

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7409366号
(P7409366)

(45)発行日 令和6年1月9日(2024.1.9)

(24)登録日 令和5年12月25日(2023.12.25)

(51)国際特許分類 F I
G 1 0 H 1/00 (2006.01) G 1 0 H 1/00 1 0 2 Z

請求項の数 7 (全26頁)

(21)出願番号	特願2021-203446(P2021-203446)	(73)特許権者	000001443
(22)出願日	令和3年12月15日(2021.12.15)		カシオ計算機株式会社
(65)公開番号	特開2023-88608(P2023-88608A)		東京都渋谷区本町1丁目6番2号
(43)公開日	令和5年6月27日(2023.6.27)	(74)代理人	110004185
審査請求日	令和4年12月22日(2022.12.22)		インフォート弁理士法人
		(74)代理人	100121083
			弁理士 青木 宏義
		(74)代理人	100138391
			弁理士 天田 昌行
		(74)代理人	100074099
			弁理士 大菅 義之
		(74)代理人	100182936
			弁理士 矢野 直樹
		(72)発明者	吉野 順
			東京都羽村市栄町3丁目2番1号 カシ
			最終頁に続く

(54)【発明の名称】 自動演奏装置、自動演奏方法、プログラム、及び電子楽器

(57)【特許請求の範囲】

【請求項1】

発音させる音数を確率的に選択する音数選択手段と、
音域に対応するボイスンタイプを決定するボイスンタイプ決定手段と、
楽曲の調とコードにより決定されるスケールに基づく複数のボイスンパターンから、
前記選択された音数と、前記決定されたボイスンタイプとの組合せに対応するボイスン
グパターンを選択するボイスンパターン選択手段と、
選択された前記ボイスンパターンに基づきボイスンされたコードの発音を音源に指
示する発音指示手段と、
を備える自動演奏装置。

10

【請求項2】

前記組合せに対応するボイスンパターンが複数ある場合は、前記複数のボイスンパ
ターンからいずれか1つを確率的に選択する、請求項1に記載の自動演奏装置。

【請求項3】

前記ボイスンタイプ決定手段は、発音すべきコードのルートの音高に基づき前記ボイ
スンタイプを決定する、請求項1又は2に記載の自動演奏装置。

【請求項4】

楽音情報を送受信する通信手段を更に備え、
前記発音指示手段は、前記通信手段を介して、前記音源に前記コードの発音を指示する、
請求項1乃至3の何れか1項に記載の自動演奏装置。

20

【請求項 5】

請求項 1 乃至 3 の何れか 1 項に記載の自動演奏装置と前記音源とを備え、前記自動演奏装置の前記発音指示手段による前記選択されたボーシングパターンに基づきボーシングされたコードの発音の指示に基づいて、前記音源が自動コード伴奏を実施する電子楽器。

【請求項 6】

発音させる音数を選択し、

音域に対応するボーシングタイプを決定し、

楽曲の調とコードにより決定されるスケールに基づく複数のボーシングパターンから、前記選択された音数と、前記決定されたボーシングタイプとの組合せに対応するボーシングパターンを選択し、

選択された前記ボーシングパターンに基づきボーシングされたコードの発音を音源に指示する、

自動演奏方法。

【請求項 7】

発音させる音数を選択し、

音域に対応するボーシングタイプを決定し、

楽曲の調とコードにより決定されるスケールに基づく複数のボーシングパターンから、前記選択された音数と、前記決定されたボーシングタイプとの組合せに対応するボーシングパターンを選択し、

選択された前記ボーシングパターンに基づきボーシングされたコードの発音を音源に指示する、

処理をコンピュータに実行させるためのプログラム。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、コードを自動演奏する自動演奏装置、自動演奏方法、プログラム、及び電子楽器に関する。

【背景技術】**【0002】**

ジャズのピアノパートやギターパート等の演奏においては、同じ構成音やコード音を繰り返し演奏する場合でも、毎回異なるタイミングで演奏をする場合がある。

また、ジャズの演奏において、演奏する楽曲のコード（和音）のボーシング（構成音）をコード譜通りに行うのではなく、ジャズ特有の緊張感をもったテンションノートを含むコード（和音）を用いるのが一般的である。テンションノートとは、長調と短調による音楽の和声と共に使用される非和声音のうち、和音の響きに緊張感を与え、かつ和音進行を阻害しない構成音をいう。テンションノートは、コード種により一律に決まるわけではない。

【0003】

電子楽器による自動演奏において、指定されたコードネームのテンションノートを用いた自動演奏を実現して粋なサウンドの演奏データを作成可能とするために、次のような従来技術が知られている（例えば特許文献 1）。自動演奏時に、ルート音データ、タイプデータ及びアヴェイラブルノートスケールデータを 1 組に含んでなる和音データが順次指定され、このアヴェイラブルノートスケールデータが参照される。

【先行技術文献】**【特許文献】****【0004】**

【文献】特開平 10 - 78779 号公報

【発明の概要】**【発明が解決しようとする課題】****【0005】**

10

20

30

40

50

しかし、従来の予め決まった演奏をデータ化した自動伴奏では、テンションノートの制御が難しく、音数が増えてテンションノートがメロディの邪魔をしてしまったり、毎回決まった演奏になってしまうなど、ジャズ等の音楽の特徴を再現することができなかった。

例えば特許文献 1 に記載のような従来技術をもってしても、アヴェイラブルノートスケールデータを含む、予め決まった和音データに基づいて自動演奏が実行されるにすぎず、同じ和音について演奏時の偶然性に基づいて演奏タイミングや小節内の音数、ボイスンク（音の構成）を微妙に変えて自動演奏させることができないという課題があった。

【 0 0 0 6 】

そこで、本発明は、演奏者による楽器のライブ演奏におけるタイミング及びボイスンクを表現可能な、自然な自動コード伴奏を実現することを目的とする。

10

【課題を解決するための手段】

【 0 0 0 7 】

態様の一例の自動演奏装置は、発音させる音数を確率的に選択する音数選択手段と、音域に対応するボイスンクタイプを決定するボイスンクタイプ決定手段と、楽曲の調とコードにより決定されるスケールに基づく複数のボイスンクパターンから、選択された音数と、決定されたボイスンクタイプとの組合せに対応するボイスンクパターンを選択するボイスンクパターン選択手段と、選択されたボイスンクパターンに基づきボイスンクされたコードの発音を音源に指示する発音指示手段と、を備える。

【発明の効果】

【 0 0 0 8 】

20

本発明によれば、確率的に選択された音数とボイスンクタイプとの組合せからボイスンクパターンを選択して、ボイスンクされたコードの発音を音源に指示するので、演奏者による楽器のライブ演奏におけるボイスンクを表現可能な、自然な自動コード伴奏の実現が可能となる。

【図面の簡単な説明】

【 0 0 0 9 】

【図 1】電子楽器の実施形態のハードウェア構成例を示す図である。

【図 2】自動演奏装置の自動コード伴奏処理の例を示すフローチャートである。

【図 3】タイミングデータ生成処理の詳細例を示すフローチャートである。

【図 4】タイミングデータ生成処理で使用される頻度テーブルのデータ構成例を示す図である。

30

【図 5】ノートタイミングテーブルのデータ構成例とそのデータによるノートタイミングの楽譜表現を示す図である。

【図 6】アンティシペーション・コード取得処理の詳細例を示すフローチャートである。

【図 7】アンティシペーション・コード取得処理の説明図である。

【図 8】ボイスンク処理の詳細例を示すフローチャートである。

【図 9】コード進行データ、スケール決定テーブル、及びスケール毎のボイスンクテーブルのデータ構成例を示す図である。

【図 10】ボイスンク処理で使用される頻度テーブルのデータ構成例を示す図である。

【図 11】スケールの例とそのスケールにおけるボイスンクのバリエーションの例の楽譜表現を示す図である。

40

【図 12】自動演奏装置と電子楽器が個別に動作する他の実施形態の接続形態を示す図である。

【図 13】自動演奏装置と電子楽器が個別に動作する他の実施形態における自動演奏装置のハードウェア構成例を示す図である。

【発明を実施するための形態】

【 0 0 1 0 】

以下、本発明を実施するための形態について図面を参照しながら詳細に説明する。図 1 は、電子楽器の一例である電子鍵盤楽器の実施形態のハードウェア構成例を示す図である。図 1 において、電子鍵盤楽器 100 は、例えば電子ピアノとして実現され、CPU（中

50

中央演算処理装置) 101、ROM (リードオンリーメモリ) 102、RAM (ランダムアクセスメモリ) 103、鍵盤部 104、スイッチ部 105、及び音源 L S I 106 を備え、それらがシステムバス 108 によって相互に接続された構成を有する。また、音源 L S I 106 の出力はサウンドシステム 107 に入力する。

【0011】

この電子鍵盤楽器 100 は、ピアノパートを自動コード伴奏する自動演奏装置の機能を備える。そして、この電子鍵盤楽器 100 の自動演奏装置は、例えばジャズの自動ピアノ伴奏の発音データを、プログラムされたデータを単に再生するのではなく、ある一定の音楽的ルールの範囲内でアルゴリズムにより自動生成することができる。

【0012】

CPU 101 は、RAM 103 を作業用メモリとして使用しながら ROM 102 に記憶された制御プログラムを RAM 103 にロードして実行することにより、図 1 の電子鍵盤楽器 100 の制御動作を実行する。特に、CPU 101 は、後述するフローチャートによって示される制御プログラムを ROM 102 から RAM 103 にロードして実行することにより、ピアノパートを自動コード伴奏する制御動作を実行する。

【0013】

鍵盤部 104 は、複数の演奏操作子としての各鍵の押鍵又は離鍵操作を検出し、CPU 101 に通知する。CPU 101 は、後述するピアノパートの自動コード伴奏のための制御動作のほかに、鍵盤部 104 から通知された押鍵又は離鍵操作の検出通知に基づいて、演奏者による鍵盤演奏に対応する楽音の発音又は消音を制御するための発音指示データを生成する処理を実行する。CPU 101 は、生成した発音指示データを、音源 L S I 106 に通知する。

【0014】

スイッチ部 105 は、演奏者による各種スイッチの操作を検出し、CPU 101 に通知する。

【0015】

音源 L S I 106 は、楽音発生のための大規模集積回路である。音源 L S I 106 は、CPU 101 から入力する発音指示データに基づいて、デジタル楽音波形データを生成し、サウンドシステム 107 に出力する。サウンドシステム 107 は、音源 L S I 106 から入力したデジタル楽音波形データをアナログ楽音波形信号に変換した後、そのアナログ楽音波形信号を内蔵のアンプで増幅して内蔵のスピーカから放音する。

【0016】

上記構成を有する電子鍵盤楽器 100 の自動演奏装置の実施形態 (以下「本自動演奏装置」と記載) によるピアノパートの自動コード伴奏処理の詳細について、以下に説明する。図 2 は、本自動演奏装置の自動コード伴奏処理の例を示すフローチャートである。この処理は、図 1 の CPU 101 が、ROM 102 に記憶されたピアノパートの自動コード伴奏の制御処理のプログラムを RAM 103 にロードして実行する処理である。

【0017】

演奏者が図 1 のスイッチ部 105 を操作して自動伴奏のジャンル (例えば「ジャズ」) やテンポを選択した後、スイッチ部 105 にある特には図示しない自動伴奏のスタートスイッチを押すと、CPU 101 は、図 2 のフローチャートで例示される自動コード伴奏処理をスタートする。

【0018】

まず、CPU 101 は、カウンタリセット処理を実行する (ステップ S 201)。具体的には、CPU 101 は、ピアノパートの自動コード伴奏の開始時からの小節数を表す RAM 103 に記憶されている小節カウンタ変数値を、ピアノパートの自動コード演奏の第 1 小節を示す値 (例えば「1」) にリセットする。また、CPU 101 は、小節内の拍数 (拍位置) を示す RAM 103 に記憶されている拍カウンタ変数値を、1 拍目を示す値 (例えば「1」) にリセットする。次に、自動演奏装置による自動ピアノ伴奏の制御は、RAM 103 に記憶される t i c k 変数の値 (以下この変数の値を「t i c k 変数値」と記

10

20

30

40

50

載)を単位として進行する。図2のROM102内には、自動コード伴奏の時間分解能を示すTime Division定数(以下この定数の値を「Time Division定数値」と呼ぶ)が予め設定されており、このTime Division定数値は4分音符の分解能を示している。この値が例えば128ならば、4分音符は「128×tick変数値」の時間長を有する。ここで、1tickが実際に何秒になるかは、自動コード伴奏のピアノパートに対して指定されるテンポによって異なる。今、ユーザ設定に従ってRAM103上のTempo変数に設定される値を「Tempo変数値[ビート/分]」とすれば、1tickの秒数(以下「tick秒数値」と記載)は、下記(1)式により算出される。

【0019】

$$\text{tick秒数値} = 60 / \text{Tempo変数値} / \text{Time Division変数値} \quad \dots (1)$$

【0020】

そこで、図2のステップS201のカウンタリセット処理において、CPU101はまず、上記(1)式に対応する演算処理により、tick秒数値を算出し、RAM103上の「tick秒変数」に記憶させる。なお、Tempo変数値としては、初期状態では図2のROM102の定数から読み出した所定の値、例えば60[ビート/秒]が、初期設定されてもよい。或いは、Tempo変数が不揮発性メモリに記憶され、電子鍵盤楽器100の電源の再投入時に、前回終了時のTempo変数値がそのまま保持されていてもよい。

【0021】

次に、CPU101は、図2のステップS201のカウンタリセット処理において、まず、RAM103上のtick変数値を0にリセットする。その後、内蔵する特には図示しないタイマのハードウェアに対して、上述のようにして算出されRAM103上のtick秒変数に保存されているtick秒数値によるタイマ割込みを設定する。この結果、タイマにおいて上記tick秒数値の秒数が経過する毎に、割込み(以下「tick割込み」と記載)が発生することになる。

【0022】

演奏者が、ピアノパートの自動コード伴奏の途中で、図1のスイッチ部105を操作して自動コード伴奏のテンポを変更した場合、CPU101は、ステップS201のカウンタリセット処理と同様にして、RAM103上のTempo変数値に再設定されたTempo変数値を用いて前述した(1)式に対応する演算処理を再度実行することにより、tick秒数値を算出する。その後、CPU101は、内蔵のタイマのハードウェアに対して、新たに算出したtick秒数値によるタイマ割込みを設定する。この結果、タイマにおいて新たに設定されたtick秒数値の秒数が経過する毎に、tick割込みが発生することになる。

【0023】

CPU101は、ステップS201のカウンタリセット処理の後、ステップS202～S211の一連の処理をループ処理として繰り返し実行する。このループ処理は、ステップS210で、自動コード伴奏データがなくなった又は演奏者が図1のスイッチ部105の特には図示しないスイッチにより自動ピアノ伴奏の終了を指示したと判定されるまで、繰り返し実行される。

【0024】

CPU101は、上記ループ処理内の、ステップS211のカウンタ更新処理において、タイマから新たなtick割込み要求が発生している場合には、tick割込み処理によりRAM103上のtickカウンタ変数値をカウントアップする。その後、CPU101は、tick割込みを解除する。CPU101は、tick割込み要求が発生していない場合には、tick割込み処理によるtickカウンタ変数値はカウントアップせずに、そのままステップS211のカウンタ更新処理を終了する。この結果、tickカウンタ変数値は、演奏者が設定したTempo変数値に対応して算出されているtick秒

10

20

30

40

50

数値の秒数毎に、カウントアップされてゆくことになる。

【0025】

CPU101は、ステップS211でtick秒数値の秒数毎にカウントアップされる上記tickカウンタ変数値を基準として、自動コード伴奏の進行を制御する。以下、このtickカウンタ変数値=1を単位とするテンポに同期した時間単位を[tick]と記載する。前述したように、4分音符の分解能を示すTimeDivision定数値が例えば128ならば、4分音符は128[tick]の時間長を有する。従って、自動コード伴奏されるピアノパートが例えば4拍子であれば、1拍=128[tick]となり、1小節=128[tick]×4拍=512[tick]となる。CPU101は、上記ループ処理のステップS211のカウンタ更新処理において、例えば4拍子のピアノパートが選択されている場合、tickカウンタ変数値が128の倍数に更新される毎にRAM103に記憶されている拍カウンタ変数値を1 2 3 4 1 2 3・・・と、1から4の間でループさせて更新する。また、CPU101は、ステップS211のカウンタ更新処理において、上記拍カウンタ変数値が変化するタイミングで、各拍の先頭からのtick時間をカウントする拍内tickカウンタ変数値を0にリセットする。更に、CPU101は、ステップS211のカウンタ更新処理において、上記拍カウンタ変数値が4から1に変化するタイミングで、RAM103に記憶されている小節カウンタ変数値を+1ずつカウントアップしてゆく。この小節カウンタ変数値はピアノパートの自動コード伴奏の開始時からの小節数を表しており、拍カウンタ変数値は小節カウンタ変数値が表す各小節内での拍数(拍位置)を表していることになる。また、拍内tickカウンタ変数値の値が、0から63(=128÷2-1)ならば表拍、64から127ならば裏拍のタイミングを示す。この値は、後述する図6のフローチャートで例示されるアンティシペーション・コード取得処理のステップS602で判定される。

【0026】

CPU101は、ループ処理として上記ステップS211を繰り返し実行してtickカウンタ変数値、拍内tickカウンタ変数値、拍カウンタ変数値、及び小節カウンタ変数値を更新しながら、以下に説明するステップS202からS210の一連の制御処理を実行する。

【0027】

図2のステップS202からS210の一連の制御処理の詳細について、以下に説明する。まず、CPU101は、現在のタイミングが小節の先頭のタイミングかを判定する(ステップS202)。具体的には、CPU101は、RAM103に記憶されている小節カウンタ変数値が、ステップS202を前回実行したときと今回実行したときとで変化している(+1されている)か否かを判定する。

【0028】

ステップS202の判定がYESならば、CPU101は、タイミングデータ生成処理を実行する(ステップS203)。この処理において、CPU101は、更新された小節カウンタ変数値が示す新たな1小節分のコードの発音タイミングを示すノートタイミングテーブルデータを生成し、RAM103に記憶させる。また、CPU101は、生成されたノートタイミングデータの各発音タイミングに対応する更新された小節カウンタ変数値が示す新たな1小節分の各自動コード伴奏データを、例えばROM102からRAM103に読み込む。自動コード伴奏データは例えば、コード及びキーを少なくとも含む。この処理の詳細は、図3のフローチャートを用いて後述する。ステップS202の判定がNOならば、CPU101は、ステップS203のタイミングデータ生成処理は実行せずにスキップする。

【0029】

次に、CPU101は、現在のタイミングがノートオフのタイミングであるか否かを判定する(ステップS204)。具体的には、CPU101は、RAM103に記憶されている現在の拍カウンタ変数値及び拍内tickカウンタ変数値が、ステップS203でRAM103に記憶したノートタイミングデータの何れかのコード消音タイミングの拍番号

10

20

30

40

50

及び [t i c k] 時間と一致するか否かを判定する。この場合の何れかのコード消音タイミングの拍番号は、後述する図 5 (c) 又は (e) に例示されるノートタイミングデータにおいて、0 でない「 G a t e 」項目値が設定されているタイミングが含まれる何れかの「拍」項目値である。また、上述の何れかのコード消音タイミングの [t i c k] 時間は、その 0 でない「 G a t e 」項目値が設定されているタイミングの「 T i c k 」項目値と当該「 G a t e 」項目値とを加算して得られる [t i c k] 時間値である。

【 0 0 3 0 】

ステップ S 2 0 4 の判定が Y E S になったら、 C P U 1 0 1 は、ノートオフ処理を実行する (ステップ S 2 0 5) 。具体的には、 C P U 1 0 1 は、ステップ S 2 0 4 で判定したタイミングに対応して後述するステップ S 2 0 8 のボイシング処理において R A M 1 0 3 に記憶されているボイシングテーブルデータが示すボイス群の消音を、音源 L S I 5 0 8 に指示する。

10

【 0 0 3 1 】

ステップ S 2 0 4 の判定が N O ならば、 C P U 1 0 1 は、現在のタイミングがノートオンのタイミングであるか否かを判定する (ステップ S 2 0 6) 。具体的には、 C P U 1 0 1 は、 R A M 1 0 3 に記憶されている現在の拍カウンタ変数値及び拍内 t i c k カウンタ変数値が、ステップ S 2 0 3 で R A M 1 0 3 に記憶したノートタイミングテーブルの何れかのコード発音タイミングの拍番号及び [t i c k] 時間と一致するか否かを判定する。この場合の何れかのコード発音タイミングの拍番号は、後述する図 5 (a) 又は (c) に例示されるノートタイミングテーブルにおいて、0 でない「 G a t e 」項目値が設定されているタイミングが含まれる何れかの「拍」項目値である。また、上述の何れかのコード発音タイミングの [t i c k] 時間は、その 0 でない「 G a t e 」項目値が設定されているタイミングの「 T i c k 」項目値である。

20

【 0 0 3 2 】

ステップ S 2 0 6 の判定が Y E S ならば、 C P U 1 0 1 は、アンティシペーション・コード取得処理を実行する (ステップ S 2 0 7) 。この処理の詳細は、図 6 に例示されるフローチャートを用いて後述する。

【 0 0 3 3 】

続いて、 C P U 1 0 1 は、ボイシング処理を実行する (ステップ S 2 0 8) 。この処理において、 C P U 1 0 1 は、 R A M 1 0 3 に記憶されている現在の小節の自動コード伴奏データから抽出した今回のノートオンに対応するコード及びキーに対してボイシングテーブルデータを決定し、 R A M 1 0 3 のノートオンエリアに記憶する。 R A M 1 0 3 に記憶されている現在の小節の自動コード伴奏データは、後に詳述するステップ S 2 0 3 のタイミングデータ生成処理にて読み込まれる。ステップ S 2 0 8 のボイシング処理の詳細は、図 8 に例示されるフローチャートを用いて後述する。

30

【 0 0 3 4 】

ステップ S 2 0 8 の処理の後、 C P U 1 0 1 は、ノートオン処理を実行する (ステップ S 2 0 9) 。この処理において、 C P U 1 0 1 は、ステップ S 2 0 7 のアンティシペーションのタイミングの発音、及びステップ S 2 0 8 のボイシング処理において R A M 1 0 3 に記憶されたボイシングテーブルデータが示すボイス群の各ボイスに対応するノートナンバーの楽音の発音を、音源 L S I 5 0 8 に指示する。このとき各ノートナンバーと共に音源 L S I 5 0 8 に指定されるペロシティは、ステップ S 2 0 6 で判定されたノートオンタイミングに対応して現在の小節のノートタイミングデータに記憶されている「 V e l o c i t y 」項目値である。ステップ S 2 0 9 の処理を実行する C P U 1 0 1 は、発音指示手段として動作する。

40

【 0 0 3 5 】

ステップ S 2 0 6 の判定が N O の場合、又はステップ S 2 0 9 の処理の後、 C P U 1 0 1 は、 R O M 1 0 2 等から読み込むべき自動コード伴奏データがまだあり、かつ演奏者が図 1 のスイッチ部 1 0 5 の特には図示しないスイッチにより自動ピアノ伴奏の終了を指示していないか否かを判定する (ステップ S 2 1 0) 。

50

【 0 0 3 6 】

ステップ S 2 1 0 の判定が Y E S ならば、C P U 1 0 1 は、ステップ S 2 1 1 で前述したカウンタ更新処理を実行した後、ステップ S 2 0 2 の処理に戻ってループ処理を継続する。ステップ S 2 1 0 の判定が N O になったら、C P U 1 0 1 は、図 2 のフローチャートで例示される自動コード伴奏処理を終了する。

【 0 0 3 7 】

図 3 は、図 2 のステップ S 2 0 3 のタイミングデータ生成処理の詳細例を示すフローチャートである。この処理において、C P U 1 0 1 は、ステップ S 2 0 2 で判定された小節の先頭のタイミングごとに、新たに更新された現在の小節内で発音するノートタイミングとゲートタイムを決定する。この場合、C P U 1 0 1 は、当該小節内でのコードの発音回数(タイミングタイプ)と、各コードをどのようなタイミングで発音させるかを指定するノートタイミングテーブルを、確率的に決定する。

10

【 0 0 3 8 】

図 3 のフローチャートにおいて、C P U 1 0 1 はまず、R A M 1 0 3 上で新たに更新されている小節カウンタ変数値に対応する小節の 1 小節分の自動コード伴奏データを、例えば R O M 1 0 2 から取得し、R A M 1 0 3 に記憶する(ステップ S 3 0 1)。1 小節分の自動コード伴奏データは例えば、1 つの組がコードを少なくとも含む 0 組以上のデータ組を含む。その小節において発音されるべきコードが無い場合には、そのデータ組は 0 組となる。なお、演奏者は、図 1 のスイッチ部 1 0 5 内の特には図示しない選択スイッチにより、自動コード伴奏データの楽曲を予め選択することができる。それにより、自動コード伴奏データの楽曲のキーと後述するテンポ範囲が決定される。

20

【 0 0 3 9 】

次に、C P U 1 0 1 は、例えば図 1 の R O M 1 0 2 に記憶されているタイミングタイプ選択用頻度テーブルを参照することにより、タイミングタイプを確率的に決定する(ステップ S 3 0 2)。タイミングタイプは、1 小節中でのコードの発音回数を指定するデータである。即ち、ステップ S 3 0 2 では、現在の小節で発音されるコードの発音回数が確率的に決定されることになる。

【 0 0 4 0 】

図 4 (a) は、ステップ S 3 0 2 の処理を実現するために図 1 の R O M 1 0 2 に記憶されるタイミングタイプ選択用頻度テーブルのデータ構成例を示す図である。図 4 (a) に例示される「Type 0」、「Type 1」、「Type 2」、「Type 3」、及び「Type C」は夫々、以下の意味を有するタイミングタイプを示している。なお、図 4 (a) 及び以下の説明では、「タイミングタイプ」を「Type」と略記する場合がある。

30

Type 0 : 1 小節中で 0 回のコードの発音を指示

Type 1 : 1 小節中で 1 回のコードの発音を指示

Type 2 : 1 小節中で 2 回のコードの発音を指示

Type 3 : 1 小節中で 3 回のコードの発音を指示

Type C : 1 小節中での先頭及びコードの変更毎にコードの発音を指示

【 0 0 4 1 】

図 4 (a) に例示されるタイミングタイプ選択用頻度テーブルの左端列に登録される「Ballad」、「Slow」、「Mid」、「Fast」、及び「Very Fast」は夫々、自動コード伴奏データのテンポ範囲を示している。演奏者が自動コード伴奏の開始時に、図 1 のスイッチ部 1 0 5 内の特には図示しない選択スイッチによって、複数提供される自動コード伴奏楽曲のうちの所望の 1 つを選択した場合、選択された自動コード伴奏データには、上記「Ballad」、「Slow」、「Mid」、「Fast」、又は「Very Fast」の何れかのテンポ範囲が予め設定されている。ここで、「Ballad」は例えば 7 0 未満のテンポ範囲に対応する。「Slow」は例えば 7 0 以上 1 0 0 未満のテンポ範囲に対応する。「Mid」は例えば 1 0 0 以上 1 5 0 未満のテンポ範囲に対応する。「Fast」は例えば 1 5 0 以上 2 5 0 未満のテンポ範囲に対応する。そして、「Very Fast」は例えば 2 5 0 以上のテンポ範囲に対応する。

40

50

【 0 0 4 2 】

CPU101は、図3のステップS302において、ROM102に記憶されている図4(a)に例示されるタイミングタイプ選択用頻度テーブルを用いて、次のような制御処理を実行する。まず、図3のステップS301にてROM102から読み込まれる自動コード伴奏データに例えばテンポ範囲「Ballad」が設定されている場合には、CPU101は、図4(a)に例示されるタイミングタイプ選択用頻度テーブルにおいて、左端項目に「Ballad」が登録されている行のデータを参照する。この行には、上述のType0、Type1、Type2、Type3、又はTypeCの各タイミングタイプが夫々、0%、10%、20%、10%、又は60%の確率で選択されることを示す各頻度値が設定されている。これに対して、CPU101は、例えば1から100までの値域を有する任意の乱数値を発生する。そして、CPU101は例えば、生成した乱数値が1から10の乱数範囲（「Type1」の頻度値10%に対応）にあれば、タイミングタイプ「Type1」を選択する。又は、CPU101は例えば、生成した乱数値が11から30の乱数範囲（「Type2」の頻度値20%に対応）にあれば、タイミングタイプ「Type2」を選択する。又は、CPU101は例えば、生成した乱数値が31から40の乱数範囲（「Type3」の頻度値10%に対応）にあれば、タイミングタイプ「Type3」を選択する。又は、CPU101は例えば、生成した乱数値が41から100の乱数範囲（「TypeC」の頻度値60%に対応）にあれば、タイミングタイプ「TypeC」を選択する。なお、「Type0」は図4(a)の例では頻度値が0%なので乱数範囲は設定されず選択されない。勿論、「Type0」に対して、それが一定の確率（頻度値）で選択されるような乱数範囲が設定されてもよい。このようにして、CPU101は、「Type0」、「Type1」、「Type2」、「Type3」、及び「TypeC」の各タイミングタイプを夫々、タイミングタイプ選択用頻度テーブルの「Ballad」行に設定された0%、10%、20%、10%、及び60%の各確率で選択することができる。

10

20

【 0 0 4 3 】

図3のステップS301にてROM102から読み込まれる自動コード伴奏データに例えばテンポ範囲「Slow」、「Mid」、「Fast」、又は「Very Fast」が設定されている場合も、「Ballad」が設定されている上述の場合と同様にして、CPU101は例えば、図4(a)に例示される構成を有するタイミングタイプ選択用頻度テーブルにおいて、左端項目に「Slow」、「Mid」、「Fast」、又は「Very Fast」が登録されている何れかの行の各頻度値を参照する。次に、CPU101は、その行の「Type0」、「Type1」、「Type2」、「Type3」、又は「TypeC」の各タイミングタイプに対して夫々設定されている頻度値[%]に応じた1から100の範囲内の各乱数範囲を設定する。そして、CPU101は、1から100の範囲で乱数値を発生させ、生成された乱数値が上記各乱数範囲のどの範囲に入るかによって、「Type0」、「Type1」、「Type2」、「Type3」、又は「TypeC」の何れかのタイミングタイプを選択する。このようにして、CPU101は、「Type0」、「Type1」、「Type2」、「Type3」、及び「TypeC」の各タイミングタイプを夫々、タイミングタイプ選択用頻度テーブルの各テンポ範囲の行に設定された各頻度値に対応する確率で選択することができる。

30

40

【 0 0 4 4 】

「Ballad」のようにテンポがゆっくりとした自動コード伴奏では、小節内での発音はコード毎に行われる場合が多いので、「TypeC」が選択される頻度値（確率）が、例えば図4(a)の60%というように大きくなる。

【 0 0 4 5 】

音楽のコード伴奏の内容は、テンポに大きな影響を受ける。例えば、テンポが速い曲に多くの発音回数のコード（=多くのノートタイミング）が含まれてしまうと、せわしない演奏になり、自然な演奏からは乖離してしまい、非常に機械的な演奏になってしまう。同時に、スローなテンポの楽曲でも、発音回数が多い演奏は不自然なものになってしまう。

50

一方、一つの楽曲の中でも、相応しい変化は必要なため、各タイミングタイプの発生確率が画一的に決定されるのはよくない。

そこで、本実施形態では、頻度テーブルという手法、即ち図4(a)に例示されるタイミングタイプ選択用頻度テーブルを用いることにより、自動コード伴奏のテンポに合う適切なタイミングタイプ(1小節中のコードの発音回数)を確率的に選択することが可能となる。

【0046】

次に、図3に例示されるフローチャートにおいて、CPU101は、ステップS302で確率的に選択したタイミングタイプの内容を判定する(ステップS303)。CPU101は、タイミングタイプが「Type1」、「Type2」、又は「Type3」の何れかである場合はステップ304を、タイミングタイプが「Type0」である場合はステップS305を、タイミングタイプが「TypeC」である場合はステップS306及びステップS307を、夫々実行する。

10

【0047】

ステップS303での判定の結果、タイミングタイプが「Type1」、「Type2」、又は「Type3」の何れかである場合、即ち小節内でのコードの発音回数が1回、2回、又は3回である場合には、CPU101は、ステップS304において、タイミングタイプ毎にROM102に記憶されている例えば複数のノートタイミングテーブルのうちの1つを確率的に選択し、RAM103に記憶する。このように、本実施形態では、確率的に選択されたタイミングタイプ(小節内でのコードの発音回数)毎に、更に複数のバリエーションの中から1つのノートタイミングテーブルを確率的に選択することができる。

20

【0048】

図5(a)及び(c)は、例えばタイミングタイプ「Type2」に対して複数(例えば8個)用意されるノートタイミングテーブル1及び2のデータ構成例を示す図である。これらの例に示されるように、1つのノートタイミングテーブルには、1小節分の例えば1~4拍及びそれを更に半拍ずつ区切った横方向8列の発音タイミングの夫々について、次のような情報が設定される。なお、この例は、自動コード伴奏が4拍子である場合の例であり、自動コード伴奏が他の拍子である場合にはその拍子数に対応した拍数に基づいて発音タイミングが区切られる。

【0049】

まず、図5(a)又は(c)の左端の列に「Tick」の文字列が設定されている行(以降、この行を「Tick行」と呼ぶ)の上記半拍単位の各先頭タイミング毎に、そのタイミングが含まれる拍の先頭からそのタイミングの先頭までの[tick]時間が設定される。これらの値としては、第1、第2、第3、及び第4拍の各拍の前半の半拍(以降「表拍」)の先頭タイミングでは、各拍の先頭なので、0[tick]が設定される。また、これらの値としては、第1、第2、第3、及び第4拍の各拍の後半の半拍(以降「裏拍」)の先頭タイミングでは、各拍(=128[tick])のちょうど半分なので、64[tick]が設定される。なお、図5(a)又は(c)は、1拍が128[tick]である場合の例である。

30

【0050】

次に、図5(a)又は(c)の左端の列に「Gate」の文字列が設定されている行(以降、この行を「Gate行」と呼ぶ)の上記半拍単位の各先頭タイミング毎に、そのタイミングがコードの発音タイミングであれば、そこで発音されるコード音の長さを[tick]時間で表した値が設定される。図5(a)に例示される「Type2ノートタイミングテーブル1」では、Gate行において、第1拍の裏拍及び第3拍の裏拍の2つのタイミングにおいて共に、コード音長=15[tick]が設定される。一方、図5(c)に例示される「Type2ノートタイミングテーブル2」では、Gate行において、第1拍の表拍及び第2拍の裏拍の2つのタイミングにおいて共に、コード音長=15[tick]が設定される。

40

【0051】

50

更に、図5(a)又は(c)の左端の列に「Velocity」の文字列が設定されている行(以降、この行を「Velocity行」と呼ぶ)の上記半拍単位の各先頭タイミング毎に、そのタイミングがコードの発音タイミングであれば、そこで発音されるコード音を構成する各ボイスのベロシティ値(最大値は127)が設定される。図5(a)に例示される「Type2ノートタイミングテーブル1」では、Velocity行において、第1拍の裏拍及び第3拍の裏拍の2つのタイミングにおいて共に、ベロシティ値=75が設定される。図5(c)に例示される「Type2ノートタイミングテーブル2」でも同様に、Velocity行において、第1拍の表拍及び第2拍の裏拍の2つのタイミングにおいて共に、ベロシティ値=75が設定される。発音が行われないタイミングでは、ベロシティ値=0が設定される。

10

【0052】

以上のようにして、図3のステップS303->ステップS304において、例えば図5(a)の「Type2ノートタイミングテーブル1」が選択された場合には、図5(b)の楽譜表現として示されるような発音タイミングで1小節分のコード音が発音され、例えば図5(c)の「Type2ノートタイミングテーブル2」が選択された場合には、図5(d)の楽譜表現として示されるような図5(b)とは異なる発音タイミングで1小節分のコード音が発音される結果になる。

【0053】

上述したようなノートタイミングテーブルとしては、「Type1」、「Type2」、及び「Type3」の夫々について、図5(a)及び(c)として例示したように、複数のノートタイミングテーブルがROM102に用意されてよい。そしてこの場合、CPU101は、ステップS302で決定したタイミングタイプに対応して、ROM102に記憶されている複数のノートタイミングテーブルの中から1つを確率的に選択し、RAM103に記憶する。

20

【0054】

この確率的な選択動作を実現するために、本実施形態では、図4(b)に例示されるようなデータ構成を有する、タイミングタイプ別ノートタイミングテーブル選択用頻度テーブルがROM102に記憶されて用いられる。この頻度テーブルとしては、タイミングタイプ別に異なる設定のものが用意されてよい。図4(b)に例示されるデータ構成を有するタイミングタイプ別ノートタイミングテーブル選択用頻度テーブルにおいて、左端項目に「No」が登録されている行には、図5(a)又は(c)に例示されるような選択され得るタイミングタイプのノートタイミングテーブルの番号が、1から順に図4(b)の例では8まで設定されている。また、左端項目に「頻度」が登録されている行の各列には、同じ列に設定されている番号を有するノートタイミングテーブルが選択される頻度値[%]が設定される。図4(b)に例示されるような頻度テーブルに対して、CPU101は、図4(a)のタイミングタイプ選択用頻度テーブルの場合と同様に、例えば1から100までの値域を有する任意の乱数値を発生する。そして、CPU101は例えば、生成された乱数値が1から20の乱数範囲(番号1のノートタイミングテーブルの選択確率20%に対応)にあれば、ノートタイミングテーブル1を選択する。又は、CPU101は例えば、生成された乱数値が21から40の乱数範囲(番号2のノートタイミングテーブルの選択確率20%に対応)にあれば、ノートタイミングテーブル2を選択する。他の番号のノートタイミングテーブルも、ノートタイミングテーブル1又は2の場合と同様に、確率的に選択される。

30

40

【0055】

以上のようにして、本実施形態では、小節毎に、まず図3のステップS302において、図4(a)に例示されるタイミングタイプ選択用頻度テーブルを用いることにより、現在選択されている自動コード伴奏のテンポに適合する小節内でのコード発音回数を、タイミングタイプとして確率的に選択することが可能となる。

そして、小節毎に、次に図3のステップS303->ステップS304において、図4(b)に例示されるタイミングタイプ別ノートタイミングテーブル選択用頻度データを用

50

いることにより、選択されたタイミングタイプ（「Type 1」、「Type 2」、又は「Type 3」）毎に用意されている、夫々異なるコード発音タイミングを有する複数のノートタイミングテーブルのうちの1つを、確率的に選択することが可能となる。

これにより、本実施形態では、小節毎にコード発音回数及びコード発音タイミングを確率的に変化させながら、自動コード伴奏を実施することが可能となる。即ち、演奏者がピアノやギターのリブジャズ演奏等において、各小節内でのコードの発音回数と半拍単位でのコードの発音タイミングを変化させながら行う音楽表現を、自動コード伴奏においても実現することが可能となる。

【0056】

図3のステップS303での判定の結果、タイミングタイプが「Type 0」である場合、CPU101は、ステップS305において、ROM102に記憶されている「Type 0」専用の1つのノートタイミングテーブルを選択しRAM103に記憶する。

10

【0057】

図5(e)は、タイミングタイプ「Type 0」に対して用意される1つのノートタイミングテーブルのデータ構成例を示す図である。基本的なデータ構成は、図5(a)又は図5(c)に例示される「Type 1~3」の場合のテーブル例と同様である。ただし、図5(e)に例示される「Type 0のノートタイミングテーブル」のGate行において、1小節分の半拍単位の8つの発音タイミングのどのタイミングにも、コード音長 = 0 [tick] が設定されている。

【0058】

以上のようにして、CPU101が、ステップS303 -> ステップS305として、例えば図5(e)の「Type 0 ノートタイミングテーブル」を選択した場合には図5(f)の楽譜表現として示されるように、その小節は全休符となり、コード音は発音されない結果となる。

20

【0059】

このようにして、本実施形態では、小節毎に、タイミングタイプ「Type 0」が確率的に選択されることにより、音楽表現としてその小節ではコード音を発音させないといった自動コード伴奏を実現することが可能となる。

【0060】

図3のステップS303での判定の結果、タイミングタイプが「Type C」である場合、まずCPU101は、ステップS306において、ステップS301でROM102から取得された自動コード伴奏データに設定されているコード位置をサーチする。

30

【0061】

そして、CPU101は、ステップS307において、ステップS306でサーチしたコード位置に合わせて、図5(a)又は(c)などと同様の形式のノートタイミングテーブルを生成しRAM103に記憶する。

【0062】

以上のようにして、CPU101が、ステップS303 -> ステップS306として、「Type C」のためのノートタイミングテーブルを生成した場合には、その小節では自動コード伴奏データによりコードが変化する毎に、変化したコードが発音される結果となる。

40

【0063】

図6は、図2のステップS207のアンティシペーション・コード取得処理の詳細例を示すフローチャートである。この処理は、アンティシペーションを発生させる。「アンティシペーション」とは、指定されたコードを半拍先取りする演奏をいう。自動コード伴奏の楽曲の曲調によってアンティシペーションさせると効果的なものとそうでないものがあるため、演奏者は、図1のスイッチ部105内の特には図示しない切替えスイッチによって、アンティシペーションのオン又はオフを切り替えることができる。又は、自動コード伴奏がROM102に記憶される工場出荷時点で、アンティシペーションのオン又はオフが設定されてもよい。

50

【 0 0 6 4 】

図 6 のフローチャートにおいて、まず CPU 1 0 1 は、ステップ S 6 0 1、ステップ S 6 0 2、及びステップ S 6 0 3 において、以下の全ての判定が YES となった場合に、ステップ S 6 0 4 に移行してアンティシペーションを発生させる。

ステップ S 6 0 1：演奏者の設定又は工場設定により、アンティシペーション処理がオンか否か。

ステップ S 6 0 2：現在のノートタイミングは裏拍であるか否か。

ステップ S 6 0 3：次拍に自動コード伴奏のコードチェンジ（現在のコードとは異なるコード）は存在するか。

【 0 0 6 5 】

図 7 は、アンティシペーション・コード取得処理の説明図である。例えば図 7 (a) に示されるように、コード進行として、C M 7（第 1 小節 1 拍目表拍）、A 7（第 2 小節 1 拍目表拍）、D m 7（第 3 小節 1 拍目表拍）、G 7（第 4 小節 1 拍目表拍）、C M 7（第 5 小節 1 拍目表拍）、A 7（第 6 小節 1 拍目表拍）、D m 7（第 7 小節 1 拍目表拍）、G 7（第 7 小節 3 拍目表拍）、及び C M 7（第 8 小節 1 拍目表拍）の各コードが与えられたとする。

また、図 7 (b) の第 7 小節の拡大図に示されるように、現在のタイミング 7 0 1 が、「現在位置」と表記してある第 7 小節 2 拍目裏拍の先頭タイミングに位置しているとする。第 7 小節 2 拍目裏拍の次の第 7 小節 3 拍目の表拍の先頭タイミングには、コード G 7 が指定されている。

この場合、図 2 のステップ S 2 0 7 のアンティシペーション・コード取得処理では、現在のタイミング 7 0 1 である第 7 小節 2 拍目裏拍のタイミングにおいて、次拍である第 7 小節 3 拍目の表拍に指定されているコード G 7 が半拍先行して発音指示される。

【 0 0 6 6 】

具体的には、今、アンティシペーション処理がオンにされているとすれば、CPU 1 0 1 は、図 7 のタイミング 7 0 1 において図 2 のステップ S 2 0 7 のアンティシペーション・コード取得処理を実行した場合、図 6 のステップ S 6 0 1、ステップ S 6 0 2、及びステップ S 6 0 3 の何れの判定も YES となる。

なお、CPU 1 0 1 は、図 2 のステップ S 2 1 1 のカウンタ更新処理の説明で前述したように、RAM 1 0 3 に記憶されている拍内 tick カウンタ変数値が例えば 6 4 [t i c k] になったか否かを判定することにより、図 6 のステップ S 6 0 2 で現在のタイミングが裏拍の先頭タイミングであるか否かを判定する。

また、CPU 1 0 1 は、RAM 1 0 3 に記憶されている現在の拍及び次の拍のコード指定を確認することにより、図 6 のステップ S 6 0 3 で次拍にコードチェンジが存在するか否かを判定する。

【 0 0 6 7 】

次拍にコードチェンジが存在しない場合（ステップ S 6 0 3 の判定が NO）、CPU 1 0 1 は現時点でのコードを取得する（ステップ S 6 0 4）。

【 0 0 6 8 】

次拍にコードチェンジが存在する場合（ステップ S 6 0 3 の判定が YES）、すなわち次拍でコードが変化する場合、CPU 1 0 1 は次拍のコードを取得する（ステップ S 6 0 5）。

【 0 0 6 9 】

最後に CPU 1 0 1 は取得したコードを、後述するボイシング処理で使用するために発音コードデータとして、RAM 1 0 3 に記憶する（ステップ S 6 0 6）。

【 0 0 7 0 】

以上のように、ステップ S 6 0 1、ステップ S 6 0 2、及びステップ S 6 0 3 の全ての判定が YES となることにより、CPU 1 0 1 はアンティシペーション処理を実行する。すなわち、次拍のコードを今回発音するコードとして取得する。

なお、発音タイミングが 4 拍目の裏拍である場合は、次小節の伴奏データを RAM 1 0

10

20

30

40

50

3に読み込み、次小節の1拍目のコードを参照してコードチェンジがあるか否かを判断すればよい。

【0071】

図8は、図2のステップS208のボイス処理の詳細例を示すフローチャートである。ボイス処理において、CPU101は、RAM103に記憶されている現在の小節の自動コード伴奏データから抽出した今回のノートオンに対応するコード及びキーに対してボイステーブルデータを決定し、RAM103のノートオンエリアに記憶する。

【0072】

CPU101は、まず、図6のステップS606でRAM103に記憶した発音コードデータが、RAM103に記憶されている前回発音時のコードと同じであるか否かを判定する(ステップS801)。

10

【0073】

ステップS801の判定がYESの場合、CPU101は、前回選択したボイステーブルデータの使用を継続し、図8のフローチャートで例示される図2のステップS208のボイス処理を終了する。この結果、CPU101は、前述した図2のステップS209のノートオン処理において、RAM103に記憶されている前回と同じボイステーブルデータが示すボイス群の各ボイスに対応するノートナンバーの楽音の発音を、音源LSI508に指示する。

【0074】

ステップS801の判定がNOの場合、CPU101は、以下に説明するボイス処理を実行する。

20

【0075】

まず、CPU101は、今回のノートオンのタイミングにおける楽曲キーを、RAM103に読み込まれている自動コード伴奏データから取得する(ステップS803)。

自動コード伴奏データは、図2のステップS203のタイミングデータ生成処理内の前述した図3のステップS301にて、RAM103に読み込まれている。コードについては、図6のステップS606でRAM103に記憶した発音コードデータを以下のボイス処理で使用する。尚、キーについては楽曲を通して変わらない場合が多いため、小節ごとに読み込むのではなく、図3のステップS301において、別途キー情報としてRAM103に読み込んでおいて、その情報をここで使用するようにしてもよい。

30

なお、CPU101は、取得したコードの情報を、次回に前述したステップS801で判定する前回のコード情報として、RAM103に記憶させる。

【0076】

今例えば、RAM103に読み込まれている自動コード伴奏データによって、図9(a)に例示されるように、キー=C(図中「Key C」と表記)、コード進行=Dm7、G7、CM7、FM7、Bm7、E7、Am7、及びA7(例えば1小節毎に1コード)が順次指定されるものとする。

ここで例えば、図9(a)に例示される第2小節中の、前述したノートタイミングデータによって指定される任意のノートオンタイミング(コード発音タイミング)において、図8のフローチャートで例示される図2のステップS208のボイス処理が実行された場合を考える。この場合、CPU101は例えば、ステップS803において、コード=G7とキー=Cを取得する。

40

また例えば、図9(a)に例示される第6小節中の、前述したノートタイミングデータによって指定される任意のノートオンタイミング(コード発音タイミング)において、図8のフローチャートで例示される図2のステップS208のボイス処理が実行された場合を考える。この場合、CPU101は例えば、ステップS803において、コード=E7とキー=Cを取得する。

【0077】

次に、CPU101は、ROM102に記憶されているスケール決定テーブルを参照することにより、スケールを決定する処理を実行する(ステップS804)。

50

図9(b)は、スケール決定テーブルのデータ構成例である。スケール決定テーブルは、取得されているコードのコードタイプ(図9(b)に例示される表の横方向の各列)と、キーの音高からそのコードのルート音の音高までの度数(図9(b)に例示される表の縦方向の各行)とに応じて、図9(b)に例示される表の各行と各列が交差する登録位置に、コードが有するスケールの名称が登録されている。

スケールとしては、図9(b)に示されるように、メジャースケール、リディアンスケール、ミクソリディアンスケール、ミクソリディアン#11スケール、ミクソリディアンスケール9スケール、ミクソリディアンスケール9 13スケール、オルタードスケール、ドリアンスケール、フィリジアンスケール、エオリアンスケール、ロクリアンスケールなどが登録可能である。その他、様々な音楽ジャンルで使用され得るスケールが登録されてよい。

10

いま例えば、CPU101が、ステップS803において、現在のノートオンタイミングである、図9(a)のコード進行中の第2小節の任意のノートオンタイミング(コード発音タイミング)において、コード=G7とキー=Cを取得している場合を考える。この場合、CPU101は、キー=CからコードG7のルート音Gまでの度数=5(図9(b)中の「V」と、コードタイプ=7とによって、図9(b)に例示されるスケール決定テーブルを参照する。この結果、CPU101は、「V」行と「7」列の交差位置から、スケール=「ミクソリディアン」を決定する。

また例えば、CPU101が、ステップS803において、現在のノートオンタイミングである、図9(a)のコード進行中の第6小節の任意のノートオンタイミング(コード発音タイミング)において、コード=E7とキー=Cを取得している場合を考える。この場合、CPU101は、キー=CからコードE7のルート音Eまでの度数=3(図9(b)中の「III」と、コードタイプ=7とによって、図9(b)に例示されるスケール決定テーブルを参照する。この結果、CPU101は、「III」行と「7」列の交差位置から、スケール=「ミクソリディアン 9 13」を決定する。

20

【0078】

次に、CPU101は、ステップS804で決定したスケール毎に予め用意されてROM102に記憶されているボイスングテーブルを、ROM102から取得する(ステップS805)。図9(c)は、スケール=「ミクソリディアン」である場合のボイスングテーブルのデータ構成例を示す図である。CPU101は例えば、前述したように、ステップS803でコード=G7とキー=Cを取得し、更にステップS804でスケール=「ミクソリディアン」を決定した場合に、図9(c)に例示されるボイスングテーブルをROM102から取得する。

30

【0079】

コード伴奏においては、コードのボイスングが重要である。ボイスングとは、1つのコードを発音させるのに、オクターブの中でどのボイスをどのように積み重ねるかを決定することである。ジャズなどの音楽ジャンルでは、コード伴奏において、ルート音に対して半音単位で9度、11度、又は13度上のいわゆるテンション音を使うことが多く、これらのボイスを使うことによって緊張感があり音楽性豊かなコード演奏が実現される。コード演奏においては、どのスケールを使うのかがポイントになり、キーとコードによって使えるテンションが異なる。そこで、本実施形態では、CPU101が、図8のステップS803及びステップS804によって、現在のノートオンタイミングで指定される例えばコード=G7とキー=Cとに基づいて、演奏可能な、例えば「ミクソリディアン」スケールを決定する。

40

【0080】

更に、本実施形態では、今回のノートオンタイミングで例えばコード=G7を「ミクソリディアン」スケールで発音させる場合に、同じ「G7のミクソリディアン」であっても、複数種類(図9(c)では例えば6種類)のボイスングのパリエーションから1つのボイスング(ボイスングパターン)を確率的に選択し、そのボイスングでコード=G7をノートオンさせることができる。

50

例えば、図9(c)に例示されるボイスングテーブルにおいて、1番のボイスングテーブルデータが選択された場合、コード=G7のノートオンにおいて、ルート音Gに対して、半音単位で4半音(長3度:B)、9半音(長6度:E)、10半音(短7度:F)、及び14半音(長9度:A)の音程を有する4音からなるボイス群が使用される。

また例えば、3番のボイスングテーブルデータが選択された場合には、コード=G7のノートオンにおいて、ルート音Gに対して、半音単位で4半音(長3度:B)、10半音(短7度:F)、及び14半音(長9度:A)の音程を有する3音からなるボイス群が使用される。

なお、ジャズ演奏などのテンションが入ったコード演奏では、一般的にルート音は発音されない場合が多いため、ボイスングテーブルのボイス群もルート音(度数1)を含まない場合が多い。

【0081】

ボイスングテーブルから1組のボイスングテーブルデータを決定するための要素としては、ジャズをはじめとする多くの音楽ジャンルで一般的にA type又はB typeと呼ばれるボイスングタイプや、何音で発音させるかを指示するポリ数が関係してくる。A type、B typeとは音域の広いボイスングであるか、音域の狭いボイスングであるかの違いである。A typeでは、テンション音を含めたボイスングで、ルート音に対して例えば3度、5度、7度、9度とボイスを積み上げてできるようなボイスングタイプをいう。B typeとは、A typeのボイスングタイプから、例えば7度、9度をオクターブ下げて、A typeと比較して音域を狭くしたボイスングタイプをいう。

図9(c)に例示されるボイスングテーブルでは、1番と2番のボイスングテーブルデータは、ボイスングタイプ=A type、ポリ数=4音である場合に選択できることを示している。これらの2つのボイスングテーブルデータは、どちらもボイスングタイプ=A type、ポリ数=4音の場合に選択できるが、その場合にどちらが選択されるかは、後述する図10(b)に例示されるボイスングテーブルデータ選択用頻度テーブルを用いて後述する図8のステップS810の処理により確率的に決定される。

また、3番のボイスングテーブルデータは、ボイスングタイプ=A type、ポリ数=3音である場合に選択できることを示している。ボイスングタイプ=A type、ポリ数=3音である場合に選択できるボイスングテーブルデータは、3番のみであるため、その場合は必ず3番のボイスングテーブルデータが選択される。

更に、4番と5番のボイスングテーブルデータは、ボイスングタイプ=B type、ポリ数=4音である場合に選択できることを示している。これらの2つのボイスングテーブルデータは、どちらもボイスングタイプ=B type、ポリ数=4音の場合に選択できるが、その場合にどちらが選択されるかは、後述する図10(b)に例示されるボイスングテーブルデータ選択用頻度テーブルを用いて後述する図8のステップS810の処理により確率的に決定される。

そして、6番のボイスングテーブルデータは、ボイスングタイプ=B type、ポリ数=3音である場合に選択できることを示している。ボイスングタイプ=B type、ポリ数=3音である場合に選択できるボイスングテーブルデータは、6番のみであるため、その場合は必ず6番のボイスングテーブルデータが選択される。

【0082】

上述のボイスングテーブルデータの選択動作を実現するために、CPU101はまず、予め用意してROM102に記憶されているポリ数選択用頻度テーブルを参照することにより、ポリ数を確率的に決定する(ステップS806)。図10(a)は、ポリ数選択用頻度テーブルのデータ構成例を示す図である。図10(a)に例示されるポリ数選択用頻度テーブルの左端列に登録される「Ballad」、「Slow」、「Mid」、「Fast」、及び「Very Fast」は夫々、図4(a)のタイミングタイプ選択用頻度テーブルの場合と同様に、自動コード伴奏データのテンポ範囲を示している。

【0083】

CPU101は、図8のステップS806において、ROM102に記憶されている図

10

20

30

40

50

10 (a) に例示されるポリ数選択用頻度テーブルを用いて、次のような制御処理を実行する。まず、図2のステップS203のタイミングデータ生成処理内の図3のステップS301にてROM102から読み込まれる自動コード伴奏データに例えばテンポ範囲「Ballad」が設定されている場合には、CPU101は、図10 (a) に例示されるポリ数選択用頻度テーブルにおいて、左端項目に「Ballad」が登録されている行のデータを参照する。この行には、ポリ数3又はポリ数4が夫々、10%又は90%の確率で選択されることを示す各頻度値 [%] が設定されている。これに対して、CPU101は、前述した図3のステップS302の場合と同様にして、例えば1から100までの値域を有する任意の乱数値を発生する。そして、CPU101は例えば、生成した乱数値が1から10の乱数範囲 (「ポリ数3」の頻度値10%に対応) にあれば、「ポリ数3」を選択する。又は、CPU101は例えば、生成した乱数値が11から100の乱数範囲 (「ポリ数4」の頻度値90%に対応) にあれば、「ポリ数4」を選択する。このようにして、CPU101は、「ポリ数3」及び「ポリ数4」の各ポリ数を夫々、ポリ数選択用頻度テーブルの「Ballad」行に設定された10%及び90%の各確率で選択することができる。

10

【0084】

図3のステップS301にてROM102から読み込まれる自動コード伴奏データに例えばテンポ範囲「Slow」、「Mid」、「Fast」、又は「Very Fast」が設定されている場合も、「Ballad」が設定されている上述の場合と同様にして、CPU101は例えば、図10 (a) に例示される構成を有するポリ数選択用頻度テーブルにおいて、左端項目に「Slow」、「Mid」、「Fast」、又は「Very Fast」が登録されている何れかの行の各頻度値を参照する。次に、CPU101は、その行の「ポリ数3」又は「ポリ数4」の各ポリ数に対して夫々設定されている頻度値 [%] に応じた1から100の範囲内の各乱数範囲を設定する。そして、CPU101は、1から100の範囲で乱数値を発生させ、生成された乱数値が上記各乱数範囲のどの範囲に入るかによって、「ポリ数3」又は「ポリ数4」の何れかのポリ数を選択する。このようにして、CPU101は、「ポリ数3」及び「ポリ数4」の各ポリ数を夫々、ポリ数選択用頻度テーブルの各テンポ範囲の行に設定された各頻度値に対応する確率で選択することができる。

20

【0085】

ポリ数は、楽曲のテンポや曲調により、自然な演奏になる内容が異なる。このため、本実施形態では、「Ballad」、「Slow」、「Mid」、「Fast」、又は「Very Fast」等の曲調ごとに、ポリ数の出現する度合いを決定する図10 (a) に例示されるデータ構成を有するポリ数選択用頻度テーブルが参照される。

30

上述のステップS806の処理を実行するCPU101は、音数選択手段として動作する。

【0086】

ステップS806でポリ数を決定した後、CPU101は、ノートオンされるコードが、前述したボイスンタイプのアタイプであるべきかBタイプであるべきかを判定する。具体的には、CPU101は、ノートオンされるコードのルート音の音高がF#以上であるか否かを判定する (ステップS807) 。

40

ステップS807の判定がNOならば、CPU101は、現在のコードのボイスンタイプとして、Aタイプを選択する (ステップS808) 。

ステップS807の判定がYESならば、CPU101は、現在のコードのボイスンタイプとして、Bタイプを選択する (ステップS809) 。

ステップS807の判定は、1オクターブを半分の位置で分割して、各コードが或る音域内に収まり、コードの遷移によって音域が飛びすぎないようにするためである。

ステップS807～S809を実行するCPU101は、ボイスンタイプ決定手段として動作する。

【0087】

50

最後に、CPU 101は、前述したステップS 805にてROM 102から取得している図9(c)に例示されるボイシングテーブルに対応して用意されROM 102に記憶されているボイシングテーブルデータ選択用頻度テーブルを用いて、ステップS 806～S 809の処理により決定したポリ数(3又は4)とボイシングタイプ(A又はB)の組合せに基づいて、図9(c)に例示されるボイシングテーブルから最適なボイシングテーブルデータを確率的に抽出し、RAM 103に記憶する(ステップS 810)。

ステップS 810を実行するCPU 101は、ボイシングパターン選択手段として動作する。

【0088】

図10(b)は、ボイシングテーブルデータ選択用頻度テーブルのデータ構成例を示す図である。図10(b)に例示されるボイシングテーブルデータ選択用頻度テーブルの左端列に登録される「4/A」、「4/B」、「3/A」、及び「3/B」は夫々、ステップS 806～S 809で決定されたポリ数(3又は4)とボイシングタイプ(A type又はB type)の組合せを示している。

【0089】

CPU 101は、図8のステップS 810において、次のような制御処理を実行する。まず、ステップS 806～S 809で決定された「ポリ数/ボイシングタイプ」が「4/A」である場合には、CPU 101は、図10(b)に例示されるボイシングテーブルデータ選択用頻度テーブルにおいて、左端項目に「4/A」が登録されている行のデータを参照する。番号1又は2のボイシングテーブルデータが夫々、60%又は40%の確率で選択されることを示す各頻度値[%]が設定されている。他の番号のボイシングテーブルデータには、頻度値として0%が設定されているため、「4/A」の組合せに対してはこれらの番号のボイシングテーブルデータは選択され得ない。これに対して、CPU 101は、前述したステップS 806の場合と同様にして、例えば1から100までの値域を有する任意の乱数値を発生する。そして、CPU 101は例えば、生成した乱数値が1から60の乱数範囲(番号1の頻度値60%に対応)にあれば、番号1のボイシングテーブルデータを選択する。又は、CPU 101は例えば、生成した乱数値が61から100の乱数範囲(番号1の頻度値40%に対応)にあれば、番号1のボイシングテーブルデータを選択する。このようにして、CPU 101は、番号1及び番号2の各ボイシングテーブルデータを夫々、ボイシングテーブルデータ選択用頻度テーブルの「4/A」行に設定された60%及び40%の各確率で選択する。

【0090】

ステップS 806～S 809で決定された「ポリ数/ボイシングタイプ」が「4/B」、「3/A」、又は「3/B」である場合も、「4/A」が設定されている上述の場合と同様にして、CPU 101は例えば、図10(b)に例示される構成を有するボイシングテーブルデータ選択用頻度テーブルにおいて、左端項目に「4/B」、「3/A」、又は「3/B」が登録されている何れかの行の各頻度値を参照する。次に、CPU 101は、その行の番号1から6の各ボイシングテーブルデータに対して夫々設定されている頻度値[%]に応じた1から100の範囲内の各乱数範囲を設定する。そして、CPU 101は、1から100の範囲で乱数値を発生させ、生成された乱数値が上記各乱数範囲のどの範囲に入るかによって、番号1から6の何れかのボイシングテーブルデータを選択する。このようにして、CPU 101は、図9(c)のボイシングテーブル上の番号1から6の各ボイシングテーブルデータを夫々、図10(b)のボイシングテーブルデータ選択用頻度テーブルの各「ポリ数/ボイシングタイプ」の行に設定された各頻度値に対応する確率で選択する。

CPU 101は、ステップS 810において、上述のようにして図9(c)のボイシングテーブルから抽出したボイシングテーブルデータを、RAM 103に記憶する。

【0091】

ステップS 810の処理の後、CPU 101は、図8のフローチャートで例示される図2のステップS 208のボイシング処理を終了する。

10

20

30

40

50

【 0 0 9 2 】

以上説明したボイスンク処理により、本実施形態では、自動コード伴奏において、ノートオンされるコード及びキーに対応して、音楽理論に則ったスケールを適切に選択し、そのスケールに対応する複数のバリエーションのボイスンクテーブルデータの候補を、ボイスンクテーブルとして提供することができる。続いて、本実施形態では、確率的に決定されるポリ数とボイスンクタイプの組合せに基づいて、上記複数のバリエーションのボイスンクテーブルデータの候補のうちの1組を、確率的に抽出することができる。そして、本実施形態では、このようにして抽出したボイスンクテーブルデータとして与えられるボイス群を使って、自動コード伴奏におけるコードのノートオン処理を実行することができる。これにより、音楽理論に則りながら多彩なバリエーションの自動コード伴奏を実現することが可能となる。

10

【 0 0 9 3 】

図11(a)はC7(ミクソリディアンスケール)の楽譜表現であり、図11(b)、(c)、(d)、(e)、(f)、及び(g)は、C7(ミクソリディアンスケール)におけるボイスンクのバリエーションの例を示す楽譜表現である。図11(b)は、ボイスンクタイプがA t y p eで9度及び13度のテンション音を含むポリ数 = 4音のC7コードの例の楽譜表現である。図11(c)は、ボイスンクタイプがA t y p eで9度のテンション音を含むポリ数 = 4音のC7コードの例の楽譜表現である。図11(d)はボイスンクタイプがA t y p eで9度のテンション音を含むポリ数 = 3音のC7コードの例の楽譜表現である。図11(e)は、ボイスンクタイプがB t y p eで9度及び13度のテンション音を含むポリ数 = 4音のC7コードの例の楽譜表現である。図11(f)は、ボイスンクタイプがB t y p eで9度のテンション音を含むポリ数 = 4音のC7コードの例の楽譜表現である。図11(g)はボイスンクタイプがB t y p eで13度のテンション音を含むポリ数 = 3音のC7コードの例の楽譜表現である。

20

【 0 0 9 4 】

また、図10(h)はマイナースケールで使用されるC7(ミクソリディアン 9 1 3)スケールの楽譜表現であり、図10(i)及び(j)は、「C7ミクソリディアン 9 1 3」スケールにおけるボイスンクのバリエーションの例を示す楽譜表現である。図11(i)は、ボイスンクタイプがA t y p eで 9度のテンション音を含むポリ数 = 4音のC7コードの例の楽譜表現である。図11(j)は、ボイスンクタイプがA t y p eで 9度及び 13度のテンション音を含むポリ数 = 4音のC7コードの例の楽譜表現である。

30

【 0 0 9 5 】

図11に例示したように、本実施形態では、多彩なボイスンクのコードで自動コード伴奏を行うことが可能である。

【 0 0 9 6 】

以上説明した実施形態は、本発明による自動演奏装置が図1の電子鍵盤楽器100に内蔵されている実施形態であった。一方、自動演奏装置と電子楽器は、夫々個別の装置であってもよい。具体的には、例えば図12に示されるように、自動演奏装置は例えばスマートフォンやタブレット端末(以下「スマートフォン等1201」と記載)に自動演奏アプリとしてインストールされ、電子楽器は例えば自動コード伴奏機能を持たない電子鍵盤楽器1202であってよい。この場合、スマートフォン等1201と電子鍵盤楽器1202は、例えばMIDI over Bluetooth Low Energy(以下「BLE-MIDI」と記載、Bluetoothは登録商標)と呼ばれる規格に基づいて無線通信する。BLE-MIDIは、無線規格Bluetooth Low Energy上で楽器間の通信の標準規格MIDI(Musical Instrument Digital Interface:楽器デジタルインタフェース)で通信が行えるようにした楽器間無線通信規格である。電子鍵盤楽器1202は、Bluetooth Low Energy規格でスマートフォン等1201に接続することができる。その状態で、スマートフォン等1201上で実行される自動演奏アプリによって、図2から図11で説明した自動

40

50

コード伴奏機能に基づく自動コード伴奏データが、BLE-MIDI規格の通信路を介して、MIDIデータとして電子鍵盤楽器1202に送信される。電子鍵盤楽器1202は、BLE-MIDI規格で受信した自動コード伴奏MIDIデータに基づいて、図2から図11で説明した自動コード伴奏を実施する。

【0097】

図13は、図12に示される接続形態を有する自動演奏装置と電子楽器が個別に動作する他の実施形態における自動演奏装置1201のハードウェア構成例を示す図である。図13において、CPU1301、ROM1302、及びRAM1303、タッチパネルディスプレイ1305は、図1のCPU101、ROM102、及びRAM103と同様の機能を有する。CPU1301が、RAM1303にダウンロードされインストールされた自動演奏アプリのプログラムを実行することにより、CPU101が制御プログラムを実行することにより実現した、図2から図11で説明した自動コード伴奏機能と同じ機能を実現する。このとき、図1のスイッチ部105と同等の機能は、タッチパネルディスプレイ1305によって提供される。そして、自動演奏アプリは、自動コード伴奏用の制御データを自動コード伴奏MIDIデータに変換してBLE-MIDI通信インタフェース1305に引き渡す。

10

【0098】

BLE-MIDI通信インタフェース1305は、自動演奏アプリにより生成された自動コード伴奏MIDIデータを、BLE-MIDI規格に従って電子鍵盤楽器1202に送信する。この結果、電子鍵盤楽器1202が、図1の電子鍵盤楽器100の場合と同様の自動コード伴奏を実施する。

20

なお、BLE-MIDI通信インタフェース1305の代わりに、有線のMIDIケーブルで電子鍵盤楽器1202に接続するMIDI通信インタフェースが用いられてもよい。

【0099】

以上説明したようにして、本実施形態では、従来技術で課題であった、ジャズ等の特徴となるテンションノートを含んだコードの発音構成（ボイスイング）と、より自然な音数、音域による自動コード伴奏が実現され、あたかもジャズ等のピアニストが演奏しているかのような、コードパートの自動伴奏を提供することが可能となる。

特に、本実施形態では、コードの発音をより自然な音域で、コードチェンジの際には音域移動が少なくすむテンションノートを含んだ演奏ポジションでの発音、コードチェンジが無い場合はボイスイングをいたずらに変化させずに自然な発音をさせ、曲調や、テーマ・ソロと言ったジャズ等の特有の楽曲構成に合わせ、同時に発音させる音数（場合によっては発音させない）もアルゴリズムにより頻度調整し、より自然なコード伴奏を提供することが可能となる。

30

これにより、演奏者があたかもジャムセッションに参加しているかのような演奏体験が得られるようになる。また、例えばジャズを演奏したいけれどジャムセッションに参加する勇気がなく、躊躇している方のトレーニングの一環として活用も可能となる。また、ジャズ等の特徴を捉えた自動コード伴奏により、ジャズ等でのソロ（アドリブ）を演奏する際のフレージングの研究や練習など、教育の一環として活用することも可能となる。

このように、本実施形態による自動演奏装置では、演奏者による楽器のライブ演奏におけるタイミング及びボイスイングを表現可能な、自然な自動コード伴奏の実現が可能となる。

40

【0100】

以上の実施形態に関して、更に以下の付記を開示する。

(付記1)

発音させる音数を確率的に選択する音数選択手段と、

音域に対応するボイスイングタイプを決定するボイスイングタイプ決定手段と、

楽曲の調とコードにより決定されるスケールに基づく複数のボイスイングパターンから、前記選択された音数と、前記決定されたボイスイングタイプとの組合せに対応するボイスイングパターンを選択するボイスイングパターン選択手段と、

選択された前記ボイスイングパターンに基づきボイスイングされたコードの発音を音源に指

50

示する発音指示手段と、
を備える自動演奏装置。

(付記 2)

前記組合せに対応するボイシングパターンが複数ある場合は、前記複数のボイシングパターンからいずれか 1 つを確率的に選択する、付記 1 に記載の自動演奏装置。

(付記 3)

前記ボイシングタイプ決定手段は、発音すべきコードのルートの音高に基づき前記ボイシングタイプを決定する、付記 1 又は 2 に記載の自動演奏装置。

(付記 4)

楽音情報を送受信する通信手段を更に備え、

前記発音指示手段は、前記通信手段を介して、前記音源に前記コードの発音を指示する、付記 1 乃至 3 の何れか 1 項に記載の自動演奏装置。

10

(付記 5)

付記 1 乃至 3 の何れか 1 項に記載の自動演奏装置と前記音源とを備え、前記自動演奏装置の前記発音指示手段による前記選択されたボイシングパターンに基づきボイシングされたコードの発音の指示に基づいて、前記音源が自動コード伴奏を実施する電子楽器。

(付記 6)

発音させる音数を選択し、

音域に対応するボイシングタイプを決定し、

楽曲の調とコードにより決定されるスケールに基づく複数のボイシングパターンから、前記選択された音数と、前記決定されたボイシングタイプとの組合せに対応するボイシングパターンを選択し、

20

選択された前記ボイシングパターンに基づきボイシングされたコードの発音を音源に指示する、

自動演奏方法。

(付記 7)

発音させる音数を選択し、

音域に対応するボイシングタイプを決定し、

楽曲の調とコードにより決定されるスケールに基づく複数のボイシングパターンから、前記選択された音数と、前記決定されたボイシングタイプとの組合せに対応するボイシングパターンを選択し、

30

選択された前記ボイシングパターンに基づきボイシングされたコードの発音を音源に指示する、

処理をコンピュータに実行させるためのプログラム。

【符号の説明】

【 0 1 0 1 】

1 0 0 電子鍵盤楽器

1 0 1 C P U

1 0 2 R O M

1 0 3 R A M

40

1 0 4 鍵盤部

1 0 5 スイッチ部

1 0 6 音源 L S I

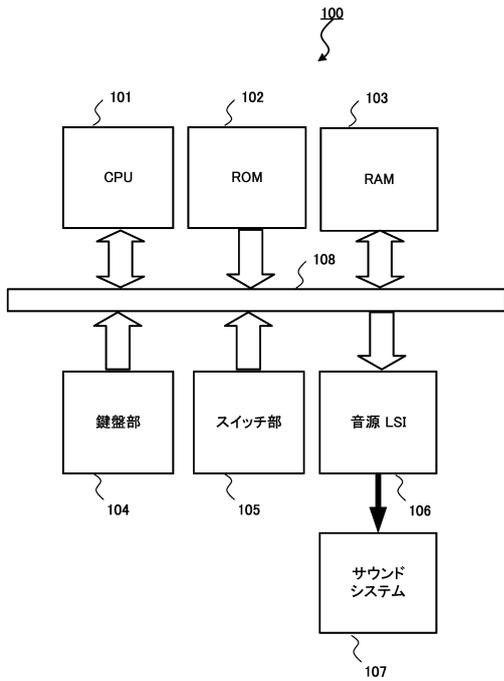
1 0 7 サウンドシステム

1 0 8 システムバス

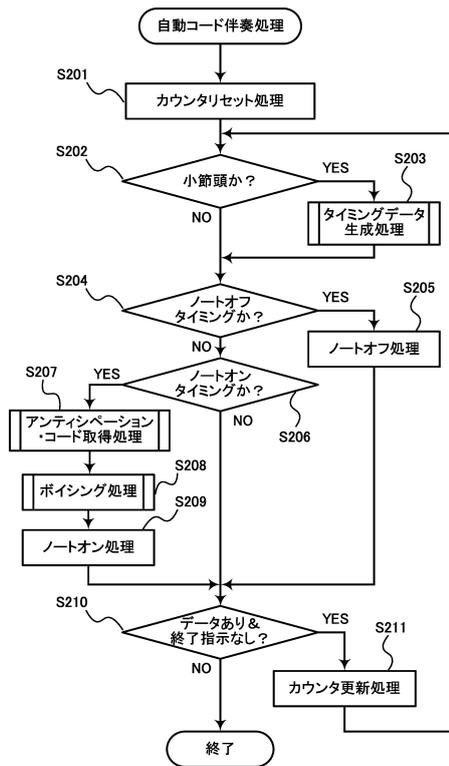
50

【図面】

【図 1】



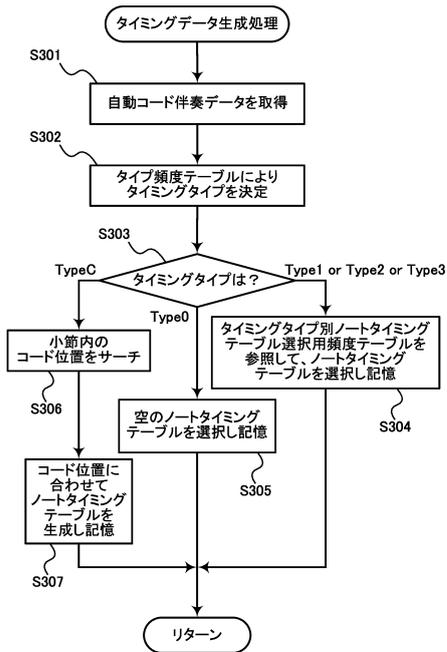
【図 2】



10

20

【図 3】



【図 4】

数字は%

	Type 0	Type 1	Type 2	Type 3	Type C
Ballad	0	10	20	10	60
Slow	0	20	30	30	20
Mid	0	40	40	10	10
Fast	10	40	20	0	30
Very Fast	40	50	10	0	0

(a) タイミングタイプ
選択用頻度テーブル

数字は%

No.	1	2	3	4	5	6	7	8
頻度	20	20	10	10	10	10	10	10

(b) タイミングタイプ別
ノートタイミングテーブル
選択用頻度テーブル

30

40

50

【 図 5 】

(a) Type2 ノートタイミングテーブル1のデータ構成例

拍	1	2	3	4
Tick	0 64	0 64	0 64	0 64
Gate	0 15	0 0 0 15	0 0 0	0 0
Velocity	0 75	0 0 0 75	0 0	0



(b) Type2 ノートタイミングテーブル1によるノートタイミングの楽譜表現

(c) Type2 ノートタイミングテーブル2のデータ構成例

拍	1	2	3	4
Tick	0 64	0 64	0 64	0 64
Gate	15 0	0 0 15	0 0 0 0	0 0
Velocity	75 0	0 0 75	0 0 0 0	0 0



(d) Type2 ノートタイミングテーブル2によるノートタイミングの楽譜表現

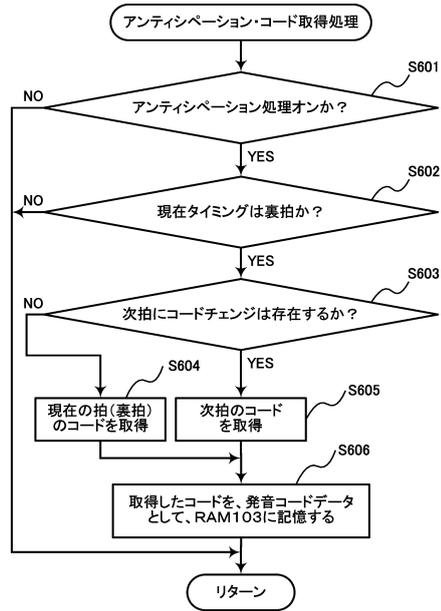
(e) Type0 ノートタイミングテーブルのデータ構成例

拍	1	2	3	4
Tick	0 64	0 64	0 64	0 64
Gate	0 0 0 0	0 0 0 0	0 0 0 0	0 0
Velocity	0 0 0 0	0 0 0 0	0 0 0 0	0 0



(f) Type0 ノートタイミングテーブルによるノートタイミングの楽譜表現

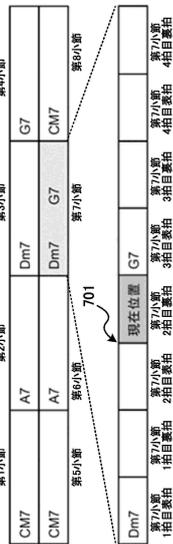
【 図 6 】



10

20

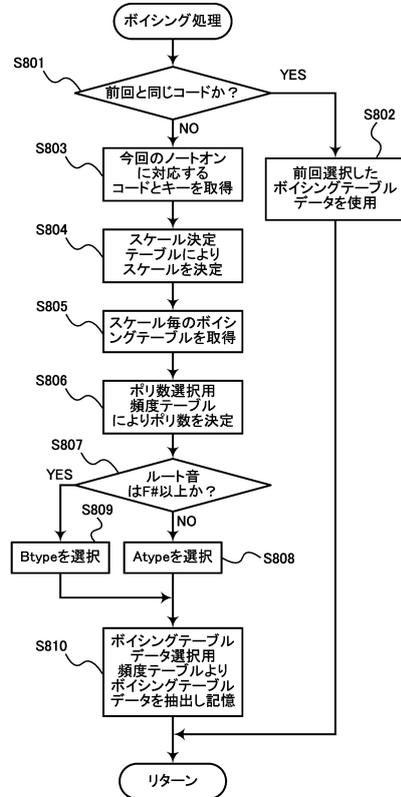
【 図 7 】



(a)

(b)

【 図 8 】



30

40

50

【 9 】

Key C

Dm7	G7	CM7	FM7
Bm7b5	E7	Am7	A7

(a)

コードタイプ	M 7	7	m 7	m 7 b 5
I	メジャー	ミクソリディアン	ドリアン	ロクリアン
II b	リディアン	ミクソリディアン# 1 1	ドリアン	ロクリアン
II	メジャー	ミクソリディアン	ドリアン	ロクリアン
III b	リディアン	ミクソリディアン# 1 1	ドリアン	ロクリアン
III	メジャー	ミクソリディアン b 9, b 1 3	フィリジアン	ロクリアン
IV	リディアン	ミクソリディアン	ドリアン	ロクリアン
V b	メジャー	ミクソリディアン	ドリアン	ロクリアン
V	メジャー	ミクソリディアン	ドリアン	ロクリアン
VI b	リディアン	ミクソリディアン	ドリアン	ロクリアン
VI	メジャー	ミクソリディアン b 9	エオリアン	ロクリアン
VII b	リディアン	ミクソリディアン# 1 1	ドリアン	ロクリアン
VII	リディアン	オルタード	ドリアン	ロクリアン

(b)

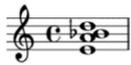
No	ボイス群	タイプ	ポリ数
1	4, 9, 10, 14	A	4
2	4, 7, 10, 14	A	4
3	4, 10, 14	A	3
4	-2, 2, 4, 10	B	4
5	-2, 2, 4, 7	B	4
6	-2, 4, 10	B	3

(c)

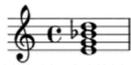
【 1 1 】



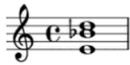
(a)



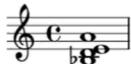
(b)



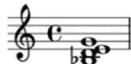
(c)



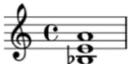
(d)



(e)



(f)



(g)



(h)



(i)



(j)

【 1 0 】

数字は%

テンポ	ポリ数:3	ポリ数:4
Ballad	10	90
Slow	20	80
Mid	40	60
Fast	50	50
Very Fast	60	40

(a) ポリ数選択用
頻度テーブル

数字は%

ポリ数/タイプ	1	2	3	4	5	6
4 / A	60	40	0	0	0	0
4 / B	0	0	0	60	40	0
3 / A	0	0	100	0	0	0
3 / B	0	0	0	0	0	100

(b) ボイシングテーブルデータ選択用
頻度テーブル

10

20

【 1 2 】

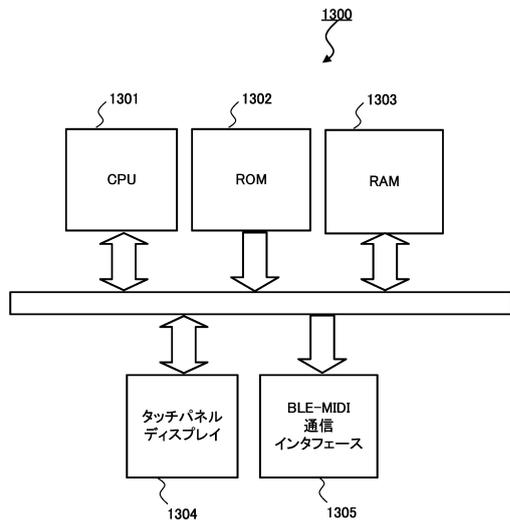


30

40

50

【図 13】



10

20

30

40

50

フロントページの続き

才計算機株式会社 羽村技術センター内

(72)発明者 橋 敏之

東京都羽村市栄町3丁目2番1号 カシオ計算機株式会社 羽村技術センター内

(72)発明者 岡野 真吾

東京都羽村市栄町3丁目2番1号 カシオ計算機株式会社 羽村技術センター内

審査官 渡部 幸和

(56)参考文献 特表2018-537727(JP,A)

特開平10-074087(JP,A)

(58)調査した分野 (Int.Cl., DB名)

G10H 1/00

G10G 1/00-3/04