に応じて異なる場合がある。ここで正確な分離結果を取得したい目的とする音を「目的音」とする。

この「目的音」が、鳴りっぱなしの継続音と突発音とのどちらであるかによって、異なる処理や設定をすることが望ましいということになる。

【0071】

上記の消し残り発生の要因の残りの1つ、すなわち、

c) マイク間隔が狭いため低い周波数において妨害音が消しきれない。

この要因はリアルタイム処理とは無関係である。しかし、以下に説明する本発明の構成

10

20

30

40

50

によって解決できる課題であるため、ここで説明する。時間周波数領域のICAにおいて、マイクロホンの間隔が狭い（例えば2～3cm程度）と、特に低い周波数で分離が十分には行なわれないことがある。これは、マイクロホン間で十分な位相差が得られないのが原因である。マイクロホン間隔を広げることで低い周波数での分離精度は向上できるが、逆に空間エリアシング（spatial aliasing）と呼ばれる現象により、高い周波数での分離精度が低下する可能性もある。また、物理的制約により、マイクロホンを広い間隔では設置できない場合もある。

【0072】

以上の問題点をまとめると、以下の通りである。

(A) 「ブロック単位処理」および「ずらし適用」を用いたリアルタイムICAでは、突発音に対して「追従遅れ」や「消し残り」が発生し、音源分離が正確に行われない場合が発生する。

(B) 音源分離を正確に行うための「追従遅れ」や「消し残り」の対処は、突発音が目的音か妨害音かで相反しているため、単一の方法では解決が難しい。

(C) 従来リアルタイムICAの枠組みでは、「追従遅れ」の短縮と「消し残り」の解消とはトレードオフの関係になる場合がある。

【先行技術文献】

【特許文献】

【0073】

【特許文献1】特開2006-238409号公報

【特許文献2】特開2008-147920号公報

【非特許文献】

【0074】

【非特許文献1】『入門・独立成分分析』（村田昇著、東京電機大学出版局）

【非特許文献2】『詳解 独立成分分析』の「19.2.4. フーリエ変換法」

【発明の概要】

【発明が解決しようとする課題】

【0075】

本発明は、このような状況に鑑みてなされたものであり、独立成分分析（Independent Component Analysis；ICA）を用いて各音源信号単位の高精度な分離処理を遅延の少ないリアルタイム処理として実行する信号処理装置、および信号処理方法、並びにプログラムを提供することを目的とする。

【課題を解決するための手段】

【0076】

本発明の第1の側面は、

複数のセンサが取得した複数音源の出力の混合信号に対する短時間フーリエ変換（STFT）によって時間周波数領域の観測信号を生成し、該観測信号に対する線形フィルタリング処理により各音源対応の音源分離結果を生成する分離処理部を有し、

前記分離処理部は、

前記観測信号に対する線形フィルタリング処理を実行して各音源対応の分離信号を生成する線形フィルタリング処理部と、

前記複数のセンサによって取得された観測信号に含まれる全ての音源方向に死角を形成した全死角空間フィルタを適用して死角方向の音を除去した全死角空間フィルタ適用信号を生成する全死角空間フィルタ適用部と、

前記分離信号と前記全死角空間フィルタ適用信号を入力し、前記分離信号に含まれる前記全死角空間フィルタ適用信号に対応する信号成分を除去するフィルタリング処理を実行する周波数フィルタリング部を有し、前記周波数フィルタリング部の処理結果を音源分離結果として生成する信号処理装置にある。

【0077】

さらに、本発明の信号処理装置の一実施態様において、前記信号処理装置は、複数音源

からの出力を混合した混合信号からなる観測信号に対して、独立成分分析（ICA：Independent Component Analysis）を適用した学習処理により、前記混合信号を分離する分離行列を求め、さらに前記観測信号から取得された全ての音源方向に死角を形成した全死角空間フィルタを生成する学習処理部を有し、前記線形フィルタリング処理部は、前記観測信号に対して、前記学習処理部の生成した分離行列を適用して前記混合信号を分離して各音源対応の分離信号を生成し、前記全死角空間フィルタ適用部は、前記観測信号に対して、前記学習処理部の生成した全死角空間フィルタを適用して死角方向の音を除去した全死角空間フィルタ適用信号を生成する。

【0078】

さらに、本発明の信号処理装置の一実施態様において、前記周波数フィルタリング部は、前記分離信号から前記全死角空間フィルタ適用信号を減算する処理により、前記分離信号に含まれる前記全死角空間フィルタ適用信号に対応する信号成分を除去するフィルタリング処理を実行する。

10

【0079】

さらに、本発明の信号処理装置の一実施態様において、前記周波数フィルタリング部は、前記全死角空間フィルタ適用信号を雑音成分としたスペクトル減算による周波数フィルタリング処理により、前記分離信号に含まれる前記全死角空間フィルタ適用信号に対応する信号成分を除去するフィルタリング処理を実行する。

【0080】

さらに、本発明の信号処理装置の一実施態様において、前記学習処理部は、前記観測信号を区分したブロック単位での学習処理を実行してブロック単位の学習結果に基づく分離行列および全死角空間フィルタを生成する処理を実行し、前記分離処理部は、前記学習処理部の生成した最新の分離行列および全死角空間フィルタを適用した処理を実行する。

20

【0081】

さらに、本発明の信号処理装置の一実施態様において、前記周波数フィルタリング部は、前記分離信号からの前記全死角空間フィルタ適用信号対応成分の除去レベルを分離信号チャンネルに応じて変更する処理を行う。

【0082】

さらに、本発明の信号処理装置の一実施態様において、前記周波数フィルタリング部は、前記分離信号からの前記全死角空間フィルタ適用信号対応成分の除去レベルを分離信号チャンネルのパワー比に応じて変更する処理を行う。

30

【0083】

さらに、本発明の信号処理装置の一実施態様において、前記分離処理部は、観測信号からの切り出しデータ単位であるフレーム中、現在の観測信号を含むフレームを適用したスケール調整としてのリスケーリング処理を実行した分離行列と、全死角空間フィルタを生成して、リスケーリング処理後の分離行列と全死角空間フィルタを適用した処理を行う。

【0084】

さらに、本発明の第2の側面は、

信号処理装置において音源分離処理を実行する信号処理方法であり、

分離処理部において、複数のセンサが取得した複数音源の出力の混合信号に対する短時間フーリエ変換（STFT）によって時間周波数領域の観測信号を生成し、該観測信号に対する線形フィルタリング処理により各音源対応の音源分離結果を生成する分離処理ステップを有し、

40

前記分離処理ステップは、

前記観測信号に対する線形フィルタリング処理を実行して各音源対応の分離信号を生成する線形フィルタリング処理ステップと、

前記複数のセンサによって取得された観測信号に含まれる全ての音源方向に死角を形成した全死角空間フィルタを適用して死角方向の音を除去した全死角空間フィルタ適用信号を生成する全死角空間フィルタ適用ステップと、

前記分離信号と前記全死角空間フィルタ適用信号を入力し、前記分離信号に含まれる前

50

記全死角空間フィルタ適用信号に対応する信号成分を除去するフィルタリング処理を実行する周波数フィルタリングステップを有し、前記周波数フィルタリングステップの処理結果を音源分離結果として生成する信号処理方法にある。

【0085】

さらに、本発明の第3の側面は、

信号処理装置において音源分離処理を実行させるプログラムであり、

分離処理部において、複数のセンサが取得した複数音源の出力の混合信号に対する短時間フーリエ変換 (STFT) によって時間周波数領域の観測信号を生成し、該観測信号に対する線形フィルタリング処理により各音源対応の音源分離結果を生成する分離処理ステップを実行させ、

10

前記分離処理ステップにおいては、さらに、

前記観測信号に対する線形フィルタリング処理を実行して各音源対応の分離信号を生成する線形フィルタリング処理ステップと、

前記複数のセンサによって取得された観測信号に含まれる全ての音源方向に死角を形成した全死角空間フィルタを適用して死角方向の音を除去した全死角空間フィルタ適用信号を生成する全死角空間フィルタ適用ステップと、

前記分離信号と前記全死角空間フィルタ適用信号を入力し、前記分離信号に含まれる前記全死角空間フィルタ適用信号に対応する信号成分を除去するフィルタリング処理を実行する周波数フィルタリングステップを実行させて、前記周波数フィルタリングステップの処理結果を音源分離結果として生成させるプログラムにある。

20

【0086】

なお、本発明のプログラムは、例えば、様々なプログラム・コードを実行可能な画像処理装置やコンピュータ・システムに対して、コンピュータ可読な形式で提供する記憶媒体、通信媒体によって提供可能なプログラムである。このようなプログラムをコンピュータ可読な形式で提供することにより、情報処理装置やコンピュータ・システム上でプログラムに応じた処理が実現される。

【0087】

本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。なお、本明細書においてシステムとは、複数の装置の論理的集合構成であり、各構成の装置が同一筐体内にあるものには限らない。

30

【発明の効果】

【0088】

本発明の一実施例の構成によれば、複数音源からの出力を混合した混合信号からなる観測信号に対して、独立成分分析 (ICA: Independent Component Analysis) を適用した学習処理により、混合信号を分離する分離行列を求めて分離信号を生成するとともに、観測信号として検出された音源に対する死角を持つ全死角空間フィルタを適用して検出音を除去した全死角空間フィルタ適用信号を生成する。さらに、分離信号に含まれる全死角空間フィルタ適用信号に対応する信号成分を除去するフィルタリング処理を実行し、周波数フィルタリング処理結果から音源分離結果を生成する。本構成により、例えば突発音等が含まれる混合信号に対する高精度な音源分離が可能となる。

40

【図面の簡単な説明】

【0089】

【図1】N個の音源から異なる音が鳴っていて、それらをn個のマイクで観測するという状況について説明する図である。

【図2】周波数ビンにおける分離 (図2(A)参照) と、全周波数ビンの分離処理 (図2(B)参照) について説明する図である。

【図3】観測信号のスペクトログラムを重複のある複数のブロック1~Nに分割し、ブロックごとに学習を行なって分離行列を求める処理例について説明する図である。

50

【図4】各ブロックから求めた分離行列を、それより以降の観測信号へ適用する「ずらし適用」について説明する図である。

【図5】重複のあるブロックから分離行列を求めるためスレッドと呼ばれる処理の単位を、時刻をずらして複数並列に起動する方式について説明する図である。

【図6】突発音の発生と観測信号との対応について説明する図である。

【図7】突発音の発生が分離結果に及ぼす影響、特に追従遅れについて説明する図である。

【図8】フレーム単位のリスキューリング処理について説明する図である。

【図9】例えば図7に示す(b1)分離結果1から全死角空間フィルタの結果を減算して突発音を相殺して音源対応出力のみを残す処理について説明する図である。

【図10】2チャンネル周波数フィルタリングについて説明する図である。

【図11】本発明の具体的な2チャンネル周波数フィルタリング処理について説明する図である。

【図12】本発明の一実施例に従った信号処理装置の構成例について説明する図である。

【図13】学習処理部のスレッド制御部の詳細構成例について説明する図である。

【図14】スレッド演算部において実行する処理について説明する図である。

【図15】学習スレッドの状態遷移について説明する図である。

【図16】学習スレッドの状態遷移について説明する図である。

【図17】音源分離処理の全体シーケンスについて説明するフローチャートを示す図である。

【図18】短時間フーリエ変換の詳細について説明する図である。

【図19】図17に示すフローチャートにおけるステップS101の初期化処理の詳細について説明するフローチャートを示す図である。

【図20】スレッド制御部131による複数の学習スレッド1,2に対する制御シーケンスを示す図である。

【図21】図17に示すフローチャートにおけるステップS105においてスレッド制御部131によって実行されるスレッド制御処理について説明するフローチャートを示す図である。

【図22】図21に示すフローチャートにおけるステップS203において実行される待機中状態における処理について説明するフローチャートを示す図である。

【図23】図21に示すフローチャートのステップS204において実行される蓄積中状態における処理について説明するフローチャートを示す図である。

【図24】図21に示すフローチャートのステップS205において実行される学習中状態における処理について説明するフローチャートを示す図である。

【図25】図24に示すフローチャートのステップS239において実行される、分離行列等更新処理について説明するフローチャートを示す図である。

【図26】図24に示すフローチャートのステップS241において実行される待機時間の設定処理について説明するフローチャートを示す図である。

【図27】図17に示すフローチャートにおけるステップS106において実行される分離処理について説明するフローチャートを示す図である。

【図28】パワー比の算出に適用する関数の例を示す図である。

【図29】学習スレッドの処理について説明するフローチャートを示す図である。

【図30】図29に示すフローチャートのステップS394において実行される、コマンド処理について説明するフローチャートを示す図である。

【図31】図30に示すフローチャートのステップS405において実行される処理の一例である分離行列の学習処理例について説明するフローチャートを示す図である。

【図32】図31に示すフローチャートのステップS420において実行する後処理について説明するフローチャートを示す図である。

【図33】「全死角空間フィルタ&周波数フィルタリング」と線形フィルタリングとを組み合わせた場合の構成例について説明する図である。

10

20

30

40

50

【図34】線形フィルタリングを行う分散最小ビームフォーマ(MVBF)の適用例について説明する図である。

【発明を実施するための形態】

【0090】

以下、図面を参照しながら本発明の信号処理装置、および信号処理方法、並びにプログラムの詳細について説明する。説明は以下の項目に従って行う。

1. 本発明の構成と処理の概要について
2. 本発明の信号処理装置の具体的実施例について
3. 本発明の信号処理装置の実行する音源分離処理について
 - 3-1. 全体シーケンス
 - 3-2. 初期化処理について
 - 3-3. スレッド制御処理について
 - 3-4. 分離処理について
4. スレッド演算部における学習スレッドの処理について
5. 本発明の信号処理装置のその他の実施例(変形例)について
6. 本発明の信号処理装置の構成に基づく効果についてのまとめ

10

【0091】

[1. 本発明の構成と処理の概要について]

まず、本発明の構成と処理の概要について説明する。

本発明は、複数の信号が混合された信号を独立成分分析(ICA: Independent Component Analysis)を用いて分離する処理を行うものである。しかしながら、前述したように、先行する観測データに基づいて生成した分離行列を用いた音源分離処理を行うと、突発音に対する分離ができないという問題がある。本発明では、例えばこのような突発音に関する問題を解決するため、例えば本出願人の先の特許出願(特開2008-147920号公報)に開示した従来リアルタイムICAシステムに対して以下の要素を新規に追加した構成を持つ。

20

【0092】

(1) 突発音の歪みの問題に対処するため、分離結果のリスケーリング(周波数間のバランスを原信号に近づける処理)をフレーム単位で行なう構成。

なお、この処理を「頻繁リスケーリング」と呼ぶ。

30

【0093】

(2) 突発音を除去するために、検出された全部の音源方向に死角を向けるフィルタ(以降「全死角空間フィルタ(all-null spatial filter)」)をICAの学習データと同一の区間から生成する構成。さらに、観測信号にICAの分離結果を適用した結果と、同観測信号に全死角空間フィルタを適用した結果との間で、周波数フィルタリングに相当する処理または周波数フィルタリングを行なう構成。

なお、この処理構成を、「全死角空間フィルタ&周波数フィルタリング」と呼ぶ。

【0094】

(3) 突発音の性質によって異なる対処をするため、ICAの各出力チャンネルが音源に対応した信号を出力しているかどうか判別し、その結果に応じて以下のどちらかの処理をする構成。

40

i) 音源に対応していると判別された場合は、「頻繁リスケーリング」と「全死角空間フィルタ&周波数フィルタリング」との両方を適用する。

その結果、突発音はそのチャンネルからは除去される。

ii) 音源に対応していないと判別された場合は、「頻繁リスケーリング」のみを適用する。その結果、突発音はそのチャンネルから出力される。

なお、この処理構成を「チャンネル別の判別」と呼ぶ。

【0095】

以下では、まず、上記(1)~(3)のそれぞれについての概要を説明する。

- (1) 頻繁リスケーリング

50

本出願人の先の特許出願である特開 2008 - 147920 では、リスケーリングは学習終了時に分離行列に対して行なわれていた。

図 5 を参照して、この分離行列のリスケーリング処理について説明する。

例えば図 5 に示すスレッド 2 の学習区間 58 の学習が終了したときに、学習データ 59 を用いて分離行列のスケール（周波数間のバランス）が決定され、次に分離行列が更新されるまでスケールは一定であった。その場合、学習データ 59 に含まれている音源に対しては正しいスケールで出力されるが、それ以外の音源（すなわち突発音）に対しては正しくないスケールで出力される可能性があった。

【0096】

そこで本発明では、リスケーリング（周波数間のバランスを原信号に近づける処理）をフレーム単位で行なうことで、突発音の歪みを低減させる。フレーム単位のリスケーリング処理について図 8 を参照して説明する。

【0097】

図 8 には、先に説明した図 4 と同様、

（A）観測信号スペクトログラム

（B）分離結果スペクトログラム

これらの各データを示している。

図 8 に示す学習データのブロック 81 は、図 4 に示す学習データのブロック 41 に対応する。

図 8 に示す現在時刻の観測信号 82 は、図 4 に示す現在時刻の観測信号 42 に対応する。

図 8 に示す分離行列 83 は、図 4 に示す分離行列 43 に対応する。図 8 に示す分離行列 83 は、学習データブロック 81 から求められた分離行列である。

【0098】

従来のリスケーリングは、学習データブロック 81 の学習データを用いて行なわれていた。それに対して、以下に説明する本発明の処理では、現在時刻を終端とする一定長のブロック、すなわち、図 8 に示す現在時刻を含むブロック 87 を設定し、この現在時刻を含むブロック 87 の区間の観測信号を用いてリスケーリングを行なう。リスケーリングの具体的な式については後述する。このようなリスケーリング処理を行うことで、突発音に対しても早い段階でスケールを合わせる（=歪みを低減する）ことができる。

【0099】

（2）全死角空間フィルタ&周波数フィルタリング

次に、突発音を除去するための有効な処理である「全死角空間フィルタ&周波数フィルタリング」処理について図 8 を用いて説明する。図 8 に示す「学習データのブロック 81」は、図 4 に示した学習データのブロック 41 と同一であり、従来はこのデータから分離行列 83（図 4 の分離行列 43 と同一）のみを生成していた。それに対して本発明では、同一のデータ（学習データのブロック 81）から図 8 に示す分離行列 83 のみならず全死角空間フィルタ 84 も生成する。この全死角空間フィルタ 84 の生成方法については後述する。

【0100】

全死角空間フィルタ 84 は、学習データのブロック 81 の区間内に存在する全ての音源方向に死角を形成したフィルタ（ベクトルまたは行列）であり、これは突発音、すなわち、学習データのブロック 81 では鳴っていなかった方向の音のみを透過する働きをする。なぜなら、学習データのブロック 81 で鳴っていた音については、位置を変えずに鳴り続けている限り、全死角空間フィルタ 84 が形成する死角によって除去されるのに対し、突発音については、その方向に死角が形成されておらず、素通ししてしまうからである。

【0101】

一方、分離行列 83 も突発音を素通しする。その結果は出力チャンネルによって異なり、あるチャンネルではそれまで出力されていた音源に突発音が重畳され（図 7 の（b1）分離結果 1）、他のあるチャンネルでは突発音のみが出力される（図 7 の（b2）分離結

10

20

30

40

50

果 2、および (b 3) 分離結果 3)。

【 0 1 0 2 】

ここで、図 7 に示す (b 1) 分離結果 1 のような結果から全死角空間フィルタの結果を減算 (またはそれに類する操作) すると、突発音が相殺され、音源に対応した出力のみが残る。その処理シーケンスについて図 9 を参照して説明する。

【 0 1 0 3 】

図 9 には、

- (a) 観測信号
- (b) 全死角空間フィルタ適用信号
- (c 1) 処理結果 1
- (c 2) 処理結果 2
- (c 3) 処理結果 3

これらの各信号を示している。左から右に時間 (t) が経過し、ブロックの高さが音量を示している。

【 0 1 0 4 】

(a) 観測信号は、先に説明した図 7 の (a) 観測信号と同じ観測信号である。観測信号には、時間 $t_0 \sim t_5$ の間、継続して鳴り続けている継続音 9 1 と、時間 $t_1 \sim t_4$ の間のみ出力される突発音 9 2 が含まれている。

【 0 1 0 5 】

図 9 に示す (a) 観測信号に対して、全死角空間フィルタを適用すると、(b) 全死角空間フィルタ適用信号が得られる。すなわち、鳴りっぱなしの継続音 9 1 はほぼ除去されるのに対し、突発音 9 2 は鳴り始めが除去されずに残る。

(b) 全死角空間フィルタ適用信号では、時間 $t_0 \sim t_5$ の間、鳴りっぱなしの継続音 9 1 はほぼ除去される。一方、突発音 9 2 は鳴り始め (時間 $t_1 \sim$) が除去されずに残る。時間 $t_1 \sim t_2$ の区間 9 4 では、突発音 9 2 は全く除去されない。

【 0 1 0 6 】

全死角空間フィルタは、時間的に先行する観測信号に含まれる音源を除去する働きを持つが、時間 $t_1 \sim t_2$ の区間 9 4 の直前の観測信号には突発音 9 2 が含まれず、全死角空間フィルタによって除去されないためである。

【 0 1 0 7 】

この図 9 (b) に示す全死角空間フィルタ適用信号を、分離行列適用結果の一つである図 7 (b 1) 分離結果 1 から減算すると、突発音が除去され、鳴りっぱなしの継続音 9 1 のみが残った結果が得られる。これが、図 9 (c 1) 処理結果 1 の信号である。すなわち、図 9 (c 1) 処理結果 1 は、図 7 (b 1) 分離信号と、図 9 (b) 全死角空間フィルタ適用信号を利用した以下の演算の結果として得られる信号である。

処理結果 1 = (分離結果 1) - (全死角空間フィルタ適用信号)

【 0 1 0 8 】

なお、減算の際に突発音を完全に除去するためには、全死角空間フィルタ適用結果のスケールを、分離行列適用結果に含まれる突発音のスケールと合わせる必要がある。これを「全死角空間フィルタのリスケージング」と呼ぶ。なおリスケージング処理は一方の信号のスケール (信号の変動の範囲) をもう一方の信号に合わせる処理として行われる。この場合は、全死角空間フィルタ適用結果のスケールを、分離行列適用結果に含まれる突発音のスケールに近づける処理として行われる。スケールは I C A の出力チャンネルごとに合わせる必要があるため、リスケージング後の全死角空間フィルタ適用結果は、I C A のチャンネル数と同一である。(リスケージング前の全死角空間フィルタ適用結果のチャンネル数は 1 である。)

【 0 1 0 9 】

上記の「減算」は、通常の引き算 (複素数領域の引き算) でも良いが、一般化して、2チャンネル周波数フィルタリングと呼ばれる処理を用いても良い。

図 10 を参照して、2チャンネル周波数フィルタリングについて説明する。

10

20

30

40

50

【0110】

一般的に、2チャンネル周波数フィルタリングは2つの入力を持つ。

一方は観測信号102 [$X(\quad, t)$]、

もう一方は推定ノイズ101 [$N(\quad, t)$]

である。これらは同一の時刻と周波数の信号である。

この2つの信号から、ゲイン104 (観測信号に乗じる係数) [$G(\quad, t)$] をゲイン推定部103で計算し、そのゲインをゲイン適用部105において観測信号に乗じることで処理結果106を得る。処理結果 $U(\quad, t)$ は、下式によって示される。

$$U(\quad, t) = G(\quad, t) \times X(\quad, t)$$

【0111】

10

具体的には、ノイズが支配的な周波数ではゲインを小さく、ノイズが少ない周波数ではゲインを大きくすることで、ノイズの除去された信号を生成するのである。通常の減算も周波数フィルタリングの一種とみなすことができるが、他に、スペクトル減算 (spectral subtraction) や Minimum Mean Square Error (MMSE) ・ Wiener フィルタ ・ Joint MAP といった既知の方式も適用可能である。

【0112】

本発明の具体的な2チャンネル周波数フィルタリング処理について図11を参照して説明する。本発明の処理では、観測信号の入力として分離行列適用結果112、すなわち、 $Y'k(\quad, t)$

20

を入力する。

また、推定ノイズの入力として、突発音である全死角空間フィルタ適用結果 (リスケーリング後) 111、すなわち、

$$Z'k(\quad, t)$$

を入力する。

【0113】

ゲイン推定部113は、これら全死角空間フィルタ適用結果111と分離行列適用結果112を入力してゲイン114 [$Gk(\quad, t)$] を求める。ゲイン適用部115が、ゲイン114 [$Gk(\quad, t)$] を、分離行列適用結果112、すなわち、 $Y'k(\quad, t)$ に乗じることで、突発音が除去された結果である $Uk(\quad, t)$ を求める。処理結果 $Uk(\quad, t)$ は下式によって示される。

30

$$Uk(\quad, t) = Gk(\quad, t) \times Y'k(\quad, t)$$

【0114】

なお、周波数フィルタリングにおいてスペクトル減算のような非線形な方式を用いると、「背景技術」の欄で述べた「消し残り」も消すことが可能である。すなわち、「消し残り」は分離行列でも全死角空間フィルタでも消せないため、それぞれの結果同士を減算すると相殺される。そのため、追従遅れと消し残りとがトレードオフになるという課題も解消できる。

【0115】

(3) チャンネル別の判別

40

前述の「全死角空間フィルタ&周波数フィルタリング」処理を全てのチャンネルに対して適用すると、かえって弊害となることがある。それは、突発音が目的音である場合である。例えば図7において、(b2) 分離結果2には突発音のみが出力されており、このチャンネルに対して、図9に示した(b) 全死角空間フィルタ適用信号との減算を行なうと、突発音の鳴り始め (図7に示す時間 $t_1 \sim t_2$ の区間74) が除去され、無音が出力される。突発音が妨害音である場合はこれでも問題ないが、突発音が目的音である場合にはこれは望ましくない。

【0116】

そこで、以下に述べる基準を用いて、「全死角空間フィルタ&周波数フィルタリング」を適用すべきか否かをチャンネルごとに判別する。あるいは、周波数フィルタリングの度

50

合いをチャンネルごとに変更する。そうすることで、鳴りっぱなしの音（突発音が発生する前から鳴っている音）のみが出力されるチャンネルと、突発音のみが出力されるチャンネルとの両方が同時に実現できる。

【 0 1 1 7 】

あるチャンネルに「全死角空間フィルタ&周波数フィルタリング」を適用するか否か、すなわち、突発音を除去するのが望ましいか否かは、突発音が発生する直前にそのチャンネルから音源に対応した信号が出力されているかに依存する。音源に対応した信号が既に出力されている場合は周波数フィルタリングを行ない（あるいは減算する量を大きくし）、そのような信号が出力されていない場合には周波数数フィルタリングをスキップする（あるいは減算する量を小さくする）。

10

【 0 1 1 8 】

例えば、図7において、突発音72が発生する直前の時間 $t_0 \sim t_1$ の区間73に注目すると、(b1)分離結果1のチャンネルには音源の継続音71に対応した信号が出力されている。このチャンネルに対しては全死角空間フィルタおよび周波数フィルタリングを適用する。そうすることで、突発音が発生しても、その突発音は除去され、継続音71に由来する信号のみが出力され続ける。

これは、図9の(c1)処理結果1に対応する。

【 0 1 1 9 】

一方、他のチャンネルである図7に示す(b2)分離結果2、(b3)分離結果3の時間 $t_0 \sim t_1$ の区間73では、継続音71由来の成分は除去され、無音に近い信号が出力されている。このようなチャンネルに対しては、周波数フィルタリングは適用しない。すなわち、図9の(c2)処理結果2、(c3)処理結果3である。これらのチャンネルには頻繁リスケーリングのみ適用する。「頻繁リスケーリング」とは、前述したように、分離結果のリスケーリング（周波数間のバランスを原信号に近づける処理）をフレーム単位で行なう処理である。

20

この様な処理を行うことで、突発音が発生したときに突発音のみからなる信号が出力される。その場合でも、頻繁リスケーリングはフレーム毎に行なわれているため、従来法と異なり、突発音の鳴り始めの歪みは低減される。

【 0 1 2 0 】

I C Aの各出力（分離行列の適用結果）が音源に対応しているか否かは、分離行列に依存する。従ってその判別はフレーム毎に行なう必要はなく、分離行列が更新されたタイミングで行えばよい。判別のための具体的な尺度については後述する。

30

【 0 1 2 1 】

なお、上記の判別を「周波数フィルタリングを適用する/しない」の2値で行なうと、適用の有無が切り替わった時点で処理結果が大きく変化してしまう。そのような現象を防ぐには、I C Aの出力が音源に対応しているか否かを連続値で表現し、その値に応じて周波数フィルタリングを作用させる度合い（減算量）を連続的に変化させる処理を行えばよい。詳細は後述する。

【 0 1 2 2 】

[2 . 本発明の信号処理装置の具体的実施例について]

40

以下、本発明の信号処理装置の具体的な実施例について説明する。本発明の信号処理装置の一構成例を図12に示す。なお、図12に示す装置構成は、本出願人が先に出願した特許出願である特開2008-147920「リアルタイム音源分離装置および方法」をベースとした構成である。この特開2008-147920において示した構成に、全死角空間フィルタおよび周波数フィルタリングに関するモジュールである共分散行列計算部125、全死角空間フィルタ適用部127、周波数フィルタリング部128、全死角空間フィルタ保持部134、パワー比保持部135を追加した構成を有する。なお、図12に示す信号処理装置は、具体的には例えばPCによって実現可能である。すなわち、図12の示す信号処理装置内の各処理部の処理は、例えば、予め規定されたプログラムに従った処理を実行するCPUによって実行可能である。

50

【 0 1 2 3 】

図 1 2 の左側に示す分離処理部 1 2 3 は主に観測信号の分離を行なう。図 1 2 の右側に示す学習処理部 1 3 0 は主に分離行列の学習を行なう。具体的には、分離行列の生成、全死角空間フィルタの生成、パワー比の算出等を実行する。全死角空間フィルタは、前述したように学習データのブロック区間で検出された全ての音源方向に死角を形成したフィルタ（ベクトルまたは行列）であり、これは突発音、すなわち、学習データブロックでは鳴っていない方向の音のみを透過する働きをする。また、パワー比は、各チャンネルの音のパワー（音量）の比率情報である。

【 0 1 2 4 】

なお、分離処理部 1 2 3 における処理と、学習処理部 1 3 0 の処理は並列に行われる。分離処理部 1 2 3 の処理は表（foreground）の処理であり、学習処理部 1 3 0 の処理は裏（background）の処理であると言える。

10

【 0 1 2 5 】

システム全体で見ると、観測信号に対する音源分離処理をフレーム毎に行なって分離結果を生成する一方で、これらの分離処理に適用する分離行列や全死角空間フィルタを、適宜、最新のものに置き換えるという動作をする。分離行列や全死角空間フィルタの提供を学習処理部 1 3 0 が行い、分離処理部 1 2 3 が、学習処理部 1 3 0 から提供された分離行列や全死角空間フィルタを適用して音源分離処理を実行する。本発明で追加された 3 つの要素の内、全死角空間フィルタの生成自体は分離行列の学習と同様に学習処理部 1 3 0 内で裏の処理として行なわれるが、分離行列および全死角空間フィルタの頻繁リスケーリングや、それぞれを観測信号に適用することや、周波数フィルタリングなどは、分離処理部 1 2 3 内で表の処理として行なわれる。

20

【 0 1 2 6 】

以下では、それぞれの構成要素の処理について説明する。

複数のマイクロホン 1 2 1 で収録された音は A D 変換部 1 2 2 でデジタル信号に変換され、分離処理部 1 2 3 のフーリエ変換部 1 2 4 へ送られる。フーリエ変換部 1 2 4 では窓付きの短時間フーリエ変換（STFT）によって周波数領域のデータへ変換する（詳細は後述する）。その際、フレームと呼ばれる一定個数のデータが生成される。以降の処理は、このフレームを単位として行なわれる。フーリエ変換されたデータは、共分散行列計算部 1 2 5、分離行列適用部 1 2 6、全死角空間フィルタ適用部 1 2 7、スレッド制御部 1 3 1 にそれぞれ送られる。

30

【 0 1 2 7 】

以下では、先に分離処理部 1 2 3 における表の処理の信号の流れについて説明し、その後で学習処理部 1 3 0 の処理について説明する。

【 0 1 2 8 】

分離処理部 1 2 3 の共分散行列計算部 1 2 5 は、フーリエ変換部 1 2 4 の生成する観測信号のフーリエ変換データを入力し、観測信号の共分散行列をフレーム毎に計算する。計算の詳細は後述する。ここで求めた共分散行列は、分離行列適用部 1 2 6 および全死角空間フィルタ適用部 1 2 7 のそれぞれで、フレーム毎にリスケーリングを行なうために利用される。また、周波数フィルタリング部 1 2 8 において周波数フィルタリングを適用する度合いを決める尺度としても利用される。

40

【 0 1 2 9 】

分離行列適用部 1 2 6 では、学習処理部 1 3 0 において現在時刻より前に求められた分離行列、すなわち分離行列保持部 1 3 3 に保持された分離行列に対してリスケーリングを行なった後、1 フレーム分の観測信号とリスケーリング後の分離行列とを乗算して 1 フレーム分の分離行列適用結果を生成する。

【 0 1 3 0 】

全死角空間フィルタ適用部 1 2 7 では、学習処理部 1 3 0 において現在時刻より前に求められた全死角空間フィルタ、すなわち全死角空間フィルタ保持部 1 3 4 に保持された全死角空間フィルタに対してリスケーリングを行なった後、1 フレーム分の観測信号とリス

50

ケーリング後の全死角空間フィルタとを乗算して1フレーム分の全死角空間フィルタ適用結果を生成する。

【0131】

周波数フィルタリング部128は、観測信号に基づくフーリエ変換データに対する分離行列の適用結果を分離行列適用部126から受け取る一方で、観測信号に基づくフーリエ変換データに対する全死角空間フィルタの適用結果を全死角空間フィルタ適用部127から受け取り、両方の適用結果から、先に図11を参照して説明した2チャンネル周波数フィルタリングを行なう。その結果は、フーリエ逆変換部129へ送られる。

【0132】

フーリエ逆変換部129へ送られた分離結果は時間領域の信号に変換され、後段処理部136へ送られる。後段処理部136の実行する後段の処理とは、例えば音声認識や話者識別やサウンド出力などである。後段の処理によっては、周波数領域のデータをそのまま使用することも可能であり、その場合、フーリエ逆変換は省略可能である。

10

【0133】

次に、フーリエ変換部124は、観測信号に基づくフーリエ変換データを学習処理部130のスレッド制御部131にも提供する。

スレッド制御部131へ送られた観測信号は、スレッド演算処理部132の複数の学習スレッド132-1~Nへ送られる。それぞれの学習スレッドは、与えられた観測信号を一定量だけ蓄積した後、ICAのバッチ処理を用いて観測信号から分離行列を求める。この処理は、先に図5を参照して説明した処理と同様の処理である。さらに、スレッド制御部131は分離行列から全死角空間フィルタおよびパワー比も計算で求める。求めた分離行列、全死角空間フィルタ、パワー比は、分離行列保持部133、全死角空間フィルタ保持部134、パワー比保持部135に保持され、それぞれスレッド制御部131の制御の下、分離処理部123の分離行列適用部126、全死角空間フィルタ適用部127、周波数フィルタリング部128へ送られる。

20

【0134】

なお、全死角空間フィルタ適用部127および分離行列適用部126からスレッド制御部131への点線は、その時点で最新のリスケーリング済み全死角空間フィルタおよび分離行列を学習初期値に反映させることを表している。詳細は後段の[3.本発明の信号処理装置のその他の実施例(変形例)]において説明する。

30

【0135】

次に、図12に示す装置構成における学習処理部130のスレッド制御部131の詳細構成について、図13を参照して説明する。

【0136】

現フレーム番号保持カウンタ151は、観測信号が1フレーム分供給されるごとに値が1インクリメントされ、所定の値に達すると初期値に戻るようになっている。

学習初期値保持部152は、それぞれのスレッドにおいて学習処理を実行する場合の分離行列Wの初期値を保持するものである。分離行列Wの初期値は、基本的には最新の分離行列と同一であるが、異なる値を用いるものとしても良い。例えば、学習初期値にはリスケーリング(周波数ビン間のパワーを調整する処理。詳細は後述)を適用する前の分離行列を用い、分離行列にはスケーリング適用後のものを用いるようにする。

40

【0137】

蓄積開始予定タイミング指定情報保持部153は、蓄積を開始するタイミングを複数のスレッド間で一定間隔にするために用いられる情報である。使用方法は後述する。なお、蓄積開始予定タイミングは、相対時刻を用いて表されていてもよいし、相対時刻の代わりにフレーム番号で管理してもよいし、時間領域信号のサンプル番号で管理してもよい。これについては他の「時刻」や「タイミング」を管理するための情報についても同様である。

【0138】

観測信号蓄積タイミング情報保持部154は、分離部127で現在使用されている分離

50

行列Wが、どのタイミングで取得された観測信号を基に学習されたものであるかを示す情報、すなわち、最新の分離行列に対応した観測信号の相対時刻またはフレーム番号を保持するものである。観測信号の蓄積タイミング情報保持部154には、対応する観測信号の蓄積開始タイミングと蓄積終了タイミングとの両方を格納しても良いが、ブロック長、すなわち、観測信号の蓄積時間が一定ならば、いずれか一方だけを保存すれば十分である。

【0139】

また、スレッド制御部131は、それぞれのスレッドへリンクされているポイントを保持したポイント保持部155を有し、これを用いて、複数のスレッド132-1~Nの処理を制御している。

【0140】

次に、図14を参照してスレッド演算部132において実行する処理について説明する。スレッド132-1~Nのそれぞれは、観測信号バッファ161、分離結果バッファ162、学習演算部163、および、分離行列保持部164の各モジュールの機能を用いて、パッチ処理のICAを実行する。

【0141】

観測信号バッファ161は、スレッド制御部131から供給される観測信号を保持する。

分離結果バッファ162には、学習演算部163により演算された分離行列収束前の分離結果が保持される。

【0142】

学習演算部163は、観測信号バッファ161に蓄積されている観測信号を、分離行列保持部164に保持されている分離処理用の分離行列Wに基づいて分離して、分離結果バッファ162に蓄積するとともに、分離結果バッファ162に蓄積される分離結果を用いて、学習中の分離行列を更新する処理を実行する。

【0143】

スレッド演算部132(=学習スレッド)は状態遷移マシンであり、現在の状態は状態格納部165に格納される。スレッドの状態はカウンタ166のカウンタ値によって、スレッド制御部131により制御される。カウンタ166は、観測信号が1フレーム分供給されるのと同期して値が変化し、この値によって状態を切り替える。詳細は後述する。

【0144】

観測信号の開始・終了タイミング保持部167には、学習に使用されている観測信号の開始タイミングと終了タイミングを示す情報のうちの少なくともいずれか一方が保持されている。タイミングを示す情報は、上述したように、フレーム番号やサンプル番号であっても良いし、相対時刻情報であっても良い。ここでも、開始タイミングと終了タイミングとの両方を格納しても良いが、ブロック長、すなわち、観測信号の蓄積時間が一定ならば、いずれか一方だけを保存すれば十分である。

【0145】

学習終了フラグ168は、学習が終了したことをスレッド制御部131に通知するために用いられるフラグである。スレッドの起動時においては、学習終了フラグ168はOFF(フラグが立っていない)にセットされ、学習が終了した時点でONにセットされる。そして、スレッド制御部131が、学習が終了したことを認識した後、スレッド制御部131の制御により、学習終了フラグ168は、再び、OFFにセットされる。

【0146】

なお、状態格納部165、カウンタ166、観測信号の開始・終了タイミング保持部167のデータは、スレッド制御部131等の外部モジュールから値を書き換えることができるものとする。例えば、スレッド演算部132において学習のループが回っている間も、スレッド制御部131はカウンタ166の値を変更することができる。

【0147】

前処理用データ保持部169は、前処理が施された観測信号を元に戻す際に必要となるデータを保存しておく領域である。具体的には、例えば、前処理において観測信号の正規化(分散を1に、平均を0にそろえる)が実行される場合、前処理用データ保持部169

10

20

30

40

50

には、分散（または標準偏差やその逆数）や平均などの値が保持されるので、これを用いて正規化前の信号を復元することができる。また、例えば、前処理として無相関化（pre-whiteningとも称される）が実行される場合、前処理用データ保持部169には、無相関化で乗じた行列が保持される。

【0148】

全死角空間フィルタ保持部160は、観測信号バッファ161に含まれる全ての音源の方向に死角を形成するフィルタであり、学習終了時の分離行列から生成する。または、観測信号バッファのデータから生成する方法も存在する。生成方法については後述する。

【0149】

次に、学習スレッド132-1~Nの状態遷移について図15および図16を参照して説明する。実装としては、カウンタ166の値に基づいて学習スレッド自身が自分の状態を変化させるという仕様にしても良いが、スレッド制御部がカウンタの値や「学習終了フラグ」168の値に応じて状態遷移コマンドを発行し、学習スレッドはそのコマンドを受けて状態を変化させるという仕様でも良い。以下の実施例では、後者の仕様を採用している。

【0150】

図15は、先に図5を参照して説明したスレッド中の1つを示している。スレッドの各々は、観測信号の「蓄積中」状態において指定された時間、すなわち1ブロック長の観測信号がバッファに蓄えられる。指定された時間が経過した後、状態は学習中に遷移する。

【0151】

学習中の状態において、分離行列Wが収束するまで（または一定回数）学習処理ループが実行され、蓄積中の状態において蓄積された観測信号に対応した分離行列が求められる。分離行列Wが収束した後（または一定回数の学習処理ループが実行された後）状態は、待機中に遷移する。

【0152】

そして、待機中の状態において、指定された時間だけ、観測信号の蓄積や学習は実行されず、待機される。待機中の状態を維持する時間は、学習にかかった時間によって決まる。すなわち、図15に示されるように、予め、「蓄積中」状態と「学習中」状態と「待機中」状態との合計の時間幅であるスレッド長（thread_len）が定められ、基本的には、「学習中」状態が終了したときからスレッド長が終了するまでの間の時間が、「待機中」状態の時間（待機時間）とされる。待機時間が過ぎた後、状態は、観測信号の「蓄積中」状態へ戻る。

【0153】

これらの時間は、例えば、ミリ秒などの単位で管理してもよいが、短時間フーリエ変換で生成されるフレームを単位として計測するものとしても良い。以降の説明では、フレームを単位として計測する（たとえば、カウントアップをおこなう）ものとする。

【0154】

図16を参照してスレッドの状態遷移についてさらに説明する。システムの起動直後において各スレッドは「初期状態」181にあるが、その内の1つは「蓄積中」183へ、残りは「待機中」182へ状態を遷移させる（状態遷移コマンドを発行する）。先に説明した図5を例にとると、スレッド1が「蓄積中」へ遷移したスレッド、それ以外が「待機中」へ遷移したスレッドである。以下では、「蓄積中」へ遷移したスレッドについて先に説明する。

【0155】

観測信号の蓄積に要する時間をブロック長（block_len）と呼ぶ（図15参照）。また、蓄積・学習・待機の1周期に要する時間をスレッド長（thread_len）と呼ぶ。これらの時間は、ミリ秒などの単位で管理してもよいが、短時間フーリエ変換で生成されるフレームを単位としても良い。以降の説明では、フレームを単位としている。

【0156】

10

20

30

40

50

「蓄積中 学習中」および「待機中 蓄積中」の状態遷移については、カウンタの値に基づいて行なう。すなわち、「蓄積中」（図15の蓄積中状態171および図16の蓄積中状態183）から始まったスレッドの中では、観測信号が1フレーム分供給されるごとにカウンタを+1し、カウンタの値がブロック長（`block_len`）と同じ値になったら、「学習中」（図15の学習中状態172および図16の学習中状態184）へ状態を遷移させる。学習は、分離処理と並列にバックグラウンドで行なわれるが、その間も観測信号のフレームと連動してカウンタを+1していく。

【0157】

学習が終了したら、状態を「待機中」（図15の待機中状態173および図16の待機中状態182）へ遷移させる。待機中も学習中と同様に、観測信号のフレームと連動してカウンタを+1していく。そしてカウンタがスレッド長（`thread_len`）と同じ値になったら、状態を「蓄積中」（図15の蓄積中状態171および図16の蓄積中状態183）へ遷移させると共に、カウンタを0（または適切な初期値）に戻す。

【0158】

一方、「初期状態」181から「待機中」（図15の待機中状態173および図16の待機中状態182）へ遷移したスレッドについては、待機させたい時間に対応した値にカウンタをセットする。例えば、図5のスレッド2は、ブロックのシフト幅（`block_shift`）だけ待機してから「蓄積中」へ遷移する。同様に、スレッド3はブロックのシフト幅の2倍（`block_shift × 2`）だけ待機している。これらを実現するためには、

スレッド2のカウントを、

$(\text{スレッド長}) - (\text{ブロックシフト幅}) : (\text{thread_len}) - (\text{block_shift})$ にセットする。

また、スレッド3のカウントを、

$(\text{スレッド長}) - (2 \times \text{ブロックシフト幅}) : (\text{thread_len}) - (\text{block_shift} \times 2)$ にセットする。

このような設定にすれば、カウンタの値がスレッド長（`thread_len`）に達した後で「蓄積中」へ遷移し、それ以降はスレッド1と同様に「蓄積中 学習中 待機中」の周期を繰り返す。

【0159】

学習スレッドを何個用意する必要があるかは、スレッド長とブロックのシフト幅とで決める。スレッド長を `thread_len`、ブロックのシフト幅を `block_shift` とすると、必要な個数は、

$(\text{スレッド長}) / (\text{ブロックシフト幅})$ 、すなわち、

$\text{thread_len} / \text{block_shift}$

で求まる。

なお、端数は切り上げる。

【0160】

例えば図5では、

$[\text{スレッド長}(\text{thread_len})] = 1.5 \times [\text{ブロック長}(\text{block_len})]$ 、

$[\text{ブロックシフト幅}(\text{block_shift})] = 0.25 \times \text{ブロック長}(\text{block_len})$

に設定してあるため、必要なスレッド数は $1.5 / 0.25 = 6$ である。

【0161】

[3. 本発明の信号処理装置の実行する音源分離処理について]

(3-1. 全体シーケンス)

次に、本発明の信号処理装置におけるリアルタイム音源分離処理の全体シーケンスについて、図17に示すフローチャートを参照して説明する。図17に示すフローチャートは、分離処理部123の処理を中心として説明するフローチャートである。学習処理部13

10

20

30

40

50

0の「裏の処理(学習)」については、分離処理とは別の処理単位(別スレッド・別プロセス・別プロセッサなど)で動かすことが可能であるため、別のフローチャートを用いて説明する。また、両者でやりとりするコマンド等については、図20に示すシーケンス図で説明する。

【0162】

最初に、図17に示すフローチャートを参照して分離処理部123の処理について説明する。システムが起動したら、ステップS101において、各種の初期化を行なう。初期化の詳細は後述する。ステップS103の音入力から、ステップS108の分離結果の送信までの処理を、システムでの処理が終了(ステップS102でYes)するまで繰り返す。

10

【0163】

ステップS103の音入力は、オーディオデバイス(実施形態によってはネットワークやファイルなど)から一定数のサンプルを取り込み(この処理を「キャプチャー」と呼ぶ)、バッファに蓄える処理である。これをマイクロホンの個数分だけ行なう。以降では、キャプチャーされたデータを観測信号と呼ぶ。

【0164】

次に、ステップS104において、観測信号を一定長ごとに切り出して、短時間フーリエ変換(STFT)を行う。短時間フーリエ変換の詳細について図18を参照して説明する。

【0165】

例えば図1に示すような環境においてk番目のマイクによって収録された観測信号 x_k を図18(a)に示す。この観測信号 x_k から一定長を切り出した切り出しデータであるフレーム191~193にハニング窓やサイン窓等の窓関数を作用させる。なお、切り出した単位をフレームと呼ぶ。1フレーム分のデータに対して、離散フーリエ変換(有限区間のフーリエ変換のこと。略称DFT)または高速フーリエ変換(FFT)を施すことにより、周波数領域のデータであるスペクトル $X_k(t)$ を得る(t はフレーム番号)。

20

【0166】

切り出すフレームの間には、図に示すフレーム191~193のように重複があってもよく、そうすることで連続するフレームのスペクトル $X_k(t-1) \sim X_k(t+1)$ を滑らかに変化させることができる。また、スペクトルをフレーム番号に従って並べたものをスペクトログラムと呼ぶ。図18(b)がスペクトログラムの例である。

30

【0167】

本発明では入力チャンネルが複数(マイクの個数分)あるため、フーリエ変換もチャンネル数だけ行なう。以降では、全チャンネル・1フレーム分のフーリエ変換結果を $X(t)$ というベクトルで表わす(先に説明した式[3.11])。式[3.11]において、 n はチャンネル数(=マイク数)である。 M は周波数ピンの総数であり、短時間フーリエ変換のポイント数を J とすると、 $M = J / 2 + 1$ である。

【0168】

図17のフローに戻り、説明を続ける。ステップS104において、観測信号を一定長ごとに切り出して、短時間フーリエ変換(STFT)を行った後、ステップS105において、それぞれの学習スレッドに対する制御を行なう。詳細は後述する。

40

【0169】

次に、ステップS105で生成された観測信号 $X(t)$ に対して、ステップS106において分離を行なう。分離行列を W とすると(前記の式[3.10])、分離結果 $Y(t)$ (式[3.4])は

$$Y(t) = WX(t)$$

で求まる(式[3.12])。

【0170】

次に、ステップS107において、分離結果 $Y(t)$ に対して逆フーリエ変換(逆FT)をかけることで、時間領域の信号に戻す。それからステップS108において、分離結

50

果を後段の処理へ渡す。以上のステップ S 1 0 3 ~ S 1 0 8 を、終了まで繰り返す。

【 0 1 7 1 】

(3 - 2 . 初期化処理について (S 1 0 1))

図 1 7 に示すフローチャートにおけるステップ S 1 0 1 の初期化処理の詳細について、図 1 9 に示すフローチャートを参照して説明する。

【 0 1 7 2 】

ステップ S 1 5 1 において、図 1 2 および図 1 3 に示すスレッド制御部 1 3 1 は、初期化処理を実行する。具体的には、図 1 3 に示す各構成についてして以下の処理を行う。

現フレーム番号保持カウンタ 1 5 1 (図 1 3 参照) を初期化してその値を 0 とする。

学習初期値保持部 1 5 2 (図 1 3 参照) に適切な初期値を代入する。例えば、初期値は単位行列でも良いし、前回のシステム終了時の分離行列 W が保存されている場合は、前回のシステム終了時の分離行列 W、またはこの分離行列に適切な変換を作用させたものを使用しても良い。また、例えば、画像や先見知識等の情報により、音源の方向がある程度の精度で推定できるような場合には、音源方向に基づいて初期値を算出して、設定するものとしてもよい。

10

【 0 1 7 3 】

さらに、

蓄積開始予定タイミング指定情報保持部 1 5 3 には、

(必要スレッド数 - 1) × [ブロックシフト幅 (b l o c k _ s h i f t)]

上記式の算出値が設定される。

20

この値は、一番大きなスレッド番号を有するスレッドの蓄積が開始するタイミング (フレーム番号) である。

そして、観測信号の蓄積タイミング情報保持部 1 3 4 には、最新の分離行列に対応した観測信号を示すタイミング情報 (フレーム番号または相対時刻情報) が保持されるので、ここでは、初期化されて、0 が保持される。

【 0 1 7 4 】

なお、分離行列保持部 1 3 3 (図 1 2 参照) にも、初期化された場合の学習初期値保持部 1 5 2 と同様に、適切な初期値が保持される。分離行列保持部 1 3 3 に保持される初期値は、単位行列でも良いし、前回のシステム終了時の分離行列が保存されている場合は、前回のシステム終了時の分離行列 W、またはこの分離行列に適切な変換を作用させたものを使用しても良い。

30

【 0 1 7 5 】

さらに、全死角空間フィルタ保持部 1 3 4 (図 1 2 参照) にも初期値を代入する。この初期値は分離行列の初期値に依存する。分離行列として単位行列を用いる場合は、全死角空間フィルタには「無効」を表わす値を代入し、この値のときは後述の周波数フィルタリングが機能しないようにしておく。一方、分離行列の初期値として他に適切な値を用いる場合は、そこから全死角空間フィルタの値を計算する。

【 0 1 7 6 】

パワー比保持部 1 3 5 (図 1 2 参照) にも初期値を代入する。初期値として例えば 0 を代入しておく、最初の分離行列が学習で求まるまでの間 (例えば図 5 の区間 5 1) は周波数フィルタリングを機能させないようにすることができる。

40

【 0 1 7 7 】

ステップ S 1 5 2 において、スレッド制御部 1 3 1 は、スレッド演算部 1 3 2 において実行されるスレッドを必要な数 N だけ確保し、それらの状態を「初期化」状態とする。

【 0 1 7 8 】

ここで、必要なスレッドの数 N は、スレッド長 / ブロックシフト幅 (t h r e a d _ l e n / b l o c k _ s h i f t) の小数点以下を切り上げる (すなわち、 t h r e a d _ l e n / b l o c k _ s h i f t よりも大きく最も値の近い整数) ことにより求められる。

【 0 1 7 9 】

50

ステップ S 1 5 3 において、スレッド制御部 1 3 1 は、スレッドループを開始して、全てのスレッドの初期化が終了するまで、初期化未処理のスレッドを検出して、ステップ S 1 5 4 乃至ステップ S 1 5 9 の処理を実行する。ステップ S 1 5 2 で生成された個数だけループを回す。なお、スレッド番号は 1 から順に振ってき、ループ内では変数 s で表わされるものとする。(ループの代わりに、学習スレッドの個数だけ並列に処理しても構わない。以降の学習スレッドのループについても同様である。)

【0180】

ステップ S 1 5 4 において、スレッド制御部 1 3 1 は、スレッド番号は 1 であるか否かを判断する。1 番目のスレッドとそれ以外では初期設定が異なるため、ステップ S 1 5 4 で処理を分岐する。

10

【0181】

ステップ S 1 5 4 において、スレッド番号が 1 であると判断された場合、ステップ S 1 5 5 において、スレッド制御部 1 3 1 は、スレッド番号 1 のスレッド(例えば、スレッド 1 3 2 - 1)を制御して、そのカウンタ 1 6 6 (図 1 4 参照)を初期化(例えば 0 にセット)する。

【0182】

ステップ S 1 5 6 において、スレッド制御部 1 3 1 は、スレッド番号 1 のスレッド(例えば、スレッド 1 3 2 - 1)に、「蓄積中」状態に状態を遷移させるための状態遷移コマンドを発行して、処理は、後述するステップ S 1 5 9 に進む。状態遷移はスレッド制御部から学習スレッドに対して「指定された状態に遷移せよ」というコマンド(以降「状態遷移コマンド」)を発行することで行なう。(以降の説明における状態遷移は全て同様である。)

20

【0183】

ステップ S 1 5 4 において、スレッド番号は 1 ではないと判断された場合、ステップ S 1 5 7 において、スレッド制御部 1 3 1 は、対応するスレッド(スレッド 1 3 2 - 2 乃至スレッド 1 3 2 - N のうちのいずれか)のカウンタ 1 6 6 の値を、`thread__len - block__shift x` (スレッド番号 - 1) に設定する。

【0184】

ステップ S 1 5 8 において、スレッド制御部 1 3 1 は、「待機中」状態に状態を遷移させるための状態遷移コマンドを発行する。

30

【0185】

ステップ S 1 5 6、または、ステップ S 1 5 8 の処理の終了後、ステップ S 1 5 9 において、スレッド制御部 1 3 1 は、スレッド内のまだ初期化されていない情報、すなわち、状態格納部 1 6 5 (図 1 4 参照)に格納された状態を示す情報、および、カウンタ 1 6 6 のカウンタ値以外の情報を初期化する。具体的には、例えば、スレッド制御部 1 3 1 は、学習終了フラグ 1 6 8 (図 1 4 参照)を OFF にセットし、観測信号の開始・終了タイミング保持部 1 6 7、および、前処理用データ保持部 1 6 9 の値を初期化(例えば、0 にセット)する。

【0186】

スレッド演算部 1 3 2 に確保された全てのスレッド、すなわち、スレッド 1 3 2 - 1 乃至スレッド 1 3 2 - N が初期化された場合、ステップ S 1 6 0 において、スレッドループが終了され、初期化は終了する。

40

【0187】

このような処理により、スレッド制御部 1 3 1 は、スレッド演算部 1 3 2 に確保された複数のスレッドのすべてを初期化する。

【0188】

なお、図 1 9 のステップ S 1 5 4 ~ S 1 5 8 の処理は、図 2 0 のシーケンス図では最初の「初期化」処理と、その直後の状態遷移コマンド送信に該当する。なお、図 2 0 は、スレッド制御部 1 3 1 による複数の学習スレッド 1, 2 に対する制御シーケンスを示している。各スレッドは、待機、蓄積、学習という処理を繰り返し実行する。スレッド制御部は

50

観測信号を各スレッドに提供して、各スレッドが観測データを蓄積した後、学習処理を行い、分離行列を生成してスレッド制御部に提供する。

【0189】

(3-3. スレッド制御処理について(S105))

次に、図21に示すフローチャートを参照して、図17のフローチャート中のステップS105の処理、すなわち、スレッド制御部131によって実行されるスレッド制御処理について説明する。

【0190】

なお、このフローチャートは、スレッド制御部131から見たものであり、学習スレッド132-1~Nから見たものではないことに注意されたい。例えば「学習中処理」とは、学習スレッドの状態が「学習中」であるときにスレッド制御部131が行なう処理のことである。(学習スレッド自体の処理については、図29を参照されたい。)

10

【0191】

ステップS201~S206は学習スレッドについてのループであり、図21に示すフローのステップS152で生成された個数だけループを回す(並列処理でも構わない)。ステップS202において、学習スレッドの現在の状態を状態保持部165(図14参照)から読み込み、その値によってそれぞれ「待機中処理」・「蓄積中処理」・「学習中処理」を実行する。それぞれの処理の詳細は後述する。

【0192】

フローの各ステップについて説明する。ステップS201において、スレッド制御部131は、スレッドループを開始し、制御実行するスレッドのスレッド番号を示す変数sをs=1として、1つのスレッドの処理が終了すると変数sを1インクリメントして、s=Nとなるまで、ステップS202乃至ステップS207のスレッドループの処理を繰り返し実行する。

20

【0193】

ステップS202において、スレッド制御部131は、変数sで示されるスレッド番号のスレッドの状態格納部165に保持されている、そのスレッドの内部状態を示す情報を取得する。変数sで示されるスレッド番号のスレッドの状態として、「待機中」状態であると検出された場合、ステップS203において、スレッド制御部131は、図22のフローチャートを用いて後述する待機中状態における処理を実行し、処理は、後述するステップS206に進む。

30

【0194】

ステップS202において、変数sで示されるスレッド番号のスレッドの状態が「蓄積中」状態であると検出された場合、ステップS204において、スレッド制御部131は、図23のフローチャートを用いて後述する蓄積中状態における処理を実行し、処理は、後述するステップS206に進む。

【0195】

ステップS202において、変数sで示されるスレッド番号のスレッドの状態が「学習中」状態であると検出された場合、ステップS205において、スレッド制御部131は、図24のフローチャートを用いて後述する学習中状態における処理を実行する。

40

【0196】

ステップS203、ステップS204、または、ステップS205の処理の終了後、ステップS206において、スレッド制御部131は、変数sを1インクリメントする。そして、制御実行するスレッドのスレッド番号を示す変数sが、s=iとなったとき、スレッドループを終了する。

【0197】

ステップS207において、スレッド制御部131は、現フレーム番号保持カウンタ151(図13参照)に保持されているフレーム番号を1インクリメントし、スレッド制御処理を終了する。

【0198】

50

このような処理により、スレッド制御部 131 は、複数のスレッドの全てを、それらの状態に応じて制御することができる。

【0199】

なお、ここでは、立ち上げられたスレッドの数 i だけ、スレッドループが繰り返されるものとして説明したが、スレッドループを繰り返す代わりに、スレッドの個数 i の並列処理を実行するものとしてもよい。

【0200】

次に、図 22 のフローチャートを参照して、図 21 に示すフローチャートにおけるステップ S203 において実行される、待機中状態における処理について説明する。

【0201】

この待機中状態における処理は、図 21 を用いて説明したスレッド制御処理における変数 s に対応するスレッドの状態が「待機中」状態であるときに、スレッド制御部 131 において実行される処理である。

【0202】

ステップ S211 において、スレッド制御部 131 は、対応するスレッド 132 のカウンタ 166 (図 14 参照) を、1 インクリメントする。

【0203】

ステップ S212 において、スレッド制御部 131 は、対応するスレッド 132 のカウンタ 166 の値は、スレッド長 (thread_len) より小さいか否かを判断する。ステップ S212 において、カウンタ 166 の値は、スレッド長より小さいと判断された場合、待機中処理を終了し、図 21 のステップ S206 に進む。

【0204】

ステップ S212 において、カウンタ 166 の値がスレッド長より小さくないと判断された場合、ステップ S213 において、スレッド制御部 131 は、「蓄積中」状態に状態を遷移させるための状態遷移コマンドを、対応するスレッド 132 に発行する。

【0205】

すなわち、スレッド制御部 131 は、図 16 を用いて説明した状態遷移図において、「待機中」であるスレッドを、「蓄積中」に遷移させるための状態遷移コマンドを発行する。

【0206】

ステップ S214 において、スレッド制御部 131 は、対応するスレッド 132 のカウンタ 166 (図 14 参照) を初期化 (例えば、0 にセット) し、観測信号の開始・終了タイミング保持部 167 (図 14 参照) に、観測信号の蓄積開始タイミング情報、すなわち、スレッド制御部 131 の現フレーム番号保持カウンタ 151 (図 13 参照) に保持されている現在のフレーム番号、または、それと同等の相対時刻情報などを設定して、待機中処理を終了し、図 21 のステップ S206 に進む。

【0207】

このような処理により、スレッド制御部 131 は、「待機中」状態であるスレッドを制御し、そのカウンタ 166 の値に基づいて、「蓄積中」に状態を遷移させることができる。

【0208】

次に、図 23 のフローチャートを参照して、図 21 に示すフローチャートのステップ S204 において実行される蓄積中状態における処理について説明する。

【0209】

この蓄積中状態における処理は、図 21 を用いて説明したスレッド制御処理における変数 s に対応するスレッドの状態が「蓄積中」状態であるときに、スレッド制御部 131 において実行される処理である。

【0210】

ステップ S221 において、スレッド制御部 131 は、1 フレーム分の観測信号 $X(t)$ を、学習のために、対応するスレッド 132 に供給する。この処理は、図 20 に示すス

10

20

30

40

50

レッド制御部からそれぞれのスレッドへの観測信号の供給に対応する。

【0211】

ステップS222において、スレッド制御部131は、対応するスレッド132のカウンタ166を、1インクリメントする。

【0212】

ステップS223において、スレッド制御部131は、対応するスレッド132のカウンタ166の値がブロック長(block_len)より小さいか否か、換言すれば、対応するスレッドの観測信号バッファ161(図14参照)が満杯であるか否かを判断する。ステップS223において、カウンタ166の値がブロック長より小さい、換言すれば、対応するスレッドの観測信号バッファ161が満杯ではないと判断された場合、蓄積中処理を終了し、図21のステップS206に進む。

10

【0213】

ステップS223において、カウンタ166の値がブロック長より小さくない、換言すれば、対応するスレッドの観測信号バッファ161が満杯であると判断された場合、ステップS224において、スレッド制御部131は、「学習中」状態に状態を遷移させるための状態遷移コマンドを、対応するスレッド132に発行して、蓄積中処理を終了し、図21のステップS206に進む。

【0214】

すなわち、スレッド制御部131は、図16を用いて説明した状態遷移図において、「蓄積中」であるスレッドを、「学習中」に遷移させるための状態遷移コマンドを発行する。

20

【0215】

このような処理により、スレッド制御部131は、「蓄積中」状態であるスレッドに観測信号を供給してその蓄積を制御し、そのカウンタ166の値に基づいて、「蓄積中」から「学習中」に状態を遷移させることができる。

【0216】

次に、図24のフローチャートを参照して、図21に示すフローチャートのステップS205において実行される、学習中状態における処理について説明する。

【0217】

この学習中状態における処理は、図21を用いて説明したスレッド制御処理における変数sに対応するスレッドの状態が「学習中」状態であるときに、スレッド制御部131において実行される処理である。

30

【0218】

ステップS231において、スレッド制御部131は、対応するスレッド132の学習終了フラグ168(図14参照)がONであるか否かを判断する。ステップS231において、学習フラグがONであると判断された場合、処理は、後述するステップS237に進む。

【0219】

ステップS231において、学習フラグがONではないと判断された場合、すなわち、対応するスレッドにおいて学習処理が実行中である場合、ステップS232に進み、時刻の比較処理を行う。「時刻の比較」とは、学習スレッド132内に記録されている、観測信号の開始時刻167(図14参照)と、スレッド制御部131に保存されている、現在の分離行列に対応した蓄積開始時刻154(図13参照)とを比較する処理である。スレッド132内に記録されている観測信号の開始時刻167(図14参照)が、スレッド制御部131に保存されている現在の分離行列に対応した蓄積開始時刻154よりも前である場合は、以降の処理をスキップする。

40

【0220】

一方、スレッド132内に記録されている観測信号の開始時刻167(図14参照)が、スレッド制御部131に保存されている現在の分離行列に対応した蓄積開始時刻154よりも後または同じである場合は、ステップS233に進む。ステップS233において

50

、スレッド制御部 131 は、対応するスレッド 132 のカウンタ 166 を、1 インクリメントする。

【0221】

次にステップ S234 において、スレッド制御部 131 は、対応するスレッド 132 のカウンタ 166 の値がスレッド長 (thread_len) より小さいか否かを判断する。ステップ S234 において、カウンタ 166 の値がスレッド長より小さいと判断された場合、学習中処理を終了し、図 21 のステップ S206 に進む。

【0222】

ステップ S234 において、カウンタ 166 の値がスレッド長より小さくないと判断された場合、ステップ S235 において、スレッド制御部 131 は、カウンタ 166 の値から所定の値を減算し、学習中処理を終了し、図 21 のステップ S206 に進む。

10

【0223】

学習中にカウンタの値がスレッド長に達した場合とは、学習にかかる時間が長くなってしまい、「待機中」状態の時間が存在しなくなった場合である。その場合、学習はまだ継続しており、観測信号バッファ 161 は利用されているため、次の蓄積を開始することができない。そこで、スレッド制御部 131 は、学習が終了するまで、次の蓄積の開始、すなわち、「蓄積中」状態へ状態を遷移させるための状態遷移コマンドの発行を延期する。そのため、スレッド制御部 131 は、カウンタ 166 の値から所定の値を減算する。減算する値は、例えば、1 であっても良いが、それよりも大きな値でも良く、例えば、スレッド長の 10% などといった値であっても良い。

20

【0224】

なお、「蓄積中」状態への遷移の延期を行なうと、蓄積開始時刻がスレッド間で不均衡となり、最悪の場合、複数のスレッドでほぼ同一の区間の観測信号を蓄積してしまう可能性もある。そうすると、いくつかのスレッドが無意味になるだけでなく、例えば、CPU が実行する OS のマルチスレッドの実装によっては、1 つの CPU で複数の学習が同時に動くことになって、更に学習時間が増大し、間隔が一層不均等になってしまう可能性がある。

【0225】

そのような事態を防ぐためには、他のスレッドの待機時間を調整して蓄積開始タイミングが再び等間隔になるように調整すればよい。この処理は、ステップ S241 において実行される。この待機時間の調整処理の詳細については後述する。

30

【0226】

ステップ S231 において、学習終了フラグが ON であると判断された場合の処理について説明する。これは、学習スレッド内の学習ループが終了する度に一回実行される処理である。ステップ S231 において、学習終了フラグが ON であり、対応するスレッドにおいて学習処理が終了したと判断した場合、ステップ S237 において、スレッド制御部 131 は、対応するスレッド 132 の学習終了フラグ 168 を OFF にする。この処理は、この分岐が連続実行されるのを防ぐための操作である。

【0227】

その後、スレッド制御部 131 はスレッドの打ち切りフラグ 170 (図 14 参照) が ON であるか OFF であるかを確認し、ON である場合は、ステップ S239 において分離行列等の更新処理を行い、ステップ S241 において待機時間の設定処理を行なう。一方、スレッドの打ち切りフラグ 170 (図 14 参照) が OFF である場合は、ステップ S239 の分離行列等の更新処理は省略し、ステップ S241 において待機時間の設定処理を行なう。ステップ S239 の分離行列等の更新処理と、ステップ S241 の待機時間の設定処理の詳細については後述する。

40

【0228】

このような処理により、スレッド制御部 131 は、対応するスレッドの学習終了フラグ 168 を参照して、「学習中」状態のスレッドの学習が終了したか否かを判断し、学習が終了した場合、分離行列 W を更新し、待機時間を設定するとともに、「学習中」状態から

50

、「待機中」または「蓄積中」に状態を遷移させることができる。

【0229】

次に、図25のフローチャートを参照して、図24に示すフローチャートのステップS239において実行される分離行列等の更新処理について説明する。これは、学習で求めた分離行列と、全死角空間フィルタとパワー比を、他のモジュールに反映させる処理である。

【0230】

ステップS251において、スレッド制御部131は、スレッドの観測信号の開始・終了タイミング保持部167（図14参照）に保持されている観測信号の開始タイミングと、観測信号の蓄積タイミング情報保持部154（図13参照）に保持されている、現在の分離行列に対応した蓄積開始タイミングとを比較し、観測信号の開始タイミングが蓄積開始タイミングより早いかなかを判断する。

【0231】

すなわち、図5に示されるように、スレッド1の学習とスレッド2の学習とは、その一部で時間が重なっている。図5では、学習区間57のほうが、学習区間58より先に終了しているが、例えば、それぞれの学習にかかる時間によっては、学習区間57よりも学習区間58のほうが先に終了してしまう場合もあり得る。

【0232】

ここで、ステップS251の判断が実行されず、学習の終了が遅いものが最新の分離行列として扱われてしまった場合、スレッド2由来の分離行列W2が、より古いタイミングで取得された観測信号によって学習されて得られたスレッド1由来の分離行列W1に上書きされてしまう。そこで、新しいタイミングで取得された観測信号によって得られた分離行列が最新の分離行列として扱われるように、観測信号の開始・終了タイミング保持部167に保持されている観測信号の開始タイミングと、観測信号の蓄積タイミング情報保持部154に保持されている現在の分離行列に対応した蓄積開始タイミングとが比較される。

【0233】

ステップS251において、観測信号の開始タイミングが現在の分離行列に対応した蓄積開始タイミングよりも早いと判断された場合、換言すれば、このスレッドの学習の結果得られた分離行列Wは、現在、観測信号の蓄積タイミング情報保持部154に保持されている分離行列Wよりも早いタイミングで観測された信号に基づいて学習されていると判断された場合、このスレッドの学習の結果得られた分離行列Wは利用されないで、分離行列更新処理は終了する。

【0234】

ステップS251において、観測信号の開始タイミングが現在の分離行列に対応した蓄積開始タイミングよりも早くないと判断された場合、すなわち、このスレッドの学習の結果得られた分離行列Wは、現在、観測信号の蓄積タイミング情報保持部154に保持されている分離行列Wよりも遅いタイミングで観測された信号に基づいて学習されていると判断された場合、ステップS252において、スレッド制御部131は、対応するスレッドの学習によって得られた分離行列Wを取得し、分離行列保持部133（図12参照）に供給して設定する。同様に同様に最新の全死角空間フィルタを全死角空間フィルタ保持部134に設定し、分離行列適用結果のパワー比をパワー比保持部135に設定する。

【0235】

ステップS253において、スレッド制御部131は、学習初期値保持部152に保持されるそれぞれのスレッドにおける学習の初期値を設定する。

【0236】

具体的には、スレッド制御部131は、学習初期値として、対応するスレッドの学習によって得られた分離行列Wを設定するものとしてもよいし、対応するスレッドの学習によって得られた分離行列Wを用いて演算される、分離行列Wとは異なる値を設定するものとしても良い。例えば、分離行列保持部133（図12参照）にはリスケーリング適用後の

10

20

30

40

50

値を代入し、学習初期値保持部 152 にはリスケージング適用前の値を代入するようにする処理としてもよい。それ以外の例については、変形例で説明する。なお、学習初期値の計算は、「分離行列の更新処理」において行なう他に、学習の前処理として行なうことも可能である。詳細は変形例を参照されたい。

【0237】

ステップ S254 において、スレッド制御部 131 は、対応するスレッドの観測信号の開始・終了タイミング保持部 167 (図 14 参照) に保持されているタイミング情報を、観測信号の蓄積タイミング情報保持部 154 (図 13 参照) に設定する。これらの処理によって、分離行列等更新処理を終了する。

【0238】

ステップ S254 の処理により、現在使用中、すなわち、分離行列保持部 133 に保持されている分離行列 W が、どの時間区間の観測信号から学習されたものであるかが示される。

【0239】

次に、図 26 のフローチャートを参照して、図 24 に示すフローチャートのステップ S241 において実行される待機時間の設定処理について説明する。

【0240】

ステップ S281 において、スレッド制御部 131 は、残りの待機時間を計算する。

【0241】

具体的には、スレッド制御部 131 は、残り待機時間 (フレーム個数) を $rest$ 、蓄積開始予定タイミング指定情報保持部 153 (図 13 参照) に保持されている蓄積開始予定タイミング (フレーム番号、または、対応する相対時刻) を Ct 、現フレーム番号保持カウンタ 151 に保持されている現フレーム番号を Ft 、ブロックのシフト幅を $block_shift$ として、残り待機時間 $rest$ を、

$$rest = Ct + block_shift - Ft$$

として算出する。すなわち、 $Ct + block_shift$ が、次々回蓄積開始予定時刻を意味するため、そこから Ft を引くことで、「次々回蓄積開始予定時刻までの残り時間」が求まるのである。

【0242】

ステップ S282 において、スレッド制御部 131 は、残りの待機時間 $rest$ の計算結果は正の値であるか否かを判断する。ステップ S282 において、残りの待機時間 $rest$ の計算結果は正の値ではない、すなわち、ゼロまたは負の値であると判断された場合、処理は、後述するステップ S286 に進む。

【0243】

ステップ S282 において、残りの待機時間 $rest$ の計算結果は正の値であると判断された場合、ステップ S283 において、スレッド制御部 131 は、「待機中」状態に状態を遷移させるための状態遷移コマンドを、対応するスレッドに発行する。

【0244】

ステップ S284 において、スレッド制御部 131 は、対応するスレッドのカウンタ 166 (図 14 参照) の値を、 $thread_len - rest$ に設定する。そうすることで、カウンタの値が、 $thread_len$ に達するまでの間は、「待機中」状態が継続される。

【0245】

ステップ S285 において、スレッド制御部 131 は、蓄積開始予定タイミング指定情報保持部 153 (図 13 参照) に保持されている値 Ct に、 $block_shift$ の値を加算する、すなわち、蓄積開始予定タイミング指定情報保持部 153 に次回の蓄積開始タイミングである、 $Ct + block_shift$ の値を設定し、残り待機時間の計算処理を終了する。

【0246】

ステップ S282 において、残りの待機時間 $rest$ の計算結果は正の値ではない、す

10

20

30

40

50

なわち、ゼロまたは負の値であると判断された場合、予定された蓄積開始タイミングを過ぎているにもかかわらず蓄積が始まっていないことを意味するので、直ちに蓄積を開始する必要がある。そこで、ステップ S 2 8 6 において、スレッド制御部 1 3 1 は、「蓄積中」状態に状態を遷移させるための状態遷移コマンドを、対応するスレッドに発行する。

【 0 2 4 7 】

ステップ S 2 8 7 において、スレッド制御部 1 3 1 は、カウンタの値を初期化（例えば 0 をセット）する。

【 0 2 4 8 】

ステップ S 2 8 8 において、スレッド制御部 1 3 1 は、蓄積開始予定タイミング指定情報保持部 1 5 3 に次回の蓄積開始タイミング、すなわち、現フレーム番号である F t を設定し、残り待機時間の計算処理を終了する。

10

【 0 2 4 9 】

このような処理により、それぞれのスレッドにおける「学習中」状態にかかる時間に応じて、「待機中」状態とする時間を設定することができる。

【 0 2 5 0 】

（ 3 - 4 . 分離処理について（ S 1 0 6 ） ）

次に、図 1 7 に示すフローチャートのステップ S 1 0 6 の処理である分離処理の詳細について、図 2 7 に示すフローチャートを参照して説明する。

【 0 2 5 1 】

図 2 7 のフローに示すステップ S 3 0 1 ~ S 3 1 0 はループ処理であり、ループ内の処理を周波数ビンごとに行なう。なお、ループ処理の代わりに、並列処理として実行してもよい。

20

【 0 2 5 2 】

ステップ S 3 0 2 において、後述のリスケーリングで必要な共分散行列をあらかじめ計算しておく。これは、図 1 2 に示す共分散行列計算部 1 2 5 に対応する処理である。リスケーリング処理は分離行列に対する処理であるステップ S 3 0 3 と全死角空間フィルタに対する処理であるステップ S 3 0 5 があるが、いずれも観測信号の共分散行列から計算可能である。そのため、ステップ S 3 0 2 では観測信号の共分散行列を以下に示す式 [4 . 3] を用いて計算する。

【 0 2 5 3 】

30

【数 4】

$$\mathbf{R}(\omega) = \left\langle \mathbf{X}(\omega, t) \mathbf{Y}(\omega, t)^H \right\rangle_t \text{diag} \left(\left\langle \mathbf{Y}(\omega, t) \mathbf{Y}(\omega, t)^H \right\rangle_t \right)^{-1} \quad \dots\dots[4.1]$$

$$= \Sigma_{XX}(\omega) \mathbf{W}(\omega)^H \text{diag} \left(\mathbf{W}(\omega) \Sigma_{XX}(\omega) \mathbf{W}(\omega)^H \right)^{-1} \quad \dots\dots[4.2]$$

$$\Sigma_{XX}(\omega) = \left\langle \mathbf{X}(\omega, t) \mathbf{X}(\omega, t)^H \right\rangle_t \quad \dots\dots[4.3]$$

$$\Sigma_{XX}(\omega) \leftarrow \Sigma_{XX}(\omega) - \frac{1}{L} \mathbf{X}(\omega, t-L) \mathbf{X}(\omega, t-L)^H + \frac{1}{L} \mathbf{X}(\omega, t) \mathbf{X}(\omega, t)^H \quad \dots\dots[4.4]$$

$$\mathbf{R}(\omega) = \begin{bmatrix} R_{11}(\omega) & \cdots & R_{1n}(\omega) \\ \vdots & \ddots & \vdots \\ R_{n1}(\omega) & \cdots & R_{nn}(\omega) \end{bmatrix} \quad \dots\dots[4.5]$$

$$\mathbf{W}'(\omega) = \text{diag} (R_{11}(\omega), \dots, R_{11}(\omega)) \mathbf{W}(\omega) \quad \dots\dots[4.6]$$

$$\mathbf{Y}'(\omega, t) = \mathbf{W}'(\omega) \mathbf{X}(\omega, t) \quad \dots\dots[4.7]$$

$$\mu_k(\omega) = \min \left\{ \frac{|X_k(\omega, t)|}{\left| \sum_{k=1}^n Y'_k(\omega, t) \right|}, 1 \right\} \quad \dots\dots[4.8]$$

$$\mathbf{Y}'(\omega, t) \leftarrow \text{diag} (\mu_1(\omega), \dots, \mu_n(\omega)) \mathbf{Y}'(\omega, t) \quad \dots\dots[4.9]$$

10

20

【0254】

ただし、平均操作 $\langle \cdot \rangle_t$ を行なう区間は、図 8 に示す現在時刻を含むブロック 87 であり、現在のフレームを含んでいる。そのため、現在のフレーム番号を t 、現在時刻を含むブロック区間 87 の長さ（フレーム数）を L とすると、式 [4.4] の操作を毎フレーム行なうことで、観測信号の共分散行列は更新される。

【0255】

次に、ステップ S303 において分離行列のリスキューリングを行なう。このリスキューリングは先の [1. 本発明の構成と処理の概要について] の欄において説明した「頻繁リスキューリング」のことであり、このリスキューリング処理の目的は突発音が出力される際の歪みを低減することである。リスキューリングの基本的な考え方は、分離結果を特定のマイクロホンへ射影することであり、「特定のマイクロホンへ射影する」とは、例えば図 1 において、1 番目のマイクロホンで観測される信号を、それぞれの音源に由来する成分にスケールを保ったまま分解することである。

30

【0256】

リスキューリング処理は、観測信号からの切り出しデータ単位であるフレーム中、現在の観測信号を含むフレームを適用して行われる。前述したように分離処理部 123 の共分散行列計算部 125 は、フーリエ変換部 124 の生成する観測信号のフーリエ変換データを入力し、観測信号の共分散行列をフレーム毎に計算する。ここで求めた共分散行列が分離行列適用部 126 および全死角空間フィルタ適用部 127 の各々においてフレーム毎のリスキューリング処理を行なうために利用される。

40

【0257】

リスキューリング処理のために、上記の式 [4.1] および式 [4.2] によってリスキューリング用の行列 $\mathbf{R}(\omega)$ をいったん求め、その次に、リスキューリング用行列 $\mathbf{R}(\omega)$ の 1 行目（1（小文字のエル）は射影先のマイクロホン番号）を要素とする対角行列を求める（式 [4.6] の右辺の第 1 項）。その対角行列をリスキューリング前の分離行列 $\mathbf{W}(\omega)$ に乗じることで、リスキューリング済みの分離行列 $\mathbf{W}'(\omega)$ を得る（式 [4.6]）。

【0258】

50

ステップ S 3 0 4 では、リスケーリング後の分離行列 W' () を観測信号 X (, t) に乗じることで (式 [4 . 7])、分離行列適用結果 Y' (, t) を得る。

$$Y' (, t) = W' () \times X (, t)$$

として、分離行列適用結果 Y' (, t) を得る。

この処理は、観測信号 X (, t) に対するリスケーリング後の分離行列 W' () を適用した線形フィルタリング処理に相当する。

【 0 2 5 9 】

ステップ S 3 0 3、S 3 0 4 の処理は、図 8 に示す処理例において、

現在時刻の観測信号 X (t) 8 2 の取得と、

分離行列 8 3 の適用処理、

これらの処理に対応する。

図 8 に示す分離行列 8 3 は、学習データブロック 8 1 から求められた分離行列である。先に説明したように、従来のリスケーリングは、学習データブロック 8 1 の学習データを用いて行なわれていた。それに対して、本発明の処理では、ステップ S 3 0 3 において、現在時刻を末端とする一定長のブロック、すなわち、図 8 に示す現在時刻を含むブロック 8 7 を設定し、この現在時刻を含むブロック 8 7 の区間の観測信号を用いてリスケーリングを行なう。この処理により突発音に対しても早い段階でスケールを合わせる (= 歪みを低減する) ことができる。

【 0 2 6 0 】

さらに必要に応じて、式 [4 . 8] および式 [4 . 9] による再調整を行なう。これは、リスケーリング後の分離行列適用結果 Y' (, t) の要素の総和が、射影先マイクロホンに対応した観測信号 X_1 (, t) の絶対値を超えていないかチェックし、超えている場合に Y' (, t) の絶対値を小さくする処理である。式 [4 . 1] で求めたリスケーリング係数は、大きな音が鳴り止んだ直後でも、その音が区間内 (図 8 の 8 7) に残っている限り大きな値となる傾向があり、その結果、現在の観測信号が無音に近い音 (背景音) であっても大きなスケールによって背景音が強調されてしまう場合がある。しかし、式 [4 . 8] および式 [4 . 9] による再調整を行なうことで、スケールが大きくなるのを防ぐことができる。

【 0 2 6 1 】

次に、ステップ S 3 0 5 において、全死角空間フィルタのリスケーリングを行なう。このリスケーリングの目的は、全死角空間フィルタの適用結果に含まれる突発音と、分離行列適用結果に含まれる突発音との間でスケールを合わせ、後述の周波数フィルタリングで突発音が相殺されるようにするためである。

【 0 2 6 2 】

図 1 2 に示す分離処理部 1 2 3 は、前述したフレーム単位の頻繁リスケーリングを行う。すなわち、観測信号からの切り出しデータ単位であるフレーム中、現在の観測信号を含むフレームを適用したスケール調整としてのリスケーリング処理を実行した分離行列と、同様にリスケーリング処理を実行した全死角空間フィルタとを、ステップ S 3 0 3 とステップ S 3 0 5 において生成する。ステップ S 3 0 4 では、リスケーリング処理後の分離行列を適用し、ステップ S 3 0 6 では、リスケーリング後の全死角空間フィルタを適用した処理を行う。

【 0 2 6 3 】

例えば図 8 に示す構成において、全死角空間フィルタ 8 4 は、学習データのブロック 8 1 の区間において鳴っている全ての音源の方向に死角を形成したフィルタ (ベクトルまたは行列) であり、これは突発音、すなわち、学習データのブロック 8 1 では鳴っていなかった方向の音のみを透過する働きをする。なぜなら、学習データのブロック 8 1 で鳴っていた音については、位置を変えずに鳴り続けている限り、フィルタが形成する死角によって除去されるのに対し、突発音については、その方向に死角が形成されておらず、素通ししてしまうからである。

【 0 2 6 4 】

10

20

30

40

50

ステップ S 3 0 5 において、全死角空間フィルタのリスケーリング処理では、以下に示す式 [7 . 1] および式 [7 . 2] によってリスケーリング用の行列 Q () を求める。(式 [7 . 1] の Y' (, t) は、式 [4 . 9] の再調整を適用する前の値である。)

【 0 2 6 5 】

【 数 5 】

$$Q(\omega) = \langle Y'(\omega, t) \overline{Z(\omega, t)} \rangle_t / \langle Z(\omega, t) \overline{Z(\omega, t)} \rangle_t \quad \dots\dots[7.1]$$

$$= W'(\omega) \Sigma_{XX}(\omega) B(\omega)^H / B(\omega) \Sigma_{XX}(\omega) B(\omega)^H \quad \dots\dots[7.2]$$

$$B'(\omega) = Q(\omega) B(\omega) \quad \dots\dots[7.3]$$

$$Z'(\omega, t) = \text{diag}(\mu_1(\omega), \dots, \mu_n(\omega)) B'(\omega) X(\omega, t) \quad \dots\dots[7.4]$$

10

【 0 2 6 6 】

ただし、式 [7 . 2] の B () はリスケーリング前の全死角空間フィルタであり、 n 個の入力から 1 個の出力を生成するフィルタである (B () の計算方法は後述する)。また、式 [7 . 1] の Z (, t) は、リスケーリング前の全死角空間フィルタ適用結果であり、以下に示す式 [5 . 5] で計算される。

20

【 0 2 6 7 】

【 数 6 】

$$B(\omega) = e_l - \sum_{k=1}^n W_k(\omega) \quad \dots\dots[5.1]$$

$$e_l = [0, \dots, 0, 1, 0, \dots, 0] \quad (l\text{番目の要素のみ } 1) \quad \dots\dots[5.2]$$

$$X_l(\omega, t) - \sum_{k=1}^n W_k(\omega) X(\omega, t) \approx 0 \quad \dots\dots[5.3]$$

$$X_l(\omega, t) - \sum_{k=1}^n W_k(\omega) X(\omega, t) = B(\omega) X(\omega, t) \quad \dots\dots[5.4]$$

$$Z(\omega, t) = B(\omega) X(\omega, t) \quad \dots\dots[5.5]$$

30

【 0 2 6 8 】

なお、 Z (, t) はベクトルではなく、スカラーである。また、 Q () は n 個の要素からなる行ベクトル (横長のベクトル) である。 Q () に B () を乗じることで (式 [7 . 3])、リスケーリング済みの全死角空間フィルタ B' () を得る。 B' () は n 行 n 列の行列である。

40

【 0 2 6 9 】

ステップ S 3 0 6 では、リスケーリング済みの全死角空間フィルタ B' () に観測信号を乗じることで (式 [7 . 4])、リスケーリング済みの全死角空間フィルタ適用結果 Z' (, t) を得る。ただし、式 [7 . 4] の μ_k () は、式 [4 . 8] で求めた値であり、 Y' (, t) を再調整した場合に Z' (, t) も再調整するためである。

【 0 2 7 0 】

全死角空間フィルタ適用結果 Z' (, t) は n 個の要素からなる列ベクトル (縦長のベクトル) であり、 k 番目の要素は $Y' k$ (, t) にスケールを合わせた全死角空間フィルタ適用結果である。

【 0 2 7 1 】

50

ステップ S 3 0 5、S 3 0 6 は、図 8 の処理例を参照して説明すると、
 現在時刻の観測信号 $X(t)$ 8 2 の取得、
 リスケーリング済みの全死角空間フィルタ $B'(\quad)$ 8 4 の生成、
 リスケーリング済みの全死角空間フィルタ $B'(\quad)$ に観測信号を乗じることで (式 [7 . 4])、リスケーリング済みの全死角空間フィルタ適用結果 $Z'(\quad, t)$ を得る処理に対応する。

【 0 2 7 2 】

次のステップ S 3 0 7 ~ S 3 1 0 はループであり、ステップ S 3 0 8 の周波数フィルタリングをチャンネルごとに行なうことを意味する。なお、ループの代わりに並列処理として実行してもよい。

10

【 0 2 7 3 】

ステップ S 3 0 8 の周波数フィルタリングは、リスケーリング済み分離行列適用結果 $Y'k(\quad, t)$ (ベクトル $Y'(\quad, t)$ の k 番目の要素) に対して、周波数ごとに異なる係数を乗じる処理であるが、本発明ではリスケーリング済み分離行列適用結果 $Y'k(\quad, t)$ からリスケーリング済み全死角空間フィルタ適用結果 (突発音にほぼ等しい) を除去するために用いる。

【 0 2 7 4 】

周波数フィルタリングの例として、以下の 3 点について説明する。

- (1) 複素数上の引き算
- (2) スペクトル減算
- (3) ウィナーフィルタ

20

【 0 2 7 5 】

まず、(1) 複素数上の引き算による周波数フィルタリングについて説明する。この処理は、分離行列を適用して生成した分離信号から全死角空間フィルタ適用信号を減算する処理により、分離信号に含まれる全死角空間フィルタ適用信号に対応する信号成分を除去するフィルタリング処理である。

以下に示す式 [8 . 1] は、複素数上の減算を表わす式である。

【 0 2 7 6 】

【数 7】

$$U_k(\omega, t) = Y_k'(\omega, t) - \alpha_k Z_k'(\omega, t) \quad \dots\dots[8.1]$$

$$U_k(\omega, t) = G_k(\omega, t) \frac{Y_k'(\omega, t)}{|Y_k'(\omega, t)|} \quad \dots\dots[8.2]$$

$$G_k(\omega, t) = \max\{|Y_k'(\omega, t)| - |\alpha_k Z_k'(\omega, t)|, |\beta Y_k'(\omega, t)|\} \quad \dots\dots[8.3]$$

$$G_k(\omega, t)^2 = \max\{|Y_k'(\omega, t)|^2 - |\alpha_k Z_k'(\omega, t)|^2, |\beta Y_k'(\omega, t)|^2\} \quad \dots\dots[8.4]$$

$$\alpha_k = \alpha f(r_k) \quad \dots\dots[8.5]$$

$$r_k = \frac{V_k}{\max_k[V_k]} \quad \dots\dots[8.6]$$

$$r_k = \frac{V_k}{V_{\max}} \quad \dots\dots[8.7]$$

$$V_k = \sum_{\omega=1}^M \langle |W_k(\omega) X(\omega, t)|^2 \rangle_t \quad \dots\dots[8.8]$$

$$= \sum_{\omega=1}^M W_k(\omega) \langle X(\omega, t) X(\omega, t)^H \rangle_t W_k(\omega)^H \quad \dots\dots[8.9]$$

$$f(r_k) = \begin{cases} f_{\min} & (r_k < r_{\min}) \\ \frac{r_k - r_{\min}}{r_{\max} - r_{\min}} & (r_{\min} \leq r_k \leq r_{\max}) \\ 1 & (r_{\max} < r_k) \end{cases} \quad \dots\dots[8.10]$$

$$r_k = \frac{V_k}{\langle V_k \rangle_k} \quad \dots\dots[8.11]$$

$$\text{SNR}_k^{[\text{post}]}(\omega, t) = \frac{|Y_k'(\omega, t)|^2}{|\alpha_k Z_k'(\omega, t)|^2} \quad \dots\dots[8.12]$$

$$\text{SNR}_k^{[\text{prior}]}(\omega, t) \leftarrow \kappa \frac{|U_k(\omega, t-1)|^2}{|\alpha_k Z_k'(\omega, t)|^2} + (1 - \kappa) \max\{\text{SNR}_k^{[\text{post}]}(\omega, t) - 1, 0\} \quad \dots\dots[8.13]$$

$$G_k(\omega, t) = \frac{\text{SNR}_k^{[\text{prior}]}(\omega, t)}{\text{SNR}_k^{[\text{prior}]}(\omega, t) + 1} |Y_k'(\omega, t)| \quad \dots\dots[8.14]$$

【0277】

上記式 [8 . 1] において、係数 α_k は 0 以上の実数であり、この係数によって、先に、[1 . 本発明の構成と処理の概要について] の欄において説明した「(3) チャンネル別の判別」を実現する。

すなわち、突発音の性質によって異なる対処をするため、ICA の各出力チャンネルが音源に対応した信号を出力しているかどうか判別し、その結果に応じて以下のどちらかの処理をする。

i) 音源に対応していると判別された場合は、「頻繁リスケーリング」と「全死角空間フィルタ & 周波数フィルタリング」との両方を適用する。

10

20

30

40

50

その結果、突発音はそのチャンネルからは除去される。

i i) 音源に対応していないと判別された場合は、「頻繁リスケーリング」のみを適用する。その結果、突発音はそのチャンネルから出力される。

これを「チャンネル別の判別」と呼ぶ。

【0278】

このように、突発音発生前に各チャンネルが音源に対応した信号を出力しているかに応じて、突発音の削減量を調整する。

【0279】

各チャンネルの出力が音源に対してはどうかを判別する方法は各種存在するが、以下の説明で用いているのは、分離行列適用結果のパワーを用いる方法である。すなわち、音源に対応しているチャンネルは比較的パワーが大きく、音源に対応していないチャンネルは比較的パワーが小さいという性質を利用する。

10

【0280】

上記式 [8 . 1] に示される係数 r_k は、式 [8 . 5] によって計算する。この式において、 r_k はチャンネル k のパワー比、 r_{max} は r_k の最大値である。パワー比は、全体の観測された音の全パワー、あるいは最大の音に対する各チャンネル (k) のパワーの比率である。パワー比 r_k はチャンネル k のパワー (音量) を V_k とし、式 [8 . 6] または式 [8 . 7] を適用して算出する。これらの式の詳細については後述する。

【0281】

$f()$ は 0 以上 1 以下を返値とする関数であり、式 [8 . 10] および図 28 に示すグラフで表される関数である。この関数の目的は、パワー比 r_k によって引き算の有無が急激に切り替わるのを防ぐためである。(逆に、 $r_{min} = r_{max}$ とすると、パワー比が閾値を跨いだ時点で、引き算の有無が急激に変化する。)

20

式 [8 . 10] の f_{min} は、0 または小さな正の値である。 f_{min} を 0 以外の値に設定する効果については、後で説明する。

【0282】

ステップ S 308 の周波数フィルタリングは、図 12 に示す周波数フィルタリング部 128 として実行される。周波数フィルタリング部 128 は、分離信号からの全死角空間フィルタ適用信号対応成分の除去レベルを分離信号チャンネルに応じて変更する処理を行う。具体的には、分離信号チャンネルのパワー比に応じて除去レベルを変更する。

30

【0283】

パワー比 r_k は式 [8 . 6] ~ [8 . 9] で計算するが、式 [8 . 8] および [8 . 9] に含まれる平均操作 $\langle \cdot \rangle_t$ は分離行列の学習で使用された観測信号と同じ区間で行なう。すなわち、図 8 に示す処理例における現在時刻を含むブロック 87 ではなく、学習データのブロック 81 の区間である。これらの式では最新フレームのデータは用いていないため、 r_k および r_{max} の計算は毎フレーム行なう必要はなく、分離行列の学習が終了したタイミングで行えばよい。そこで、 r_k の具体的な計算方法については、図 31 に示す分離行列学習のフローチャートのステップ S 420 の後処理の詳細について説明した図 32 のフローを参照して後段で説明する。

【0284】

40

複素数上の引き算 (式 [8 . 1]) でも突発音は除去可能となる。しかし、線形フィルタリングの一種であるため、「従来技術の問題点」で述べた「追従遅れと消し残りとのトレードオフ」という課題は解消できない。一方、以下で述べる非線形な周波数フィルタリングを用いると、そのトレードオフも解消することが可能となる。

【0285】

上記した式 [8 . 2] は、周波数フィルタリングの一般式である。すなわち、リスケーリング済みの分離行列適用結果 $Y'k(\omega, t)$ を絶対値で正規化した項、

$$Y'k(\omega, t) / |Y'k(\omega, t)|$$

に対して、ゲイン $G_k(\omega, t)$ を乗じる。周波数フィルタリングの手法によってゲインの計算方法は様々であるが、以下で説明するスペクトル減算法 (spectral s

50

u b t r a c t i o n) では、スペクトル振幅の差分から求める。

【 0 2 8 6 】

(2) スペクトル減算による周波数フィルタリングについて説明する。

スペクトル減算による周波数フィルタリング処理は、全死角空間フィルタ適用信号を雑音成分としたスペクトル減算による周波数フィルタリング処理により、分離行列を適用して生成した分離信号に含まれる全死角空間フィルタ適用信号に対応する信号成分を除去するフィルタリング処理である。

【 0 2 8 7 】

スペクトル減算法の式は、上記の式 [8 . 3] および式 [8 . 4] に示される通りである。式 [8 . 3] は振幅そのものの減算であり、Magnitude Spectral Subtraction と呼ばれる。式 [8 . 4] は振幅の2乗の減算であり、Power Spectral Subtraction と呼ばれる。両式において、 $\max \{ A , B \}$ は2つの引数の内で大きい方を返り値とする操作を表わす。 α_k は、一般的には over - subtraction factor と呼ばれる項であるが、本発明では式 [8 . 5] の演算をすることで、「音源に対応した信号が出力されているか」に応じて減算量を調整する働きもしている。 β は flooring factor と呼ばれ、0に近い小さな値(例えば0.01)である。 $\max \{ \}$ の第2項によって、減算後のゲインが0や負の値になるのを防ぐ。

【 0 2 8 8 】

α_k の計算は、複素数上の引き算の場合と同様に、式 [8 . 5] ~ [8 . 1 0] に従って行なう。なお、式 [8 . 1 0] において、 f_{\min} を0の代わりに正の小さな値とすると、

$$r_k < r_{\min}$$

の場合にも周波数フィルタリングが小さく作用するため、「消し残り」をある程度除去することができる。

【 0 2 8 9 】

(3) ウィナーフィルタによる周波数フィルタリングについて説明する。

ウィナーフィルタとは、目的音と妨害音とのパワーの比である事前SNR (priori SNR) に基づいて係数 $G_k (\cdot , t)$ を計算する方式である。事前SNRが既知であれば、ウィナーフィルタによって求まる係数は、妨害音の除去性能が二乗誤差最小の意味で最適であることが知られている。ウィナーフィルタの詳細については、例えば以下を参照されたい。

特許出願 2 0 0 7 - 5 3 3 3 3 1 [H 1 8 . 8 . 3 1]

特再 W O 0 7 / 0 2 6 8 2 7 [H 2 1 . 3 . 1 2]

【 発明の名称 】 マイクロホンアレイ用ポストフィルタ

【 出願人 】 北陸先端科学技術大学院大学，トヨタ自動車(株)

【 発明者 】 赤木 正人，李 軍鋒，上地 正昭，佐々木 和也

【 0 2 9 0 】

ウィナーフィルタに基づいて係数を計算するためには事前SNRの値が必要だが、一般にその値は未知である。そこで、事前SNRの代わりに、観測信号と妨害音とのパワー比である事後SNR (posteriori SNR) と、直前のフレームでの処理結果を目的音とみなした1フレーム分の事前SNRとから、事前SNRをフレームごとに推定する方法が提案されており、それを Decision Directed (DD) 法と呼ぶ。DD法を用いて突発音を除去する方法について、式 [8 . 1 2] ~ 式 [8 . 1 4] を用いて説明する。(これらの式において、上付きの [post] と [prior] は、それぞれ「事後」「事前」を区別するためのものである。)

【 0 2 9 1 】

式 [8 . 1 2] は、1フレーム分の事後SNRを求める式である。この式において α_k は式 [8 . 5] などから求める。ただし、ウィナーフィルタにおいては、over - subtractionを行なう必要はないため、 $\alpha_k = 1$ でよい。あるいは、 $\alpha_k < 1$ とするこ

10

20

30

40

50

とで、突発音の除去効果を小さくすることもできる。次に、式 [8 . 1 3] を用いて、事前 S N R の推定値を求める。この式の α は忘却係数であり、1 未満かつ 1 に近い値を用いる。

事前 S N R の推定値から、式 [8 . 1 4] を用いて周波数フィルタリングの係数 $G_k (\omega, t)$ を計算する。

【 0 2 9 2 】

周波数フィルタリングの方法として、上記では、

- (1) 複素数上の引き算
- (2) スペクトル減算
- (3) ウィナーフィルタ

10

これらについて説明したが、これらの方法以外にも、以下の方法も適用可能である。

(4) Minimum Mean Square Error (MMSE) Short Time Spectral Amplitude (STSA)、または MMSE Log Spectral Amplitude (LSA)

【 0 2 9 3 】

なお、これらの詳細については以下を参照されたい。

* 『独立成分分析を用いた雑音推定に基づく MMSE STSA』

岡本 亮維, 高橋 祐, 猿渡 洋, 鹿野 清宏,

日本音響学会講演論文集, 2 - 9 - 6, pp. 663 - - 666, March 2009.

20

"MMSE STSA with Noise Estimation Based on Independent Component Analysis"

Ryo OKAMOTO, Yu TAKAHASHI, Hiroshi SARUWATARI and Kiyohiro SHIKANO

* 登録特許 4 1 7 2 5 3 0 号公報 雑音抑圧の方法及び装置並びにコンピュータプログラム

* "Diffuse noise suppression by crystal-array-based post-filter design"

Nobutaka ITO, Nobutaka ONO, and Shigeki SAGAYAMA

30

【 0 2 9 4 】

図 2 7 に示すフローチャートに従った分離処理により、従来法の分離結果よりも精度の高い分離結果である $U_1 (\omega, t) \sim U_n (\omega, t)$ が生成されるようになる。

【 0 2 9 5 】

[4 . スレッド演算部における学習スレッドの処理について]

図 1 2 に示すスレッド制御部 1 3 1 と、各学習スレッド 1 3 2 - 1 ~ N を適用したスレッド演算部 1 3 2 の処理は並列で動作しており、学習スレッドはスレッド制御部とは別のフローに基づいて動いている。以降では、スレッド演算部における学習スレッドの処理について、図 2 9 に示すフローチャートを用いて説明する。

【 0 2 9 6 】

40

スレッド演算部 1 3 2 は、起動後、ステップ S 3 9 1 において、初期設定される。起動のタイミングは、図 1 7 の全体フローのステップ S 1 0 1 の初期化処理の期間であり、図 1 9 に示すフローのステップ S 1 5 2 の学習スレッドの確保処理のタイミングとなる。

【 0 2 9 7 】

スレッド演算部 1 3 2 において学習スレッドは、起動後、ステップ S 3 9 1 において初期設定され、その後、イベントが発生するまで待機する (処理をブロックする) 。 (この「待機」は、学習スレッドの状態の 1 つである「待機中」とは別である。) イベントは、以下のどれかのアクションが行なわれたときに発生する。

- ・状態遷移コマンドが発行された。
- ・フレームデータが転送された。

50

・終了コマンドが発行された。

どのイベントが発生したかによって、以降の処理を分岐する（ステップ S 3 9 2）。すなわち、スレッド制御部 1 3 1 から入力されたイベントによって、続く処理が分岐される。

【 0 2 9 8 】

ステップ S 3 9 3 において、状態遷移コマンドが入力されたと判断された場合、ステップ S 3 9 4 において対応するコマンド処理が実行される。

【 0 2 9 9 】

ステップ S 3 9 3 において、フレームデータの転送イベントの入力を受けたと判断された場合、ステップ S 3 9 5 において、スレッド 1 3 2 は、フレームデータを取得する。次に、ステップ S 3 9 6 において、スレッド 1 3 2 は、取得したフレームデータを、観測信号バッファ 1 6 1（図 1 4 参照）に蓄積し、ステップ S 3 9 2 に戻り、次イベントを待機する。

10

【 0 3 0 0 】

観測信号バッファ 1 6 1（図 1 4 参照）は、配列またはスタックの構造をしており、観測信号はカウンタと同じ番号の個所に格納されるものとする。

【 0 3 0 1 】

ステップ S 3 9 3 において、終了コマンドが入力されたと判断された場合、ステップ S 3 9 7 において、スレッド 1 3 2 は、例えば、メモリの開放などの適切な終了前処理を実行し、処理が終了される。

20

【 0 3 0 2 】

このような処理により、スレッド制御部 1 3 1 の制御に基づいて、それぞれのスレッドにおいて処理が実行される。

【 0 3 0 3 】

次に、図 3 0 のフローチャートを参照して、図 2 9 に示すフローチャートのステップ S 3 9 4 において実行される、コマンド処理について説明する。

【 0 3 0 4 】

ステップ S 4 0 1 において、スレッド 1 3 2 は、供給された状態遷移コマンドに応じて、それ以降の処理を分岐する。なお、以降では「 の状態へ遷移する」というコマンドを「状態遷移コマンド『 』」と表現する。

30

【 0 3 0 5 】

ステップ S 4 0 1 において、供給された状態遷移コマンドが、「待機中」状態への遷移を指令する「状態遷移コマンド『待機中』」である場合、ステップ S 4 0 2 において、スレッド 1 3 2 は、状態格納部 1 6 5（図 1 4 参照）に、状態が「待機中」であることを示す情報を格納する、すなわち、状態を「待機中」に遷移して、コマンド処理を終了する。

【 0 3 0 6 】

ステップ S 4 0 1 において、供給された状態遷移コマンドが「蓄積中」状態への遷移を指令する「状態遷移コマンド『蓄積中』」である場合、ステップ S 4 0 3 において、スレッド 1 3 2 は、状態格納部 1 6 5 に、状態が「蓄積中」であることを示す情報を格納する、すなわち、状態を「蓄積中」に遷移して、コマンド処理を終了する。

40

【 0 3 0 7 】

ステップ S 4 0 1 において、供給された状態遷移コマンドが「学習中」状態への遷移を指令する「状態遷移コマンド『学習中』」である場合、ステップ S 4 0 4 において、スレッド 1 3 2 は、状態格納部 1 6 5 に、状態が「学習中」であることを示す情報を格納する、すなわち、状態を「学習中」に遷移する。

【 0 3 0 8 】

さらに、ステップ S 4 0 5 において分離行列の学習処理を実行する。この処理の詳細については後述する。

【 0 3 0 9 】

ステップ S 4 0 6 において、スレッド 1 3 2 は、学習が終了したことをスレッド制御部

50

131へ通知するために、学習終了フラグ168をONにして処理を終了する。フラグを立てることで、学習が終了した直後であることをスレッド制御部131へ通知するのである。

【0310】

このような処理により、スレッド制御部131から供給された状態遷移コマンドに基づいて、それぞれのスレッドの状態が遷移される。

【0311】

次に、図31のフローチャートを参照して、図30に示すフローチャートのステップS405において実行される処理の一例である分離行列の学習処理例について説明する。これは、バッチによって分離行列を求める処理であり、バッチ処理であればどのアルゴリズムも適用可能である。ただし、パーミュテーション(permutation)を起こしにくい方式を用いる必要がある。以下では、本出願人の先の出願である特開2006-238409『音声信号分離装置・雑音除去装置および方法』に開示した構成を適用した例について説明する。

【0312】

ステップS411において、スレッド132の学習演算部163(図14参照)は、観測信号バッファ161に蓄積された観測信号に対して、必要に応じて、前処理を実行する。

【0313】

具体的には、学習演算部163は、学習のループに入る前に、必要に応じて、観測信号バッファ161に蓄積された観測信号に対して、正規化(normalization)や無相関化(uncorrelationまたはpre-whitening)などの処理を行なう。例えば、正規化を行なう場合、学習演算部163は、ブロック内のフレームについて観測信号の標準偏差を求め、標準偏差の逆数からなる対角行列をSとして、以下に示す式[9.1]により、 $X' = SX$ を計算する。ただしXは、ブロック内の全フレーム分の観測信号からなる行列であり、図8の学習データのブロック81で表わされる区間である。

【0314】

【数8】

$$X'(\omega, t) = S(\omega)X(\omega, t) \quad \dots\dots[9.1]$$

$$Y(\omega, t) = W(\omega)X'(\omega, t) \quad \dots\dots[9.2]$$

$$W(\omega) \leftarrow R(\omega)W(\omega) \quad \dots\dots[9.3]$$

$$W(\omega) \leftarrow R(\omega)W(\omega)S(\omega) \quad \dots\dots[9.4]$$

$$\lambda_k(\omega) = \frac{\langle X_l(\omega, t) \overline{Y_k(\omega, t)} \rangle_t}{\langle Y_k(\omega, t) \overline{Y_k(\omega, t)} \rangle_t} \quad \dots\dots[9.5]$$

$$R(\omega) = \text{diag}(\lambda_1(\omega), \dots, \lambda_n(\omega)) \quad \dots\dots[9.6]$$

$$\Sigma_{XX}(\omega) = \langle X(\omega, t)X(\omega, t)^H \rangle_t \quad \dots\dots[9.7]$$

$$\Sigma_{XX}(\omega) = [p_1, \dots, p_n] \text{diag}(\lambda_1, \dots, \lambda_n) [p_1, \dots, p_n]^H \quad \dots\dots[9.8]$$

$$P(\omega) = \text{diag}(\lambda_1, \dots, \lambda_n)^{-1/2} [p_1, \dots, p_n]^H \quad \dots\dots[9.9]$$

$$X'(\omega, t) = P(\omega)X(\omega, t) \quad \dots\dots[9.10]$$

$$\langle X'(\omega, t)X'(\omega, t)^H \rangle_t = I \quad \dots\dots[9.11]$$

【0315】

一方、無相関化は共分散行列が単位行列となるような変換である。無相関化の方法は何通りがあるが、ここでは共分散行列の固有値 (Eigenvalue) および固有ベクトル (Eigenvector) を用いる方法を説明する。

【 0316 】

蓄積された観測信号 (例えば図 8 の学習データのブロック 81) から、式 [9.7] を用いて周波数ビンごとに共分散行列 $X X^T$ () を計算する。次にこの行列に対して固有値展開を適用すると、 $X X^T$ () は固有値 $\lambda_1 \sim \lambda_n$ および固有ベクトル $p_1 \sim p_n$ を用いて式 [9.8] のように分解できる。ただし、固有ベクトルは単位ベクトルかつ互いに直交しているものとする。固有値および固有ベクトルから式 [9.9] のような行列 P () を生成すると、 P () は無相関化の行列となっている。

10

【 0317 】

すなわち、 P () を観測信号 X (, t) に乗じたものを X' (, t) とすると (式 [9.10])、 X' (, t) の共分散行列は式 [9.11] の関係を満たしている。

【 0318 】

このような無相関化を前処理として行なうことにより、学習において収束までのループ回数を少なくすることができる。また、本発明では、固有ベクトルから全死角空間フィルタを生成することも可能になる。(詳細は後述する)

【 0319 】

以下の式に現われる観測信号 X は、前処理を行なった観測信号 X' も表わし得るものとする。

20

次に、ステップ S 412 において、学習演算部 163 は、分離行列の初期値として、スレッド制御部 131 から、スレッド制御部 131 の学習初期値保持部 152 に保持されている学習初期値 W を取得する。

【 0320 】

ステップ S 413 ~ S 419 の処理は、学習のループであり、これらの処理を W が収束するか打ち切りフラグが ON になるまで繰り返す。打ち切りフラグは、先に説明した図 24 の学習中処理のフローのステップ S 236 において ON に設定されるフラグである。後から開始した学習がそれより前に開始した学習よりも早く終了した場合に ON になるものである。ステップ S 413 において、打ち切りフラグが ON であると判断した場合は、処理を終了する。

30

【 0321 】

ステップ S 413 において、打ち切りフラグが OFF であると判断した場合は、ステップ S 414 に進む。ステップ S 414 において、学習演算部 163 は、分離行列 W の値が収束したか否かを判断する。分離行列 W の値が収束したか否かは、例えば、行列のノルムを用いて判定する。分離行列 W のノルム (全要素の 2 乗和) である $\|W\|_F$ と、 W のノルムである $\|W\|_2$ とをそれぞれ計算し、両者の比である $\|W\|_F / \|W\|_2$ が一定の値 (例えば $1 / 1000$) よりも小さければ、 W が収束したと判定する。または単純に、ループが一定数 (例えば 50 回) 回ったかどうかで判定しても構わない。

【 0322 】

ステップ S 414 において、分離行列 W の値が収束したと判断された場合、処理は、後述するステップ S 420 に進み、後処理を実行して処理を終了する。すなわち、分離行列 W が収束するまで、学習処理ループが実行される。

40

【 0323 】

ステップ S 414 において、分離行列 W の値が収束していないと判断された場合 (またはループ回数が所定の値に達していない場合)、ステップ S 415 ~ S 419 の学習のループの中に進む。学習は、全ての周波数ビンにおいて、先に説明した式 [3.1] ~ 式 [3.3] を繰り返す処理として行なわれる。すなわち、分離行列 W を求めるために、式 [3.1] から式 [3.3] までを分離行列 W が収束するまで (または一定回数) 繰り返し実行する。この繰り返し実行が「学習」である。なお、分離結果 Y (t) は式 [3.4]

50

で表わされる。

【0324】

ステップS416が、式[3.1]に対応する。
 ステップS417が、式[3.2]に対応する。
 ステップS418が、式[3.3]に対応している。

【0325】

式[3.1]～式[3.3]は周波数ビンごとの式であるため、ステップS415とステップS419で周波数ビンについてのループを回すことで、全周波数ビンのWを求めている。

【0326】

なお、ICAのアルゴリズムとしては、式[3.2]以外も適用可能である。たとえば、前処理として無相関化を行なった場合は、正規直交制約(orthonormal constraint)に基づく勾配法である以下に示す式[3.13]～[3.15]を用いても良い。なお、式[3.13]の $X'(\quad , t)$ は、無相関化後の観測信号である。

【0327】

【数9】

$$Y(\omega, t) = W(\omega)X'(\omega, t) \quad \dots\dots[3.13]$$

$$D(\omega) = \left\langle \varphi_{\omega}(Y(t))Y(\omega, t)^H \right\rangle_t \quad \dots\dots[3.14]$$

$$\Delta W(\omega) = \{D(\omega) - D(\omega)^H\}W(\omega) \quad \dots\dots[3.15]$$

【0328】

これらのループ処理の終了後に、ステップS413に戻り打ち切りフラグの判定、ステップS414における分離行列の収束判定を行う。打ち切りフラグがONであれば処理を終了する。ステップS414で分離行列の収束が確認された場合(あるいは規定ループ数に達した場合は、ステップS420に進む。

【0329】

ステップS420の後処理の詳細について、図32に示すフローチャートを参照して説明する。

ステップS420の後処理においては、以下の処理を実行する。

- (1) 分離行列を、正規化前の観測信号に対応させる。
- (2) 周波数ビンの間のバランスを調整する(リスケージング)。

【0330】

まず、(1)分離行列を、正規化前の観測信号に対応させる処理について説明する。

前処理として正規化が行なわれた場合、上述した処理(図31のステップS415～S419)により求められる分離行列Wは、正規化前の観測信号Xを分離するためのものではなく、正規化後の観測信号 X' を分離するためのものである。すなわち、WにXを直接乗じても、それは分離された信号ではない。そこで、上述した処理により求められた分離行列W()を補正して、正規化前の観測信号 $X(\quad , t)$ を分離するためのものへと変換する。

【0331】

具体的には、正規化の際に作用させた行列を $S(\quad)$ とすると、 $W(\quad)$ を正規化前の観測信号に対応させるには、

$$W(\quad) = W(\quad)S(\quad)$$

という補正を行なえばよい(式[9.1])。

前処理として無相関化を行なった場合も同様に、

10

20

30

40

50

$W(\quad) W(\quad) P(\quad)$

という補正を行なう。($P(\quad)$ は無相関化の行列)

【0332】

次に(2)周波数ピンの間のバランスを調整する(リスケーリング)処理について説明する。

ICAのアルゴリズムによっては、分離結果 Y の周波数ピン間のバランス(スケール)が、予想される原信号のものと異なっている場合がある(特開2006-238409『音声信号分離装置・雑音除去装置および方法』はそのような例である)。そのような場合、後処理で周波数ピンのスケールを補正する必要がある。スケールの補正のために、式[9.5]と式[9.6]から補正用の行列を計算する。式[9.5]の l (小文字のエル)は、射影先のマイクロホンの番号である。補正用の行列が求まったら、式[9.3]によって分離行列 $W(\quad)$ を補正する。

【0333】

なお、

(1)分離行列を、正規化前の観測信号に対応させる。

(2)周波数ピンの間のバランスを調整する(リスケーリング)。

これらをまとめて、式[9.4]を適用して一気に補正しても構わない。こうしてリスケーリングされた分離行列は、図12に示す分離行列保持部133に格納され、必要に応じて分離処理部123の実行する分離処理(表の処理)において参照される。

【0334】

次に、ステップS453の全死角空間フィルタの生成処理に進む。全死角空間フィルタの生成方法には以下の2通りが可能である。

(1)分離行列から生成

(2)観測信号共分散行列の固有ベクトルから生成

【0335】

まず、「(1)分離行列から全死角空間フィルタを生成する方法」について説明する。

ステップS452においてリスケーリングされた分離行列を $W(\quad)$ 、その行ベクトルを $W_1(\quad) \sim W_n(\quad)$ とすると、全死角空間フィルタ $B(\quad)$ は、先に示した式[5.1]で計算できる。ただし、 l (小文字のエル)は射影先のマイクロホン番号を表わす。 e_l は n 次元の行ベクトルであり、 l 番目の要素のみが1、それ以外を0とした行列である。

【0336】

式[5.1]に従って求めた全死角空間フィルタ $B(\quad)$ を観測信号 $X(\quad)$ に乘じると、その結果 $Z(\quad, t)$ は全死角空間フィルタ適用結果となる(式[5.4])。

こうして計算した全死角空間フィルタ $B(\quad)$ が全死角空間フィルタとして機能する理由は、式[5.3]で説明できる。

【0337】

この式[5.3]において、

$W_k(\quad) X(\quad, t)$

は、分離行列適用結果の k チャンネル目である。

分離行列は、先に図32を参照して説明した分離処理フローのステップS452の分離行列のリスケーリング処理においてリスケーリングされているため、分離行列適用結果を全チャンネルで総和すると、射影先マイクロホンの観測信号である $X_1(\quad, t)$ とほぼ等しくなる。

【0338】

従って、式[5.3]の左辺は0に近い値になるはずである。また、式[5.3]の左辺は、式[5.1]の全死角空間フィルタ $B(\quad)$ を用いて式[5.4]の右辺のように変形できる。すなわち $B(\quad)$ は、観測信号 $X(\quad, t)$ から0に近い信号を生成するフィルタ、すなわち全死角空間フィルタと見なすことができるのである。

【 0 3 3 9 】

なお、分離行列の収束が不十分の場合、そのような分離行列から生成される全死角空間フィルタは、学習データの区間に含まれる音源もある程度は透過させる性質をもつ。例えば、先に図7を参照して説明した従来法では、時間 $t_2 \sim t_3$ の区間 75 においては分離行列が収束しておらず、そのために突発音もある程度は出力されるが、その区間の分離行列から生成された全死角空間フィルタも、やはり突発音をある程度は透過する。それが図9における時間 $t_2 \sim t_3$ の区間 95 である。しかし、図7に示す時間 $t_2 \sim t_3$ の区間 75 と、図9に示す時間 $t_2 \sim t_3$ の区間 95 において突発音が同様に透過しているため、周波数フィルタリングによって相殺される。すなわち、図9に示す (c1) 処理結果1において時間 $t_2 \sim t_3$ の区間 95 に対応する区間でも突発音は消える。

10

【 0 3 4 0 】

次に、「(2) 観測信号共分散行列の固有ベクトルから全死角空間フィルタを生成する方法」について説明する。

【 0 3 4 1 】

図31を参照して説明した分離行列の学習処理におけるステップ S411の「前処理」として無相関化を用いた場合、観測信号の共分散行列に対する固有値分解はすでに完了している。すなわち、以下に示す式 [6.1] (式 [9.8] と同一) のように、観測信号共分散行列 $X X^H$ は、固有値 $\lambda_1 \sim \lambda_n$ と固有ベクトル $p_1 \sim p_n$ を用いて表される。

20

【 0 3 4 2 】

【 数 1 0 】

$$\Sigma_{XX}(\omega) = [p_1, \dots, p_n] \text{diag}(\lambda_1, \dots, \lambda_n) [p_1, \dots, p_n]^H \quad \dots\dots[6.1]$$

λ_k : 固有値 (降順) p_n : 固有ベクトル

$$B(\omega) = p_n^H \quad \dots\dots[6.2]$$

【 0 3 4 3 】

ここで、固有値は全て 0 以上かつ降順に並んでいるとする。すなわち、

30

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$$

を満たすとする。この場合、最小の固有値 λ_n に対応した固有ベクトル p_n は、全死角空間フィルタの性質を持っている。従って、式 [6.2] のように全死角空間フィルタ $B(\omega)$ を設定すれば、以降は「(1) 分離行列から生成」の場合と同様に全死角空間フィルタ $B(\omega)$ を使用することができる。

【 0 3 4 4 】

この方法は、ICA 以外でも、時間周波数領域で観測信号にベクトルや行列を乗じて音源分離を行なう方式と組み合わせることで、前述した「消し残り」を軽減することができる。

【 0 3 4 5 】

40

こうして生成された全死角空間フィルタは、図12に示す全死角空間フィルタ保持部 134 に格納され、必要に応じて、分離処理部 123 の実行する分離処理 (表の処理) に際して参照される。

以上で、ステップ S453 の全死角空間フィルタの生成処理についての説明を終了する。

【 0 3 4 6 】

次に、ステップ S454 の「パワー比を計算」する処理について説明する。パワー比は、例えば、先に図27を参照して説明した分離処理におけるステップ S308 の「周波数フィルタリング」処理において参照されるが、パワー比の計算で用いる観測信号は学習データの区間 (例えば図8に示す学習データのブロック 81) と同一であるため、パワー比

50

の計算自体は学習終了時に一度行えば、次回に分離行列が更新されるまでの間はその値が有効である。

【0347】

パワー比を求める前に、まず前記した式[8.8]または式[8.9]を用いて、チャンネルごとにパワー（区間内の要素の2乗和）を計算する。ただし、分離行列 W_k （ ）は、ステップS452においてリスケーリングされた分離行列であり、また、平均操作 $\langle \cdot \rangle_t$ は学習データの区間（図8の例では、学習データのブロック81）で行なう。

【0348】

パワー比の計算は、前記の式[8.6]・式[8.7]・式[8.11]のいずれかを適用して行なう。チャンネル k のパワー（分散）を V_k として、式[8.6]・式[8.7]・式[8.11]のいずれかを適用してパワー比 r_k を算出する。3つの式の違いは分母にある。式[8.6]の分母は同一区間内でチャンネル間でパワーを比較して最大のものである。式[8.7]の分母は非常に大きな音が入力されたときのパワーを V_{max} としてあらかじめ計算しておいたものである。式[8.11]の分母はパワー V_k をチャンネル間で平均したものである。どれを使用するかは使用環境に応じて使い分ければよく、例えば比較的静かな環境で使用される場合は式[8.7]を、背景ノイズが比較的大きな環境で使用される場合は式[8.6]を用いる。それに対し、式[8.11]を用い、かつ、 $r_{min} = 1/r_{max}$ となるように r_{min} と r_{max} を設定した場合は、広範囲の環境において比較的安定して動作する。なぜなら、周波数フィルタリングが適用されないチャンネルと適用されるチャンネルが、それぞれ少なくとも1つは存在するため、全チャンネルに対して突発音が除去されたり残ったりすることが起こらないからである。

【0349】

こうして計算されたチャンネル対応のパワー比 r_k は、図12に示すパワー比保持部135に格納され、必要に応じて分離処理部123の実行する分離処理（表の処理）に際して参照される。すなわち、パワー比に基づく関数（式[8.10]および図28）を用いることで、周波数フィルタリング（図27のステップS308）の実行態様をチャンネルごとに決定する際に利用される。

以上で、ステップS454のパワー比の計算処理についての説明を終了する。

【0350】

[5.本発明の信号処理装置のその他の実施例（変形例）について]

次に、上述した実施例と異なる実施例としての変形例について説明する。

（5-1.変形例1）

上述した実施例では、周波数フィルタリングのチャンネルごとの適用形態を決定する方法として、パワー比に基づく関数（式[8.10]および図28）を用いる方法について説明した。

【0351】

別の手段として、「チャンネル間で分離行列適用結果のパワーを比較し、パワーが最小のチャンネル以外のチャンネルに対して周波数フィルタリングを適用する」という方法も可能である。すなわち、パワーが最小のチャンネルを突発音の出力用として常に確保しておくのである。パワーが最小のチャンネルはどの音源にも対応していない可能性が高いため、このような簡易的な方法でも十分に実用になる。

【0352】

ただし、突発音の出力されるチャンネルが頻繁に切り替わる（例えば、突発音が鳴っている間に切り替わってしまう）のを防ぐために、工夫が必要である。ここでは、そのような工夫として以下の2点、すなわち、

（1）パワー比計算の平滑化

（2）全死角空間フィルタを学習初期値に反映させる。

これらの2点について説明する。

【0353】

（1）パワー比計算の平滑化

10

20

30

40

50

まず、パワー比計算の平滑化について説明する。

チャンネルごとのパワーを、以下に示す式 [1 0 . 1] に基づいて計算する。

【 0 3 5 4 】

【 数 1 1 】

$$r_k \leftarrow \gamma r_k + (1 - r_k) \frac{V_k}{V_{\max}} \quad \dots\dots[10.1]$$

$$0 \leq \gamma \leq 1 \quad \dots\dots[10.2]$$

$$\alpha_k = \begin{cases} \alpha_{\min} & (r_k = \min_k [r_k]) \\ \alpha & (\text{Otherwise}) \end{cases} \quad \dots\dots[10.3]$$

10

$$W(\omega) = W'(\omega) - \text{diag}(\alpha'_1, \dots, \alpha'_n) B'(\omega) \quad \dots\dots[10.4]$$

$$\alpha'_k = \begin{cases} \alpha'_{\min} & (r_k = \min_k [r_k]) \\ \alpha' & (\text{Otherwise}) \end{cases} \quad \dots\dots[10.5]$$

$$W(\omega) = \text{normalize}(W'(\omega) - \text{diag}(\alpha'_1, \dots, \alpha'_n) B'(\omega)) \quad \dots\dots[10.6]$$

$$W(\omega) \leftarrow \mu W(\omega) + (1 - \mu) [\text{normalize}(W'(\omega) - \text{diag}(\alpha'_1, \dots, \alpha'_n) B'(\omega))] \quad \dots\dots[10.7]$$

20

【 0 3 5 5 】

上記式 [1 0 . 1] に基づいてチャンネルごとのパワーを計算すると、パワーがほぼ同じ出力チャンネルが複数存在する場合に、パワー最小のチャンネルが頻繁に切り替わりやすくなる。例えば、観測信号が無音に近い場合、全ての出力チャンネルも無音に近くなる、すなわち出力パワーがほとんど同じになり、パワー最小チャンネルが僅差で決定されるため、それが頻繁に切り替えるという現象が発生し得る。

30

【 0 3 5 6 】

そのような現象を防ぐため、減算量（または `over-subtraction factor`） α_k は、先に示した式 [8 . 5] の代わりに式 [1 0 . 3] で計算する。ただし、 α_{\min} は 0 か 0 に近い正の値であり、 α は式 [8 . 5] と同様に α_k の最大値である。すなわち、パワー最小のチャンネルに対しては周波数フィルタリングを非適用に近い状態とし、それ以外のチャンネルに対しては周波数フィルタリングをそのまま適用する。なお、 α_{\min} を 0 に近い正の値とすることで、突発音用に確保されたチャンネルであっても、「消し残り」（「従来法の問題点」を参照）をある程度低減することができる。

【 0 3 5 7 】

（ 2 ）全死角空間フィルタを学習初期値に反映させる

40

次に、全死角空間フィルタを学習初期値に反映させる手法について説明する。パワー最小のチャンネルに対してのみ周波数フィルタリングを非適用（または、非適用に近い状態）とすることで、突発音はそのチャンネルのみに出力される。一方、突発音が鳴り続けると、やがて分離行列に反映され、周波数フィルタリングの作用がなくても一つのチャンネルのみに出力されるようになる。例えば図 7 に示す（ b 2 ）分離結果 2 のチャンネルである。そのため、両者でチャンネルを一致させるような工夫をしないと、突発音が鳴っている途中で出力先チャンネルが交替するという現象が発生し得る。

【 0 3 5 8 】

周波数フィルタリングを非適用にしたチャンネルにその後も突発音出力され続けるようにする（ = チャンネル交替を防ぐ ） ためには、「どのチャンネルに周波数フィルタリン

50

グが適用されたか（または適用されなかったか）」という情報が、次回の学習の初期値に反映されるようになっていけばよい。その方法について以下で説明する。

【0359】

前述した実施例では、学習初期値の設定を図25を参照して説明した「分離行列更新処理」のステップS253において（すなわち学習終了直後に）行なっていたが、変形例では、図31を参照して説明した「分離行列学習処理」のステップS412の分離行列Wの初期値設定時において（すなわち次の学習の直前で）行なう。理由は、学習開始直前における最新の分離行列および全死角空間フィルタの値を学習初期値に反映させるためである。（図12の、分離行列適用部126および全死角空間フィルタ適用部127からスレッド制御部131への矢印を参照。）

10

【0360】

図31を参照して説明した「分離行列学習処理」のステップS412の分離行列Wの初期値の計算として、先に示した式[10-4]を全周波数ビンについて行なう。ただし、この式[10-4]の左辺の $W(\quad)$ は学習初期値として、図13に示すスレッド制御部131の学習初期値保持部152および図14に示すスレッド演算部132の分離行列保持部164に格納される値、右辺の $W'(\quad)$ および $B'(\quad)$ はそれぞれ頻繁リスケーリング後の分離行列と全死角空間フィルタである。 $'k$ は式[10.3]の k と同一の値を用いてもよいが、式[10.5]のように異なる値を用いてもよい。例えば、周波数フィルタリングとしてスペクトル減算を用いた場合、式[10.3]では $\quad = 1.5$ とする一方、式[10.5]では $' = 1.0$ とする。（スペクトル減算においては $\quad > 1$ とすることで、*over-subtraction*という効果が得られるが、通常の引き算では $\quad = 1$ の方が望ましいため。）

20

【0361】

$' = 1$ かつ $'_{min} = 0$ （または0に近い正の値）とすると、式[10.4]で計算される分離行列 $W(\quad)$ は、パワー最小のチャンネルは突発音を出力し、それ以外のチャンネルは突発音を抑圧するという性質をもつ。従って、そのような値を学習初期値とすることで、学習後も同一のチャンネルに突発音出力され続ける可能性が高くなる。

【0362】

なお、必要に応じて、式[10.4]の代わりに式[10.6]の操作を行なってもよい。この式において、*normalize*(\quad)は、カッコ内の行列に対して、各行ベクトルのノルムを1に正規化する操作を表わす。

30

【0363】

また、学習スレッド同士で学習時間が重複する可能性がある場合（例えば図5では学習時間57と学習時間58とで時間が重複している）は、最新以外の分離行列も学習初期値に反映させた方が、どのチャンネルにどの音源が出力されるかが安定する。（理由は特開2008-147920を参照されたい。）変形例において、最新以外の分離行列を学習初期値に反映させるためには、式[10.6]の代わりに式[10.7]を用いる。この式において、右辺の $W(\quad)$ は前回計算された学習初期値であり、図13に示す学習初期値保持部152に格納されている。 μ は忘却係数であり、0以上、1以下の値をとる。

40

【0364】

この変形例による判別法を先に示した式[8.1]～[8.10]と比較すると、ちょうど n 個（マイクロホン数と同じ）の音源が連続的に鳴っている状態で新たに突発音が鳴った場合のみ不具合が発生する。すなわち、 $n-1$ 個の出力チャンネルに対しては周波数フィルタリングによって突発音が除去されるが、最もパワーが小さかったチャンネルに対しては周波数フィルタリングが適用されないため、突発音が重畳して出力される。（その場合でも、従来法と比べると $n-1$ 個のチャンネルに対してはメリットがある。）

【0365】

一方、突発音が鳴る前の音源数が n よりも少ない場合は、どのチャンネルから突発音出力されるかがあらかじめ予想できる。そのため、主に突発音の方を目的音として使用するようなアプリケーション（例えば、音楽が鳴っている環境において、ときどき音声でコ

50

マンドを入力する)においては、ICAの複数の出力チャンネルの内のどれが目的音なのかを特定するのが容易になるという利点がある。

【0366】

(5-2. 変形例2)

ICA以外の線形フィルタリングとの組み合わせ

上述した実施例では、リアルタイムICAに対して全死角空間フィルタと周波数フィルタリング(減算)とを組み合わせたが、ICA以外の線形フィルタリング処理と組み合わせることも可能であり、そうすることで「消し残り」を低減することが可能である。ここでは、線形フィルタリングと組み合わせた場合の構成例について説明した後、線形フィルタリングの具体例として分散最小ビームフォーマ(MVBF: Minimal Variance Beamformer)を用いた場合の処理について説明する。

10

【0367】

図33は、「全死角空間フィルタ&周波数フィルタリング」と線形フィルタリングとを組み合わせた場合の構成例を示す図である。図33に示す構成によって実行する処理は、図12に示す分離処理部123の実行する観測信号の分離処理(表の処理)とほぼ同様の処理である。

【0368】

何らかの線形フィルタの生成および適用を行なう系(フーリエ変換部303 線形フィルタ生成&適用部305)と全死角空間フィルタの生成および適用を行なう系(フーリエ変換部303 全死角空間フィルタ生成&適用部304)とを用意し、それぞれの適用結果に対して周波数フィルタリング(減算)を行なう。線形フィルタ生成&適用部305から全死角空間フィルタ生成&適用部304への破線は、全死角空間フィルタの適用結果に対して必要に応じてリスケーリング(全死角空間フィルタ適用結果のスケールを線形フィルタ適用結果のスケールに合わせる)ことを意味する。

20

【0369】

なお、ここでいう線形フィルタリングとは、分離行列 $W(\quad)$ を行列またはベクトルとし、 $W(\quad)$ と観測信号ベクトル $X(\quad, t)$ とを乗算することで(すなわち、分離結果: $Y(\quad, t) = W(\quad) X(\quad, t)$ の形式で)信号の分離・抽出・除去などを行なう処理のことを意味する。

【0370】

以下では、線形フィルタリングとして分散最小ビームフォーマを用いた場合について説明する。分散最小ビームフォーマは、目的音と妨害音とが混在する環境において、目的音の方向等の情報などを用いて目的音を抽出する技術の一つであり、適応ビームフォーマ(Adaptive beam former: ABF)と呼ばれる技術の一種である。詳細については、例えば以下の資料を参照されたい。

30

『音場の計測と指向性制御』小野順貴, 安藤繁

第22回センシングフォーラム資料, pp. 305-310, 9月, 2005. <http://hil.t.u-tokyo.ac.jp/publications/download.php?bib=Ono2005SensingForum09.pdf>

【0371】

以下では、図34を用いて分散最小ビームフォーマ(MVBF)について簡単に説明した後、全死角空間フィルタおよび周波数フィルタリングとの組み合わせを説明する。図34に示すような、目的音354(音源数1)と妨害音355(音源数1以上)とが混在する環境において、両者の音が混合した信号をn個のマイクロホン351~353で観測する。観測信号からなるベクトルを、前記した式[2.2]と同様に $X(\quad, t)$ とする。

40

【0372】

音源から各マイクロホンまでの伝達関数(インパルス応答)である $H_1(\quad) \sim H_n(\quad)$ は既知とし、それらを要素とするベクトルを $H(\quad)$ とする。ベクトル $H(\quad)$ は下に示す式[11.1]によって定義される。

【0373】

50

【数 1 2】

$$\mathbf{H}(\omega) = \begin{bmatrix} H_1(\omega) \\ \vdots \\ H_n(\omega) \end{bmatrix} \quad \dots\dots[11.1]$$

$$\mathbf{D}(\omega) = [D_1(\omega), \dots, D_n(\omega)] \quad \dots\dots[11.2]$$

$$Y(\omega, t) = \mathbf{D}(\omega) \mathbf{X}(\omega, t) \quad \dots\dots[11.3]$$

$$\mathbf{D}(\omega) \mathbf{H}(\omega) = 1 \quad \dots\dots[11.4]$$

$$\mathbf{D}(\omega) = \frac{\mathbf{H}(\omega)^H \Sigma_{XX}(\omega)^{-1}}{\mathbf{H}(\omega)^H \Sigma_{XX}(\omega)^{-1} \mathbf{H}(\omega)} \quad \dots\dots[11.5]$$

$$\Sigma_{XX}(\omega)^{-1} = \left\langle \mathbf{P}(\omega)^{-1} \mathbf{X}'(\omega, t) \mathbf{X}'(\omega, t)^H (\mathbf{P}(\omega)^{-1})^H \right\rangle_t^{-1} \quad \dots\dots[11.6]$$

$$= \mathbf{P}(\omega)^H \mathbf{P}(\omega) \quad \dots\dots[11.7]$$

$$\mathbf{D}(\omega) = \frac{\mathbf{H}(\omega)^H \mathbf{P}(\omega)^H \mathbf{P}(\omega)}{\mathbf{H}(\omega)^H \mathbf{P}(\omega)^H \mathbf{P}(\omega) \mathbf{H}(\omega)} \quad \dots\dots[11.8]$$

$$Q(\omega) = \left\langle Y(\omega, t) \overline{Z(\omega, t)} \right\rangle_t / \left\langle Z(\omega, t) \overline{Z(\omega, t)} \right\rangle_t \quad \dots\dots[11.9]$$

$$= \mathbf{D}(\omega) \Sigma_{XX}(\omega) \mathbf{B}(\omega)^H / \mathbf{B}(\omega) \Sigma_{XX}(\omega) \mathbf{B}(\omega)^H \quad \dots\dots[11.10]$$

$$\mathbf{B}'(\omega) = Q(\omega) \mathbf{B}(\omega) \quad \dots\dots[11.11]$$

$$Z'(\omega, t) = \mathbf{B}'(\omega) \mathbf{X}(\omega, t) \quad \dots\dots[11.12]$$

10

20

30

【0374】

ベクトル $\mathbf{H}(\quad)$ をステアリングベクトルと呼ぶ。なお、線形フィルタリングの具体例である分散最小ビームフォーマ (MVB F) においては、真の伝達関数を使用しなくても、 $H_1(\quad) \sim H_n(\quad)$ の間の比が正しければ目的音の抽出は可能である。そのため、ステアリングベクトルは、目的音の音源方向や位置などから算出したり、目的音のみが鳴っている (妨害音はすべて止まっている) 区間の観測信号から推定したりすることも可能である。

【0375】

図 3 4 に示すように、観測信号 $X_1(\quad, t) \sim X_n(\quad, t)$ にフィルタ係数 ($D_1(\quad) \sim D_n(\quad)$) を乗じるフィルタ 3 5 8 を通して総和したものを分離結果 $Y(\quad, t)$ 3 5 9 とする。分離結果 $Y(\quad, t)$ 3 5 9 は、フィルタ係数を要素とするベクトル $\mathbf{D}(\quad)$ (式 [11.2]) を用いて式 [11.3] のように表わせる。ICA の場合と異なり、出力は 1 チャンネル、すなわち $Y(\quad, t)$ はスカラーである。

40

【0376】

分散最小ビームフォーマ (MVB F) のフィルタである $\mathbf{D}(\quad)$ は、式 [11.5] で求められる。この式において、 $\Sigma_{XX}(\quad)$ は観測信号の共分散行列であり、ICA の場合と同様に前記した式 [4.4] での操作で得られる。なお、式 [11.5] は、「目的音 3 5 4 由来の音はそのまま残す」という制約 (式 [11.4] に相当) の下で、 $Y(\quad, t)$ の分散 $\langle |Y(\quad, t)|^2 \rangle$ を最小にする MVB F フィルタ $\mathbf{D}(\quad)$ を求めるとい

50

D () は、目的音の方向のゲインを 1 に保つ一方で、それぞれの妨害音の方向には死角を形成している。

【 0 3 7 7 】

しかし、M V B F による音源抽出には、I C A における「消し残り」と同様の課題がある。すなわち、妨害音の音源数がマイク数以上の場合や、妨害音が無指向性の場合 (= 点音源でない場合) などには、妨害音を死角で消しきれなくなるため、抽出の性能が低下する。また、マイクロホンの配置によって、ある周波数帯域での抽出精度が低下する可能性がある。

【 0 3 7 8 】

また、計算量の制限などにより、フィルタの更新を毎フレームではなく複数フレームに 1 回の頻度でしか行なえない場合もある。その場合は、「追従遅れ」と同様の現象も発生する。例えば、フィルタの更新を 10 フレームに 1 回の頻度で行なった場合、突発音が鳴ってから最大で 9 フレームの間は、その音が除去されずに出力されてしまう。

【 0 3 7 9 】

一方、本発明の全死角空間フィルタと周波数フィルタリングを M V B F と組み合わせることで、「消し残り」にも「追従遅れ」にも対処可能となる。その際、共分散行列に対して固有値分解を行なうことで、計算量の増加なしに全死角空間フィルタを計算することができる。以下、その方法について説明する。

【 0 3 8 0 】

観測信号の共分散行列を、前記した式 [4 . 4] でフレームごとに計算する。そして M V B F フィルタの更新頻度に合わせ、共分散行列に対して固有値分解を行なう (前記式 [6 . 1])。I C A と組み合わせた場合と同様に、全死角空間フィルタは、最小の固有値に対応した固有ベクトルの転置である (式 [6 . 2])。

【 0 3 8 1 】

固有値分解の結果を用いると、逆行列を含まないシンプルな式で M V B F のフィルタを計算することができる。前記の式 [9 . 9] から計算される無相関化行列 $P ()$ を用いると、観測信号の共分散行列は式 [1 1 . 7] のように書け、それを用いて M V B F のフィルタは式 [1 1 . 8] のように書くことができるからである。言い換えると、式 [1 1 . 5] において観測信号の共分散行列を求める手段として固有値分解を用いれば、同時に全死角空間フィルタも求まっている。

【 0 3 8 2 】

こうして求めた全死角空間フィルタ $B ()$ に対して、リスケーリング (全死角空間フィルタ適用結果のスケールを M V B F フィルタ適用結果のスケールに合わせる処理) を行なう。リスケーリングは、式 [1 1 . 9] で計算される係数 $Q ()$ を全死角空間フィルタ $B ()$ に乗じることで行なう (式 [1 1 . 1 1])。リスケーリング済み全死角空間フィルタの適用結果 $Z' (, t)$ は、式 [1 1 . 1 2] で行なう。M V B F 側の出力が 1 チャンネルなので、 $Z' (, t)$ も 1 チャンネル (すなわち $Z' (, t)$ はスカラ) である。

【 0 3 8 3 】

こうして生成された M V B F の結果 (式 [1 1 . 3]) と全死角空間フィルタ適用結果 (式 [1 1 . 1 2]) との間で、周波数フィルタリング (広い意味での減算) を行なう。それによって、M V B F の結果から「消し残り」が除去されるようになる。また、M V B F フィルタの更新を複数フレームごとに行なったために「追従遅れ」が発生している場合でも、突発音を除去することができるようになる。

【 0 3 8 4 】

[6 . 本発明の信号処理装置の構成に基づく効果についてのまとめ]

以下、本発明の信号処理装置の構成に基づく効果についてまとめて記載する。本発明の信号処理装置の構成に基づく効果としては以下のような効果がある。

(1) 独立成分分析を用いたリアルタイム音源分離システムにおいて、分離行列適用結果の他に全死角空間フィルタ適用結果も生成し、両者の間で周波数フィルタリングまたは

10

20

30

40

50

減算を行なうことにより、突発音を除去することができる。

(2) 周波数フィルタリングを適用する強さ(または減算の量)を、音源に対応した信号が突発音発生前に出力されていたかに応じて切り替えることにより、

a) 音源に対応した信号が出力されているチャンネルからは突発音が除去され、

b) 音源に対応した信号が出力されていないチャンネルからは突発音出力されることが可能になる。

(3) 分離行列に対して最短で1フレーム毎にリスケーリングを行なうことにより、突発音出力される際の歪みを低減することができる。

【0385】

以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、特許請求の範囲の欄を参酌すべきである。

【0386】

また、明細書中において説明した一連の処理はハードウェア、またはソフトウェア、あるいは両者の複合構成によって実行することが可能である。ソフトウェアによる処理を実行する場合は、処理シーケンスを記録したプログラムを、専用のハードウェアに組み込まれたコンピュータ内のメモリにインストールして実行させるか、あるいは、各種処理が実行可能な汎用コンピュータにプログラムをインストールして実行させることが可能である。例えば、プログラムは記録媒体に予め記録しておくことができる。記録媒体からコンピュータにインストールする他、LAN(Local Area Network)、インターネットといったネットワークを介してプログラムを受信し、内蔵するハードディスク等の記録媒体にインストールすることができる。

【0387】

なお、明細書に記載された各種の処理は、記載に従って時系列に実行されるのみならず、処理を実行する装置の処理能力あるいは必要に応じて並列的あるいは個別に実行されてもよい。また、本明細書においてシステムとは、複数の装置の論理的集合構成であり、各構成の装置が同一筐体内にあるものには限らない。

【産業上の利用可能性】

【0388】

以上、説明したように、本発明の一実施例の構成によれば、複数音源からの出力を混合した混合信号からなる観測信号に対して、独立成分分析(ICA: Independent Component Analysis)を適用した学習処理により、混合信号を分離する分離行列を求めて分離信号を生成するとともに、観測信号として検出された音源に対する死角を持つ全死角空間フィルタを適用して検出音を除去した全死角空間フィルタ適用信号を生成する。さらに、分離信号に含まれる全死角空間フィルタ適用信号に対応する信号成分を除去するフィルタリング処理を実行し、周波数フィルタリング処理結果から音源分離結果を生成する。本構成により、例えば突発音等が含まれる混合信号に対する高精度な音源分離が可能となる。

【符号の説明】

【0389】

- 41 学習データブロック
- 42 現在時刻の観測信号
- 43 分離行列
- 44 現在時刻の分離結果
- 45 分離結果スペクトログラム区間
- 46 現在時刻を含む観測信号ブロック
- 51 ~ 53 適用分離行列規定区間
- 54 観測信号蓄積区間
- 55 学習開始タイミング

10

20

30

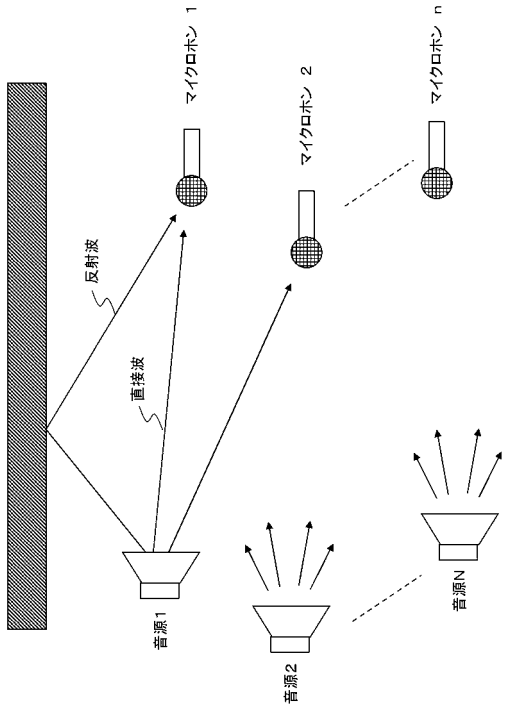
40

50

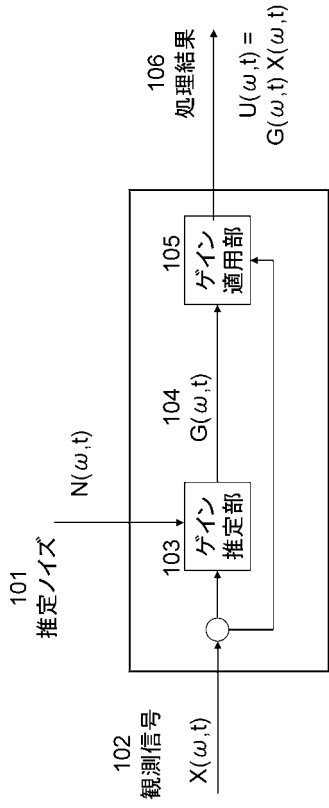
5 7 , 5 8	学習区間	
6 1 , 6 2	音源 1 出力区間	
6 3	音源 2 出力区間	
6 4	学習データブロック	
6 5	ブロック終端から現在時刻までの区間	
6 7	音源 1 の無音区間	
6 9	現在時刻の音源出力	
7 1	継続音	
7 2	突発音	
7 8 , 7 9	消し残り	10
8 1	学習データブロック区間	
8 2	現在時刻の観測信号	
8 3	分離行列	
8 4	全死角空間フィルタ	
8 5	周波数フィルタリング	
9 1	継続音	
9 2	突発音	
1 0 1	推定ノイズ	
1 0 2	観測信号	
1 0 3	ゲイン推定部	20
1 0 4	ゲイン	
1 0 5	ゲイン適用部	
1 0 6	処理結果	
1 1 1	推定ノイズ	
1 1 2	観測信号	
1 1 3	ゲイン推定部	
1 1 4	ゲイン	
1 1 5	ゲイン適用部	
1 1 6	処理結果	
1 2 1	マイクロホン	30
1 2 2	A D 変換部	
1 2 3	分離処理部	
1 2 4	フーリエ変換部	
1 2 5	共分散行列計算部	
1 2 6	分離行列適用部	
1 2 7	全死角空間フィルタ適用部	
1 2 8	周波数フィルタリング部	
1 2 9	フーリエ逆変換部	
1 3 0	学習処理部	
1 3 1	スレッド制御部	40
1 3 2	スレッド演算部	
1 3 3	分離行列保持部	
1 3 4	全死角空間フィルタ保持部	
1 3 5	パワー比保持部	
1 3 6	後段処理部	
1 5 1	現フレーム番号保持カウンタ	
1 5 2	学習初期値保持部	
1 5 3	蓄積開始予定タイミング指定情報保持部	
1 5 4	観測信号蓄積タイミング情報保持部	
1 5 5	ポインタ保持部	50

1 6 0	全死角空間フィルタ保持部	
1 6 1	観測信号バッファ	
1 6 2	分離結果バッファ	
1 6 3	学習演算部	
1 6 4	分離行列保持部	
1 6 5	状態格納部	
1 6 6	カウンタ	
1 6 7	観測信号の開始・終了タイミング保持部	
1 6 8	学習終了フラグ	
1 6 9	前処理用データ保持部	10
1 7 0	打ち切りフラグ	
1 7 1 , 1 7 4	蓄積中状態	
1 7 2	学習中状態	
1 7 3	待機中状態	
1 8 1	初期状態	
1 8 2	待機中状態	
1 8 3	蓄積中状態	
1 8 4	学習中状態	
1 9 1 ~ 1 9 3	フレーム	
3 0 1	マイクロホン	20
3 0 2	A D変換部	
3 0 3	フーリエ変換部	
3 0 4	全死角空間フィルタ生成 & 適用部	
3 0 5	線形フィルタ生成 & 適用部	
3 0 6	周波数フィルタリング部	
3 0 7	フーリエ逆変換部	
3 0 8	後段処理部	
3 5 1 ~ 3 5 3	マイクロホン	
3 5 4	目的音	
3 5 5	妨害音	30
3 5 8	フィルタ	
3 5 9	分離結果	

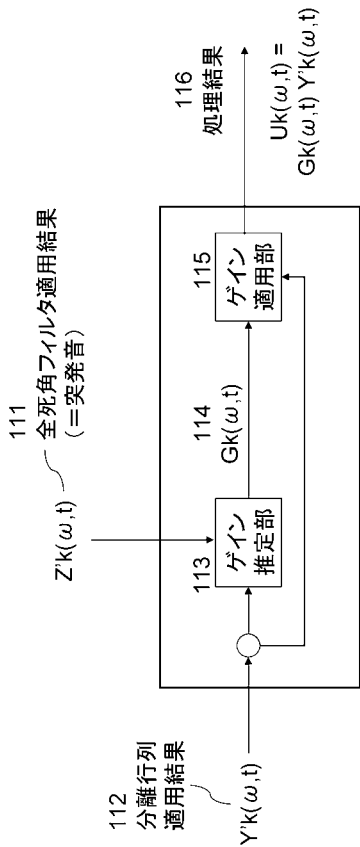
【図 1】



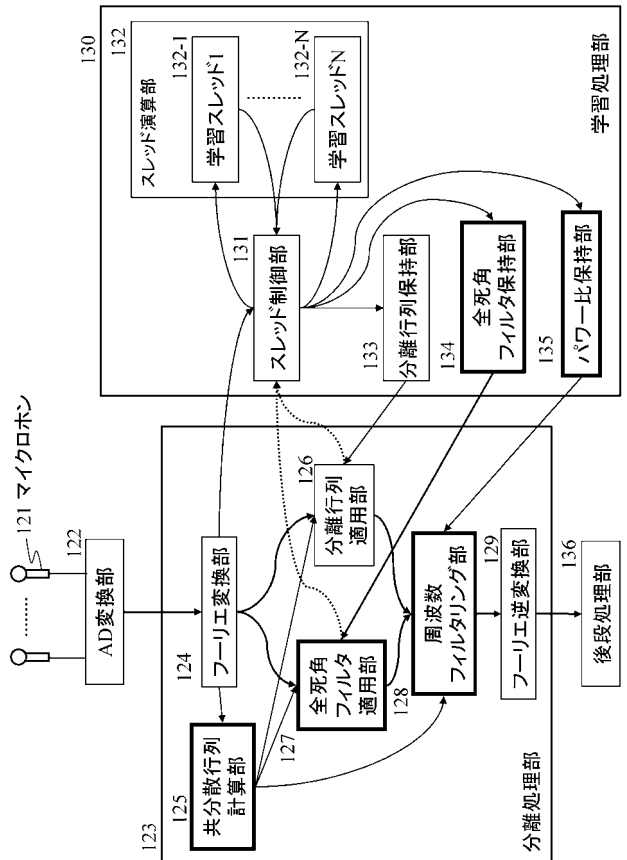
【図 10】



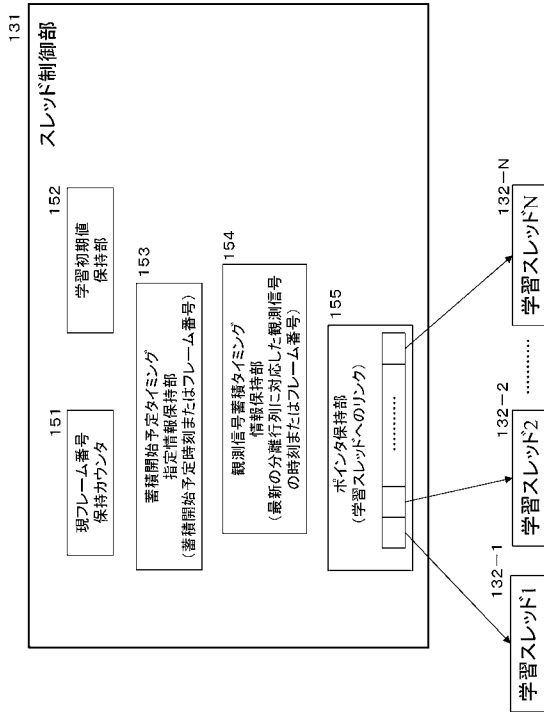
【図 11】



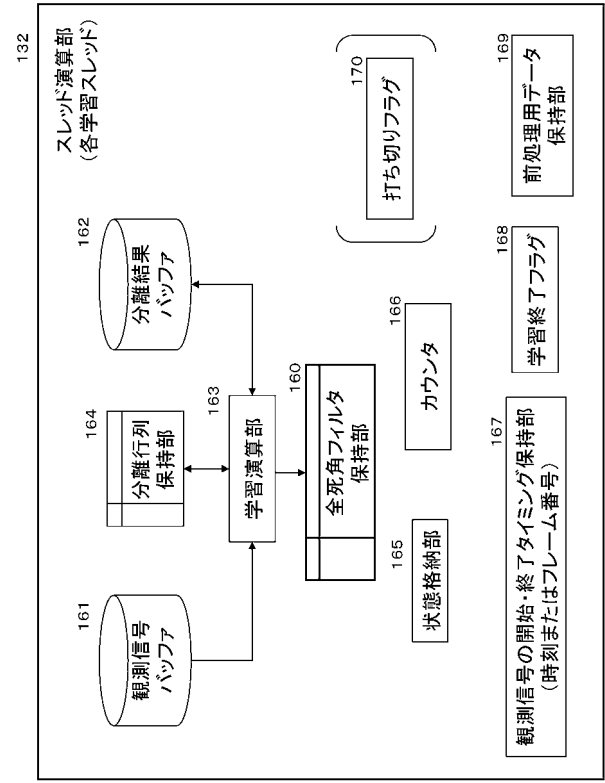
【図 12】



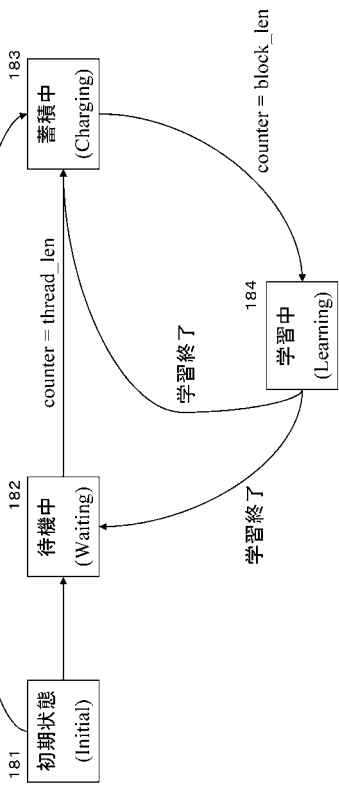
【 図 1 3 】



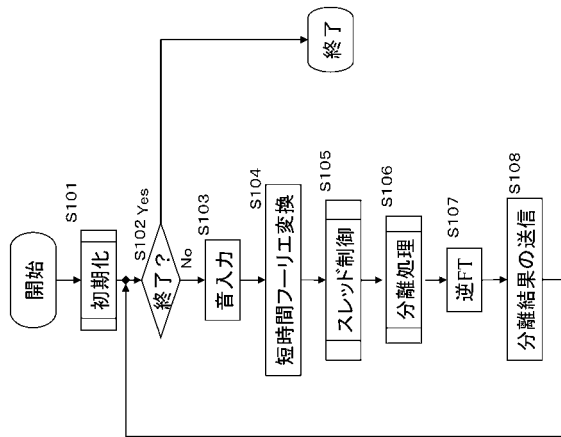
【 図 1 4 】



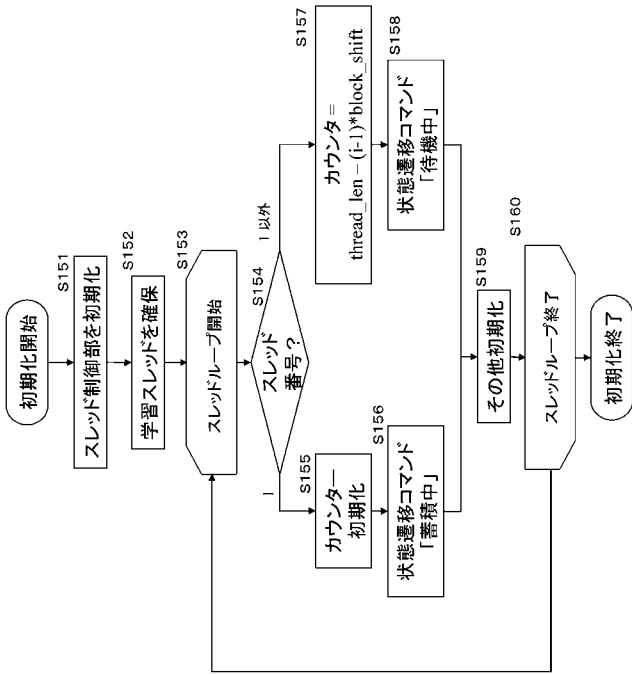
【 図 1 6 】



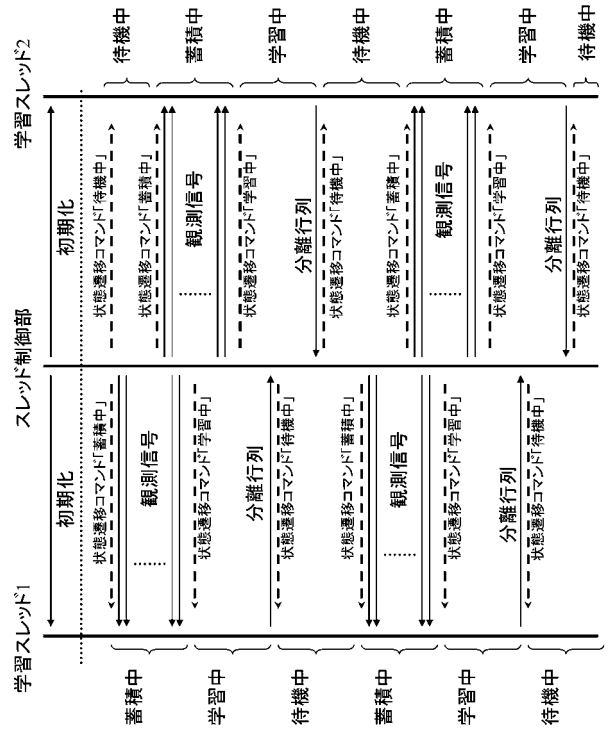
【 図 1 7 】



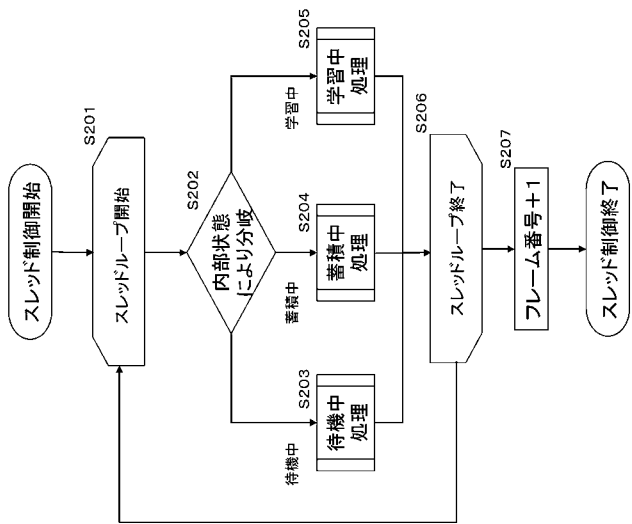
【図 19】



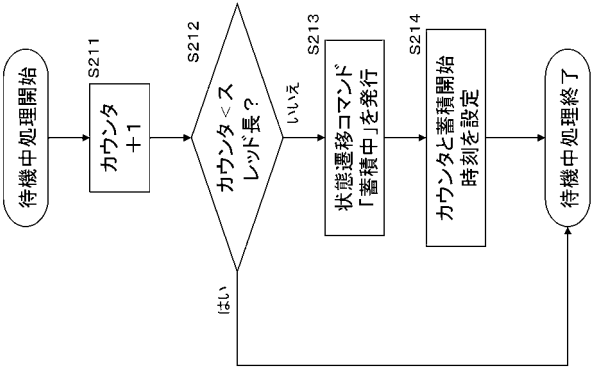
【図 20】



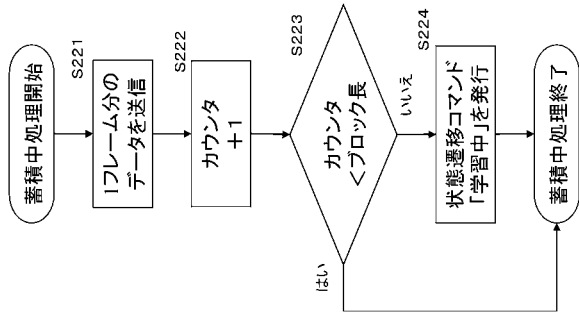
【図 21】



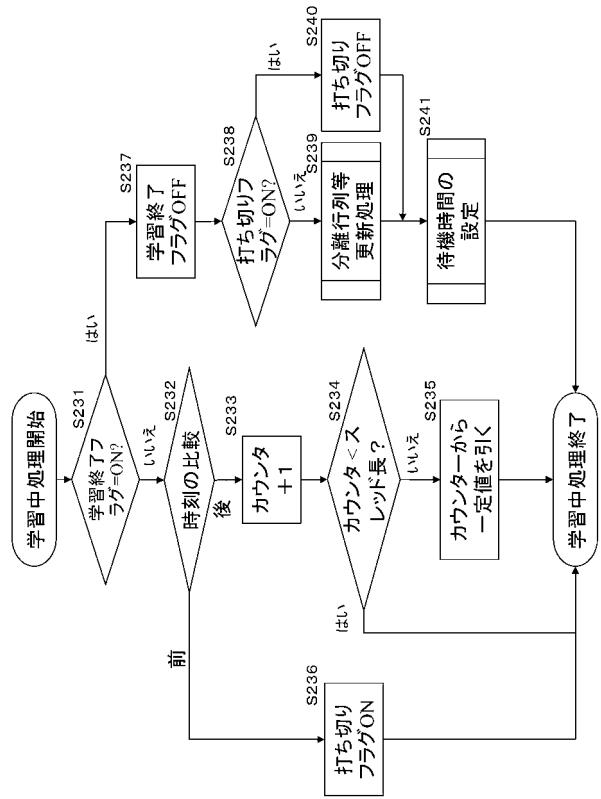
【図 22】



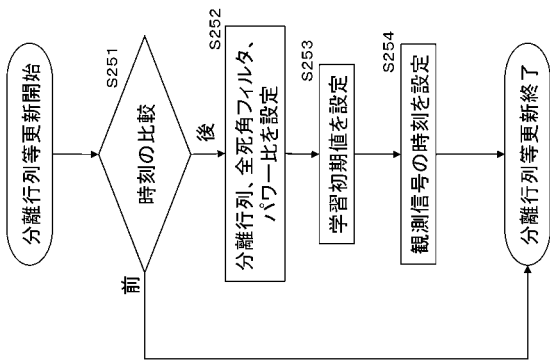
【図 2 3】



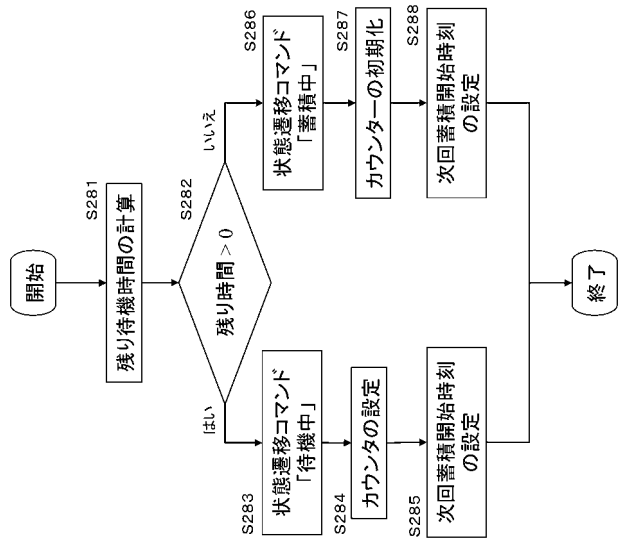
【図 2 4】



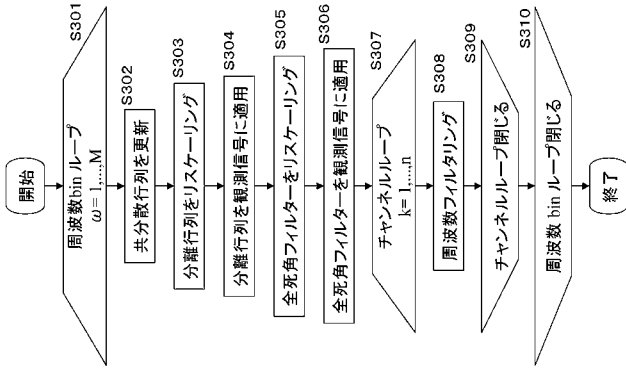
【図 2 5】



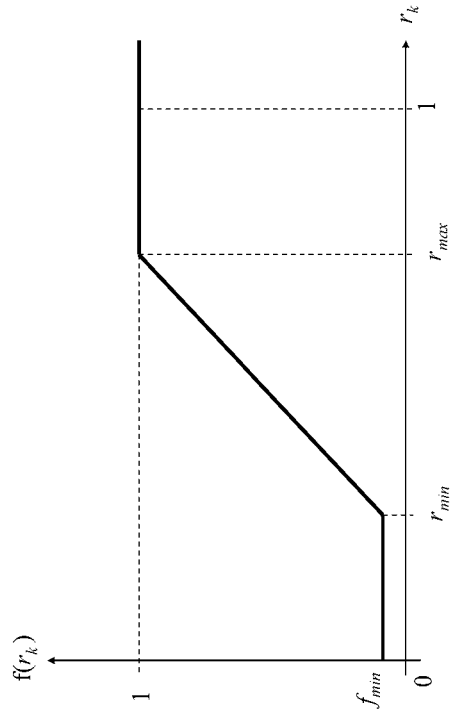
【図 2 6】



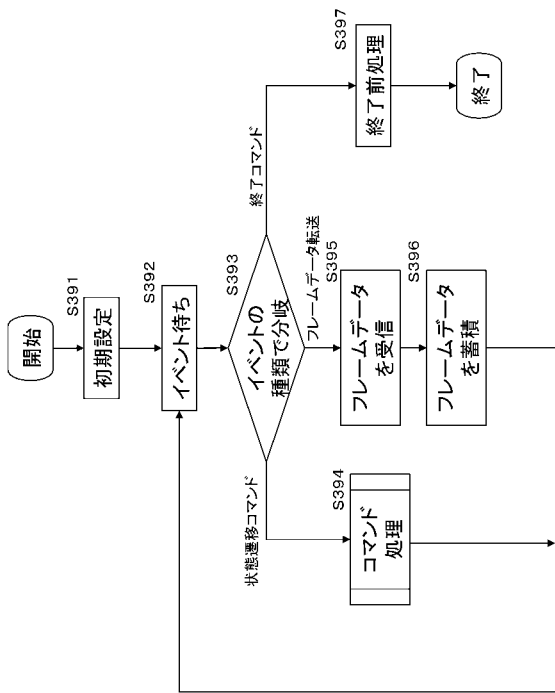
【 図 2 7 】



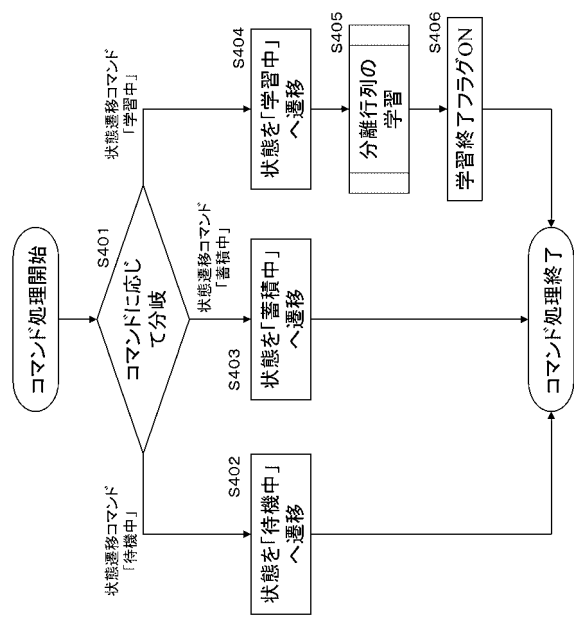
【 図 2 8 】



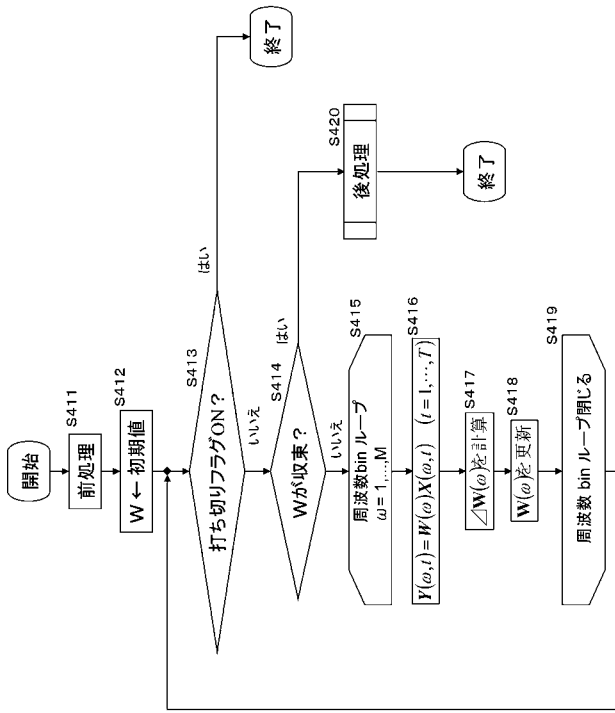
【 図 2 9 】



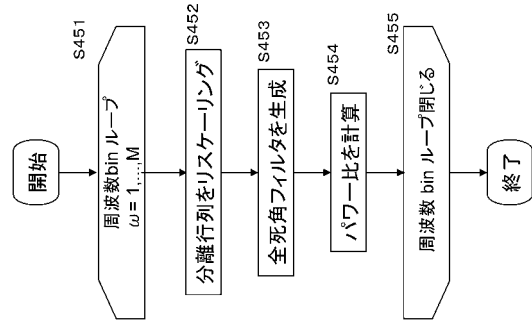
【 図 3 0 】



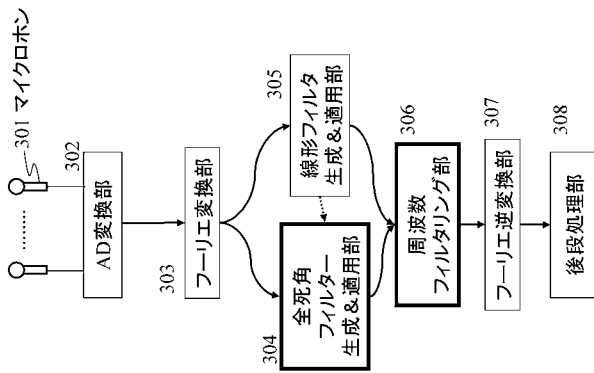
【 図 3 1 】



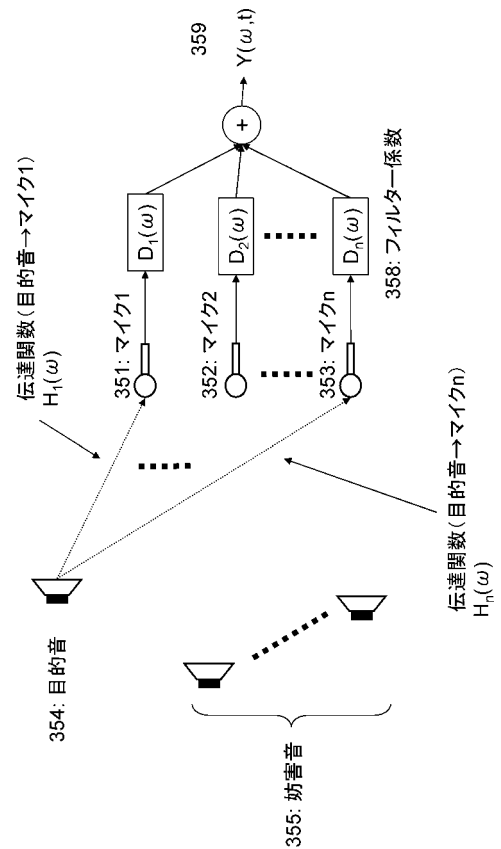
【 図 3 2 】



【 図 3 3 】

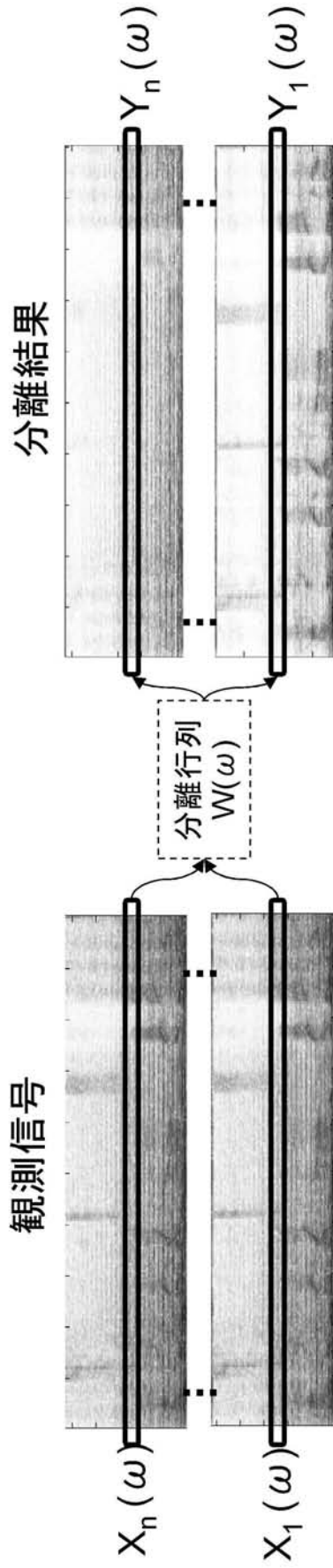


【 図 3 4 】

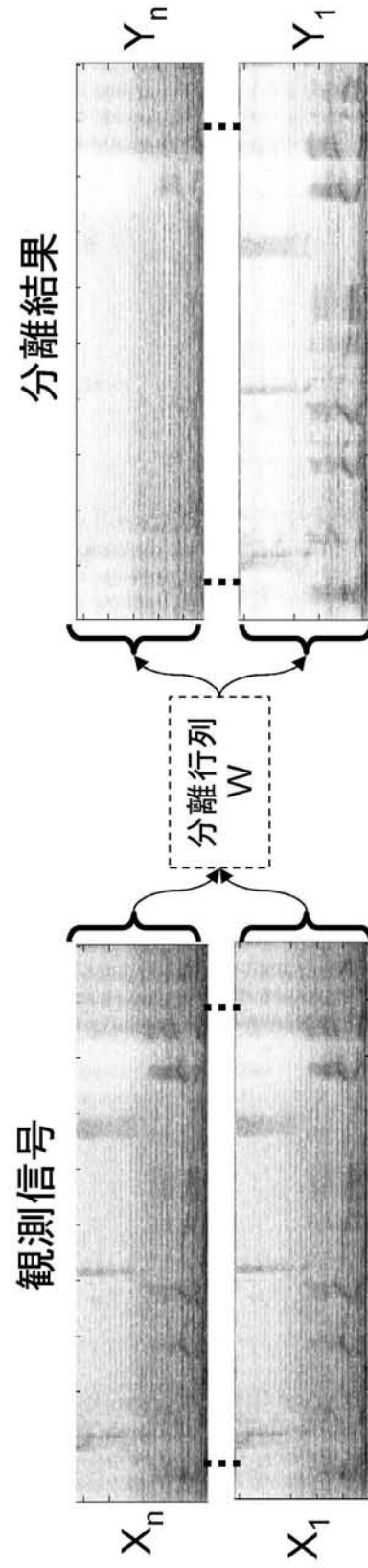


【 図 2 】

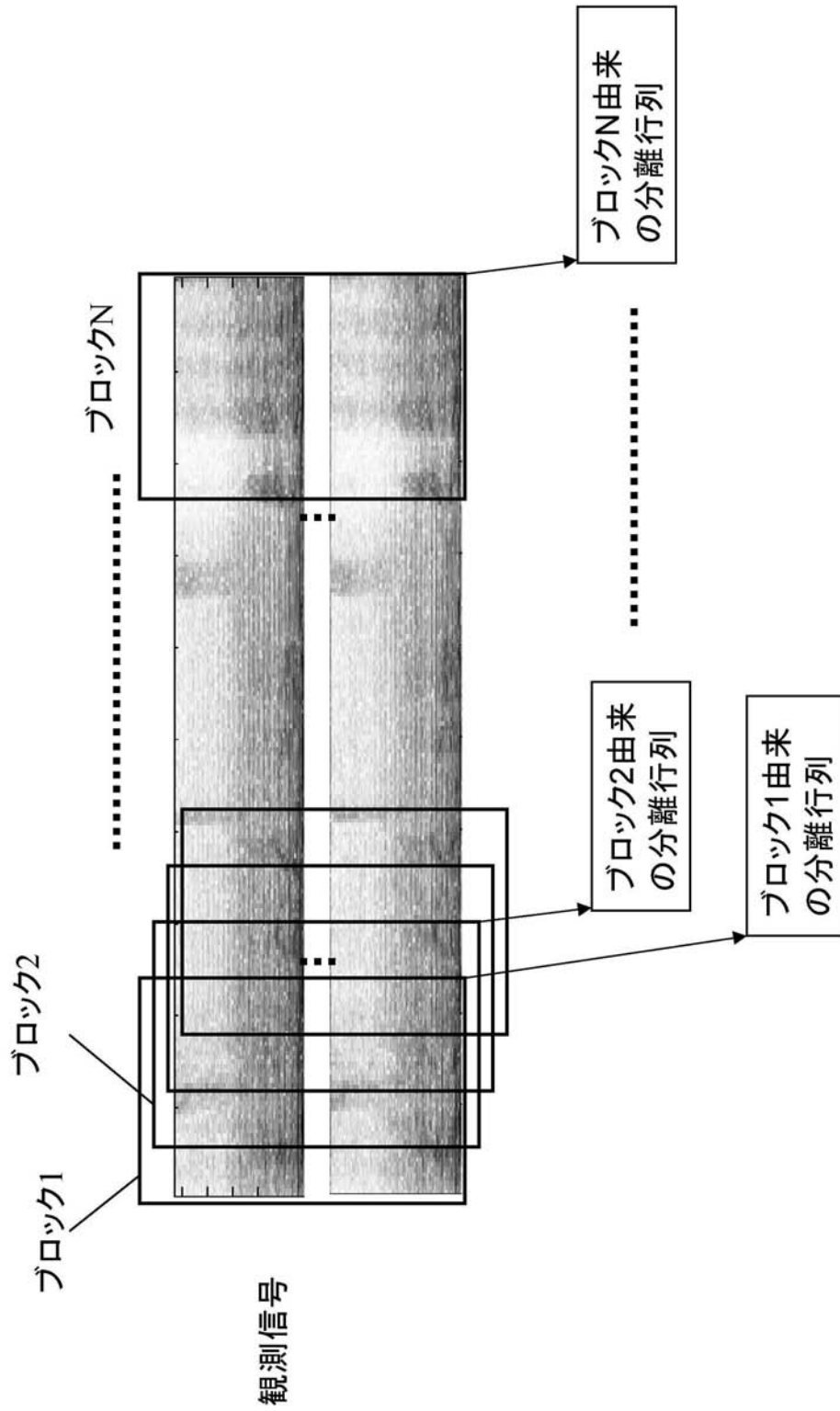
(a) 周波数ビンごとの分離



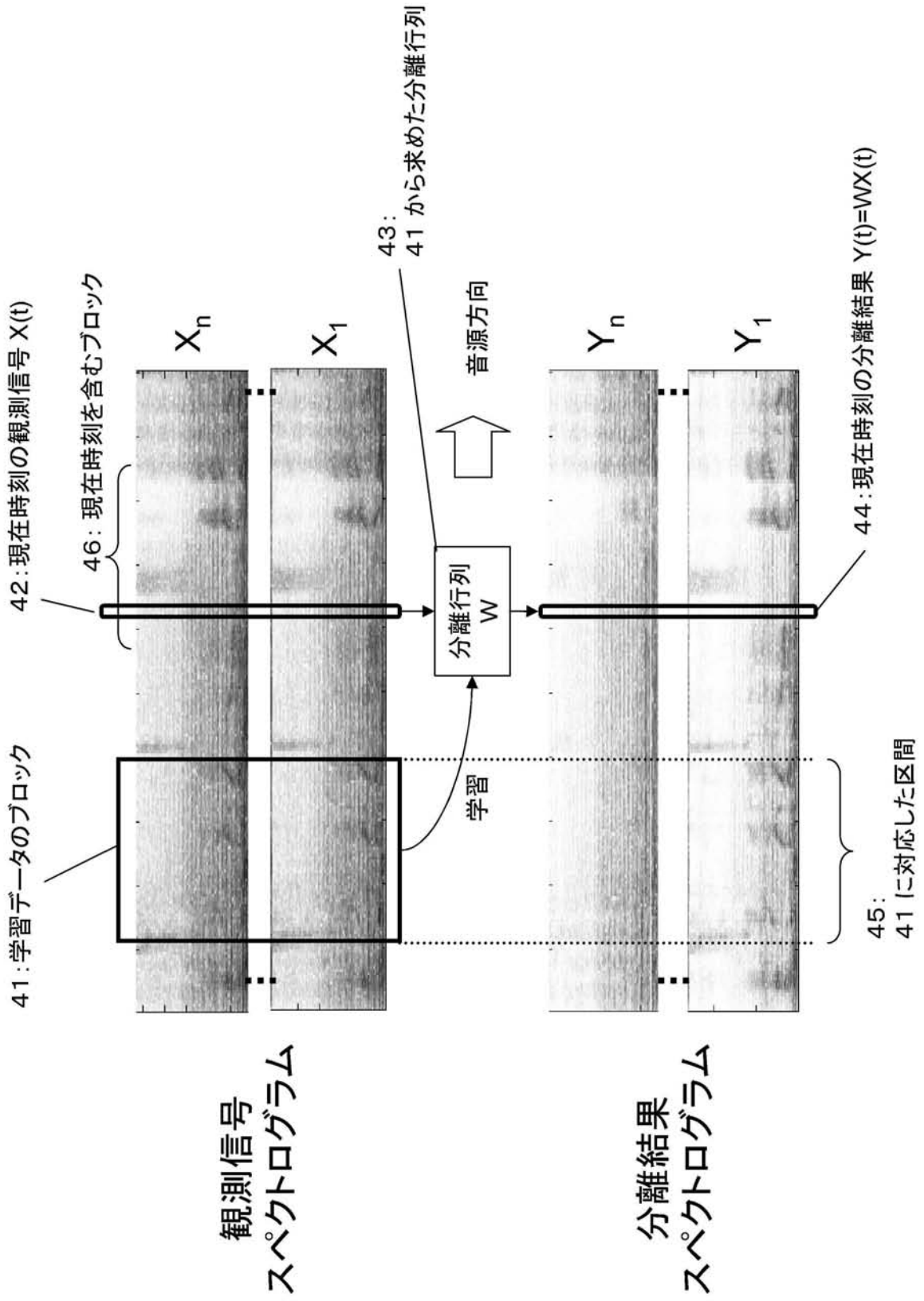
(b) 全周波数ビンの分離



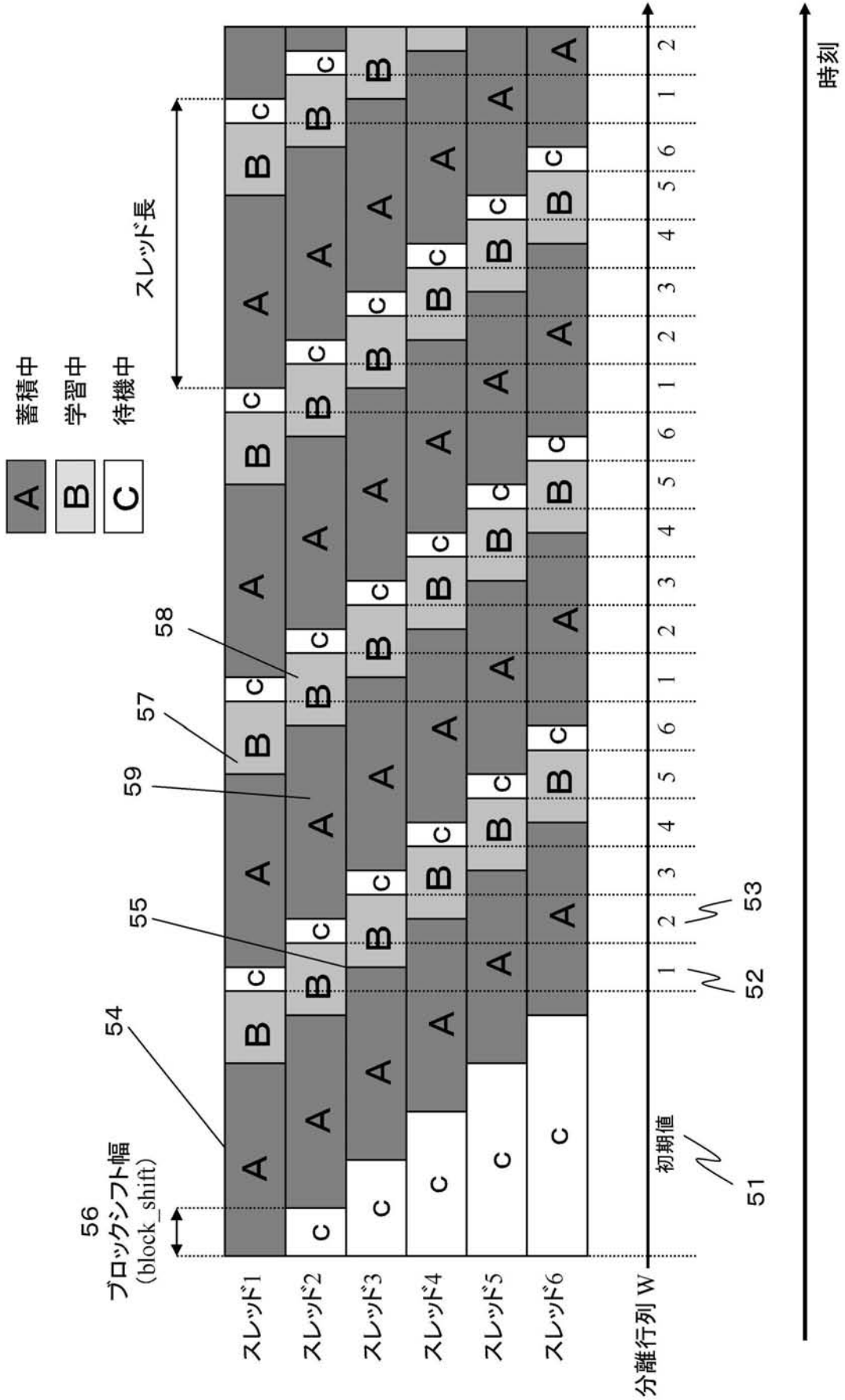
【図3】



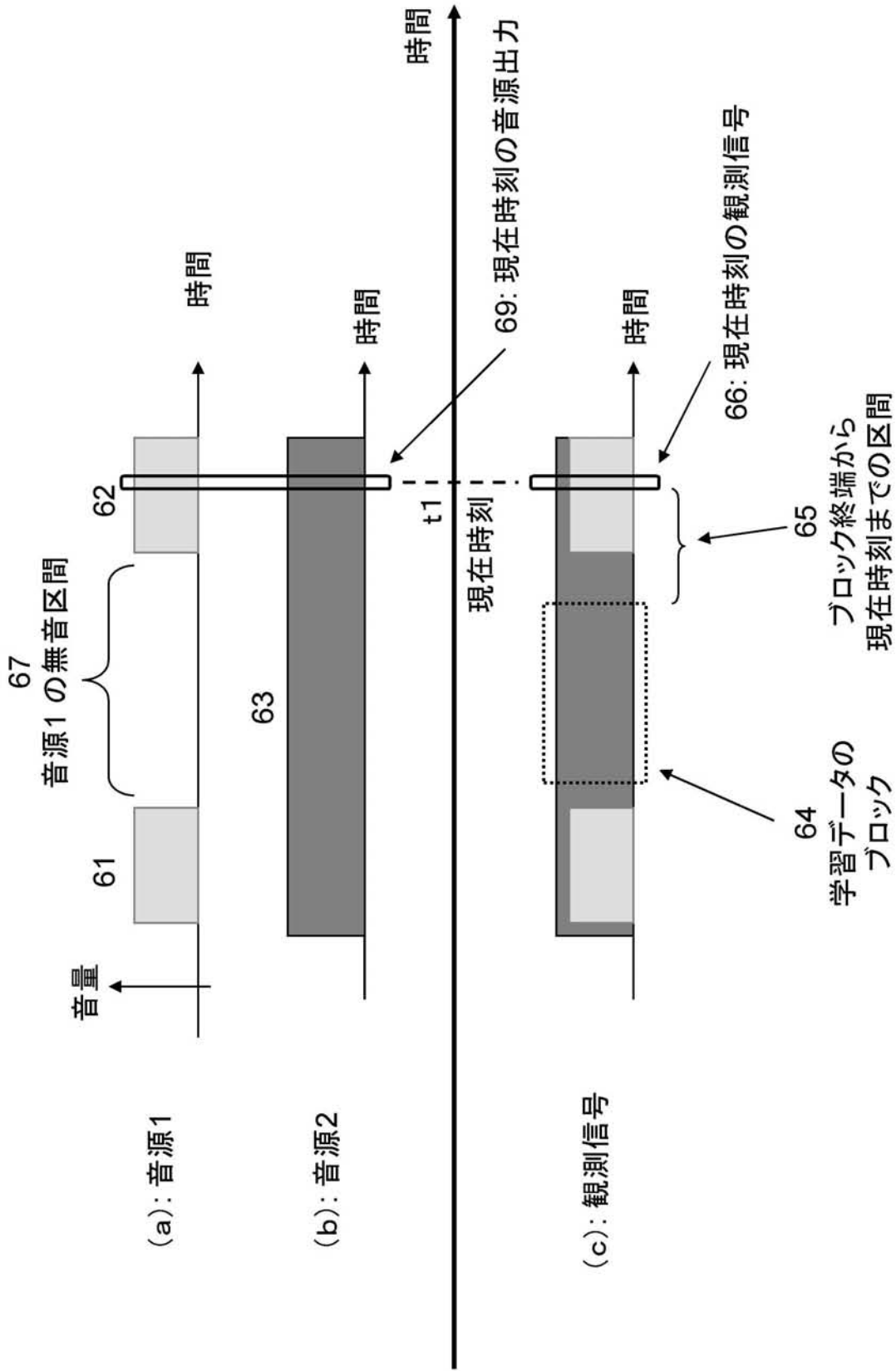
【 図 4 】



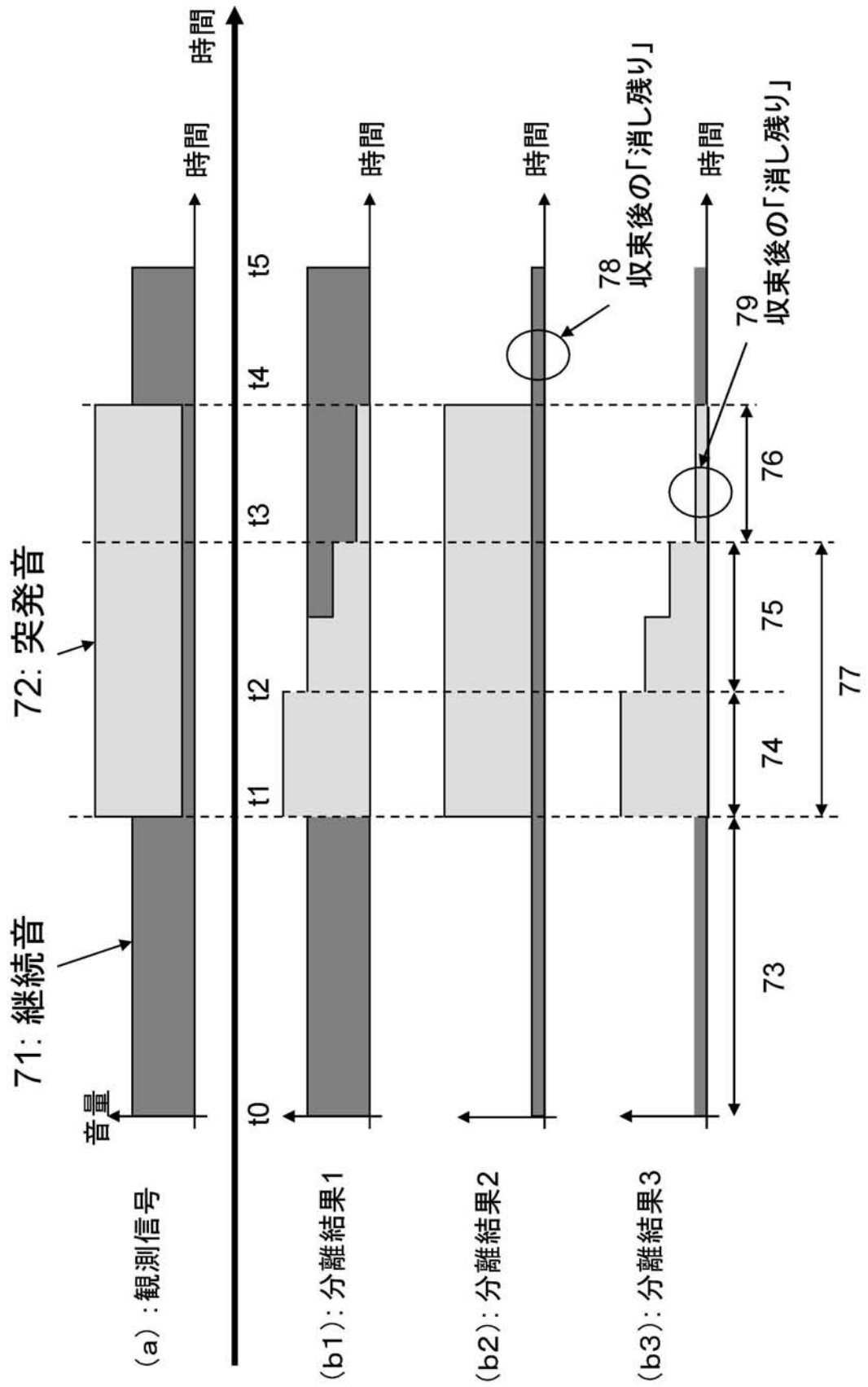
【図5】



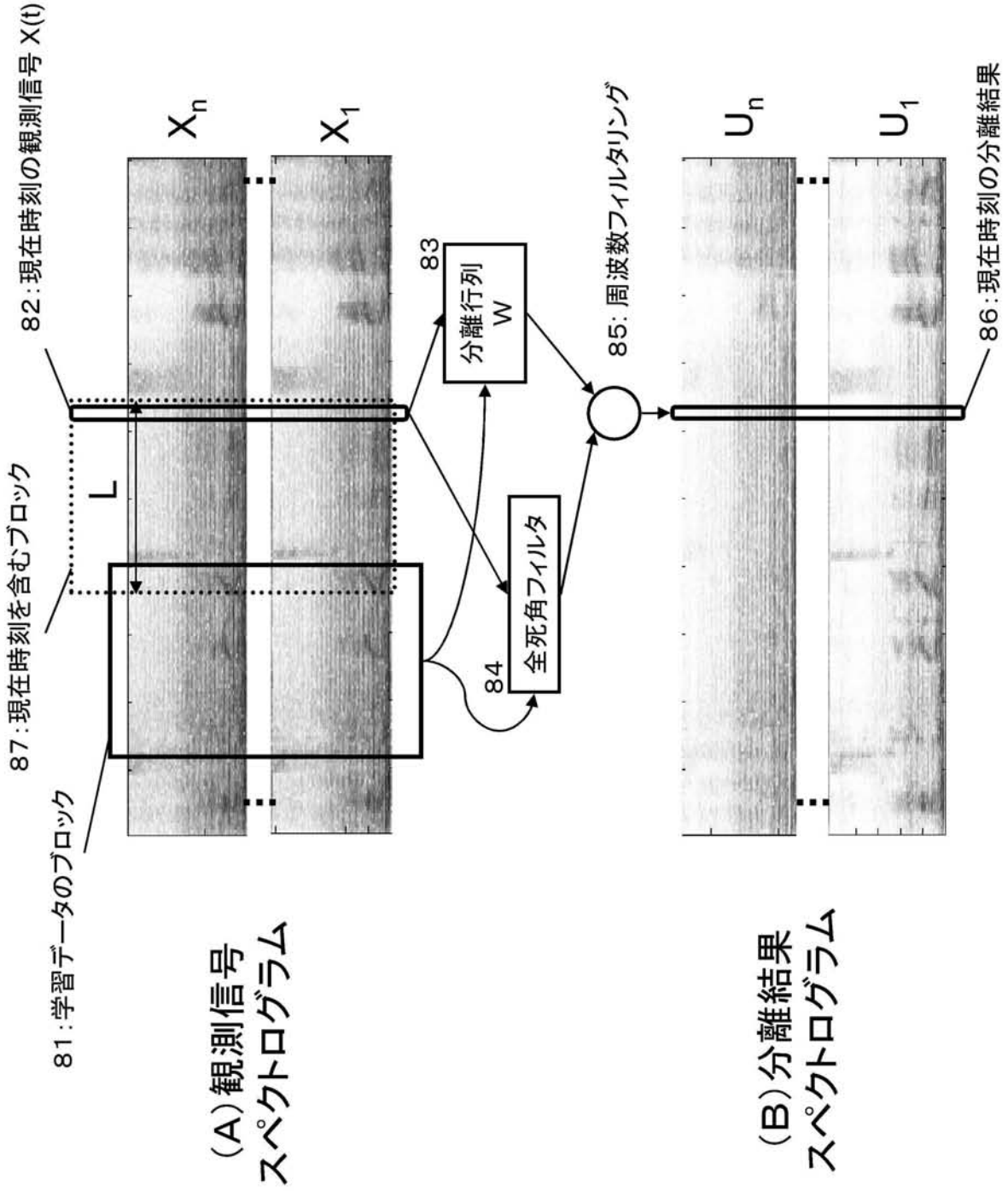
【図6】



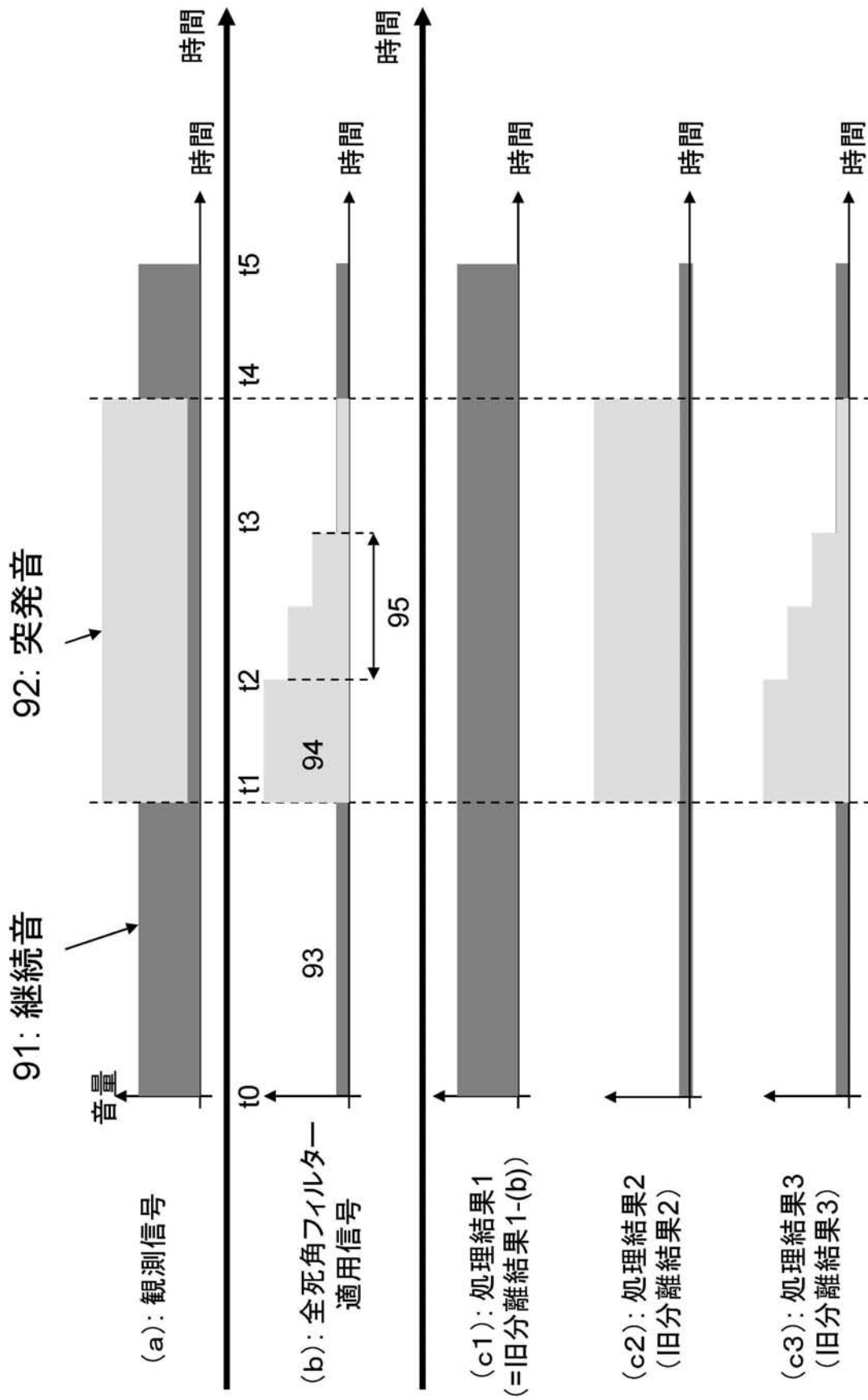
【 図 7 】



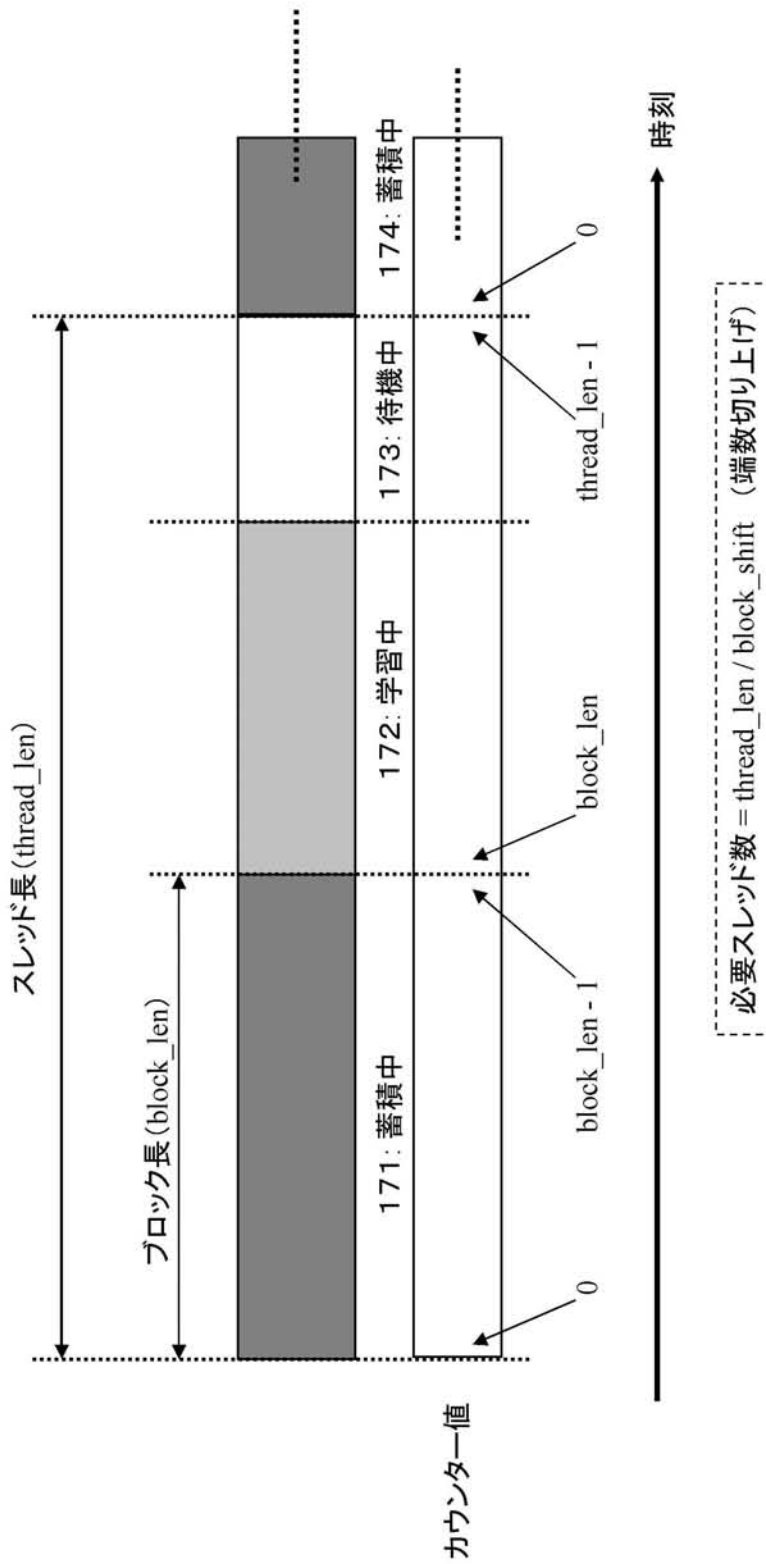
【 図 8 】



【 図 9 】



【図 15】



【 図 1 8 】

