(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2007-264035
(P2007-264035A)
(43)公開日　平成19年10月11日(2007.10.11)

(51)Int.Cl.			F I			テーマコード（参考）
G10F	1/02	(2006.01)	G10F	1/02	B	5D378
G10H	1/00	(2006.01)	G10H	1/00	101B	

審査請求 未請求　請求項の数 3　OL　（全 25 頁）

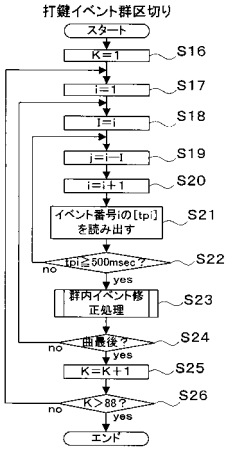
(21)出願番号　特願2006-85258(P2006-85258)
(22)出願日　平成18年3月27日(2006.3.27)

(71)出願人　000004075
ヤマハ株式会社
静岡県浜松市中区中沢町１０番１号
(74)代理人　100077539
弁理士　飯塚　義仁
(72)発明者　藤原　祐二
静岡県浜松市中沢町１０番１号　ヤマハ株式会社内

Ｆターム(参考)　5D378 MM82　MM90

(54)【発明の名称】鍵盤楽器

(57)【要約】
【課題】自動演奏ピアノにおいて、例えば連打打鍵演奏の部分など、自機で再生が不得手な部分を良好に再生できるようにする。
【解決手段】自動演奏ピアノは、１曲分の演奏情報から或る鍵１について全ての打鍵イベントを抽出する。該或る鍵１について抽出された複数の打鍵イベントのうち、連続する複数の打鍵イベントを一群の打鍵イベント群に含めるかどうかを、直前の打鍵イベントｊから当該打鍵イベントｉまでの時間間隔に基づき判断し（ステップＳ１６～Ｓ２２）、所定時間以上であれば、打鍵イベントｉは他の打鍵イベント群に属するものと判断し、該直前イベントｊを含む打鍵イベント群を区切る。ステップＳ２２は、打鍵イベント群内において、各打鍵イベントの少なくとも１つの楽音要素を揃えるよう修正を行なう。これにより、該打鍵イベント群内の楽音要素の不揃いを揃え、再生不得手な演奏も良好に再生できる。
【選択図】図７



【特許請求の範囲】
【請求項１】
鍵と、
前記鍵を駆動するための駆動手段と、
打鍵イベントのシーケンスからなる演奏情報を供給するための供給手段と、
前記演奏情報を構成する打鍵イベントの中から、同一鍵に対応する打鍵イベントを１乃至複数抽出する抽出手段と、
前記抽出された１乃至複数の打鍵イベントのうち、連続して発生する複数の打鍵イベントを一群の打鍵イベント群に含めるかどうかを所定の閾値に基づき判断する判断手段と、
前記判断手段により一群の打鍵イベント群に含めるよう判断された複数の打鍵イベントからなる打鍵イベント群内において、各打鍵イベントの少なくとも１つの楽音要素を揃えるよう修正を行なう修正手段と、
前記修正手段により修正された打鍵イベントに基づき、該打鍵イベントに対応する鍵を駆動するために前記駆動手段を制御する制御手段と
を具えることを特徴とする鍵盤楽器。
【請求項２】
鍵と、
前記鍵を駆動するための駆動手段と、
打鍵イベントのシーケンスからなる演奏情報を供給するための供給手段と、
前記演奏情報を構成する打鍵イベントの中から、同一鍵に対応する打鍵イベントを１乃至複数抽出する抽出手段と、
前記抽出された１乃至複数の打鍵イベントのそれぞれに基づき、駆動されるべき鍵の打鍵運動を規定するために、少なくとも該鍵の動きの位置成分の時間的変遷を表す軌道データを１乃至複数作成する軌道データ作成手段と、
前記作成された１乃至複数の軌道データのうち、連続して発生する複数の軌道データを一群の軌道データ群に含めるかどうかを所定の閾値に基づき判断する判断手段と、
前記判断手段により一群の軌道データ群に含めるよう判断された複数の軌道データからなる軌道データ群内において、各軌道データの少なくとも１つの楽音要素を揃えるよう修正を行なう修正手段と、
前記修正手段により修正された軌道データに基づき、該軌道データに対応する鍵を駆動するために前記駆動手段を制御する制御手段と
を具えることを特徴とする鍵盤楽器。
【請求項３】
前記抽出手段において、抽出した同一鍵に対応する１乃至複数の打鍵イベントを、打鍵イベント発生順に記憶する打鍵イベント記憶手段を更に具えることを特徴とする請求項１又は２に記載の鍵盤楽器。
【発明の詳細な説明】
【技術分野】
【０００１】
この発明は、鍵を自動的に駆動する機構を備えた鍵盤楽器に関し、特に該鍵盤楽器における連打演奏など、再生の難しい演奏に関する再生性能の改良に関する。
【背景技術】
【０００２】
自動演奏ピアノにおいては、ピアノの各鍵に対応して設けられたソレノイドを演奏情報に基づき選択的に励磁することで、各ソレノイドに対応する鍵を押鍵駆動し、該押鍵駆動に応じて当該鍵に対応するハンマが打弦運動することで、当該ハンマに対応する弦を打弦する。なお、この明細書において、時間経過に対する鍵の位置変化を「軌道」と言う。自動演奏ピアノにおいては、演奏情報に基づき鍵の動きを規定する軌道データを作成し、該軌道データに基づき鍵の動きの制御（つまりソレノイドの駆動制御）することで、演奏情報に基づく自動演奏を行なうことが従来から知られている。

【０００３】
ピアノ奏法の一つとして、鍵をエンド位置に押し切る前に離鍵を開始する、或いは、離鍵途中から（鍵をレスト位置に戻しきらずに）次の押鍵に入る奏法、所謂ハーフストローク奏法がある。従来の自動演奏ピアノにおいて、ハーフストローク奏法を実現するための技術の一例として、押鍵時の鍵軌道（押鍵軌道）と離鍵時の鍵軌道（離鍵軌道）から両軌道の交差位置を調べて、両軌道が鍵のストロークエンド位置の手前で交差している場合には、押鍵途中から離鍵動作を或いは離鍵途中から押鍵動作を行なわせる技術があった（例えば、下記特許文献１を参照）。
【特許文献１】特許第３５４１４１１号公報
【０００４】
また、ハーフストローク奏法は、同一鍵を連続して押鍵する演奏（連打演奏）を行なう際に利用されることがある。自動演奏ピアノにおいて連打演奏を行なう場合には、連打演奏のノートオン（発音指示）の周期に対してハンマの打弦動作が追従できずに、正しい発音が行なわれれなくなる現象、所謂「音抜け」等の不具合が生じ易い。そこで、連打演奏の性能を向上させる技術として、例えば、再生すべきノートオン（発音指示）データによって指示される発音強度に対応する押鍵速度よりも速い押鍵を行なわせること、或いは、ノートオフ（消音指示）データによって指示されるタイミングよりも離鍵タイミングを早めることで、連続的な打弦（ハンマの往復動作）に要する時間を短縮する方法があった（例えば、下記特許文献２参照）。
【特許文献２】特許第３５５１５０７号公報
【０００５】
周知の通りアップライトピアノとグランドピアノとでは各々に具わるアクション機構の構造やその動作特性が異なっている。グランドライトピアノのアクション機構は連打性能に優れており、速い連打演奏にも対応できる（概ね１３Ｈｚ程度の周期で繰り返される連打まで対応可能と言われている）。一方、アップライトピアノのアクション機構は、連打性能についてグランドピアに対して劣っており、速い連打演奏には追従できない（概ね、８Ｈｚ程度の周期で繰り返される連打までしか対応できないと言われている）。
【０００６】
ところで、自動演奏ピアノにおいて再生する演奏情報は、自機において行なわれたピアノ演奏を記録したデータに限らない。例えば、アップライトピアノ型自動演奏ピアノで収録した演奏情報をグランドピアノ型自動演奏ピアノで再生する場合や、逆に、グランドピアノ型自動演奏ピアノで収録した演奏情報をアップライトピアノ型自動演奏ピアノで再生する場合など他の異なる種類のピアノで収録した演奏情報を再生するケース、或いは、ピアノ以外の機器で記録乃至作成された演奏情報を再生するケースなどが考えられる。
【０００７】
例えば、グランドピアノで収録した演奏情報をアップライトピアノ型の自動演奏ピアノで再生する場合、アップライトピアノに具わるアクション機構は、グランドピアノのアクション機構に比べて連打性能が劣るため、高速な連打演奏等に追従できず、良好な再生を行なうことができなかった。すなわち、従来の自動演奏ピアノにおいて、種類が異なる機器で収録された演奏情報に対して、自機で再生が不得手な部分を良好に再生できないという不都合があった。
【０００８】
上記の不都合について、前記特許文献１においては、演奏情報を収録した装置と、該演奏情報を再生する装置との機種の違いについて考慮していない。また、前記特許文献２においては、演奏情報の収録装置と再生装置の機種の違いを配慮した記載が見られるものの、この技術では、速い連打に対応させようとすると、再生される連打演奏の打弦タイミングや打弦速度が元の演奏情報から大きくズレてしまい、ピアノ演奏の再現性が損なわれるという不都合があった。更に、上記特許文献１乃至２に代表される従来の技術では、特にアップライトピアノにおける速い連打演奏の再生性能が不十分であった。
【発明の開示】

【発明が解決しようとする課題】
【０００９】
この発明は上述の点に鑑みてなされたもので、自動的に鍵盤を駆動して演奏を行う鍵盤楽器において、例えば連打打鍵演奏の部分など、自機で再生が不得手な部分を良好に再生できるようにした鍵盤楽器を提供しようとするものである。
【課題を解決するための手段】
【００１０】
この発明は、鍵と、前記鍵を駆動するための駆動手段と、打鍵イベントのシーケンスからなる演奏情報を供給するための供給手段と、前記演奏情報を構成する打鍵イベントの中から、同一鍵に対応する打鍵イベントを１乃至複数抽出する抽出手段と、前記抽出された１乃至複数の打鍵イベントのうち、連続して発生する複数の打鍵イベントを一群の打鍵イベント群に含めるかどうかを所定の閾値に基づき判断する判断手段と、前記判断手段により一群の打鍵イベント群に含めるよう判断された複数の打鍵イベントからなる打鍵イベント群内において、各打鍵イベントの少なくとも１つの楽音要素を揃えるよう修正を行なう修正手段と、前記修正手段により修正された打鍵イベントに基づき、該打鍵イベントに対応する鍵を駆動するために前記駆動手段を制御する制御手段とを具える鍵盤楽器である。
【００１１】
また、この発明は、鍵と、前記鍵を駆動するための駆動手段と、打鍵イベントのシーケンスからなる演奏情報を供給するための供給手段と、前記演奏情報を構成する打鍵イベントの中から、同一鍵に対応する打鍵イベントを１乃至複数抽出する抽出手段と、前記抽出された１乃至複数の打鍵イベントのそれぞれに基づき、駆動されるべき鍵の打鍵運動を規定するために、少なくとも該鍵の動きの位置成分の時間的変遷を表す軌道データを１乃至複数作成する軌道データ作成手段と、前記作成された１乃至複数の軌道データのうち、連続して発生する複数の軌道データを一群の軌道データ群に含めるかどうかを所定の閾値に基づき判断する判断手段と、前記判断手段により一群の軌道データ群に含めるよう判断された複数の軌道データからなる軌道データ群内において、各軌道データの少なくとも１つの楽音要素を揃えるよう修正を行なう修正手段と、前記修正手段により修正された軌道データに基づき、該軌道データに対応する鍵を駆動するために前記駆動手段を制御する制御手段とを具える鍵盤楽器である。
【発明の効果】
【００１２】
この発明によれば、演奏情報を構成する打鍵イベントの中から、同一鍵に対応する打鍵イベントを１乃至複数抽出し、該抽出された１乃至複数の打鍵イベントのうち、連続して発生する複数の打鍵イベントを一群の打鍵イベント群に含めるかどうかを所定の閾値に基づき判断する判断し、一群の打鍵イベント群に含めるよう判断された複数の打鍵イベントからなる打鍵イベント群内において、各打鍵イベントの少なくとも１つの楽音要素を揃えるよう修正を行なうことで、該打鍵イベント群内の楽音要素の不揃いを揃えることができる。従って、再生の難しい連打打鍵など、当該鍵盤楽器が再生不得手な演奏であっても、これを一群の打鍵イベント群と捉えて、該打鍵イベント群内の楽音要素を揃えることで、例えば連打演奏内の急激な楽音要素の変化等を無くし、安定した良好な再生を行なうことができるという優れた効果を奏する。
【００１３】
また、この発明によれば、前記抽出された１乃至複数の打鍵イベントのそれぞれに基づき、駆動されるべき鍵の打鍵運動を規定するために、少なくとも該鍵の動きの位置成分の時間的変遷を表す軌道データを１乃至複数作成し、作成された１乃至複数の軌道データのうち、連続して発生する複数の軌道データを一群の軌道データ群に含めるかどうかを所定の閾値に基づき判断し、一群の軌道データ群に含めるよう判断された複数の軌道データからなる軌道データ群内において、各軌道データの少なくとも１つの楽音要素を揃えるよう修正を行なうことで、該軌道データ群内の楽音要素の不揃いを揃えることができる。従って、再生の難しい連打打鍵など、当該鍵盤楽器が再生不得手な演奏であっても、これを軌

道データ群と捉えて、該軌道データ群内の楽音要素を揃えることで、例えば連打演奏内の急激な楽音要素の変化を無くし、安定した良好な再生動作を行なうことができるようになるという優れた効果を奏する。

【発明を実施するための最良の形態】

【００１４】

以下、添付図面を参照してこの発明の一実施例について説明する。

図１は、この発明の一実施例に係る自動演奏ピアノの構成例を説明するための図であって、機械的な発音機構の要部を抽出して示すと共に、電気的制御系の機能ブロックを示している。図１に示すように、自動演奏ピアノは、鍵１と、該鍵１の運動をハンマに伝達するためのアクション機構２と、対応する鍵１の運動に連動して打弦運動するハンマ３と、該ハンマ２によって打撃される弦４と、電気的制御に基づき鍵１を駆動する電磁ソレノイド５とを含む発音機構を具える。この実施例においては、アップライトピアノ型自動演奏ピアノを想定しており、図１には弦４が略鉛直に配置されるアップライトピアノ型の発音機構が示されている。上記の構成は、一般的なアップライトピアノ型自動演奏ピアノと同様である。なお、後述するように、この実施例においては、電磁ソレノイド５の駆動をサーボ制御する自動演奏制御構成が適用されている。このため、ソレノイド５には、プランジャ動作を検出して、その検出出力をフィードバック信号として後述するサーボコントローラ１２に供給するためのプランジャセンサ８が具備される。

【００１５】

鍵１の下面側には、鍵１の動きに応じた適宜の物理量（位置、速度等）を検出するためのキーセンサ６が配設されている。キーセンサ６の出力は、後述する演奏記録部１３とサーボコントローラ１２の双方に供給され、演奏収録時の演奏情報の作成と自動演奏制御のサーボ制御とに利用される。また、符号７はハンマ３の動きに応じた適宜の物理量（位置、速度等）を検出するためのハンマセンサである。ハンマセンサ７の出力信号は、演奏記録部１３に供給されて、当該演奏収録時の演奏情報の生成に利用される。キーセンサ６とハンマセンサ７は、センサ出力信号に基づき、押鍵タイミング、押鍵速度、離鍵タイミング、離鍵速度、打弦タイミング、打弦速度等、ピアノ演奏操作に関わる種々の情報を計測することができるセンサであれば、従来から知られるどのようなセンサ構成を採用してもよい。この実施例では、キーセンサ６及びハンマセンサ７の一例として、鍵１乃至ハンマ３の動作ストロークの全工程について連続的な位置情報を示すアナログ信号を出力可能な光学式の位置センサを採用するものとする。周知のとおり位置情報を適宜微分することで速度情報を得ることができるので、キーセンサ６乃至ハンマセンサ７として位置センサを採用した場合も打弦速度等の速度情報を取得できる。

【００１６】

鍵１は、バランスピンＰに貫通された位置を凡その支点として、上下揺動可能に支持されており、非押鍵状態（外力を加えない状態）では図１に示すレスト位置（ストローク量０ｍｍの位置）にあり、該レスト位置からエンド位置（この実施例ではレスト位置から１０ｍｍ押し込んだ位置）の範囲で往復動可能である。ソレノイド５が駆動（励磁）されると、ソレノイド５のプランジャが上方に突出して、鍵１の後端を突き上げる。これに応じて、鍵１はバランスピンＰを支点に揺動して、その演奏者側端部（図において右側）が下がり、押鍵が行なわれる。鍵１の押鍵動作に連動して、アクション機構２が作動して、ハンマ３が回動して弦４を打弦することで、ピアノ音が発音される。また、ソレノイド５の消磁に応じてプランジャが下がれば、該プランジャによる鍵１の後端突き上げが解除されるので、鍵１はレスト位置に戻る。これが離鍵の動作である。なお、この明細書において押鍵と該押鍵に対応する離鍵を合わせて「打鍵」と言う。

【００１７】

なお、図１では、或る１つの鍵１についての機械的な発音機構の要部のみを示しているが、自動演奏ピアノの鍵盤を構成する複数（典型的には８８個）の鍵の各々に対して図１に示すものと同様な発音機構が具備される。

【００１８】

図２は図１に示す自動演奏ピアノの電気的ハードウェア構成の概略を示すブロック図である。図２に示すように自動演奏ピアノは、ＣＰＵ２０、ＲＯＭ２１、ＲＡＭ２２、記憶装置２３、インターフェース（Ｉ／Ｏ）２４を含み、各装置間がデータ及びアドレスバス２０Ｂを介して接続される。ＣＰＵ２０は、自動演奏ピアノの全体的な動作を制御すると共に、この自動演奏ピアノの基本的な動作、即ち、演奏情報の再生処理やピアノ演奏の記録処理等、各種信号処理を実行する。この実施例において、前記の各種信号処理は、当該処理を実行するためのソフトウェアプログラムによって構成及び実施されるものとする。該ソフトウェアプログラムは、例えばＲＯＭ２１内に記憶されていてよい。

【００１９】

また、ＲＡＭ２２は、各種信号処理の実行中に発生した各種データを格納するワークメモリとして使用される。この実施例においては、ＲＡＭ２２には、この実施例においては、再生すべき演奏情報をロードするための領域や、この実施例に係る「鍵別打鍵イベントブロック記憶領域」や、作成された軌道データをバッファする領域等が用意される。

【００２０】

センサ２５は、図１に示すキーセンサ６、ハンマセンサ７及びソレノイド５に設けられたプランジャセンサ８に相当する。センサ２５の出力信号（アナログ信号）はＡＤ変換器を含むインターフェース（Ｉ／Ｏ）２４を介してディジタル信号に変換され、所定のサンプリング周期で制御系に取り込まれる。また、後述する演奏情報再生時に生成される軌道データは、ＰＷＭ発生器２６を介してＰＷＭ形式の電流信号（ＰＷＭ信号）に変換され、ソレノイド５に供給される。

【００２１】

記憶装置２３は、ハードディスク、フレキシブルディスク又はフロッピー（登録商標）ディスク、コンパクトディスク（ＣＤ‐ＲＯＭ）、光磁気ディスク（ＭＯ）、ＺＩＰディスク、ＤＶＤ（Ｄｉｇｉｔａｌ　Ｖｅｒｓａｔｉｌｅ　Ｄｉｓｋ）等、適宜の記録媒体で構成されてよい。記憶装置２３は、再生すべき演奏情報（自動演奏データ）の供給手段として、或いは、ピアノ演奏を記録するための記録媒体として利用されるもので、複数曲分の曲データファイル（複数曲分の演奏情報）を記憶することができる。

【００２２】

なお、当該自動演奏ピアノには、この他にも演奏情報の再生や演奏記録に関する各種指示をユーザが入力するための操作子群を含む操作部や該操作子部を駆動・検出するための機構、その機構を制御するためのソフトウェアや、表示器や、電子音源装置や、その他の外部機器（コンピュータやＭＩＤＩ機器など）、インターネット等の通信ネットワークに接続するためめの通信インターフェース等が適宜に具備されてよい。

【００２３】

自動演奏ピアノの基本的な機能である「演奏情報の再生」と「ピアノ演奏の収録」の概要について説明する。図１において、再生前処理部１０、モーションコントローラ１１及びサーボコントローラ１２は、演奏データの再生処理を担うモジュールである。また、演奏記録部１３及び記録後処理１４は、キーセンサ６及びハンマセンサ７の出力に基づき、演奏者が行ったピアノ演奏の演奏内容を記録する処理（演奏の記録処理）を担うモジュールである。また、この実施例において、これら各モジュールにて実施する信号処理は、ＣＰＵ２０が実行するソフトウェアプログラムによって実現される。

【００２４】

ユーザは、図示しない操作パネル上のスイッチ操作等により、ピアノ演奏の収録の開始指示を行うことができる。演奏記録部１３は、キーセンサ６及びハンマセンサ７から供給されるセンサ出力信号を取り込み、取り込んだセンサ出力信号に基づき、押鍵タイミング、押鍵速度、離鍵タイミング、離鍵速度、打弦タイミング、打弦速度等、ピアノ演奏に関わる種々の情報を計測する。記録後処理部１４は、前記計測された種々の情報に対して所定の正規化処理を行なった後、該種々の情報に基づきＭＩＤＩ形式等の適宜のデータフォーマットの演奏情報（自動演奏データ）を生成する。生成された演奏情報は、例えば１つの曲データファイルとして記憶装置２３に記録されたり、図示しない通信ネットワークを

介して該ネットワーク上の他の装置にリアルタイムで供給されてよい。なお、前記正規化処理とは、ピアノの個体差を吸収するための処理である。センサにより検出した種々の物理情報は、各ピアノにおけるセンサの位置や、構造上の違い、あるいは、機械的誤差によって各ピアノ固体に固有の傾向を持つものとなるため、標準となるピアノを想定して、該基準となるピアノに対応する演奏情報に変換する正規化処理を行なうのである。

【００２５】

演奏情報の構成例を図３に示す。この実施例では、一例として演奏情報のデータはＳＭＦ（スタンダードＭＩＤＩファイル）形式で記述されるものとする。図３に示す通り、演奏情報は、或る楽曲の自動演奏を実現するためのＭＩＤＩイベント（打鍵イベント）により構成される。打鍵イベントにはノートオンイベント、ノートオフイベント及びデルタタイムの３つのメッセージが含まれる。ノートオンイベント及びノートオフイベントにおいてステータスバイト（ノートオン＝「９ｎ」、ノートオフ＝「８ｎ」）は、メッセージ種類の識別子とＭＩＤＩチャンネル番号からなる。ステータスバイトに続く１つ目のデータバイト「ｋｋ」はノート番号（音高）を表し、２つ目のデータバイト「ｖｖ」はベロシティ値を表す。ノート番号「ｋｋ」は、音高を１２８段階のＭＩＤＩ値で記述したデータであり、自動演奏ピアノ用の演奏情報ではノート番号「ｋｋ」はピアノ鍵盤の８８鍵の各鍵に割り当てられたキーナンバ（１～８８）にそれぞれ対応付けられる。ノートオンのベロシティ値は楽音の音量（ハンマ３の打弦速度）を１２７段階のＭＩＤＩ値で表したデータであり、ノートオフのベロシティ値は離鍵速度を１２８段階のＭＩＤＩ値で表したデータとする。また、デルタタイムデータ「ｔｔ」は、ノートオンイベント又はノートオフイベントの発生タイミングを、直前イベントから当該イベントの間の相対的時間間隔を示すデータ（ＭＩＤＩ値）である。このデルタタイムの累積値により曲先頭から当該イベントまでの累積経過時間を得ることができる。なお、この明細書においては、曲の先頭から当該イベントまでの累積経過時間のことを「時刻」乃至「絶対時刻」と言い、また、デルタタイムが表す前後する２つのイベント間の相対的時間間隔を「相対時間」と呼ぶものとする。

【００２６】

《第１実施例》

図４は、この発明の第１の実施形態に係る演奏情報の再生処理の全体的な流れ一例を示すフローチャートである。なお、図４のフローチャートに示す再生処理の全体的な流れは、或る１つの鍵についての打鍵（再生処理）を表すものであり、同様な処理が８８鍵の全ての鍵について実行される。以下、図１と図４のフローチャートとを参照して、再生動作の概要を説明する。

【００２７】

ユーザは、例えば自動演奏ピアノのコントローラ等に具備された再生スタートスイッチの操作により任意の楽曲の再生（自動演奏）開始を指示できる。ユーザが或る楽曲について再生開始指示を行うと（ステップＳ１）、ステップＳ２においてサーボコントローラ１２が起動する。サーボコントローラ１２は当該再生処理とは別ルーチンで動作する。このサーボコントローラ１２の動作については後述する。

【００２８】

再生前処理部１０は、前記再生開始指示された楽曲の演奏情報を記憶装置２３や図示しないリアルタイム通信装置等から読み出して（ステップＳ３）、該読み出した演奏情報に対して適宜の正規化や単位合わせ等を行うと共に、図５を参照して後述する「鍵毎の打鍵イベント抽出」処理を行い（ステップＳ４）、該抽出された鍵毎のイベントに対して図７を参照して後述する「打鍵イベント群区切り」処理を行う（ステップＳ５）。後に詳しく述べる通り、この実施例に係る自動演奏の再生処理においては、ステップＳ４及びＳ５の処理により、同一鍵について連続する打鍵イベントを１群の「打鍵イベント群」として把握するところに主要な特徴を有する。なお、前記演奏情報に対する正規化とは、ピアノの個体差を吸収するための処理であり、同単位合わせとは、例えば、ＭＩＤＩ形式で記述された打鍵イベントのベロシティ値をミリメートル毎秒単位、デルタタイムをミリ秒単位等

の記述単位に変換する処理である。

【００２９】

モーションコントローラ１１においては再現すべき打鍵軌道をサーボコントローラ１２に指示するための軌道データを作成する処理が行なわれる（ステップＳ６）。「軌道データ」は押鍵軌道データとこれに後続する離鍵軌道データのセット、つまり１回の打鍵を構成する鍵の動きを表す。ここで作成された軌道データのデータはＲＡＭ２２に一時格納される。そして、前記ステップＳ６において作成された軌道データの再生タイミングが到来したら（ステップＳ７のｙｅｓ）、ステップＳ８において、現在のタイミングで再生すべき軌道データをＲＡＭ２２から読み出してサーボコントローラ１２に供給し、サーボコントローラ１２は該供給された軌道データに基づき鍵１の挙動をサーボ制御する。すなわち、サーボコントローラ１２は、モーションコントローラ１１から供給された軌道データ（軌道リファランスｒｅｆ）と、キーセンサ６及びプランジャセンサ８から帰還入力されるフィードバック信号とに基づく励磁電流（図２のＰＷＭ発生器２６によって発生されるＰＷＭ形式の電流信号）をソレノイド５に供給し、該ソレノイド５の駆動をサーボ制御する。これにより、鍵１が押鍵され、該鍵１の押鍵動作に応じてアクション機構２が作動して、ハンマ３が打弦運動する。

【００３０】

ステップＳ１０では、当該軌道データの軌道データを最後まで再生したかどうかチェックして、最後でなければ（ステップＳ７のｎｏ）、ステップＳ８以下を繰り返して、後続の軌道（押鍵軌道に対する離鍵軌道）を再生することで当該軌道データの再生を行なう。そして、１曲分の演奏情報の再生が終了するまでステップＳ６以降を繰り返す（ステップＳ８）。演奏情報が終了したら（ステップＳ８のｙｅｓ）、ステップＳ９においてサーボコントローラ１２を停止する。

【００３１】

図５は、前記図４のステップＳ４における「鍵毎の打鍵イベント抽出」処理の手順の一例を示すフローチャートである。ステップＳ１２において、前記図４のステップＳ３において読み出される演奏情報（打鍵イベント）を当該楽曲の１曲分全てＲＡＭ２２にロードする。ＲＡＭ２２上には、デルタタイムとイベント（ノートオン又はノートオフイベント）からなるシーケンスが、楽曲の先頭から終端までイベント発生順に順次書き込まれる（図３の演奏情報構成例を参照）。

【００３２】

ステップＳ１３では、ＲＡＭ２２にロードした演奏情報について、各打鍵イベントのメッセージ（ノートオン又はノートオフイベント、各イベントの発生タイミングを示すデルタタイム）を曲の先頭から順次読み取り、ステップＳ１４において、前記読み取った打鍵イベントのメッセージに基づき、図６に示す「鍵別打鍵イベントブロック記憶領域」を作成する。「鍵別打鍵イベントブロック記憶領域」は、１曲分の演奏情報に含まれる全ての打鍵イベントを８８鍵の各鍵別に管理するためのデータ記憶領域であって、ＲＡＭ２２において８８鍵の各鍵毎に１曲分の全ての打鍵イベントを格納しうる十分な容量を確保して用意される。

【００３３】

図６の「鍵別打鍵イベントブロック記憶領域」で使用する変数について説明する。

「Ｋ」はノート番号（キーナンバ）である。変数「ｉ」は当該鍵についての打鍵イベントの順番を示す「打鍵イベント番号」であって、曲の先頭からの通し番号で記述されている。なお、打鍵イベント番号はｉ≧１の整数であり、当該楽曲中の当該ノート番号について最後のイベント番号は変数Ｍで表す。変数「ｖｐｉ」は曲頭からｉ番目のノートオンベロシティ値である。また、変数「ｖｎｉ」は曲頭からｉ番目のノートオフベロシティ値（ノートオンベロシティｖｐｉに後続するノートオフベロシティ値）である。また、変数「ｔｐｉ」は、ノートオンベロシティ「ｖｐｉ」の発生タイミング、すなわち直前ノートオフ「ｖｎ（ｉ－１）」から「ｖｐｉ」までの相対時間である（但し、曲開始後の最初の打鍵イベントについては曲開始からの相対時間）。また、変数「ｔｎｉ」は「ｖｐｉ」とこ

れに後続する「ｖｎｉ」の間の相対時間である。

【００３４】

図６においては、特定の鍵「Ｋ」＝１について打鍵イベントの記述例を一例として示す。鍵別打鍵イベントブロック記憶領域の或る鍵「Ｋ」＝１の打鍵イベントブロックには、当該「Ｋ」＝１について、或る楽曲１曲中の全ての打鍵イベントが、打鍵イベント番号ｉ（ｉ＝１，２，３・・・Ｍ）によりイベント発生順に管理されており、各打鍵イベントについて「ｔｐｉ」，「ｖｐｉ」，「ｔｎｉ」，「ｖｎｉ」の各データが記憶されている。以下、他のノート番号Ｋ＝２～８８に対応する各打鍵イベントブロックもそれぞれ上記と同様な内部構造により構成される。

【００３５】

前記ステップＳ１４においては、前記ステップＳ１３にて読み出した打鍵イベントのメッセージに応じて、上記図６の鍵別打鍵イベントブロック記憶領域に「ｔｐｉ」，「ｖｐｉ」，「ｔｎｉ」，「ｖｎｉ」の各データを書き込むことで、当該楽曲についての鍵別打鍵イベントブロック記憶領域を作成する。打鍵イベントの読み出しは曲先頭からイベント発生順に行なわれるので、読み出されるイベントの順序は、或る１つの鍵についてのイベントにのみ着目すれば、デルタタイム、ノートオン、デルタタイム、ノートオフ、デルタタイム、ノートオン・・・の順である。従って、当該ステップＳ１４における鍵別打鍵イベントブロック記憶領域に対する上記各データの書き込み動作を、或る１つ鍵に対応する打鍵イベントブロックにのみ着目すれば、先ず、「ｔｐｉ」欄にデータが書き込まれ、ついで、「ｖｐｉ」欄、「ｔｎｉ」欄、そして、「ｖｎｉ」欄に順次データが書き込まれ、或る１つの打鍵イベント番号ｉに対応する１行が完成し、１つの打鍵イベントの記述が終る。１つの打鍵イベントの記述が終ると、次の打鍵イベント番号（ｉ＋１）の行の書き込みに移行する。

【００３６】

前記ステップＳ１４における鍵別打鍵イベントブロック記憶領域に対する各変数の書き込み処理を具体的に説明する。なお、打鍵イベントブロック記憶領域において、データを書き込むべき欄を「末尾欄」と呼ぶ。

前記ステップＳ１３において読み出した打鍵イベントがデルタタイムであった場合は、８８鍵の全ての鍵に対応する打鍵イベントブロックの各末尾欄（ｔｐｉ又はｔｎｉ）に読み出したデルタタイムの値を書き込む。ここで、ｔｐｉ又はｔｎｉ欄に既に何らかの値が書き込まれているものの、対応するｖｐｉ又はｖｎｉ欄（図６においてそれぞれ右隣の欄）が空欄である場合には、当該ｔｐｉ又はｔｎｉ欄を末尾欄とみなし、読み出したデルタタイムの値は当該ｔｐｉ又はｔｎｉに加算される。従って、対応するｖｐｉ又はｖｎｉ欄に値が書き込まれるまでは、当該ｔｐｉ又はｔｎｉの値が累積されることになる。

【００３７】

また、ノートオンイベントが読み出された場合には、当該ノート番号Ｋの打鍵イベントブロックの末尾欄（ｖｐｉの欄）にベロシティ値を書き込む。この時点で、当該ｖｐｉに対応するｔｐｉの値が確定する。確定したｔｐｉの値は、直前のノートオフｖｎ（ｉ－１）を書き込んだ時点から当該ｖｐｉを書き込んだ時点までに読み出したデルタタイムの累積値であるため、該直前ノートオフｖｎ（ｉ－１）から当該ｖｐｉまでの相対時間を表す値となる。

【００３８】

また、ノートオフイベントが読み出された場合には、当該ノート番号Ｋの打鍵イベントブロックの末尾欄（ｖｎｉの欄）にベロシティ値を書き込む。この時点で、当該ｖｎｉに対応するｔｎｉの値が確定する。当該ｖｎｉの直前のノートオンｖｐｉが書き込まれた時点以後に読み出されるデルタタイムイベントは、順次、当該ｖｎｉに対応するｔｎｉ欄に書き込み・加算される。すなわち、前記確定したｔｎｉの値は、直前のノートオンｖｐｉから当該ｖｎｉまでに読み出したデルタタイムの累積値であり、該直前ノートオンｖｐｉから当該ｖｎｉまでの相対時間を表す値となる。

【００３９】

上記ステップＳ１３のイベントの読み出しとステップＳ１４の処理を、当該曲の演奏情報の終りまで繰り返すことで、１曲分の演奏情報について「鍵別打鍵イベントブロック記憶領域」を作成する（ステップＳ１５）。

【００４０】

図７は、前記図４のステップＳ５における「打鍵イベント群区切り」処理の手順の一例を示すフローチャートである。ステップＳ１６において、ノート番号Ｋ＝１にセットして、処理対象となる鍵をノート番号Ｋ＝１に対応する鍵に設定する。ステップＳ１７において、打鍵イベント番号ｉ＝１にセットすることで、以下の処理を曲頭から１番目の打鍵イベントから始めるよう打鍵イベント番号ｉを初期化する。ステップＳ１８において、前記ステップＳ１７においてセットした変数ｉ＝１を変数Ｉにセットする。変数Ｉは、以下の処理において確定すべき「打鍵イベント群」の先頭の打鍵イベント番号ｉを定義するためのものである。ステップＳ１９において、変数ｊに「ｉ－Ｉ」すなわち前記ステップＳ１７でセットした変数ｉから前記ステップＳ１８でセットした変数Ｉを減算した値をセットする。変数ｊは、前記確定すべき打鍵イベント群内における打鍵イベント番号を表す変数である。この変数ｊにより、前記打鍵イベント群は、先頭番号「０」～群内終端番号「Ｎ」までのＮ＋１個の打鍵イベントからなる打鍵イベント群と表現できる。先に述べた通り、「ｊ＝ｉ－Ｉ」と定義されるので、打鍵イベント群は変数Ｉ～Ｉ＋Ｎ番目までのＮ＋１個の打鍵イベント群と表現することもできる。

【００４１】

ステップＳ２０において、現在の打鍵イベント番号ｉの値を１つインクリメント（ｉ＋１）して、前記打鍵イベント番号ｉに後続する打鍵イベント番号（ｉ＋１）を新規打鍵イベント番号ｉにセットする。そして、ステップＳ２１において、当該ノート番号Ｋについての鍵別打鍵イベントブロック記憶領域（図６参照）から、前記ステップＳ２０でセットした打鍵イベント番号ｉの「ｔｐｉ」を読み出す。ここで読み出した「ｔｐｉ」は、群内打鍵イベント番号ｊ（つまり、打鍵イベント番号ｉの直前の打鍵イベント）におけるノートオフ「ｖｎｊ」から当該打鍵イベント番号ｉのノートオン「ｖｐｉ」までの相対時間を表すデータである。

【００４２】

ステップＳ２２において、前記ステップＳ２１で読み出した「ｔｐｉ」の値に基づき、直前「ｖｎｊ」から「ｖｐｉ」までの相対時間の間隔を調べる。この実施例では、「ｔｐｉ」が５００ミリ秒（msec）以上であるかどうかを判断している。直前「ｖｎｊ」から「ｖｐｉ」までの相対時間の間隔（ｔｐｉ）が５００msec以下に近接している場合（ステップＳ２２のｎｏ）は、直前の打鍵イベント番号ｊと打鍵イベント番号ｉとを同一打鍵イベント群内に属する打鍵イベントと判断して、ステップＳ１９に戻る。ステップＳ１９では、現在の打鍵イベント番号（ｉ－Ｉ）つまり番号（ｊ＋１）を、新規の群内番号ｊにセットして、以下、ステップＳ２０～Ｓ２２により上記と同様な処理を行なう。

【００４３】

一方、直前「ｖｎｊ」から「ｖｐｉ」までの相対時間「ｔｐｉ」が５００msec以上離れている場合（ステップＳ２２のｙｅｓ）は、直前の群内打鍵イベント番号ｊと打鍵イベント番号ｉとは別の打鍵イベント群に属するものと判断して、打鍵イベント番号Ｉから現在の群内番号ｊまでを区切りとして、この区切りに含まれる全ての打鍵イベントを１群の「打鍵イベント群」として確定する。なお、ステップＳ２２のｙｅｓの場合、現時点の群内打番号ｊの値が群内終端番号「Ｎ」に相当するものである。ステップＳ２３では、前記確定した打鍵イベント群に対して「修正処理」のサブルーチンを行なう。この「修正処理」の詳細な内容は図８を参照して後述する。

ステップＳ２４では、当該鍵について、当該楽曲の終端まで打鍵イベントを調べたか（打鍵イベント番号ｉ＝Ｍに達したか）どうか判断し、未だ楽曲の終りまで処理していなければ（ステップＳ２４のｎｏ）、ステップＳ１８に戻り、現時点の打鍵イベント番号ｉを変数Ｉにセットすることで、次ぎに確定すべき「打鍵イベント群」の先頭の打鍵イベント番号ｉを定義し、以下、上記と同様な処理によって新たな打鍵イベント群を確定する。当

該鍵について当該楽曲の終りまで処理したら、つまり打鍵イベント番号ｉ＝Ｍの場合（ステップＳ２４のｙｅｓ）、ステップＳ２５においてノート番号Ｋをインクリメント（Ｋ＋１）することで新たな鍵を処理対象に設定し、ステップＳ１７以下の処理により、８８鍵の全ての鍵について当該楽曲の終りまで打鍵イベント群の確定を行なう（ステップＳ２６）。

【００４４】

以上の処理により、同一鍵について、或る打鍵イベントのノートオンタイミングと直前打鍵イベントのノートオフタイミングが５００msec以内の間隔で連続している「打鍵イベント群」を確定する。言い換えれば、「打鍵イベント群」は打鍵イベント発生間隔が５００msec以内の連打打鍵のイベント群と捉えることができる。なお、前記ステップＳ２２において当該打鍵イベント番号ｉ＝Ｍ、すなわち、当該曲中の当該鍵の最終打鍵イベントの場合も、ステップＳ２２をｙｅｓに分岐して、打鍵イベント群を確定する。

【００４５】

図８は、前記ステップＳ２３に示す打鍵イベント群に対する「修正処理」の手順の一例を示すフローチャートである。以下に説明する修正処理により打鍵イベント群内の各打鍵イベントを分析し、該各打鍵イベントを修正する。なお、処理対象の打鍵イベント群内の打鍵イベント番号ｉはｉ＝［Ｉ～Ｉ＋Ｎ］であり、これは群内番号ｊ［０～Ｎ］に対応する。

先ず、ステップＳ２７において、前記ステップＳ１９においてセットされてた群内番号ｊの値に応じて当該打鍵イベント群内の打鍵イベント数を確定し、該確定したイベント数に基づき変数Ｊをセットする。当該修正処理は、前述の通りステップＳ２２の判断により、打鍵イベント群の区切りを確定した後に行なわれるので、現時点の群内番号ｊは群内終端番号「Ｎ」となっている。従って、現時点の群内番号ｊに対応して打鍵イベント数を確定することができる。ここでは、群内番号ｊの値を変数Ｊに記憶するものとする。前述の通り、変数ｊは「０」～「Ｎ」までの整数であるから、変数Ｊの値は「実際のイベント数－１」の数値となる。

【００４６】

ステップＳ２８において、変数Ｊの値が０より大きいか調べる。変数ｊは「０」～「Ｎ」までの整数であるから、変数Ｊ＝０の場合とは当該打鍵イベント群内のイベント数が１打鍵イベントのみの場合である。変数Ｊ＞０の場合はステップＳ２８をｎｏに分岐することで、打鍵イベント群内のイベント数が１つ（つまり単打）の場合には、以下の処理を行なわないようにしている。

【００４７】

変数Ｊ＞０、つまり打鍵イベント群内のイベント数が複数の場合（ステップＳ２８のｙｅｓ）、当該打鍵イベント群は同一鍵の連打打鍵を表す打鍵イベント群である。この場合、この実施例においては、打鍵イベント群内における各打鍵イベントの楽音要素の修正内容の一具体例として、ステップＳ２９～Ｓ３２により、処理対象の打鍵イベント群内の全ての打鍵イベントに対して、ノートオンベロシティｖｐｉ、ノートオフベロシティｖｎｉ、ノートオン発生タイミングを示す相対時間ｔｐｉ及びノートオフ発生タイミングを示す相対時間ｔｎｉの各データの算術平均（相加平均）を求め、各データを求めた平均値に均一化する処理を行なう。

【００４８】

ステップＳ２９では、下記の式（１）により、当該打鍵イベント群内の全てのノートオンベロシティｖｐｉを平均化し、打鍵イベント群内のノートオンベロシティ平均値ｖｐａｖを得る。なお、式（１）において群内番号の変数ｊは「０」から「Ｊ」である。

【数１】

$$vpav=\sum_{j=0}^{J}(vpj)\ /\ (J+1)\cdots\text{式（１）}$$

【００４９】

ステップＳ３０では、下記の式（２）により、当該打鍵イベント群内の全てのノートオフベロシティｖｎｉを平均化し、打鍵イベント群内のノートオフベロシティ平均値ｖｎａｖを得る。なお、式（２）において群内番号の変数ｊは「０」から「Ｊ」である。

【数２】

$$vnav=\sum_{j=0}^{J}(vnj)/(J+1) \quad \cdots \text{式（２）}$$

【００５０】

ステップＳ３１では、下記の式（３）により、当該打鍵イベント群内のイベント相互時間（ノートオンタイミング）ｔｐｉを平均化し、打鍵イベント群内のノートオンタイミング平均値ｔｐａｖを得る。なお、式（３）において群内番号の変数ｊは「１」から「Ｊ」である。これにより、当該打鍵イベント群内の先頭のノートオンの発生タイミングｔｐ０を修正せずに保持する。従って、直前打鍵イベント群の末尾から当該打鍵イベント群の先頭までの相互間隔には影響を与えない。

【数３】

$$tpav=\sum_{j=1}^{J}(tpj)/(J) \quad \cdots \text{式（３）}$$

【００５１】

ステップＳ３２では、下記の式（４）により、当該打鍵イベント群内のイベント相互時間（ノートオフタイミング）ｔｎｉを平均化し、打鍵イベント群内のノートオフタイミング平均値ｔｎａｖを得る。なお、式（４）において群内番号の変数ｊは「０」から「Ｊ－１」である。これにより、当該打鍵イベント群内の末尾のノートオフの発生タイミングｔｎＪを修正せずに保持する。従って、当該打鍵イベント群の末尾から後続する打鍵イベント群の先頭の相互間隔に影響を与えない。

【数４】

$$tnav=\sum_{j=0}^{J-1}(tnj)/(J) \quad \cdots \text{式（４）}$$

【００５２】

上記ステップＳ３１とＳ３２の処理により、当該打鍵イベント群の先頭のノートオン発生タイミングと、当該打鍵イベント群の末尾のノートオフ発生タイミングを修正しないようにしているので、ステップＳ２９～Ｓ３２による平均化範囲は後述図９（ｂ）において矢印Ａで示す範囲となる。なお、当該打鍵イベント群内の末尾のノートオン発生タイミングｔｐＪも結果として変化しない。

【００５３】

ステップＳ３３では、前記図６に示す鍵別打鍵イベントブロック記憶領域において、当該打鍵イベント群に含まれる各打鍵イベント毎の値を、上記ステップＳ２９～Ｓ３２による修正結果に基づき書き換える。すなわち、当該打鍵イベント群に含まれる全ての打鍵イベント番号ｊの各ノートオンベロシティｖｐｊを上記ステップＳ２９で求めたノートオンベロシティ平均値ｖｐａｖに、また、各ノートオフベロシティｖｎｊを上記ステップＳ３０で求めたノートオフベロシティ平均値ｖｎａｖにそれぞれ書き換える。また、当該打鍵イベント群内の先頭の打鍵イベント番号［ｊ＝０（つまり、ｉ＝Ｉ）］におけるノートオン発生タイミングｔｐ０を除く、当該打鍵イベント群内の全てのｔｐｊを上記ステップＳ３１で求めたノートオン発生タイミング平均値ｔｐａｖに書き換える。また、当該打鍵イベント群内の末尾の打鍵イベント番号［ｊ＝Ｊ（つまり、ｉ＝Ｉ＋Ｎ）］におけるノートオフ発生タイミングｔｎＪを除く、当該打鍵イベント群内の全てのｔｎｊを上記ステップＳ３２で求めたノートオン発生タイミング平均値ｔｎａｖに書き換える。

【００５４】

図９は或る１つの打鍵イベント群についての修正処理を模式的に示す図である。図９に

おいて、縦軸はベロシティ値、横軸は時間ｔを示し、上向き矢印によりノートオンイベント、下向き矢印によりノートオフイベントを示す。すなわち、各矢印の長さにより当該イベントのベロシティ値（ｖｐｉ又はｖｎｉ）を表し、横軸上の位置は各イベント間の相互時間間隔（ｔｐｉ又はｔｎｉ）に基づく。また、同一打鍵イベント群に属するノートオン・ノートオフイベントを実線で示し、他の打鍵イベント群に属するものは点線で示した。（ａ）は修正前の打鍵イベント群の模式図、（ｂ）は図８の修正処理により平均化を施した打鍵イベント群の模式図である。なお、当該打鍵イベント群内の各打鍵イベント番号ｉは「Ｉ」から「（Ｉ＋Ｎ）」であり、先に述べた通り、変数ｊ＝ｉ－Ｉと定義されている。上記ステップＳ２７～Ｓ３３の修正処理により、当該打鍵イベント群内の各打鍵イベントは、図９（ａ）に示す修正前の状態から（ｂ）修正後の状態、つまり、各ｖｐｊが平均値ｖｐａｖに、また、各ｖｎｊが平均値ｖｎａｖに、また、ｔｐ０以外の全てのｔｐｊが平均値ｔｐａｖに、ｔｎＪ以外の全てのｔｎｊが平均値ｔｎａｖに修正される。これにより、当該打鍵イベント群内の楽音要素の不揃いを均一に揃えることができる。

【００５５】

当該打鍵イベント群について鍵別打鍵イベントブロック記憶領域の書き換えが終ると、図８に示す「修正処理」のルーチンを抜けて、図７に示すステップＳ２５以降に戻り、８８鍵の全ての鍵の全ての打鍵イベント群に対して図８に示す「修正処理」のルーチンを行い、各打鍵イベント群内の各打鍵イベントを修正する。

【００５６】

図１０はモーションコントローラ１１が実行する「軌道データ作成処理（上記図４のステップＳ６）」の手順の一例を示すフローチャートである。この実施例において、軌道データ作成に利用される打鍵イベントは、図７及び図８の処理により打鍵イベント群毎に適宜修正された打鍵イベントである。なお、図１０において、或る１つの鍵についての１つの打鍵イベントに対応する軌道データ作成について説明する。また、図１０においては、押鍵から離鍵に至る軌道データの作成について説明するが、離鍵から押鍵に至る軌道データの作成についても大略同様なアルゴリズムを適用できる。

【００５７】

ステップＳ３４において、当該鍵に対応する鍵別打鍵イベントブロック記憶領域から、１つの打鍵イベントについてノートオンイベント、つまり、ノートオンベロシティｖｐｉのデータと、当該ノートオンの発生タイミングを示すｔｐｉを取得する。ノートオンベロシティｖｐｉは、図３を参照して前述した通りハンマ３による打弦速度ＶＨを示すデータであり、該ｖｐｉの発生タイミングを示すｔｐｉは即ち打弦時刻ＴＨに対応する。ｔｐｉは相対時間（デルタタイム）のデータであるが、曲先頭から当該イベントまでのデルタタイム（ｔｐｉ及びｔｎｉ）の累積値により、打弦時刻ＴＨの絶対時刻を得ることができる。すなわち、打弦時刻ＴＨは当該ノートオンイベントの発生時刻を曲先頭からの絶対時刻で表すデータである。

【００５８】

ステップＳ３５において、打弦速度ＶＨ及び打弦時刻ＴＨに基づき、押鍵リファランス速度Ｖｒと押鍵リファランス時刻Ｔｒを算出する。押鍵リファランス速度Ｖｒとは、鍵１の所定のストローク位置（リファランス位置Ｘ）における鍵の押鍵速度を指す。リファランス位置Ｘとは、当該位置における鍵の速度を確定することで所望のハンマ打弦速度を高精度で再現できる鍵のストローク位置として定義された所定位置であり、これは概ねレスト位置から９．０～９．５ｍｍ鍵を押し込んだ位置が適切であることが実験等により判っている。このリファランス位置Ｘにおいて、打弦速度ＶＨを忠実に再現するための押鍵リファランス速度Ｖｒは直線近似関数式（５）によって推定できる。

Vr=α*VH+β　・・・式（５）

なお、この明細書において数式中の記号「＊」は乗算を示す。

【００５９】

また、押鍵リファランス時刻Ｔｒは、打弦時刻ＴＨにて打弦を実行するために鍵１が前記リファランス位置Ｘを通過すべき時刻である。打弦時刻ＴＨと押鍵リファランス時刻Ｔ

ｒの時間差Δｔとすると、該時間差Δｔと打弦速度ＶＨの関係は双曲線によって良好に近似されることが実験から判っている。従って、時間差Δｔは打弦速度ＶＨを分母とする１変数式（２）で近似することができる。

Δt=-(γ/VH)+δ・・・式（６）

【００６０】

上記式（２）により時間差Δｔが求まれば、打弦時刻ＴＨ（絶対時刻）から時間差Δｔを減算［ＴＨ－Δｔ］することで押鍵リファランス時刻Ｔｒを算出することができる。押鍵開始時刻ＴＲは、押鍵リファランス時刻Ｔｒ（絶対時刻）と、速度Ｖｒの速さで変位する鍵１がレスト位置（ストローク量０ｍｍ）から所定のリファランス位置Ｘまで変位するのにかかる相対時間の差分として求めることができる。すなわち、押鍵開始時刻ＴＲは、下記式（７）により求めることができる。

TR＝Tr-X/Vr・・・式（７）

【００６１】

従って、押鍵リファランス速度Ｖｒ及び押鍵リファランス時刻Ｔｒを満たす等速軌道は、レスト位置（鍵の押し込み量０ｍｍ）ＸＲとし、時刻ｔを絶対時刻とすると「Vr*(t-TR)+XR」で表すことができる。これにより、当該ノートオンイベントに対応する等速押鍵軌道を求めることができる。なお、ここでは等速軌道を想定しているので押鍵リファランス速度Ｖｒは当該等速軌道の押鍵速度に等しい。

なお、式（５）のα及びβと、式（６）のγ及びδは、それぞれ、ピアノ機種やリファランス位置Ｘの設定等に応じて実験によって決定される定数である。

【００６２】

ステップＳ３６において、当該鍵に対応する鍵別打鍵イベントブロック記憶領域から、前記ステップＳ３４にてノートオンを取得した打鍵イベントについて、ノートオフイベントｖｎｉのデータと、当該ノートオフの発生タイミングを示すｔｎｉを取得する。ノートオフベロシティｖｎｉは、図３を参照して前述した通り離鍵速度ＶＫＮ（＜０）を示すデータであり、該ｖｎｉの発生タイミングを示すｔｎｉは即ち離鍵時刻ＴＫＮに対応する。曲先頭から当該イベントまでのデルタタイム（ｔｐｉ及びｔｎｉ）の累積値により、離鍵時刻ＴＫＮの絶対時刻を得ることができる。すなわち、離鍵時刻ＴＫＮは当該ノートオフイベントの発生時刻を曲先頭からの絶対時刻で表すデータである。

【００６３】

ステップＳ３７において、離鍵速度ＶＫＮ及び離鍵時刻ＴＫＮに基づき離鍵リファランス速度ＶｒＮ（＜０）と離鍵リファランス時刻ＴｒＮを算出する。離鍵軌道については、ダンパーが弦４に当接する（弦４の振動を減衰開始させる）時点における鍵１のストローク位置を離鍵リファランス位置ＸＮとする。離鍵リファランス速度ＶｒＮは該離鍵リファランス位置ＸＮにおける鍵の速度であり、また、離鍵リファランス時刻ＴｒＮはストロークエンド位置から離鍵開始した鍵１が該離鍵リファランス位置ＸＮに達する時刻である。ここで、離鍵動作開始を基準時点（＝０）、鍵のストロークエンド位置をＸＥ（鍵の押し込み量１０ｍｍ）、鍵が離鍵リファランス位置ＸＮを通過する時刻をＴｒＮ´とすると離鍵リファランス位置ＸＮは式（８）となる。

XN＝VrN＊TrN´＋XE・・・式（８）

［なお、等速軌道を想定しているので初速度＝ＶｒＮ＝ＶＫＮ（＜０）］

【００６４】

上記式（８）からＴｒＮ´を求めることができる。ＴｒＮ´は鍵１が速度ＶｒＮの速さでエンド位置ＸＥから離鍵リファランス位置ＸＮまで変位するのにかかる相対時間であるから、該ＴｒＮ´と離鍵リファランス時刻ＴｒＮ（絶対時刻）との差分から、絶対時刻で表現した離鍵開始時刻ＴＥＮを求めることができる。

離鍵開始時刻ＴＥＮが決まれば、離鍵リファランス速度ＶｒＮ及び離鍵リファランス時刻ＴｒＮを満たす等速離鍵軌道は、時刻ｔを絶対時刻とすると「VrN*(t-TEN)+XE」で表すことができる。これにより、当該ノートオフイベントに対応する等速離鍵軌道を求めることができる。

【００６５】
ステップＳ３８において、前記ステップＳ３５で求めた押鍵軌道と、前記ステップＳ３７でも止めた離鍵軌道を１組の軌道データとして、ＲＡＭ２２に格納する。前記軌道データは、レスト位置を時刻ＴＲに出発して時刻ＴＥにエンド位置に到着する押鍵速度Ｖｒの押鍵軌道と、時刻ＴＥから時刻ＴＥＮの静止区間（速度＝０）と、時刻ＴＥＮにエンド位置を出発して時刻ＴＲＮにレスト位置に到着する離鍵速度ＶｒＮの離鍵軌道とからなる。この軌道データのデータが軌道リファランスｒｅｆとしてサーボコントローラ１２に供給される。
【００６６】
なお、上記軌道データ作成処理において、所謂ハーフストローク奏法に対応すべく、押鍵軌道と離鍵軌道の軌道交差判定を行ない、軌道が交差する（ハーフストローク奏法である）場合には、両軌道の交差時刻を算出し、該交差時刻で押鍵軌道と離鍵軌道の切り替えを行なう交差軌道データを作成してもよい。また、作成する軌道は等速軌道（直線軌道）に限らず曲線軌道等であってもよい。
【００６７】
図１１は、当該自動演奏ピアノにおけるサーボ制御の制御構成の一例を機能的に示すブロック図である。以下に、当該自動演奏ピアノにおける鍵１の打鍵操作のサーボ制御の動作（サーボコントローラ１２の動作）の一例について図１１を参照して説明する。
【００６８】
目標値生成部５０には、図１０を参照して説明した処理により生成された軌道データ（軌道リファランスｒｅｆ）が、当該軌道データの供給タイミングに応じて供給される。目標値生成部５０は該供給された軌道リファランスに基づき、或る時刻ｔにおける位置目標値ｒｘ及び速度目標値ｒｖを生成する。速度目標値ｒｖは、或る時刻ｔにおける打鍵速度を例えばミリメートル毎秒単位で記述したデータである。この実施例では等速軌道を想定しているので、速度目標値ｒｖは常に一定であり、押鍵軌道については前記ステップＳ３５で求めた押鍵速度Ｖｒ、離鍵軌道については前記ステップＳ３７で求めた離鍵リファランス速度ＶｒＮである。また、位置目標値ｒｘは該或る時刻ｔにおける鍵の位置を表すデータであり軌道データに基づき生成できる。
【００６９】
目標値生成部５０で生成された速度目標値ｒｖと位置目標値ｒｘは、所定のサンプル時間毎（例えば１msec毎）に、後段の速度比較部５１、位置比較部５２にそれぞれ並行に送出される。速度比較部５１では、速度目標値ｒｖと速度成分のフィードバック信号ｙｖの差（速度偏差ｅｖ）を求める。また、位置比較部５２では、位置目標値ｒｘと位置成分のフィードバック信号ｙｘの差（位置偏差ｅｘ）を求める。
【００７０】
前記速度成分のフィードバック信号ｙｖと前記位置成分のフィードバック信号ｙｘは、ソレノイド５の制御量（プランジャ速度）ｙｍを検出する速度センサ（プランジャセンサ）８及び該ソレノイド５により打鍵された鍵１の制御量（鍵ストローク位置）ｙｋを検出する位置センサ（キーセンサ）６の出力に基づき生成される。速度センサ８から出力されたプランジャ速度ｙｍに基づく速度アナログ信号ｙｖｍａ及び位置センサ６から出力された鍵ストローク位置ｙｋに基づく位置アナログ信号ｙｘｋａはそれぞれＡＤ変換器（図２のインターフェース２４に相当）５６ａ，５６ｂを介して、速度ディジタル信号ｙｖｍｄと位置ディジタル信号ｙｘｋｄに変換される。正規化部５７ａ，５７ｂは、速度ディジタル信号ｙｖｍｄ及び位置ディジタル信号ｙｘｋｄに対して所定の正規化処理を行ない、速度ディジタル信号ｙｖｍｄ及び位置ディジタル信号ｙｘｋｄをそれぞれ正規化した「プランジャ速度値ｙｖｍ」と「鍵位置値ｙｘｋ」を生成する。なお、前記所定の正規化処理は例えば、速度ディジタル信号ｙｖｍｄの記述単位を速度目標値ｒｖの記述単位（例えばミリメートル毎秒単位）に換算したり、位置ディジタル信号ｙｘｋｄの記述単位を位置目標値ｒｘの記述単位（例えばミリメートル）に換算したりする処理や、各装置個体に固有の値ずれの補正などである。

【００７１】
速度生成部５８は、鍵位置値ｙｘｋを適宜微分演算（例えば多項式適合等）することにより、鍵１の速度情報（鍵速度値ｙｖｋ）を生成する。また、位置生成部５９は、プランジャ速度値ｙｖｍを適宜積分演算することにより、プランジャの位置情報（プランジャ位置値ｙｘｍ）を生成する。そして、速度成分加算部６０において、プランジャ速度値ｙｖｍと鍵速度値ｙｖｋを加算して一本化することで、速度成分のフィードバック信号ｙｖを生成する。この速度成分のフィードバック信号ｙｖは前記速度比較部５１へ帰還入力（負帰還）される。また、位置成分加算部６１において、鍵位置値ｙｖｋとプランジャ位置値ｙｘｍを加算して一本化することで、位置成分のフィードバック信号ｙｘを生成する。この位置成分のフィードバック信号ｙｘは前記位置比較部５２へ帰還入力（負帰還）される。

【００７２】
速度比較部５１で求めた速度目標値ｒｖと速度成分フィードバック信号ｙｖとの差（速度偏差ｅｖ）は、速度増幅部５３において所定のゲイン値Ｋｖで増幅された後、速度制御信号ｕｖとして加算器５５に供給される。また、位置比較部５２で求めた位置目標値ｒｘと位置成分フィードバック信号ｙｘの差（位置偏差ｅｘ）は、位置増幅部５４において所定のゲイン値Ｋｘで増幅された後、位置制御信号ｕｘとして加算器５５に供給される。加算器５５では、速度制御信号ｕｖと位置制御信号ｕｘを加算することで、これら各事象の制御信号を一本化する。この加算結果がソレノイド５を駆動するための制御信号ｕである。制御信号ｕはＰＷＭ発生器２６を介してＰＷＭ形式のソレノイド励磁電流信号ｕｉに変換され、この励磁電流信号ｕｉに基づきソレノイド５が駆動される。

【００７３】
以上説明した通り、この発明に係る第１実施例によれば、同一鍵について複数の打鍵イベントの相互間隔が５００msec以内に近接する場合に、該複数の打鍵イベントを連続する打鍵イベント群として捉えて、該打鍵イベント群内の楽音要素の不揃いを揃えることで、連打演奏内の急激な楽音要素の変化を無くし、安定した連打打鍵用の軌道データを供給できる。従って、再生の難しい連打打鍵など、自動演奏ピアノが再生不得手な演奏も、比較的安定して再生できるようになる。また、記録状態の不安定（楽音要素にバラツキの多い）な連打演奏や記録精度の悪い演奏情報の再生を行なう場合や、再生を行う自動演奏ピアノの動作確動作が悪い場合でも、安定した再生動作を行なうことができるようになる。

【００７４】
《第２実施例》
上記第１実施例においては、再生前処理部１０における処理（図４のステップＳ５）として、打鍵イベント（ＭＩＤＩデータ）に対して打鍵イベント群の区切り（図７）や、該打鍵イベント群に対する修正処理（図８）を行なう例を示した。これに対して、第２実施例では、モーションコントローラ１１において作成した軌道データに対して「軌道データ群」の区切りや、軌道データ群の修正処理を行なう。図１２は第２実施例の概要を説明するための図であって、複数の軌道データによる打鍵軌道の一例を示す模式図である。同図において、横軸に時間ｔをとり、縦軸に鍵のストローク位置をとる。また、ＸＲはレスト位置、ＸＥはエンド位置を示しており、ＸＭは鍵ストロークのレスト位置から鍵ストロークのエンド位置の中間位置を示す。なお、上向き矢印が押鍵軌道に対応し、下向き矢印が離鍵軌道に対応する。（ａ）は１群の軌道データ群であって、修正前のものを示し、（ｂ）に修正後の軌道データ群を示している。

【００７５】
図１３は、第２実施例に係る再生処理の全体の流れを示すフローチャートである。前記図４に示す第１実施例に係る再生処理と同様な構成については、適宜その説明を省略する。図１３に示す通り、第２実施例に係る再生処理においては、再生前処理部１０が読み出した演奏情報を鍵別に抽出する処理（ステップＳ４３）を行なうことで、前記図６に示すものと同様な鍵別打鍵イベントブロック記憶領域をＲＡＭ２２上に作成する。モーションコントローラ１１は、前記鍵別打鍵イベントブロック記憶領域から順次打鍵イベントを読

み出して、該読み出した打鍵イベントに基づき軌道データ作成処理（ステップＳ４４）を行なう。軌道データ作成処理自体は、前記図１０に示す処理と同様であり、作成する等速軌道としては等速軌道を想定する。モーションコントローラ１１において作成された軌道データは打鍵（押鍵と離鍵）発生順に鍵別軌道データブロック記憶領域に書き込まれる。鍵別軌道データブロック記憶領域は、前記鍵別打鍵イベントブロック記憶領域と同じものであってよい。すなわち、軌道データ作成に使用した打鍵イベントの行に対して、当該作成した軌道データに基づくデータを上書きすればよい。そうして、ステップＳ４５において、前記作成された軌道データについて「鍵軌道データ群区切り」処理を行なうところが前記図４の処理とは異なる。そして、以下、前記図４の処理と同様に、軌道データをサーボコントローラ１２に供給して、軌道データ毎に鍵１の駆動を行い（ステップＳ４６、Ｓ４７及びＳ４８）、１曲分の演奏情報が終るまで上記を繰り返す。

【００７６】

図１４は前記鍵別軌道データブロック記憶領域の構成例を示す図である。鍵別軌道データブロック記憶領域の構成の概要は第１実施例の鍵別打鍵イベントブロック記憶領域と同様であり、「Ｋ」と変数「ｉ」についても既述と同様である。なお、第２実施例では処理対象とするデータは軌道データであり「イベント」データではないが、本明細書においては１回の打鍵の発生を「イベント」と捉えて差し支えない。即ち、変数ｉは上記と同様に当該鍵について打鍵イベントの順番を示す打鍵イベント番号（打鍵番号）である。

【００７７】

図１４に示す鍵別軌道データブロック記憶領域においては、或る１つの鍵について軌道データに基づく物理量を下記の通り既述する。変数「ＶＰｉ」は、或る１つの鍵に着目した場合に、曲頭からｉ番目に到来する打鍵の押鍵速度Ｖｒに対応し、ミリメートル毎秒単位で記述された物理量のデータである。また、変数「ＶＮｉ」は、或る１つの鍵に着目した場合に、曲頭からｉ番目に到来する打鍵の離鍵速度ＶｒＮに対応し、ミリメートル毎秒単位で記述された物理量のデータである。また、変数「ＴＰｉ」は、前記図１２に示す通り、或る打鍵番号ｉに対して直前の打鍵番号（ｉ－１）の離鍵軌道が中央位置ＸＭを通過する時刻から当該打鍵番号ｉの押鍵軌道のエンド位置ＸＥ到着時刻までの時間差をミリ秒単位で記述したデータである。また、変数「ＴＮｉ」は、前記図１２に示す通り、或る打鍵番号ｉの押鍵軌道のエンド位置ＸＥ到着時刻から同打鍵番号ｉの離鍵軌道の中央位置ＸＭ通過時刻までの時間差をミリ秒単位で記述したデータである。

【００７８】

モーションコントローラ１１は、前記ステップＳ４４において作成した軌道データ（押鍵軌道と離鍵軌道）に基づき、上記「ＶＰｉ」、「ＶＮｉ」「ＴＰｉ」及び「ＴＮｉ」の各物理量のデータを求め、当該鍵についての鍵別軌道データブロック記憶領域における打鍵番号ｉの行に、該各物理量のデータをそれぞれ書き込む。

【００７９】

図１５は、前記ステップＳ４５における「軌道データ群区切り」処理の手順の一例を示すフローチャートである。この処理の大略は前記第１実施例の図７に示す「軌道データ群区切り」処理と同様である。すなわち、ノート番号Ｋ＝１にセットして、処理対象となる鍵をノート番号Ｋ＝１に対応する鍵に設定し、打鍵番号の変数ｉ＝１にセットして、変数ｉを初期化する（ステップＳ５１，Ｓ５２）。ステップＳ５３において、現在の変数ｉを変数Ｉにセットして、確定すべき「軌道データ群」の先頭の打鍵番号ｉを定義し、ステップＳ５４において、該確定すべき「軌道データ群」内での群内打鍵番号を表す変数ｊに「ｉ－Ｉ」をセットする。

【００８０】

ステップＳ５５において、当該ノート番号Ｋについての鍵別軌道データブロック記憶領域（図１４参照）から、前記ステップＳ５４でセットした群内番号ｊの離鍵速度データ「ＶＮｊ」を読み出して、変数ＶＮにセットする。ステップＳ５６において、現在の打鍵番号ｉの値をインクリメント（ｉ＋１）して、該現在の打鍵番号ｉに後続する打鍵番号（ｉ＋１）を新規打鍵番号ｉにセットする。ステップＳ５７では、当該ノート番号Ｋについて

の鍵別軌道データブロック記憶領域（図１４参照）から、前記ステップＳ５６でセットした打鍵番号ｉの時間差データ「ＴＰｉ」を読み出して、変数ＴＰにセットする。前記セットされた「ＴＰｉ」は前記ＶＮにセットされた離鍵速度データＶＮｊの離鍵軌道が中央位置ＸＭを通過する時刻から当該打鍵番号ｉの押鍵軌道のエンド位置ＸＥ到着時刻までの時間差である。

【００８１】

ステップＳ５８では、「TP－10/VN*1000」で求まる時間値（ミリ秒単位）が１００ミリ秒以上かどうかを調べることで、打鍵番号ｊの離鍵による鍵のレスト位置ＸＲへの戻り後の待機時間（後続打鍵番号ｉの押鍵開始までの間隔）が１００ミリ秒以上開いているかどうかを判定する。１００ミリ秒以内であれば（ステップＳ５８のｎｏ）、直前の郡内打鍵番号ｊと打鍵番号ｉとを同じ軌道データ群内に属する打鍵軌道と判断して、ステップＳ５４に戻り、現在の打鍵番号ｉつまり群内番号（ｊ＋１）を、新規の群内番号ｊにセットして、以下、ステップＳ５５～Ｓ５８により上記と同様な処理を行なう。

【００８２】

一方、離鍵レスト位置ＸＲ戻り後の待機時間が１００ミリ秒以上開いている場合（ステップＳ５８のｙｅｓ）は、直前の群内打鍵番号ｊと現在の打鍵番号ｉとは別の軌道データ群内に属する打鍵軌道と判断して、打鍵番号Ｉから打鍵番号ｊまでを区切りとして、この区切りに含まれる全ての軌道データを１群の「軌道データ群」として確定する。そして、ステップＳ５９のにおいて前記確定した軌道データ群に対して「修正処理」のサブルーチンを行なう。なお、当該打鍵番号ｉ＝Ｍ、すなわち、当該曲中の当該鍵の最後の打鍵の場合も、ステップＳ２２をｙｅｓに分岐して、鍵軌道データ群を区切る。

【００８３】

以下、当該鍵について順次新たな鍵軌道データ群を確定してゆき、当該楽曲の終端まで行なったら、別の鍵について鍵軌道データ群の確定を行なうことで、これを８８鍵の全てに行なう（ステップＳ６０～Ｓ６２）。

【００８４】

図１６は、前記ステップＳ５９に示す軌道データ群に対する「修正処理」の手順の一例を示すフローチャートである。処理対象となる軌道データ群の打鍵番号ｉはｉ＝［Ｉ～Ｉ＋Ｎ］であり、これは群内番号ｊ［０～Ｎ］に対応する。この処理において変数Ｊは、既述と同様に、現時点の群内番号ｊの値がセットされ、当該打鍵群内の打鍵数（正確には打鍵数－１）を表す。また、当該軌道データ群内の打鍵数が１つ（つまり単打）の場合には、以下の処理を行なわないようにしている（ステップＳ６３，Ｓ６４）。この実施例においては、当該軌道データ群内の各軌道データ修正の一具体例として、ステップＳ６５～Ｓ６８により、押鍵速度と離鍵軌道は幾何平均（相乗平均）により、また、時間差ＴＰｉと時間差ＴＮｉは算術平均（相加平均）により、各物理量毎に値を揃えるものとする。

【００８５】

ステップＳ６５では、下記の式（８）により、当該軌道データ群内の全ての押鍵速度ＶＰｉを幾何平均化することで、軌道データ群内の押鍵速度幾何平均値ＶＰａｖを得る。なお、式（８）において群内番号の変数ｊは「０」から「Ｊ」である。

【数５】

$$VPav=\left\{\prod_{j=0}^{J}(VPj)\right\}^{1/(J+1)} \quad \cdots \text{式（８）}$$

【００８６】

ステップＳ６６では、下記の式（９）により、当該軌道データ群内の全ての離鍵速度ＶＮｉを幾何平均化することで、軌道データ群内の離鍵速度幾何平均値ＶＮａｖを得る。なお、式（９）において群内番号の変数ｊは「０」から「Ｊ」である。

【数６】

$$VNav=\left\{\prod_{j=0}^{J}(VNj)\right\}^{1/(J+1)} \quad \cdots \text{式（９）}$$

【００８７】

ステップＳ６７では、下記の式（１０）により、当該軌道データ群内における押鍵時の時間差ＴＰｉを平均化することで、軌道データ群内の押鍵動き出しまでの時間の平均値ＴＰａｖを得る。なお、式（１０）において群内番号の変数ｊは「１」から「Ｊ」である。これにより、当該軌道データ群内の先頭打鍵の押鍵タイミングを修正せずに保持する。従って、直前軌道データ群の末尾から当該軌道データ群の先頭までの相互間隔には影響を与えない。

【数７】

$$TPav=\sum_{j=1}^{J}(TPj)/(J) \quad \cdots \text{式（１０）}$$

【００８８】

ステップＳ６８では、下記の式（１１）により、当該軌道データ群内における離鍵時の時間差ＴＮｉを平均化することで、軌道データ群内の離鍵動き出しまでの時間の平均値ＴＮａｖを得る。なお、式（４）において群内番号お変数ｊは「０」から「Ｊ－１」である。これにより、当該軌道データ群内の末尾打鍵の離鍵タイミングを修正せずに保持する。従って、当該軌道データ群の末尾から後続軌道データ群の先頭までの相互間隔には影響を与えない。

【数８】

$$TNav=\sum_{j=0}^{J-1}(TNj)/(J) \quad \cdots \text{式（１１）}$$

【００８９】

上記ステップＳ６５～Ｓ６８の処理による各修正結果は軌道データブロック記憶領域の対応する欄に上書きされる。この修正処理による修正結果は図１２（ｂ）に示す通りであり、その平均化（幾何平均と算術平均）範囲は図１２（ｂ）において矢印Ｂで示す範囲となる。図１２において（ａ）に示す元の軌道データ群に含まれる打鍵の楽音要素（押鍵速度、押鍵タイミング、離鍵速度、離鍵タイミング、押鍵深さ、離鍵戻し量等）のバラツキが、（ｂ）に示す通り均一化されるので、安定した連打打鍵の軌道データ群をサーボコントローラ１２に供給できるようになる。

【００９０】

従って、以上説明した第２実施例においても、同一鍵について複数の打鍵の相互間隔が近接する（離鍵のレスト位置戻りから１００msec以内に後続打鍵が始まる）場合には連続する軌道データ群として捉え、該軌道データ群内の楽音要素の不揃いを揃えることで、連打演奏内の急激な楽音要素の変化を無くし、安定した連打打鍵用の軌道データを供給できるようになり、再生の難しい連打打鍵など、自動演奏ピアノが再生不得手な演奏も、比較的安定して再生できるようになる。また、記録状態の不安定（楽音要素にバラツキの多い）な連打演奏や記録精度の悪い演奏情報の再生を行なう場合や、再生を行う自動演奏ピアノの動作確動作が悪い場合でも、安定した再生動作を行なうことができるようになる。

【００９１】

なお、上記第１実施例及び第２実施例において、図７に示す打鍵イベント群の区切り又は図１５に示す軌道データ群区切りの判断は、図７においては先行打鍵イベント末尾からの時間間隔（５００msec以上か？）、また、図１５においては離鍵レスト戻り後の待機時間（１００msec以上か？）をそれぞれ基準としたが、双方とも判断の基準として用いる時間設定は一例であって、これに限定されない。また、判断基準の変更例として、打鍵イベ

ント群又は軌道データ群の先頭からの時間経過（例えば２秒経過後等）に応じて当該群を区切るようにしたり、或いは、該先頭からの打鍵数（例えば１０打鍵経過後等）に応じて当該群を区切るようにしたりしてもよい。また、１つの打鍵イベントにおけるノートオンとノートオフ間の時間間隔ｔｎｉ（例えば１秒以上経過したか？等）に応じて打鍵イベント群を区切ってもよく、軌道データの場合にはノートオンとノートオフ間の間隔に準ずる適宜の時間間隔を基準にすればよい。また、ベロシティ値の違いの大きさを基準にしてもよい。例えば、或るノートオンベロシテ値ｖｐｉと後続する打鍵イベントのノートオンベロシテ値ｖｐ（ｉ＋１）の差が３２（ＭＩＤＩ値）より大きい場合とか、或るノートオンベロシテ値ｖｐｉとこれに続くノートオフベロシティ値ｖｎｉの差が１６（ＭＩＤＩ値）より大きい場合等に異なる打鍵イベント群と判断するようにしてもよく、押鍵速度・離鍵速度についても同種の基準で軌道データ群の判断をしうる。また、上記の各種判断基準を適宜組み合わせても良い。また、軌道データ群の場合は、更に、或る打鍵について、押鍵軌道と離鍵軌道の軌道交差や、交差しない場合であっても押鍵軌道と離鍵軌道の相互間隔（エンド位置での待機時間１００msec以内等、或いは、離鍵軌道と押鍵軌道の相互間隔（レスト位置での待機時間１００msec以内等）を条件に、軌道データ群の判断をしてもよい。

【００９２】

また、上記第２実施例において、図１４の鍵別軌道データブロック記憶領域の時間差ＴＰｉ、ＴＮｉは上記説明した変数に限らず、先行軌道の到着時刻から当該軌道の出発時刻までの時間差を設定してもよい。

【００９３】

また、図８又は図１６に示す修正処理においては、上記各実施例に示した群内の各楽音要素のデータ（打弦タイミング［ｔｐｉ，ＴＰｉに対応］、打弦速度［ｖｐｉ，ＶＰｉに対応］、止音タイミング［ｔｎｉ，ＴＮｉに対応］、止音速度［ｖｎｉ，ＶＮｉに対応］）の平均を求める処理例に限らず、処理対象の打鍵イベント群又は軌道データ群内の打鍵イベント乃至打鍵について平均打鍵周波数（つまり連打打鍵の周期の速さ）を求めるような変形例が可能である。この場合、平均打鍵周波数が装置性能の限界を越える場合に、平均打鍵周波数を該装置性能の限界値（例えば、アップライトピアノ型装置であれば８Ｈｚ等）にクリップさせることで、当該連打打鍵を当該装置にて再生可能な連打打鍵に修正することができる。この修正に際して、打鍵イベント群内に納まらない打鍵イベントが生じた場合には、これを削除して他の打鍵イベントを平均化するなど適宜の補正を行なうとよい。なお、この修正についても当該打鍵イベント群の先頭のノートオン発生タイミングと、当該打鍵イベント群の末尾のノートオフ発生タイミングを修正しないようにするとよい。

【００９４】

また、図８又は図１６に示す修正処理の別の変形例としては、処理対象の打鍵イベント群又は軌道データ群内の各楽音要素のデータの回帰直線或いは回帰曲線を求め、求めた回帰直線或いは回帰曲線に沿って各楽音要素を修正することができる。なお、図９（ｃ）には打鍵イベント群内の各楽音要素をそれぞれ回帰直線に沿って修正した例をに示し、また、図１２（ｃ）には軌道データ群内の各楽音要素をそれぞれ回帰直線に沿って修正した例をに示す。図９（ｃ）及び図１２（ｃ）に示す通り回帰直線に沿って楽音要素を修正することで、群内の各楽音要素の変化傾向に合わせた修正を行なうことができる。例えば、図９（ｃ）の例では、ベロシティが徐々に増大し、各イベント間の時間間隔が狭まって行く。このように、回帰直線或いは回帰曲線による修正を行なうことで、元の演奏情報が持つ音楽的特質を残しておくことができる。

【００９５】

更に、修正処理の別の変形例として、処理対象の打鍵イベント群又は軌道データ群内の各楽音要素のデータの標準偏差を求め、求めた標準偏差を保ちつつ各楽音要素に任意の補正（例えば乱数によりベロシティに揺らぎを作る等）することができる。これにより、修正後の演奏情報（打鍵イベント）に対して更なる音楽的特質を与えることも可能である。

また、鍵軌道データ群の修正処理においては、更なる変形例として、群内の各楽音要素に基づき軌道データを等速軌道から、等加速度軌道、等躍動軌道等の組み合わせに変更したり、押鍵深さや離鍵戻り量の修正などを施すようにしても良い。また、打鍵イベント群の修正処理においても、群内の各楽音要素に基づき、当該打鍵イベントに基づき作成されるべき軌道種類が決定されるようにしてもよい。

【００９６】

また、上記に挙げた修正処理の変形例を適宜に組み合わせても良いし、ユーザが修正メニュー等から所望の修正内容を選択できてもよい。

【００９７】

また、図８又は図１６に示す修正処理において、修正対象となる各楽音要素（打弦タイミング、打弦速度、止音タイミング、止音速度、更には軌道種類等）をユーザ選択できてもよい。また、修正対象となる各楽音要素に優先順位を設定し、元のデータの特徴を尊重する度合いを任意に調整できてもよい。

【００９８】

また、上記第１実施例及び第２実施例において、再生対象の演奏情報を収録した装置の種類（例えば、グランドピアノ型の装置、アップライトピアノ型の装置、或いは、その他の電子楽器等、と言った装置の種類）を判断する処理を更に行なうようにし、自機とは異なる種類の装置で収録した演奏情報について、前記図８又は前記図１６に示す修正処理を実行するようにしてもよい。この場合、例えば、演奏情報のヘッダー部に収録装置のＩＤ情報を記録しておくと共に、再生装置（自機）では自機の装置種類を自機ＩＤ情報により認識できるようにしておき、図４のステップＳ３又は図１３のステップＳ４２における演奏情報の読み出しに先立って、装置種類判断を行なうようにすればよい。これによれば、例えば、自機とは異なる種類の装置で収録された演奏情報に含まれる自機で不得手な演奏についても、自機の再生能力に適したデータに修正して、良好な再生を行なうことが可能である。

【００９９】

また、更に別の実施形態として、前記図８又は前記図１６に示す修正処理を実行するかどうかを、ユーザの選択操作に応じて任意に設定できてもよい。

【０１００】

また、上記実施例においては、図４のステップＳ４又は図１２のステップＳ４３の「鍵毎の打鍵イベント抽出」において、再生対象の演奏情報を１曲分ＲＡＭ２２にロードしてから処理する構成、つまり、外部からの演奏情報の読込みに対して非リアルタイムで処理する構成を想定している。この処理構成について変更例としては、再生対象の演奏情報を１曲分を完全にＲＡＭ２２にロードしてから、他の処理を開始する構成に限らず、ある程度纏まった量のＭＩＤＩイベント（例えば演奏時間にして数秒分程度のイベント）をロードしたら、演奏情報のロードに並行して他の処理を行なうようにしても差し支えない。

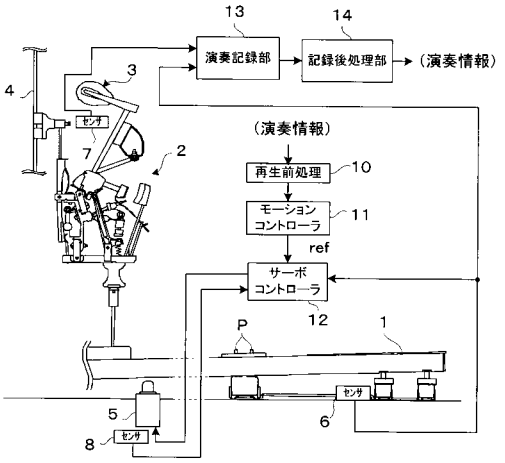
また、非リアルタイム処理に限らず、ある程度纏まった量のＭＩＤＩイベントをバッファしうるよう適宜のバッファ時間（例えば５００msec程度の遅延時間)を設けることで、該バッファ時間内でのリアルタイム処理は可能である。

【０１０１】

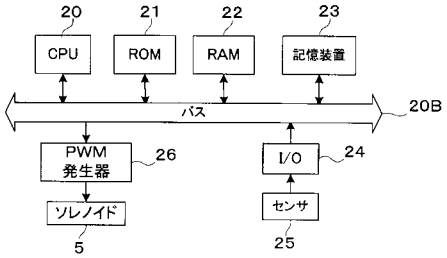
なお、上記実施例においては、図３に示す通り、演奏情報が、ノートオンイベントにおいて１２７段階で記述されたノートンベロシティ値と、ノートオフイベントにおいて１２８段階で記述されたノートオフベロシティ値とを個別に持つものとし、該ノートオンイベントに対応して押鍵起動が生成され、該ノートオフイベントに対応して離鍵起動が生成される例について説明したが、これに限らず、ノートンベロシティ値の特定の１つの値（典型的にはＭＩＤＩ値＝０）をノートオフイベントとして扱う場合も、この発明を適用可能である。この場合は、当該ノートオフイベントに対応するノートオンベロシティ値（ＭＩＤＩ値＝１～１２７）に基づきノートオフベロシティ値を推測して離鍵軌道を作成し、この発明に係る打鍵イベント群（軌道データ群）の区切り処理や打鍵イベント毎の修正処理等を行なうようにしてよい。

【０１０２】
　また、上記図１１に示すサーボコントローラ１２のサーボ制御構成は一例であって、サーボ制御の適用例は、フィードバック制御する物理量（位置、速度、加速度等）の種類及びその組み合わせ、或いは、各物理量の重み付けゲインの調整やバイアス電流の付加等、種々の要素によって多様である。すなわち、サーボコントローラ１２のサーボ制御構成は従来から知られるどのような構成を適用しても差し支えない。
【０１０３】
　なお、上記実施例においては、この実施例に係る自動演奏ピアノをアコースティックピアノによって構成する例について説明したが、これに限らず、自動演奏機能を具えた電子ピアノなど、鍵を自動的に動かす機構を備えた電子鍵盤楽器に対して、この発明を適用してもよい。
　また、この発明はハードウェア装置により構成されてもよいし、コンピュータにおいて本発明の動作を実行させるためのソフトウェアプログラム、或いは、ＤＳＰに本発明の動作を実行させるためのマイクロプログラムによって構成されてもよい。
【図面の簡単な説明】
【０１０４】
【図１】この発明の一実施例に係る自動演奏ピアノの全体構成を示す図。
【図２】同実施例に係る自動演奏ピアノの電気的ハードウェア構成を示すブロック図。
【図３】同実施例に係る自動演奏ピアノに供給される演奏情報の構成例を示す。
【図４】同実施例の第１実施例に係る自動演奏ピアノにおける再生動作の手順の一例を示すフローチャート。
【図５】同実施例に係る鍵毎の打鍵イベント抽出処理の手順の一例を示すフローチャート。
【図６】同実施例に係る鍵別打鍵イベントブロック記憶領域の構成例を示す図。
【図７】同実施例に係る打鍵イベント群区切り処理の手順の一例を示すフローチャート。
【図８】同実施例に係る打鍵イベント群の修正処理の手順の一例を示すフローチャート。
【図９】同実施例に係る打鍵イベント群の修正処理を模式的に示す図であって、（ａ）は修正前、（ｂ）は修正後をそれぞれ示す。また、（ｃ）は別の修正例を示す。
【図１０】同実施例に係る軌道データ作成処理の手順の一例を示すフローチャート。
【図１１】同実施例に係るサーボ制御構成例を示すブロック図。
【図１２】この発明の別の実施例（第２実施例）に係る軌道データ群の修正処理を模式的に示す図であって、（ａ）は修正前、（ｂ）は修正後をそれぞれ示す。また、（ｃ）は別の修正例を示す。
【図１３】同別の実施例に係る再生動作の手順の一例を示すフローチャート。
【図１４】同別の実施例に係る鍵別軌道データブロック記憶領域の構成例を示す図。
【図１５】同別の実施例に係る軌道データ群区切り処理の手順の一例を示すフローチャート。
【図１６】同別の実施例に係る軌道データ群の修正処理のの手順の一例を示すフローチャート。
【符号の説明】
【０１０５】
　１　鍵、２　アクション機構、３　ハンマ、４　弦、５　電磁ソレノイド、６　キーセンサ、７　ハンマセンサ、８　プランジャセンサ、１０　再生前処理部、１１　モーションコントローラ、１２　サーボコントローラ、１３　演奏記録部、１４　記録後処理部、２０　ＣＰＵ、２１　ＲＯＭ、２２　ＲＡＭ、２３　記憶装置

【図１】



【図２】



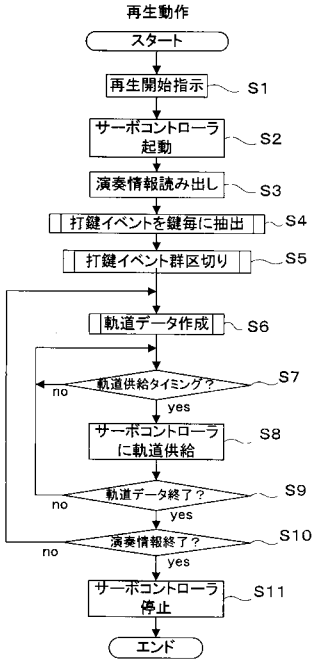
【図３】

演奏情報の構成

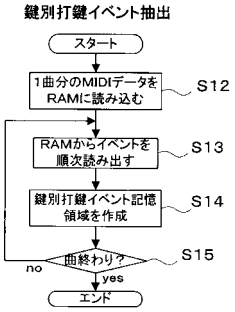
デルタタイム	[tt]
ノートオンイベント	[9n kk vv]
デルタタイム	[tt]
ノートオンイベント	[9n kk vv]
デルタタイム	[tt]
ノートオフイベント	[8n kk vv]
デルタタイム	[tt]
ノートオフイベント	[8n kk vv]
⋮	

ノートオンイベント
［9n kk vv］ n:チャンネル kk:ノート番号 vv:ベロシティ
ノートオフイベント
［8n kk vv］ n:チャンネル kk:ノート番号 vv:ベロシティ
デルタタイム
［tt］ tt:前後イベント間の経過時間

【図４】



【図５】



【図６】

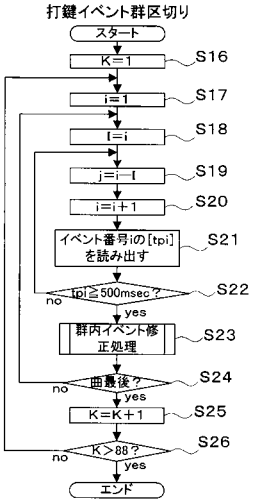
鍵別打鍵イベント記憶領域の構成

K＝1, i＝1～M			
tp1	vp1	tn1	vn1
tp2	vp2	tn2	vn2
⋮	⋮	⋮	⋮
tpi	vpi	tni	vni
⋮	⋮	⋮	⋮
tpM	vpM	tnM	vnM

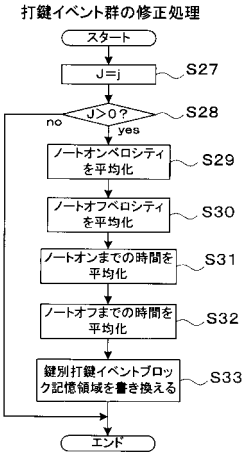
←1つの打鍵

以下K＝2～88まで、
内部構造は同様

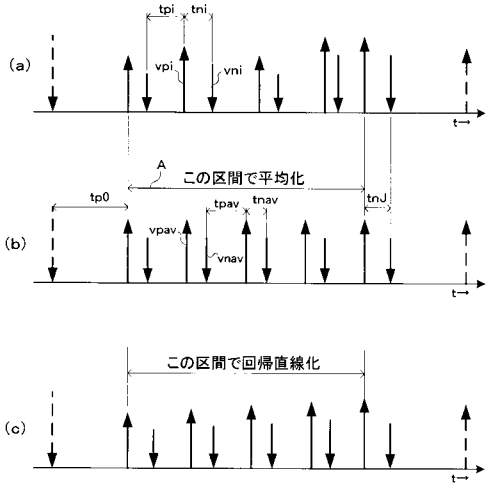
【図７】



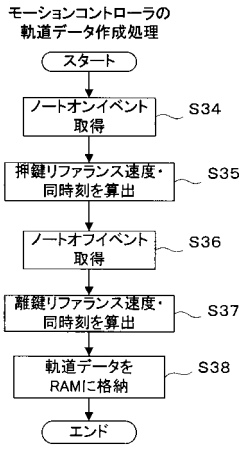
【図８】



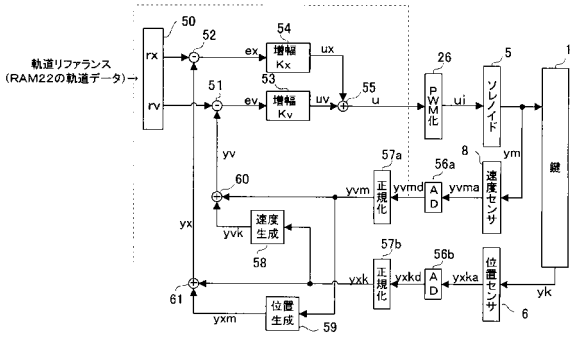
【図９】



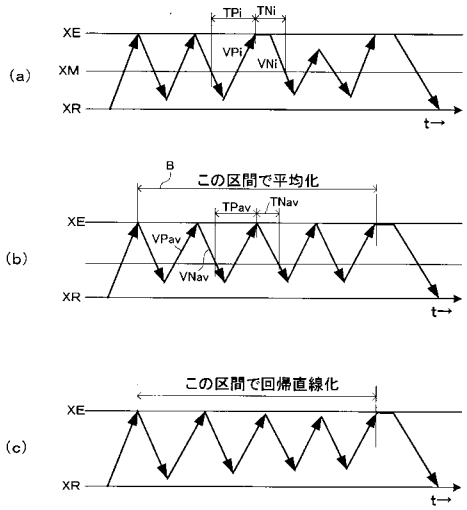
【図１０】



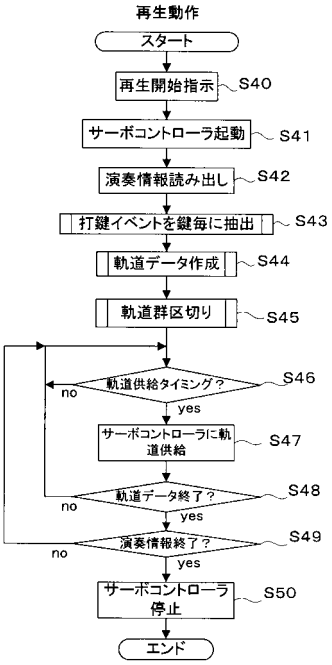
【図１１】



【図１２】



【図１３】



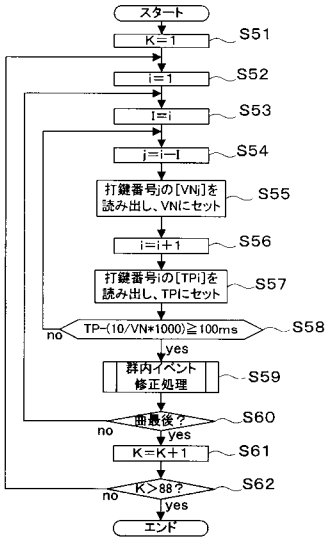
【図１４】

鍵別軌道データブロック記憶領域の構成

K＝1，i＝1～M				
TP1	VP1	TN1	VN1	←1つの軌道データ
TP2	VP2	TN2	VN2	
⋮	⋮	⋮	⋮	
TPi	VPi	TNi	VNi	
⋮	⋮	⋮	⋮	
TPM	VPM	TNM	VNM	

以下K＝2～88まで、
内部構造は同様

【図１５】



【図１６】