

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3654083号
(P3654083)

(45) 発行日 平成17年6月2日(2005.6.2)

(24) 登録日 平成17年3月11日(2005.3.11)

(51) Int. Cl.⁷

G10H 7/02

F I

G10H 7/00 521F

G10H 7/00 521K

請求項の数 7 (全 25 頁)

(21) 出願番号	特願平11-273176	(73) 特許権者	000004075 ヤマハ株式会社 静岡県浜松市中沢町10番1号
(22) 出願日	平成11年9月27日(1999.9.27)	(74) 代理人	100077539 弁理士 飯塚 義仁
(65) 公開番号	特開2001-100762(P2001-100762A)	(72) 発明者	鈴木 秀雄 静岡県浜松市中沢町10番1号 ヤマハ株式会社内
(43) 公開日	平成13年4月13日(2001.4.13)	(72) 発明者	田邑 元一 静岡県浜松市中沢町10番1号 ヤマハ株式会社内
審査請求日	平成13年6月26日(2001.6.26)	(72) 発明者	宇佐 聡史 静岡県浜松市中沢町10番1号 ヤマハ株式会社内

最終頁に続く

(54) 【発明の名称】 波形生成方法及び装置

(57) 【特許請求の範囲】

【請求項1】

調和成分と調和外成分の各々についてサンプル識別情報と時刻位置情報の組を順次受け取るステップと、

受け取ったサンプル識別情報によりデータベースを参照して調和成分のサンプルデータと調和外成分のサンプルデータを取得するステップと、

受け取った時刻位置情報に従って、前記取得した調和成分のサンプルデータと調和外成分のサンプルデータを時間軸上に配置するステップと、

配置された調和成分のサンプルデータと調和外成分のサンプルデータを混合することにより波形を生成するステップと

を具備する波形生成方法。

【請求項2】

前記配置するステップは、前記調和外成分のサンプルデータを前記調和成分のサンプルデータに同期するように時間軸上に配置するものであることを特徴とする請求項1に記載の波形生成方法。

【請求項3】

前記調和成分のサンプルデータは、波形形状データ、振幅データ、ピッチデータ、時間軸制御データのうちのいずれかの要素についてのサンプルデータであり、前記調和外成分のサンプルデータは、波形形状データ、振幅データ、時間軸制御データのうちのいずれかの要素についてのサンプルデータである請求項1又は2に記載の波形生成方法。

【請求項4】

前記配置するステップでは、各成分毎のサンプルデータにおける前記振幅データ又はピッチデータを、それに対応する前記波形形状データに同期するように時間軸上に配置することを特徴とする請求項3に記載の波形生成方法。

【請求項5】

補正情報を受け取るステップを更に具備し、波形生成のために使用する前記サンプルデータを該補正情報に応じて補正するようにした請求項1乃至4のいずれかに記載の波形生成方法。

【請求項6】

演奏データを受け取るステップと、
前記演奏データによってテーブルを参照して、調和成分と調和外成分の各々についてサンプル識別情報と時刻位置情報の組を取得するステップと、
前記取得した時刻位置情報を前記演奏データに応じて補正するステップと、
前記取得したサンプル識別情報によりデータベースを参照して調和成分のサンプルデータと調和外成分のサンプルデータを取得するステップと、
前記補正した時刻位置情報に従って、前記取得した調和成分のサンプルデータと調和外成分のサンプルデータを時間軸上に配置するステップと、
配置された調和成分のサンプルデータと調和外成分のサンプルデータを混合することにより波形を生成するステップと
を具備する波形生成方法。

10

20

【請求項7】

調和成分と調和外成分の各々についてサンプル識別情報と時刻位置情報の組を受け取る手段と、
前記サンプル識別情報によりデータベースを参照して調和成分のサンプルデータと調和外成分のサンプルデータを取得する手段と、
前記時刻位置情報に従って、前記取得した調和成分のサンプルデータと調和外成分のサンプルデータを時間軸上に配置する手段と
を具備し、時間軸上に配置された調和成分のサンプルデータと調和外成分のサンプルデータを混合することにより波形を生成することを特徴とする波形生成装置。

【発明の詳細な説明】

30

【0001】

【発明の属する技術分野】

この発明は、波形メモリ等からの波形データの読み出し等に基づき、楽音あるいは音声若しくはその他任意の音の波形を生成する方法及び装置に関し、特に、演奏者により行われた自然楽器固有の各種奏法若しくはアーティキュレーションによる音色変化を忠実に表現した波形を生成するものに関する。この発明は、電子楽器は勿論のこと、自動演奏装置、コンピュータ、電子ゲーム装置その他のマルチメディア機器等、楽音あるいは音声若しくはその他任意の音を発生する機能を有するあらゆる分野の機器若しくは装置または方法において広範囲に適用できるものである。なお、この明細書において、楽音波形という場合、音楽的な音の波形に限るものではなく、音声あるいはその他任意の音の波形を含んでい

40

【0002】

【従来の技術】

波形メモリにおいて、PCM（パルス符号変調）あるいはDPCM（差分PCM）又はADPCM（適応差分PCM）等の任意の符号化方式で符号化した波形データ（つまり波形サンプルデータ）を記憶しておき、これを所望の音楽ピッチに対応して読み出すことにより、楽音波形を形成するようにした、いわゆる「波形メモリ読み出し」技術は既に公知であり、また、様々なタイプの「波形メモリ読み出し方式」技術が知られている。従来知られた「波形メモリ読み出し方式」技術のほとんどは、発音開始から終了までの1つの音の波形を発生するためのものである。一例として、発音開始から終了までの1音の全波形の

50

波形データを記憶する方式がある。また、別の例として、変化の複雑なアタック部などについてはその全波形の波形データを記憶し、変化のあまりないサステイン部などについては所定のループ波形を記憶する方式がある。なお、本明細書において、「ループ波形」とは繰り返し読み出し（ループ読み出し）される波形という意味で用いるものとする。

【0003】

【発明が解決しようとする課題】

ところで、従来の発音開始から終了までの1音の全波形の波形データを記憶する方式やアタック部などの波形の一部において全波形の波形データを記憶する方式の「波形メモリ読み出し方式」技術においては、各種奏法（若しくはアーティキュレーション）に対応する様々な波形データを多数記憶しておかなければならず、この多数の波形データを記憶する

10

ために大きな記憶容量が必要であった。また、上述の全波形の波形データを記憶する方式では、自然楽器固有の各種奏法（若しくはアーティキュレーション）による音色変化を忠実に表現することが可能であるが、記憶した波形データの通りしか楽音を再生することができないので、制御性に乏しく、また、編集性にも乏しかった。例えば、所望の奏法（若しくはアーティキュレーション）に対応する波形データを演奏データに応じた時間軸制御等の特性制御を行うことが非常に困難であった。

【0004】

本発明は上述の点に鑑みてなされたもので、様々な奏法（若しくはアーティキュレーション）に対応する高品質な波形データを容易かつ簡便にまた制御性豊かに生成することのできる波形生成技術を提供することを目的とする。特に、本発明は、高品質な波形（すなわち奏法又はアーティキュレーションを表現した波形）に対応する波形データをそっくりそのまま記憶しておくのではなく、波形形状の変化表現に必要な一部の波形（サンプルデータ）のみを抽出して記憶しておき、これを用いて波形生成を行うことによって、波形データを記憶するために必要とする記憶量を節約することのでき、しかも波形再現性を確保すると共にその際の制御性を豊かにした、波形生成方法並びに装置を提供しようとするものである。

20

【0007】

【課題を解決するための手段】

本発明に係る波形生成方法は、調和成分と調和外成分の各々についてサンプル識別情報と時刻位置情報の組を順次受け取るステップと、受け取ったサンプル識別情報によりデータベースを参照して調和成分のサンプルデータと調和外成分のサンプルデータを取得するステップと、受け取った時刻位置情報に従って、前記取得した調和成分のサンプルデータと調和外成分のサンプルデータを時間軸上に配置するステップと、配置された調和成分のサンプルデータと調和外成分のサンプルデータを混合することにより波形を生成するステップとを具備する。この場合、時刻位置情報と組をなすのは、サンプルデータそのものではなく、サンプル識別情報（サンプルID、つまりサンプルデータを指示するデータ）である。このサンプル識別情報によってデータベースから調和成分のサンプルデータと調和外成分のサンプルデータが得られる。この調和成分のサンプルデータと調和外成分のサンプルデータは、時刻位置情報に従って適切な時刻位置に配置される、従って、波形データの記憶量節約と精巧な波形再現性の確保を実現することができ、また、調和成分と調和外成分とを別々に制御できるため制御性にも富むものとなる。

30

40

【0008】

前記調和成分のサンプルデータは、波形形状データ、振幅データ、ピッチデータ、時間軸制御データのうちのいずれかの要素についてのサンプルデータであり、前記調和外成分のサンプルデータは、波形形状データ、振幅データ、時間軸制御データのうちのいずれかの要素についてのサンプルデータである。実施例において、これらの各要素についてのサンプルデータはベクトルデータとも呼ばれている。すなわち、ベクトルは、当該波形を生成するための異なる種類の基本的な要素に対応している。その種の基本的な要素としては、例えば、波形形状（音色若しくはティンバーを設定する波形形状）、ピッチの時間的変化、あ

50

るいは振幅の時間的変化などがあり、それらのベクトルを波形ベクトル、ピッチベクトル、振幅ベクトルと呼ぶ。更には、波形の時間軸の進行を伸縮制御するタイムベクトルが含まれていてもよい。このタイムベクトルによって、波形ベクトル、ピッチベクトル、振幅ベクトルなどの時間軸を制御することができる。

補正情報を受け取るステップを更に具備し、波形生成のために使用する前記サンプルデータを該補正情報に応じて補正するようにしてもよい。そうすれば、例えば、サンプル識別情報が指示する通りのサンプルデータが予め用意（記憶）されていなかったとしても、サンプル識別情報の指示に近いサンプルデータを取りあえず取得して、これを補正情報に応じて補正することで希望のサンプルデータを得ることができる。よって、サンプルデータを予め用意（記憶）するための構成を簡略化することができる、すなわち記憶量を節約することができる。また、サンプル識別情報が指示する通りのサンプルデータが予め用意（記憶）されている場合であっても、補正情報に応じて補正することでサンプルデータを自由に変形することができ、制御性に富むものとなる。

10

【0009】

更に、本発明に係る波形生成方法は、演奏データを受け取るステップと、前記演奏データによってテーブルを参照して、調和成分と調和外成分の各々についてサンプル識別情報と時刻位置情報の組を取得するステップと、前記取得した時刻位置情報を前記演奏データに応じて補正するステップと、前記取得したサンプル識別情報によりデータベースを参照して調和成分のサンプルデータと調和外成分のサンプルデータを取得するステップと、前記補正した時刻位置情報に従って、前記取得した調和成分のサンプルデータと調和外成分の サンプルデータを時間軸上に配置するステップと、配置された調和成分のサンプルデータと調和外成分のサンプルデータを混合することにより波形を生成するステップとを具備する。この場合は、演奏データに基づき調和成分と調和外成分の各々についてサンプル識別情報と時刻位置情報の組が取得される。予め用意された（例えば記憶された）サンプル識別情報と時刻位置情報の多数の組の中に、演奏データが意図する演奏音に最適の組がないかもしれない。しかし、そのような場合でも、演奏データの意図するものに近い組を選択して、そこで取得した時刻位置情報を演奏データに応じて補正することで、演奏データの意図する時刻位置情報を得ることができる。一方、演奏データがタッチやベロシティのような可変制御ファクタを含む場合は、その可変制御ファクタに応じて時刻位置情報を補正することで、演奏データに応じた時刻位置情報の可変制御を行うことができる。従って、時刻位置情報を予め用意（記憶）するための構成を簡略化することができる、すなわち記憶量を節約することができる。また、波形を特徴づけるサンプルデータの時刻位置を可変制御することができ、制御性に富むものとなる。また、調和成分と調和外成分とを別々に制御できるため、この点でも制御性にも富むものとなる。

20

30

【0010】

実施例において、演奏データは、奏法識別情報（以下、奏法IDともいう）を含む形態で与えられる。例えば、奏法識別情報は、演奏音の奏法的特徴に応じて、アタックやボディあるいはリリース等の音の部分的区間に対応して与えられることがあり、また、スラーのような音と音のつながりの区間（ジョイント部）に対応して与えられることもあれば、ピブラートのような音の特殊演奏部分に対応して与えられることもあり、また、フレーズのように複数音符に対応して与えられることもある。再生すべき演奏に応じて、所要の奏法識別情報が与えられる。

40

本発明は、方法の発明として構成し実施することができるのみならず、装置の発明として構成し実施することができる。また、本発明は、コンピュータまたはDSP等のプロセッサのプログラムの形態で実施することができるし、そのようなプログラムを記憶した記憶媒体の形態で実施することもできる。

【0011】

【発明の実施の形態】

以下、この発明の実施の形態を添付図面に従って詳細に説明する。

【0012】

50

図1は、この発明に係る波形生成装置のハードウェア構成例を示すブロック図である。ここに示されたハードウェア構成例はコンピュータを用いて構成されており、そこにおいて、波形生成処理は、コンピュータがこの発明に係る波形生成処理を実現する所定のプログラム(ソフトウェア)を実行することにより実施される。勿論、この波形生成処理はコンピュータソフトウェアの形態に限らず、DSP(デジタル・シグナル・プロセッサ)によって処理されるマイクロプログラムの形態でも実施可能であり、また、この種のプログラムの形態に限らず、ディスクリート回路又は集積回路若しくは大規模集積回路等を含んで構成された専用ハードウェア装置の形態で実施してもよい。また、この波形生成装置は、電子楽器あるいはカラオケ装置又は電子ゲーム装置又はその他のマルチメディア機器又はパーソナルコンピュータ等、任意の製品応用形態をとってよい。

10

【0013】

図1に示されたハードウェア構成例においては、コンピュータのメイン制御部としてのCPU101に対して、バスラインBL(データあるいはアドレスバス等)を介してリードオンリメモリ(ROM)102、ランダムアクセスメモリ(RAM)103、パネルスイッチ104、パネル表示器105、ドライブ106、波形取込部107、波形出力部108、ハードディスク109、通信インタフェース111がそれぞれ接続されている。CPU101は、後述する「波形データベース作成」や「制作したデータベースに基づく楽音合成(ソフトウェア音源)」等の処理を、所定のプログラムに基づいて実行する。これらのプログラムは、通信インタフェース111を介したネットワークあるいはドライブ106に装着されたCDやMO等の外部記憶メディア106A等から供給されてハードディスク109に記憶される。そして、実行時にハードディスク109からRAM103にロードされる。あるいは、ROM102にプログラムが記録されていてもよい。ROM102は、CPU101により実行あるいは参照される各種プログラムや各種データ等を格納するものである。ROM103は、演奏に関する各種情報やCPU101がプログラムを実行する際に発生する各種データを一時的に記憶するワーキングメモリとして、あるいは現在実行中のプログラムやそれに関連するデータを記憶するメモリとして使用される。RAM103の所定のアドレス領域がそれぞれの機能に割り当てられ、レジスタやフラグ、テーブル、メモリなどとして利用される。パネルスイッチ104は、楽音をサンプリングする指示やサンプリングされた波形データ等のエディットや各種情報の入力等を行うための各種の操作子を含んで構成される。例えば、数値データ入力用のテンキーや文字データ入力用のキーボード、あるいはパネルスイッチ等である。この他にも音高、音色、効果等を選択・設定・制御するための各種操作子を含んでよい。パネル表示器105は、パネルスイッチ104により入力された各種情報やサンプリングされた波形データ等を表示する、例えば液晶表示パネル(LCD)やCRT等のディスプレイである。

20

30

【0014】

波形取込部107はA/D変換器を内蔵し、外部波形入力(例えば、マイクロフォンなどからの入力)されたアナログ楽音信号をデジタルデータに変換(サンプリング)してRAM103あるいはハードディスク109に該デジタル波形データをオリジナル波形データ(生成すべき波形データの素材となる波形データ)として取り込むものである。CPU101によって実行する「波形データベース作成」処理では、上記取り込んだオリジナル波形データを基にして本発明に従う「波形データベース」の作成を行う。また、CPU101によって実行する「データベースに基づく楽音合成」処理では、上記「波形データベース」を使用して演奏情報に応じた任意の楽音信号の波形データを生成する。勿論、複数の楽音信号の同時発生が可能である。生成された楽音信号の波形データはバスラインBLを介して波形出力部108に与えられ、適宜バッファ記憶される。波形出力部108ではバッファ記憶された波形データを所定の出力サンプリング周波数にしたがって出力し、これをD/A変換してサウンドシステム108Aに送出する。こうして、波形出力部108から出力された楽音信号は、サウンドシステム108Aを介して発音される。ハードディスク109は、波形データや奏法に応じた波形を合成するためのデータ(後述する奏法テーブル、コードブック等のデータ)、各種音色パラメータ等からなる音色データなどのよう

40

50

な演奏に関する複数種類のデータを記憶したり、前記CPU101が実行する各種プログラム等の制御に関するデータを記憶したりするものである。

【0015】

ドライブ106は、波形データや奏法に応じた波形を合成するためのデータ（後述する奏法テーブル、コードブック等の各種データ）、多種多様な音色パラメータ等からなる音色データなどのような演奏に関する複数種類のデータを記憶したり、前記CPU101が実行する各種プログラム等の制御に関するデータを記憶したりするための着脱可能なディスク（外部記憶メディア106A）をドライブするものである。なお、前記ドライブ106によりドライブされる外部記憶メディア106Aはフロッピーディスク（FD）の他に、コンパクトディスク（CD-ROM・CD-RAM）、光磁気ディスク（MO）、あるいはDVD（Digital Versatile Diskの略）等の着脱自在な様々な形態の記憶メディアであってよい。制御プログラムを記憶した外部記憶メディア106Aをドライブ106にセットし、その内容（制御プログラム）をハードディスク109に落とさずに、RAM103に直接ロードしてもよい。なお、外部記憶メディア106Aを用いて、あるいはネットワークを介して制御プログラムを提供するやり方は、制御プログラムの追加やバージョンアップ等を容易に行うことができるので好都合である。

10

【0016】

通信インタフェース111は、例えばLANやインターネット、電話回線等の通信ネットワーク（図示せず）に接続されており、該通信ネットワークを介して、サーバコンピュータ等（図示せず）と接続され、当該サーバコンピュータ等から制御プログラムや各種データあるいは演奏情報などを波形生成装置側に取り込むためのものである。すなわち、ROM102やハードディスク109に制御プログラムや各種データが記憶されていない場合に、サーバコンピュータから制御プログラムや各種データをダウンロードするために用いられる。クライアントとなる波形生成装置は、通信インタフェース111を介してサーバコンピュータへと制御プログラムや各種データのダウンロードを要求するコマンドを送信する。サーバコンピュータは、このコマンドを受け、要求された制御プログラムやデータなどを通信インタフェース111を介してハードディスク109に蓄積することにより、ダウンロードが完了する。更に、MIDIインタフェースを含み、MIDIの演奏情報を受け取るようにしてもよいのは勿論である。また、音楽演奏用キーボードや演奏操作機器をバスラインBLに接続し、リアルタイム演奏によって演奏情報を供給するようにしてもよいのは言うまでもない。勿論、所望の音楽曲の演奏情報を記憶した外部記憶メディア106Aを使用して、演奏情報を供給するようにしてもよい。

20

30

【0017】

図2は、上述した波形生成装置において実行される「波形データベース作成処理」の一実施例を示すフローチャートである。当該処理は、いろいろな奏法（若しくはアーティキュレーション）に対応するために、いろいろな奏法（若しくはアーティキュレーション）で演奏された演奏音の波形を素材としてベクトルデータを作成するための処理である。ステップS1では、後述する奏法テーブル及びコードブックを記憶するためのデータベースを準備する。このデータベースとなる媒体としては、例えばハードディスク109を使用する。そして、様々な自然楽器の様々な演奏態様による波形データを収集する（ステップS2）。すなわち、様々な自然楽器の様々な実際の演奏音を外部波形入力（例えば、マイクロフォン等）から波形取込部107を介して取り込み、それらの演奏音の波形データ（オリジナル波形データ）をハードディスク109の所定のエリアに記憶する。この際に取り込む演奏音の波形データは演奏全体の波形データであってもよいし、あるフレーズ、あるいは1音、あるいはアタック部やリリース部といった特徴のある演奏の一部の波形データだけであってもよい。次に、こうして得られた自然楽器固有の様々な演奏態様による演奏音の波形データを特徴的な部分毎に切り分けて、チューニング及びファイル名付けする（ステップS3）。すなわち、取り込んだオリジナル波形データを波形形状の変化を代表する一部の波形（例えば、アタック部波形、ボディ部波形、リリース部波形、ジョイント部波形等）毎に分離して（1 切り分け）、分離した1周期乃至複数周期の波形デー

40

50

タがそれぞれいかなるピッチであるかを判定し(2 チューニング)、さらにそれぞれ分離した波形データに対してファイル名を付与する(3 ファイル名付け)。ただし、アタック部分やリリース部分といった演奏の一部の波形データを取り込んでいる場合には、このような波形の分離(1 切り分け)を省略できる。

次に、周波数分析による成分分離を行う(ステップS4)。すなわち、ステップS3で分離生成された一部の波形データをFFT(高速フーリエ変換)分析して複数成分に分離し(この実施例では、調和成分と調和外成分に分離する)、さらに各成分(調和成分、調和外成分等)から波形、ピッチ、振幅の各要素毎の特徴抽出、つまり特徴分離を行う(ただし、調和成分と調和外成分に分離する場合、調和外成分はピッチを持たないものであることから調和外成分についてのピッチ分離は行わなくてよい)。例えば「波形」(Timbre)要素は、ピッチと振幅をノーマライズした波形形状のみ特徴を抽出したものである。「ピッチ」(Pitch)要素は、基準ピッチに対するピッチ変動特性を抽出したものである。「振幅」(Amplitude)要素は、振幅エンベロープ特性を抽出したものである。

【0018】

ステップS5では、ベクトルデータの作成が行われる。すなわち、分離された各成分(調和成分、調和外成分等)の波形(Timbre)やピッチ(Pitch)や振幅(Amplitude)の各要素毎に複数のサンプル値を分散的に又は必要に応じて連続的に抽出し、当該サンプル値列に対して各々異なったベクトルID(識別情報)を付与して、サンプル値の時刻位置のデータとともにコードブックに記憶する(以下、このようなサンプルデータをベクトルデータと呼ぶ)。この実施例では、調和成分の波形(Timbre)要素のベクトルデータ、ピッチ(Pitch)要素のベクトルデータ、振幅(Amplitude)要素のベクトルデータと、調和外成分の波形(Timbre)要素のベクトルデータ、振幅(Amplitude)要素のベクトルデータとがそれぞれ作成される。このように、これらの各成分要素毎のベクトルデータは、時間軸の進行に伴い変化しうるデータである。次に、奏法モジュールのデータ(詳しい内容については後述する)を作成して奏法モジュールを奏法テーブルに記憶する。こうして作成された奏法モジュール及びベクトルデータは、データベースにおける奏法テーブル及びコードブックへ書き込まれ(ステップS6)、データベースへのデータ蓄積がはかれる。上述したように、ベクトルデータは取り込んだオリジナル波形データそのままではなく取り込んだオリジナル波形の形状を代表する波形を各要素毎に分離したデータであって、このベクトルデータは各々が最終的には奏法モジュールの単位となるデータである。このように、コードブックには抽出した波形形状の変化を代表する一部の波形データを圧縮した形で記憶する。一方、奏法テーブルには、奏法モジュールのデータ(つまり、圧縮された形で記憶されたベクトルデータを元の波形形状の波形データに戻すために必要な各種のデータや、コードブックに記憶されたベクトルデータを指定するためのIDデータ)などが記憶される(詳しくは後述する)。

【0019】

上述した特徴分離(ステップS4参照)の際に、振幅、ピッチ、波形要素の他に時間を要素として特徴抽出を行う(以下、抽出された時間要素のベクトルデータのことを「タイムベクトルデータ」と呼ぶ)。この時間要素については、分離生成された一部の波形データの時間区間におけるオリジナル波形データの時間長をそのまま用いる。従って、当該時間区間のオリジナルの時間長(可変値である)を比「1」で示すこととすれば、当該「波形データベース作成処理」時においてこの時間長をあえて分析・測定する必要はない。その場合、時間要素についてのデータ(すなわち、「タイムベクトルデータ」)はどの時間区間でも同じ値“1”であるから、これをコードブックにあえて記憶しておかなくてもよい。勿論、これに限らず、この実際の時間長を分析・測定し、これを「タイムベクトルデータ」としてコードブックに記憶するようにする変形例も実施可能である。

【0020】

そして、データベース作成が充分に行われたか否かを判定する(ステップS7)。すなわち、外部波形入力から得られた様々な自然楽器の様々な演奏態様による演奏音のオリジナル波形データの収集を充分に行って、様々な奏法モジュールのデータ及びベクトルデータ

10

20

30

40

50

を十分に得たか否かを判定する。この判定は自動判定に限らず、ユーザによるスイッチ入力操作に基づく処理続行可否指示に従って行うようにしてもよい。オリジナル波形データの収集とそれに基づくベクトルデータの作成が充分に行われたと判定されたならば（ステップS7のYES）、当該処理を終了する。引き続き、オリジナル波形データの収集とそれに基づくベクトルデータの作成を行う場合（ステップS7のNO）、ステップS2の処理へ戻り、上述した各処理（ステップS2～ステップS7）を繰り返し実行する。上記「ベクトルデータの作成が充分に行われたか否か」の判定（ステップS7）は、「作成したベクトルデータを実際を使用して楽音を生成してみる」ことにより行われてもよい。すなわち、ステップS7で一旦「ベクトルデータを充分作成した」（ステップS7のYES）と判断して図2に示すフローを抜けた後に、「その作成したベクトルデータを使用して楽音を再生してみたら満足できなかったので、再びステップS2以降の処理を行ってベクトルデータを追加する」というような処理を行ってもよい。つまり、「ベクトルデータを作成してデータベースに追加する」という処理は、必要に応じて随時行われる。

10

なお、上述の「波形データベース作成処理」において、奏法モジュールを任意に追加・削除したり、あるいは奏法モジュールのデータ等の編集を行うことができるようにしてもよい。

【0021】

ここで、奏法モジュールのデータについて具体的に説明する。

奏法モジュールはハードディスク109上にデータベース化されて構成される奏法テーブルに記憶され、1つの奏法モジュールは「奏法ID」と「奏法パラメータ」の組み合わせによって指定することができるようになっている。「奏法ID」は、その中に楽器情報及びモジュールパーツ名を含む。例えば、「奏法ID」は次のように定義される。例えば、1つの「奏法ID」が32ビット（第0～31ビット）列で表現されているとすると、そのうちの6ビットを使用して楽器情報を表現する。例えば、当該6ビット列が「000000」であればAltoSax（アルト・サクス）を示し、「001000」であればViolin（バイオリン）を示す楽器情報である。この楽器情報は前記6ビット列のうち上位3ビット列を楽器種類の分類に使用し、下位3ビット列を楽器種類の小分類のために使用するなどしてよい。また、32ビット列の別の6ビット列を使用してモジュールパーツ名を表現する。例えば、当該6ビット列が「000000」であればNormalAttack、「000001」であればBendAttack、「000010」であればGraceNoteAttack、「001000」であればNormalShortBody、「001001」であればVibBody、「001010」であればNormalLongBody、「010000」であればNormalRelease、「011000」であればNormalJoint、「011001」であればGraceNoteJointを示すモジュールパーツ名である。勿論、上述した構成に限られないことは言うまでもない。

20

30

【0022】

上述したように、個々の奏法モジュールは、上記「奏法ID」と「奏法パラメータ」との組み合わせで特定される。すなわち、「奏法ID」に応じて所定の奏法モジュールが特定され、その内容が「奏法パラメータ」に応じて可変設定される。この「奏法パラメータ」は該奏法モジュールに対応する波形データを特徴付ける、若しくは制御するパラメータであり、各奏法モジュール毎に所定の種類の「奏法パラメータ」が存在している。例えば、AltoSax [NormalAttack]モジュールの場合にはAttack直後の絶対音高やAttack直後の音量などの種類の奏法パラメータが与えられてよいし、AltoSax [BendUpAttack]モジュールの場合にはBendUpAttack終了時の絶対音高、BendUpAttack時のBend深さの初期値、BendUpAttack開始（ノートオンタイミング）～終了までの時間、Attack直後の音量、あるいはBendUpAttack中のデフォルトのカーブの時間的な伸縮などの種類の奏法パラメータが与えられてよい。また、AltoSax [NormalShortBody]モジュールの場合には当該モジュールの絶対音高、NormalShortBodyの終了時刻 - 開始時刻、NormalShortBody開始時のダイナミクス、NormalShortBody終了時のダイナミクスなどの種類の奏法パラメータが与えられてよい。なお、奏法モジュールには、「奏法パラメータ」の採りうる全ての値に対応するデータ（後述する要素データ）を必ずしも有しない。「奏法パラメータ」の飛び飛びの一部の値だ

40

50

けに応じたデータを記憶している場合もある。すなわち、例えばAltoSax [NormalAttack] モジュールの場合、Attack直後の絶対音高やAttack直後の音量の全ての値ではなく、一部のデータだけに対応したデータを記憶していてもよい。

このように、奏法モジュールを「奏法ID」と「奏法パラメータ」で指定できるようにすることで、例えばAltoSax [NormalAttack]であればアルトサックスのノーマルアタック部を示す複数データ（後述する要素データ）の中から所望の奏法パラメータに応じたデータを指定することができるし、Violin[BendAttack]であればバイオリンのバンドアタック部を示す複数データ（後述する要素データ）の中から所望の奏法パラメータに応じたデータを指定することができる。

【0023】

奏法テーブルにおいては、個々の奏法モジュールにつき、当該奏法モジュールに対応する波形を生成するために必要なデータ、例えば各成分要素毎のベクトルデータ（波形要素、ピッチ要素（ピッチエンベロープ）、振幅要素（振幅エンベロープ）等）を指定するためのベクトルIDや代表点値列（複数サンプル列の中の補正のための代表的サンプル点を指示するデータ）あるいは各成分要素毎のベクトルデータ（波形要素、ピッチ要素（ピッチエンベロープ）、振幅要素（振幅エンベロープ））の開始時間位置や終了時間位置などの情報等を記憶している。つまり、ベクトルデータという圧縮された形でデータベースに記憶されている波形から通常形状の波形を再生するために必要な各種のデータを記憶している（以下、このようなデータを「要素データ」とも呼ぶ）。奏法テーブルにおいて、1つの奏法モジュールに対応して記憶する具体的なデータの一例をAltoSax [NormalAttack]モ

ジュールの場合について説明すると、次の通りである。

データ1：奏法モジュールのサンプル長。

データ2：ノートオンタイミングの位置。

データ3：調和成分の振幅（Amplitude）要素のベクトルIDと代表点値列。

データ4：調和成分のピッチ（Pitch）要素のベクトルIDと代表点値列。

データ5：調和成分の波形（Timbre）要素のベクトルID。

データ6：調和外成分の振幅（Amplitude）要素のベクトルIDと代表点値列。

データ7：調和外成分の波形（Timbre）要素のベクトルID。

データ8：調和成分の波形（Timbre）要素の塊部の開始位置。

データ9：調和成分の波形（Timbre）要素の塊部の終了位置（調和成分の波形（Timbre）

要素のループ部の開始位置）。

データ10：調和外成分の波形（Timbre）要素の塊部の開始位置。

データ11：調和外成分の波形（Timbre）要素の塊部の終了位置（調和外成分の波形（Timbre）要素のループ部の開始位置）。

データ12：調和外成分の波形（Timbre）要素のループ部の終了位置。

【0024】

上記データ1～12について、図3を参照して説明する。

図3は、当該奏法モジュールに対応する実波形区間を構成する各成分及び要素の一例を模式的に示す図であり、上から当該区間における調和成分の振幅（Amplitude）要素、調和成分のピッチ（Pitch）要素、調和成分の波形（Timbre）要素、調和外成分の振幅（Amplitude）要素、調和外成分の波形（Timbre）要素の一例を示す。なお、図に示している数字は上記各データの番号に対応するように付してある。

1は、当該奏法モジュールに該当する波形のサンプル長（波形区間長）である。例えば、当該奏法モジュールの基となったオリジナル波形データの全体の時間長さに対応している。2はノートオンタイミングの位置であり、当該奏法モジュールのどの時間位置にも可変に設定することが可能である。原則的には、このノートオンタイミングの位置から当該波形に従った演奏音の発音が開始されるが、バンドアタックなどの奏法によってはノートオンタイミングよりも波形成分の立ち上がり開始時点が先行する場合がある。3は、コードブックに記憶された調和成分の振幅（Amplitude）要素のベクトルデータを指し示すためのベクトルID及び代表点値列を示す（図において、黒く塗りつぶした正方形で示す2点

10

20

30

40

50

が代表点を示す)。4は、調和成分のピッチ(Pitch)要素のベクトルデータを指し示すためのベクトルID及び代表点値列を示す。6は、調和外成分の振幅(Amplitude)要素のベクトルデータを指し示すためのベクトルID及び代表点値列を示す。代表点値列データはベクトルIDによって指示されるベクトルデータ(複数サンプル列からなる)を変更制御するためのデータであり、代表的サンプル点のいくつかを指示(特定)するものである。特定された代表的サンプル点に関してその時間位置(横軸)とレベル軸(縦軸)を変更若しくは補正することにより、他の残りのサンプル点も連動して変更し、もってベクトルの形状を変更する。例えば、そのサンプル数より少ない数の分散的サンプルを示すデータであるが、勿論これに限らず、代表点値列データはサンプルとサンプルの間の中間位置のデータであってもよいし、あるいは所定の範囲(連続的な複数サンプル)にわたるデータであってもよい。また、サンプル値そのものでなく、差分や乗数等のデータであってもよい。この代表点を横軸及び/又は縦軸(時間軸)に移動することによって、各ベクトルデータの形状を変えることができる。つまり、エンベロープ波形の形状を変えることができる。5は、調和成分の波形(Timbre)要素のベクトルデータを指し示すためのベクトルIDである。7は、調和外成分の波形(Timbre)要素のベクトルデータを指し示すためのベクトルIDである。8は、調和成分の波形(Timbre)要素の波形の塊部の開始位置である。9は、調和成分の波形(Timbre)要素の波形の塊部の終了位置(あるいは、調和成分の波形(Timbre)要素の波形のループ部の開始位置)である。すなわち、8から開始する三角形は特徴のある波形形状が連続的に記憶されているノンループ波形の部分を示し、その後続く9から開始する長方形は繰り返し読み出しすることのできるループ波形の部分を示す。ノンループ波形は、奏法(若しくはアーティキュレーション)等の特徴を有する高品質な波形である。ループ波形は、1周期または適当な複数周期分の波形からなる比較的単調な音部分の単位波形である。10は、調和外成分の波形(Timbre)要素の波形の塊部の開始位置である。11は、調和外成分の波形(Timbre)要素の波形の塊部の終了位置(あるいは、調和外成分の波形(Timbre)要素の波形のループ部の開始位置)である。12は、調和外成分の波形(Timbre)要素の波形のループ部の終了位置である。上記データ3~データ7は各成分要素毎にコードブックに記憶されているベクトルデータを指し示すための識別情報のデータであり、上記データ2及びデータ8~データ12はベクトルデータから元の(分離前の)波形を組み立てるための時間情報のデータである。このように、奏法モジュールのデータはベクトルデータを指し示すためのデータと時間情報のデータとから構成される。このような奏法テーブルに記憶されている奏法モジュールのデータを使用することにより、コードブックに記憶されている波形の素材(ベクトルデータ)を使って、波形を自由に組み立てることができることになる。つまり、奏法モジュールは、奏法(若しくはアーティキュレーション)に応じて生成する波形の挙動を表すデータである。なお、奏法モジュールのデータの種類や数は各奏法モジュール毎に異なっておりよい。また、上述したデータ以外にも他の情報等を持っていてよい。例えば、波形の時間軸を伸長/圧縮制御するためのデータなどを持っていてもよい。

【0025】

また、上述の例では説明を理解しやすくするために、1つの奏法モジュールが調和成分の各要素(波形、ピッチ、振幅)及び調和外成分の各要素(波形、振幅)の全てを具備している例について説明したが、これに限らず、奏法モジュールが調和成分の各要素(波形、ピッチ、振幅)や調和外成分の各要素(波形、振幅)の1つからなっておりよいのは勿論である。例えば、奏法モジュールが調和成分の波形(Timbre)要素、調和成分のピッチ(Pitch)要素、調和成分の振幅(Amplitude)要素、調和外成分の波形(Timbre)要素、調和外成分の振幅(Amplitude)要素のいずれか1つの要素からなっておりよい。こうすると、各成分毎に奏法モジュールを自由に組み合わせ使用することができることになり好ましい。

【0026】

このように、様々な自然楽器の様々な演奏態様による演奏音の波形データを全波形データで持つのではなく、波形形状の変化に必要な一部の波形(例えば、アタック部波形、ボデ

ィ部波形、リリース部波形、ジョイント部波形等)のみを抽出し、さらに成分、要素、代表点といった階層的な圧縮手法を用いて、データ圧縮された形で波形データをハードディスク109に記憶することから、波形データを記憶するために必要なハードディスク109の記憶容量を削減することができるようになっている。

【0027】

図1に示す波形生成装置において、波形の合成はコンピュータがこの発明に係る波形合成処理を実現する所定のプログラム(ソフトウェア)を実行することにより実施される。図4Aは、前記波形合成処理を実現する所定のプログラム(「データベースに基づく楽音合成処理」)のフローチャートの一実施例を示したものである。また、この種のプログラムの形態に限らず、波形合成処理を専用ハードウェア装置の形態で実施するようにしてもよい。図4Bは、図4Aと同様の波形合成処理を専用ハードウェア装置の形態で構成した場合の一実施例を示すブロック図である。主に、図4Bに従って説明し、図4Aについては対応するステップを括弧書きして示す。

曲データ再生部101Aは、奏法記号付き曲データの再生処理を行う(ステップS11)。最初に、曲データ再生部101Aは奏法記号付き曲データ(演奏情報)を受信する。通常の楽譜には、そのままではMIDIデータとならないような強弱記号(クレッシェンドやデクレッシェンド等)、テンポ記号(アレグロやリタルダンド等)、スラー記号、テヌート記号、アクセント記号等の音楽記号が付されている。そこで、これらの記号を「奏法記号」としてデータ化して、この「奏法記号」を含むMIDI曲データが「奏法記号付き曲データ」である。「奏法記号」は、チャートIDとチャートパラメータとから構成する。チャートIDは楽譜に記載される音楽記号を示すIDであり、チャートパラメータはチャートIDで示される音楽記号の内容の程度を示すパラメータである。例えば、チャートIDが「ピブラート」を示す場合にはピブラートの速さや深さ等がチャートパラメータとして付与され、チャートIDが「クレッシェンド」を示す場合にはクレッシェンドのスタート時の音量、クレッシェンドのエンド時の音量、音量変化する時間長等がチャートパラメータとして付与される。

【0028】

楽譜解釈部(プレーヤー)101Bでは、楽譜解釈処理を行う(ステップS12)。具体的には、曲データに含まれるMIDIデータと上述した「奏法記号」(チャートIDとチャートパラメータ)を奏法指定情報(奏法IDと奏法パラメータ)に変換し、時刻情報とともに奏法合成部(アーティキュレーター)101Cに出力する。一般的に、同じ音楽記号でも演奏家により記号の解釈が異なって、演奏家毎に異なった演奏方法(すなわち、奏法若しくはアーティキュレーション)で演奏が行われることがある。あるいは、音符の並び方等によっても、演奏家毎に異なった演奏方法で演奏が行われることもある。そこで、そのような楽譜上の記号(音楽記号や音符の並び方等)を解釈する知識をエキスパートシステム化したものが楽譜解釈部101Bである。楽譜解釈部101Bにおける楽譜上の記号を解釈する際の基準の一例としては、以下のようなものがある。例えば、ピブラートは8分音符以上でないとかげられない。スタッカートでは自然にダイナミクスが大きくなる。テヌート度で音符の減衰率が決まる。レガートは1音中で減衰しない。8分音符ピブラートのスピードは音価でほぼ決まる。音高によってダイナミクスは異なる。更には、1フレーズ内の音高の上昇又は下降によるダイナミクスの変化、減衰ダイナミクスはdbリニア、テヌートやスタッカート等に応じた音符の長さの変化、アタック部のベンドアップの記号に応じたベンドアップの幅とカーブ、といったような各種の解釈基準がある。楽譜解釈部101Bはこのような基準に従って解釈を楽譜に対して行うことにより、楽譜を音に変換する。更に、楽譜解釈部101Bは、ユーザからのプレーヤー指定、すなわちユーザにより誰の演奏か(奏法か)の指定に応じて上述の楽譜解釈処理を行う。楽譜解釈部101Bは、このプレーヤー指定に応じて楽譜の解釈方法を異ならせて楽譜を解釈する。例えば、この複数プレーヤーに対応した異なる楽譜解釈方法はデータベースに蓄積されており、楽譜解釈部101Bはユーザからのプレーヤー指定に応じて選択的に楽譜解釈方法を異ならせて楽譜の解釈を行う。

10

20

30

40

50

【0029】

なお、楽譜の解釈結果を示すデータを予め含むように曲データ（演奏情報）を構成してもよい。そのような予め楽譜を解釈した結果のデータを含む曲データを入力した場合には、上述した処理を行う必要がないことは言うまでもない。また、楽譜解釈部101B（ステップS12）における楽譜の解釈処理は全自動で行うようにしてもよいし、ユーザによる人為的入力操作を適宜介在させて行うようにしてもよい。

【0030】

奏法合成部（アーティキュレーター）101Cは楽譜解釈部（プレーヤー）101Bにより変換された奏法指定（奏法ID+奏法パラメータ）に基づいて奏法テーブルを参照して、奏法指定（奏法ID+奏法パラメータ）に応じたパケットストリーム（あるいはベクトルストリームとも呼ぶ）及び奏法パラメータに応じた該ストリームに関するベクトルパラメータを生成し、波形合成部101Dに供給する（ステップS13）。パケットストリームとして波形合成部101Dに供給されるデータは、ピッチ（Pitch）要素及び振幅（Amplitude）要素に関してはパケットの時刻情報、ベクトルID、代表点値列などであり、波形（Timbre）要素に関してはベクトルID、時刻情報などである（詳しくは後述する）。次に、波形合成部101Dはパケットストリームに応じてコードブックからベクトルデータを取り出し、該ベクトルデータをベクトルパラメータに応じて変形し、変形したベクトルデータに基づいて波形を合成する（ステップS14）。それから、他パートの波形生成処理を行う（ステップS15）。ここで、他パートとは、複数の演奏パートのうち奏法合成処理を行わない、通常の楽音波形合成処理が適用される演奏パートである。例えば、これらの他のパートは通常の波形メモリ音源方式で楽音生成を行う。この「他パートの波形生成処理」は、専用のハードウェア音源（外部の音源ユニットやコンピュータに装着可能な音源カード）に行わせてもよい。説明を簡略化するために、この実施例では奏法（若しくはアーティキュレーション）に応じた楽音生成を行うのは1パートのみの場合とする。勿論、複数パートで奏法再生してもよい。

【0031】

図5は、上述した奏法合成部101Cにおける奏法合成処理の流れを説明するためのブロック図である。ただし、図5では奏法モジュールとコードブックが別々に記憶されているように図示したが、実際には両方ともハードディスク109のデータベース内に記憶されている。

奏法合成部101Cは、楽譜解釈部101Bからの奏法指定（奏法ID+奏法パラメータ）と時刻情報のデータに基づいて、波形合成部101Dに供給する各種パケットストリームを作成する。奏法合成部101Cで各音色毎に使用している奏法モジュールは固定的ではなく、ユーザが新たに奏法モジュールを使用中の奏法モジュールに追加したり、使用している奏法モジュールの一部の奏法モジュールの使用を中止したりすることができる。また、奏法合成部101Cでは、選択された要素データと奏法パラメータの値との間のズレ分を補正するための補正情報を作成する処理や、前後の奏法モジュールの波形特性を滑らかに接続する接続部の平滑化などの処理も行う（詳しくは後述する）。

なお、標準的には楽譜解釈部101Bから奏法合成部101Cに対してデータが与えられるがそれに限らず、前述のとおり、楽譜解釈部101Bにより既に解釈の終わっている奏法指定付き曲データ乃至人間が楽譜の解釈をして奏法IDや奏法パラメータを付与した奏法指定付き曲データを用意して、それを再生したデータを奏法合成部101Cに供給するようにしてもよい。

【0032】

図6は、奏法合成処理の一実施例を詳細に示したフローチャートである。

奏法合成部101Cは、奏法ID及び奏法パラメータに応じて奏法テーブルから奏法モジュールの選択を行う（ステップS21）。すなわち、楽譜解釈部101Bから送信された奏法ID（楽器情報+モジュールパーツ名）と奏法パラメータに応じて1つの奏法モジュールを選択する。この際に、楽譜解釈部101Bは楽譜を解釈する前に楽器情報の示す音色に対応してどのようなモジュールパーツが奏法テーブルに存在するかを予めデータベ

10

20

30

40

50

スをチェックして確認し、存在しているパーツの範囲で奏法IDを指定する。なお、存在しないパーツが指定された場合には、その代わりに類似の特性を有する奏法IDが選択されるようにしてもよい。次に、該指定された奏法IDと奏法パラメータに応じて複数の要素データを選択する(ステップS22)。すなわち、指定された奏法IDと奏法パラメータとにより奏法テーブルを参照することにより、奏法モジュールを特定し、該モジュールから該奏法パラメータに対応した複数の要素データを選択する。この際に、奏法モジュール中に奏法パラメータに完全一致する要素データが存在しない場合には、その値に近い奏法パラメータに対応した要素データが選択される。

【0033】

次に、時刻情報に応じて要素データ中の各位置の時刻を算出する(ステップS23)。すなわち、各要素データを、時刻情報に基づいて絶対的な時間位置に配置する。具体的には、時刻情報に基づいて、各相対的な時間位置を示す要素データから対応する絶対時間を算出する。こうして、各要素データのタイミングを決定する(図3参照)。そして、奏法パラメータに応じて各要素データの値を補正する(ステップS24)。すなわち、選択された要素データと奏法パラメータの値との間のずれ分を補正する。例えば、楽譜解釈部101Bから送信されたAltoSax [NormalAttack]モジュールのAttack直後の音量(奏法パラメータ)が「95」であり、奏法テーブルに存在するAltoSax [NormalAttack]モジュールのAttack直後の音量が「100」である場合、奏法合成部101CはAttack直後の音量が「100」であるAltoSax [NormalAttack]モジュールの要素データを選択する。しかし、このままではAttack直後の音量が「100」のままであることから、選択された要素データの代表点に対して補正を行うことによってAttack直後の音量を「95」に補正する。このように、選択された要素データの値を送信された奏法パラメータの値に近づけるようにして補正を行う。また、設定されているマイクロチューニング(楽器の調律)の値に応じた補正や楽器の音量変化特性に応じた音量の補正等も行う。これらの補正は各要素データの代表点値を変化することにより行われ、代表点値を大きく変化することもある。すなわち、補正を行うのに必要十分なデータが代表点であり、この代表点をコントロールすることによって各種の補正を行う。

なお、上記ステップS23では、上記奏法パラメータのような補正情報によって、上記時刻情報が示す時間位置を補正するようにしてもよい。例えば、演奏データに基づいて得られる時間位置と上記時刻情報が示す時間位置とが一致しない場合に、演奏データに基づいて得られる時間位置に近い時間位置を示す時刻情報を選択して、そこで取得した時刻位置情報を演奏データに応じて補正することで、演奏データの意図する時刻位置情報を得ることができる。また、演奏データがタッチやベロシティのような可変制御ファクタを含む場合は、その可変制御ファクタに応じて時刻位置情報を補正することで、演奏データに応じた時刻位置情報の可変制御を行うことができる。補正情報は、このような時刻位置補正を行うための情報を含む。

【0034】

更に、各要素データを調整して隣り合う奏法モジュールの接続部を平滑化するためにリンク処理を行う(ステップS25)。すなわち、前後の奏法モジュールにおける接続部の代表点を互いに接近させて接続することによって、前後の奏法モジュールの波形特性が滑らかになるようにする。このような接続若しくはリンク処理は、調和成分の波形(Timbre)、振幅(Amplitude)、ピッチ(Pitch)等の各要素毎に、あるいは調和外成分の波形(Timbre)、振幅(Amplitude)の各要素毎に、別々に行われる。

この際、前の奏法モジュールの「リンク開始点」から、後の奏法モジュールの「リンク終了点」までの範囲で調整を行う。すなわち、「リンク開始点」から「リンク終了点」の範囲内にある代表点を「歩みより率」に基づいて調整する。この「歩みより率」は、前の奏法モジュールと後の奏法モジュールからそれぞれどれだけ歩み寄ったところで接続するかを制御するためのパラメータであり、後述するように前後の奏法モジュールの組み合わせに従って決定される。また、前後の奏法モジュールを接続した際に、波形の接続がうまく行かない場合には、前後いずれかの奏法モジュールでその波形特性のベクトルIDを間引

くことにより接続を滑らかにする。この間引きを実現するために、「奏法モジュール組み合わせテーブル」と、これから参照される「間引き実行パラメータ範囲テーブル」と、さらにこれから参照される「間引き時間テーブル」を用意する。

この他にも、以下のような楽譜解釈部 101Bにおけるリンク処理により波形特性を滑らかに接続することができる。例えば、奏法モジュールとは関係なく、奏法パラメータ（ダイナミクス値、ピッチパラメータ値等）の不連続部分を滑らかに接続する。あるいは、ビブラートからリリースへと移行する場合にビブラートを早めに減少させることにより、滑らかに接続する。

【0035】

ここで、上述のリンク処理について詳しく説明する。すなわち、前後の奏法モジュールの接続部を平滑化する（ステップS25参照）ための各要素データの調整について簡単に説明する。まず、図7を用いて奏法モジュールが振幅（Amplitude）要素又はピッチ（Pitch）要素と対応する場合のリンク処理について説明する。

前の奏法モジュールと後の奏法モジュールとの接続部における代表点の値の不連続により両者間の接続部に段差が生じている場合、まずダイナミクス接続点（Amplitudeの場合）あるいはピッチ接続点（Pitchの場合）の目標値を、前後どちらの奏法モジュール側の値により近づけるかという指標の「歩みより率」を決定する。本実施例では「歩みより率」が図示のようなテーブルによって与えられるとする。例えば前の奏法モジュールのベクトルIDが「3」であり、後の奏法モジュールのベクトルIDが「7」である場合の「歩みより率」はテーブルから「30」と決定される。こうして決定された「歩みより率」により前の奏法モジュールの「リンク開始点」から「奏法モジュールの終了点」まで、徐々に目標値に向けてエンベロープ形状を変形する。また、後の奏法モジュールの「リンク終了点」から「奏法モジュール開始点」まで、徐々に目標値に向けてエンベロープ形状を変形する。例えば「歩みより率」が「30」と決定された場合、前の奏法モジュールに対する目標値は「30」であり、前の奏法モジュールは後の奏法モジュール側に「30」%歩みよりを行う（本実施例では、前の奏法モジュールにおける最後の代表点が下方に「30」%歩みよりする）。一方、後の奏法モジュールは前の奏法モジュール側に「70」（100 - 30）%歩みよりを行う（本実施例では、後の奏法モジュールにおける最初の代表点が上方に「70」%歩みよりする）。また、リンク開始点からリンク終了点までに存在する前後の奏法モジュールの複数代表点が上記歩みよりに伴って各々上下に歩みよりを行う。このように、歩みよりは前後する奏法モジュールの複数の代表点で行われる。なお、リンク開始点とリンク終了点は適宜定めてよいが、リンク開始点やリンク終了点を所望の代表点と同一の点に設定すると、図に示したようなリンク開始点やリンク終了点におけるエンベロープ形状の折れ曲がりがなくなるので望ましい。勿論、リンク開始点やリンク終了点を所望の代表点と同一の点に設定していない場合でも、エンベロープ形状に折れ曲がりが生じないように歩みよりを行うようにしてよいことは言うまでもない。

【0036】

なお、「歩みより率」の決定は上述した例に限られるものではない。例えば、接続点の前後で指定された奏法パラメータに基づいて決定してもよい。または、奏法IDや奏法パラメータになる前の演奏データに基づいて決定してもよい。あるいは、それらのデータの組み合わせに基づいて決定してもよい。また、上述の例では「歩みより率」により歩みよりする代表点は1つであり、その他の代表点はその歩みよりに伴って適量だけ歩みよりするようにしたが、複数の代表点各々について別々に「歩みより率」を決定し、それに従って複数の代表点を各々「歩みより率」分だけ歩みよりするようにしてもよい。

【0037】

次に、奏法モジュールが波形（Timbre）要素である場合のリンク処理について説明する。図8A～図8Dは、奏法モジュールが波形（Timbre）要素である場合のリンク処理を説明するための概念図である。図8Aはアタック部波形とボディ部波形とを接続した場合の波形の間引きを説明するための概念図であり、図8Bはボディ部波形とリリース部波形とを接続した場合における波形の間引きを説明するための概念図である。図8Aでは、ボディ

10

20

30

40

50

部波形は5つのループ波形L1～L5からなり、各々所定の時間範囲でループ再生されるものとする。同様に、図8Bのボディ部波形は6つのループ波形L1'～L6'からなるものとする。

波形に関する要素データの調整（つまり、波形のリンク処理）の方法には種々あるが、その一例として、例えばアタック部あるいはジョイント部の奏法モジュールとボディ部の奏法モジュールとの接続（あるいは、ボディ部の奏法モジュールとリリース部あるいはジョイント部の奏法モジュールとの接続）において、波形の部分的間引きにより滑らかに接続する方法を提案する。波形と波形とを接続する際に、クロスフェード合成することはよく知られている。しかし、図8Aの例の場合のように、接続時点から最初のループ波形L1の開始位置までの時間tが短い場合、短い時間t内で急なクロスフェード合成をしなければならなくなる。そのような急なクロスフェード波形合成、つまり接続する波形と波形との間の時間が非常に接近している場合に当該波形間でクロスフェード波形合成を行うと、それに伴って大きなノイズを発生する波形を生ずることになり、好ましくない。そこで、波形の一部を間引き（削除）して接続する波形と波形との時間間隔を広げることにより、急なクロスフェード波形合成を行わないようにする。この場合に、アタック部やリリース部あるいはジョイント部における波形は1つの塊であって、波形を間引くことができないので、この場合はボディ部側のループ波形の間引きを行う。図8A及び図8Bでは、黒く塗りつぶした長方形で示したループ波形L1、L6'を間引きする。例えば、図8Aでは接続時点からの時間差が比較的長い2番目のループ波形L2とアタック部波形の末尾波形とをクロスフェード合成し、最初のループ波形L1は使用しない。同様に、図8Bではループ波形L5'とリリース部波形との間でクロスフェード合成を行い、波形L6'は使用しない。

なお、ジョイント部とは音と音の間（又は音部分と音部分の間）を任意の奏法でつなぐ波形区間のことである。

【0038】

また、アタック部の奏法モジュールとリリース部あるいはジョイント部の奏法モジュールとの接続を波形間引きにより滑らかにする。図8C及び図8Dは、アタック部波形とリリース部波形とを接続する場合における波形の間引きを説明するための概念図である。

この場合には、アタック部あるいはリリース部等の奏法モジュールが波形間引きできる場合とできない場合とがある。アタック部の奏法モジュールが波形間引きできる例としてはバンドアタック部（後半にいくつかのループ波形を持つ）がある。また、前半にいくつかのループ波形を持つリリース部の場合も波形間引きが行える。このように、波形間引きできる側の奏法モジュールを波形間引きする。例えば、バンドアタック部とリリース部とを接続する場合には、図8Cに示すようにバンドアタック部側のループ波形を間引きする（図8Cでは、バンドアタック部側の黒く塗りつぶした長方形で示したループ波形を1つ間引きする）。また、ノーマルアタック部とループ波形を有するリリース部とを接続する場合には図8Dに示すようにリリース部側のループ波形を間引きする（図8Dでは、リリース部側の黒く塗りつぶした長方形で示したループ波形を1つ間引きする）。

なお、間引く対象とするループ波形は奏法モジュールと奏法モジュールとの接続部に最も近いループ波形（先頭あるいは最後に位置するループ波形）とすることに限らず、複数ループ波形から所定の優先順位に従って間引く対象とするループ波形を特定するようにしてもよい。

【0039】

このように、ある奏法モジュールの組み合わせにおいて、ある奏法パラメータの範囲で接続がうまく行かない場合に波形を間引くが、これを実現するために、例えば「奏法モジュール組み合わせテーブル」と、これから参照される「間引き実行パラメータ範囲テーブル」と、更にこれから参照される「間引き時間テーブル」を用意する。「奏法モジュール組み合わせテーブル」は、接続する前後の奏法モジュールの組み合わせにより所定のパラメータを決定するためのテーブルである。「間引き実行パラメータ範囲テーブル」は、上記パラメータ毎に間引きを行う時間の範囲を決定するためのテーブルである。「間引き時間

10

20

30

40

50

テーブル」は、間引き時間を決定するためのテーブルである。接続時点と最初の（又は最後の）ループ波形 L 1（又は L 6'）との時間差（図 8 A ~ 図 8 D に示す時間 t）が基準の間引き時間より短い場合に、該ループ波形を間引く。

【 0 0 4 0 】

さらに、奏法モジュールのサンプル長が短く、当該奏法モジュールの後に続く奏法モジュールが開始されるよりも前に終了してしまう場合の波形接続について図 9 を用いて説明する。ただし、図 9 では、図の左側から右側に時系列に A.Sax[BendUpAttack]、A.Sax[NormalShortBody]、A.Sax[VibratoBody]、A.Sax[NormalRelease] の 4 つの奏法モジュールで波形（Timbre）要素のパケットストリームを形成している場合について説明する。各奏法モジュールのサンプル長（区間長）は、“length” で示す長さで表せられる。図 9 において、最上段に記載されている「ノートオン」と「ノートオフ」は、MIDI データのイベントタイミングである。また、中段に記載されている A.Sax[BendUpAttack] 等はそれぞれ奏法 ID の発生タイミングであり、note、dynamics、depth 等はそれぞれ奏法パラメータの発生タイミングを示す。

A.Sax [BendUpAttack] モジュールは、時刻 t 0 から開始される。また、時刻 t 1 は当該モジュール内のノートオンのタイミングであり、指示されたノートオンタイミングにあわせる。また、当該モジュールのパケットストリームの内容は、上記 note、dynamics、depth 等の奏法パラメータに基づいて制御される。A.Sax[NormalShortBody] モジュールは、アタックモジュール直後の時刻 t 2 から開始される。時刻 t 3 は、接続部において、その途中からビブラート奏法がスタートしているタイミングである。このタイミングは、例えば、曲データに付与されたビブラート記号の開始タイミングに基づいて決定される。時刻 t 5 は、A.Sax[NormalRelease] モジュール内のノートオフタイミングであり、指示されたノートオフタイミングにあわせる。A.Sax[NormalRelease] モジュールの始まりの時刻 t 4 はそれに応じて特定される。すなわち、時刻 t 1 においてノートオンされ、時刻 t 5 においてノートオフされることから、実際に当該パケットストリームから生成される波形に従って発音される時間は時刻 t 1 から時刻 t 5 までの時間である。このようなパケットストリームの場合に、時刻 t 2 から時刻 t 4 までの時間長と、その間の A.Sax[NormalRelease] モジュールと A.Sax[VibratoBody] モジュールの各サンプル長 length の合計が合わないことが多く、適切に対処する必要が生ずる。このような場合、同じ奏法モジュールを繰り返すことによってサンプル長 length の合計を前記時間長にあわせるか、奏法モジュールのサンプル長を可変して前記時間長に合わせるか、あるいは前記両方を組み合わせて用いて前記時間長を合わせる。このようにして、各モジュール間で調節して波形接続を行うようになっている。上述の例では、A.Sax[NormalShortBody] モジュールを繰り返すことにより、その後続く A.Sax[VibratoBody] モジュールと波形接続を行っている。同様に、A.Sax[VibratoBody] モジュールを繰り返すことにより、その後続く A.Sax[NormalRelease] モジュールとの波形接続を行っている。

【 0 0 4 1 】

上記のように、奏法モジュールを複数回繰り返して波形接続を行う場合、複数繰り返す側の奏法モジュールの時間長を可変する。この時間長の可変制御は、上述した例では A.Sax[NormalShortBody] モジュールあるいは A.Sax[VibratoBody] モジュールにおける代表点を移動することによってなされる。すなわち、A.Sax[NormalShortBody] モジュールあるいは A.Sax[VibratoBody] モジュールを構成する複数のループ波形間のクロスフェード接続の時間を変化するなどの適宜の方法によって実現する。ループ波形の場合は、ループ回数乃至ループ継続時間を可変することによって、比較的簡単に、ループ再生波形全体の時間長を可変制御することができる。一方、ノンループ波形の場合は、時間軸上におけるその存在長を可変制御することはそれほど簡単ではない。従って、上記のように、ノンループ波形とループ波形とからなる一連の音の波形において、ループ読出区間の波形データの時間軸を伸縮可変制御することで全体の発音時間長を可変制御するように工夫することは、時間伸縮制御を容易にするので極めて好ましい。そのために、特開平 10 307586 号で本出願人が先に提案した「Time Stretch & Compress」制御（略して「TSC 制御」）

10

20

30

40

50

を用いるとよい。特に、特殊な奏法に対応するノンループ波形において時間軸の長さを可変するために、上記「TSC制御」は好ましく応用できる。

【0042】

このようにして作成されるパケットストリームの一列を概念的に示すと、図10のようである。図10では、上から順に調和成分の振幅(Amplitude)要素、ピッチ(Pitch)要素、波形(Timbre)要素、調和外成分の振幅(Amplitude)要素、波形(Timbre)要素における各パケットストリームを示す。また、調和成分の振幅(Amplitude)要素、ピッチ(Pitch)要素、調和外成分の振幅(Amplitude)要素において、黒く塗りつぶした正方形は代表点であり、これらを結ぶ曲線はパケットストリームの中のパケットに含まれるベクトルIDで示されたベクトルの形状を示す。調和成分の波形(Timbre)要素、調和外成分の波形(Timbre)要素において、白抜きの長方形Lはループ波形を示すものであり、その他の長方形NLはノンループ波形を示すものである。なお、ノンループ波形のうち、斜線を引いたものは特に特徴のあるノンループ波形を示すものである。さらに、この実施例では、NormalAttackモジュールにおける調和成分と調和外成分の波形(Timbre)要素をそれぞれ2個のベクトルで構成し、調和成分の振幅(Amplitude)要素とピッチ(Pitch)要素及び調和外成分の振幅(Amplitude)要素をそれぞれ1個のベクトルで構成している。本実施例では、調和成分、調和外成分とも、波形(Timbre)要素がノンループ波形となっている部分については、振幅(Amplitude)要素及びピッチ(Pitch)要素がベクトルを持たないようになっている。しかしながら、波形(Timbre)要素がノンループ波形であるところでも、振幅(Amplitude)要素、ピッチ(Pitch)要素のベクトルを持ち生成波形を制御するようにしてもよい。VibratoBodyモジュールでは、調和成分の波形(Timbre)要素を5個のベクトルで構成し、調和成分の振幅(Amplitude)要素とピッチ(Pitch)要素及び調和外成分の波形(Timbre)要素と振幅(Amplitude)要素をそれぞれ1個のベクトルで構成している。ここで、VibratoBodyモジュールは3回繰り返されているが、各繰り返し毎に各ベクトルの形状が異なっていることに注意してほしい。これは各繰り返し毎に異なる奏法パラメータが指定されたからである。異なる奏法パラメータに応じて異なる要素データが選択されたり、異なる奏法パラメータに応じて異なるレベル制御、時間軸制御が行われたりする。NormalJointモジュールでは、調和成分と調和外成分の波形(Timbre)要素をそれぞれ3個のベクトルで構成し、調和成分の振幅(Amplitude)要素とピッチ(Pitch)要素及び調和外成分の振幅(Amplitude)要素をそれぞれ2個のベクトルで構成している。なお、NormalBodyモジュールの説明は省略する。

奏法合成部101Cは、上述したようにパケットストリームを各成分(調和成分及び調和外成分)毎に生成する。これらのパケットストリームは複数個のパケットで構成されてなり、1個1個のパケットはベクトルIDとパケットの時刻情報とを具える。それに加えて、調和成分の振幅(Amplitude)要素、ピッチ(Pitch)要素、調和外成分の振幅(Amplitude)要素の場合には各代表点の確定値などを具える。勿論、これに限られるものではなく、ベクトルIDとパケットの時刻情報に加えて他の情報を具えてよい。このような1つ1つのパケットの内容に従って、各成分毎にパケットストリームが構成されている。このように、パケットストリームには複数のパケット及びその各パケットの時刻情報(開始時刻)が含まれる。

なお、楽器の種類等によってパケットストリームの数が異なっていてよいことは言うまでもない。

【0043】

波形合成部101Dは、奏法合成部101Cから供給される各成分毎のパケットストリーム(ベクトルID、時刻情報、補正情報等を含む複数のパケットの列)に基づいて波形を合成する。図11は、波形合成部101Dにおける動作を説明するために全体構成を示した概念図である。図12~図15は、波形合成部101Dにおける各動作を詳細に説明するためのブロック図である。図12は、波形合成の全体の流れを簡単に示すブロック図である。図13は、ベクトルローダを説明するためのブロック図である。図14は、ベクトルオペレータを説明するためのブロック図である。図15は、ベクトルデコーダを説明す

るためのブロック図である。

奏法合成部（アーティキュレーター）101Cで作成された各成分要素毎の packets ストリームは、波形合成部101Dにおける各成分要素毎に対応して設けられる所定の packets キューバッファ21～25に順次に packets 入力（つまり、packets 単位での入力）される。入力された packets は packets キューバッファ21～25に蓄積され、順次所定の順番でベクトルローダ20に送られる。ベクトルローダ20では packets 内のベクトルIDを参照して、当該ベクトルIDに対応するオリジナルのベクトルデータをコードブック26から読み出し（ロード）する。読み出されたベクトルデータは、各成分要素毎に対応して設けられる所定のベクトルデコーダ31～35へ送られ、ベクトルデコーダ31～35で各成分要素毎の波形が生成される。さらに、ベクトルデコーダ31～35では、生成された各成分要素毎の波形を各ベクトルデコーダ31～35間で同期しながら各成分毎（調和成分及び調和外成分）の波形を生成する。こうして生成された各成分毎の波形は、ミキサ38に送られる。奏法合成部（アーティキュレーター）101Cでは、packets キューバッファ21～25に対して packets を入力する他、ストリーム管理（個々のベクトルデータの生成や削除あるいはベクトルデータ間の接続に関する管理）や再生コントロール（所望の波形生成の実行あるいは生成された所望の波形の再生/停止などのコントロール）などの各種の制御を波形合成部101Dに対して実行する。

【0044】

上述したように、ベクトルローダ20には packets キューバッファ21に蓄積された packets ストリームを構成する packets を順次に入力され、ベクトルデコーダ20は各 packets 内のベクトルIDに基づいて、コードブック26から当該ベクトルIDに対応するベクトルデータを読み出し、読み出したベクトルデータをベクトルデコーダ31に送る（図12参照）。この際に、読み出した各 packets 内に補正情報（例えば、代表点に関する補正情報）が付されている場合がある。このような場合、ベクトルローダ20では読み出したオリジナルのベクトルデータを補正情報に従って変形し、変形したベクトルデータ（これをオリジナルのベクトルデータと区別するためにベクトル情報データと呼ぶ）を情報として持つ packets （これを奏法合成部101Cから入力された packets と区別するためにベクトル packets と呼ぶ）をベクトルデコーダ31～35へ出力する。このように、ベクトルローダ20は奏法合成部（アーティキュレーター）101Cから入力された packets のベクトルIDに基づきオリジナルのベクトルデータをコードブック26から読み出し、必要に応じて補正情報でベクトルデータの補正を行った上でベクトルデコーダ31～35にベクトル packets を渡す（図13参照）。上述したようなベクトルデータの代表点に関する補正情報には、例えば乱数に基づいて時刻情報をずらすための補正情報等いろいろな補正情報がありうる。

【0045】

図14に示すように、ベクトルデコーダ31～35は入力されたベクトル packets を処理するためのベクトルオペレータの生成や破棄、ベクトルオペレータ間の接続・同期管理、時刻管理、他ベクトルIDストリームからの入力のベクトルオペレータ毎のパラメータへの変換設定などの各種のオペレータ動作管理を行うものである。ベクトルオペレータ36及び37では、ベクトル情報データの読み出し、あるいはベクトル情報データの読み出し位置制御（Speed入力）やゲイン制御（Gain入力）などが行われる。このベクトルオペレータ36及び37に設定される各種のパラメータは、ベクトルデコーダ31～35で管理される。各成分要素毎に対応するようにベクトルデコーダ31～35が設けられており、当該ベクトルデコーダ31～35がベクトル packets 内のベクトル情報データを読み出して所望の波形の時系列的生成を行う。例えば、図15に示すように、ベクトルデコーダ31は調和成分の振幅（Amplitude）要素のエンベロープ波形を生成し、ベクトルデコーダ32は調和成分のピッチ（Pitch）要素のエンベロープ波形を生成し、ベクトルデコーダ33は調和成分の波形（Timbre）要素の波形を生成し、ベクトルデコーダ34は調和外成分の振幅（Amplitude）要素のエンベロープ波形を生成し、ベクトルデコーダ35は調和外成分の波形（Timbre）要素のエンベロープ波形を生成する。ベクトルデコーダ33は、

ベクトルデコーダ 3 1 及び 3 2 で生成された調和成分の振幅 (Amplitude) 要素のエンベロープ波形と調和成分のピッチ (Pitch) 要素のエンベロープ波形を付与した調和成分の波形を生成してミキサ 3 8 へ出力する。すなわち、ベクトルデコーダ 3 3 に対して、調和成分の振幅 (Amplitude) 要素のエンベロープ波形をゲイン制御 (Gain入力) を行うためのベクトルオペレータとして、調和成分のピッチ (Pitch) 要素のエンベロープ波形をベクトル情報データの読み出し位置制御 (Speed入力) を行うためのベクトルオペレータとして入力する。また、ベクトルデコーダ 3 5 では、ベクトルデコーダ 3 4 で生成された調和外成分の振幅 (Amplitude) 要素のエンベロープ波形を付与した調和外成分の波形を生成してミキサ 3 8 へ出力する。すなわち、ベクトルデコーダ 3 5 に対しては、調和外成分の振幅 (Amplitude) 要素のエンベロープ波形をゲイン制御 (Gain入力) を行うための制

10

【 0 0 4 6 】

さらに、各成分、要素における波形の時系列的生成の際には、各ベクトルデコーダ 3 1 ~ 3 5 間で波形の同期を行いながら波形生成が行われる。例えば、波形 (Timbre) 要素のベクトルパケットと振幅 (Amplitude) 要素のベクトルパケットを入力した場合、波形 (Timbre) 要素のベクトルパケットに基づく波形合生成の時間を基準として、それに同期して振幅 (Amplitude) 要素のベクトルパケットに基づく振幅波形を生成する。該振幅波形により、波形 (Timbre) 要素のベクトルパケットに基づいて生成される波形の振幅が制御される。波形 (Timbre) 要素のベクトルパケットとピッチ (Pitch) 要素のベクトルパケットを入力した場合、波形 (Timbre) 要素のベクトルパケットに基づく波形生成の時間を基

20

準として、それに同期してピッチ (Pitch) 要素のベクトルパケットに基づくピッチ波形を合成する。該ピッチ波形により、波形 (Timbre) 要素のベクトルパケットに基づいて生成される波形のピッチが制御される。調和成分の波形 (Timbre) 要素のベクトルパケットと調和外成分の波形 (Timbre) 要素のベクトルパケットを入力する場合、調和成分の波形 (Timbre) 要素のベクトルパケットに基づく調和外成分合成の時間を基準として、それに同期して調和外成分の波形 (Timbre) 要素のベクトルパケットに基づく調和外成分が合成される。合成された調和成分と調和外成分の波形を混合することにより、所望の楽音波形を生成する。

30

【 0 0 4 7 】

なお、この実施例において、調和成分と調和外成分の同期または非同期を選択できるように構成し、同期が選択された場合にのみにおいて、上述の調和成分の波形 (Timbre) 要素のベクトルパケットに基づいて生成される調和成分の波形合成の時間を基準として、それに同期して調和外成分の波形 (Timbre) 要素のベクトルパケットに基づいて生成される調和外成分の波形を合成するようにしてもよい。

上述したように、パケットストリームは複数パケット列で構成されており、ベクトルパケットのパケットストリームの場合、各パケットには各々ベクトルデータが含まれている。すなわち、パケットストリームとはベクトルデータが時間方向に並んだものであり、振幅 (Amplitude) 要素のベクトルデータもピッチ (Pitch) 要素のベクトルデータも波形 (Timbre) 要素のベクトルデータもデータ構造や意味は違うが、ベクトルオペレータ 3 6 及び 3 7 から見た場合は基本的に同じものである。

40

図 1 6 は、ベクトルデータのデータ構造の一実施例を概念的に示す概念図である。例えば、ベクトルデータの読み出し位置の単位は [SEC] で、読み出し速度を等速とした場合、ベクトルデータ上での 1 サンプルは出力波形の 1 サンプルと一致する。また、読み出し速度の単位は $1 / 1200$ [cent] (= 2 の n 乗) で、指数 n が 0 だと等速、1.0 だと 2 倍 (例えば、波形 (Timbre) 要素の場合には 1 オクターブ上がる)、-1.0 だと 0.5 倍 (例えば、波形 (Timbre) 要素の場合には 1 オクターブ下がる) となる (図 1 6 の上段図参照)。また、コードブック 2 6 には実際のベクトルデータが格納される。例えば、振幅 (Amplitude) 要素のベクトルデータあるいはピッチ (Pitch) 要素のベクトルデータは VECTORPOINT 構造体の配列と、代表点データとからなっている。VECTORPOINT 構造体の配列には各点のサンプル位置と値が順番に入っており、例えば振幅 (Amplitude) 要素のベクトルデータ

50

の値は[db]単位であり、ピッチ(Pitch)要素のベクトルデータの値はMIDIのノートナンバー0を0.0としたときの1/1200[cent]単位である。また、代表点データはDWORD型の配列で、代表点となるVECTORPOINT構造体の配列のインデックス番号が格納されている(図16の下段図参照)。勿論、上述した例に限られるものでないことは言うまでもない。

【0048】

なお、上述したような波形生成装置を電子楽器に用いた場合、電子楽器は鍵盤楽器の形態に限らず、弦楽器や管楽器、あるいは打楽器等のようなタイプの形態でもよい。また、その場合に、曲データ再生部101A、楽譜解釈部101B、奏法合成部101C、波形合成部101D等を1つの電子楽器本体内に内蔵したものに限らず、それぞれが別々に構成され、MIDIインタフェースや各種ネットワーク等の通信手段を用いて各構成部を接続するように構成されたものにも同様に適用できることはいうまでもない。また、パソコンとアプリケーションソフトウェアという構成であってもよく、この場合処理プログラムを磁気ディスク、光ディスクあるいは半導体メモリ等の記憶メディアから供給したり、ネットワークを介して供給するものであってもよい。さらに、自動演奏ピアノのような自動演奏装置などにも適用してよい。

10

【0049】

【発明の効果】

以上の通り、本発明によれば、波形形状の変化を代表する一部の波形データを抽出して記憶手段に記憶することにより、波形形状の変化表現に必要な波形データを圧縮して記憶することができ、波形データ記憶量を節約することができると共に、抽出した一部の波形データをその時刻位置の情報と共に記憶手段に記憶することで、波形再生の際に、精巧な波形再現性を確保することができる、という優れた効果を奏する。また、波形形状の変化を代表する一部の波形データは、それと組になって与えられる時刻位置の情報に従って適切な時刻位置に配置されるので、楽音波形の大事な部分、つまり波形形状の変化を代表する波形に関して、適切な時刻位置で再現されることとなり、精巧な波形再現性を確保することができる、という優れた効果を奏する。また、サンプル識別情報と時刻位置情報の組を使用して、サンプルデータを時刻位置情報に従って適切な時刻位置に配置するようにしたので、波形データの記憶量節約と精巧な波形再現性の確保を実現することができる、という優れた効果を奏する。また、演奏データに基づき上記サンプル識別情報と時刻位置情報の組を取得し、取得した時刻位置情報を演奏データに応じて補正することで、演奏データの意図する時刻位置情報を得ることができると共に、演奏データに応じた時刻位置情報の可変制御を行うことができるので、記憶量の節約と、制御性に富む波形生成を実現できる、という優れた効果を奏する。こうして、奏法若しくはアーティキュレーションを考慮した高品質な波形の生成を、簡素化された構成でかつ制御性豊かに実現することができる。

20

30

【図面の簡単な説明】

【図1】 この発明に係る波形生成装置のハードウェア構成例を示すブロック図。

【図2】 波形生成装置において実行される「波形データベース作成処理」の一実施例を示すフローチャート。

【図3】 奏法モジュールに対応する実波形区間を構成する各成分及び要素の一例を模式的に示す図。

40

【図4A】 「データベースに基づく楽音合成処理」の一実施例を示すフローチャート。

【図4B】 図4Aと同様の波形合成処理を専用ハードウェアの形態で構成した場合の一実施例を示すブロック図。

【図5】 上述した奏法合成部における奏法合成処理の流れを説明するためのブロック図。

【図6】 奏法合成部で行われる奏法合成処理の一実施例を詳細に示したフローチャート。

【図7】 奏法モジュールが振幅要素又はピッチ要素に対応する場合におけるリンク処理を説明するための概念図。

【図8A】 アタック波形とボディ波形とを接続した場合の波形の間引きを説明するため

50

の概念図。

【図 8 B】 ボディ波形とリリース波形とを接続した場合の波形の間引きを説明するための概念図。

【図 8 C】 ベンドアタック波形とリリース波形とを接続した場合の波形の間引きを説明するための概念図。

【図 8 D】 ノーマルアタック波形とループ部を有するリリース波形とを接続した場合の波形の間引きを説明するための概念図。

【図 9】 後に続く奏法モジュールが開始されるよりも前の奏法モジュールが終了してしまう場合の波形接続について説明するための概念図。

【図 10】 奏法合成部で生成されるパケットストリームを説明するための概念図。

【図 11】 波形合成部における動作を説明するために全体構成の一実施例を示した概念図。

【図 12】 波形合成の全体の流れを簡単に示すブロック図。

【図 13】 ベクトルローダを説明するためのブロック図。

【図 14】 ベクトルオペレータを説明するためのブロック図。

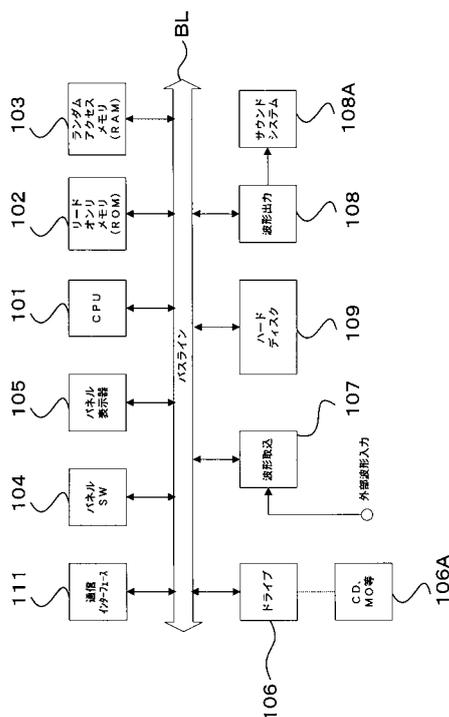
【図 15】 ベクトルレコーダを説明するためのブロック図。

【図 16】 ベクトルデータのデータ構造の一実施例を概念的に示す概念図。

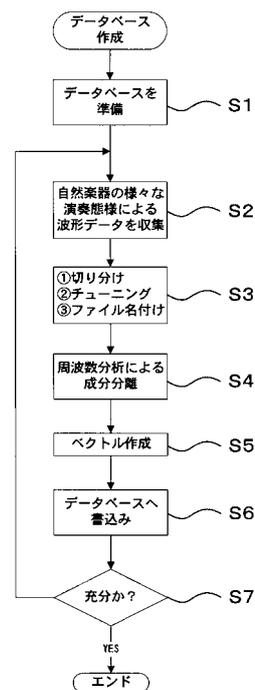
【符号の説明】

101... CPU、102... リードオンリメモリ (ROM)、103... ランダムアクセスメモリ (RAM)、104... パネルスイッチ、105... パネル表示器、106... ドライブ、
 106A... 外部記憶メディア、107... 波形取込部、108... 波形出力部、108A... サウンドシステム、
 109... ハードディスク、111... 通信インタフェース、BL... バスライン、101A... 曲データ再生部、
 101B... 楽譜解釈部、101C... 奏法合成部、101D... 波形合成部、20... ベクトルローダ、
 31 (32 ~ 35)... ベクトルデコーダ、36 (37)... ベクトルオペレータ、38... ミキサ

【図 1】



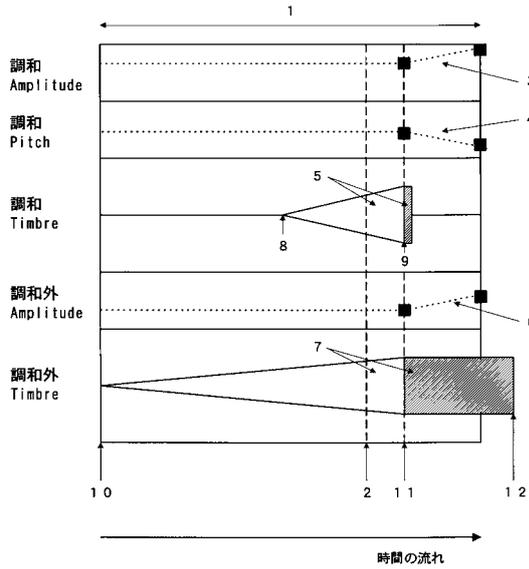
【図 2】



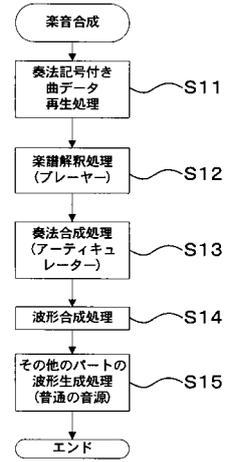
10

20

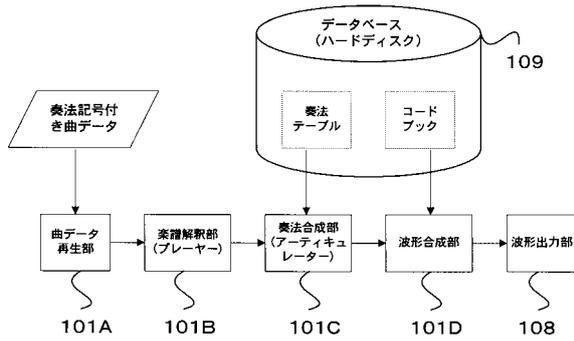
【 図 3 】



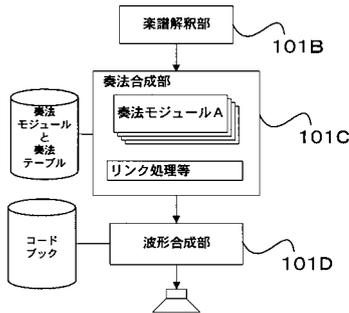
【 図 4 A 】



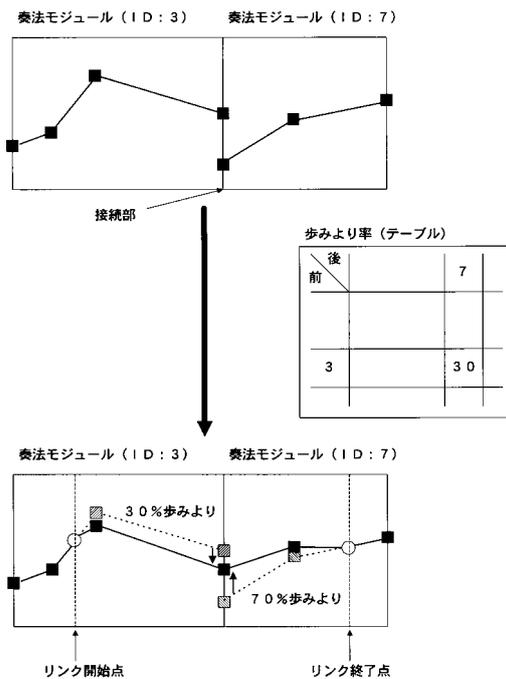
【 図 4 B 】



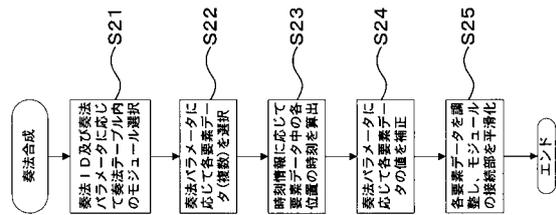
【 図 5 】



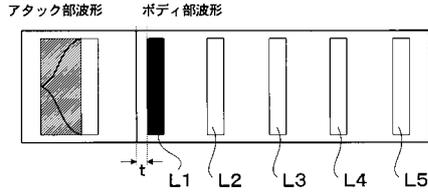
【 図 7 】



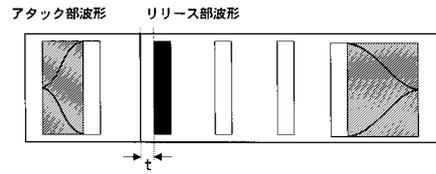
【 図 6 】



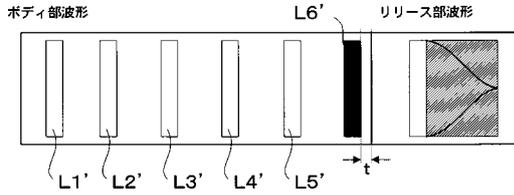
【 図 8 A 】



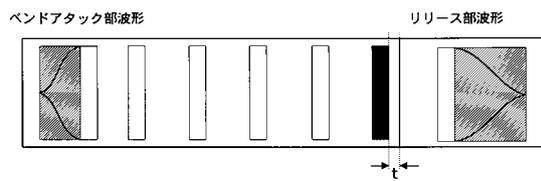
【 図 8 D 】



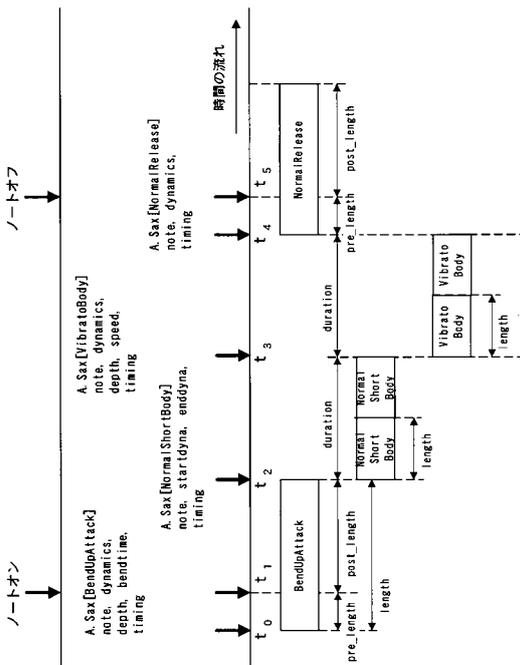
【 図 8 B 】



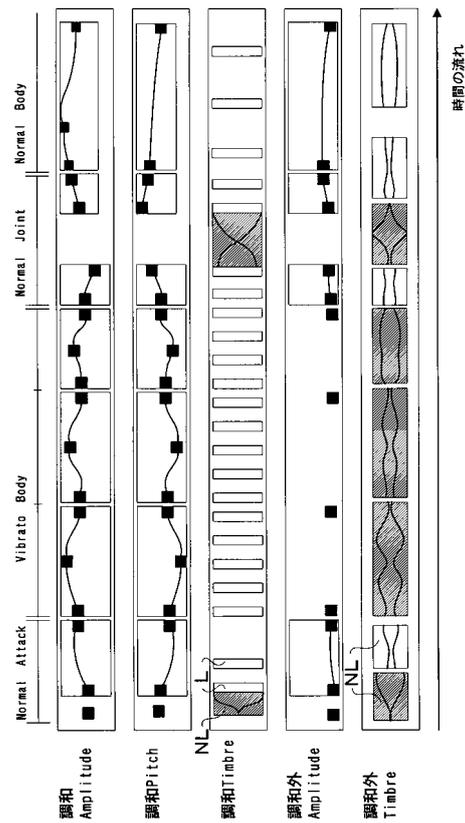
【 図 8 C 】



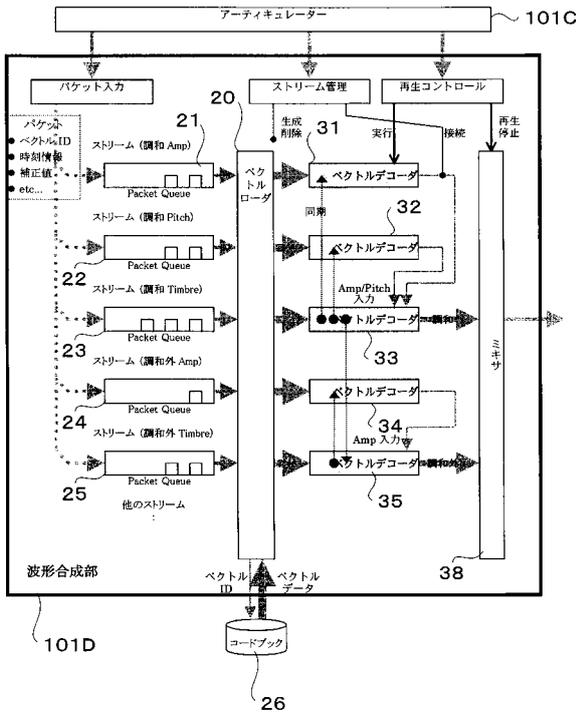
【 図 9 】



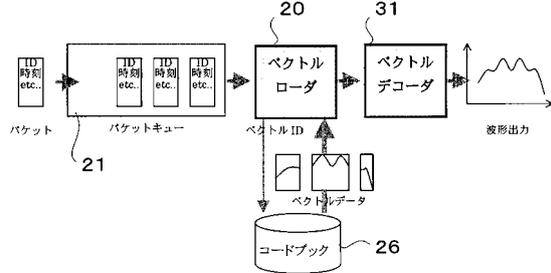
【 図 10 】



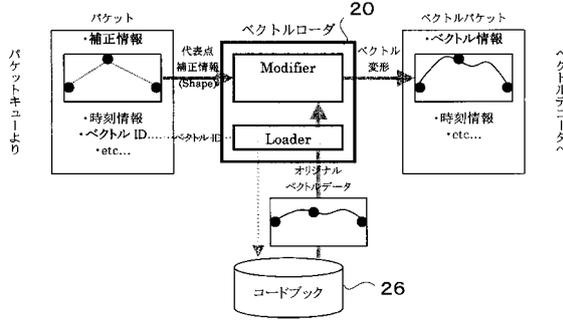
【図11】



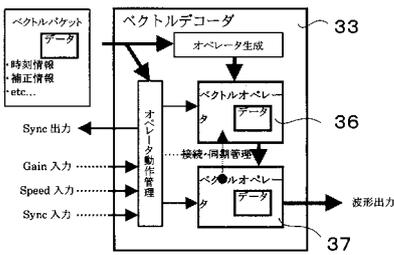
【図12】



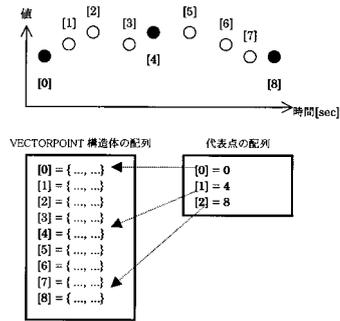
【図13】



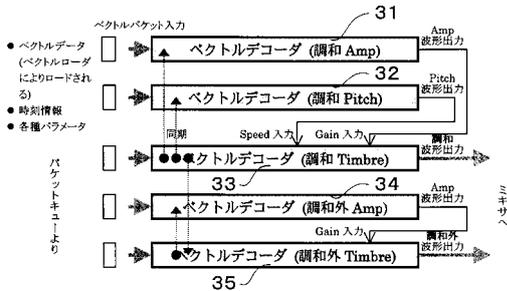
【図14】



【図16】



【図15】



フロントページの続き

審査官 小宮 慎司

- (56)参考文献 特開昭59-096513(JP,A)
特開平06-110455(JP,A)
特開平10-214083(JP,A)
特開平11-167382(JP,A)
特開平04-024699(JP,A)
特開昭53-084708(JP,A)

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