(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7276292号
(P7276292)

(45)発行日　令和5年5月18日(2023.5.18)

(24)登録日　令和5年5月10日(2023.5.10)

(51)国際特許分類		FI		
G10H	*1/053(2006.01)*	G10H	1/053	Z
G10L	*13/00 (2006.01)*	G10L	13/00	100Y
G10L	*13/033 (2013.01)*	G10L	13/033	102B

請求項の数　9　（全30頁）

(21)出願番号　特願2020-152926(P2020-152926)
(22)出願日　令和2年9月11日(2020.9.11)
(65)公開番号　特開2022-47167(P2022-47167A)
(43)公開日　令和4年3月24日(2022.3.24)
審査請求日　令和3年9月3日(2021.9.3)

(73)特許権者　000001443
カシオ計算機株式会社
東京都渋谷区本町１丁目６番２号
(74)代理人　100121083
弁理士　青木 宏義
(74)代理人　100138391
弁理士　天田 昌行
(74)代理人　100074099
弁理士　大菅 義之
(74)代理人　100182936
弁理士　矢野 直樹
(72)発明者　岩瀬 広
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　山下 剛史

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】

演奏時に指定される演奏時音高データを出力する音高指定部と、

前記演奏時に演奏によって音高が順次指定される時間間隔を示す演奏テンポデータを、前記演奏時の演奏形態を示す演奏時演奏形態データとして出力する演奏形態出力部と、

前記演奏時に、前記演奏時音高データ及び前記演奏時演奏形態データを学習済み音響モデルに入力することにより推論される音響モデルパラメータに基づいて、前記演奏時音高データ及び前記演奏時演奏形態データに対応する楽音データを合成し出力する発音モデル部と、

を備える電子楽器。

【請求項２】

演奏時の歌詞を示す演奏時歌詞データを出力する歌詞出力部と、

前記演奏時に前記歌詞の出力に合わせて指定される演奏時音高データを出力する音高指定部と、

前記演奏時に演奏によって音高が順次指定される時間間隔を示す演奏テンポデータを、前記演奏時の演奏形態を示す演奏時演奏形態データとして出力する演奏形態出力部と、

前記演奏時に、前記演奏時歌詞データ、前記演奏時音高データ、及び前記演奏時演奏形態データを学習済み音響モデルに入力することにより推論される音響モデルパラメータに基づいて、前記演奏時歌詞データ、前記演奏時音高データ、及び前記演奏時演奏形態データに対応する歌声音声データを合成し出力する発声モデル部と、

を備える電子楽器。
【請求項３】
前記楽音データは、前記演奏テンポデータの時間間隔に応じて発音される子音部の時間長が調整されている、請求項１に記載の電子楽器。
【請求項４】
前記歌声音声データは、前記演奏テンポデータの時間間隔に応じて発音される子音部の時間長が調整されている、請求項２に記載の電子楽器。
【請求項５】
前記演奏形態出力部は、順次得られる前記演奏テンポデータを演奏者に意図的に変更させる変更手段を含む、請求項１乃至請求項４のいずれかに記載の電子楽器。
【請求項６】
電子楽器のプロセッサに、
演奏時に指定される演奏時音高データを出力し、
前記演奏時に演奏によって音高が順次指定される時間間隔を示す演奏テンポデータを、前記演奏時の演奏形態を示す演奏時演奏形態データとして出力し、
前記演奏時に、前記演奏時音高データ及び前記演奏時演奏形態データを学習済み音響モデルに入力することにより推論される音響モデルパラメータに基づいて、前記演奏時音高データ及び前記演奏時演奏形態データに対応する楽音データを合成し出力する、
処理を実行させる電子楽器の制御方法。
【請求項７】
電子楽器のプロセッサに、
演奏時の歌詞を示す演奏時歌詞データを出力し、
前記演奏時に前記歌詞の出力に合わせて指定される演奏時音高データを出力し、
前記演奏時に演奏によって音高が順次指定される時間間隔を示す演奏テンポデータを、前記演奏時の演奏形態を示す演奏時演奏形態データとして出力し、
前記演奏時に、前記演奏時歌詞データ、前記演奏時音高データ、及び前記演奏時演奏形態データを学習済み音響モデルに入力することにより推論される音響モデルパラメータに基づいて、前記演奏時歌詞データ、前記演奏時音高データ、及び前記演奏時演奏形態データに対応する歌声音声データを合成し出力する、
処理を実行させる電子楽器の制御方法。
【請求項８】
電子楽器のプロセッサに、
演奏時に指定される演奏時音高データを出力し、
前記演奏時に演奏によって音高が順次指定される時間間隔を示す演奏テンポデータを、前記演奏時の演奏形態を示す演奏時演奏形態データとして出力し、
前記演奏時に、前記演奏時音高データ及び前記演奏時演奏形態データを学習済み音響モデルに入力することにより推論される音響モデルパラメータに基づいて、前記演奏時音高データ及び前記演奏時演奏形態データに対応する楽音データを合成し出力する、
処理を実行させるためのプログラム。
【請求項９】
電子楽器のプロセッサに、
演奏時の歌詞を示す演奏時歌詞データを出力し、
前記演奏時に前記歌詞の出力に合わせて指定される演奏時音高データを出力し、
前記演奏時に演奏によって音高が順次指定される時間間隔を示す演奏テンポデータを、前記演奏時の演奏形態を示す演奏時演奏形態データとして出力し、
前記演奏時に、前記演奏時歌詞データ、前記演奏時音高データ、及び前記演奏時演奏形態データを学習済み音響モデルに入力することにより推論される音響モデルパラメータに基づいて、前記演奏時歌詞データ、前記演奏時音高データ、及び前記演奏時演奏形態データに対応する歌声音声データを合成し出力する、
処理を実行させるためのプログラム。

【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、鍵盤等の操作子の操作に応じて学習済み音響モデルを駆動して音声を出力する電子楽器、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】
電子楽器において、従来のＰＣＭ（Ｐｕｌｓｅ　Ｃｏｄｅ　Ｍｏｄｕｌａｔｉｏｎ：パルス符号変調）方式の表現力の弱点である歌唱音声や生楽器の表現力を補うために、人間の発声機構やアコースティック楽器の発音機構をデジタル信号処理でモデル化した音響モデルを、歌唱動作や演奏動作に基づく機械学習により学習させ、その学習済み音響モデルを実際の演奏操作に基づいて駆動して歌声や楽音の音声波形データを推論して出力する技術が考案され実用化されつつある（例えば特許文献１）。
【先行技術文献】
【特許文献】
【０００３】
【文献】特許第６６１０７１４号公報
【発明の概要】
【発明が解決しようとする課題】
【０００４】
機械学習により例えば歌声波形や楽音波形を作り出す場合、演奏されるテンポやフレーズの歌い方や演奏形態の変化によって生成波形が変化することが多い。例えば、ボーカル音声の子音部の発音時間長、管楽器音のブロー音の発音時間長、擦弦楽器の弦をこすり始めるときのノイズ成分の時間長が、音符の少ないゆっくりとした演奏では長い時間になって表情豊かな生々しい音になり、音符が多いテンポの速い演奏では短い時間になって歯切れのよい音で演奏される。
【０００５】
しかし、ユーザが鍵盤等でリアルタイムに演奏する場合には、音源装置に各音符の譜割りの変化や演奏フレーズの違いに対応して変化する音符間の演奏速度を伝える手段がないため、音響モデルが音符間の演奏速度の変化に応じた適切な音声波形を推論することができず、例えば、ゆっくりとした演奏のときの表現力が不足したり、逆に、テンポの速い演奏に対して生成される音声波形の立上りが遅れて演奏しずらくなってしまう、といった問題があった。
【０００６】
そこで、本発明の目的は、リアルタイムに変化する音符間の演奏速度の変化に合った適切な音声波形を推論可能とすることにある。
【課題を解決するための手段】
【０００７】
態様の一例の電子楽器は、演奏時に指定される演奏時音高データを出力する音高指定部と、前記演奏時に演奏によって音高が順次指定される時間間隔を示す演奏テンポデータを、演奏時の演奏形態を示す演奏時演奏形態データとして出力する演奏形態出力部と、演奏時に、演奏時音高データ及び演奏時演奏形態データを学習済み音響モデルに入力することにより推論される音響モデルパラメータに基づいて、演奏時音高データ及び演奏時演奏形態データに対応する楽音データを合成し出力する発音モデル部と、を備える。
【０００８】
態様の他の一例の電子楽器は、演奏時の歌詞を示す演奏時歌詞データを出力する歌詞出力部と、演奏時に歌詞の出力に合わせて指定される演奏時音高データを出力する音高指定部と、前記演奏時に演奏によって音高が順次指定される時間間隔を示す演奏テンポデータを、演奏時の演奏形態を示す演奏時演奏形態データとして出力する演奏形態出力部と、演奏時に、演奏時歌詞データ、演奏時音高データ、及び演奏時演奏形態データを学習済み音

響モデルに入力することにより推論される音響モデルパラメータに基づいて、演奏時歌詞データ、演奏時音高データ、及び演奏時演奏形態データに対応する歌声音声データを合成し出力する発声モデル部と、を備える。

【発明の効果】

【０００９】

本発明によれば、リアルタイムに変化する音符間の演奏速度の変化に合った適切な音声波形を推論することが可能となる。

【図面の簡単な説明】

【００１０】

【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。

【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】歌い方のもととなる譜割りの例を示す説明図である。

【図５】演奏テンポの差により生じる歌声音声の波形変化を示す図である。

【図６】歌詞出力部、音高指定部、及び演奏形態出力部の構成例を示すブロック図である。

【図７】本実施形態のデータ構成例を示す図である。

【図８】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９】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１０】スイッチ処理の詳細例を示すフローチャートである。

【図１１】鍵盤処理の詳細例を示すフローチャートである。

【図１２】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１３】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】

【００１１】

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。

【００１２】

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、操作子としての複数の鍵からなる鍵盤１０１と、音量の指定、後述するソング再生のテンポ設定、後述する演奏テンポモードの設定、後述する演奏テンポのアジャスト設定、後述するソング再生開始、後述する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、後述するソング再生時の歌詞、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、特には図示しないが、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。

【００１３】

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、図１のＬＣＤ１０４が接続されるＬＣＤコントローラ２０８、外部のネットワークとＭＩＤＩデータ等のやりとりを行うネットワークインタフェース２１９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データ２１８及び歌声音声データ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、特には図示しないスピーカ又は出力端子から出力される。

【００１４】

ＣＰＵ２０１は、ＲＡＭ２０３をワークメモリとして使用しながらＲＯＭ２０２からＲＡＭ２０３にロードし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。

【００１５】

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。

【００１６】

音源ＬＳＩ２０４は、ＣＰＵ２０１からの発音制御データ２１６に従って、例えば特には図示しない波形ＲＯＭから楽音波形データを読み出し、楽音データ２１８としてＤ／Ａコンバータ２１１に出力する。音源ＬＳＩ２０４は、同時に最大２５６ボイスを発音させる能力を有する。

【００１７】

音声合成ＬＳＩ２０５は、ＣＰＵ２０１から、歌詞のテキストデータ（演奏時歌詞データ）と各歌詞に対応する各音高を指定するデータ（演奏時音高データ）と歌い方に関するデータ（演奏時演奏形態データ）を演奏時歌声データ２１５として与えられると、それに対応する歌声音声データ２１７を合成し、Ｄ／Ａコンバータ２１２に出力する。

【００１８】

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。

【００１９】

ＬＣＤコントローラ２０８は、ＬＣＤ１０４の表示状態を制御するＩＣ（集積回路）である。

【００２０】

図３は、本実施形態における音声合成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は、図２の音声合成ＬＳＩ２０５が実行する一機能として電子鍵盤楽器１００に内蔵される。

【００２１】

音声合成部３０２は、後述する歌詞の自動再生（以下「ソング再生」と記載）処理により図１の鍵盤１０１上の押鍵に基づいて図２のキースキャナ２０６を介してＣＰＵ２０１から指示される歌詞、音高、及び歌い方の情報を含む演奏時歌声データ２１５を入力することにより、歌声音声データ２１７を合成し出力する。このとき音声合成部３０２のプロセッサは、鍵盤１０１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ＣＰＵ２０１により生成された歌詞情報と、いずれかの鍵に対応付けられている音高情報と、歌い方に関する情報を含む演奏時歌声データ２１５を演奏時歌声解析部３０７に入力し、そこから出力される演奏時言語特徴量系列３１６を音響モデル部３０６に記憶されている学習済み音響モデルに入力し、その結果、音響モデル部３０６が出力したスペクトル情報３１８と音源情報３１９とに基づいて、歌い手の歌声を推論した歌声音声データ２１７を出力する発声処理を実行する。

【００２２】

音声学習部３０１は例えば、図３に示されるように、図１の電子鍵盤楽器１００とは別に外部に存在するサーバコンピュータ３００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３０１は、図２の音声合成ＬＳＩ２０５の処理能力に余裕があれば、音声合成ＬＳＩ２０５が実行する一機能として電子鍵盤楽器１００に内蔵されてもよい。

【００２３】

図２の音声学習部３０１及び音声合成部３０２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。

【００２４】

（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７３巻１号（２０１７），ｐｐ．５５－６２

【００２５】

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３００が実行する機能である図２の音声学習部３０１は、学習用歌声解析部３０３と学習用音響特徴量抽出部３０４とモデル学習部３０５とを含む。

【００２６】

音声学習部３０１において、学習用歌声音声データ３１２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３１１としては、各歌唱曲の歌詞のテキストデータ（学習用歌詞データ）と各歌詞に対応する各音高を指定するデータ（学習用音高データ）と学習用歌声音声データ３１２の歌い方を示すデータ（学習用演奏形態データ）とが用意される。学習用演奏形態データとしては、上記学習用音高データが順次指定される時間間隔が順次計測され、順次計測された時間間隔を示す各データが指定される。

【００２７】

学習用歌声解析部３０３は、学習用歌詞データ、学習用音高データ、及び学習用演奏形態データを含む学習用歌声データ３１１を入力してそのデータを解析する。この結果、学習用歌声解析部３０３は、学習用歌声データ３１１に対応する音素、音高、歌い方を表現する離散数値系列である学習用言語特徴量系列３１３を推定して出力する。

【００２８】

学習用音響特徴量抽出部３０４は、上記学習用歌声データ３１１の入力に合わせてその学習用歌声データ３１１に対応する歌詞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３１２を入力して分析する。この結果、学習用音響特徴量抽出部３０４は、学習用歌声音声データ３１２に対応する音声の特徴量を表す学習用音響特徴量系列３１４を抽出し、それを教師データとして出力する。

【００２９】

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

λ

と置く）とから、学習用音響特徴量系列３１４（これを

o

と置く）が生成される確率（これを

$P(o \mid l, \lambda)$

と置く）を最大にするような音響モデル

$\hat{\lambda}$

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。

【数１】

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} P(\boldsymbol{o} \mid \boldsymbol{l}, \boldsymbol{\lambda}) \qquad (1)$$

【００３０】

ここで、

$$\arg\max$$

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。

【００３１】

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

$$\hat{\boldsymbol{\lambda}}$$

を表現する学習結果データ３１５を出力する。

【００３２】

この学習結果データ３１５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ＲＯＭ２０２に記憶され、電子鍵盤楽器１００のパワーオン時に、図２のＲＯＭ２０２から音声合成ＬＳＩ２０５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、学習結果データ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、特には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ＬＳＩ２０５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。或いは、音声合成ＬＳＩ２０５とは別に、学習済み音響モデルをＦＰＧＡ（Ｆｉｅｌｄ－Ｐｒｏｇｒａｍｍａｂｌｅ　Ｇａｔｅ　Ａｒｒａｙ）等によりハードウェア化し、これをもって音響モデル部としてもよい。

【００３３】

音声合成ＬＳＩ２０５が実行する機能である音声合成部３０２は、演奏時歌声解析部３０７と音響モデル部３０６と発声モデル部３０８とを含む。音声合成部３０２は、演奏時に順次入力される演奏時歌声データ２１５に対応する歌声音声データ２１７を、音響モデル部３０６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順次合成し出力する、統計的音声合成処理を実行する。

【００３４】

演奏時歌声解析部３０７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ＣＰＵ２０１より指定される演奏時歌詞データ（歌詞テキストに対応する歌詞の音素）と演奏時音高データと演奏時演奏形態データ（歌い方データ）に関する情報を含む演奏時歌声データ２１５を入力し、そのデータを解析する。この結果、演奏時歌声解析部３０７は、演奏時歌声データ２１５に対応する音素、品詞、単語と音高と歌い方を表現する演奏時言語特徴量系列３１６を解析して出力する。

【００３５】

音響モデル部３０６は、演奏時言語特徴量系列３１６を入力することにより、それに対応する音響モデルパラメータである演奏時音響特徴量系列３１７を推定して出力する。即

ち音響モデル部３０６は、下記（２）式に従って、演奏時歌声解析部３０７から入力する演奏時言語特徴量系列３１６（これを再度

$$l$$

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

$$\hat{\lambda}$$

とに基づいて、演奏時音響特徴量系列３１７（これを再度

$$o$$

と置く）が生成される確率（これを

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

と置く）を最大にするような音響モデルパラメータである演奏時音響特徴量系列３１７の推定値

$$\hat{o}$$

を推定する。

【数２】

$$\hat{o} = \arg\max_{o} P(o \mid l, \hat{\lambda}) \qquad (2)$$

【００３６】

発声モデル部３０８は、演奏時音響特徴量系列３１７を入力することにより、ＣＰＵ２０１より指定される演奏時歌声データ２１５に対応する歌声音声データ２１７を合成し出力する。この歌声音声データ２１７は、図２のＤ／Ａコンバータ２１２からミキサ２１３及びアンプ２１４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。

【００３７】

学習用音響特徴量系列３１４や演奏時音響特徴量系列３１７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（パラメータ）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３０８は、音源生成部３０９と合成フィルタ部３１０とを含む。音源生成部３０９は、人間の声帯をモデル化した部分であり、音響モデル部３０６から入力する音源情報３１９の系列を順次入力することにより、例えば、音源情報３１９に含まれる基本周波数（Ｆ０）及びパワー値で周期的に繰り返されるパルス列データ（有声音音素の場合）、又は音源情報３１９に含まれるパワー値を有するホワイトノイズデータ（無声音音素の場合）、或いはそれらが混合されたデータからなる音源信号データを生成する。合成フィルタ部３１０は、人間の声道をモデル化した部分であり、音響モデル部３０６から順次入力するスペクトル情報３１８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

、音源生成部３０９から入力する音源信号データを励振源信号データとして、デジタル信号データである歌声音声データ３２１を生成し出力する。

【００３８】

学習用歌声音声データ３１２及び歌声音声データ２１７に対するサンプリング周波数は、例えば１６ＫＨｚ（キロヘルツ）である。また、学習用音響特徴量系列３１４及び演奏時音響特徴量系列３１７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５ｍｓｅｃ（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２５ｍｓｅｃ、窓関数はブラックマン窓、分析次数は２４次である。

【００３９】

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具体的な処理としては例えば、音響モデル部３０６に設定される学習結果データ３１５によって表現される音響モデルとして、ＨＭＭ（Ｈｉｄｄｅｎ　Ｍａｒｋｏｖ　Ｍｏｄｅｌ：隠れマルコフモデル）を用いる方法や、ＤＮＮ（Ｄｅｅｐ　Ｎｅｕｒａｌ　Ｎｅｔｗｏｒｋ：ディープニューラルネットワーク）を用いる方法を採用することができる。これらの具体的な実施形態については、前述した特許文献１に開示されているので、本出願では、その詳細な説明は省略する。

【００４０】

図３に例示した音声学習部３０１及び音声合成部３０２からなる統計的音声合成処理により、或る歌手の歌声を学習した学習済み音響モデルを搭載した音響モデル部３０６に、ソング再生される歌詞と演奏者により押鍵指定される音高とを含む演奏時歌声データ２１５を順次入力させることにより、或る歌手が良好に歌う歌声音声データ２１７を出力する電子鍵盤楽器１００が実現される。

【００４１】

ここで、歌唱音声では、速いパッセージのメロディとゆっくりしたパッセージのメロディとでは、歌い方に差がでるのが通常である。図４は、歌い方のもととなる譜割りの例を示す説明図である。図４（ａ）に速いパッセージの歌詞メロディの楽譜例、図４（ｂ）にゆっくりしたパッセージの歌詞メロディの楽譜例を示す。この例では、音高変化のパターンは同様であるが、図４（ａ）は、１６分音符（音符の長さが四分音符の４分の１）の連続の譜割りであるが、図４（ｂ）は、４分音符の連続の譜割りとなっている。従って、音高を変化させる速度については、図４（ａ）の譜割りは図４（ｂ）の譜割りの４倍の速度となる。速いパッセージの曲では、歌唱音声の子音部は短くしないとうまく歌唱（演奏）できない。逆に、ゆっくりしたパッセージの曲では、歌唱音声の子音部を長くしたほうが、表現力の高い歌唱（演奏）ができる。上述のように、音高の変化パターンが同じでも、歌唱メロディの音符ひとつひとつの長さの違い（四分音符、八分音符、十六分音符等）により、歌唱（演奏）速度に差が生じるが、全く同じ楽譜が歌唱（演奏）されても、演奏時のテンポが変化すれば演奏速度に差が生じるのは言うまでもない。以下の説明では、上述の２つの要因により生じる音符間の時間間隔（発音速度）を通常の楽曲のテンポと区別して「演奏テンポ」と記載することにする。

【００４２】

図５は、図４に例示したような演奏テンポの差により生じる歌声音声の波形変化を示す図である。図５に示される例は、／ｇａ／の音声が発音された場合の歌声音声の波形例を示している。／ｇａ／の音声は、子音の／ｇ／と、母音の／ａ／が組み合わさった音声である。子音部の音長（時間長）は、通常は数１０ミリ秒から２００ミリ秒程度であることが多い。ここで、図５（ａ）は速いパッセージで歌唱された場合の歌声音声波形の例、図５（ｂ）はゆっくりしたパッセージで歌唱された場合の歌声音声波形の例を示している。図５（ａ）と（ｂ）の波形の違いは、子音／ｇ／の部分の長さが異なることである。速いパッセージで歌唱された場合には、図５（ａ）に示されるように、子音部の発音時間長が短く、逆に、ゆっくりしたパッセージで歌唱される場合には、図５（ｂ）に示されるように、子音部の発音時間長が長くなっていることがわかる。速いパッセージでの歌唱では子

音をはっきり歌わず、発音開始速度を優先するが、ゆっくりしたパッセージでは、子音を長くはっきり発音させることで、言葉としての明瞭度を上げる発音になることが多い。

【００４３】

上述したような、演奏テンポの差を歌声音声データの変化に反映させるために、本実施形態における図３に例示した音声学習部３０１及び音声合成部３０２からなる統計的音声合成処理において、音声学習部３０１において入力される学習用歌声データ３１１に、歌詞を示す学習用歌詞データと、音高を示す学習用音高データに、歌い方を示す学習用演奏形態データが追加され、この学習用演奏形態データに演奏テンポの情報が含ませられる。音声学習部３０１内の学習用歌声解析部３０３は、このような学習用歌声データ３１１を解析することにより、学習用言語特徴量系列３１３を生成する。そして、音声学習部３０１内のモデル学習部３０５が、この学習用言語特徴量系列３１３を用いて機械学習を行う。この結果、モデル学習部３０５が、演奏テンポの情報を含む学習済み音響モデルを学習結果データ３１５として出力し、音声合成ＬＳＩ２０５の音声合成部３０２内の音響モデル部３０６に記憶させることができる。学習用演奏形態データとしては、上記学習用音高データが順次指定される時間間隔が順次計測され、順次計測された時間間隔を示す各演奏テンポデータが指定される。このように、本実施形態におけるモデル学習部３０５は、歌い方による演奏テンポの違いが加味された学習済み音響モデルを導きだせるような学習を行うことができる。

【００４４】

一方、上述のような学習済み音響モデルがセットされた音響モデル部３０６を含む音声合成部３０２においては、演奏時歌声データ２１５に、歌詞を示す演奏時歌詞データと、音高を示す演奏時音高データに、歌い方を示す演奏時演奏形態データが追加され、この演奏時演奏形態データに演奏テンポの情報を含ませることができる。音声合成部３０２内の演奏時歌声解析部３０７は、このような演奏時歌声データ２１５を解析することにより、演奏時言語特徴量系列３１６を生成する。そして、音声合成部３０２内の音響モデル部３０６は、この演奏時言語特徴量系列３１６を学習済み音響モデルに入力させることにより、対応するスペクトル情報３１８及び音源情報３１９を出力し、それぞれ発声モデル部３０８内の合成フィルタ部３１０及び音源生成部３０９に供給する。この結果、発声モデル部３０８は、歌い方による演奏テンポの違いにより例えば図５（ａ）及び（ｂ）に例示したような子音の長さ等の変化が反映された歌声音声データ２１７を出力することができる。即ち、リアルタイムに変化する音符間の演奏速度の変化に合った、適切な歌声音声データ２１７を推論することが可能となる。

【００４５】

図６は、上述した演奏時歌声データ２１５を生成するための、図２のＣＰＵ２０１が後述する図８から図１１のフローチャートで例示される制御処理の機能として実現する歌詞出力部、音高指定部、及び演奏形態出力部の構成例を示すブロック図である。

【００４６】

歌詞出力部６０１は、演奏時の歌詞を示す各演奏時歌詞データ６０９を、図２の音声合成ＬＳＩ２０５に出力する各演奏時歌声データ２１５に含ませて出力する。具体的には、歌詞出力部６０１は、図２においてＣＰＵ２０１が予めＲＯＭ２０２からＲＡＭ２０３にロードしたソング再生の曲データ６０４中の各タイミングデータ６０５を順次読み出しながら、各タイミングデータ６０５が示すタイミングに従って、各タイミングデータ６０５と組で曲データ６０４として記憶されている各イベントデータ６０６中の各歌詞データ（歌詞テキスト）６０８を順次読み出し、それぞれを各演奏時歌詞データ６０９とする。

【００４７】

音高指定部６０２は、演奏時に各歌詞の出力に合わせて指定される各音高を示す各演奏時音高データ６１０を、図２の音声合成ＬＳＩ２０５に出力する各演奏時歌声データ２１５に含ませて出力する。具体的には、音高指定部６０２は、ＲＡＭ２０３にロードされた上記ソング再生用の曲データ６０４中の各タイミングデータ６０５を順次読み出しながら、各タイミングデータ６０５が示すタイミングにおいて、演奏者が図１の鍵盤１０１で何

れかの鍵を押鍵操作してその押鍵された鍵の音高情報がキースキャナ２０６を介して入力されている場合には、その音高情報を演奏時音高データ６１０とする。また、音高指定部６０２は、各タイミングデータ６０５が示すタイミングにおいて、演奏者が図１の鍵盤１０１でどの鍵も押鍵操作していない場合には、そのタイミングデータ６０５と組で曲データ６０４として記憶されているイベントデータ６０６中の音高データ６０７を演奏時音高データ６１０とする。

【００４８】

演奏形態出力部６０３は、演奏時の演奏形態である歌い方を示す演奏時演奏形態データ６１１を、図２の音声合成ＬＳＩ２０５に出力する各演奏時歌声データ２１５に含ませて出力する。

【００４９】

具体的には、演奏形態出力部６０３は、演奏者が図１の第１のスイッチパネル１０２上で、後述するように演奏テンポモードをフリーモードに設定している場合には、演奏時に演奏者の押鍵によって音高が指定される時間間隔を順次計測し、順次計測された時間間隔を示す各演奏テンポデータを、各演奏時演奏形態データ６１１とする。

【００５０】

一方、演奏形態出力部６０３は、演奏者が図１の第１のスイッチパネル１０２上で、後述するように演奏テンポモードをフリーモードに設定していない場合には、ＲＡＭ２０３にロードされた上記ソング再生用の曲データ６０４から順次読み出される各タイミングデータ６０５が示す各時間間隔に対応する各演奏テンポデータを、各演奏時演奏形態データ６１１とする。

【００５１】

また、演奏形態出力部６０３は、演奏者が図１の第１のスイッチパネル１０２上で、後述するように演奏テンポモードを意図的に変更する演奏テンポアジャスト設定を行った場合には、その演奏テンポアジャスト設定の値に基づいて、上述のようにして順次得られる各演奏テンポデータの値を意図的に変更し、変更後の各演奏テンポデータを演奏時演奏形態データ６１１とする。

【００５２】

以上のようにして、図２のＣＰＵ２０１が実行する歌詞出力部６０１、音高指定部６０２、及び演奏形態出力部６０３の各機能は、演奏者の押鍵操作又はソング再生による押鍵イベントが発生したタイミングで、演奏時歌詞データ６０９、演奏時音高データ６１０、及び演奏時演奏形態データ６１１を含む演奏時歌声データ２１５を生成し、それを図２又は図３の構成を有する音声合成ＬＳＩ２０５内の音声合成部３０２に対して発行することができる。

【００５３】

図３から図６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７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詳細な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つ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。

【００５４】

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"MThd"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、Ｆｏｒ

ｍａｔＴｙｐｅ、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。ＮｕｍｂｅｒＯｆＴｒａｃｋ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。ＴｉｍｅＤｉｖｉｓｉｏｎ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。

【００５５】

トラックチャンク１は、歌詞パートを示し、図６の曲データ６０４に対応し、ＣｈｕｎｋＩＤと、ＣｈｕｎｋＳｉｚｅと、図６のタイミングデータ６０５に対応するＤｅｌｔａＴｉｍｅ＿１［ｉ］及び図６のイベントデータ６０６に対応するＥｖｅｎｔ＿１［ｉ］からなる演奏データ組（０≦ｉ≦Ｌ－１）とからなる。また、トラックチャンク２は、伴奏パートに対応し、ＣｈｕｎｋＩＤと、ＣｈｕｎｋＳｉｚｅと、伴奏パートのタイミングデータであるＤｅｌｔａＴｉｍｅ＿２［ｉ］及び伴奏パートのイベントデータであるＥｖｅｎｔ＿２［ｊ］からなる演奏データ組（０≦ｊ≦Ｍ－１）とからなる。

【００５６】

トラックチャンク１、２における各ＣｈｕｎｋＩＤ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。トラックチャンク１、２における各ＣｈｕｎｋＳｉｚｅは、各トラックチャンクにおいて、ＣｈｕｎｋＩＤとＣｈｕｎｋＳｉｚｅを除く部分のデータ長を示す４バイトデータである。

【００５７】

図６のタイミングデータ６０５であるＤｅｌｔａＴｉｍｅ＿１［ｉ］は、その直前の図６のイベントデータ６０６である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、伴奏パートのタイミングデータであるＤｅｌｔａＴｉｍｅ＿２［ｉ］は、その直前の伴奏パートのイベントデータである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。

【００５８】

図６のイベントデータ６０６であるＥｖｅｎｔ＿１［ｉ］は、本実施例のトラックチャンク１／歌詞パートにおいては、歌詞の発声テキストと音高の２つの情報を持つメタイベントである。伴奏パートのイベントデータであるＥｖｅｎｔ＿２［ｉ］は、トラックチャンク２／伴奏パートにおいて、伴奏音のノートオン又はノートオフを指示するＭＩＤＩイベント、又は伴奏音の拍子を指示するメタイベントである。

【００５９】

トラックチャンク１／歌詞パートの、各演奏データ組ＤｅｌｔａＴｉｍｅ＿１［ｉ］及びＥｖｅｎｔ＿１［ｉ］において、その直前のイベントデータ６０６であるＥｖｅｎｔ＿１［ｉ－１］の実行時刻からタイミングデータ６０５であるＤｅｌｔａＴｉｍｅ＿１［ｉ］だけ待った上でイベントデータ６０６であるＥｖｅｎｔ＿１［ｉ］が実行されることにより、ソング再生の進行が実現される。一方、トラックチャンク２／伴奏パートの、各演奏データ組ＤｅｌｔａＴｉｍｅ＿２［ｉ］及びＥｖｅｎｔ＿２［ｉ］において、その直前のイベントデータＥｖｅｎｔ＿２［ｉ－１］の実行時刻からタイミングデータＤｅｌｔａＴｉｍｅ＿２［ｉ］だけ待った上でイベントデータ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。

【００６０】

図８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。

【００６１】
ＣＰＵ２０１は、まず初期化処理を実行した後（ステップＳ８０１）、ステップＳ８０２からＳ８０８の一連の処理を繰り返し実行する。
【００６２】
この繰返し処理において、ＣＰＵ２０１はまず、スイッチ処理を実行する（ステップＳ８０２）。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。スイッチ処理の詳細は、図１０のフローチャートを用いて後述する。
【００６３】
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Ｓ８０３）。鍵盤処理では、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。また、鍵盤処理において、ＣＰＵ２０１は、直前の押鍵から現在の押鍵までの時間間隔を演奏テンポデータとして算出する処理を実行する。鍵盤処理の詳細は、図１１のフローチャートを用いて後述する。
【００６４】
次に、ＣＰＵ２０１は、図１のＬＣＤ１０４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ＬＣＤ１０４に表示する表示処理を実行する（ステップＳ８０４）。ＬＣＤ１０４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及び伴奏の楽譜や、各種設定情報がある。
【００６５】
次に、ＣＰＵ２０１は、ソング再生処理を実行する（ステップＳ８０５）。ソング再生処理では、ＣＰＵ２０１は、ソング再生に基づいて音声合成ＬＳＩ２０５を動作させるための歌詞、発声音高、及び演奏テンポを含む演奏時歌声データ２１５を生成して音声合成ＬＳＩ２０５に発行する。ソング再生処理の詳細は、図１３のフローチャートを用いて後述する。
【００６６】
続いて、ＣＰＵ２０１は、音源処理を実行する（ステップＳ８０６）。音源処理において、ＣＰＵ２０１は、音源ＬＳＩ２０４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。
【００６７】
続いて、ＣＰＵ２０１は、音声合成処理を実行する（ステップＳ８０７）。音声合成処理において、ＣＰＵ２０１は、音声合成ＬＳＩ２０５による音声合成の実行を制御する。
【００６８】
最後にＣＰＵ２０１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Ｓ８０８）。ステップＳ８０８の判定がＮＯならば、ＣＰＵ２０１は、ステップＳ８０２の処理に戻る。ステップＳ８０８の判定がＹＥＳならば、ＣＰＵ２０１は、図８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。
【００６９】
図９（ａ）、（ｂ）、及び（ｃ）はそれぞれ、図８のステップＳ８０１の初期化処理、図８のステップＳ８０２のスイッチ処理における後述する図１０のステップＳ１００２のテンポ変更処理、及び同じく図１０のステップＳ１００６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。
【００７０】
まず、図８のステップＳ８０１の初期化処理の詳細例を示す図９（ａ）において、ＣＰＵ２０１は、ＴｉｃｋＴｉｍｅ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、ＴｉｃｋＴｉｍｅという時間を単位として進行する。図７に例示される曲データのヘッダチャンク内のＴｉｍｅＤｉｖｉｓｉｏｎ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４８０ＴｉｃｋＴｉｍｅの時間長を有する。また、図７に例示される曲データの各トラックチャンク内の待ち時間ＤｅｌｔａＴｉｍｅ＿１［ｉ］の値及びＤｅｌｔａＴｉｍｅ＿２［ｉ］の値も、ＴｉｃｋＴｉｍｅの時間単位によりカウントされる。ここで、１ＴｉｃｋＴｉｍｅ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ＴｉｍｅＤｉｖｉｓｉｏｎとすれば、ＴｉｃｋＴｉｍｅの秒数は、下記（３）式により算出される。

【００７１】

ＴｉｃｋＴｉｍｅ［秒］＝６０／Ｔｅｍｐｏ／ＴｉｍｅＤｉｖｉｓｉｏｎ
・・・（３）

【００７２】

そこで、図９（ａ）のフローチャートで例示される初期化処理において、ＣＰＵ２０１はまず、上記（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Ｓ９０１）。なお、テンポ値Ｔｅｍｐｏは、初期状態では図２のＲＯＭ２０２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。

【００７３】

次に、ＣＰＵ２０１は、図２のタイマ２１０に対して、ステップＳ９０１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Ｓ９０２）。この結果、タイマ２１０において上記ＴｉｃｋＴｉｍｅ［秒］が経過する毎に、ＣＰＵ２０１に対してソング再生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ＣＰＵ２０１で実行される自動演奏割込み処理（後述する図１２）では、１ＴｉｃｋＴｉｍｅ毎にソング再生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。

【００７４】

続いて、ＣＰＵ２０１は、図２のＲＡＭ２０３の初期化等のその他初期化処理を実行する（ステップＳ９０３）。その後、ＣＰＵ２０１は、図９（ａ）のフローチャートで例示される図８のステップＳ８０１の初期化処理を終了する。

【００７５】

図９（ｂ）及び（ｃ）のフローチャートについては、後述する。図１０は、図８のステップＳ８０２のスイッチ処理の詳細例を示すフローチャートである。

【００７６】

ＣＰＵ２０１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Ｓ１００１）。その判定がＹＥＳならば、ＣＰＵ２０１は、テンポ変更処理を実行する（ステップＳ１００２）。この処理の詳細は、図９（ｂ）を用いて後述する。ステップＳ１００１の判定がＮＯならば、ＣＰＵ２０１は、ステップＳ１００２の処理はスキップする。

【００７７】

次に、ＣＰＵ２０１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Ｓ１００３）。その判定がＹＥＳならば、ＣＰＵ２０１は、ソング曲読込み処理を実行する（ステップＳ１００４）。この処理は、図７で説明したデータ構造を有する曲データを、図２のＲＯＭ２０２からＲＡＭ２０３に読み込む処理である。なお、ソング曲読込み処理は、演奏中でなくても、演奏開始前でもよい。これ以降、図７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、ＲＡＭ２０３に読み込まれた曲データに対して実行される。ステップＳ１００３の判定がＮＯならば、ＣＰＵ２０１は、ステップＳ１００４の処理はスキップする。

【００７８】

続いて、ＣＰＵ２０１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Ｓ１００５）。その判定がＹＥＳならば、ＣＰＵ２０１は、ソング開始処理を実行する（ステップＳ１００６）。この処理の詳細は、図９（ｃ）を用いて後述する。ステップＳ１００５の判定がＮＯならば、ＣＰＵ２０１は、ステップＳ１００６の処理はスキップする。

【００７９】

続いて、ＣＰＵ２０１は、図１の第１のスイッチパネル１０２においてフリーモードスイッチが操作されたか否かを判定する（ステップＳ１００７）。その判定がＹＥＳならば、ＣＰＵ２０１は、ＲＡＭ２０３上の変数ＦｒｅｅＭｏｄｅの値を変更するフリーモードセット処理を実行する（ステップＳ１００８）。フリーモードスイッチは例えばトグル動作になっており、変数ＦｒｅｅＭｏｄｅの値は、例えば図９ステップＳ９０３で、例えば値１に初期設定されている。その状態でフリーモードスイッチが押されると変数ＦｒｅｅＭｏｄｅの値は０になり、もう一度押されるとその値は１になる、というようにフリーモードスイッチが押される毎に変数ＦｒｅｅＭｏｄｅの値が０と１で交互に切り替えられる。変数ＦｒｅｅＭｏｄｅの値が、１のときにはフリーモードが設定され、値０のときにはフリーモードの設定が解除される。ステップＳ１００７の判定がＮＯならば、ＣＰＵ２０１は、ステップＳ１００８の処理はスキップする。

【００８０】

続いて、ＣＰＵ２０１は、図１の第１のスイッチパネル１０２において演奏テンポアジャストスイッチが操作されたか否かを判定する（ステップＳ１００９）。その判定がＹＥＳならば、ＣＰＵ２０１は、ＲＡＭ２０３上の変数ＳｈｉｎＡｄｊｕｓｔの値を、上記演奏テンポアジャストスイッチの操作に続いて第１のスイッチパネル１０２上の数値キーによって指定された値に変更する演奏テンポアジャスト設定処理を実行する（ステップＳ１０１０）。変数ＳｈｉｎＡｄｊｕｓｔの値は、例えば図９のステップＳ９０３で、値０に初期設定される。ステップＳ１００９の判定がＮＯならば、ＣＰＵ２０１は、ステップＳ１０１０の処理はスキップする。

【００８１】

最後に、ＣＰＵ２０１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Ｓ１０１１）。その後、ＣＰＵ２０１は、図１０のフローチャートで例示される図８のステップＳ８０２のスイッチ処理を終了する。

【００８２】

図９（ｂ）は、図１０のステップＳ１００２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９（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。

【００８３】

まず、ＣＰＵ２０１は、図８のステップＳ８０１の初期化処理で実行された図９（ａ）のステップＳ９０１の場合と同様にして、前述した（３）式に対応する演算処理により、ＴｉｃｋＴｉｍｅ［秒］を算出する（ステップＳ９１１）。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。

【００８４】

次に、ＣＰＵ２０１は、図８のステップＳ８０１の初期化処理で実行された図９（ａ）のステップＳ９０２の場合と同様にして、図２のタイマ２１０に対して、ステップＳ９１１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Ｓ９１２）。その後、ＣＰＵ２０１は、図９（ｂ）のフローチャートで例示される図１０のステップＳ１００２のテンポ変更処理を終了する。

【００８５】

図９（ｃ）は、図１０のステップＳ１００６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。

【００８６】
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上のタイミングデータ変数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７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［ｊ］及びＥｖｅｎｔ＿２［ｊ］（１≦ｊ≦Ｍ－１）の夫々ｊ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Ｓ９２１）。これにより、図７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。

【００８７】
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Ｎｕｌｌ値に初期設定する（ステップＳ９２２）。Ｎｕｌｌ値は通常０と定義されることが多いが、インデックス番号が０である場合があることから、本実施例においてはＮｕｌｌ値を－１と定義する。

【００８８】
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Ｓ９２３）。

【００８９】
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Ｓ９２４）。

【００９０】
ステップＳ９２４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Ｓ９２５）。逆に、ステップＳ９２４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Ｓ９２６）。ステップＳ９２５又はＳ９２６の処理の後、ＣＰＵ２０１は、図９（ｃ）のフローチャートで例示される図１０のステップＳ１００６のソング開始処理を終了する。

【００９１】
図１１は、図８のステップＳ８０３の鍵盤処理の詳細例を示すフローチャートである。まず、ＣＰＵ２０１は、図２のキースキャナ２０６を介して図１の鍵盤１０１上の何れかの鍵が操作されたか否かを判定する（ステップＳ１１０１）。

【００９２】
ステップＳ１１０１の判定がＮＯならば、ＣＰＵ２０１は、そのまま図１１のフローチャートで例示される図８のステップＳ８０３の鍵盤処理を終了する。

【００９３】
ステップＳ１１０１の判定がＹＥＳならば、ＣＰＵ２０１は、押鍵がなされたか離鍵がなされたかを判定する（ステップＳ１１０２）。

【００９４】
ステップＳ１１０２の判定において離鍵がなされたと判定された場合には、ＣＰＵ２０１は、音声合成ＬＳＩ２０５に対して、離鍵された音高（又はキーナンバ）に対応する歌声音声データ２１７の発声の消音を指示する（ステップＳ１１１３）。この指示に従って、音声合成ＬＳＩ２０５内の図３の音声合成部３０２は、該当する歌声音声データ２１７の発声を中止する。その後、ＣＰＵ２０１は、図１１のフローチャートで例示される図８

のステップＳ８０３の鍵盤処理を終了する。
【００９５】
ステップＳ１１０２の判定において押鍵がなされたと判定された場合には、ＣＰＵ２０１は、ＲＡＭ２０３上の変数ＦｒｅｅＭｏｄｅの値を判定する（ステップＳ１１０３）。この変数ＦｒｅｅＭｏｄｅの値は、前述した図１０のステップＳ１００８で設定され、変数フリーモードが値１のときにはフリーモードが設定され、値０のときにはフリーモードの設定が解除される。
【００９６】
ステップ１１０３で変数フリーモードの値が０であってフリーモードの設定が解除されていると判定された場合には、ＣＰＵ２０１は、図６の演奏形態出力部６０３の説明で前述したように、ＲＡＭ２０３にロードされたソング再生用の曲データ６０４から順次読み出される各タイミングデータ６０５である後述するＤｅｌｔａＴｉｍｅ＿１［ＡｕｔｏＩｎｄｅｘ＿１］を用いて下記（４）式で例示される演算処理により算出される値を、図６の演奏時演奏形態データ６１１に対応する演奏テンポを示すＲＡＭ２０３上の変数ＰｌａｙＴｅｍｐｏにセットする（ステップＳ１１０９）。
【００９７】

$$PlayTempo = \left(1 / DeltaTime\_1[AutoIndex\_1]\right) \times 所定の係数 \quad \cdots (4)$$

【００９８】
（４）式において、所定の係数は本実施例においては曲データのＴｉｍｅＤｉｖｉｓｉｏｎ値×６０である。すなわちＴｉｍｅＤｉｖｉｓｉｏｎ値が４８０であれば、ＤｅｌｔａＴｉｍｅ＿１［ＡｕｔｏＩｎｄｅｘ＿１］が４８０のときＰｌａｙＴｅｍｐｏは６０（通常のテンポ６０に相当）となる。ＤｅｌｔａＴｉｍｅ＿１［ＡｕｔｏＩｎｄｅｘ＿１］が２４０のときはＰｌａｙＴｅｍｐｏは１２０（通常のテンポ１２０に相当）となる。
【００９９】
フリーモードの設定が解除されている場合には、演奏テンポは、ソング再生のタイミング情報に同期して設定されることになる。
【０１００】
ステップ１１０３で変数フリーモードの値が１であると判定された場合には、ＣＰＵ２０１は更に、ＲＡＭ２０３上の変数ＮｏｔｅＯｎＴｉｍｅの値がＮｕｌｌ値であるか否かを判定する（ステップＳ１１０４）。ソング再生の開始時には、例えば図９のステップＳ９０３において、変数ＮｏｔｅＯｎＴｉｍｅの値はＮｕｌｌ値に初期設定されており、ソング再生開始後は後述するステップＳ１１１０において図２のタイマ２１０の現在時刻が順次セットされる。
【０１０１】
ソング再生の開始時であってステップＳ１１０４の判定がＹＥＳになった場合は、演奏者の押鍵操作から演奏テンポを決定することができないので、ＣＰＵ２０１は、ＲＡＭ２０３上のタイミングデータ６０５であるＤｅｌｔａＴｉｍｅ＿１［ＡｕｔｏＩｎｄｅｘ＿１］を用いて前述した（４）式で例示される演算処理により算出される値を、ＲＡＭ２０３上の変数ＰｌａｙＴｅｍｐｏにセットする（ステップＳ１１０９）。このようにソング再生の開始時には、演奏テンポは、暫定的にソング再生のタイミング情報に同期して設定されることになる。
【０１０２】
ソング再生の開始後であってステップＳ１１０４の判定がＮＯになった場合は、ＣＰＵ２０１は、まず図２のタイマ２１０が示す現在時刻から前回の押鍵時刻を示しているＲＡＭ２０３上の変数ＮｏｔｅＯｎＴｉｍｅの値を減算して得られる差分時間をＲＡＭ２０３上の変数ＤｅｌｔａＴｉｍｅにセットする（ステップＳ１１０５）。
【０１０３】
次に、ＣＰＵ２０１は、前回の押鍵から今回の押鍵までの差分時間を示す変数Ｄｅｌｔ

ａＴｉｍｅの値が、コード演奏（和音）による同時押鍵とみなす所定の最大時間よりも小さいか否かを判定する（ステップＳ１１０６）。

【０１０４】

ステップＳ１１０６の判定がＹＥＳで、今回の押鍵がコード演奏（和音）による同時押鍵であると判定された場合には、ＣＰＵ２０１は、演奏テンポを決定するための処理は実行せずに、後述するステップＳ１１１０の処理に移行する。

【０１０５】

ステップＳ１１０６の判定がＮＯで、今回の押鍵がコード演奏（和音）による同時押鍵ではないと判定された場合には、ＣＰＵ２０１は更に、前回の押鍵から今回の押鍵までの差分時間を示す変数ＤｅｌｔａＴｉｍｅの値が、演奏が途切れたとみなす最小時間よりも大きいか否かを判定する（ステップＳ１１０７）。

【０１０６】

ステップＳ１１０７の判定がＹＥＳで、しばらく演奏が途切れた後の押鍵（演奏フレーズの先頭）であると判定された場合には、演奏フレーズの演奏テンポを決定することができないので、ＣＰＵ２０１は、ＲＡＭ２０３上のタイミングデータ６０５であるＤｅｌｔａＴｉｍｅ＿１［ＡｕｔｏＩｎｄｅｘ＿１］を用いて前述した（４）式で例示される演算処理により算出される値を、ＲＡＭ２０３上の変数ＰｌａｙＴｅｍｐｏにセットする（ステップＳ１１０９）。このように、しばらく演奏が途切れた後の押鍵（演奏フレーズの先頭）である場合には、演奏テンポは、暫定的にソング再生のタイミング情報に同期して設定されることになる。

【０１０７】

ステップＳ１１０７の判定がＮＯで、今回の押鍵がコード演奏（和音）による同時押鍵でもなく演奏フレーズの先頭での押鍵でもないと判定された場合には、ＣＰＵ２０１は、下記（５）式に例示されるように、前回の押鍵から今回の押鍵までの差分時間を示す変数ＤｅｌｔａＴｉｍｅの逆数に所定の係数を乗算して得られる値を、図６の演奏時演奏形態データ６１１に対応する演奏テンポを示すＲＡＭ２０３上の変数ＰｌａｙＴｅｍｐｏにセットする（ステップＳ１１０８）。

【０１０８】

ＰｌａｙＴｅｍｐｏ＝（１／ＤｅｌｔａＴｉｍｅ）×所定の係数　・・（５）

【０１０９】

ステップＳ１１０８での処理により、前回の押鍵と今回の押鍵の時間差を示す変数ＤｅｌｔａＴｉｍｅの値が小さい場合には、演奏テンポであるＰｌａｙＴｅｍｐｏの値は大きくなり（演奏テンポが速くなり）、演奏フレーズが速いパッセージであるとみなされ、音声合成ＬＳＩ２０５内の音声合成部３０２において、図５（ａ）に例示したように子音部の時間長が短い歌声音声データ２１７の音声波形が推論される。一方、時間差を示す変数ＤｅｌｔａＴｉｍｅの値が大きい場合には、演奏テンポの値は小さくなり（演奏テンポが遅くなり）、演奏フレーズがゆっくりとしたパッセージであるとみなされ、音声合成部３０２において、図５（ｂ）に例示したように子音部の時間長が長い歌声音声データ２１７の音声波形が推論される。

【０１１０】

前述したステップＳ１１０８の処理の後、前述したステップＳ１１０９の処理の後、又は前述したステップＳ１１０６の判定がＹＥＳとなった後に、ＣＰＵ２０１は、前回の押鍵時刻を示すＲＡＭ２０３上の変数ＮｏｔｅＯｎＴｉｍｅに、図２のタイマ２１０が示す現在時刻をセットする（ステップＳ１１１０）。

【０１１１】

最後に、ＣＰＵ２０１は、ステップＳ１１０８又はＳ１１０９で決定された演奏テンポを示すＲＡＭ２０３上の変数ＰｌａｙＴｅｍｐｏの値に、演奏者が意図的に設定した演奏テンポアジャスト値が設定されているＲＡＭ２０３上の変数ＳｈｉｎＡｄｊｕｓｔ（図１０のステップＳ１０１０参照）の値を加算して得られる値を、新たな変数ＰｌａｙＴｅｍｐｏの値としてセットする（ステップＳ１１１１）。その後、ＣＰＵ２０１は、図１１

のフローチャートで例示される図８のステップＳ８０３の鍵盤処理を終了する。
【０１１２】
ステップＳ１１１１の処理により、演奏者は、音声合成部３０２で合成される歌声音声データ２１７における子音部の時間長を意図的に調整（アジャスト）することができる。演奏者は、曲目や嗜好により歌い方を調整したい場合がある。例えば、ある曲では全体的に音を短く切って歯切れよく演奏したい場合は、子音を短くして早口で歌ったような音声を発音してほしい、逆に、ある曲では全体的にゆったり演奏したい場合は、ゆっくり歌ったような子音の息遣いをはっきり聞かせることができる音声を発音してほしいという場合がある。そこで、本実施形態では、演奏者が、例えば図１の第１のスイッチパネル１０２上の演奏テンポアジャストスイッチを操作することにより、変数ＳｈｉｎＡｄｊｕｓｔの値を変更し、これに基づいて変数ＰｌａｙＴｅｍｐｏの値を調整することにより、演奏者の意図を反映した歌声音声データ２１７を合成することができる。スイッチ操作以外にも電子鍵盤楽器１００に接続される可変抵抗を利用したペダルを足で操作することにより、ＳｈｉｎＡｄｊｕｓｔの値を楽曲中の任意のタイミングで細かく制御することもできる。
【０１１３】
以上の鍵盤処理によって変数ＰｌａｙＴｅｍｐｏに設定された演奏テンポ値は、後述するソング再生処理において、演奏時歌声データ２１５の一部として設定されて（後述する図１３のステップＳ１３０５参照）、音声合成ＬＳＩ２０５に発行される。
【０１１４】
以上の鍵盤処理において、特に、ステップＳ１１０３からＳ１１０９、及びステップＳ１１１１の処理は、図６の演奏形態出力部６０３の機能に対応する。
【０１１５】
図１２は、図２のタイマ２１０においてＴｉｃｋＴｉｍｅ［秒］毎に発生する割込み（図９（ａ）のステップＳ９０２又は図９（ｂ）のステップＳ９１２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７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。
【０１１６】
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Ｓ１２０１からＳ１２０６）を実行する。始めにＣＰＵ２０１は、ＳｏｎｇＳｔａｒｔ値が１であるか否か（図１０のステップＳ１００６及び図９のステップＳ９２３参照）、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Ｓ１２０１）。
【０１１７】
歌詞及び伴奏の進行が指示されていないと判定された（ステップＳ１２０１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１２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。
【０１１８】
歌詞及び伴奏の進行が指示されていると判定された（ステップＳ１２０１の判定がＹＥＳである）場合には、ＣＰＵ２０１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ＲＡＭ２０３上の変数ＤｅｌｔａＴ＿１の値が、ＲＡＭ２０３上の変数ＡｕｔｏＩｎｄｅｘ＿１の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を示すタイミングデータ６０５（図６）であるＲＡＭ２０３上のＤｅｌｔａＴｉｍｅ＿１［ＡｕｔｏＩｎｄｅｘ＿１］に一致したか否かを判定する（ステップＳ１２０２）。
【０１１９】
ステップＳ１２０２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変数ＤｅｌｔａＴ＿１の値を＋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Ｓ１２０３）。その後、ＣＰＵ２０１は、後述するステップＳ１２０７に移行する。

【０１２０】
ステップＳ１２０２の判定がＹＥＳになると、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変数ＡｕｔｏＩｎｄｅｘ＿１の値を、ＲＡＭ２０３上の変数ＳｏｎｇＩｎｄｅｘに格納する（ステップＳ１２０４）。
【０１２１】
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変数ＡｕｔｏＩｎｄｅｘ＿１の値を＋１インクリメントする（ステップＳ１２０５）。
【０１２２】
また、ＣＰＵ２０１は、トラックチャンク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変数ＤｅｌｔａＴ＿１値を０にリセットする（ステップＳ１２０６）。その後、ＣＰＵ２０１は、ステップＳ１２０７の処理に移行する。
【０１２３】
次に、ＣＰＵ２０１は、トラックチャンク２に対応する一連の処理（ステップＳ１２０７からＳ１２１３）を実行する。始めにＣＰＵ２０１は、トラックチャンク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ＲＡＭ２０３上の変数ＤｅｌｔａＴ＿２値が、ＲＡＭ２０３上の変数ＡｕｔｏＩｎｄｅｘ＿２の値が示すこれから実行しようとする演奏データ組のＲＡＭ２０３上のタイミングデータＤｅｌｔａＴｉｍｅ＿２［ＡｕｔｏＩｎｄｅｘ＿２］に一致したか否かを判定する（ステップＳ１２０７）。
【０１２４】
ステップＳ１２０７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変数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Ｓ１２０８）。その後、ＣＰＵ２０１は、図１２のフローチャートで例示される自動演奏割込み処理を終了する。
【０１２５】
ステップＳ１２０７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（伴奏なし）を判定する（ステップＳ１２０９）（図９（ｃ）のステップＳ９２４からＳ９２６を参照）。
【０１２６】
ステップＳ１２０９の判定がＹＥＳならば、ＣＰＵ２０１は、変数ＡｕｔｏＩｎｄｅｘ＿２値が示すトラックチャック２の伴奏に関するＲＡＭ２０３上のイベントデータＥｖｅｎｔ＿２［ＡｕｔｏＩｎｄｅｘ＿２］が示す処理を実行する（ステップＳ１２１０）。ここで実行されるイベントデータＥｖｅｎｔ＿２［ＡｕｔｏＩｎｄｅｘ＿２］が示す処理が、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指示が発行される。一方、イベントデータＥｖｅｎｔ＿２［ＡｕｔｏＩｎｄｅｘ＿２］が示す処理が、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指示が発行される。
【０１２７】
一方、ステップＳ１２０９の判定がＮＯならば、ＣＰＵ２０１は、ステップＳ１２１０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データＥｖｅｎｔ＿２［ＡｕｔｏＩｎｄｅｘ＿２］が示す処理は実行せずに、歌詞に同期した進行のために、次のステップＳ１２１１の処理に進んで、イベントの進行を進める制御処理のみを実行する。
【０１２８】
ステップＳ１２１０の後又はステップＳ１２０９の判定がＮＯの場合に、ＣＰＵ２０１は、トラックチャンク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変数ＡｕｔｏＩｎｄｅｘ＿２の値を＋１インクリメントする（ステップＳ１２１１）。
【０１２９】
次に、ＣＰＵ２０１は、トラックチャンク２に関して今回実行したイベントデータの発

生時刻からの相対時刻を示す変数ＤｅｌｔａＴ＿２の値を０にリセットする（ステップＳ１２１２）。

【０１３０】

そして、ＣＰＵ２０１は、変数ＡｕｔｏＩｎｄｅｘ＿２の値が示す次に実行されるトラックチャンク２上の演奏データ組のＲＡＭ２０３上のタイミングデータＤｅｌｔａＴｉｍｅ＿２［ＡｕｔｏＩｎｄｅｘ＿２］の値が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Ｓ１２１３）。

【０１３１】

ステップＳ１２１３の判定がＮＯならば、ＣＰＵ２０１は、図１２のフローチャートで例示される今回の自動演奏割込み処理を終了する。

【０１３２】

ステップＳ１２１３の判定がＹＥＳならば、ＣＰＵ２０１は、ステップＳ１２０９の処理に戻って、変数ＡｕｔｏＩｎｄｅｘ＿２の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ＲＡＭ２０３上のイベントデータ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Ｓ１２０９からＳ１２１３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。

【０１３３】

図１３は、図８のステップＳ８０５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。

【０１３４】

まずＣＰＵ２０１は、図１２の自動演奏割込み処理におけるステップＳ１２０４で、ＲＡＭ２０３上の変数ＳｏｎｇＩｎｄｅｘにＮｕｌｌ値でない新たな値がセットされて、ソング再生状態になったか否かを判定する（ステップＳ１３０１）。変数ＳｏｎｇＩｎｄｅｘには、ソング開始時は前述した図９（ｃ）のステップＳ９２２でＮｕｌｌ値が初期設定され、歌声の再生タイミングが到来する毎に図１２の自動演奏割込み処理における前述したステップＳ１２０２の判定がＹＥＳとなって、続くステップＳ１２０４で、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変数ＡｕｔｏＩｎｄｅｘ＿１の有効な値がセットされ、更に図１３のフローチャートで例示されるソング再生処理が１回実行される毎に、後述するステップＳ１３０７で再びＮｕｌｌ値にリセットされる。即ち、変数ＳｏｎｇＩｎｄｅｘの値にＮｕｌｌ値以外の有効な値がセットされているか否かは、現在のタイミングがソング再生のタイミングになっているか否かを示すものである。

【０１３５】

ステップＳ１３０１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８のステップＳ８０３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Ｓ１３０２）。

【０１３６】

ステップＳ１３０２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Ｓ１３０３）。

【０１３７】

一方、ステップＳ１３０１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Ｓ１３０２の判定がＮＯ、即ち現時点で新規押鍵が検出されていないと判定された場合には、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データＥｖｅｎｔ＿１［ＳｏｎｇＩｎｄｅｘ］から音高データ（図６のイベントデータ６０６中の音高データ６０７に対応）を読み出し、この音高データ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Ｓ１３０４）。

【０１３８】

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（図６のイベントデータ６０６中の歌詞データ６０８に対応）を読み出す。そして、ＣＰＵ２０１は、読み出した歌詞文字列（図６の演奏時歌詞データ６０９に対応）と、ステップＳ１３０３又はＳ１３０４で取得された発声音高（図６の演奏時音高データ６１０に対応）と、前述した図８のステップＳ８０３に対応する図１０のステップＳ１１１１にてＲＡＭ２０３上の変数ＰｌａｙＴｅｍｐｏに得られた演奏テンポ（図６の演奏時演奏形態データ６１１に対応）がセットされた演奏時歌声データ２１５を、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Ｓ１３０５）。

【０１３９】

続いて、ＣＰＵ２０１は、ステップＳ１３０５で作成した演奏時歌声データ２１５を、図２の音声合成ＬＳＩ２０５の図３の音声合成部３０２に対して発行する（ステップＳ１３０６）。音声合成ＬＳＩ２０５は、図３から図６を用いて説明したように、演奏時歌声データ２１５によって指定される歌詞を、演奏時歌声データ２１５によって指定される演奏者が鍵盤１０１上で押鍵した鍵又はソング再生により音高データ６０７（図６参照）として自動的に指定される音高にリアルタイムに対応し、更に演奏時歌声データ２１５によって指定される演奏テンポ（歌い方）で適切に歌う歌声音声データ２１７を推論、合成して出力する。

【０１４０】

最後に、ＣＰＵ２０１は、変数ＳｏｎｇＩｎｄｅｘ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Ｓ１３０７）。その後、ＣＰＵ２０１は、図１３のフローチャートで例示される図８のステップＳ８０５のソング再生処理を終了する。

【０１４１】

以上のソング再生処理において、特に、ステップＳ１３０２からＳ１３０４の処理は、図６の音高指定部６０２の機能に対応する。また、特に、ステップＳ１３０５の処理は、図６の歌詞出力部６０１の機能に対応する。

【０１４２】

以上説明した一実施形態により、演奏する曲の種類や、演奏フレーズにより、ボーカル音声の子音部の発音時間長が、ゆっくりとしたパッセージの音符の少ない演奏では長く表情豊かな生々しい音にすることができ、テンポが速い、又は音符が多い演奏では、短く歯切れのよい音にすることができる等、演奏フレーズに合った音色変化を得ることが可能となる。

【０１４３】

上述した一実施形態は、歌声音声データを生成する電子楽器の実施形態であったが、他の実施形態として、管楽器音や弦楽器音を生成する電子楽器の実施形態も実施可能である。この場合、図３の音響モデル部３０６に対応する音響モデル部は、音高を指定する学習用音高データとその音高に対応する管楽器や弦楽器の或る音源ソースの音響を示す学習用音響データに対応する教師データと学習用音響データの演奏形態（例えば演奏テンポ）を示す学習用演奏形態データとで機械学習させられ、入力される音高データと演奏形態データとに対応する音響モデルパラメータを出力する学習済み音響モデルを記憶する。また、音高指定部（図６の音高指定部６０２に対応）は、演奏時に演奏者の演奏操作により指定される音高を示す演奏時音高データを出力する。更に、演奏形態出力部（図６の演奏形態出力部６０３に対応）は、上述の演奏時の演奏形態、例えば演奏テンポを示す演奏時演奏形態データを出力する。そして、発音モデル部（図３の発声モデル部３０８に対応）は、演奏時に、上述の演奏時音高データと演奏時演奏形態データとを音響モデル部が記憶する学習済み音響モデルに入力することにより出力される音響モデルパラメータに基づいて、或る音源ソースの音声を推論する楽音データを合成し出力する。このような電子楽器の実施形態においては、例えば速いパッセージの曲では、管楽器の吹き始めのブロー音や弦楽

器の弦を弓で擦る瞬間の弓をあてる速度が短くなるような音高データが推論されて合成されることにより、歯切れのよい演奏が可能となる。逆に、ゆっくりしたパッセージの曲では、管楽器の吹き始めのブロー音、弦を弓で擦る瞬間の弓があたる音の時間が長くなるような音高データが推論されて合成されることにより、演奏表現力の高い演奏が可能となる。

【０１４４】

上述した一実施形態において、初回の押鍵時や演奏フレーズの最初の押鍵のような演奏フレーズの速度が推定できない場合は、強く歌ったり弾いたりした場合は、子音や音の立ち上がり部分は短くなり、弱く歌ったり弾いたりした場合は子音や音の立ち上がり部分は長くなる傾向があることを利用して、鍵盤を弾く強さ（押鍵時のベロシティー値）を演奏テンポの値の算出時のよりどころとして使用してもよい。

【０１４５】

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。

【０１４６】

更に、音声合成方式としては、ＨＭＭ音響モデルを用いた統計的音声合成処理、ＤＮＮ音響モデルを用いた統計的音声合成処理に基づく音声合成方式のほか、ＨＭＭとＤＮＮを組み合わせた音響モデル等、機械学習に基づく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。

【０１４７】

以上説明した実施形態では、演奏時歌詞データ６０９は予め記憶された曲データ６０４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。

【０１４８】

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）

演奏時に指定される演奏時音高データを出力する音高指定部と、

前記演奏時の演奏形態を示す演奏時演奏形態データを出力する演奏形態出力部と、

前記演奏時に、前記演奏時音高データ及び前記演奏時演奏形態データを学習済み音響モデルに入力することにより推論される音響モデルパラメータに基づいて、前記演奏時音高データ及び前記演奏時演奏形態データに対応する楽音データを合成し出力する発音モデル部と、

を備える電子楽器。

（付記２）

演奏時の歌詞を示す演奏時歌詞データを出力する歌詞出力部と、

前記演奏時に前記歌詞の出力に合わせて指定される演奏時音高データを出力する音高指定部と、

前記演奏時の演奏形態を示す演奏時演奏形態データを出力する演奏形態出力部と、

前記演奏時に、前記演奏時歌詞データ、前記演奏時音高データ、及び前記演奏時演奏形態データを学習済み音響モデルに入力することにより推論される音響モデルパラメータに基づいて、前記演奏時歌詞データ、前記演奏時音高データ、及び前記演奏時演奏形態データに対応する歌声音声データを合成し出力する発声モデル部と、

を備える電子楽器。

（付記３）

前記演奏形態出力部は、前記演奏時に前記音高が指定される時間間隔を順次計測し、順次計測された前記時間間隔を示す演奏テンポデータを前記演奏時演奏形態データとして順次出力する、付記１又は２の何れかに記載の電子楽器。

（付記４）

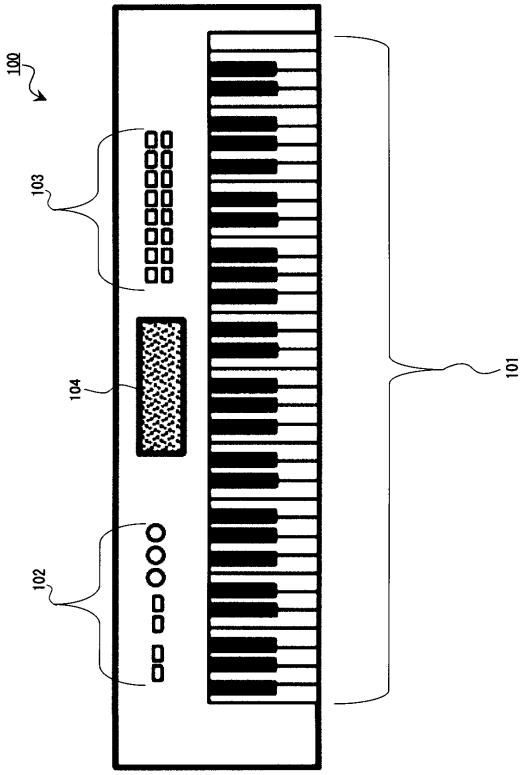
前記演奏形態出力部は、順次得られる前記演奏テンポデータを演奏者に意図的に変更させる変更手段を含む、付記３に記載の電子楽器。

（付記５）
電子楽器のプロセッサに、
演奏時に指定される演奏時音高データを出力し、
前記演奏時の演奏形態を示す演奏時演奏形態データを出力し、
前記演奏時に、前記演奏時音高データ及び前記演奏時演奏形態データを学習済み音響モデルに入力することにより推論される音響モデルパラメータに基づいて、前記演奏時音高データ及び前記演奏時演奏形態データに対応する楽音データを合成し出力する、
処理を実行させる電子楽器の制御方法。
（付記６）
電子楽器のプロセッサに、
演奏時の歌詞を示す演奏時歌詞データを出力し、
前記演奏時に前記歌詞の出力に合わせて指定される演奏時音高データを出力し、
前記演奏時の演奏形態を示す前記演奏時演奏形態データを出力し、
前記演奏時に、前記演奏時歌詞データ、前記演奏時音高データ、及び前記演奏時演奏形態データを学習済み音響モデルに入力することにより推論される音響モデルパラメータに基づいて、前記演奏時歌詞データ、前記演奏時音高データ、及び前記演奏時演奏形態データに対応する歌声音声データを合成し出力する、
処理を実行させる電子楽器の制御方法。
（付記７）
電子楽器のプロセッサに、
演奏時に指定される演奏時音高データを出力し、
前記演奏時の演奏形態を示す演奏時演奏形態データを出力し、
前記演奏時に、前記演奏時音高データ及び前記演奏時演奏形態データを学習済み音響モデルに入力することにより推論される音響モデルパラメータに基づいて、前記演奏時音高データ及び前記演奏時演奏形態データに対応する楽音データを合成し出力する、
処理を実行させるためのプログラム。
（付記８）
電子楽器のプロセッサに、
演奏時の歌詞を示す演奏時歌詞データを出力し、
前記演奏時に前記歌詞の出力に合わせて指定される演奏時音高データを出力し、
前記演奏時の演奏形態を示す前記演奏時演奏形態データを出力し、
前記演奏時に、前記演奏時歌詞データ、前記演奏時音高データ、及び前記演奏時演奏形態データを学習済み音響モデルに入力することにより推論される音響モデルパラメータに基づいて、前記演奏時歌詞データ、前記演奏時音高データ、及び前記演奏時演奏形態データに対応する歌声音声データを合成し出力する、
処理を実行させるためのプログラム。
【符号の説明】
【０１４９】
１００　電子鍵盤楽器
１０１　鍵盤
１０２　第１のスイッチパネル
１０３　第２のスイッチパネル
１０４　ＬＣＤ
２００　制御システム
２０１　ＣＰＵ
２０２　ＲＯＭ
２０３　ＲＡＭ
２０４　音源ＬＳＩ
２０５　音声合成ＬＳＩ
２０６　キースキャナ

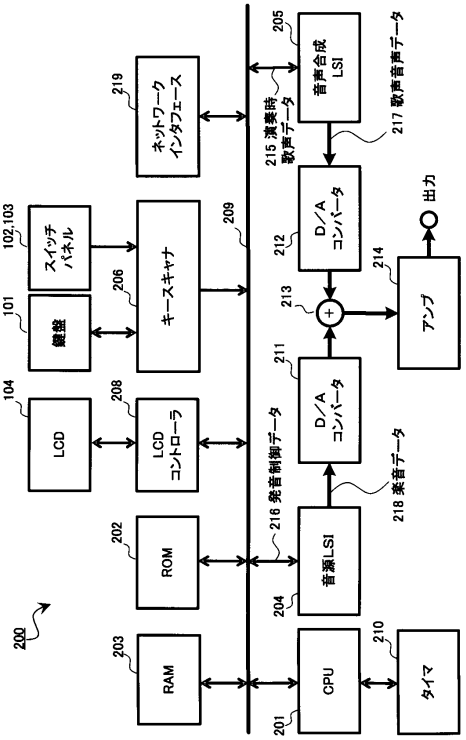
２０８　ＬＣＤコントローラ
２０９　システムバス
２１０　タイマ
２１１、２１２　Ｄ／Ａコンバータ
２１３　ミキサ
２１４　アンプ
２１５　歌声データ
２１６　発音制御データ
２１７　歌声音声データ
２１８　楽音データ
２１９　ネットワークインタフェース
３００　サーバコンピュータ
３０１　音声学習部
３０２　音声合成部
３０３　学習用歌声解析部
３０４　学習用音響特徴量抽出
３０５　モデル学習部
３０６　音響モデル部
３０７　演奏時歌声解析部
３０８　発声モデル部
３０９　音源生成部
３１０　合成フィルタ部
３１１　学習用歌声データ
３１２　学習用歌声音声データ
３１３　学習用言語特徴量系列
３１４　学習用音響特徴量系列
３１５　学習結果データ
３１６　演奏時言語情報量系列
３１７　演奏時音響特徴量系列
３１８　スペクトル情報
３１９　音源情報
６０１　歌詞出力部
６０２　音高指定部
６０３　演奏形態出力部
６０４　曲データ
６０５　タイミングデータ
６０６　イベントデータ
６０７　音高データ
６０８　歌詞データ
６０９　演奏時歌詞データ
６１０　演奏時音高データ
６１１　演奏時演奏形態データ

【図面】

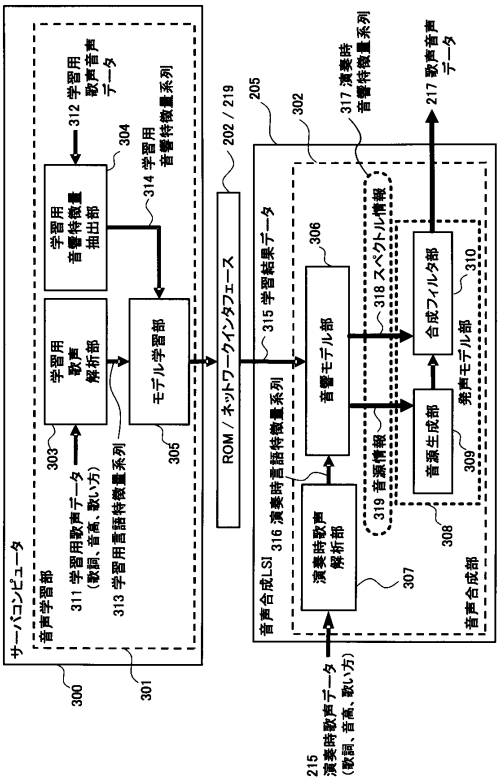
【図１】


100
103
104
102
101

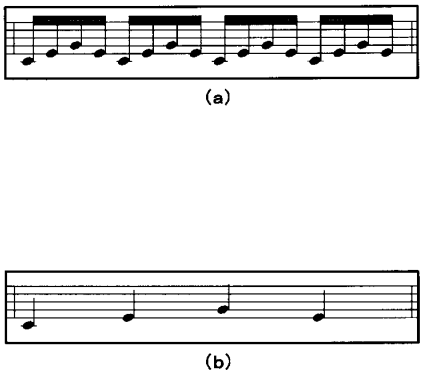
【図２】


200
RAM
ROM
LCD
鍵盤
スイッチパネル
CPU
タイマ
音源LSI
LCDコントローラ
キースキャナ
D/Aコンバータ
D/Aコンバータ
アンプ
出力
ネットワークインタフェース
音声合成LSI
216 発音制御データ
218 楽音データ
215 演奏時歌声データ
217 歌声音声データ

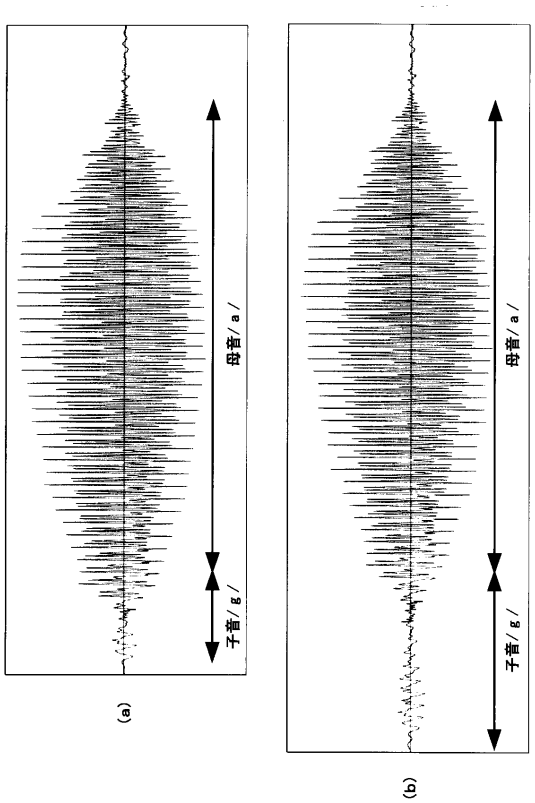
【図３】


サーバコンピュータ
音声学習部
学習用歌声解析部
学習用音響特徴量抽出部
モデル学習部
312 学習用歌声音声データ
314 学習用音響特徴量系列
313 学習用言語特徴量系列
311 学習用歌声データ（歌詞、音高、歌い方）
ROM / ネットワークインタフェース
音声合成LSI
315 学習結果データ
316 演奏時言語特徴量系列
演奏時歌声解析部
音響モデル部
音源生成部
合成フィルタ部
発声モデル部
318 スペクトル情報
319 音源情報
317 演奏時音響特徴量系列
217 歌声音声データ
215 演奏時歌声データ（歌詞、音高、歌い方）
音声合成部

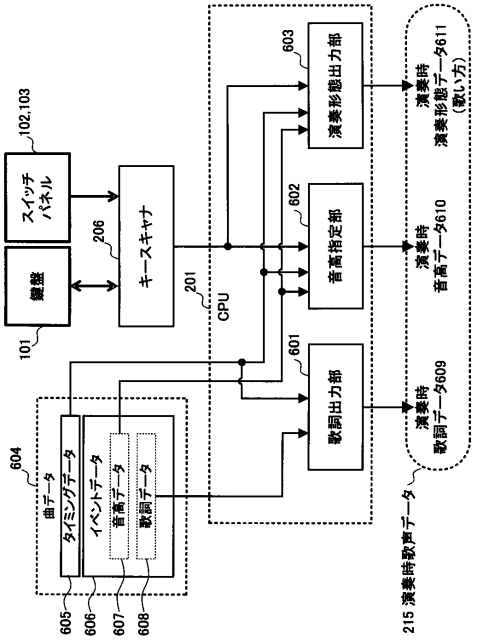
【図４】


(a)
(b)

【図５】



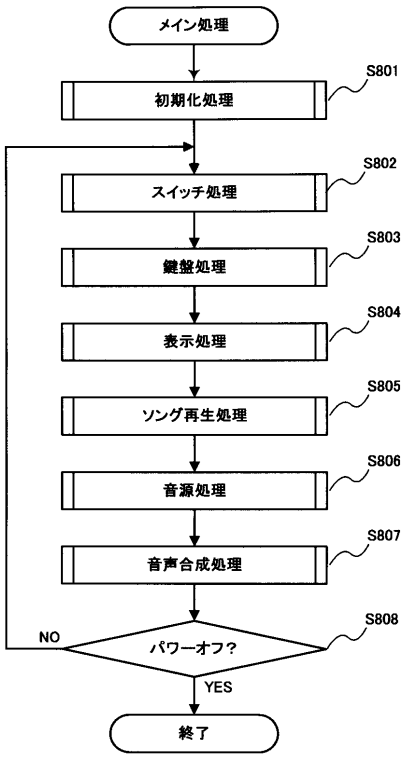
【図６】



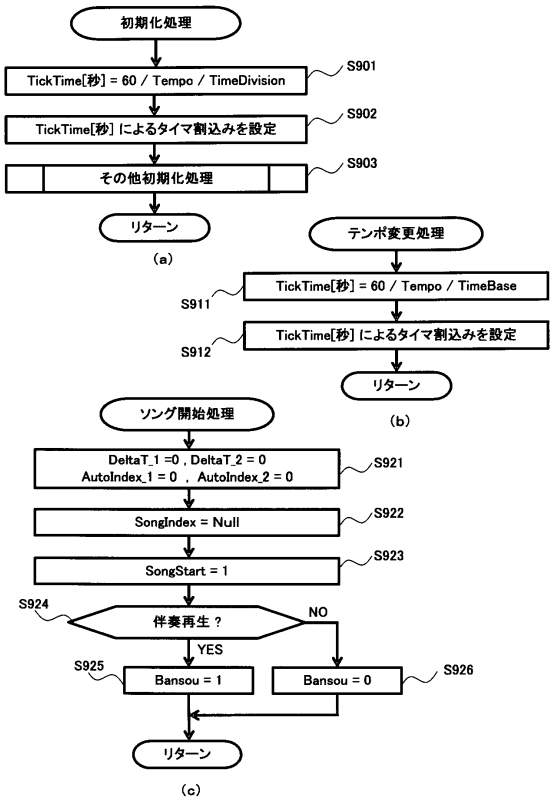
【図７】

ヘッダチャンク		ChunkID	固定文字列"MThd"=0x4d546864
		ChunkSize	ヘッダチャンクの長さ=0x00000006
		FormatType	フォーマット：例えば0x0001
		NumberOfTrack	トラック数：例えば0x0002
		TimeDivision	タイムベース：例えば0x1e0
トラックチャンク1（歌詞パート）		ChunkID	固定文字列"MTrk"=0x4d54726b
		ChunkSize_1	トラックチャンク1の長さ
	DeltaTime_1[0]	DeltaTime	直前イベントからの待ち時間
	Event_1[0]	Event	イベント
	DeltaTime_1[1]	DeltaTime	直前イベントからの待ち時間
	Event_1[1]	Event	イベント
	・・・	・・・	
	DeltaTime_1[L-1]	DeltaTime	直前イベントからの待ち時間
	Event_1[L-1]	Event	トラックの最後は必ず[End of Track]
トラックチャンク2（伴奏パート）		ChunkID	固定文字列"MTrk"=0x4d54726b
		ChunkSize_2	トラックチャンク2の長さ
	DeltaTime_2[0]	DeltaTime	直前イベントからの待ち時間
	Event_2[0]	Event	イベント
	DeltaTime_2[1]	DeltaTime	直前イベントからの待ち時間
	Event_2[1]	Event	イベント
	・・・	・・・	
	DeltaTime_2[M-1]	DeltaTime	直前イベントからの待ち時間
	Event_2[M-1]	Event	トラックの最後は必ず[End of Track]

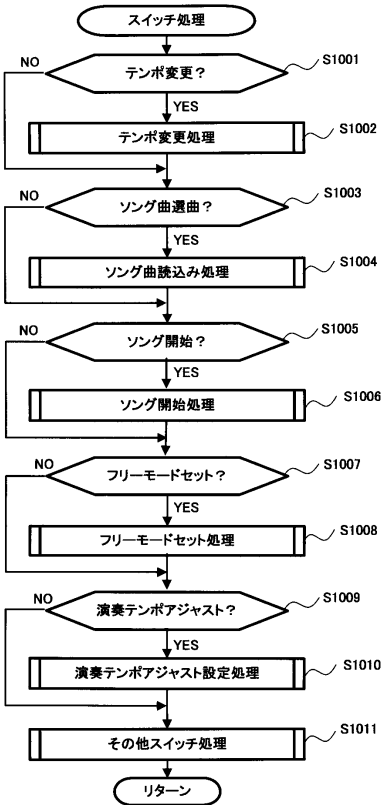
【図８】



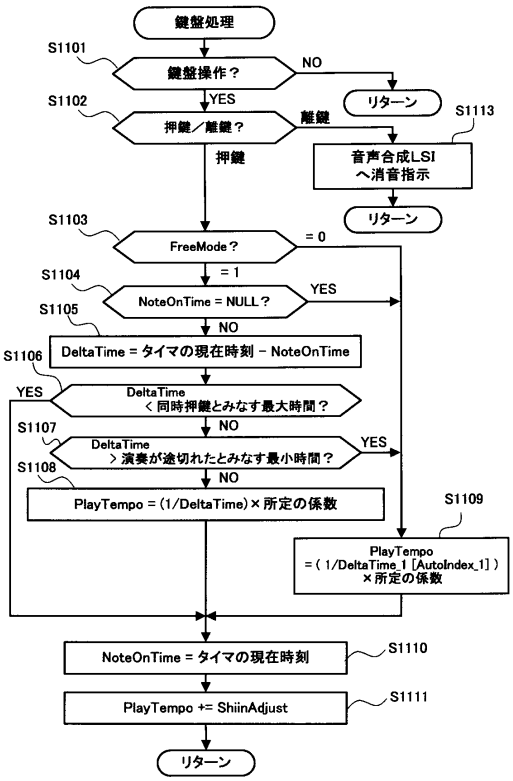
【図９】



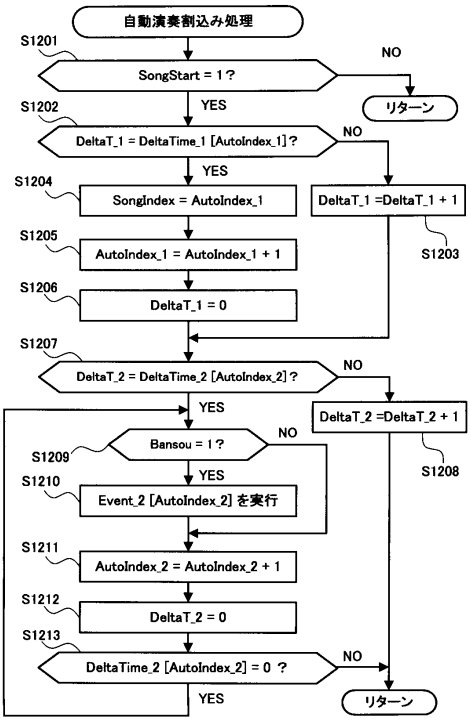
【図１０】



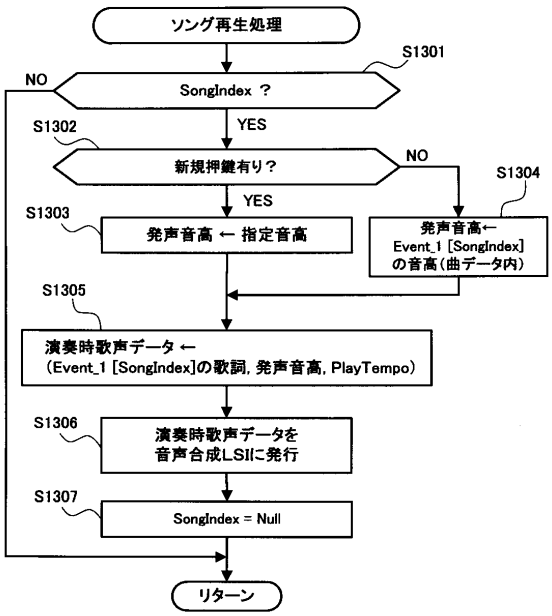
【図１１】



【図１２】



【図１３】


ソング再生処理
S1301
NO
SongIndex ？
YES
S1302
新規押鍵有り？
NO
S1304
YES
S1303
発声音高 ← 指定音高
発声音高←
Event_1 [SongIndex]
の音高（曲データ内）
S1305
演奏時歌声データ ←
（Event_1 [SongIndex]の歌詞, 発声音高, PlayTempo）
S1306
演奏時歌声データを
音声合成LSIに発行
S1307
SongIndex = Null
リターン

フロントページの続き

(56)参考文献　特開２０１９－２１９５６８（ＪＰ，Ａ）
特開２０１７－１０７２２８（ＪＰ，Ａ）
特開２０１５－７５５７４（ＪＰ，Ａ）
国際公開第２０１８／０１６５８１（WO，Ａ１）
特開２０１９－１８４９３５（ＪＰ，Ａ）
米国特許出願公開第２００５／０２６２９８９（ＵＳ，Ａ１）

(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｇ１０Ｈ　１／００－　７／１２
Ｇ１０Ｌ　１３／００－１３／１０