(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3620063号
(P3620063)
(45)発行日　平成17年2月16日(2005.2.16)　(24)登録日　平成16年11月26日(2004.11.26)

(51)Int.Cl.[7]　F I
G10F　1/02　G10F　1/02　B

請求項の数 4　(全 37 頁)

(21)出願番号　特願平6-79604
(22)出願日　平成6年3月25日(1994.3.25)
(65)公開番号　特開平7-271355
(43)公開日　平成7年10月20日(1995.10.20)
審査請求日　平成12年10月26日(2000.10.26)

(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
(74)代理人　100098084
弁理士　川▲崎▼　研二
(74)代理人　100096884
弁理士　末成　幹生
(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
(72)発明者　川端　太郎
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】自動ピアノおよび演奏データ加工装置

(57)【特許請求の範囲】
【請求項1】
所定の鍵が押鍵されたことを表す押鍵イベント、該鍵の駆動により発せられる音の発音時刻及び発音強度に関連したデータからなるハンマーイベント、離鍵時刻及び離鍵速度からなる離鍵イベントに係るイベントデータを複数含む演奏データを供給する演奏情報供給手段と、
該演奏情報供給手段によって演奏データが供給されたときに、対応する3つのイベントの順序関係に基づいて、欠落したイベントが存在するか否かを判定する判定手段と、
前記判定手段において欠落したイベントが存在すると判定されると、欠落したと判定されたイベントに対応するダミーのイベントデータを前記演奏データに適宜追加するイベント修復手段と
を具備することを特徴とする演奏データ加工装置。
【請求項2】
前記判定手段は、前記押鍵イベント、前記ハンマーイベント、前記離鍵イベントの順序で前記各イベントデータが順次供給されなかった場合に、欠落したイベントデータが存在すると判定する
ことを特徴とする請求項1記載の演奏データ加工装置。
【請求項3】
前記判定手段は、押鍵イベントと該押鍵イベントに後続して供給されたハンマーイベントとの時間間隔が所定時間以上である場合には、該押鍵イベントと該ハンマーイベントとの

間で他のハンマーイベントが欠落したものと判定するものであり、
前記イベント修復手段は、前記他のハンマーイベントに係るイベントデータを前記演奏データに追加するとともに、前記他のハンマーイベントにおける打弦速度を、打弦時に発音が行われない程度の速度に設定する
ことを特徴とする請求項１記載の演奏データ加工装置。
【請求項４】
所定の鍵について押鍵イベントと、該鍵の駆動により発せられる音の発音時刻及び発音強度に関連したデータからなるハンマーイベントと、離鍵時刻及び離鍵速度からなる離鍵イベントとを検出するイベント検出手段と、
前記イベント検出手段により検出された前記押鍵イベントと前記ハンマーイベントと前記離鍵イベントとを記憶する記憶手段と、
前記記憶手段の内容を読出し、前記押鍵イベントと前記ハンマーイベントと前記離鍵イベントのうちで欠落したイベントがあるか否かを各イベントの順序関係に基づいて判定し、欠落したイベントがあると判定した場合には、その欠落したイベントに対応するダミーのイベントを追加し、前記押鍵イベントと前記ハンマーイベントと前記離鍵イベントとから構成される演奏データとして出力するイベント修復手段と、
前記イベント修復手段によって、欠落したイベントが修復された演奏データに基づいて鍵軌道を生成し、この鍵軌道に沿って前記鍵を駆動する鍵駆動手段と
を具備することを特徴とする自動ピアノ。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】
この発明は、例えば、自動的に鍵盤を駆動して演奏を行う自動ピアノおよびこの自動ピアノに用いて好適な演奏データ加工装置に関する。
【０００２】
【従来の技術】
従来の自動ピアノにおいては、各鍵および各ハンマーに対してセンサが設けられており、これらセンサによって検出したオン／オフイベント、押鍵速度またはハンマー速度等に基づいて演奏データを記録し、この演奏データに基づいて押鍵／離鍵時刻および押鍵速度を制御することにより演奏状態を再生することが可能である。また、自動ピアノにおいて検出した演奏データを、他の自動ピアノあるいは電子楽器等に直接供給することにより、複数箇所でリアルタイムに生演奏を行うことも可能である。
【０００３】
また、これは公知技術ではないが、単に押鍵／離鍵時刻および押鍵速度等に基づいて各鍵を駆動したのでは演奏の微妙なニュアンスを再現することが困難であるため、リファレンス軌道（鍵軌道の目標値）を生成し連続的に各鍵の動きを駆動する技術が本出願人により提案されている（平成５年１２月１７日付の特許出願，整理番号９３７０１３）。
【０００４】
【発明が解決しようとする課題】
ところで、上述した自動ピアノは、押鍵イベント（押鍵時刻，押鍵速度）、ハンマーイベント（打弦時刻、打弦速度）等に基づいてリファレンス軌道を生成するため、なんらかの理由により何れかのイベントが抜けた場合は、リファレンス軌道を生成することが困難であった。
この発明は上述した事情に鑑みてなされたものであり、取得された限りの情報に基づいて正確な鍵の動作再生を行う自動ピアノを提供することを第１の目的としている。また、かかる鍵の動作再生をリアルタイムに実現するために、この自動ピアノに用いて好適な演奏データ加工装置を提供することを第２の目的としている。
【０００５】
【課題を解決するための手段】
上記課題を解決するため、請求項１に係る発明にあっては、所定の鍵が押鍵されたことを

表す押鍵イベント、該鍵の駆動により発せられる音の発音時刻及び発音強度に関連したデータからなるハンマーイベント、離鍵時刻及び離鍵速度からなる離鍵イベントに係るイベントデータを複数含む演奏データを供給する演奏情報供給手段と、該演奏情報供給手段によって演奏データが供給されたときに、対応する３つのイベントの順序関係に基づいて、欠落したイベントが存在するか否かを判定する判定手段と、前記判定手段において欠落したイベントが存在すると判定されると、欠落したと判定されたイベントに対応するダミーのイベントデータを前記演奏データに適宜追加するイベント修復手段とを具備することを特徴としている。

【０００８】

また、請求項２に係る構成にあっては、請求項１において、前記判定手段は、前記押鍵イベント、前記ハンマーイベント、前記離鍵イベントの順序で前記各イベントデータが順次供給されなかった場合に、欠落したイベントデータが存在すると判定することを特徴としている。

【０００９】

また、請求項３に係る構成にあっては、請求項１において、前記判定手段は、押鍵イベントと該押鍵イベントに後続して供給されたハンマーイベントとの時間間隔が所定時間以上である場合には、該押鍵イベントと該ハンマーイベントとの間で他のハンマーイベントが欠落したものと判定するものであり、前記イベント修復手段は、前記他のハンマーイベントに係るイベントデータを前記演奏データに追加するとともに、前記他のハンマーイベントにおける打弦速度を、打弦時に発音が行われない程度の速度に設定することを特徴としている。

【００１１】

また、請求項４に係る構成にあっては、所定の鍵について押鍵イベントと、該鍵の駆動により発せられる音の発音時刻及び発音強度に関連したデータからなるハンマーイベントと、離鍵時刻及び離鍵速度からなる離鍵イベントとを検出するイベント検出手段と、前記イベント検出手段により検出された前記押鍵イベントと前記ハンマーイベントと前記離鍵イベントとを記憶する記憶手段と、前記記憶手段の内容を読出し、前記押鍵イベントと前記ハンマーイベントと前記離鍵イベントのうちで欠落したイベントがあるか否かを各イベントの順序関係に基づいて判定し、欠落したイベントがあると判定した場合には、その欠落したイベントに対応するダミーのイベントを追加し、前記押鍵イベントと前記ハンマーイベントと前記離鍵イベントとから構成される演奏データとして出力するイベント修復手段と、前記イベント修復手段によって、欠落したイベントが修復された演奏データに基づいて鍵軌道を生成し、この鍵軌道に沿って前記鍵を駆動する鍵駆動手段とを具備することを特徴としている。

【００１２】

【作用】

請求項１に係る構成にあっては、判定手段は、複数のイベントデータ相互の関係に基づいて、欠落したイベントデータが存在するか否かを判定する。そして、欠落したイベントデータが存在すると判定されると、イベント修復手段は、欠落したと判定されたイベントに対応するダミーのイベントデータを演奏データに適宜追加する。

【００１５】

また、請求項２に係る構成にあっては、押鍵イベント、ハンマーイベント、離鍵イベントの順序で各イベントデータが順次供給されなかった場合に、判定手段は、欠落したイベントデータが存在すると判定するから、各イベントの順序に基づいて抜けイベントの有無を判定することができる。

【００１６】

また、請求項３に係る構成にあっては、判定手段は、押鍵イベントと該押鍵イベントに後続して供給されたハンマーイベントとの時間間隔が所定時間以上である場合には、該押鍵イベントと該ハンマーイベントとの間で他のハンマーイベントが欠落したものと判定し、イベント修復手段は該他のハンマーイベントに係るイベントデータを演奏データに追加す

るとともに、該他のハンマーイベントにおける打弦速度を、打弦時に発音が行われない程度の速度に設定する。これにより、ハンマーイベントによる発音を抑制しつつダンパ操作を行うことができる。

【００１７】

また、請求項４に係る構成にあっては、イベント検出手段が所定の鍵について押鍵イベントと、該鍵の駆動により発せられる音の発音時刻及び発音強度に関連したデータからなるハンマーイベントと、離鍵時刻及び離鍵速度からなる離鍵イベントとを検出すると、これらイベントの内容が記憶手段に記憶される。イベント修復手段はこれら各イベントを読出し、欠落したイベントがあるか否かを各イベント相互の関係に基づいて判定し、欠落したイベントがあると判定した場合には、その欠落したイベントに対応するダミーのイベントを追加し、前記押鍵イベントと前記ハンマーイベントと前記離鍵イベントとから構成される演奏データとして出力する。鍵駆動手段は、この演奏データに基づいて鍵を駆動するから、イベントの欠落があった場合であっても鍵を駆動することが可能になる。

【００１８】

【実施例】

Ｉ．第１実施例

Ａ．実施例の制御原理

▲１▼リファレンスポイント

図２０は、一般的な自動ピアノの要部の構成を示す断面図である。図に示すように、自動ピアノにおいては、鍵１と、鍵１の運動をハンマー２に伝達するアクション３と、ハンマー２によって打弦される弦４と、鍵１を駆動するソレノイド５とを有している。そして、ソレノイド５のプランジャが突出すると、鍵１がバランスピンＰを中心に回動し、その演奏者側が下がり（以下、この状態を押鍵状態という）、また、これに連動してアクション３が作動し、ダンパー６が弦４から離れるとともに、ハンマー２が回動して打弦する。

一方、演奏者が弾く場合は、指で鍵１を押下することにより、上述と同様の作用が生じて打弦が行われる。

【００１９】

なお、図において、ＳＥ１，ＳＥ２は、打弦速度を計測するためのセンサであり、ハンマー２がこれらのセンサＳＥ１，ＳＥ２の間を通過する時間を計測することにより、ハンマー２の速度、すなわち、打弦速度が計測される。また、ハンマー２がセンサＳＥ１を通過する時刻が打弦時刻として計測される。

ところで、鍵１を押し下げる速度に応じてハンマー２の打弦速度が決まるが、鍵１の速度は初め遅くて次第に早くなる場合や、その逆の場合もあり、さらには、ほとんど一定の速さで押される場合もある。この場合、鍵１のレスト位置（鍵１を押してない場合の初期位置）からエンド位置（鍵を押し切った位置）に至るまでの速度と、ハンマー２の打弦速度とがどのような関係になっているのかが重要である。なぜならば、その関係を考察せず、打弦強度データに応じて鍵速度（初期速度など）を制御しても、記録時の打弦速度を再生することはできないからである。

【００２０】

実験によれば、鍵１のある位置における速度とハンマー２の打弦速度とが極めて良い対応を示すことが判った。この位置は、ピアノの個体差にもよるが、概ねレスト位置から９．０ｍｍ～９．５ｍｍ程度押し下げた位置であった。したがって、鍵１がこの位置に達するときの速度を、打弦強度データに応じて制御すれば、記録時の打弦速度を忠実に再現することができる。なお、以下においては、上述の所定位置をリファレンスポイントＸｒという。

【００２１】

▲２▼リファレンス速度

次に、上述のようにして求めたリファレンスポイントＸｒにおいて、どのような鍵速度にすれば、打弦速度を忠実に再現することができるかを設定する必要がある。なお、以下においては、リファレンスポイントＸｒにおける鍵速度をリファレンス速度Ｖｒという。

ここで、図２はリファレンスポイントＸｒを９．５ｍｍに設定したときの鍵速度と打弦速度の関係を示す図である。図中、白点は鍵をエンド位置まで押し切る単打奏法を行った場合の結果を示し、黒点は鍵をエンド位置まで押し切らずに連打する連打奏法を行った場合の結果を示している。また、Ｃ１は１次最小自乗法近似による直線、Ｃ２は６次最小自乗法による曲線を示している。

【００２２】

図２から明らかなように、リファレンス速度Ｖｒは、直線Ｃ１あるいは曲線Ｃ２のいずれによっても近似できる。したがって、近似性のよい関数を適宜選択すれば、この関数を用いて任意の打弦強度データ（記録時の打弦速度情報）からリファレンス速度Ｖｒを決定することができる。

この実施例においては、計算が簡単で誤差の少ない１次関数近似を採用している。したがって、リファレンス速度Ｖｒは、次式によって求められる。

【００２３】

【数１】

$$Vr = \alpha \cdot VH + \beta$$

数１において、ＶＨは打弦速度（打弦強度データ）であり、αおよびβは定数である。定数αおよびβは、ピアノの機種等に応じ実験等によって決定する。なお、αおよびβは、同一ピアノであっても、リファレンスポイントＸｒをどこにするかによって変動する。

【００２４】

▲３▼リファレンス時間差

さて、演奏データに含まれる打弦時刻データは、前述したように、相対時刻あるいは絶対時刻で記録されているが、いずれにしても再生側自動ピアノにおいて打弦時刻データを読みとって処理することにより、再生時の各音の打弦絶対時刻が求められる。そこで、このようにして求めた打弦絶対時刻において正確に打弦を行わせるには、鍵が何時リファレンスポイントＸｒを通過すればよいかを求める必要がある。

【００２５】

ここで、鍵１がリファレンスポイントＸｒを通過する時刻（以下、リファレンス時刻ｔｒという）と打弦時刻（正確には、ハンマーが打弦位置直前にあるセンサＳＥ１を通過した時刻）との時間差をリファレンス時間差Ｔｒと定義し、これと打弦速度との関係を実験により求めたものが図３である。図３において、白点は単打奏法による結果、黒点は連打奏法による結果を示している。そして、図３を縮尺２倍にしたものが図４であり、縮尺４倍にしたものが図５である。これらの図から判るように、リファレンス時間差Ｔｒと打弦速度との関係は、双曲線により極めて良好に近似される。すなわち、このリファレンス時間差Ｔｒは、打弦速度ＶＨを分母にする１変数式で近似することができ、次式によって算出される

。

【００２６】

【数２】

$$Tr = -(\gamma / vH) + \delta$$

なお、数２における定数γおよびδは、ピアノの機種等に応じ実験等によって決定する。なお、γおよびδは、同一ピアノであっても、リファレンスポイントＸｒをどこにするかによって変動する。これは、数１におけるα、βの場合と同様である。

【００２７】

さて、数２によって、リファレンス時間差Ｔｒが求まれば、再生側の打弦絶対時刻からリファレンス時間差Ｔｒを減算することによって、リファレンス時刻

ｔｒが求められ、結局、上述した▲１▼、▲２▼、▲３▼の処理により、リファレンスポイントＸｒ、リファレンス速度Ｖｒ、およびリファレンス時刻ｔｒが求められる。したがって、リファレンス時刻ｔｒにリファレンスポイントＸｒに達し、かつ、その時の速度がリファレンス速度Ｖｒとなるように鍵１を駆動すれば、記録時の打弦状態を忠実に再現することができることが判る。

なお、鍵１がリファレンスポイントＸｒに達したときに打弦が行われるのであれば、リファレンス時間差Ｔｒを求める処理は不要になる。

【００２８】

以上のようにして、予め設定したリファレンスポイントＸｒにおけるリファレンス速度Ｖｒが求められれば、鍵１の挙動をこれに応じて制御することにより、記録時の打弦速度を再生することができる。

この場合、レスト位置からエンド位置までの鍵の軌道（時間経過に対する鍵の位置）を設定すれば、鍵の押下開始位置からリファレンスポイントＸｒに至るまでの各位置と時刻の関係（速度と時刻の関係等）が求められるから、これに応じて鍵の位置をフィードバック制御すればよいことが判る。この場合における鍵の軌道は、例えば、リファレンスポイントＸｒを通る直線軌道（等速運動の場合）や放物線軌道（等加速度運動の場合）の他、任意の軌道を設定することができる。ただし、打弦状態の再現性が良く、また、制御し易い軌道を選択することが好ましい。また、設定する軌道によっては、リファレンス点における加速度等（運動を決定する他の要素）を求める必要が生じるが、これについては軌道の種類に応じて適宜算出すればよい。

【００２９】

▲４▼ダミーイベントの追加

ところで、上述したような再生処理を実現するためには、演奏記録時において、押鍵時刻、打弦時刻および打弦速度を得ておく必要がある。しかし、演奏状態あるいは自動ピアノの性能等により、押鍵イベントあるいはハンマーイベント等が抜ける虞がある。その詳細を図２１，図２２を参照し説明する。まず、図２１（ａ）は、演奏記録時における鍵１の押鍵深さを示している。図において一点鎖線$Ｌ_{1}$は、センサによって押鍵または離鍵イベントが検出される押鍵深さである。また、同図（ｂ）は、押鍵深さの軌道が一点鎖線$Ｌ_{1}$と交差するタイミング、すなわち押鍵または離鍵イベントが発生するタイミングを示している。また、同図（ｃ）は、センサＳＥ１，ＳＥ２によりハンマーイベントの検出された時刻（正確には、ハンマーが打弦位置直前にあるセンサＳＥ１を通過した時刻）と、検出されたハンマー速度$Ｖｈ_{1}$，$Ｖｈ_{2}$を示す。

【００３０】

図２１（ａ）によれば、時刻$ｔ_{0}$から鍵１の押鍵が開始され、時刻$ｔ_{1}$において押鍵イベントが検出されている。また、時刻$ｔ_{15}$においては、この押鍵動作に対応するハンマーイベントが生じている。次に、時刻$ｔ_{2}$においては、鍵１が若干離鍵されているが、離鍵イベントが検出される前に再び押鍵が開始され、この押鍵動作に対応するハンマーイベントが時刻$ｔ_{25}$に生じている。すなわち、ハーフストロークによる押鍵が行われている。そして、その後に鍵１が離鍵され、時刻$ｔ_{3}$において離鍵イベントが検出されている。すなわち、図２１（ａ）～（ｃ）によれば、時刻$ｔ_{15}$のハンマーイベントに対応する離鍵イベントと、時刻$ｔ_{25}$のハンマーイベントに対応する押鍵イベントは存在しない。このような状態を、以下「イベント抜け」という。

【００３１】

また、図２２（ａ）～（ｃ）においても、ハーフストローク時における押鍵深さ、鍵イベントおよびハンマーイベントが示されている。まず時刻$ｔ_{1}$においては押鍵イベントが検出され、これに対応するハンマーイベントおよび離鍵イベントが時刻$ｔ_{15}$および時刻$ｔ_{2}$に各々検出されている。次に、時刻$ｔ_{3}$においては、ハーフストロークによる押鍵イベントが検出されているが、この押鍵は、ダンパー操作を行うための弱い押鍵であるため、ハンマー２は打弦を行う前にレスト位置に戻っている。従って、同図（ｃ）を参照すると、時刻$ｔ_{3}$の押鍵イベントに対応するハンマーイベントは存在しない。すなわち、ハンマーイベントのイベント抜けが生じている。

【００３２】

本実施例においては、以上のようなイベント抜けが検出された場合には、仮想のイベント（ダミーイベント）が適宜追加される。すなわち、第１のハンマーイベントと第２のハンマーイベントとの間において、離鍵イベントおよび押鍵イベントが抜けている場合には、

第１のハンマーイベントの発生時刻において、ダミーの離鍵イベントとダミーの押鍵イベントとが追加される。例えば、図２１（ｄ）においては、ハンマーイベントの発生する時刻t_{15}においてダミーの離鍵イベントとダミーの押鍵イベントとが追加されている。一方、押鍵イベントと離鍵イベントとの間においてハンマーイベントが抜けている場合は、当該押鍵イベントの発生時刻において弱いハンマーイベントが追加される。ここで、「弱いハンマーイベント」とは、発音が行われない程度に、きわめてハンマー速度が遅いイベントをいう。例えば、図２２（ｄ）においては、押鍵イベントが発生する時刻t_3において、ダミーの弱いハンマーイベントが追加されている。ダミーイベントの追加は、押鍵イベント、ハンマーイベント、離鍵イベントの順序、あるいはイベントの時間間隔などに応じて行われる。その詳細については後述する。

【００３３】

以上のように追加されるダミーイベントの発生時刻は既存のイベントと同一の時刻であるが、これら発生時刻は適切なリファレンス軌道を生成するために、適宜変更される。その詳細については後述する。

また、これらダミーイベントを追加する理由を説明しておく。まず、本実施例においては、上述したように、演奏記録時において、押鍵時刻、打弦時刻および打弦速度を得ておく必要がある。従って、これらのうち一部が抜けた場合には、正確なリファレンス軌道を生成することが困難になる。例えば、本実施例において、各種のダミーイベントを追加することにより、図２１（ｅ）、図２２（ｅ）、図２３（ｆ）の実線に示すようなリファレンス軌道を得ることができる（詳細は後述する）。これに対して、ダミーイベントを追加しなかった場合には、例えば同図の破線L_2～L_4に示すような、実際の演奏状態と大幅に異なるリファレンス軌道が生成されることになる。すなわち、本実施例においては、ダミーイベントを適宜追加することにより、実際の演奏状態に近いリファレンス軌道が得られるのである。

【００３４】

▲５▼処理の概要

以上のようにして、予め設定したリファレンスポイントＸｒにおけるリファレンス速度Ｖｒが求められれば、鍵１の挙動をこれに応じて制御することにより、記録時の打弦速度を再生することができる。また、イベント抜けが検出された場合には適宜ダミーイベントが追加されるから、ハーフストローク等の奏法が採られた場合においても、安定して演奏状態を再生することが可能である。

【００３５】

この場合、レスト位置からエンド位置までの鍵の軌道（時間経過に対する鍵の位置）を設定すれば、鍵の押下開始位置からリファレンスポイントＸｒに至るまでの各位置と時刻の関係（速度と時刻の関係等）が求められるから、これに応じて鍵の位置をフィードバック制御すればよいことが判る。この場合における鍵の軌道は、例えば、リファレンスポイントＸｒを通る直線軌道（等速運動の場合）や放物線軌道（等加速度運動の場合）の他、任意の軌道を設定することができる。ただし、打弦状態の再現性が良く、また、制御し易い軌道を選択することが好ましい。また、設定する軌道によっては、リファレンス点における加速度等（運動を決定する他の要素）を求める必要が生じるが、これについては軌道の種類に応じて適宜算出すればよい。

以上がこの実施例の制御原理である。

【００３６】

Ｂ．実施例の構成

Ｂ－１．全体構成

次に、本実施例の全体構成を説明する。図１は、この実施例の構成を示すブロック図である。なお、前述した図２０の各部と対応する部分には同一の符号を付け、その説明を省略する。

【００３７】

図において、１０は再生前処理部であり、記録メディアあるいはリアルタイム通信装置か

ら供給される演奏データに基づいて、鍵の軌道データを作成する回路であり、まず、リファレンス時刻ｔｒ、リファレンス速度ｖｒを求め、さらに、加速度の推定が必要な場合はこれを演算する。そして、軌道の種類（直線、放物線、その他任意の軌道）に応じて各音に対する鍵軌道を生成する。

１１は、再生前処理部１０が作成した軌道データに基づいて、ソレノイド５のプランジャの動きを示すデータを作成するモーションコントローラである。また、１２はサーボコントローラであり、モーションコントローラ１１から供給されるデータに応じてソレノイド５の励磁電流を制御する。この場合、ソレノイド５には、プランジャの位置を検出する検出機構が設けられており、これにより検出されたプランジャ位置をフィードバック信号としてサーボコントローラ１２にフィードバックするようになっている。そして、サーボコントローラ１２は、モーションコントローラ１１から供給されるデータとソレノイド５からのフィードバック信号とを照合しながら、両者が一致するように励磁電流を制御する。

【００３８】

次に、２６は、鍵１の下面に取り付けられた板状のシャッタである。２５は、上下方向に所定距離隔て設けられる２組のフォトセンサによって構成されているキーセンサであり、鍵１が押下され始めると、まず上方のフォトセンサが遮光され、次いで、下方のフォトセンサが遮光される。離鍵の際には、下方のフォトセンサが受光状態になり、ついで、上方のセンサが受光状態になる。

【００３９】

キーセンサ２５の出力信号は、演奏記録部３０に供給され、演奏記録部３０は、キーセンサ２５内の下方のフォトセンサが受光状態になってから上方のフォトセンサが受光状態になるまでの時間を測定し、ここから、離鍵速度を検出する。また、演奏記録部３０は上方のフォトセンサが受光状態になった時刻を離鍵時刻として処理する。また、演奏記録部３０は、センサＳＥ２が遮光されてからセンサＳＥ１が遮光されるまでの時間を計測し、ここから打弦速度を検出する。また、センサＳＥ１が遮光された時刻を打弦時刻として処理する。

【００４０】

すなわち、演奏記録部３０は、演奏が開始されると、センサＳＥ１，ＳＥ２の出力信号に基づいて、打弦時刻および打弦速度を検出し、かつ、キーセンサ２５の出力信号に基づいて離鍵時刻および離鍵速度を検出する。以上のようにして検出された各情報は、記録後処理部３１に供給される。

【００４１】

記録後処理部３１においては、演奏記録部３０から供給される各種情報に対し、正規化処理を施した後に、外部の記録媒体に演奏データとして供給する。ここで、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。すなわち、打弦速度、打弦時刻、離鍵速度、離鍵時刻等は、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固有の傾向を持つため、標準となるピアノを想定し、そのピアノにおける打弦速度、打弦時刻等に変換するための処理である。また、記録後処理部３１は、演奏記録部３０で記録された情報にイベント抜けを検出すると、演奏データに適宜ダミーイベントを追加する（詳細は後述する）。

【００４２】

上述した構成において自動ピアノを演奏すると、演奏記録部３０においてハンマーイベントおよび鍵イベント等が記録され、記録後処理部３１においてこれらの情報に正規化処理等が施された後、演奏データが出力される。この演奏データは、他の自動楽器に供給され、あるいは適当な記録媒体に記録される。次に、記録された演奏データを再生すると、演奏データ中の打弦強度データに応じて、リファレンスポイントＸｒを通過する鍵軌道が生成され、鍵がこの軌道を再現するようにソレノイド５の励磁電流が制御される。

この結果、鍵１がリファレンスポイントＸｒに達したときの速度は、記録時の打弦速度に対応した速度（すなわち、リファレンス速度Ｖｒ）になり、ハンマー２は記録時と同じ速度で打弦を行う。

【００４３】
以上のように、この実施例においては、リファレンスポイントという制御点を導入し、この点における鍵の運動属性を再現するとともに、イベント抜けが生じた場合は演奏データに適宜ダミーイベントを追加するため、センサ等を増やすことなく、忠実に打弦速度を再現することができる。
【００４４】
Ｃ．実施例の動作
Ｃ－１．演奏記録動作
演奏記録を行う場合、演奏者は、まず演奏記録部３０を動作状態にしておき、自動ピアノを演奏する。これにより、演奏記録部３０においてハンマーイベントおよび鍵イベント等が記録される。
次に、演奏者が記録後処理部３１を動作させると、図６に示すプログラムが実行される。図において処理が開始されると、ステップＳＰ１００において各種の変数の初期設定が行われる。次に処理がステップＳＰ１０１に進むと、演奏記録部３０からハンマーイベントおよび鍵イベント等のデータが読出される。また、ステップＳＰ１０１においては、読出されたデータに対して対応関係にあるデータも併せて読出される。例えば、最初に読出されたデータが所定のキーコードＫＣの押鍵イベント（押鍵時刻）であった場合、そのキーコードＫＣの離鍵イベント（離鍵時刻）およびハンマーイベント（打弦時刻、打弦速度）が検索され、存在するならばこれらのデータも演奏記録部３０から読出される。
【００４５】
▲１▼一般的な演奏状態に対する動作
次に、処理がステップＳＰ１０２を介してステップＳＰ１０３に進むと、先に読出されたデータの対応関係がどのような状態であるか判定される。一般的な
（正常な）演奏状態では、押鍵イベントと離鍵イベントとの間に１回のハンマーイベントが存在する。かかる場合は処理がステップＳＰ１０４に進む。ステップＳＰ１０４においては、押鍵時刻が打弦時刻以下であるか否かが判定される。一般的な状態では押鍵イベントが生じた後にハンマーイベントが生じるから「ＹＥＳ」と判定され、処理がステップＳＰ１０５に進む。ステップＳＰ１０５においては、押鍵時刻および打弦時刻の時間差が所定値Ａ以上であるか否かが判定される。
【００４６】
ここで、所定値Ａは一般的な押鍵状態における押鍵時刻，打弦時刻間の時間よりも若干余裕を持った値、例えば「１０ｍｓｅｃ」に設定されている。従って、一般的な演奏状態では「ＮＯ」と判定され、処理がステップＳＰ１１３に進む。ステップＳＰ１１３においては、先にステップＳＰ１０１で読出されたデータに上述した正規化処理が施される。すなわち、ステップＳＰ１０１で読出された打弦速度、打弦時刻、離鍵速度、離鍵時刻等のデータが、標準となるピアノにおける打弦速度、打弦時刻等に変換されるのである。そして、このデータは、ステップＳＰ１１４において、演奏データとして外部に出力される。出力された演奏データは磁気ディスク等の記録媒体に適宜記録される。
【００４７】
以上の処理が終了すると、処理がステップＳＰ１０１に戻り、演奏記録部３０から次のデータが読出され、上述したのと同様の処理が行われる。そして、演奏記録部３０に記録されたデータの全てが既に読み出され、ステップＳＰ１０１において新たなデータを読み出せなかった場合は、ステップＳＰ１０２で「ＹＥＳ」と判定され、記録後処理部３１の動作が終了する。
このように、一般的な演奏状態においては、演奏記録部３０から読出されたデータに対して単に正規化処理が施され、その結果が演奏データとして出力されることが判る。
【００４８】
▲２▼遅い押鍵に対する動作
演奏記録時において何れかの鍵がゆっくりと押下されると、押鍵時刻から打弦時刻までの時間が所定値Ａよりも長くなる場合がある。かかる演奏状態が演奏記録部３０に記録され

た場合は、ステップＳＰ１０５において「ＹＥＳ」と判定され、処理はステップＳＰ１０６に進む。ステップＳＰ１０６においては、所定値Ｂが押鍵速度で除算され、その商は押鍵時刻，打弦時刻の時間差以上であるか否かが判定される。ここで「ＮＯ」と判定されると、処理はステップＳＰ１１３に進む。すなわち、押鍵速度が遅い場合には、時間差が所定値Ａを越えている場合であっても、この時間差が押鍵速度に鑑みて妥当な値であれば、一般的な場合と同様に処理をステップＳＰ１１３に進めることとしたものである。ここで、所定値Ｂは、例えば「１０ｍｍ」（所定値Ｂのディメンジョンはｓ・ｍｍ／ｓ＝ｍｍになる）程度に設定するとよい。

【００４９】

▲３▼離鍵／押鍵イベント抜けに対する動作

先に図２１について説明したように、ハーフストロークの連打を行った場合等においては、複数のハンマーイベントの間に離鍵／押鍵イベントが存在しない場合がある。かかる状態のデータが読出され処理がステップＳＰ１０３に進むと、離鍵／押鍵イベント抜けであると判定され、処理がステップＳＰ１１１に進む。ステップＳＰ１１１においては、最後のハンマーイベントを除く全てのハンマーイベントに対応して、各打弦時刻と同一時刻のダミーの離鍵および押鍵イベントが追加される。

【００５０】

次に、処理がステップＳＰ１１２に進むと、追加されたダミーの離鍵および押鍵イベントの内容が修正される。その内容を、以下詳述する。

まず、離鍵／押鍵イベント抜けが生じた場合においても、ハンマーイベントの内容、すなわち打弦時刻および打弦速度は既知である。また、先に図２～図５において説明したように、打弦速度が既知であればリファレンス速度Ｖｒおよびリファレンス時間差Ｔｒを求めることができる。従って、「打弦時刻よりもリファレンス時間差Ｔｒだけ前の時刻にリファレンス点を通過し、かつ、リファレンス速度Ｖｒの傾きを有する直線」は一意に決定される。

【００５１】

本実施例においては、押鍵時の鍵軌道は上記直線に沿った等速打鍵のものと仮定し、これによってダミーの押鍵イベントの内容が修正される。すなわち、ダミーの押鍵イベントについて、押鍵時刻は上記直線がセンサ位置と交差する時刻に設定され、押鍵速度はリファレンス速度Ｖｒに等しくなるように（すなわち等速打鍵になるように）設定される。

【００５２】

また、ダミーの離鍵速度は、当該自動ピアノの機構上実現できる最大の離鍵速度、すなわち演奏者が鍵に全く触れていない場合の離鍵速度に設定される。この場合、ハンマーイベントが生じた後に直ちに離鍵が開始されたものと仮定する。従って、離鍵時刻は、ハンマーイベントを起点とする最大傾斜の離鍵軌道がセンサ位置と交差する時刻に設定される。このように、ダミーの離鍵および押鍵イベントが追加され、これらの内容が修正されると、処理がステップＳＰ１１３に進み、一般的な場合と同様の処理が行われる。

【００５３】

▲４▼ハンマーイベント抜けに対する動作

先に図２２の時刻t_3～t_4について説明したように、ハーフストローク奏法を行った場合は、押鍵イベントおよび離鍵イベントは生ずるが、これらに対応するハンマーイベントは発生しないことがある。かかる状態のデータが読出され処理がステップＳＰ１０３に進むと、ハンマーイベント抜けであると判定され、処理がステップＳＰ１０７に進む。ステップＳＰ１０７においては、押鍵時刻と同一時刻にダミーのハンマーイベントが追加される。

【００５４】

次に、処理がステップＳＰ１０８に進むと、ダミーのハンマーイベントに係る打弦速度が微少値、すなわち発音されない程度の値に設定される。また、かかる微少な打弦速度に対応してリファレンス速度Ｖｒおよびリファレンス時間差Ｔｒが求まるから、上記ステップＳＰ１１２と同様に、「打弦時刻よりもリファレンス時間差Ｔｒだけ前の時刻にリファレ

ンス点を通過し、かつ、リファレンス速度Ｖｒの傾きを有する直線」が一意に決定され、その内容に基づいて押鍵イベントの内容が修正される。
このように、弱いハンマーイベントが追加されるとともに押鍵イベントの内容が修正されると、処理がステップＳＰ１１３に進み、一般的な場合と同様の処理が行われる。
【００５５】
▲５▼押鍵時刻，打弦時刻の時間差が大き過ぎる場合の処理
押鍵時刻，打弦時刻の時間差が大きすぎる場合の具体例を、図２３（ａ）～（ｃ）に示す。これらの図において、鍵軌道は時刻t_1にセンサ位置に達しており、この時刻t_1に押鍵イベントが発生している。しかし、この時の押鍵動作はダンパー操作を行うためのハーフストロークであり、ハンマーイベントは発生していない。その後、センサ位置近くまで鍵が戻されるが、センサ位置に達する前の時刻t_2において再び鍵が押下され、時刻t_3にハンマーイベントが発生し、時刻t_4に離鍵イベントが発生している。
【００５６】
上記例では、押鍵イベントと離鍵イベントの間に１回のハンマーイベントが存在する。従って、ステップＳＰ１０３においては、各イベント間の関係は一応正常なものと判定され、処理はステップＳＰ１０４に進む。しかし、上記例では押鍵時刻と打弦時刻の時間差が大きすぎるため、ステップＳＰ１０５，１０６において何れも「ＹＥＳ」と判定され、処理がステップＳＰ１０７に進む。ステップＳＰ１０７においては、上記“ハンマーイベント抜け”の場合と同様に、ダミーの弱いハンマーイベントが追加される（同図（ｄ）参照）。しかし、ダミーのハンマーイベントを追加したために、押鍵イベントと離鍵イベントの間に２回のハンマーイベントが生ずることになる。これは、“離鍵／押鍵イベント抜け”の場合と同様の状態であるため、ダミーのハンマーイベントと同一時刻に、ダミーの離鍵および押鍵イベントがさらに追加される（同図（ｅ）参照）。
【００５７】
次に、処理がステップＳＰ１０８に進むと、ダミーのハンマーイベントに係る打弦速度が微少値に設定され、これに伴って時刻t_1に発生した押鍵イベントの内容も修正される。そして、以上の処理が終了すると、処理がステップＳＰ１１３に進み、一般的な場合と同様の処理が行われる。また、ダミーの離鍵および押鍵イベントは、前述した離鍵／押鍵イベント抜けに対する動作と同様にその内容が修正される。
【００５８】
▲６▼打弦時刻が押鍵時刻よりも早い場合の処理
一般的な演奏状態では、最初に押鍵イベントが生じた後、ハンマーイベントが発生する。従って、打弦時刻が押鍵時刻よりも早くなることは、ほとんど起り得ないと考えられる。しかし、例えば同一の鍵を高速に連打した場合などにおいて、ハンマー２やアクション３の運動状態によっては、かかる事態が生じる可能性も否定できない。
【００５９】
かかる場合は、ステップＳＰ１０４において「ＮＯ」と判定され、処理がステップＳＰ１０９に進む。ステップＳＰ１０９においては、打弦時刻と押鍵時刻の時間差が所定値Ｃ（例えば１０ｍｓｅｃ）以上であるか否かが判定される。ここで「ＹＥＳ」と判定されると、ステップＳＰ１１０において所定値Ｄ（例えば１０ｍｍ）が打弦速度で除算され、その商は上記時間差以上であるか否かが判定される。
【００６０】
ここで、打弦時刻と押鍵時刻との時間差が大きい場合には、ステップＳＰ１０９，１１０の何れにおいても「ＹＥＳ」と判定されるのであるが、かかる場合にはハンマーイベントは押鍵イベントから独立したものと推定することができる。すなわち、同一鍵の高速連打等を行った場合はハンマー２やアクション３が特殊な運動状態になるため、レスト位置からセンサ位置直前までの短い距離の押鍵動作（この押鍵動作はキーセンサ２５によっては検出されない）によってもハンマーイベントが生じることも考えられる。かかる打弦動作が行われた後、ダンパー操作のみを行う弱い押鍵動作（この操作によって押鍵／離鍵イベントは生じるが、ハンマーイベントは生じない）が行われると、結果的にハンマーイベン

ト、押鍵イベント、および離鍵イベントの順で各イベントが発生する。

【００６１】

そこで、かかる場合は、処理をステップＳＰ１１１に進め、ハンマーイベントと同時刻にダミーの押鍵／離鍵イベントを追加するとともに、その後に生じる押鍵イベントと離鍵イベントとの間にダミーの弱いハンマーイベントを追加することとしている。

【００６２】

一方、打弦時刻と押鍵時刻との時間差が小さい場合には、ステップＳＰ１０９，１１０の何れかにおいて「ＮＯ」と判定されるのであるが、かかる場合には押鍵イベントとハンマーイベントとがなんらかの関係を有していると推定されるため、各々独立にダミーイベントを追加したのでは、かえって不都合な結果を招くことも考えられる。そこで、かかる場合は、処理をステップＳＰ１１３に進め、ステップＳＰ１０１において読出されたデータに対して、ダミーイベントを追加することなく正規化処理を施すこととしている。なお、ハンマーイベントの後に押鍵イベントが発生するような状態では、後にリファレンス軌道を生成する際に不都合が生じるため、押鍵時刻が打弦時刻の前になるように押鍵イベントの内容を変更しておくと良い。そして、処理がステップＳＰ１１２に進むと、押鍵イベントおよび離鍵イベントの内容が修正される。

【００６３】

なお、上述した各処理においては、ダミーのイベントを適宜追加するために、現実にはあり得ない順番で鍵イベントが配列されることがある。例えば、図２２（ｅ）に示す例にあっては、時刻t_{21}に押鍵イベントが発生した後、再び時刻t_{22}に押鍵イベントが発生し、時刻t_{23}，t_{24}においても連続して離鍵イベントが発生している。従って、どの押鍵イベントがどの離鍵イベントと対応するのかを明確にするために、演奏データを外部に出力する際に識別情報を付与しておくとよい。また、同図に示すように、複数の押鍵／離鍵イベントに基づいて生成されるリファレンス軌道は相互に交差することもあるが、その対策については後述する。

【００６４】

Ｃ－２．演奏再生動作

さて、上述のように記録後処理部３１を介して出力された演奏データは磁気ディスク等の記録媒体に適宜記録されるが、この演奏データに基づいて自動演奏を行う場合には、演奏データを再生前処理部１０に供給する。以下、本実施例における再生前処理部１０について説明する。この実施例においては、鍵軌道として直線を想定するので、次のような手法で軌道データの作成を行う。

【００６５】

▲１▼押鍵時の軌道データ作成

図７は、鍵の押鍵軌道（直線軌道）を示す図であり、レスト位置X_0から等速運動をしてエンド位置Ｘｅに至っている。ここで、鍵の初速度をV_0、鍵の位置をＸ、鍵の駆動開始時点からの時刻をｔとすれば、鍵の軌道は、

【００６６】

【数３】

$$X = V_0 \cdot t + X_0$$

と表される。

また、鍵がリファレンスポイントＸｒに達する時刻をｔｒ’とすると、

【００６７】

【数４】

$$Xr = V_0 \cdot tr' + X_0$$

なる式が成り立つから、この数４から時刻ｔｒ’を求めることができる。したがって、押鍵を開始する絶対時刻（以下、押鍵開始時刻という）t_0は、次式によって求めることができる。

【００６８】

【数５】

$t_0 = tr - tr'$
$= tr - (Xr - X_0) / V_0$
なお、リファレンス時刻ｔｒは、前述のように、打弦時刻からリファレンス時間差Ｔｒを減算することによって求める。
上記数５によって押鍵開始時刻t_0を求め、この時刻から、数３で示される軌道に従って鍵１を駆動すれば、鍵１は、リファレンス時刻ｔｒにおいて正確にリファレンスポイントＸｒに達し、しかも、その時の速度は、打弦強度データに対応したリファレンス速度Ｖｒとなる。
【００６９】
なお、鍵の挙動については、直線軌道（等速運動）を想定しているから、リファレンス速度Ｖｒと初速度V_0は等しい。そして、リファレンス速度Ｖｒは、前述の数１によって求められるから、数５で求めた押鍵開始時刻t_0から一定速度Ｖｒで鍵を駆動するように制御（速度制御）しても上記と同様の結果を得ることができる。
【００７０】
▲２▼離鍵時の軌道データ作成
次に、離鍵時の軌道データ作成について説明する。
まず、鍵の位置をＸＮ、離鍵初速度をV_0N（＜０）、離鍵開始時点からの時刻をｔＮ、エンド位置をＸｅとすれば、離鍵時の鍵軌道は、次式で表される。
【００７１】
【数６】
$XN = V_0N \cdot tN + Xe$
ここで、図８は数６で示される軌道を示す図である。
さて、前述のように、演奏記録部３０（図１参照）は、キーセンサ２５内の下方のフォトセンサが受光状態になってから上方のフォトセンサが受光状態になるまでの時間を測定して離鍵速度ｖｋＮを検出し、また、上方のフォトセンサが受光状態になった時刻を離鍵時刻ｔｋＮとして検出する。この場合、離鍵時刻ｔｋＮにおけるダンパー６は、弦４に接して音の減衰を開始する状態なっている（そのような状態になるようフォトセンサの位置が調整されている）。そして、このようにして検出された離鍵速度ＶｋＮおよび離鍵時刻ｔｋＮは、それぞれ演奏情報を構成するデータとして記録され、再生時に読み出される。
【００７２】
ここで、ダンパー６が弦４に接するときの鍵の位置を離鍵リファレンスポイントＸｒＮと定義すれば、鍵１が離鍵リファレンスポイントＸｒＮに達したときに、離鍵状態になったということができる。したがって、鍵１が離鍵リファレンスポイントＸｒＮに達する時刻（以下、離鍵リファレンス時刻ｔｒＮという）と、演奏情報中の離鍵時刻ｔｋＮとが一致するように鍵位置を制御すれば、正確な離鍵タイミング制御を行うことができる。
【００７３】
また、ダンパー６が弦４に接する速さは、音の減衰状態に影響を与えるから、これを忠実に再現することが望ましい。この速さは、離鍵速度ＶｋＮに対応するから、結局、離鍵リファレンスポイントＸｒＮにおける鍵速度（以下、離鍵リファレンス速度ＶｒＮという）を正確に離鍵速度ＶｋＮに一致させれば、音の減衰状態が正確に再現される。
ここで、鍵の駆動が開始される時刻を基準（＝０）にして、鍵がリファレンスポイントＸｒＮに達する時刻をｔｒＮ’とすると、
【００７４】
【数７】
$XrN = V_0N \cdot trN' + XeN$
（ただし、直線軌道だから$V_0N = VrN = VkN$）
なる関係が成り立ち、この数７より時刻ｔｒＮ’を求めることができる。したがって、次式によって離鍵開始時刻t_0Nを求めることができる。
【００７５】
【数８】

$$t_0N = trN - trN'$$
$$= trN - (XrN - XeN) / V_0N$$

この数８によって離鍵開始時刻ｔ$_0$Ｎを求め、この時刻から、数６で示される軌道に従って鍵を駆動すれば、鍵は離鍵時刻ｔｋＮにおいて離鍵リファレンスポイントＸｒＮに達し、記録時の離鍵状態を忠実に再現することができる。
なお、時刻ｔ$_0$から速度Ｖ$_0$Ｎ（＝ｖｋＮ：離鍵速度）で鍵駆動するように制御（速度制御）しても上記と同様の結果を得ることができる。

【００７６】

▲３▼交差時の軌道データ作成（交差処理）

押鍵軌道および離鍵軌道は上述のようにして作成されるが、鍵の操作には、離鍵の途中から次の押鍵に移ったり、あるいは、押鍵の途中から離鍵され場合がある。このような場合においては、作成した押鍵軌道と離鍵軌道とが交差する。
例えば、図９はこのような軌道の交差状態を示しており、図示の状態では、時刻ｔ$_0$から時刻ｔｃまで押鍵が行われ、時刻ｔ$_1$から時刻ｔ$_4$まで離鍵が行われている。このとき、上述の方法によって生成される押鍵軌道は、時刻ｔ$_0$にレスト位置Ｘ$_0$を離れ、時刻ｔ$_3$においてエンド位置Ｘｅに達する軌道であり、また、離鍵軌道は時刻ｔ$_0$Ｎにエンド位置Ｘｅを離れ、時刻ｔ$_4$においてレスト位置Ｘ$_0$に達する軌道である。

【００７７】

ここで、交差する時刻ｔｃを求めることができれば、ｔ$_1$～ｔｃまでは押鍵軌道に基づいて鍵１を制御し、ｔｃ～ｔ$_3$までは離鍵軌道に基づいて鍵を制御すればよい。ここで、図９に示すような、押鍵の後に発生した離鍵の軌道が交差する場合は、交差時刻ｔｃは次のようにして求めることができる。

【００７８】

【数９】

$$tc = (V_0 \cdot t_3 - V_0N \cdot t_0N) / (V_0 - V_0N)$$
$$= t_0N + V_0 \cdot (t_3 - t_0N) / (V_0 - V_0N)$$

なお、数９におけるｔ$_3$は、次式により算出される。

【００７９】

【数１０】

ｔ$_3$＝ｔ$_0$＋（Ｘｅ－Ｘ$_0$）／Ｖ$_0$

また、離鍵の後に発生した押鍵の軌道が交差する場合も、２つの直線軌道の交点を求めればよいので、上記と同様の考え方により交差時刻を求めることができる。また、上述したように、本実施例にあっては適宜ダミーイベントを追加するから、図２２（ｅ）のように、押鍵軌道同士あるいは離鍵軌道同士が交差することも考えられる。押鍵軌道同士が交差する場合には、同図に示すように、先にエンド位置に達する軌道（押鍵速度の速い軌道）を採用するのが妥当である。これは、強いアタックを忠実に再現するためである。一方、離鍵軌道同士が交差する場合には、最初に離鍵速度の速い軌道を採用し、交差点を過ぎた後は、離鍵速度の遅い軌道を採用するとよい。

【００８０】

このようにして、交差時刻を求め、押鍵軌道と離鍵軌道を組み合わせることにより、交差時の軌道データを作成する。
以上が、再生前処理部１０における軌道データの作成であり、このようにして作成された軌道データは、図１に示すモーションコントローラ１１に供給される。モーションコントローラ１１においては、作成された軌道データに基づいて、各時刻における鍵１の位置に対応した位置制御データ（Ｘ）を作成し、サーボコントローラ１２に供給する。サーボコントローラ１２は、位置制御データ（Ｘ）に応じた励磁電流をソレノイド５に供給するとともに、ソレノイド５から供給されるフィードバック信号と制御データ（Ｘ）を比較し、両者が一致するようにサーボ制御を行う。

【００８１】

▲４▼作成された軌道データの具体例

以下、図２１、図２２、図２３について、上述した軌道データを作成した場合の具体例を説明する。

まず、図２１（ｅ）によれば、時刻t_{15}以前の再生リファレンス軌道は、レスト位置から等速運動をしてエンド位置に至っている。その速度はハンマー速度Vh_1に対応したリファレンス速度である。その後、リファレンス軌道は、直ちに最大離鍵速度でレスト位置に向かっている。これは、ダミーの離鍵イベントが、当初においてはハンマーイベントと同一の時刻（時刻t_{15}，同図（ｄ）参照）に追加されたためである。この離鍵時の軌道は、時刻t_{25}のハンマーイベントに対応する押鍵軌道と交差しているため、上述した交差処理が行われる。この結果、リファレンス軌道は同図の実線に示すようになる。

【００８２】

次に、図２２（ｅ）によれば、時刻t_1の押鍵イベント、時刻t_2の離鍵イベントおよび時刻t_{15}のハンマーイベントに対応して、時刻t_{22}およびt_{23}に一点鎖線L_1と交差する軌道が生成されている。一方、同図（ｄ）によれば、時刻t_3に弱いハンマーイベントが追加されているから、これに対応して傾きの小さな軌道（時刻t_{21}に一点鎖線L_1と交差する軌道）が生成されている。両者は交差するから、上述した場合と同様に交差処理が行われる。

【００８３】

次に、図２３（ｆ）によれば、時刻t_1に至るまで、傾きの小さな押鍵軌道が生成されている。これは、時刻t_1に追加された弱いハンマーイベント（同図（ｄ）参照）に対応するものである。そして、時刻t_1以降は、最大離鍵速度でレスト位置に向かう離鍵軌道が生成されている。その後、時刻t_3のハンマーイベントに対応して押鍵軌道が生成されており、その傾きはハンマー速度Vh_1に対応したリファレンス速度に等しい。この例においては、押鍵／離鍵軌道は交差しないから、得られた押鍵／離鍵軌道がそのままリファレンス軌道になる。

【００８４】

▲５▼演奏再生の全体動作

次に、演奏再生時における実施例の全体動作を説明する。

まず、再生前処理部１０は、記録媒体から演奏情報を読み出し、その中の打弦時刻データおよび打弦速度データに基づいて、押鍵軌道を作成する。

作成された押鍵軌道データは、モーションコントローラ１１に供給され、ここで、位置制御データ（Ｘ）に変換される。すなわち、押鍵開始時刻t_0になると、数３に示される軌道データがモーションコントローラ１１に供給され、モーションコントローラ１１は、数３のＸを時間経過とともに順次演算し、位置制御データ（Ｘ）を作成する。この位置制御データ（Ｘ）は、サーボコントローラ１２に供給され、これにより、数３のＸに対応した励磁電流がソレノイド５に供給される。この場合、サーボコントローラ１２は、ソレノイド５のフィードバック信号と位置制御データ（Ｘ）とを比較し、両者が一致するように励磁電流を制御するから、ソレノイド５のプランジャの突出量は、数３のＸに対応したものとなる。したがって、鍵１は数３で示される直線軌道に従って押下されていき、リファレンスポイントＸｒにおいてリファレンス速度Ｖｒを有する運動を行う。これにより、記録時の打弦強度を忠実に再現する打弦が行われる。

【００８５】

次に、再生前処理部１０は、演奏情報の中の離鍵時刻データ、離鍵速度データに応じて、離鍵軌道を作成し、モーションコントローラ１１に供給する。モーションコントローラ１１は、上述の場合と同様にして、数６のＸＮを時間の経過に応じて順次演算し、位置制御データ（Ｘ）を作成し、サーボコントローラ１２に供給する。この結果、サーボコントローラ１２は数６に対応した励磁電流制御を行い、鍵１は数６で示される直線軌道に従ってレスト位置X_0に戻っていく。これにより、鍵１の運動は、離鍵リファレンスポイントＸ

ｒＮにおいて離鍵リファレンス速度ＶｒＮを有するものとなり、記録時の離鍵状態が忠実に再現される。

【００８６】

一方、連打等が行われ、押鍵軌道と離鍵軌道が交差するときは、再生前処理部１０が数１０によって交差時刻ｔｃを演算し、この時刻を境に、押鍵軌道と離鍵軌道を切り換えてモーションコントローラ１１に供給する。これにより、鍵１は押鍵軌道の途中から離鍵軌道に、あるいは離鍵軌道の途中から押鍵軌道に切り換えられ、いわゆる、ハーフストロークの奏法に従った運動を行う。

【００８７】

ＩＩ．第２実施例

Ａ．実施例の制御原理

次に、この発明の第２実施例について説明する。この実施例が前述した第１実施例と異なる点は、再生前処理部１０において放物線の軌道データを作成する点である。そこで、説明の簡略化のため、第２実施例については、再生前処理部１０の動作原理および動作についてのみ説明する。

この実施例における再生前処理部１０は、押鍵時の鍵軌道として放物線、離鍵の鍵軌道として直線を想定するので、次のような手法で軌道データの作成を行う。

【００８８】

▲１▼押鍵時の軌道データ作成

押鍵軌道を次のように規定する。

【００８９】

【数１１】

$X = a/2 \cdot t^2 + b \cdot t + c$

ここで、ａは加速度であり、次式によって求める。

【００９０】

【数１２】

$a = (Vr - Vk) / (tr - (tk_1 + tk_2)/2)$

数１２におけるリファレンス速度Ｖｒは、前述の実施例と同様に、数１を用いて打弦速度から求める。また、tk_1は、次式によって求めることができる。

【００９１】

【数１３】

$tk_1 = tk_2 - Xd/Vk$

ここで、Ｘｄは、キーセンサ２５内の２つのセンサの取り付け間隔である。また、図１０は、この実施例における放物線軌道を示す図であり、図において、Xk_2、Xk_1はキーセンサ２５内の２つのセンサの取り付け位置である。この図からも判るように、キーオン速度Ｖｋは、シャッタ２６がXk_1とXk_2の間を通過する間の平均速度であり、放物線軌道においては、中間時刻である時刻$(tk_1 + tk_2)/2$における速度である。

前述した数１２は、リファレンス速度Ｖｒとキーオン速度Ｖｋの速度差を、リファレンス時刻ｔｒと中間時刻$(tk_1 + tk_2)/2$の時間差で除したもので、言い替えれば、リファレンスポイントＸｒとセンサ中間点との間の速度変化から加速度を算出するものである。なお、加速度一定という推定のもとでは、任意の２点間の速度変化を基にして加速度を検出することができる。

【００９２】

以上のことから判るように、演奏データ中の打弦時刻と打弦速度から、リファレンス時刻ｔｒとリファレンス速度Ｖｒが求められ、さらに、キーオン速度Ｖｋとキーオン時刻tk_2を参照して加速度ａが求められる。

したがって、あとは押鍵開始時刻と初速度が求められれば、図１０に示す放物線軌道に沿ってキーを駆動できることになる。次に、これらの求め方について説明する。

まず、図１１に示すように、押鍵開始時刻をt_0、初速度をV_0とすれば、数１１における係数ｂは、

【００９３】
【数１４】
$b = V_0 - a \cdot t_0$
と表され、また、数１１における定数ｃは、
【００９４】
【数１５】
$c = X_0 - (a/2) \cdot t_0^{\,2} + b \cdot t_0$
と表される（ただし、数１５におけるX_0は前述のようにレスト位置）。
ここで、押鍵開始時刻を基準（＝０）にした場合のリファレンス時刻をｔｒ’とし、さらに、数１３、数１４を用いて、数１１をリファレンスポイントＸｒにおける式に書き直すと次式になる。
【００９５】
【数１６】
$Xr = (a/2) \cdot tr'^{\,2} + V_0 \cdot tr' + X_0$
ここで、リファレンス速度Ｖｒは、
【００９６】
【数１７】
$Vr = a \cdot tr' + V_0$
と表すことができるから、この関係を数１６に代入して整理すると、
【００９７】
【数１８】
$0 = (a/2) \cdot tr' - Vr \cdot tr' - (X_0 - Xr)$
となり、ここから、ｔｒ’を求めると、
【００９８】
【数１９】
$tr' = (Vr - (Vr^2 + 2a(X_0 - Xr))^{1/2})/a$
となる。したがって、加速度ａ、リファレンス速度Ｖｒ、リファレンスポイントＸｒ、およびレスト位置X_0が既知であれば、鍵が押鍵開始時からリファレンスポイントＸｒに至るまでの時刻ｔｒ’が求められる。また、リファレンス時刻ｔｒと押鍵開始時刻t_0の関係は、
【００９９】
【数２０】
$tr' = tr - t_0$
であるから、数２０を用いて、押鍵開始時刻t_0が求められる。
また、数１６に数１７を代入して初速度V_0について整理して解けば、
【０１００】
【数２１】
$V_0 = (Vr^2 + 2a(X_0 - Xr))^{1/2}$
となり、これにより初速度V_0を求めることができる。
したがって、以上のようにして求めた初速度V_0、加速度ａを、数１１に代入して軌道を求め、押鍵開始時刻t_0からその軌道に従って鍵を駆動することにより、打弦時刻と打弦速度とを正確に再現することができる。しかも、人間の鍵操作に近い放物線軌道なので、演奏者の微妙なニュアンスまで表現することが可能になる。
なお、時刻t_0から初速度V_0で鍵を駆動し、後は、加速度ａに対応した速度変化分を用いて速度制御を行っても、上記と同様の効果が奏することができる。
【０１０１】
▲２▼離鍵時の軌道データ作成
離鍵については、直線軌道を想定するので、前述した第１実施例と同様の離鍵軌道データ作成を行う。
【０１０２】

▲３▼交差時の軌道データ作成

この実施例における押鍵軌道と離鍵軌道の交差は、放物線軌道と直線軌道との交差であるので、交差時刻ｔｃは、次のようにして求めることができる。

【０１０３】

【数２２】

$$tc=(-\Delta b+(\Delta b^{2}-2\cdot\Delta a\cdot\Delta c)^{1/2})/\Delta a$$

ただし、数２２におけるΔａは、

【０１０４】

【数２３】

$$\Delta a=a-aN$$

であり、ａＮは、離鍵時の加速度である。この実施例の場合は、離鍵時の加速度は０であるので、Δａ＝ａ（押鍵時の加速度）となる。

また、数２２におけるΔｂは、

【０１０５】

【数２４】

$$\begin{aligned}\Delta b&=Vr-VrN-(a\cdot tr-aN\cdot trN)\\&=Vr-VrN-a\cdot tr\qquad(\because aN=0)\end{aligned}$$

であり、Δｃは、

【０１０６】

【数２５】

$$\begin{aligned}\Delta c&=Xr-XrN\\&\quad+(a\cdot tr^{2}-aN\cdot trN^{2})/2\\&\quad-(Vr\cdot tr-VrN\cdot trN)\\&=Xr-XrN\\&\quad+(a\cdot tr^{2})/2\\&\quad-(Vr\cdot tr-VrN\cdot trN)\qquad(\because aN=0)\end{aligned}$$

である。このようにして、交差時刻ｔｃを求め、押鍵軌道と離鍵軌道を組み合わせることにより、交差時の軌道データを作成する。

【０１０７】

以上が、第２実施例における再生前処理部１０の軌道データ作成処理である。このようにして作成された軌道データは、図１に示すモーションコントローラ１１に供給され、モーションコントローラ１１においては、作成された軌道データに基づいて、各時刻における鍵１の位置に対応した位置制御データ（Ｘ）を作成し、サーボコントローラ１２に供給する。このモーションコントローラ１１とサーボコントローラ１２の構成は、第１実施例と同様になっている。

【０１０８】

Ｂ．第２実施例の動作

次に、第２実施例の動作について説明するが、第２実施例の記録動作は第１実施例と同様である。また、再生動作も概ね第１実施例の動作と同じであるため、異なる点だけを説明する。

【０１０９】

まず、再生前処理部１０は、記録媒体から演奏情報を読み出し、その中の打弦時刻データ、打弦速度データ、キーオン速度データおよびキーオン時刻データに基づいて、放物線の押鍵軌道を作成する。

作成された放物線の押鍵軌道データは、モーションコントローラ１１に供給され、ここで、位置制御データ（Ｘ）に変換される。すなわち、押鍵開始時刻t_0になると、数１１に

示される軌道データがモーションコントローラ１１に供給され、モーションコントローラ１１は、数１１のＸを時間経過とともに順次演算し、位置制御データ（Ｘ）を作成する。この位置制御データ（Ｘ）は、サーボコントローラ１２に供給され、これにより、数１１のＸに対応した励磁電流がソレノイド５に供給される。この場合、サーボコントローラ１２は、ソレノイド５のフィードバック信号と位置制御データ（Ｘ）とを比較し、両者が一致するように励磁電流を制御するから、ソレノイド５のプランジャの突出量は、数１１のＸに対応したものとなる。したがって、鍵１は数１１で示される放物線軌道に従って押下されていき、リファレンスポイントＸｒにおいてリファレンス速度Ｖｒおよび加速度ａを有する運動を行う。この結果、記録時の打弦強度を忠実に再現する打弦が行われる。しかも、人間の演奏状態に近い放物線軌道により、加速度をも再現するので、演奏の微妙なニュアンスも再現される。

また、押鍵軌道と離鍵軌道が交差するときは、再生前処理部１０が数２２によって交差時刻ｔｃを演算し、この時刻を境に、押鍵軌道と離鍵軌道を切り換えてモーションコントローラ１１に供給する。これにより、鍵１は放物線の押鍵軌道の途中から直線の離鍵軌道に、あるいは直線の離鍵軌道の途中から放物線の押鍵軌道に切り換えられ、いわゆる、ハーフストロークの奏法に従った運動を行う。

【０１１０】

ここで、図１２に本実施例によって鍵駆動した場合の実験例を示す。この図１２に示す３つのグラフは、いずれも横軸が時間であり、縦軸が鍵のレスト位置X_0からの移動量を示している。

この図に示す（イ）は、演奏者が実際に演奏を行った場合の鍵の軌道を示しており、図に示す時刻t_{10}、t_{11}の付近はハーフストローク奏法のために、鍵がレスト位置X_0に戻る前に次の押鍵動作に入っている。また、時刻t_{12}付近は、鍵は動いているが打弦は行われなかった部分である。

また、図１２（イ）に示す演奏から得られる打弦時刻、打弦速度、キーオン時刻、キーオン速度、離鍵時刻および離鍵速度を基にして、本実施例において再現した押鍵軌道および離鍵軌道が同図（ハ）に示す軌道である。また、同図（ロ）は、同図（イ）および（ハ）に示す２つの軌道を重ねたものであり、両者が良く一致していることが判る。例えば、ハーフストロークの部分については、再生軌道もハーフストローク軌道になっており、打弦が行われなかった時刻t_{12}の部分については、再生軌道は鍵を押し切った状態が継続されている。

【０１１１】

また、図１３は、図１２の一部を拡大したものであり、この図からも実際の演奏による鍵軌道と、実施例において再現した鍵軌道との一致性が確認される。なお、実際の演奏による鍵軌道に対して、実施例で再現した鍵軌道は揺らぎが少なく一見単純に見えるが、図１２（イ）の軌道に含まれる揺らぎは、演奏者の意図とは違った部分（ミスによる部分）がほとんどである。したがって、実施例による軌道は、ミス部分を除外し、理想に近づいた軌道であると言える。

【０１１２】

Ｃ．第２実施例の変形

▲１▼ダンパー離弦タイミングの再現

鍵を押下していくと、まず、ダンパー６（図１参照）が弦から離れ、次いで、ハンマー２が打弦を行うことは周知の通りである。この場合、ダンパー６が弦から離れても、打弦が行われるまでは音は発生しないから、ダンパー６の離弦を正確に再現しても意味がないという考えもある。しかしながら、複数の鍵を同時あるいは近接したタイミングで押鍵する場合などには、他の弦の振動や共鳴の影響のために、打弦前であっても、ダンパーが離れていれば、その弦が振動することがある。したがって、演奏における他の弦の影響を忠実に再現するために、ダンパーの離弦タイミングを正確に再現することも重要であると言える。

そこで、以下のようにして、ダンパー離弦タイミングを推定する。まず、前述のように、

キーオン時刻をtk_2、キーオン速度をVkとし、キーオン位置（鍵１に設けられたシャッタ２６がキーセンサ２５内の下方のフォトセンサを遮光したときの鍵の位置）をXk_2とする。次に、図１４に示すように、ダンパーの離弦に対応する鍵の位置をX_1とし、キーオン速度Vkの傾きに沿って直線を引く。そして、この直線と位置X_1との交点の時刻をダンパー離弦時刻t_1と推定する。すなわち、ダンパー離弦時刻t_1は次式によって求められる。

【０１１３】

【数２６】

$$t_1 = tk_2 - (Xk_2 - X_1) / Vk$$

以上のようにして、ダンパー離弦時刻t_1が求められれば、前述の各実施例と同様にしてリファレンス速度Vrおよびリファレンス時刻trを求め、これらを満たす放物線軌道を求める。この放物線軌道を図１５に示す。この軌道は、リファレンス時刻trとリファレンスポイントXrを原点にとれば、次のように表される。

【０１１４】

【数２７】

$$X = ((X_1 - Vr \cdot (t_1 - tr)) / (t_1 - tr)^2) \cdot t^2 + Vr \cdot t$$

したがって、数２７に示される軌道にしたがって、ソレノイドの励磁電流を制御すれば、ダンパー離弦時刻を正確に再現することができる。なお、励磁電流の制御は、前述した各実施例と同様に行えばよい。

【０１１５】

▲２▼加速度、初速度の固定

第２実施例においては、押鍵時の加速度ａと初速度Ｖ０の双方を演算によって求めるようにしたが、いずれか一方を固定値にして、計算を簡略化してもよい。

この場合の固定値は、実験等を繰り返すことによって、統計的に適切な値を決めることが望ましい。

例えば、加速度をａを固定する場合には、ａ＝２．５ｍ／ｓ^{2}に設定すると、比較的良好な結果が得られることが判った。ここで、ａ＝２．５ｍ／ｓ^{2}にした場合の実験結果を図１６に示す。

【０１１６】

この図に示す（イ）は、演奏者が実際に演奏を行った場合の鍵の軌道を示しており、（ハ）は同図（イ）に示す演奏から得られる打弦時刻、打弦速度を基にするとともに、加速度を固定して再現した押鍵軌道を示す。すなわち、前述した数１１に示す軌道における加速度ａの値を固定し、数１２に示す加速度算出を省略して押鍵軌道を求めたものである。なお、離鍵軌道は、直線軌道であるため、前述の実施例と同様である。

また、同図（ロ）は、同図（イ）および（ハ）に示す２つの軌道を重ねたものであり、両者が良く一致していることが判る。例えば、ハーフストロークの部分についても、良好な一致性が認められる。なお、図１７（イ）、（ロ）、（ハ）は、図１６（イ）、（ロ）、（ハ）の軌道を一部拡大したものである。

以上のように加速度を固定すると、演算が省略できるため、処理速度が早くなる。

【０１１７】

次に、初速度V_0を固定した場合について説明する。この場合には、数２１の演算を省略し、予め設定した初速度V_0を用いて数１１に示す軌道を演算すればよい。

ここで、初速度V_0を０．１ｍ／ｓにした場合の実験結果を図１８に示す。

この図に示す（イ）は、演奏者が実際に演奏を行った場合の鍵の軌道を示しており、（ハ）は同図（イ）に示す演奏から得られる打弦時刻、打弦速度を基にするとともに、初速度V_0を固定して再現した押鍵軌道を示す。すなわち、前述した数２１の初速度演算を省略して初速度V_0を固定し、この条件の基で数１１に示す押鍵軌道を求めたものである。なお、離鍵軌道は、直線軌道であるため、前述の実施例と同様である。そして、同図（ロ）は、同図（イ）および（ハ）に示す２つの軌道を重ねたものであり、両者が良く一致していることが判る。なお、図１９（イ）、（ロ）、（ハ）は、図１８（イ）、（ロ）、（ハ

）の軌道を一部拡大したものである。

【０１１８】

ＩＩＩ．第３実施例

Ａ．実施例の制御原理

▲１▼リアルタイムの再生

上記第１，第２実施例においては、まず演奏記録部３０において各種鍵イベントおよびハンマーイベントを記録しておき、演奏記録が終了した後に記録後処理部３１による正規化処理、あるいはダミーイベントの追加等が行われていた。

しかし、演奏データをリアルタイムに出力しなければならない場合もある。例えば、演奏者による自動ピアノの生演奏を行いながら演奏データを検出し、これによって別室の他の自動ピアノを駆動することも考えられる。このような場合、データ送信側自動ピアノにおける演奏記録部３０と記録後処理部３１、および受信側自動ピアノにおける再生前処理部１０の処理をリアルタイムに実行する必要がある。しかし、これを実現するためには、以下のような点について留意する必要がある。

【０１１９】

▲２▼データ遅延の必要性

上記第１，第２実施例においては、押鍵速度や打弦速度等に基づいてリファレンス軌道を生成し、これに基づいて各鍵を駆動していたため、少なくとも押鍵速度または打弦速度のうち一方が得られなければリファレンス軌道を生成できず、鍵を駆動することができない。本実施例においてもリファレンス軌道を求める原理は第１，第２実施例と共通するから、受信側自動ピアノの演奏タイミングは送信側自動ピアノのものと比較して若干遅延せざるを得ない。

【０１２０】

すなわち、送信側自動ピアノにおいて鍵１が押鍵されると、やがて鍵１はセンサ位置に達し、押鍵時刻と押鍵速度が検出される。その後、ハンマー２がセンサＳＥ１，ＳＥ２を通過すると、打弦時刻と打弦速度が検出される。これらのデータに基づいて押鍵時のリファレンス軌道が生成された後、初めて受信側自動ピアノの鍵の駆動を開始することができるのである。また、ダミーイベントが有る場合にはイベントの内容を修正する処理を行うので、各押離鍵に対応して行われるリファレンス計算に要する時間もダミーイベントの有無によって各押離鍵で異なる。そこで、本実施例においては、各イベント毎にタイムコード（時刻情報）を付与し、各イベント間の相対的な時間差を保持しつつ演奏データを遅延させる。このときの遅延時間は情報の最大処理時間、すなわち押鍵開始時刻から打弦時刻までの最大時間とリファレンス軌道の計算時間と記録後処理部３１の処理時間との合計値である。以下、前二者の内容について各々について説明しておく。

【０１２１】

（ｉ）押鍵開始時刻から打弦時刻までの最大時間

ここで、押鍵開始時刻から打弦時刻までの最大時間とは、ステップＳＰ１０６（図６参照）で説明した値“Ｂ／押鍵速度”よりも若干大きい値である。すなわち、ステップＳＰ１０６で比較対象にされている「時間差」とは押鍵時刻から打弦時刻までの時間であって、押鍵開始時刻から打弦時刻までの時間は、さらに長くなるからである。この最大値を決定するためには、押鍵速度の最小値を予め決定する必要がある。すなわち、押鍵速度の最小値は発音されない程度の速度にしておき、実際に検出された押鍵速度が最小値よりもさらに小さい場合には、押鍵速度を強制的に該最小値に設定するとよい。

（ｉｉ）リファレンス軌道の最大計算時間

リファレンス軌道の計算は、複数の鍵について同時に行われる場合もあるため、一つの押鍵イベントに要する処理の数倍の時間を想定しておくとよい。

【０１２２】

▲３▼イベントデータの並び替えの必要性

自動演奏を再生する場合には、タイムコードの小さい順に各イベントを再生する必要がある。このことは、一般的な自動ピアノでは、検出された順序で各イベントを記録しておき

、記録した順序で再生することにより、自然に解決される。

【０１２３】

しかし、本実施例のようにリファレンス軌道を生成する場合には、検出時のイベント順序と比較して、再生時のイベント順序は異なる。このことを図２６（イ）～（ハ）を参照し説明する。同図（イ）において、時刻t_4にキーコードＫＣ１の押鍵イベントが検出されており、その後、時刻t_5にキーコードＫＣ２の押鍵イベントが検出されている。これらキーコードＫＣ１，２のリファレンス軌道は同図（ロ）に示すようになるが、キーコードＫＣ２の押鍵開始時刻t_{23}はキーコードＫＣ１の押鍵開始時刻t_{24}よりも早くなっている。従って、リファレンス軌道を押鍵開始等のイベントによって表現するならば、キーコードＫＣ１，２のイベントの順序が逆転することになる。

【０１２４】

従って、処理を行う順番で各イベントを配置するために、イベントデータの順序を並び替える必要が生じる。ここで、各イベントデータの内容としては、キーコード、押鍵開始時刻（タイムコード）、押鍵初速度、加速度等の内容を含める必要があり、メモリ上でこれらのデータ自体を入れ替えるのは多大な処理時間を要する。そこで、本実施例にあっては、メモリ上のイベントデータを構造体によって表現し、前後のイベントに係る構造体のアドレスを指定するポインタ変数を、この構造体のメンバとして含ませることにより、イベントの並び替えを高速に行うこととしている。

【０１２５】

その詳細を図２４（イ）～（ハ）を参照し説明する。同図（イ）においてST_0～ST_2は構造体であり、x_0～x_2はメモリ上における各構造体の先頭アドレスである。構造体ST_0～ST_2は、各々一つのイベントに係るデータ（キーコード、押鍵開始時刻等）と、ポインタ＊ＰＲＩＯＲ，＊ＮＥＸＴをメンバとして含んでいる。ここで、構造体ST_0～ST_2の順序で各イベントを実行する場合には、各構造体のポインタ＊ＮＥＸＴには、次に実行すべきイベントに係る構造体の先頭アドレスが格納される。すなわち、構造体ST_0のポインタ＊ＮＥＸＴには構造体ST_1の先頭アドレスであるアドレスx_1が格納され、同様に構造体ST_1のポインタ＊ＮＥＸＴにはアドレスx_2が格納されている。また、最後に実行されるイベントに係る構造体ST_2のポインタ＊ＮＥＸＴには、最初に実行されるイベントに係る構造体ST_0の先頭アドレスx_0が格納されている。

【０１２６】

また、各構造体のポインタ＊ＰＲＩＯＲには、先に実行すべきイベントに係る構造体の先頭アドレスが格納されている。但し、最初に実行すべきイベントに係る構造体ST_0のポインタ＊ＰＲＩＯＲには、最後に実行されるイベントに係る構造体ST_2の先頭アドレスx_2が格納されている。かかるデータ構造においては、各構造体はポインタ＊ＰＲＩＯＲ，＊ＮＥＸＴによってリンクされており、各構造体のポインタ＊ＰＲＩＯＲ，＊ＮＥＸＴを順次追跡していくことにより、昇順または逆順に各イベントに係る構造体を高速にアクセスすることができる。

【０１２７】

ここで、何れかのイベントを削除する場合は、そのイベントに係る構造体をリンクから外してしまえば良い。同図（ロ）に示す例にあっては、構造体ST_0のポインタ＊ＮＥＸＴがアドレスx_2に変更され、構造体ST_2のポインタ＊ＰＲＩＯＲがアドレスx_0に変更されており、これによって構造体ST_1がリンクから外されている。すなわち、ここに言う「削除」とは、上述のように何れかの構造体をリンクから外すことであり、削除された構造体の内容自体は、実際に消去されない限りメモリ内に残存することになる。

【０１２８】

また、任意の箇所になんらかのイベントを追加する場合は、挿入箇所におけるポインタ＊ＰＲＩＯＲ，＊ＮＥＸＴを変更すればよい。同図（ハ）は、先頭アドレスx_{new}に記憶されている構造体ST_{new}をリンクに組込む例を示している。図において構造体

ST_1 のポインタ ＊ＮＥＸＴ および構造体ST_2 のポインタ ＊ＰＲＩＯＲ は、各々アドレスx_{new} に変更されており、構造体ST_{new} のポインタ ＊ＰＲＩＯＲ はアドレスx_1 に、ポインタ ＊ＮＥＸＴ はアドレスx_2 に各々設定されている。これにより、構造体ST_{new} は、上記リンクにおいて、構造体ST_1 と構造体ST_2 の間に追加される事になる。イベントの順序を変更する場合は、移動したいイベントに係る構造体を削除し、削除したイベントを所望の箇所に追加すればよい。

【０１２９】

Ｂ．実施例の構成

次に、本実施例の構成を図２７を参照し説明する。図において１００，２００は自動ピアノであり、図１のものと同様に構成されている。３００はリアルタイム処理装置であり、ＣＰＵ、ＲＡＭ、ＲＯＭ等によって構成されている。リアルタイム処理装置３００は、後述する処理プログラムに基づいて、図１における演奏記録部３０、記録後処理部３１および再生前処理部１０と同等の処理をリアルタイムに実行するものである。すなわち、リアルタイム処理装置３００は、自動ピアノ１００内の各キーコードに対応するセンサＳＥ１，ＳＥ２，２５に接続され自動ピアノ１００における各種イベントを検出するとともに、これらのイベントに基づいてリファレンス軌道を生成し、自動ピアノ２００内のモーションコントローラ１１を介して自動ピアノ２００を駆動する。

【０１３０】

Ｃ．実施例の動作

Ｃ－１．リアルタイム処理装置３００の動作

次に、本実施例の具体的動作を説明するが、最初にリアルタイム処理装置３００の動作について詳述しておく。

まず、リアルタイム処理装置３００の電源を投入すると、図２５に示す処理プログラムが起動される。図において処理が開始されると、ステップＳＰ１において所定の初期設定が行われる。このステップＳＰ１においては、所定の時刻レジスタが「０」に設定される。ところで、リアルタイム処理装置３００においては「１ｍｓｅｃ」毎にタイマ割込が発生し、図２８に示す割込処理ルーチンが起動され、上記時刻レジスタが「１」だけインクリメントされる。従って、この時刻レジスタは、ミリ秒単位でリアルタイム処理装置３００内の絶対時刻を表示することになる。

【０１３１】

次に、図２５において処理がステップＳＰ２に進むと、時間監視キーがタイムアップしたか否かが判定される。ここで、時間監視キーとは、押鍵イベントが検出される毎にメモリ内に設けられ、該押鍵イベントに対するハンマーイベントが発生すべき最も遅い時刻を記憶しておく変数である。この時点では時間監視キーは設定されていないから「ＮＯ」と判定され、処理がステップＳＰ４に進む。ステップＳＰ４においては、自動ピアノ１００から新たなイベントデータを受信したか否かが判定される。新たなデータを受信していなければ「ＮＯ」と判定され処理がステップＳＰ１２に進む。

【０１３２】

ステップＳＰ１２においては、現在時刻が出力時刻（詳細は後述する）を超過したか否かが判定される。この時点では出力時刻が全く設定されていないため「ＮＯ」と判定され、処理がステップＳＰ２に戻る。以後、なんらかのイベントデータが入力されるまで、ステップＳＰ２，４，１２から成るループが繰り返されることになる。

【０１３３】

次に、実際にイベントデータが入力された場合の処理を図２６を参照して説明する。同図（イ）においては、時刻t_1 にキーコードＫＣ１の押鍵イベントが発生し、リアルタイム処理装置３００に受信される。このイベントが検出された後、処理がステップＳＰ４に進むと、ここで「ＹＥＳ」と判定され処理がステップＳＰ５に進む。ステップＳＰ５においては、受信したイベントに係る構造体がメモリ内に生成される。すなわち、この構造体は、キーコード、押鍵時刻（この時点における時刻レジスタの内容）、押鍵速度、およびポインタ ＊ＰＲＩＯＲ，＊ＮＥＸＴ 等をメンバとするものであり、ポインタ ＊ＰＲ

ＩＯＲ，＊ＮＥＸＴ にはその構造体自体の先頭アドレスが格納される。

【０１３４】

なお、この押鍵イベントはリアルタイム処理装置３００が動作した後最初に受信されたイベントであるため、既存のイベントとリンクされることは無いが、以後受信されるイベントは、構造体としてメモリに記憶される際に、順次既存の受信イベントとリンクされる。このリンクを以後「記録フレーム」という。また、ステップＳＰ５においては、この押鍵イベントに対応する時間監視キーがメモリ内に設けられる。その値は、時刻t_1 と、「押鍵開始時刻から打弦時刻までの最大時間」との合計値である。これにより、以後ステップＳＰ２が実行される毎に時間監視キーが参照され、これがタイムアップすると、ステップＳＰ３を介して、ダミーイベントが記録フレームに追加されることになる。

【０１３５】

次に処理がステップＳＰ６に進むと、完成したイベントが記録フレーム内に生じたか否かが判定される。ここで「完成したイベント」とは、リファレンス軌道を生成するに足るイベントである。上記時刻t_1 における押鍵イベントは、後のハンマーイベントと組合わせることによって初めてリファレンス軌道を生成できるものであるから、この時点では完成していない。従って、ここでは「ＮＯ」と判定され、処理がステップＳＰ１２に進む。以後、ステップＳＰ２，４，１２から成るループが繰り返されることになる。

【０１３６】

次に、図２６の時刻t_2 において、自動ピアノ１００でキーコードＫＣ１のハンマーイベントが発生すると、次にステップＳＰ４が実行された際に「ＹＥＳ」と判定され、処理がステップＳＰ５に進む。ステップＳＰ５においては、ハンマーイベントの内容が構造体としてメモリ内に記憶され、この構造体は記録フレームにリンクされる。これにより、キーコードＫＣ１については押鍵イベントとハンマーイベントが揃ったため、リファレンス軌道を求めることが可能になった。すなわち、両者は「完成したイベント」になった。

【０１３７】

従って、次に処理がステップＳＰ６に進むと「ＹＥＳ」と判定され、処理がステップＳＰ７に進む。ステップＳＰ７においては、完成したイベント内にダミーイベントを含むか否かが判定される。上記例においてはダミーイベントは含まれないため「ＮＯ」と判定され、処理がステップＳＰ１０に進む。ステップＳＰ１０においては、押鍵開始時刻、交差点あるいは加速度等についてリファレンス計算が行われ、その結果が構造体としてメモリ内に格納される。以後、リファレンス計算の結果は、記録フレームと同様に、構造体としてメモリ内に記憶されるとともに押鍵開始時刻の早い順にリンクされる。このようにリファレンス計算の結果（構造体）をリンクしたものを、以後「リファレンスフレーム」という。

【０１３８】

ところで、時刻t_1，t_2 におけるイベントに基づいて押鍵開始時刻を演算すると、例えば同図（ハ）に示す時刻t_{21} が得られる。この時刻t_{21} は、演算の完了した時点では既に過去の時刻であるから、時刻t_{21} を押鍵開始時刻とするようなリファレンス軌道を生成することは不可能である。また、検出された各イベントについてリファレンス計算することにより、検出時の各イベント間の時間間隔とリファレンス計算後の各イベント間の時間間隔が異なることがあり、最たる場合には、前述したように、後に検出されたイベントを先に検出されたイベントよりも早く再生する必要がある。従って、キーコードＫＣ１の押鍵開始時刻は、時刻t_{21} から所定の遅延時間ｔｄだけ遅らせた時刻t_{31} に設定される。すなわち、実際のリファレンスフレームにおいては、同図（ニ）に従って押鍵開始時刻等が設定される。

【０１３９】

次に、処理がステップＳＰ１２に進むと、リファレンスフレーム内の先頭にある構造体について、出力時刻が超過したか否かが判定される。この時点では未だ時刻t_{31} に達していないため、「ＮＯ」と判定され処理がステップＳＰ２に戻る。以下同様に、自動ピアノ１００において新たなイベントが生ずる毎に、その内容が記録フレームに追加され、「

完成したイベント」が生ずると、その内容に基づいて押鍵開始時刻、交差点あるいは加速度等についてリファレンス計算が行われ、その結果がリファレンスフレームに追加される。

【０１４０】

次に、時刻がt_{31} に達した後に処理がステップＳＰ１２に進むと、ここで「ＹＥＳ」と判定され、処理がステップＳＰ１３に進む。ステップＳＰ１３においては、出力時刻の超過したリファレンスフレーム内の構造体の内容に基づいて、リファレンス軌道が生成され、その内容が自動ピアノ２００のモーションコントローラ１１に供給される。リファレンス軌道がモーションコントローラ１１に供給されると、記録フレームおよびリファレンスフレーム内の対応するイベントデータは用済になるため、各々のフレームから削除される。削除されたイベントデータに係るメモリ領域は解放され、あらたなデータを格納することが可能になる。

【０１４１】

ところで、上述のようにステップＳＰ２において「時間監視キーがタイムアップした」、と判定されるとダミーイベントが追加されるのであるが、この場合の処理について詳述しておく。ダミーイベントが追加された場合は、基本的には、ダミーイベントが追加されなかった場合と同様の処理が行われる。すなわち、ダミーイベントに基づいてリファレンス計算が行われる。しかし、ステップＳＰ３でダミーイベントが追加された直後のステップＳＰ４において、本来の（ダミーではない）イベントが受信されると、イベントが重複するおそれがある。このようなケースは稀ではあるが、本実施例においてはかかる場合も不具合が生じないように対応している。

【０１４２】

まず、ダミーイベントを含む「完成したイベント」が生じた場合、ステップＳＰ７において「ＹＥＳ」と判定され、処理がステップＳＰ８に進む。ステップＳＰ８では当該ダミーイベントを採用するか否かについて最終的な判断が行われる。すなわち、上述したように本来のイベントを受信した場合には、その受信したイベントを用いることが好適であるから、かかる場合には処理をステップＳＰ９に進め、ダミーイベントを記録フレームから削除することとしたものである。

【０１４３】

Ｃ－２．演奏再生の全体動作

次に、演奏再生時における実施例の全体動作を説明する。

まず、演奏者が自動ピアノ１００を演奏すると、この演奏による鍵イベントおよびハンマーイベントが自動ピアノ１００内のセンサＳＥ１，ＳＥ２，２５によって検出され、押鍵速度、あるいは離鍵速度とともにリアルタイム処理装置３００に供給される。リアルタイム処理装置３００においては、受信した各イベントの内容が記録フレームに追加され、イベントの完成とともにリファレンス計算が行われると、その計算結果がリファレンスフレームに追加される。

【０１４４】

そして、リファレンスフレーム内の各イベントについて出力時間が超過すると、リファレンス軌道が演算され、その結果が自動ピアノ２００のモーションコントローラ１１に供給される。まず、押鍵軌道データがモーションコントローラ１１に供給されると、この押鍵軌道データは位置制御データ（Ｘ）に変換される。すなわち、数３に示される軌道データがモーションコントローラ１１に供給され、モーションコントローラ１１は、数３のＸを時間経過とともに順次演算し、位置制御データ（Ｘ）を作成する。

【０１４５】

この位置制御データ（Ｘ）は、サーボコントローラ１２に供給され、これにより、数３のＸに対応した励磁電流がソレノイド５に供給される。この場合、サーボコントローラ１２は、ソレノイド５のフィードバック信号と位置制御データ（Ｘ）とを比較し、両者が一致するように励磁電流を制御するから、ソレノイド５のプランジャの突出量は、数３のＸに

対応したものとなる。したがって、鍵１は数３で示される直線軌道に従って押下されていき、リファレンスポイントＸｒにおいてリファレンス速度Ｖｒを有する運動を行う。これにより、自動ピアノ１００における打弦強度は、自動ピアノ２００においてリアルタイムに、かつ忠実に再現される。

【０１４６】

次に、リアルタイム処理装置３００は、リファレンスフレーム内の離鍵終了時刻データ、離鍵速度データに応じて、離鍵軌道を作成し、自動ピアノ２００のモーションコントローラ１１に供給する。モーションコントローラ１１は、上述の場合と同様にして、数６のＸＮを時間の経過に応じて順次演算し、位置制御データ（Ｘ）を作成し、サーボコントローラ１２に供給する。この結果、サーボコントローラ１２は数６に対応した励磁電流制御を行い、鍵１は数６で示される直線軌道に従ってレスト位置$Ｘ_0$に戻っていく。これにより、自動ピアノ２００における鍵１の運動は、離鍵リファレンスポイントＸｒＮにおいて離鍵リファレンス速度ＶｒＮを有するものとなり、自動ピアノ１００の離鍵状態が忠実に再現される。

【０１４７】

以上のように、本実施例によれば、イベントに時刻情報を含ませるとともにリファレンス軌道のイベントを所定時間遅延させるようにしたので、後に検出されたイベントを先に検出されたイベントよりも早く再生する必要がある場合など、検出時の各イベント間の時間間隔とリファレンス計算後の各イベント間の時間間隔が異なったとしても、リファレンス計算後の各イベント間の時間間隔に応じた再生を行うことができる。

また、本実施例によれば、イベントに時間情報を含ませるとともにリファレンス軌道のイベントを所定時間遅延させるようにしたので、ダミーイベントの有無によってイベント毎に処理時間が異なっても、リファレンス計算後のイベント間の時間間隔は保持される。

【０１４８】

なお、イベントを所定時間遅延させてイベント毎の処理時間を吸収する手法は、上述したダミーイベントの有無によるリファレンス計算以外にも適用可能であって、要するに、イベント毎に処理時間が異なるものであれば適用可能である。例えば、リファレンス計算において、押鍵イベントについては加速度の計算を行い、離鍵イベントについては加速度計算を行わない場合には、両者における計算時間は当然異なる筈である。また、リファレンス計算に直接関係の無い用途としては、例えばキーコードに応じて打弦速度を正規化するような処理（低音域と高音域の打弦速度は少し大きくし中音域の打弦速度は変更せずに出力すること）が挙げられる。

【０１４９】

ＩＶ．変形例

この発明においては、上述した各実施例に対し、以下に述べる種々の変形が可能である。

Ａ．サーボ制御の変形など

上述した各実施例においては、モーションコントローラ１１とサーボコントローラ１２によって位置サーボ制御を行っていたが、これに代えて、第１実施例では速度を指示する速度サーボ制御を行っても良い。すなわち、モーションコントローラ１１が押鍵（あるいは離鍵）開示時刻において初速度を指示し、サーボコントローラ１２は鍵１が与えられた速度を保つようにサーボ制御を行うようにしてもよい。また、第２実施例においては、加速度制御を行うようにしてもよい。すなわち、モーションコントローラ１１が押鍵開始時刻において初速度を指定し、以後は時間の経過とともに速度の変化分を順次指示する。そして、サーボコントローラ１２は、速度変化分を累算して、現時点の速度を算出し、鍵１の速度が累算速度に一致するようにサーボ制御を行う。

【０１５０】

Ｂ．推定した加速度情報の電子楽器への応用

第２実施例において推定した加速度ａは、電子ピアノの押鍵制御に限らず、電子楽器等の楽音制御に用いることができる。すなわち、加速度の推定は、押鍵速度情報と打弦速度情報がそれぞれ１以上あれば可能であるから、これらの情報を基に加速度を推定し、それを

楽音制御情報として用いることができる。例えば、楽音波形を複数記憶し、これらの波形を音の強さや、高さに応じて選択して読み出す電子楽器があるが、この電子楽器に、さらに加速度情報に応じた波形選択を行わせると、発生楽音の表情がいっそう豊かになという利点が得られる。

【０１５１】

Ｃ．リファレンスポイントＸｒにおける運動属性の変形

上述した各実施例においては、リファレンスポイントＸｒにおける鍵の速度や加速度を再現するようにしたが、鍵の運動を一意的に再現できるのであれば、その他の運動属性を再現するようにしてもよい。例えば、速度、加速度、力のいずれか、あるいは、これらの組み合わせを用いても良い。

【０１５２】

Ｄ．離鍵時の加速度軌道

上述した各実施例においては、離鍵時の軌道は全て直線軌道であったが、これを加速度軌道としてもよい。この場合には、離鍵速度を２カ所以上で測定するようにセンサを追加し、得られた速度の変化から加速度を推定するように構成すればよい。例えば、キーセンサ２５内のフォトセンサの数を増やし、演奏記録部３０がこれらのセンサの出力信号から２カ所以上の離鍵速度を検出するようにすればよい。

さらに、押鍵時と離鍵時の軌道を直線とするか放物線とするかは、任意に組み合わせることが可能である。

【０１５３】

Ｅ．開平演算態様

第２実施例における初速算出演算（数２０）および押鍵開始時刻算出演算（数１８、１９）は、開平演算が入るが、ソフトウエア上の演算速度が問題になる場合は、例えば、２分探索法を用いて、整数演算をしてもよい。

【０１５４】

Ｆ．弱音の再生

実際の押鍵操作においては、強打よりも弱打において放物線軌道をとる傾向が強く、直線軌道での再現では再現性が悪化してくる。特に、最弱打（ｐｐｐｐ）に近い押鍵を直線軌道で再現しようとすると、ピアノのアクション部の構造などの理由から、ハンマーへの力伝達が行われず不安定な打弦となることがある。そこで、第１実施例においては、弱音だけを検出して放物線軌道で再現するようにし、その他の音は直線軌道で再現するように構成してもよい。すなわち、再生前処理部３０において、打弦速度データを所定のしきい値で振り分け、弱音と判定された音に対しては、予め設定した加速度ａを用いて放物線軌道を算出し、これによって鍵を駆動する。

【０１５５】

Ｇ．打弦時刻、打弦速度の代用

例えば、特開平１－２３９５９４号では、押鍵速度から打弦速度を推定して記録するようにしているが、このように、打弦速度に代わるデータが記録されている場合は、前述した各実施例において、それを用いることも可能である。したがって、ＭＩＤＩ信号中のキーオン速度（キーベロシティ）のように、打弦速度に対応するデータが記録されている場合においては、これを用いて軌道演算を行うことができる。打弦時刻についても、これに代わるデータがあれば、代用することができる。要は、発音時刻や発音強度に関連したデータであれば、用いることができる。

【０１５６】

Ｈ．記録フレーム、リファレンスフレームの代用

第３実施例では構造体内のポインタを用いて各構造体をリンクし記録フレーム、およびリファレンスフレームを構成したが、各構造体をリンクする手段は構造体内のポインタに限られない。例えば、磁気ディスク記録に用いられているＦＡＴ（ファイル・アロケーション・テーブル）と同様な技術を用いることもできる。すなわち、メモリ内の所定領域において、タイムコード順にアドレス情報を順次配置しておき、このアドレス情報を参照する

ことによって各種データをアクセスしてもよい。

【０１５７】

Ｉ．ダミーイベント追加条件の変更

図６においては、ステップＳＰ１０６，１１０を省略してもよい。すなわち、ステップＳＰ１０５または１０９において「ＹＥＳ」と判定されると直ちにステップＳＰ１０７または１１１を実行するようにしてもよい。

【０１５８】

Ｊ．第１実施例においては、演奏データを外部に出力する（記録媒体に記録する）前に抜けイベントの修復を行っていたが、要はリファレンス計算を行う前に抜けイベントの修復を行えばよいのであって、センサ出力を何ら修正することなく記録媒体に記録し、これを読み出して自動演奏をする際に抜けイベントの修復を行うようにしてもよい。

また、同様に、第３実施例において、自動ピアノ１００あるいは自動ピアノ２００にリアルタイム処理装置３００を内蔵して自動ピアノ１００あるいは自動ピアノ２００のどちらか一方でイベントの修復処理とリファレンス計算を行うようにしてもよいし、自動ピアノ１００でイベントの修復処理のみを行い、自動ピアノ２００でリファレンス計算のみを行うようにしてもよい。

【０１５９】

Ｋ．ダミーイベントの内容を修正する際の態様は、上述した態様に限らず、どのような態様であってもよい。例えば、上述した第１実施例においては、離鍵／押鍵イベントが抜けた場合に、図２１の直線L_5に示すように、ダミーの離鍵速度は最大の離鍵速度に設定されるようにしたが、図２１の二点鎖線L_6に示すように、ダミーの押鍵イベントの押鍵時刻に向けて離鍵させたときの速度にしてもよい。

【０１６０】

Ｌ．第３実施例においては、入力されるイベントの時間間隔に応じてダミーイベントを追加するようにしたが、これに限らず、第１実施例と同様に、押鍵イベント、ハンマーイベント、離鍵イベントの入力される順序に応じて離鍵／押鍵イベント抜け、ハンマーイベント抜け等を検出してダミーイベントを追加するようにしてもよい。

【０１６１】

Ｍ．上記各実施例において開示したイベント修復技術は、リファレンス計算にのみ適用されるものではなく、他の種々の用途に用いて良いことは言うまでも無い。例えば、特開平１－２３９５９４号公報に開示されたもののように、イベントデータを用いて各種の処理を行う装置に対して、かかるイベント修復技術は適用可能である。

【０１６２】

Ｎ．第１実施例においては、ステップＳＰ１０７，１１１において既存のイベントと同一の発生時刻を有する各種のダミーイベントが演奏データに追加され、ステップＳＰ１０８，１１２においてこれらダミーイベントの発生時刻が修正されている。しかし、ステップＳＰ１０７，１１１において、ダミーイベントを追加する際に、その発生時刻等を最初から適切な値に設定し、ステップＳＰ１０７，１１１を省略してもよい。一方、第１実施例のようにダミーイベントの発生時刻を既存のイベントと同一時刻に設定しておくと、そのこと自体によって当該イベントは例外的なイベントであることを表現でき、ダミーイベントと通常のイベントとを区別するための特別のコードを設ける必要がなくなる。

【０１６３】

【発明の効果】

以上説明したように、請求項４に係る自動ピアノによれば、取得された限りの情報に基づいて正確な鍵の動作再生を行うことができる。また、請求項１，２，３に係る演奏データ加工装置によれば、欠落したイベントが存在すると判定されると、演奏データに新たなイベントデータを適宜追加することができる。

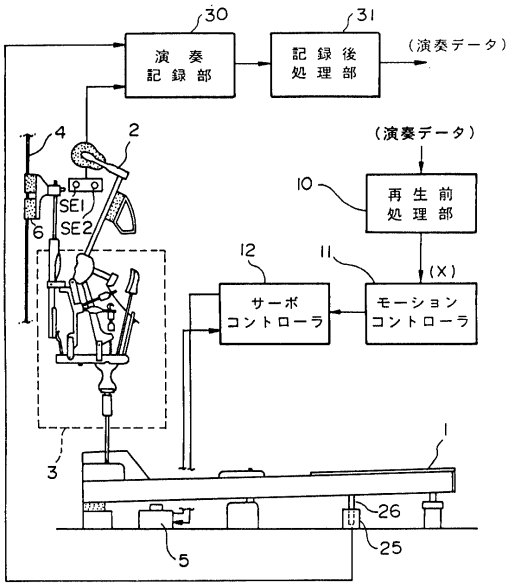
【図面の簡単な説明】

【図１】この発明の第１実施例の構成を示すブロック図である。

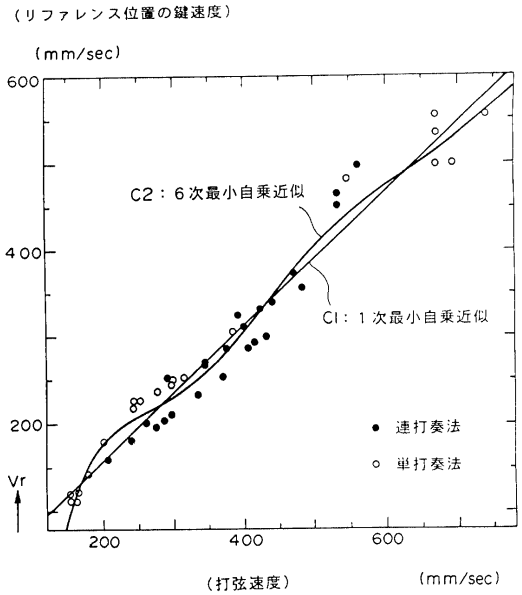
【図２】リファレンスポイントを９．５ｍｍに設定したときの鍵速度と打弦速度の関係を

示す図である。
【図３】リファレンス時間差Ｔｒと打弦速度との関係を示す図である。
【図４】図３を縮尺２倍にした図である。
【図５】図３を縮尺４倍にした図である。
【図６】記録後処理部３１のフローチャートである。
【図７】鍵の押鍵軌道（直線軌道）を示す図である。
【図８】数６で示される軌道を示す図である。
【図９】押鍵の後に発生した離鍵の軌道が交差する場合を示す図である。
【図１０】第２実施例で用いる放物線軌道を示す図である。
【図１１】第２実施例における押鍵開始時刻の算出を説明するための図である。
【図１２】第２実施例において、鍵駆動した場合の実験例を示す図である。
【図１３】図１２の一部を拡大した図である。
【図１４】ダンパーの離弦時刻推定方法を説明するための図である。
【図１５】ダンパー離弦時刻を再現した場合の放物線軌道を示す図である。
【図１６】第２実施例において加速度を固定した場合の実験結果を示す図である。
【図１７】図１６の一部を拡大した図である。
【図１８】第２実施例において初速度を固定した場合の実験結果を示す図である。
【図１９】図１８の一部を拡大した図である。
【図２０】一般的な自動ピアノの構成を示す概略図である。
【図２１】第３実施例の動作説明図である。
【図２２】第３実施例の動作説明図である。
【図２３】第３実施例の動作説明図である。
【図２４】第３実施例のデータ構造の説明図である。
【図２５】リアルタイム処理装置３００のメインルーチンのフローチャートである。
【図２６】リアルタイム処理装置３００のタイムチャートである。
【図２７】この発明の第３実施例の構成を示すブロック図である。
【図２８】リアルタイム処理装置３００のタイマ割込処理ルーチンのフローチャートである。
【符号の説明】
１・・・・鍵、３０・・・・演奏記録部（記憶手段）、３１・・・・記録後処理部（判定手段、イベント修復手段）、１０……再生前処理部（鍵駆動手段）、１１……モーションコントローラ（鍵駆動手段）、１２……サーボコントローラ（鍵駆動手段）、２５・・・・キーセンサ（イベント検出手段）、ＳＥ１，ＳＥ２・・・・センサ（イベント検出手段）、３００・・・・リアルタイム処理装置（イベントデータ加工手段，演奏データ加工装置，遅延手段，遅延時間設定手段）。

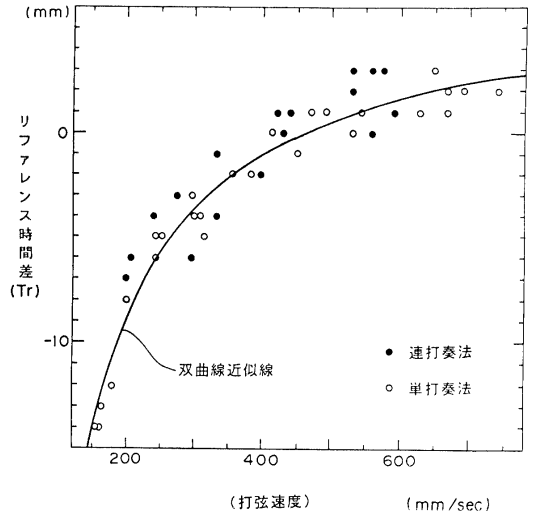
【図１】


30
31
演奏記録部
記録後処理部
(演奏データ)
(演奏データ)
4
2
10
再生前処理部
SE1
SE2
6
12
11
(X)
サーボコントローラ
モーションコントローラ
1
3
26
25
5

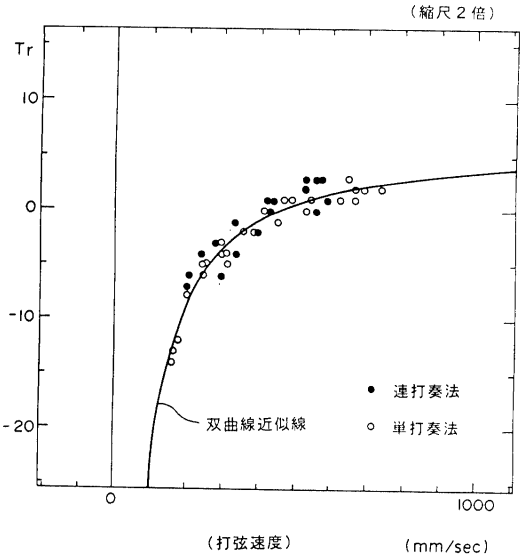
【図２】


(リファレンス位置の鍵速度)
(mm/sec)
600
400
200
Vr
C2：6次最小自乗近似
C1：1次最小自乗近似
● 連打奏法
○ 単打奏法
200
400
600
(打弦速度)
(mm/sec)

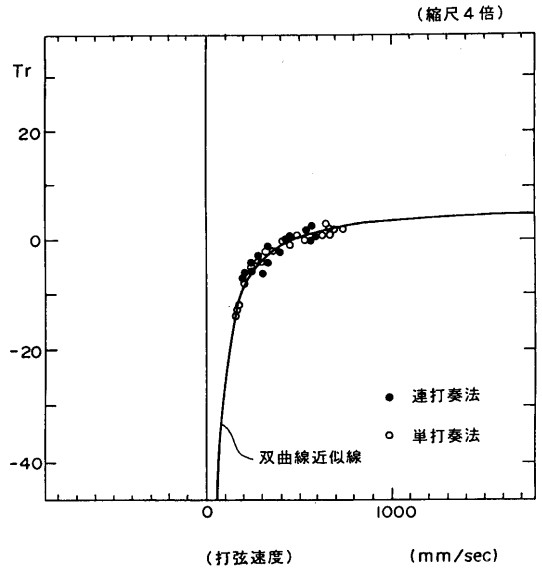
【図３】


(mm)
リファレンス時間差(Tr)
0
-10
双曲線近似線
● 連打奏法
○ 単打奏法
200
400
600
(打弦速度)
(mm/sec)

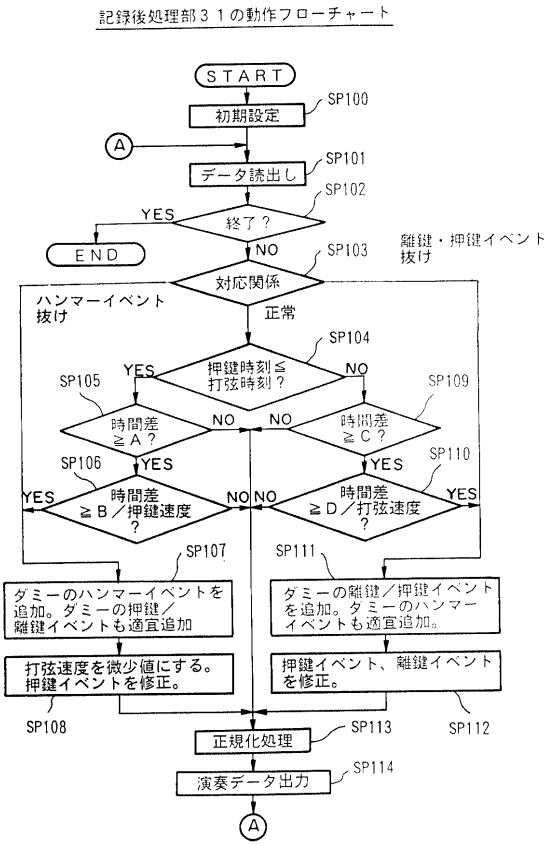
【図４】


(縮尺２倍)
Tr
10
0
-10
-20
双曲線近似線
● 連打奏法
○ 単打奏法
0
1000
(打弦速度)
(mm/sec)

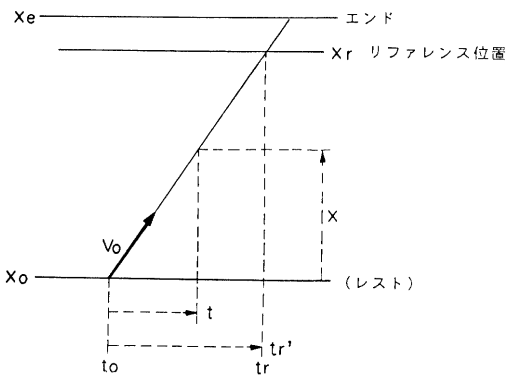
【図５】


（縮尺４倍）
Tr
20
0
-20
-40
0
1000
（打弦速度）
（mm/sec）
● 連打奏法
○ 単打奏法
双曲線近似線

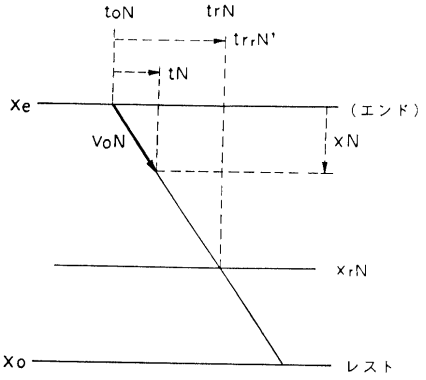
【図６】


記録後処理部３１の動作フローチャート
START
初期設定
SP100
A
データ読出し
SP101
終了？
SP102
YES
END
NO
対応関係
SP103
離鍵・押鍵イベント抜け
ハンマーイベント抜け
正常
押鍵時刻≦打弦時刻？
SP104
YES
NO
時間差≧A？
SP105
NO
時間差≧C？
SP109
YES
時間差≧B／押鍵速度？
SP106
YES
NO
時間差≧D／打弦速度？
SP110
YES
NO
ダミーのハンマーイベントを追加。ダミーの押鍵／離鍵イベントも適宜追加
SP107
打弦速度を微少値にする。押鍵イベントを修正。
SP108
ダミーの離鍵／押鍵イベントを追加。ダミーのハンマーイベントも適宜追加。
SP111
押鍵イベント、離鍵イベントを修正。
SP112
正規化処理
SP113
演奏データ出力
SP114
A

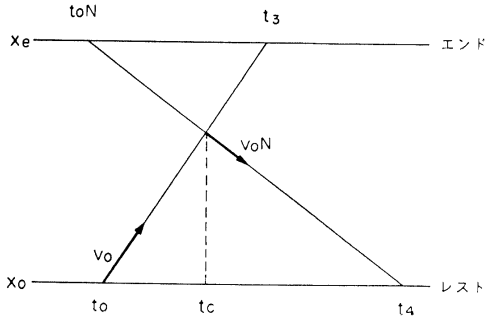
【図７】


Xe
エンド
Xr リファレンス位置
X
Vo
Xo
（レスト）
t
tr'
to
tr

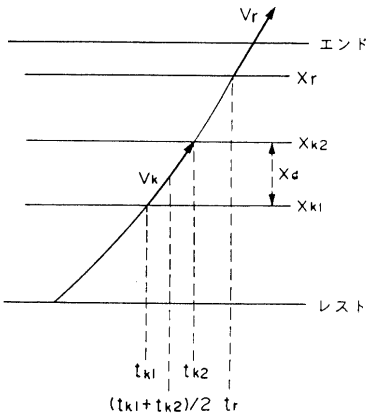
【図８】


toN
trN
trrN'
tN
Xe
（エンド）
VoN
xN
xrN
Xo
レスト

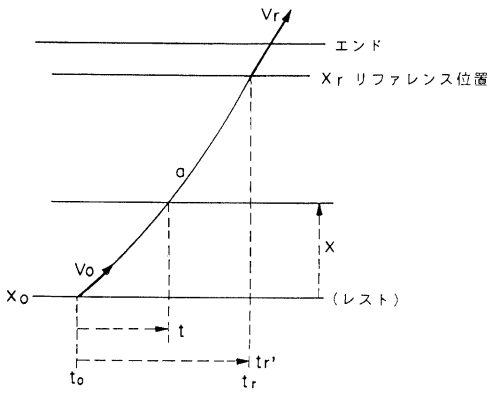
【図９】



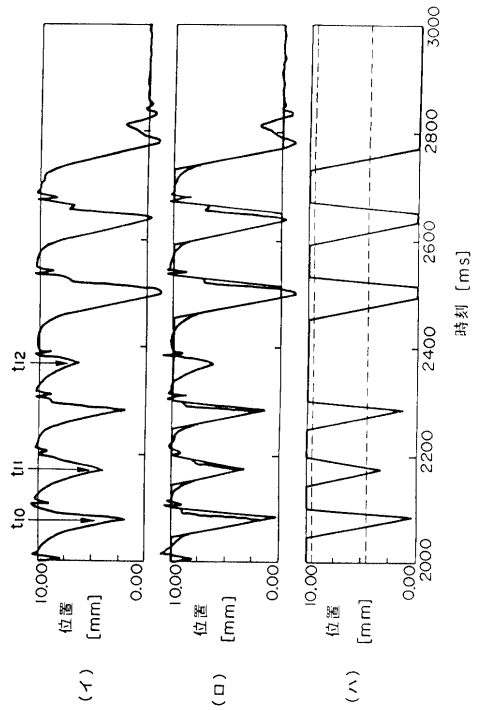
【図１０】



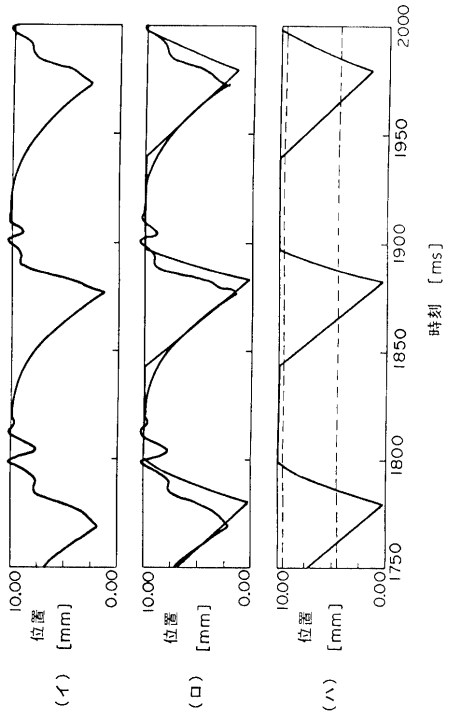
【図１１】



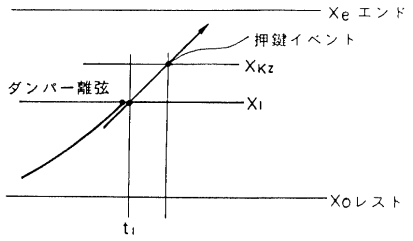
【図１２】



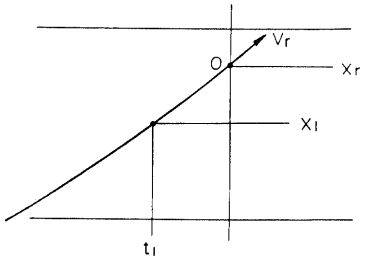
【図１３】



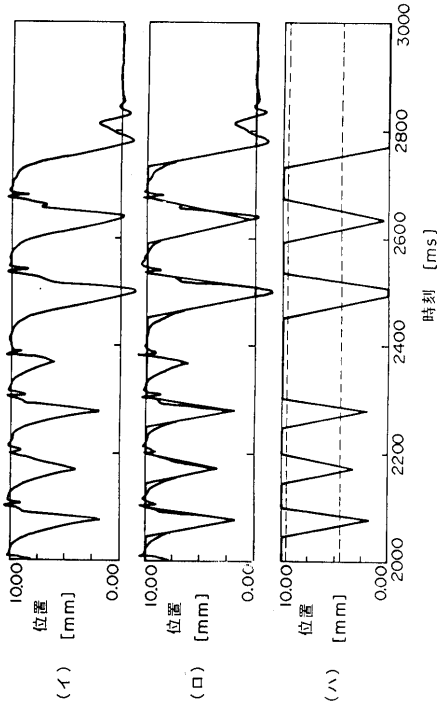
【図１４】



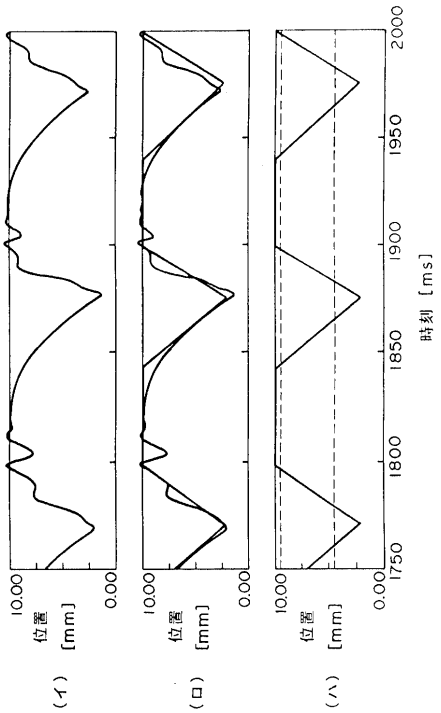
【図１５】



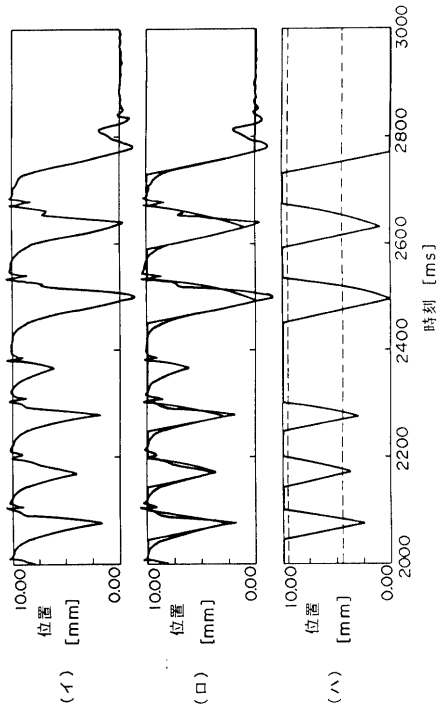
【図１６】



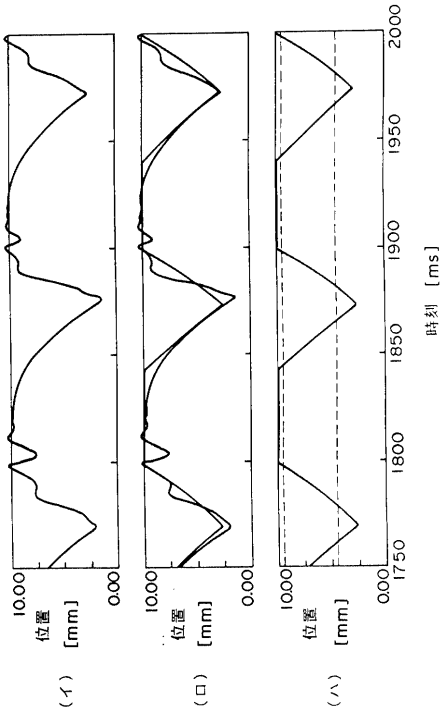
【図１７】



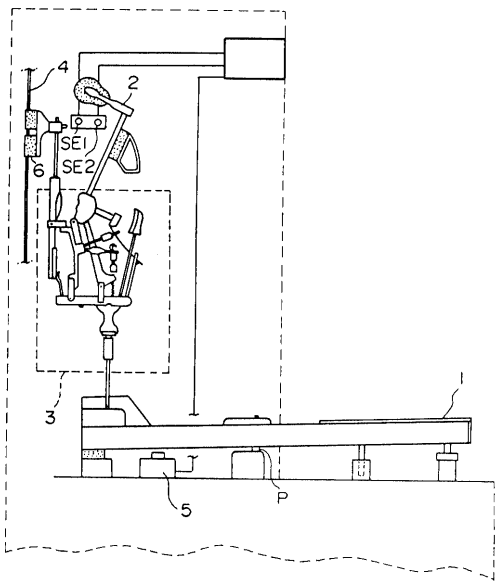
【図１８】



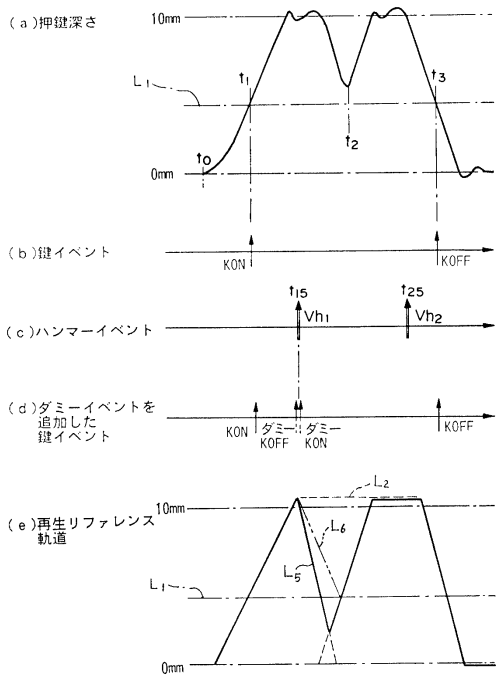
【図１９】



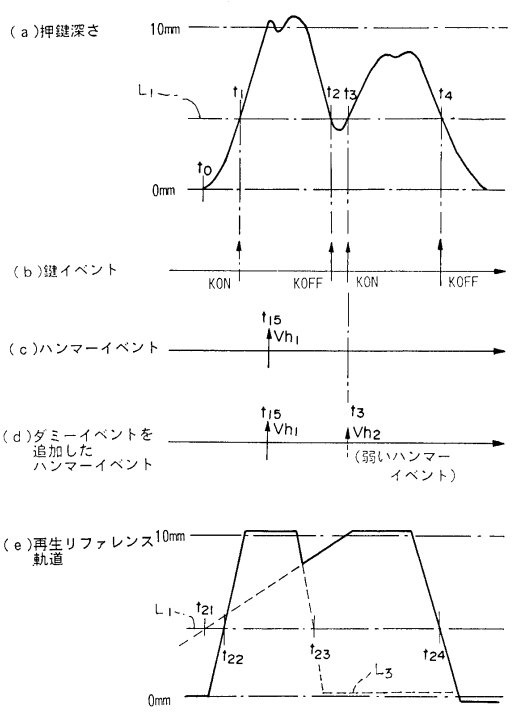
【図２０】



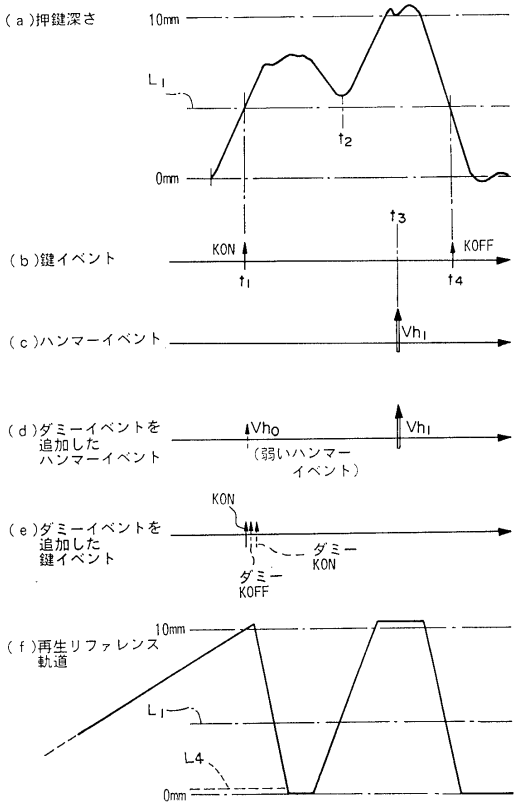
【図２１】



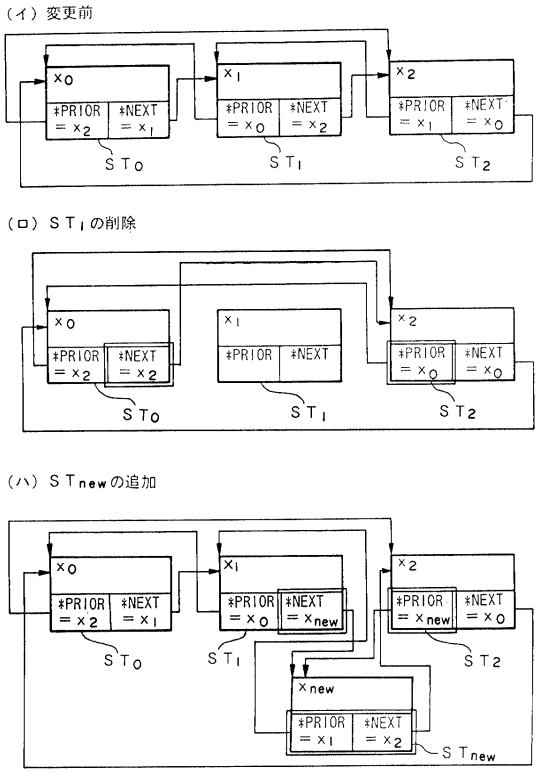
【図２２】



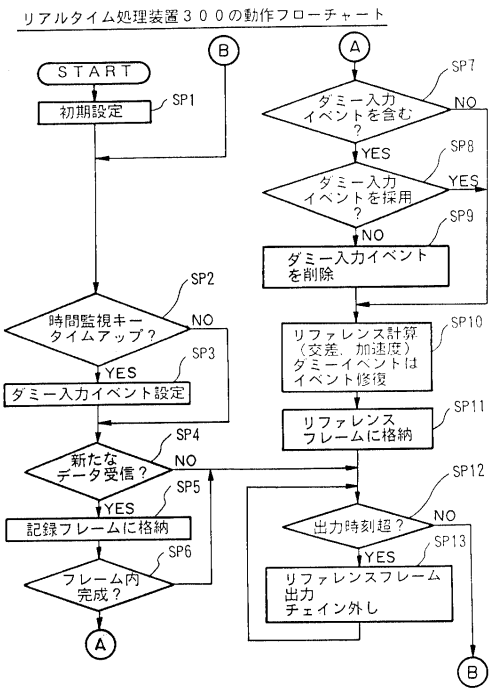
【図２３】



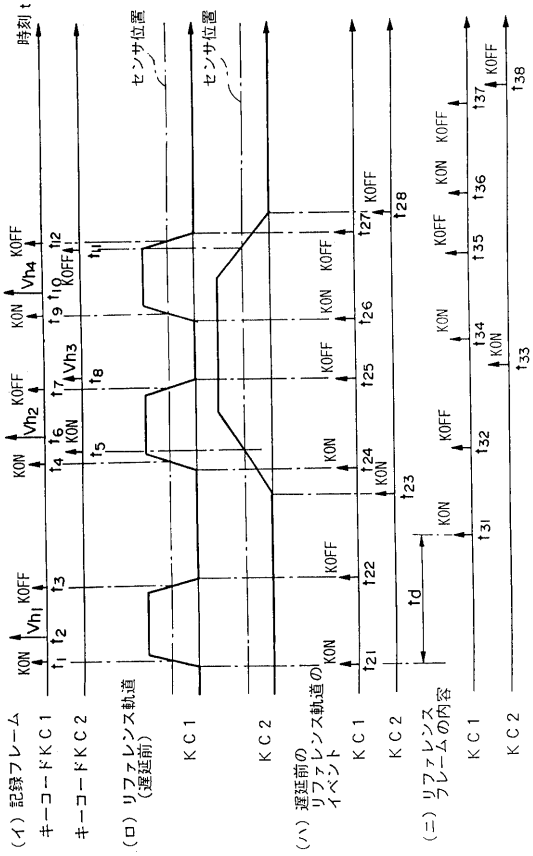
【図２４】



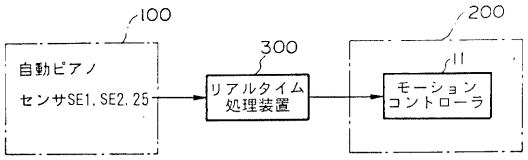
【図２５】



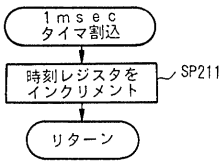
【図２６】


時刻 t
センサ位置
センサ位置
(イ) 記録フレーム
キーコードKC1
キーコードKC2
(ロ) リファレンス軌道
(遅延前)
(ハ) 遅延前の
リファレンス軌道の
イベント
(ニ) リファレンス
フレームの内容
KC1
KC2
KON
KOFF
Vh1
Vh2
Vh3
Vh4
td

【図２７】


100
自動ピアノ
センサSE1, SE2, 25
300
リアルタイム
処理装置
200
11
モーション
コントローラ

【図２８】


1msec
タイマ割込
時刻レジスタを
インクリメント
SP211
リターン

フロントページの続き

(72)発明者　磯崎　善政
静岡県浜松市中沢町１０番１号　ヤマハ株式会社内
(72)発明者　大場　保彦
静岡県浜松市中沢町１０番１号　ヤマハ株式会社内

審査官　板橋　通孝

(56)参考文献　特開平０５－１８１４６７（ＪＰ，Ａ）
特開平０２－２３０５９１（ＪＰ，Ａ）
特開平０３－２６６２８３（ＪＰ，Ａ）
特開平０１－２３９５９４（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.^{7}，ＤＢ名)
G10F　1/02
G10H　1/00 - 7/12