

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6835182号  
(P6835182)

(45) 発行日 令和3年2月24日 (2021.2.24)

(24) 登録日 令和3年2月8日 (2021.2.8)

(51) Int. Cl.	F I
G 1 O L 13/06 (2013.01)	G 1 O L 13/06 1 4 O
G 1 O H 1/00 (2006.01)	G 1 O H 1/00 B
G 1 O L 13/02 (2013.01)	G 1 O L 13/02 1 1 O Z
G 1 O L 13/033 (2013.01)	G 1 O L 13/033 1 O 2 B
G 1 O L 13/00 (2006.01)	G 1 O L 13/00 1 O O Y

請求項の数 7 (全 35 頁)

(21) 出願番号	特願2019-196824 (P2019-196824)	(73) 特許権者	000001443
(22) 出願日	令和1年10月30日 (2019.10.30)		カシオ計算機株式会社
(62) 分割の表示	特願2018-118056 (P2018-118056) の分割		東京都渋谷区本町1丁目6番2号
原出願日	平成30年6月21日 (2018.6.21)	(74) 代理人	100074099 弁理士 大菅 義之
(65) 公開番号	特開2020-24456 (P2020-24456A)	(74) 代理人	100121083 弁理士 青木 宏義
(43) 公開日	令和2年2月13日 (2020.2.13)	(74) 代理人	100138391 弁理士 天田 昌行
審査請求日	令和2年1月9日 (2020.1.9)	(72) 発明者	段城 真 東京都羽村市栄町3丁目2番1号 カシオ 計算機株式会社 羽村技術センター内
		(72) 発明者	太田 文章 東京都羽村市栄町3丁目2番1号 カシオ 計算機株式会社 羽村技術センター内 最終頁に続く

(54) 【発明の名称】 電子楽器、電子楽器の制御方法、及びプログラム

(57) 【特許請求の範囲】

【請求項1】

設定された音色に応じた楽器音波形データであって、ユーザによる和音の演奏操作に応じて取得された複数の音高データそれぞれに応じた楽器音波形データを、前記和音の演奏操作に応じて取得し、

或る歌い手の歌声を学習した学習済みモデルに、演奏前からメモリに記憶されている歌詞データを入力することにより、前記学習済みモデルが出力する音響特徴量データを、前記和音の演奏操作に応じて取得し、

取得された前記楽器音波形データに、取得された前記音響特徴量データが示す音響特徴量を合成し、

合成することにより生成される和音の推論歌声データを、ユーザが歌わなくても出力する、

電子楽器。

【請求項2】

励振源信号として複数の第1発音チャネルからそれぞれ発振された前記楽器音波形データに、前記音響特徴量データが示す音響特徴量を合成することにより、前記和音の推論歌声データを生成する、

請求項1に記載の電子楽器。

【請求項3】

前記複数の第1発音チャネル以外の第2発音チャネルは、伴奏データの出力のために使

用され、

前記第 2 発音チャネルからの出力には、前記音響特徴量は合成されない、  
請求項 2 に記載の電子楽器。

【請求項 4】

前記和音の推論歌声データを、設定されている第 1 テンポで出力し、  
ユーザによるテンポ変更指示操作に応じて、前記第 1 テンポから第 2 テンポに変更し、  
変更された前記第 2 テンポで、前記和音の推論歌声データを出力する、  
請求項 1 乃至 3 のいずれかに記載の電子楽器。

【請求項 5】

前記設定された音色は、少なくともブラス音、ストリング音、オルガン音のいずれかを  
含む、請求項 1 乃至 4 のいずれかに記載の電子楽器。

10

【請求項 6】

電子楽器のコンピュータに、  
設定された音色に応じた楽器音波形データであって、ユーザによる和音の演奏操作に応  
じて取得された複数の音高データそれぞれに応じた楽器音波形データを、前記和音の演奏  
操作に応じて取得させ、

或る歌い手の歌声を学習した学習済みモデルに、演奏前からメモリに記憶されている歌  
詞データを入力することにより、前記学習済みモデルが出力する音響特徴量データを、前  
記和音の演奏操作に応じて取得させ、

取得された前記楽器音波形データに、取得された前記音響特徴量データが示す音響特徴  
量を合成させ、

20

合成することにより生成される和音の推論歌声データを、ユーザが歌わなくても出力さ  
せる、

電子楽器の制御方法。

【請求項 7】

電子楽器のコンピュータに、  
設定された音色に応じた楽器音波形データであって、ユーザによる和音の演奏操作に応  
じて取得された複数の音高データそれぞれに応じた楽器音波形データを、前記和音の演奏  
操作に応じて取得させ、

或る歌い手の歌声を学習した学習済みモデルに、演奏前からメモリに記憶されている歌  
詞データを入力することにより、前記学習済みモデルが出力する音響特徴量データを、前  
記和音の演奏操作に応じて取得させ、

30

取得された前記楽器音波形データに、取得された前記音響特徴量データが示す音響特徴  
量を合成させ、

合成することにより生成される和音の推論歌声データを、ユーザが歌わなくても出力さ  
せる、

プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

40

本発明は、鍵盤等の操作子の操作に応じて歌声を再生する電子楽器、電子楽器の制御方法、及びプログラムに関する。

【背景技術】

【0002】

従来、録音された音声の素片を接続し加工する素片連結型の合成方式により音声合成された歌声を出力する電子楽器が知られている（例えば特許文献 1）。

【先行技術文献】

【特許文献】

【0003】

【特許文献 1】特開平 9 - 050287

50

## 【発明の概要】

## 【発明が解決しようとする課題】

## 【0004】

しかし、PCM(Pulse Code Modulation:パルス符号変調)方式の延長ともいえるこの方式は、開発時に長時間の録音作業が必要であり、また、録音された音声の素片同士を滑らかに繋ぐための複雑な計算処理や、自然な歌声にするための調整が必要であった。

## 【0005】

そこで、本発明の目的は、或る歌手の歌声を学習した学習済みモデルを搭載することにより、演奏者による各操作子の操作により指定された音高で、前記或る歌手が良好に歌う電子楽器を提供することにある。

10

## 【課題を解決するための手段】

## 【0006】

態様の一例の電子楽器では、設定された音色に応じた楽器音波形データであって、ユーザによる和音の演奏操作に応じて取得された複数の音高データそれぞれに応じた楽器音波形データを、前記和音の演奏操作に応じて取得し、或る歌い手の歌声を学習した学習済みモデルに、演奏前からメモリに記憶されている歌詞データを入力することにより、前記学習済みモデルが出力する音響特徴量データを、前記和音の演奏操作に応じて取得し、取得された前記楽器音波形データに、取得された前記音響特徴量データが示す音響特徴量を合成し、合成することにより生成される和音の推論歌声データを、ユーザが歌わなくても出力する。

20

## 【発明の効果】

## 【0007】

本発明によれば、或る歌手の歌声を学習した学習済みモデルを搭載することにより、演奏者による各操作子の操作により指定された音高で、前記或る歌手が良好に歌う電子楽器を提供できる。

## 【図面の簡単な説明】

## 【0008】

【図1】電子鍵盤楽器の一実施形態の外観例を示す図である。

30

【図2】電子鍵盤楽器の制御システムの一実施形態のハードウェア構成例を示すブロック図である。

【図3】音声学習部及び音声合成部の構成例を示すブロック図である。

【図4】統計的音声合成処理の第1の実施形態の説明図である。

【図5】統計的音声合成処理の第2の実施形態の説明図である。

【図6】本実施形態のデータ構成例を示す図である。

【図7】本実施形態における電子楽器の制御処理例を示すメインフローチャートである。

【図8】初期化処理、テンポ変更処理、及びソング開始処理の詳細例を示すフローチャートである。

【図9】スイッチ処理の詳細例を示すフローチャートである。

40

【図10】自動演奏割込み処理の詳細例を示すフローチャートである。

【図11】ソング再生処理の詳細例を示すフローチャートである。

## 【発明を実施するための形態】

## 【0009】

以下、本発明を実施するための形態について図面を参照しながら詳細に説明する。

## 【0010】

図1は、電子鍵盤楽器の一実施形態100の外観例を示す図である。電子鍵盤楽器100は、演奏操作子としての複数の鍵からなる鍵盤101と、音量の指定、ソング再生のテンポ設定、ソング再生開始、伴奏再生等の各種設定を指示する第1のスイッチパネル102と、ソングや伴奏の選曲や音色の選択等を行う第2のスイッチパネル103と、ソング

50

再生時の歌詞、楽譜や各種設定情報を表示するＬＣＤ１０４（Ｌｉｑｕｉｄ　Ｃｒｙｓｔａｌ　Ｄｉｓｐｌａｙ：液晶ディスプレイ）等を備える。また、電子鍵盤楽器１００は、特に図示しないが、演奏により生成された楽音を放音するスピーカを裏面部、側面部、又は背面部等に備える。

#### 【００１１】

図２は、図１の電子鍵盤楽器１００の制御システム２００の一実施形態のハードウェア構成例を示す図である。図２において、制御システム２００は、ＣＰＵ（中央演算処理装置）２０１、ＲＯＭ（リードオンリーメモリ）２０２、ＲＡＭ（ランダムアクセスメモリ）２０３、音源ＬＳＩ（大規模集積回路）２０４、音声合成ＬＳＩ２０５、図１の鍵盤１０１、第１のスイッチパネル１０２、及び第２のスイッチパネル１０３が接続されるキースキャナ２０６、及び図１のＬＣＤ１０４が接続されるＬＣＤコントローラ２０８が、それぞれシステムバス２０９に接続されている。また、ＣＰＵ２０１には、自動演奏のシーケンスを制御するためのタイマ２１０が接続される。更に、音源ＬＳＩ２０４及び音声合成ＬＳＩ２０５からそれぞれ出力される楽音出力データ２１８（楽器音波形データ）及び歌声音声出力データ２１７は、Ｄ／Ａコンバータ２１１、２１２によりそれぞれアナログ楽音出力信号及びアナログ歌声音声出力信号に変換される。アナログ楽音出力信号及びアナログ歌声音声出力信号は、ミキサ２１３で混合され、その混合信号がアンプ２１４で増幅された後に、特に図示しないスピーカ又は出力端子から出力される。また、音源ＬＳＩ２０４の出力は、音声合成ＬＳＩ２０５に入力される。音源ＬＳＩ２０４と音声合成ＬＳＩ２０５は、１つのＬＳＩに統合されてもよい。

#### 【００１２】

ＣＰＵ２０１は、ＲＡＭ２０３をワークメモリとして使用しながらＲＯＭ２０２に記憶された制御プログラムを実行することにより、図１の電子鍵盤楽器１００の制御動作を実行する。また、ＲＯＭ２０２は、上記制御プログラム及び各種固定データのほか、歌詞データ及び伴奏データを含む曲データを記憶する。

#### 【００１３】

ＣＰＵ２０１には、本実施形態で使用するタイマ２１０が実装されており、例えば電子鍵盤楽器１００における自動演奏の進行をカウントする。

#### 【００１４】

音源ＬＳＩ２０４は、ＣＰＵ２０１からの発音制御指示に従って、例えば特に図示しない波形ＲＯＭから楽音波形データを読み出し、Ｄ／Ａコンバータ２１１に出力する。音源ＬＳＩ２０４は、同時に最大２５６ボイスを発振させる能力を有する。

#### 【００１５】

音声合成ＬＳＩ２０５は、ＣＰＵ２０１から、歌詞のテキストデータと音高に関する情報を歌声データ２１５として与えられると、それに対応する歌声の音声データを合成し、Ｄ／Ａコンバータ２１２に出力する。

#### 【００１６】

なお、音源ＬＳＩ２０４から出力される所定の発音チャネル（複数チャネル可能）の楽音出力データは、発声音源用楽音出力データ２２０として、音声合成ＬＳＩ２０５に入力される。

#### 【００１７】

キースキャナ２０６は、図１の鍵盤１０１の押鍵／離鍵状態、第１のスイッチパネル１０２、及び第２のスイッチパネル１０３のスイッチ操作状態を定常的に走査し、ＣＰＵ２０１に割り込みを掛けて状態変化を伝える。

#### 【００１８】

ＬＣＤコントローラ６０９は、ＬＣＤ５０５の表示状態を制御するＩＣ（集積回路）である。

#### 【００１９】

図３は、本実施形態における音声合成部、音響効果付加部、及び音声学習部の構成例を示すブロック図である。ここで、音声合成部３０２及び音響効果付加部３２０は、図２の

10

20

30

40

50

音声合成 L S I 2 0 5 が実行する一機能として電子鍵盤楽器 1 0 0 に内蔵される。

【 0 0 2 0 】

音声合成部 3 0 2 は、後述するソング再生処理により図 1 の鍵盤 1 0 1 上の押鍵に基づいて図 2 のキースキャナ 2 0 6 を介して C P U 2 0 1 から指示される歌詞及び音高の情報を含む歌声データ 2 1 5 を入力することにより、歌声音声出力データ 3 2 1 を合成し出力する。このとき音声合成部 3 0 2 のプロセッサは、鍵盤 1 0 1 上の複数の鍵（操作子）のなかのいずれかの鍵への操作に応じて、音響モデル部 3 0 6 に設定されている学習済み音響モデルに対して、歌詞情報と、いずれかの鍵に対応付けられている音高情報を含む歌声データ 2 1 5 を入力するとともに、その入力に応じて音響モデル部 3 0 6 が出力したスペクトル情報 3 1 8 と音源 L S I 2 0 4 が出力する発声音源用楽音出力データ 2 2 0 とに基づいて、歌い手の歌声を推論した歌声音声出力データ 3 2 1（第 1 音源情報使用推論歌声データ）を出力する音源情報使用推論歌声データ出力処理を実行する。

10

【 0 0 2 1 】

音響効果付加部 3 2 0 は、更に効果の情報が含まれる歌声データ 2 1 5 を入力することにより、音声合成部 3 0 2 が出力する歌声音声出力データ 3 2 1 に対して、ビブラート効果、トレモロ効果、又はワウ効果などの音響効果を付加し、最終的な歌声音声出力データ 2 1 7（図 2 参照）を出力する。

【 0 0 2 2 】

音声学習部 3 0 1 は例えば、図 3 に示されるように、図 1 の電子鍵盤楽器 1 0 0 とは別に外部に存在するサーバコンピュータ 3 0 0 が実行する一機能として実装されてよい。或いは、図 3 には図示していないが、音声学習部 3 0 1 は、図 2 の音声合成 L S I 2 0 5 の処理能力に余裕があれば、音声合成 L S I 2 0 5 が実行する一機能として電子鍵盤楽器 1 0 0 に内蔵されてもよい。

20

【 0 0 2 3 】

図 2 の音声学習部 3 0 1 及び音声合成部 3 0 2 は、例えば下記非特許文献 1 に記載の「深層学習に基づく統計的音声合成」の技術に基づいて実装される。

【 0 0 2 4 】

（非特許文献 1）

橋本佳，高木信二「深層学習に基づく統計的音声合成」日本音響学会誌 7 3 巻 1 号（2 0 1 7），p p . 5 5 - 6 2

30

【 0 0 2 5 】

図 3 に示されるように例えば外部のサーバコンピュータ 3 0 0 が実行する機能である図 2 の音声学習部 3 0 1 は、学習用テキスト解析部 3 0 3 と学習用音響特徴量抽出部 3 0 4 とモデル学習部 3 0 5 とを含む。

【 0 0 2 6 】

音声学習部 3 0 1 において、学習用歌声音声データ 3 1 2 としては、例えば適当なジャンルの複数の歌唱曲を或る歌い手が歌った音声を録音したものが使用される。また、学習用歌声データ 3 1 1 としては、各歌唱曲の歌詞テキストが用意される。

【 0 0 2 7 】

学習用テキスト解析部 3 0 3 は、歌詞テキストを含む学習用歌声データ 3 1 1 を入力してそのデータを解析する。この結果、学習用テキスト解析部 3 0 3 は、学習用歌声データ 3 1 1 に対応する音素、音高等を表現する離散数値系列である学習用言語特徴量系列 3 1 3 を推定して出力する。

40

【 0 0 2 8 】

学習用音響特徴量抽出部 3 0 4 は、上記学習用歌声データ 3 1 1 の入力に合わせてその学習用歌声データ 3 1 1 に対応する歌詞テキストを或る歌い手が歌うことによりマイク等を介して集録された学習用歌声音声データ 3 1 2 を入力して分析する。この結果、学習用音響特徴量抽出部 3 0 4 は、学習用歌声音声データ 3 1 2 に対応する音声の特徴を表す学習用音響特徴量系列 3 1 4 を抽出して出力する。

【 0 0 2 9 】

50

モデル学習部 305 は、下記 (1) 式に従って、学習用言語特徴量系列 313 (これを

$l$

と置く)と、音響モデル(これを

$\lambda$

と置く)とから、学習用音響特徴量系列 314 (これを

$o$

10

と置く)が生成される確率(これを

$P(o | l, \lambda)$

と置く)を最大にするような音響モデル

$\hat{\lambda}$

20

を、機械学習により推定する。即ち、テキストである言語特徴量系列と音声である音響特徴量系列との関係が、音響モデルという統計モデルによって表現される。

【0030】

【数1】

$$\hat{\lambda} = \arg \max_{\lambda} P(o | l, \lambda) \quad (1)$$

【0031】

ここで、

30

$\arg \max$

は、その右側に記載される関数に関して最大値を与える、その下側に記載されている引数を算出する演算を示す。

【0032】

モデル学習部 305 は、(1) 式によって機械学習を行った結果算出される音響モデル

$\hat{\lambda}$

を表現するモデルパラメータを学習結果 315 として出力する。

40

【0033】

この学習結果 315 (モデルパラメータ) は例えば、図 3 に示されるように、図 1 の電子鍵盤楽器 100 の工場出荷時に、図 2 の電子鍵盤楽器 100 の制御システムの ROM 202 に記憶され、電子鍵盤楽器 100 のパワーオン時に、図 2 の ROM 202 から音声合成 LSI 205 内の後述する音響モデル部 306 にロードされてよい。或いは、学習結果 315 は例えば、図 3 に示されるように、演奏者が電子鍵盤楽器 100 の第 2 のスイッチパネル 103 を操作することにより、特に図示しないインターネットや USB (Universal Serial Bus) ケーブル等のネットワークからネットワークインタフェース 219 を介して、音声合成 LSI 205 内の後述する音響モデル部 306 にダウンロードされてもよい。

50

## 【 0 0 3 4 】

音声合成 L S I 2 0 5 が実行する機能である音声合成部 3 0 2 は、テキスト解析部 3 0 7 と音響モデル部 3 0 6 と発声モデル部 3 0 8 とを含む。音声合成部 3 0 2 は、歌詞テキストを含む歌声データ 2 1 5 に対応する歌声音声出力データ 3 2 1 を、音響モデル部 3 0 6 に設定された音響モデルという統計モデルを用いて予測することにより合成する、統計的音声合成処理を実行する。

## 【 0 0 3 5 】

テキスト解析部 3 0 7 は、自動演奏に合わせた演奏者の演奏の結果として、図 2 の C P U 2 0 1 より指定される歌詞の音素、音高等に関する情報を含む歌声データ 2 1 5 を入力し、そのデータを解析する。この結果、テキスト解析部 3 0 7 は、歌声データ 2 1 5 に対応する音素、品詞、単語等を表現する言語特徴量系列 3 1 6 を解析して出力する。

10

## 【 0 0 3 6 】

音響モデル部 3 0 6 は、言語特徴量系列 3 1 6 を入力することにより、それに対応する音響特徴量系列 3 1 7 を推定して出力する。即ち音響モデル部 3 0 6 は、下記 ( 2 ) 式に従って、テキスト解析部 3 0 7 から入力する言語特徴量系列 3 1 6 (これを再度

$$l$$

と置く)と、モデル学習部 3 0 5 での機械学習により学習結果 3 1 5 として設定された音響モデル

20

$$\hat{\lambda}$$

とに基づいて、音響特徴量系列 3 1 7 (これを再度

$$o$$

と置く)が生成される確率(これを

$$P(o | l, \hat{\lambda})$$

30

と置く)を最大にするような音響特徴量系列 3 1 7 の推定値

$$\hat{o}$$

を推定する。

## 【 0 0 3 7 】

## 【 数 2 】

$$\hat{o} = \arg \max_o P(o | l, \hat{\lambda}) \quad (2)$$

40

## 【 0 0 3 8 】

発声モデル部 3 0 8 は、音響特徴量系列 3 1 7 を入力することにより、C P U 2 0 1 より指定される歌詞テキストを含む歌声データ 2 1 5 に対応する歌声音声出力データ 3 2 1 を生成する。歌声音声出力データ 3 2 1 は、後述する音響効果付加部 3 2 0 にて音響効果が付加されることにより最終的な歌声音声出力データ 2 1 7 に変換され、図 2 の D / A コンバータ 2 1 2 からミキサ 2 1 3 及びアンプ 2 1 4 を介して出力され、特には図示しないスピーカから放音される。

## 【 0 0 3 9 】

50

学習用音響特徴量系列 3 1 4 や音響特徴量系列 3 1 7 で表される音響特徴量は、人間の声道をモデル化したスペクトル情報と、人間の声帯をモデル化した音源情報とを含む。スペクトル情報としては例えば、メルケプストラムや線スペクトル対 (Line Spectral Pairs: LSP) 等を採用できる。音源情報としては、人間の音声のピッチ周波数を示す基本周波数 (F0) 及びパワー値を採用できる。発声モデル部 3 0 8 は、合成フィルタ部 3 1 0 を含む。合成フィルタ部 3 1 0 には、図 2 の音源 L S I 2 0 4 の所定の発音チャネル (複数チャネル可能) の発声音源用楽音出力データ 2 2 0 の出力が入力される。合成フィルタ部 3 1 0 は、人間の声道をモデル化した部分であり、音響モデル部 3 0 6 から順次入力するスペクトル情報 3 1 8 の系列に基づいて声道をモデル化するデジタルフィルタを形成し、音源 L S I 2 0 4 から入力する所定の発音チャネル (複数チャネル可能) の発声音源用楽音出力データ 2 2 0 を励振源信号として、デジタル信号の歌声音声出力データ 3 2 1 を生成し出力する。音源 L S I 2 0 4 から入力する発声音源用楽音出力データ 2 2 0 は、所定の発音チャネル分のポリフォニックである。

#### 【0040】

上述のように、演奏者の鍵盤 1 0 1 (図 1) での演奏に基づいて音源 L S I 2 0 4 が生成、出力する発声音源用楽音出力データ 2 2 0 が、音響モデル部 3 0 6 から入力するスペクトル情報 3 1 8 に基づいて動作する合成フィルタ部 3 1 0 に入力して、合成フィルタ部 3 1 0 から歌声音声出力データ 3 2 1 が出力される。このようにして生成、出力される歌声音声出力データ 3 2 1 は、音源 L S I 2 0 4 で生成された楽器音を音源信号としている。このため、歌い手の歌声と比べると、忠実性は若干失われるが、音源 L S I 2 0 4 で設定された楽器音の雰囲気が良く残ると共に、歌い手の歌声の声質も良く残った歌声となり、効果的な歌声音声出力データ 3 2 1 を出力させることが可能となる。更に、ボコーダモードでは、ポリフォニック動作が可能であるため、複数の歌声がハモるような効果を奏することも可能となる。

#### 【0041】

なお、音源 L S I 2 0 4 は、例えば複数の所定の発音チャネルの出力を発声音源用楽音出力データ 2 2 0 として音声合成 L S I 2 0 5 に供給すると同時に、他のチャネルの出力を通常の楽音出力データ 2 1 8 として出力するように動作してもよい。これにより、伴奏音は通常の楽器音で発音させたり、メロディラインの楽器音を発音させると同時にそのメロディの歌声を音声合成 L S I 2 0 5 から発音させるというような動作も可能である。

#### 【0042】

なお、ボコーダモードにおいて合成フィルタ部 3 1 0 に入力される発声音源用楽音出力データ 2 2 0 は、どのような信号でもよいが、音源信号として性質上、倍音成分を多く含み、かつ長く持続する、例えばブラス音、ストリング音、オルガン音のような楽器音が好ましい。勿論、大きな効果を狙って、このような基準に全く従わないような楽器音、例えば動物の鳴き声のような楽器音が使用されても、非常におもしろい効果が得られる。具体的な実施例として、例えば愛犬の鳴き声をサンプリングして得られたデータを楽器音として合成フィルタ部 3 1 0 に入力する。そして、合成フィルタ部 3 1 0 及び音響効果付加部 3 2 0 から出力される歌声音声出力データ 2 1 7 に基づいてスピーカから発音させる。そうすると、まるで愛犬が歌詞を歌っているように聞こえるという非常におもしろい効果が得られる。

#### 【0043】

学習用歌声音声データ 3 1 2 に対するサンプリング周波数は、例えば 1 6 K H z (キロヘルツ) である。また、学習用音響特徴量系列 3 1 4 及び音響特徴量系列 3 1 7 に含まれるスペクトルパラメータとして、例えばメルケプストラム分析処理により得られるメルケプストラムパラメータが採用される場合、その更新フレーム周期は、例えば 5 m s e c (ミリ秒) である。更に、メルケプストラム分析処理の場合、分析窓長は 2 5 m s e c、窓関数はブラックマン窓、分析次数は 2 4 次である。

#### 【0044】

音声合成部 3 0 2 から出力される歌声音声出力データ 3 2 1 は更に、音声合成 L S I 2

10

20

30

40

50



05内の音響効果付加部320により、ビブラート効果、トレモロ効果、又はワウ効果などの音響効果が付加される。

【0045】

ビブラート効果とは、歌唱において音を伸ばすとき、音高の高さを所定の振れ幅（深さ）で周期的に振らす効果をいう。ビブラート効果を付加するための音響効果付加部320の構成としては、例えば下記特許文献2又は3に記載の技術を採用することができる。

<特許文献2>

特開平06-167976号公報

<特許文献3>

特開平07-199931号公報

10

【0046】

トレモロ効果とは、同一又は複数の音を小刻みに演奏する効果をいう。トレモロ効果を付加するための音響効果付加部320の構成としては、例えば下記特許文献4に記載の技術を採用することができる。

<特許文献4>

特開平07-028471号公報

【0047】

ワウ効果とは、バンドパスフィルタの利得がピークとなる周波数を動かすことによって、「ワウワウ」としゃべっているような効果を得るものである。ワウ効果を付加するための音響効果付加部320の構成としては、例えば下記特許文献5に記載の技術を採用することができる。

20

<特許文献5>

特開平05-006173号公報

【0048】

演奏者は、歌声音声を指示するための鍵盤101（図1）上の第1鍵（第1操作子）によって、歌声音声出力データ321の出力を継続させている状態（第1鍵を押している状態）で、鍵盤101上の第2鍵（第2操作子）を繰り返し連打する操作を行うと、音響効果付加部320において、ビブラート効果、トレモロ効果、又はワウ効果のうち、第1のスイッチパネル102（図1）で予め選択した音響効果を付加させることができる。

【0049】

30

この場合更に、演奏者は、歌声指定している第1鍵の音高に対して、連打する第2鍵を、第2鍵と第1鍵の音高差が所望の音高差となるように指定することにより、音響効果付加部320における音高効果の程度を可変させることができる。例えば、第2鍵と第1鍵の間の音高差が1オクターブ離れていれば、音響効果の深さ（デプス）の最大値が設定され、音高差が小さくなるにつれて音響効果の程度が弱くなるように可変させることができる。

【0050】

なお、連打する鍵盤101上の第2鍵は、白鍵でもよいが、例えば黒鍵とした場合には、歌声音声の音高を指定するための第1鍵の演奏操作を邪魔しにくくて済む。

【0051】

40

このように、本実施形態では、音声合成部302から出力される歌声音声出力データ321に対して、音響効果付加部320にて更に多彩な音響効果を付加させて、最終的な歌声音声出力データ217を生成することが可能となる。

なお、設定された時間（たとえば、数百ミリ秒）、第2鍵に対する押鍵を検出しない場合に、音響効果の付加を終了する。

【0052】

別の実施例として、第1鍵が押されている状態で第2鍵が1度押されただけで、すなわち上述のように第2鍵が連打されていなくても、このような音響効果を付加してもよい。この場合も、第1鍵と第2鍵との間の音高差に応じて、このような音響効果の深さを変更してもよい。また、第2鍵が押されている間、音響効果を付加させて、第2鍵に対する離

50

鍵の検出に応じて、音響効果の付加を終了してもよい。

【 0 0 5 3 】

また別の実施例としては、第 1 鍵が押されている状態で第 2 鍵が押された後、第 1 鍵が離鍵されても、このような音響効果を付加してもよい。また、第 1 鍵及び第 2 鍵を連打する「トリル」を検出することにより、このような音高効果を付加してもよい。

【 0 0 5 4 】

なお、本明細書においては、これらの音響効果を付加する演奏法のことを、便宜上、「いわゆるレガート奏法」と言う場合がある。

【 0 0 5 5 】

次に、図 3 の音声学習部 3 0 1 及び音声合成部 3 0 2 からなる統計的音声合成処理の第 1 の実施形態について説明する。統計的音声合成処理の第 1 の実施形態では、音響モデル部 3 0 6 に設定される学習結果 3 1 5 (モデルパラメータ)によって表現される音響モデルとして、前述した非特許文献 1、及び下記非特許文献 2 に記載の HMM (Hidden Markov Model : 隠れマルコフモデル)を用いる。

【 0 0 5 6 】

(非特許文献 2)

酒向慎司、才野慶二郎、南角吉彦、徳田恵一、北村正「声質と歌唱スタイルを自動学習可能な歌声合成システム」情報処理学会研究報告音楽情報科学 (MUS) 2 0 0 8 ( 1 2 ( 2 0 0 8 - MUS - 0 7 4 ) ) , p p . 3 9 - 4 4 , 2 0 0 8 - 0 2 - 0 8

【 0 0 5 7 】

統計的音声合成処理の第 1 の実施形態では、ユーザが或るメロディーにそった歌詞を発声する際、声帯の振動や声道特性の歌声の特徴パラメータがどのような時間変化をしながら発声されるかが、HMM 音響モデルによって学習される。より具体的には、HMM 音響モデルは、学習用の歌声データから求めたスペクトル、基本周波数、およびそれらの時間構造を音素単位でモデル化したものである。

【 0 0 5 8 】

まず、HMM 音響モデルが採用される図 3 の音声学習部 3 0 1 の処理について説明する。音声学習部 3 0 1 内のモデル学習部 3 0 5 は、学習用テキスト解析部 3 0 3 が出力する学習用言語特徴量系列 3 1 3 と、学習用音響特徴量抽出部 3 0 4 が出力する上記学習用音響特徴量系列 3 1 4 とを入力することにより、前述した ( 1 ) 式に基づいて、尤度が最大となる HMM 音響モデルの学習を行う。HMM 音響モデルの尤度関数は下記 ( 3 ) 式で示される。

【 0 0 5 9 】

【数 3】

$$\begin{aligned}
 P(o | l, \lambda) &= \sum_q P(o | q, \lambda) P(q | l, \lambda) \\
 &= \sum_q \prod_{t=1}^T P(o_t | q_t, \lambda) P(q_t | q_{t-1}, l, \lambda) \\
 &= \sum_q \prod_{t=1}^T \mathcal{N}(o_t | \mu_{q_t}, \Sigma_{q_t}) a_{q_{t-1} q_t}
 \end{aligned} \tag{3}$$

【 0 0 6 0 】

ここで、

$o_t$

はフレーム t における音響特徴量、T はフレーム数、

$$\mathbf{q} = (q_1, \dots, q_T)$$

は H M M 音響モデルの状態系列、

$$q_t$$

はフレーム t における H M M 音響モデルの状態番号を表す。また、

$$a_{q_{t-1}q_t}$$

10

は状態

$$q_{t-1}$$

から状態

$$q_t$$

への状態遷移確率を表し、

$$\mathcal{N}(\mathbf{o}_t \mid \boldsymbol{\mu}_{q_t}, \boldsymbol{\Sigma}_{q_t})$$

20

は平均ベクトル

$$\boldsymbol{\mu}_{q_t}$$

、共分散行列

$$\boldsymbol{\Sigma}_{q_t}$$

の正規分布であり、状態

$$q_t$$

30

の出力確率分布を表す。尤度最大化基準による H M M 音響モデルの学習は、期待値最大化 ( Expectation Maximization : E M ) アルゴリズムを利用することで効率良く行われる。

【 0 0 6 1 】

歌声音声のスペクトルパラメータは、連続 H M M によってモデル化することができる。一方、対数基本周波数 ( F 0 ) は有声区間では連続値をとり、無声区間では値を持たない可変次元の時間系列信号であるため、通常の連続 H M M や離散 H M M で直接モデル化することはできない。そこで、可変次元に対応した多空間上の確率分布に基づく H M M である M S D - H M M ( M u l t i - S p a c e p r o b a b i l i t y D i s t r i b u t i o n H M M ) を用い、スペクトルパラメータとしてメルケプストラムを多次元ガウス分布、対数基本周波数 ( F 0 ) の有声音を1次元空間、無声音を0次元空間のガウス分布として同時にモデル化する。

40

【 0 0 6 2 】

また、歌声を構成する音素の特徴は、音響的な特徴は同一の音素であっても、様々な要因の影響を受けて変動することが知られている。例えば、基本的な音韻単位である音素のスペクトルや対数基本周波数 ( F 0 ) は、歌唱スタイルやテンポ、或いは、前後の歌詞や音高等によって異なる。このような音響特徴量に影響を与える要因のことをコンテキストと呼ぶ。第 1 の実施形態の統計的音声合成処理では、音声の音響的な特徴を精度良くモデ

50

ル化するために、コンテキストを考慮したHMM音響モデル（コンテキスト依存モデル）を採用することができる。具体的には、学習用テキスト解析部303は、フレーム毎の音素、音高だけでなく、直前、直後の音素や、現在位置、直前、直後のビブラートやアクセント等も考慮した学習用言語特徴量系列313を出力してよい。更に、コンテキストの組合せの効率化のために、決定木に基づくコンテキストクラスタリングが用いられてよい。これは、二分木を用いてHMM音響モデルの集合を木構造に分割することで、類似したコンテキストの組合せごとにHMM音響モデルをクラスタリングする手法である。木の各ノードには、「直前の音素は/a/であるか？」等のコンテキストを二分する質問があり、各リーフノードには、特定のHMM音響モデルに相当する学習結果315（モデルパラメータ）がある。任意のコンテキストの組合せは、ノードにある質問に沿って木を辿ること  
10  
で、何れかのリーフノードに到達でき、そのリーフノードに対応する学習結果315（モデルパラメータ）を選択できる。適切な決定木構造を選択することで、高精度かつ汎化性能の高いHMM音響モデル（コンテキスト依存モデル）を推定することができる。

#### 【0063】

図4は、統計的音声合成処理の第1の実施形態における、HMMの決定木についての説明図である。コンテキストに依存するそれぞれの音素について、その音素の各状態は、例えば図4(a)に示される#1、#2、#3の3つの状態401からなるHMMに対応付けられる。各状態に対して入出力する矢印は、状態遷移を示している。例えば、状態401（#1）は例えば、その音素の開始付近をモデル化する状態である。また、状態401（#2）は例えば、その音素の中心付近をモデル化する状態である。更に、状態401（  
20  
#3）は例えば、その音素の終了付近をモデル化する状態である。

#### 【0064】

また、音素長に依存して、図4(a)のHMMが示す#1から#3の各状態401が継続する長さが、図4(b)の状態継続長モデルにより決定される。図3のモデル学習部305は、図3の学習用テキスト解析部303が図3の学習用歌声データ311から抽出した状態継続長に関する多数の音素のコンテキストに対応する学習用言語特徴量系列313から、状態継続長を決定するための状態継続長決定木402を学習により生成し、学習結果315として音声合成部302内の音響モデル部306に設定する。

#### 【0065】

また、図3のモデル学習部305は例えば、図3の学習用音響特徴量抽出部304が図3の学習用歌声音声データ312から抽出したメルケプストラムパラメータに関する多数の音素に対応する学習用音響特徴量系列314から、メルケプストラムパラメータを決定するためのメルケプストラムパラメータ決定木403を学習により生成し、学習結果315として音声合成部302内の音響モデル部306に設定する。  
30

#### 【0066】

更に、図3のモデル学習部305は例えば、図3の学習用音響特徴量抽出部304が図3の学習用歌声音声データ312から抽出した対数基本周波数(F0)に関する多数の音素に対応する学習用音響特徴量系列314から、対数基本周波数(F0)を決定するための対数基本周波数決定木404を学習により生成し、学習結果315として音声合成部302内の音響モデル部306に設定する。なお、前述したように、対数基本周波数(F0)の有声区間と無声区間はそれぞれ、可変次元に対応したMSD-HMMにより、1次元及び0次元のガウス分布としてモデル化され、対数基本周波数決定木404が生成される。  
40

#### 【0067】

その他、図3のモデル学習部305は、図3の学習用テキスト解析部303が図3の学習用歌声データ311から抽出した状態継続長に関する多数の音素のコンテキストに対応する学習用言語特徴量系列313から、音高のビブラートやアクセント等のコンテキストを決定するための決定木を学習により生成し、学習結果315として音声合成部302内の音響モデル部306に設定してもよい。

#### 【0068】

10

20

30

40

50

次に、HMM音響モデルが採用される図3の音声合成部302の処理について説明する。音響モデル部306は、テキスト解析部307が出力する歌詞の音素、音高、その他のコンテキストに関する言語特徴量系列316を入力することにより、コンテキスト毎に図4に例示される各決定木402、403、404等を参照してHMMを連結し、連結した各HMMから出力確立が最大となる音響特徴量系列317（スペクトル情報318と音源情報319）を予測する。

【0069】

このとき、音響モデル部306は、前述した(2)式に従って、テキスト解析部307から入力する言語特徴量系列316(=

$l$

10

)と、モデル学習部305での機械学習により学習結果315として設定された音響モデル(=

$\hat{\lambda}$

)とに基づいて、音響特徴量系列317(=

$o$

20

)が生成される確率(=

$P(o | l, \hat{\lambda})$

)を最大にするような音響特徴量系列317の推定値(=

$\hat{o}$

)を推定する。ここで、前述した(2)式は、図4(b)の状態継続長モデルによって推定される状態系列

30

$\hat{q} = \arg \max_q P(q | l, \hat{\lambda})$

を用いることで、下記(4)式により近似される。

【0070】

【数4】

$\hat{o} = \arg \max_o \sum_q P(o | q, \hat{\lambda}) P(q | l, \hat{\lambda})$

40

$\approx \arg \max_o P(o | \hat{q}, \hat{\lambda})$

$= \arg \max_o \mathcal{N}(o | \mu_{\hat{q}}, \Sigma_{\hat{q}})$

(4)

$= \mu_{\hat{q}}$

【0071】

ここで、

50

$$\mu_{\hat{q}} = [\mu_{\hat{q}_1}^\top, \dots, \mu_{\hat{q}_T}^\top]^\top$$

$$\Sigma_{\hat{q}} = \text{diag}[\Sigma_{\hat{q}_1}, \dots, \Sigma_{\hat{q}_T}]$$

であり、

$$\mu_{\hat{q}_t}$$

10

と

$$\Sigma_{\hat{q}_t}$$

はそれぞれ状態

$$\hat{q}_t$$

における平均ベクトルと共分散行列である。言語特徴量系列

$$l$$

20

を用い、音響モデル部 3 0 6 に設定された各決定木を辿ることにより、平均ベクトルと共分散行列が算出される。(4)式より、音響特徴量系列 3 1 7 の推定値(=

$$\hat{o}$$

)は平均ベクトル(=

$$\mu_{\hat{q}}$$

30

)によって得られるが、

$$\mu_{\hat{q}}$$

は状態が遷移する部分で階段状に変化する不連続な系列となる。このような不連続な音響特徴量系列 3 1 7 から合成フィルタ部 3 1 0 が歌声音声出力データ 3 2 1 を合成した場合、自然性という観点からは低品質な合成音声になってしまう。そこで、統計的音声合成処理の第 1 の実施形態では、モデル学習部 3 0 5 において動的特徴量を考慮した学習結果 3 1 5 (モデルパラメータ)の生成アルゴリズムが採用されてよい。静的特徴量

$$c_t$$

40

と動的特徴量

$$\Delta c_t$$

からフレーム t における音響特徴量系列(=

$$o_t = [c_t^\top, \Delta c_t^\top]^\top$$

50

）が構成されるとき、各時刻の音響特徴量系列（＝

$$\boldsymbol{o} = [\boldsymbol{o}_1^\top, \dots, \boldsymbol{o}_T^\top]^\top$$

）は、下記（５）式で示される。

【００７２】

【数５】

$$\boldsymbol{o} = \boldsymbol{W}\boldsymbol{c} \quad (5)$$

10

【００７３】

ここで、

$$\boldsymbol{W}$$

は静的特徴量系列

$$\boldsymbol{c} = [\boldsymbol{c}_1^\top, \dots, \boldsymbol{c}_T^\top]^\top$$

20

から動的特徴量を含む音響特徴量系列

$$\boldsymbol{o}$$

を求める行列である。モデル学習部３０５は、上述の（５）式を制約として前述の（４）式を、下記（６）式に示されるように解く。

【００７４】

【数６】

$$\hat{\boldsymbol{c}} = \arg \max_{\boldsymbol{c}} \mathcal{N}(\boldsymbol{W}\boldsymbol{c} \mid \boldsymbol{\mu}_{\hat{\boldsymbol{q}}}, \boldsymbol{\Sigma}_{\hat{\boldsymbol{q}}}) \quad (6)$$

30

【００７５】

ここで、

$$\hat{\boldsymbol{c}}$$

は、動的特徴量を制約としながら出力確率が最大となる静的特徴量系列である。動的特徴量を考慮することで状態境界の不連続が解決され、滑らかに変化する音響特徴量系列３１７を得ることができ、合成フィルタ部３１０において、高品質な歌声音声出力データ３２１を生成することが可能となる。

40

【００７６】

ここで、歌声データの音素境界は楽譜が定める音符の境界とは一致しないことが多い。このような時間的な揺らぎは、音楽表現という観点からは本質的なものといえる。そこで、上述のＨＭＭ音響モデルを採用した統計的音声合成処理の第１の実施形態において、歌声の発声において、発声する際の音韻の違いや音高やリズムなど、様々な影響を受けた時間の偏りがあると仮定し、学習データにおける発声のタイミングと楽譜とのずれをモデル化する技術が採用されてもよい。具体的には、音符単位のずれモデルとして、音符単位に見た歌声と楽譜のずれを１次元ガウス分布によって表し、他のスペクトルパラメータや対数基本周波数（Ｆ０）等と同様にして、コンテキストに依存したＨＭＭ音響モデルとして扱ってよい。このような、「ずれ」のコンテキストを含むＨＭＭ音響モデルを用いた歌声

50

合成では、まず楽譜の表す時間境界を定めた後、音符単位のずれモデルと音素状態継続長モデルの両者の同時確率を最大化することで、学習データにおける音符の揺らぎを考慮した時間構造を決定することが可能となる。

#### 【0077】

次に、図3の音声学習部301及び音声合成部302からなる統計的音声合成処理の第2の実施形態について説明する。統計的音声合成処理の第1の実施形態では、言語特徴量系列316から音響特徴量系列317を予測するために、音響モデル部306がディープニューラルネットワーク(Deep Neural Network: DNN)により実装される。これに対応して、音声学習部301内のモデル学習部305は、言語特徴量から音響特徴量へのDNN内の各ニューロンの非線形変換関数を表すモデルパラメータを学習し、そのモデルパラメータを学習結果315として音声合成部302内の音響モデル部306のDNNに出力する。

10

#### 【0078】

通常、音響特徴量は例えば5.1ms(ミリ秒)幅のフレームを単位として算出され、言語特徴量は音素を単位として算出される。従って、音響特徴量と言語特徴量は時間単位が異なる。HMM音響モデルを採用した統計的音声合成処理の第1の実施形態では、音響特徴量と言語特徴量の対応はHMMの状態系列によって表現され、モデル学習部305は、音響特徴量と言語特徴量の対応関係を、図3の学習用歌声データ311及び学習用歌声音声データ312に基づき自動学習した。これに対して、DNNを採用した統計的音声合成処理の第2の実施形態では、音響モデル部306に設定されるDNNは、入力である言語特徴量系列316と出力である音響特徴量系列317の一対一の対応関係を表すモデルであるため、時間単位の異なる入出力データ対を用いてDNNを学習させることはできない。このため、統計的音声合成処理の第2の実施形態では、予めフレーム単位の音響特徴量系列と言語特徴量系列の対応関係が設定され、フレーム単位の音響特徴量と言語特徴量の対が生成される。

20

#### 【0079】

図5は、上述の対応関係を示す音声合成LSI205の動作説明図である。例えば、童謡「きらきら星」の歌い出しの歌詞文字列「き」「ら」「き」(図5(a))に対応する言語特徴量系列である歌声音素列「/k/」「/i/」「/r/」「/a/」「/k/」「/i/」(図5(b))が得られているときに、これらの言語特徴量系列が、フレーム単位の音響特徴量系列(図5(c))に対して、1対多の関係(図5の(b)と(c)の関係)で対応付けられる。なお、言語特徴量は音響モデル部306におけるDNNへの入力として使用されるため、数値データとして表現する必要がある。このため、言語特徴量系列としては、「直前の音素は「/a/」であるか?」や「現在の単語に含まれる音素の数は?」等のコンテキストに関する質問に対する二値のデータ(0又は1)、或いは、連続値での回答を連結して得られる数値データが用意される。

30

#### 【0080】

統計的音声合成処理の第2の実施形態における図3の音声学習部301内のモデル学習部305は、図5の破線矢印群501として示されるように、フレーム単位で、図5(b)に対応する学習用言語特徴量系列313の音素列と図5(c)に対応する学習用音響特徴量系列314の対を音響モデル部306のDNNに順次与えて学習を行う。なお、音響モデル部306内のDNNは、図5のグレー色の丸印群として示されるように、入力層、1つ以上の中間層、及び出力層からなるニューロン群を含む。

40

#### 【0081】

一方、音声合成時には、上記フレーム単位で、図5(b)に対応する言語特徴量系列316の音素列が音響モデル部306のDNNに入力される。この結果、音響モデル部306のDNNは、図5の太実線矢印群502として示されるように、上記フレーム単位で、音響特徴量系列317を出力する。従って、発声モデル部308においても、上述のフレーム単位で、音響特徴量系列317に含まれる音源情報319及びスペクトル情報318がそれぞれ音源生成部309及び合成フィルタ部310に与えられて、音声合成が実行さ

50



れる。

【 0 0 8 2 】

この結果、発声モデル部 3 0 8 は、図 5 の太実線矢印群 5 0 3 として示されるように、フレーム毎に、例えば 2 2 5 サンプル ( s a m p l e s ) ずつの歌声音声出力データ 3 2 1 を出力する。フレームは 5 . 1 m s e c の時間幅を有するため、1 サンプルは「 5 . 1 m s e c ÷ 2 2 5 = 0 . 0 2 2 7 m s e c 」であり、従って、歌声音声出力データ 3 2 1 のサンプリング周波数は 1 / 0 . 0 2 2 7 = 4 4 k H z ( キロヘルツ ) である。

【 0 0 8 3 】

D N N の学習は、フレーム単位の音響特徴量と言語特徴量の対を用いて、下記の ( 7 ) 式で演算される二乗誤差最小化基準によって行われる。

10

【 0 0 8 4 】

【数 7】

$$\hat{\lambda} = \arg \min_{\lambda} \frac{1}{2} \sum_{t=1}^T \|o_t - g_{\lambda}(l_t)\|^2 \quad (7)$$

【 0 0 8 5 】

ここで、

20

$o_t$

と

$l_t$

はそれぞれ t 番目のフレーム t における音響特徴量と言語特徴量、

$\hat{\lambda}$

30

は音響モデル部 3 0 6 の D N N のモデルパラメータ、

$g_{\lambda}(\cdot)$

は D N N によって表される非線形変換関数である。D N N のモデルパラメータは誤差逆伝播法によって効率良く推定することができる。前述した ( 1 ) 式によって表される統計的音声合成におけるモデル学習部 3 0 5 の処理との対応関係を考慮すると、D N N の学習は下記の ( 8 ) 式のように表すことができる。

【 0 0 8 6 】

40

【数 8】

$$\begin{aligned} \hat{\lambda} &= \arg \max_{\lambda} P(o | l, \lambda) \\ &= \arg \max_{\lambda} \prod_{t=1}^T \mathcal{N}(o_t | \tilde{\mu}_t, \tilde{\Sigma}_t) \end{aligned} \quad (8)$$

【 0 0 8 7 】

ここで、下記 ( 9 ) 式が成立する。

50

【数 9】

$$\tilde{\mu}_t = g_{\lambda}(l_t) \quad (9)$$

【0088】

上記(8)式及び(9)式のように、音響特徴量と言語特徴量の関係は、DNNの出力を平均ベクトルとする正規分布

$$\mathcal{N}(o_t | \tilde{\mu}_t, \tilde{\Sigma}_t)$$

10

によって表すことができる。DNNを用いた統計的音声合成処理の第2の実施形態では、通常、言語特徴量系列

$$l_t$$

に非依存な共分散行列、即ち全てのフレームにおいて同じ共分散行列

$$\tilde{\Sigma}_g$$

が用いられる。また、共分散行列

20

$$\tilde{\Sigma}_g$$

を単位行列とすると、(8)式は(7)式と等価な学習処理を示している。

【0089】

図5で説明したように、音響モデル部306のDNNは、フレーム毎に独立に音響特徴量系列317を推定する。このため、得られる音響特徴量系列317には、合成音声の品質を低下させるような不連続が含まれる。そこで、本実施形態では例えば、統計的音声合成処理の第1の実施形態の場合と同様な、動的特徴量を用いたパラメータ生成アルゴリズムを利用することにより、合成音声の品質を改善することができる。

30

【0090】

図3から図5で説明した統計的音声合成処理を利用した図1及び図2の電子鍵盤楽器100の実施形態の動作について、以下に詳細に説明する。図6は、本実施形態において、図2のROM202からRAM203に読み込まれる曲データのデータ構成例を示す図である。このデータ構成例は、MIDI(Musical Instrument Digital Interface)用ファイルフォーマットの一種であるスタンダードMIDIファイルのフォーマットに準拠している。この曲データは、チャンクと呼ばれるデータブロックから構成される。具体的には、曲データは、ファイルの先頭にあるヘッダチャンクと、それに続く歌詞パート用の歌詞データが格納されるトラックチャンク1と、伴奏パート用の演奏データが格納されるトラックチャンク2とから構成される。

40

【0091】

ヘッダチャンクは、ChunkID、ChunkSize、FormatType、NumberOfTrack、及びTimeDivisionの4つの値からなる。ChunkIDは、ヘッダチャンクであることを示す"MTld"という半角4文字に対応する4バイトのアスキーコード「4D 54 68 64」(数字は16進数)である。ChunkSizeは、ヘッダチャンクにおいて、ChunkIDとChunkSizeを除く、FormatType、NumberOfTrack、及びTimeDivisionの部分のデータ長を示す4バイトデータであり、データ長は6バイト:「00 00 00 06」(数字は16進数)に固定されている。FormatTypeは、本実施形態の場合、複数トラックを使用するフォーマット1を意味する2バイトのデータ「00 01」(数字は16進数)である

50

。Number Of Trackは、本実施形態の場合、歌詞パートと伴奏パートに対応する2トラックを使用することを示す2バイトのデータ「00 02」（数字は16進数）である。Time Divisionは、4分音符あたりの分解能を示すタイムベース値を示すデータであり、本実施形態の場合、10進法で480を示す2バイトのデータ「01 E0」（数字は16進数）である。

#### 【0092】

トラックチャンク1、2はそれぞれ、Chunk ID、Chunk Sizeと、Delta Time\_\_1[i]及びEvent\_\_1[i]（トラックチャンク1／歌詞パートの場合）又はDelta Time\_\_2[i]及びEvent\_\_2[i]（トラックチャンク2／伴奏パートの場合）からなる演奏データ組（0 i L：トラックチャンク1／歌詞パートの場合、0 i M：トラックチャンク2／伴奏パートの場合）とからなる。Chunk IDは、トラックチャンクであることを示す"MTrk"という半角4文字に対応する4バイトのアスキーコード「4D 54 72 6B」（数字は16進数）である。Chunk Sizeは、各トラックチャンクにおいて、Chunk IDとChunk Sizeを除く部分のデータ長を示す4バイトデータである。

#### 【0093】

Delta Time\_\_1[i]は、その直前のEvent\_\_1[i-1]の実行時刻からの待ち時間（相対時間）を示す1～4バイトの可変長データである。同様に、Delta Time\_\_2[i]は、その直前のEvent\_\_2[i-1]の実行時刻からの待ち時間（相対時間）を示す1～4バイトの可変長データである。Event\_\_1[i]は、トラックチャンク1／歌詞パートにおいて、歌詞の発声タイミングと音高を指示するメタイベント（タイミング情報）である。Event\_\_2[i]は、トラックチャンク2／伴奏パートにおいて、ノートオン又はノートオフを指示するMIDIイベント、又は拍子を指示するメタイベント（タイミング情報）である。トラックチャンク1／歌詞パートに対して、各演奏データ組Delta Time\_\_1[i]及びEvent\_\_1[i]において、その直前のEvent\_\_1[i-1]の実行時刻からDelta Time\_\_1[i]だけ待った上でEvent\_\_1[i]が実行されることにより、歌詞の発声進行が実現される。一方、トラックチャンク2／伴奏パートに対して、各演奏データ組Delta Time\_\_2[i]及びEvent\_\_2[i]において、その直前のEvent\_\_2[i-1]の実行時刻からDelta Time\_\_2[i]だけ待った上でEvent\_\_2[i]が実行されることにより、自動伴奏の進行が実現される。

#### 【0094】

図7は、本実施形態における電子楽器の制御処理例を示すメインフローチャートである。この制御処理は例えば、図2のCPU201が、ROM202からRAM203にロードされた制御処理プログラムを実行する動作である。

#### 【0095】

CPU201は、まず初期化処理を実行した後（ステップS701）、ステップS702からS708の一連の処理を繰り返し実行する。

#### 【0096】

この繰り返し処理において、CPU201はまず、スイッチ処理を実行する（ステップS702）。ここでは、CPU201は、図2のキースキャナ206からの割込みに基づいて、図1の第1のスイッチパネル102又は第2のスイッチパネル103のスイッチ操作に対応する処理を実行する。

#### 【0097】

次に、CPU201は、図2のキー・スキャナ206からの割込みに基づいて図1の鍵盤101の何れかの鍵が操作されたか否かを判定して処理する鍵盤処理を実行する（ステップS703）。ここでは、CPU201は、演奏者による何れかの鍵の押鍵又は離鍵の操作に応じて、図2の音源LSI204に対して、発音開始又は発音停止を指示する楽音制御データ216を出力する。

#### 【0098】

10

20

30

40

50

次に、CPU 201は、図1のLCD 104に表示すべきデータを処理し、そのデータを、図2のLCDコントローラ208を介してLCD 104に表示する表示処理を実行する(ステップS704)。LCD 104に表示されるデータとしては、例えば演奏される歌声音声出力データ217に対応する歌詞とその歌詞に対応するメロディの楽譜や、各種設定情報がある。

【0099】

次に、CPU 201は、ソング再生処理を実行する(ステップS705)。この処理においては、CPU 201が、演奏者の演奏に基づいて図5で説明した制御処理を実行し、歌声データ215を生成して音声合成LSI 205に出力する。

【0100】

続いて、CPU 201は、音源処理を実行する(ステップS706)。音源処理において、CPU 201は、音源LSI 204における発音中の楽音のエンベロープ制御等の制御処理を実行する。

【0101】

続いて、CPU 201は、音声合成処理を実行する(ステップS707)。音声合成処理において、CPU 201は、音声合成LSI 205による音声合成の実行を制御する。

【0102】

最後にCPU 201は、演奏者が特には図示しないパワーオフスイッチを押してパワーオフしたか否かを判定する(ステップS708)。ステップS708の判定がNOならば、CPU 201は、ステップS702の処理に戻る。ステップS708の判定がYESならば、CPU 201は、図7のフローチャートで示される制御処理を終了し、電子鍵盤楽器100の電源を切る。

【0103】

図8(a)、(b)、及び(c)はそれぞれ、図7のステップS701の初期化処理、図7のステップS702のスイッチ処理における後述する図9のステップS902のテンポ変更処理、及び同じく図9のステップS906のソング開始処理の詳細例を示すフローチャートである。

【0104】

まず、図7のステップS701の初期化処理の詳細例を示す図8(a)において、CPU 201は、Tick Timeの初期化処理を実行する。本実施形態において、歌詞の進行及び自動伴奏は、Tick Timeという時間を単位として進行する。図6の曲データのヘッダチャンク内のTime Division値として指定されるタイムベース値は4分音符の分解能を示しており、この値が例えば480ならば、4分音符は480 Tick Timeの時間長を有する。また、図6の曲データのトラックチャンク内の待ち時間Delta Time\_\_1[i]値及びDelta Time\_\_2[i]値も、Tick Timeの時間単位によりカウントされる。ここで、1 Tick Timeが実際に何秒になるかは、曲データに対して指定されるテンポによって異なる。今、テンポ値をTempo[ビート/分]、上記タイムベース値をTime Divisionとすれば、Tick Timeの秒数は、次式により算出される。

【0105】

$$\text{Tick Time [秒]} = 60 / \text{Tempo} / \text{Time Division} \quad (10)$$

【0106】

そこで、図8(a)のフローチャートで例示される初期化処理において、CPU 201はまず、上記(10)式に対応する演算処理により、Tick Time[秒]を算出する(ステップS801)。なお、テンポ値Tempoは、初期状態では図2のROM 202に所定の値、例えば60[ビート/秒]が記憶されているとする。或いは、不揮発性メモリに、前回終了時のテンポ値が記憶されていてもよい。

【0107】

次に、CPU 201は、図2のタイマ210に対して、ステップS801で算出したTick Time[秒]によるタイマ割込みを設定する(ステップS802)。この結果、

タイマ 210 において上記 T i c k T i m e [ 秒 ] が経過する毎に、C P U 201 に対して歌詞進行及び自動伴奏のための割込み（以下「自動演奏割込み」と記載）が発生する。従って、この自動演奏割込みに基づいて C P U 201 で実行される自動演奏割込み処理（後述する図 10）では、1 T i c k T i m e 毎に歌詞進行及び自動伴奏を進行させる制御処理が実行されることになる。

【 0108 】

続いて、C P U 201 は、図 2 の R A M 203 の初期化等のその他初期化処理を実行する（ステップ S 803）。その後、C P U 201 は、図 8（a）のフローチャートで例示される図 7 のステップ S 701 の初期化処理を終了する。

【 0109 】

図 8（b）及び（c）のフローチャートについては、後述する。図 9 は、図 7 のステップ S 702 のスイッチ処理の詳細例を示すフローチャートである。

【 0110 】

C P U 201 はまず、図 1 の第 1 のスイッチパネル 102 内のテンポ変更スイッチにより歌詞進行及び自動伴奏のテンポが変更されたか否かを判定する（ステップ S 901）。その判定が Y E S ならば、C P U 201 は、テンポ変更処理を実行する（ステップ S 902）。この処理の詳細は、図 8（b）を用いて後述する。ステップ S 901 の判定が N O ならば、C P U 201 は、ステップ S 902 の処理はスキップする。

【 0111 】

次に、C P U 201 は、図 1 の第 2 のスイッチパネル 103 において何れかのソング曲が選曲されたか否かを判定する（ステップ S 903）。その判定が Y E S ならば、C P U 201 は、ソング曲読み込み処理を実行する（ステップ S 904）。この処理は、図 6 で説明したデータ構造を有する曲データを、図 2 の R O M 202 から R A M 203 に読み込む処理である。なお、ソング曲読み込み処理は、演奏中でなくても、演奏開始前でもよい。これ以降、図 6 に例示されるデータ構造内のトラックチャンク 1 又は 2 に対するデータアクセスは、R A M 203 に読み込まれた曲データに対して実行される。ステップ S 903 の判定が N O ならば、C P U 201 は、ステップ S 904 の処理はスキップする。

【 0112 】

続いて、C P U 201 は、図 1 の第 1 のスイッチパネル 102 においてソング開始スイッチが操作されたか否かを判定する（ステップ S 905）。その判定が Y E S ならば、C P U 201 は、ソング開始処理を実行する（ステップ S 906）。この処理の詳細は、図 8（c）を用いて後述する。ステップ S 905 の判定が N O ならば、C P U 201 は、ステップ S 906 の処理はスキップする。

【 0113 】

続いて、C P U 201 は、図 1 の第 1 のスイッチパネル 102 においてエフェクト選択スイッチが操作されたか否かを判定する（ステップ S 907）。その判定が Y E S ならば、C P U 201 は、エフェクト選択処理を実行する（ステップ S 908）。ここでは、前述したように、図 3 の音響効果付加部 320 が歌声音声出力データ 321 の発声音声に対して音響効果を付加するときに、ビブラート効果、トレモロ効果、又はワウ効果のうちの音響効果を付加するかを、第 1 のスイッチパネル 102 によって演奏者に選択させる。この選択の結果、C P U 201 は、上記音響効果のうち演奏者が選択した何れかを、音声合成 L S I 205 内の音響効果付加部 320 に設定する。ステップ S 907 の判定が N O ならば、C P U 201 は、ステップ S 908 の処理はスキップする。

【 0114 】

設定によっては、複数の効果を同時に付加するようにしてもよい。

【 0115 】

最後に、C P U 201 は、図 1 の第 1 のスイッチパネル 102 又は第 2 のスイッチパネル 103 においてその他のスイッチが操作されたか否かを判定し、各スイッチ操作に対応する処理を実行する（ステップ S 909）。この処理は、演奏者が図 2 又は図 3 の音源 L S I 204 から音声合成 L S I 205 内の発声モデル部 308 に供給される発声音源用楽

10

20

30

40

50

音出力データ220の楽器音として、少なくともプラス音、ストリング音、オルガン音、及び動物の鳴き声のいずれかを含む複数の楽器音のなかからプラス音、ストリング音、オルガン音、及び動物の鳴き声のうちのいずれかの楽器音を選択させる第2のスイッチパネル103上の音色選択スイッチ（選択操作子）に対する処理を含む。

#### 【0116】

その後、CPU201は、図9のフローチャートで例示される図7のステップS702のスイッチ処理を終了する。この処理は、例えば発声音源用楽音出力データ220の音色の選択や、発声音源用楽音出力データ220の所定の発音チャンネルの選択のスイッチ動作を含む。

#### 【0117】

図8(b)は、図9のステップS902のテンポ変更処理の詳細例を示すフローチャートである。前述したように、テンポ値が変更されるとTickTime[秒]も変更になる。図8(b)のフローチャートでは、CPU201は、このTickTime[秒]の変更に関する制御処理を実行する。

#### 【0118】

まず、CPU201は、図7のステップS701の初期化処理で実行された図8(a)のステップS801の場合と同様にして、前述した(10)式に対応する演算処理により、TickTime[秒]を算出する(ステップS811)。なお、テンポ値Tempoは、図1の第1のスイッチパネル102内のテンポ変更スイッチにより変更された後の値がRAM203等に記憶されているものとする。

#### 【0119】

次に、CPU201は、図7のステップS701の初期化処理で実行された図8(a)のステップS802の場合と同様にして、図2のタイマ210に対して、ステップS811で算出したTickTime[秒]によるタイマ割込みを設定する(ステップS812)。その後、CPU201は、図8(b)のフローチャートで例示される図9のステップS902のテンポ変更処理を終了する。

#### 【0120】

図8(c)は、図9のステップS906のソング開始処理の詳細例を示すフローチャートである。

#### 【0121】

まず、CPU201は、自動演奏の進行において、TickTimeを単位として、直前のイベントの発生時刻からの相対時間をカウントするためのRAM203上の変数DeltaT\_\_1(トラックチャンク1)及びDeltaT\_\_2(トラックチャンク2)の値を共に0に初期設定する。次に、CPU201は、図6に例示される曲データのトラックチャンク1内の演奏データ組DeltaTime\_\_1[i]及びEvent\_\_1[i](1 ≤ i ≤ L - 1)の夫々iの値を指定するためのRAM203上の変数AutoIndex\_\_1と、同じくトラックチャンク2内の演奏データ組DeltaTime\_\_2[i]及びEvent\_\_2[i](1 ≤ i ≤ M - 1)の夫々iを指定するためのRAM203上の変数AutoIndex\_\_2の各値を共に0に初期設定する(以上、ステップS821)。これにより、図6の例では、初期状態としてまず、トラックチャンク1内の先頭の演奏データ組DeltaTime\_\_1[0]とEvent\_\_1[0]、及びトラックチャンク2内の先頭の演奏データ組DeltaTime\_\_2[0]とEvent\_\_2[0]がそれぞれ参照される。

#### 【0122】

次に、CPU201は、現在のソング位置を指示するRAM203上の変数SongIndexの値を0に初期設定する(ステップS822)。

#### 【0123】

更に、CPU201は、歌詞及び伴奏の進行をするか(=1)しないか(=0)を示すRAM203上の変数SongStartの値を1(進行する)に初期設定する(ステップS823)。

10

20

30

40

50

## 【0124】

その後、CPU201は、演奏者が、図1の第1のスイッチパネル102により歌詞の再生に合わせて伴奏の再生を行う設定を行っているか否かを判定する（ステップS824）。

## 【0125】

ステップS824の判定がYESならば、CPU201は、RAM203上の変数Bansouの値を1（伴奏有り）に設定する（ステップS825）。逆に、ステップS824の判定がNOならば、CPU201は、変数Bansouの値を0（伴奏無し）に設定する（ステップS826）。ステップS825又はS826の処理の後、CPU201は、図8（c）のフローチャートで示される図9のステップS906のソング開始処理を終了する。

10

## 【0126】

図13は、図2のタイマ210においてTickTime[秒]毎に発生する割込み（図8（a）のステップS802又は図8（b）のステップS812を参照）に基づいて実行される自動演奏割込み処理の詳細例を示すフローチャートである。以下の処理は、図6に例示される曲データのトラックチャンク1及び2の演奏データ組に対して実行される。

## 【0127】

まず、CPU201は、トラックチャンク1に対応する一連の処理（ステップS1001からS1006）を実行する。始めにCPU201は、SongStart値が1であるか否か、即ち歌詞及び伴奏の進行が指示されているか否かを判定する（ステップS1001）。

20

## 【0128】

CPU201は、歌詞及び伴奏の進行が指示されていないと判定した（ステップS1001の判定がNOである）場合には、CPU201は、歌詞及び伴奏の進行は行わずに図10のフローチャートで例示される自動演奏割込み処理をそのまま終了する。

## 【0129】

CPU201は、歌詞及び伴奏の進行が指示されていると判定した（ステップS1001の判定がYESである）場合には、トラックチャンク1に関する前回のイベントの発生時刻からの相対時刻を示すDeltaT\_1値が、AutoIndex\_1値が示すこれから実行しようとする演奏データ組の待ち時間DeltaTime\_1[AutoIndex\_1]に一致したか否かを判定する（ステップS1002）。

30

## 【0130】

ステップS1002の判定がNOならば、CPU201は、トラックチャック1に関して、前回のイベントの発生時刻からの相対時刻を示すDeltaT\_1値を+1インクリメントさせて、今回の割込みに対応する1TickTime単位分だけ時刻を進行させる（ステップS1003）。その後、CPU201は、後述するステップS1007に移行する。

## 【0131】

ステップS1002の判定がYESになると、CPU201は、トラックチャック1に関して、AutoIndex\_1値が示す演奏データ組のイベントEvent[AutoIndex\_1]を実行する（ステップS1004）。このイベントは、歌詞データを含むソングイベントである。

40

## 【0132】

続いて、CPU201は、トラックチャンク1内の次に実行すべきソングイベントの位置を示すAutoIndex\_1値を、RAM203上の変数SongIndexに格納する（ステップS1004）。

## 【0133】

更に、CPU201は、トラックチャンク1内の演奏データ組を参照するためのAutoIndex\_1値を+1インクリメントする（ステップS1005）。

## 【0134】

50

また、CPU201は、トラックチャック1に関して今回参照したソングイベントの発生時刻からの相対時刻を示すDeltaT\_\_1値を0にリセットする(ステップS1006)。その後、CPU201は、ステップS1007の処理に移行する。

【0135】

次に、CPU201は、トラックチャック2に対応する一連の処理(ステップS1007からS1013)を実行する。始めにCPU201は、トラックチャック2に関する前回のイベントの発生時刻からの相対時刻を示すDeltaT\_\_2値が、AutoIndex\_\_2値が示すこれから実行しようとする演奏データ組の待ち時間DeltaTime\_\_2[AutoIndex\_\_2]に一致したか否かを判定する(ステップS1007)。

【0136】

ステップS1007の判定がNOならば、CPU201は、トラックチャック2に関して、前回のイベントの発生時刻からの相対時刻を示すDeltaT\_\_2値を+1インクリメントさせて、今回の割込みに対応する1TickTime単位分だけ時刻を進行させる(ステップS1008)。その後、CPU201は、図10のフローチャートで示される自動演奏割込み処理を終了する。

【0137】

ステップS1007の判定がYESならば、CPU201は、伴奏再生を指示するRAM203上の変数Bansouの値が1(伴奏有り)であるか否かを判定する(ステップS1009)(図8(c)のステップS824からS826を参照)。

【0138】

ステップS1009の判定がYESならば、CPU201は、AutoIndex\_\_2値が示すトラックチャック2に関する伴奏に関するイベントEvent\_\_2[AutoIndex\_\_2]を実行する(ステップS1010)。ここで実行されるイベントEvent\_\_2[AutoIndex\_\_2]が、例えばノートオンイベントであれば、そのノートオンイベントにより指定されるキーナンバー及びベロシティにより、図2の音源LSI204に対して伴奏用の楽音の発音命令が発行される。一方、イベントEvent\_\_2[AutoIndex\_\_2]が、例えばノートオフイベントであれば、そのノートオフイベントにより指定されるキーナンバー及びベロシティにより、図2の音源LSI204に対して発音中の伴奏用の楽音の消音命令が発行される。

【0139】

一方、ステップS1009の判定がNOならば、CPU201は、ステップS1010をスキップすることにより、今回の伴奏に関するイベントEvent\_\_2[AutoIndex\_\_2]は実行せずに、歌詞に同期した進行のために、次のステップS1011の処理に進んで、イベントを進める制御処理のみを実行する。

【0140】

ステップS1010の後又はステップS1009の判定がNOの場合に、CPU201は、トラックチャック2上の伴奏データのための演奏データ組を参照するためのAutoIndex\_\_2値を+1インクリメントする(ステップS1011)。

【0141】

また、CPU201は、トラックチャック2に関して今回実行したイベントの発生時刻からの相対時刻を示すDeltaT\_\_2値を0にリセットする(ステップS1012)。

【0142】

そして、CPU201は、AutoIndex\_\_2値が示す次に実行されるトラックチャック2上の演奏データ組の待ち時間DeltaTime\_\_2[AutoIndex\_\_2]が0であるか否か、即ち、今回のイベントと同時に実行されるイベントであるか否かを判定する(ステップS1013)。

【0143】

ステップS1013の判定がNOならば、CPU201は、図10のフローチャートで示される今回の自動演奏割込み処理を終了する。

【0144】

10

20

30

40

50



ステップS1013の判定がYESならば、CPU201は、ステップS1009に戻って、AutoIndex\_\_2値が示すトラックチャンク2上で次に実行される演奏データ組のイベントEvent\_\_2[AutoIndex\_\_2]に関する制御処理を繰り返す。CPU201は、今回同時に実行される回数分だけ、ステップS1009からS1013の処理を繰り返し実行する。以上の処理シーケンスは、例えば和音等のように複数のノートオンイベントが同時タイミングで発音されるような場合に実行される。

【0145】

図11は、図7のステップS705のソング再生処理の詳細例を示すフローチャートである。

【0146】

まずCPU201は、図10の自動演奏割込み処理におけるステップS1004で、RAM203上の変数SongIndexに、値がセットされてNull値でなくなっているか否かを判定する(ステップS1101)。このSongIndex値は、現在のタイミングが歌声の再生タイミングになっているか否かを示すものである。

【0147】

ステップS1101の判定がYESになった、即ち現時点がソング再生のタイミングになったら、CPU201は、図7のステップS703の鍵盤処理により演奏者による図1の鍵盤101上で新たな押鍵が検出されているか否かを判定する(ステップS1102)。

【0148】

ステップS1102の判定がYESならば、CPU201は、演奏者による押鍵により指定された音高を、発声音高として特には図示しないレジスタ又はRAM203上の変数にセットする(ステップS1103)。

【0149】

次に、CPU201は、ステップS1103で設定された押鍵に基づく音高がセットされた発声音高で図9のステップS909で予め設定されている楽音の音色と所定の発音チャンネルで楽音を発音させるためのノートオンデータを生成し、音源LSI204に対して楽音の発音処理を指示する(ステップS1105)。音源LSI204は、CPU201から指定された所定の音色の所定の発音チャンネル分の楽音信号を生成、発声音源用楽音出力データ220として、音声合成LSI205内のボコーダモードスイッチ320を介して合成フィルタ部310に入力させる。

【0150】

続いて、CPU201は、RAM203上の変数SongIndexが示すRAM203上の曲データのトラックチャンク1上のソングイベントEvent\_\_1[SongIndex]から、歌詞文字列を読み出す。CPU201は、読み出した歌詞文字列に対応する歌声音声出力データ321を、ステップS1103で設定された押鍵に基づく音高がセットされた発声音高で発音させるための歌声データ215を生成し、音声合成LSI205に対して発声処理を指示する(ステップS1105)。音声合成LSI205は、図3から図5を用いて説明した統計的音声合成処理の第1の実施形態又は第2の実施形態を実行することにより、RAM203から曲データとして指定される歌詞を、演奏者が鍵盤101上で押鍵した鍵の音高にリアルタイムに対応して歌う歌声音声出力データ321を合成して出力する。

【0151】

この結果、演奏者の鍵盤101(図1)での演奏に基づいて音源LSI204が生成、出力する発声音源用楽音出力データ220が、音響モデル部306から入力するスペクトル情報318に基づいて動作する合成フィルタ部310に入力して、ポリフォニック動作で合成フィルタ部310から歌声音声出力データ321が出力される。

【0152】

一方、ステップS1101の判定により現時点がソング再生のタイミングになったと判定されると共に、ステップS1102の判定がNO、即ち現時点で新規押鍵が検出されて

10

20

30

40

50

いないと判定された場合には、CPU 201は、RAM 203上の変数Song Indexが示すRAM 203上の曲データのトラックチャンク1上のソングイベントEvent\_\_1 [Song Index]から音高のデータを読み出し、この音高を発声音高として特には図示しないレジスタ又はRAM 203上の変数にセットする(ステップS 1104)。

#### 【0153】

その後、CPU 201は、前述したステップS 1105以降の処理を実行することにより、音声合成LSI 205に対して歌声音声出力データ321の発声処理を指示する(ステップS 1105、S 1106)。音声合成LSI 205は、図3から図5を用いて説明した統計的音声合成処理の第1の実施形態又は第2の実施形態を実行することにより、演奏者が鍵盤101上でいずれの鍵も押鍵していなくても、RAM 203から曲データとして指定される歌詞を、同じく曲データとしてデフォルト指定されている音高に対応して歌う歌声音声出力データ321を合成して出力する。

10

#### 【0154】

ステップS 1105の処理の後、CPU 201は、RAM 203上の変数Song Indexが示す再生を行ったソング位置を、RAM 203上の変数Song Index\_\_preに記憶させる(ステップS 1107)。

#### 【0155】

更に、CPU 201は、変数Song Indexの値をNull値にクリアして、これ以降のタイミングをソング再生のタイミングでない状態にする(ステップS 1108)。その後、CPU 201は、図11のフローチャートで示される図7のステップS 705のソング再生処理を終了する。

20

#### 【0156】

前述したステップS 1101の判定がNOである、即ち現時点がソング再生のタイミングではないときには、CPU 201は、図7のステップS 703の鍵盤処理により演奏者による図1の鍵盤101上で効果付加のための「いわゆるレガート奏法」が検出されているか否かを判定する(ステップS 1109)。前述したように、このレガート奏法は、例えば、ステップS 1102でソング再生のための第1鍵が押鍵されている状態で、他の第2鍵を繰り返し連打する奏法である。この場合、CPU 201は、ステップS 11080において、第2鍵の押鍵を検出したときに、その押鍵の繰返し速度が所定の速度以上であるときに、レガート奏法が実行されていると判定する。

30

#### 【0157】

ステップS 1109の判定がNOならば、CPU 201はそのまま、図11のフローチャートで示される図7のステップS 705のソング再生処理を終了する。

#### 【0158】

ステップS 1109の判定がYESの場合には、CPU 201は、ステップS 1103で設定されている発声音高と、「いわゆるレガート奏法」により図1の鍵盤101上で繰り返し連打されている鍵の音高との音高差を計算する(ステップS 1110)。

#### 【0159】

続いて、CPU 201は、ステップS 1109で計算した音高差に応じたエフェクト量を、図2の音声合成LSI 205内の音響効果付加部320(図3)に設定する(ステップS 1111)。この結果、音響効果付加部320は、音声合成部302内の合成フィルタ部310から出力される歌声音声出力データ321に対して、図9のステップS 908で選択された音響効果の付加処理を上記エフェクト量で実行し、最終的な歌声音声出力データ217(図2、図3)を出力する。

40

#### 【0160】

以上のステップS 1110とステップS 1111の処理により、音声合成部302から出力される歌声音声出力データ321に対して、ビブラート効果、トレモロ効果、又はワウ効果などの音響効果を付加することが可能となり、多彩な歌声表現が実現される。

#### 【0161】

50

ステップS 1 1 0 9の処理の後、CPU 2 0 1は、図 1 1のフローチャートで示される図 7のステップS 7 0 5のソング再生処理を終了する。

【 0 1 6 2 】

図 3 及び図 4 を用いて説明したHMM音響モデルを採用した統計的音声合成処理の第 1 の実施形態では、特定の歌い手や歌唱スタイルなどの微妙な音楽表現を再現することが可能となり、接続歪みのない滑らかな歌声音質を実現することが可能となる。更に、学習結果 3 1 5 (モデルパラメータ)の変換により、別の歌い手への適応や、多様な声質や感情を表現することが可能となる。更に、HMM音響モデルにおける全てのモデルパラメータを、学習用歌声データ 3 1 1 及び学習用歌声音声データ 3 1 2 から自働学習できることにより、特定の歌い手の特徴をHMM音響モデルとして獲得し、合成時にそれらの特徴を再現するような歌声合成システムを自動的に構築することが可能となる。歌声の基本周波数や長さは楽譜のメロディやテンポに従うものであり、ピッチの時間変化やリズムの時間構造を楽譜から一意に定めることもできるが、そこから合成される歌声は単調で機械的なものになり、歌声としての魅力に欠けるものである。実際の歌声には、楽譜通りの画一化されたものだけでなく、声質のほかに声の高さやそれらの時間的な構造の変化により、それぞれの歌い手独自のスタイルが存在している。HMM音響モデルを採用する統計的音声合成処理の第 1 の実施形態では、歌声におけるスペクトル情報とピッチ情報の時系列変化をコンテキストに基づいてモデル化することができ、さらに楽譜情報を考慮することで、実際の歌声により近い歌声再生が可能となる。更に、統計的音声合成処理の第 1 の実施形態で採用されるHMM音響モデルは、あるメロディに沿った歌詞を発声する際、歌い手の声帯の振動や声道特性における歌声の音響特徴量系列がどのような時間変化をしながら発声されるか、という生成モデルに相当する。更に、統計的音声合成処理の第 1 の実施形態において、音符と歌声の「ずれ」のコンテキストを含むHMM音響モデルを用いることにより、歌い手の発声特性に依存して複雑に変化する傾向を有する歌唱法を正確に再現できる歌声音声の合成が実現される。このようなHMM音響モデルを採用する統計的音声合成処理の第 1 の実施形態の技術が、例えば電子鍵盤楽器 1 0 0 によるリアルタイム演奏の技術と融合されることにより、素片合成方式等による従来の電子楽器では不可能であった、モデルとなる歌い手の歌唱法及び声質を正確に反映させることのでき、まるでその歌い手が実際に歌っているような歌声の演奏を、電子鍵盤楽器 1 0 0 の鍵盤演奏等に合わせ、実現することが可能となる。

【 0 1 6 3 】

図 3 及び図 5 を用いて説明したDNN音響モデルを採用した統計的音声合成処理の第 2 の実施形態では、言語特徴量系列と音響特徴量系列の関係の表現として、統計的音声合成処理の第 1 の実施形態における決定木に基づくコンテキストに依存したHMM音響モデルが、DNNに置き換えられる。これにより、決定木では表現することが困難な複雑な非線形変換関数によって言語特徴量系列と音響特徴量系列の関係を表現することが可能となる。また、決定木に基づくコンテキストに依存したHMM音響モデルでは、決定木に基づいて対応する学習データも分類されるため、各コンテキストに依存したHMM音響モデルに割り当てられる学習データが減少してしまう。これに対し、DNN音響モデルでは学習データ全体から単一のDNNを学習するため、学習データを効率良く利用することが可能となる。このため、DNN音響モデルはHMM音響モデルよりも高精度に音響特徴量を予測することが可能となり、合成音声の自然性を大幅に改善することが可能となる。更に、DNN音響モデルでは、フレームに関する言語特徴量系列を利用可能することが可能となる。即ち、DNN音響モデルでは、予め音響特徴量系列と言語特徴量系列の時間的な対応関係が決められるため、HMM音響モデルでは考慮することが困難であった「現在の音素の継続フレーム数」、「現在のフレームの音素内位置」などのフレームに関する言語特徴量を利用することが可能となる。これにより、フレームに関する言語特徴量を用いることで、より詳細な特徴をモデル化することが可能となり、合成音声の自然性を改善することが可能となる。このようなDNN音響モデルを採用する統計的音声合成処理の第 2 の実施形態の技術が、例えば電子鍵盤楽器 1 0 0 によるリアルタイム演奏の技術と融合されること

により、鍵盤演奏等に基づく歌声の演奏を、モデルとなる歌い手の歌唱法及び声質に更に自然に近づけることが可能となる。

【0164】

以上説明した実施形態では、音声合成方式として統計的音声合成処理の技術を採用することにより、従来の素片合成方式に比較して格段に少ないメモリ容量を実現することが可能となる。例えば、素片合成方式の電子楽器では、音声素片データのために数百メガバイトに及ぶ記憶容量を有するメモリが必要であったが、本実施形態では、図3の学習結果315のモデルパラメータを記憶させるために、わずか数メガバイトの記憶容量を有するメモリのみで済む。このため、より低価格の電子楽器を実現することが可能となり、高音質の歌声演奏システムをより広いユーザ層に利用してもらうことが可能となる。

10

【0165】

更に、従来の素片データ方式では、素片データの入手による調整が必要なため、歌声演奏のためのデータの作成に膨大な時間（年単位）と労力を必要としていたが、本実施形態によるHMM音響モデル又はDNN音響モデルのための学習結果315のモデルパラメータの作成では、データの調整がほとんど必要ないため、数分の一の作成時間と労力で済む。これによっても、より低価格の電子楽器を実現することが可能となる。また、一般ユーザが、クラウドサービスとして利用可能なサーバコンピュータ300や或いは音声合成LSI205に内蔵された学習機能を使って、自分の声、家族の声、或いは有名人の声等を学習させ、それをモデル音声として電子楽器で歌声演奏させることも可能となる。この場合にも、従来よりも格段に自然で高音質な歌声演奏を、より低価格の電子楽器として実現

20

【0166】

本実施形態では特に、音源LSI204で生成された楽器音の発声音源用楽音出力データ220を音源信号とするため、音源LSI204で設定された楽器音の雰囲気が残ると共に、歌い手の歌声の声質も良く残った歌声となり、効果的な歌声音源出力データ217を出力させることが可能となる。更に、ポリフォニック動作が可能であるため、複数の歌声がハモるような効果を奏することも可能となる。これにより、演奏者に指定される各音高に基づいて学習された歌い手の歌声に応じた歌声で良好に歌う電子楽器を提供することが可能となる。

【0167】

30

以上説明した実施形態は、電子鍵盤楽器について本発明を実施したものであるが、本発明は電子弦楽器等の電子楽器にも適用することができる。

【0168】

また、図3の発声モデル部308として採用可能な音声合成方式は、ケプストラム音声合成方式には限定されず、LSP音声合成方式をはじめとして様々な音声合成方式を採用することが可能である。

【0169】

更に、以上説明した実施形態では、HMM音響モデルを用いた統計的音声合成処理の第1の実施形態又はDNN音響モデルを用いた遠後の第2の実施形態の音声合成方式について説明したが、本発明はこれに限られるものではなく、例えばHMMとDNNを組み合わせた音響モデル等、統計的音声合成処理を用いた技術であればどのような音声合成方式が採用されてもよい。

40

【0170】

以上説明した実施形態では、歌詞情報は曲データとして与えられたが、演奏者がリアルタイムに歌う内容を音声認識して得られるテキストデータが歌詞情報としてリアルタイムに与えられてもよい。

【0171】

以上の実施形態に関して、更に以下の付記を開示する。

（付記1）

各音高情報がそれぞれ対応付けられている複数の操作子と、

50

学習用歌詞情報及び学習用音高情報を含む学習用楽譜データと、歌い手の学習用歌声データと、を用いた機械学習処理により学習させた学習済み音響モデルであって、歌わせる歌詞情報と、音高情報と、を入力することにより、前記歌い手の声道をモデル化したスペクトル情報を出力する学習済み音響モデルを記憶しているメモリと、

プロセッサと、

を含み、

前記プロセッサは、前記複数の操作子のなかのいずれかの操作子への操作に応じて、前記学習済み音響モデルに対して、前記歌詞情報と、前記いずれかの操作子に対応付けられている音高情報と、を入力するとともに、前記入力に応じて前記学習済み音響モデルが出力した前記スペクトル情報と、前記いずれかの操作子に対応付けられている音高情報に応じた楽器音波形データと、に基づいて前記歌い手の歌声を推論した第1楽器音使用推論歌声データを出力する楽器音使用推論歌声データ出力処理、

を実行する電子楽器。

(付記2)

付記1に記載の電子楽器において、

少なくともプラス音、ストリング音、オルガン音、及び動物の鳴き声のいずれかを含む複数の楽器音のなかから前記プラス音、前記ストリング音、前記オルガン音、及び前記動物の鳴き声のうちのいずれかの楽器音を選択させる選択操作子、

を有し、

前記楽器音使用推論歌声データ出力処理は、前記選択操作子により選択された楽器音に応じた前記楽器音波形データに基づいて、前記第1楽器音使用推論歌声データを出力する。

(付記3)

付記1または付記2に記載の電子楽器において、

前記メモリは、メロディデータと、自動演奏させる伴奏データと、を有する曲データを記憶しており、

前記メロディデータは、各音高情報と、前記各音高情報に応じた音を出力する各タイミング情報と、前記各音高情報に対応付けられている各歌詞情報と、を含み、

前記プロセッサは、

前記伴奏データに基づいて発音部に発音させる伴奏データ自動演奏処理と、

前記学習済み音響モデルに対して、操作された前記いずれかの操作子に対応付けられている音高情報を入力する代わりに、前記メロディデータに含まれる音高情報を入力するとともに、前記メロディデータに含まれる歌詞情報を入力する入力処理と、(\*入力するタイミングは、演奏中でなくても、演奏開始前でもよいことを明細書中に含めてください)

を実行し、

前記楽器音使用推論歌声データ出力処理は、前記メロディデータに含まれるタイミング情報が示すタイミングに応じて前記複数の操作子のなかのいずれの操作子も演奏者に操作されなかった場合に、前記入力処理に応じて前記学習済み音響モデルから出力された前記スペクトル情報と、前記入力処理により入力された前記メロディデータに含まれる音高情報に応じた前記楽器音波形データと、に基づいて、前記歌い手の歌声を推論した第2楽器音使用推論歌声データを、前記メロディデータに含まれるタイミング情報が示すタイミングに合わせて出力する。

(付記4)

付記1から付記3のいずれかに記載の電子楽器において、

前記複数の操作子は、操作された前記いずれかの操作子としての第1操作子と、前記第1操作子からみて設定された条件を満たす第2操作子と、を含み、

前記楽器音使用推論歌声データ出力処理により前記第1楽器音使用推論歌声データが出力されている際に、前記第1操作子への操作が継続している状態で前記第2操作子が繰り返し操作された場合に、前記第1楽器音使用推論歌声データに対して少なくともビブラート、トレモロ及びワウワウのいずれかのエフェクトをかけるエフェクト処理、

を実行する。

(付記 5)

付記 4 に記載の電子楽器において、

前記エフェクト処理は、前記第 1 操作子に対応付けられている音高情報が示す第 1 音高と、前記第 2 操作子に対応付けられている音高情報が示す第 2 音高と、の音高差に応じて前記エフェクトの程度を変更している。

(付記 6)

付記 4 または 5 に記載の電子楽器において、

前記第 2 操作子は、黒鍵である。

(付記 7)

各音高情報がそれぞれ対応付けられている複数の操作子と、

学習用歌詞情報及び学習用音高情報を含む学習用楽譜データと、歌い手の学習用歌声データと、を用いた機械学習処理により学習させた学習済み音響モデルであって、歌わせる歌詞情報と、音高情報と、を入力することにより、前記歌い手の声道をモデル化したスペクトル情報を出力する学習済み音響モデルを記憶しているメモリと、

を含む電子楽器のコンピュータに、

前記複数の操作子のなかのいずれかの操作子への操作に応じて、前記学習済み音響モデルに対して、前記歌詞情報と、前記いずれかの操作子に対応付けられている音高情報と、を入力するとともに、前記入力に応じて前記学習済み音響モデルが出力した前記スペクトル情報と、前記いずれかの操作子に対応付けられている音高情報に応じた楽器音波形データと、に基づいて前記歌い手の歌声を推論した第 1 楽器音使用推論歌声データを出力する楽器音使用推論歌声データ出力処理、

を実行させる方法。

(付記 8)

学習用歌詞情報及び学習用音高情報を含む学習用楽譜データと、歌い手の学習用歌声データと、を用いた機械学習処理により学習させた学習済み音響モデルであって、歌わせる歌詞情報と、音高情報と、を入力することにより、前記歌い手の声道をモデル化したスペクトル情報を出力する学習済み音響モデルを記憶しているメモリと、

を含む電子楽器のコンピュータに、

前記複数の操作子のなかのいずれかの操作子への操作に応じて、前記学習済み音響モデルに対して、前記歌詞情報と、前記いずれかの操作子に対応付けられている音高情報と、を入力するとともに、前記入力に応じて前記学習済み音響モデルが出力した前記スペクトル情報と、前記いずれかの操作子に対応付けられている音高情報に応じた楽器音波形データと、に基づいて前記歌い手の歌声を推論した第 1 楽器音使用推論歌声データを出力する楽器音使用推論歌声データ出力処理、

を実行させるプログラム。

【符号の説明】

【 0 1 7 2 】

1 0 0 電子鍵盤楽器

1 0 1 鍵盤

1 0 2 第 1 のスイッチパネル

1 0 3 第 2 のスイッチパネル

1 0 4 L C D

2 0 0 制御システム

2 0 1 C P U

2 0 2 R O M

2 0 3 R A M

2 0 4 音源 L S I

2 0 5 音声合成 L S I

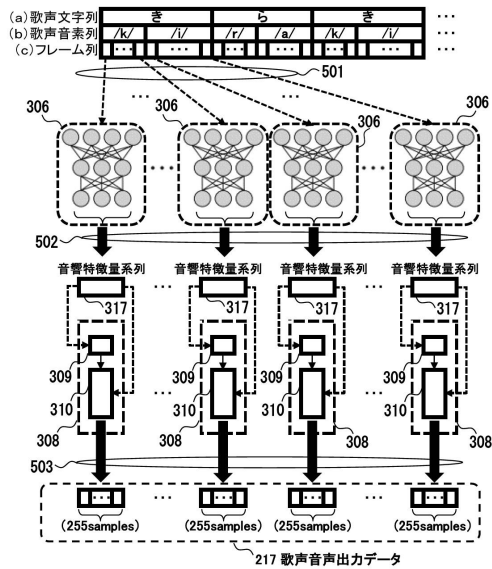
2 0 6 キースキャナ

2 0 8	L C D コントローラ	
2 0 9	システムバス	
2 1 0	タイマ	
2 1 1、2 1 2	D / A コンバータ	
2 1 3	ミキサ	
2 1 4	アンプ	
2 1 5	歌声データ	
2 1 6	発音制御データ	
2 1 7、3 2 1	歌声音声出力データ	
2 1 8	楽音出力データ	10
2 1 9	ネットワークインタフェース	
2 2 0	発声音源用楽音出力データ	
3 0 0	サーバコンピュータ	
3 0 1	音声学習部	
3 0 2	音声合成部	
3 0 3	学習用テキスト解析部	
3 0 4	学習用音響特徴量抽出	
3 0 5	モデル学習部	
3 0 6	音響モデル部	
3 0 7	テキスト解析部	20
3 0 8	発声モデル部	
3 0 9	音源生成部	
3 1 0	合成フィルタ部	
3 1 1	学習用歌声データ	
3 1 2	学習用歌声音声データ	
3 1 3	学習用言語特徴量系列	
3 1 4	学習用音響特徴量系列	
3 1 5	学習結果	
3 1 6	言語情報量系列	
3 1 7	音響特徴量系列	30
3 1 8	スペクトル情報	
3 1 9	音源情報	
3 2 0	音響効果付加部	





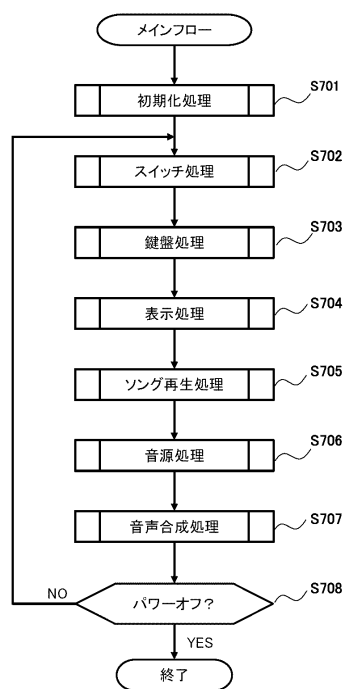
【 図 5 】



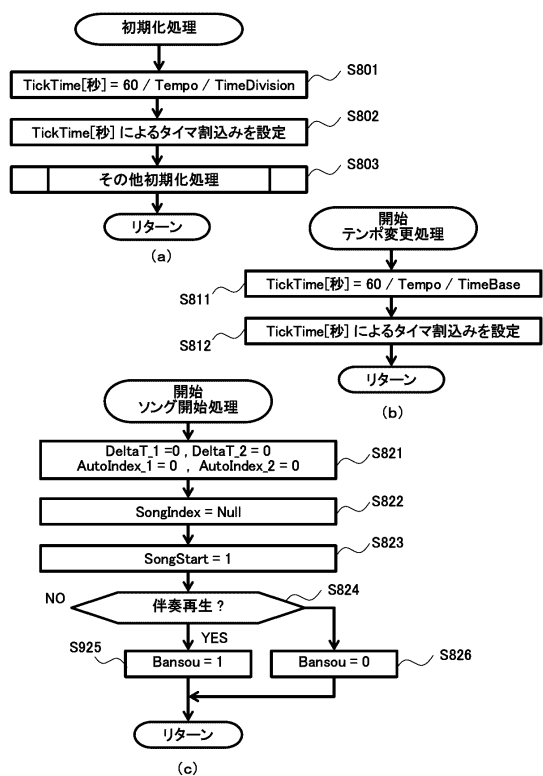
【 図 6 】



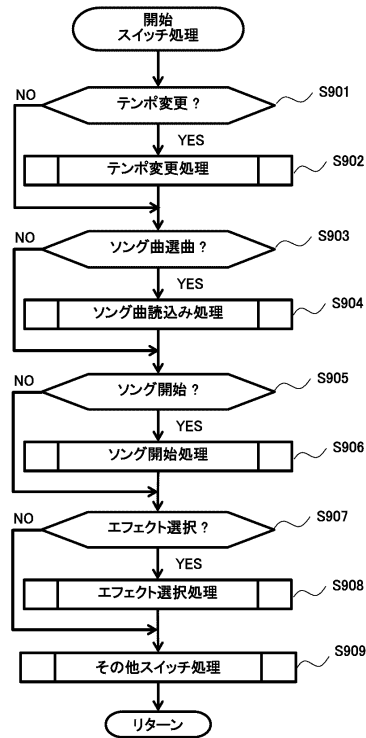
【圖 7】



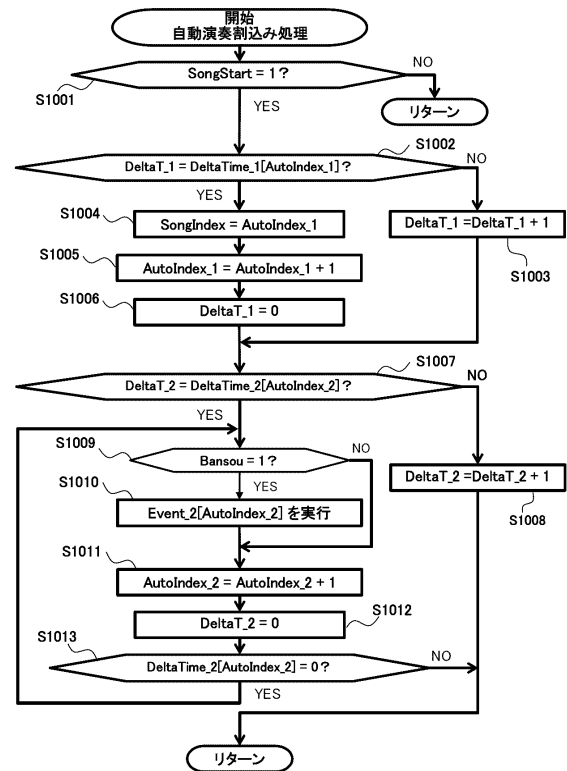
【 図 8 】



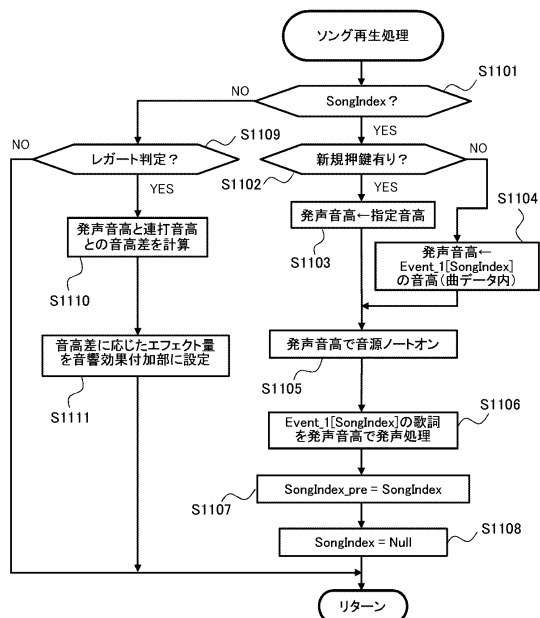
【図 9】



【図 10】



【図 11】



---

フロントページの続き

(72)発明者 瀬戸口 克

東京都羽村市栄町3丁目2番1号 カシオ計算機株式会社 羽村技術センター内

(72)発明者 中村 厚士

東京都羽村市栄町3丁目2番1号 カシオ計算機株式会社 羽村技術センター内

審査官 菊池 智紀

(56)参考文献 特開平4-349497(JP,A)

特開平5-341793(JP,A)

特開2010-139592(JP,A)

特開2011-13454(JP,A)

特開2017-107228(JP,A)

特開2006-146095(JP,A)

特開2005-84660(JP,A)

特開2004-287099(JP,A)

実開昭55-73999(JP,U)

酒向慎司 他, "声質と歌唱スタイルを自動学習可能な歌声合成システム", 情報処理学会研究報告, 2008年 2月 8日, Vol.2008, No.12, pp.39-44

(58)調査した分野(Int.Cl., DB名)

G10L 13/00 - 13/10

G10H 1/00 - 7/12