

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3900188号
(P3900188)

(45) 発行日 平成19年4月4日(2007.4.4)

(24) 登録日 平成19年1月12日(2007.1.12)

(51) Int. Cl.		F I		
G 1 0 H	1/00	(2006.01)	G 1 0 H	1/00 1 0 2 Z
G 1 0 H	1/46	(2006.01)	G 1 0 H	1/00 Z
			G 1 0 H	1/46

請求項の数 1 (全 99 頁)

(21) 出願番号	特願2005-38700 (P2005-38700)	(73) 特許権者	000004075 ヤマハ株式会社 静岡県浜松市中沢町10番1号
(22) 出願日	平成17年2月16日(2005.2.16)	(74) 代理人	100107995 弁理士 岡部 恵行
(62) 分割の表示	特願2000-77340 (P2000-77340) の分割	(72) 発明者	西元 哲夫 静岡県浜松市中沢町10番1号 ヤマハ株 式会社内
原出願日	平成12年3月21日(2000.3.21)	(72) 発明者	柿下 正尋 静岡県浜松市中沢町10番1号 ヤマハ株 式会社内
(65) 公開番号	特開2005-173632 (P2005-173632A)	(72) 発明者	東儀 温 静岡県浜松市中沢町10番1号 ヤマハ株 式会社内
(43) 公開日	平成17年6月30日(2005.6.30)		
審査請求日	平成17年2月22日(2005.2.22)		
(31) 優先権主張番号	特願平11-224782		
(32) 優先日	平成11年8月9日(1999.8.9)		
(33) 優先権主張国	日本国(JP)		
(31) 優先権主張番号	特願平11-271400		
(32) 優先日	平成11年9月24日(1999.9.24)		
(33) 優先権主張国	日本国(JP)		

最終頁に続く

(54) 【発明の名称】 演奏データ作成装置

(57) 【特許請求の範囲】

【請求項1】

音符情報及び音量を表わす楽音制御情報をそれぞれ含む複数のパートから成る演奏データを供給する演奏データ供給手段と、

上記演奏データから同音色のパート及びそのパート数を検出するパート検出手段と、

上記パート数に基づいて、上記同音色のパートの音量が減少するように当該各パートの音量値を算出する音量値算出手段と、

上記音量値を表わす楽音制御情報を該当パートに付与する音量値付与手段とを具備することを特徴とする演奏データ作成装置。

【発明の詳細な説明】

【技術分野】

【0001】

この発明は、表情が付与された演奏データを作成するための演奏データ作成装置、換言すれば、供給された演奏データ中の特徴に基づいて、演奏データに各種表情を付加するためのパラメータの値を自動的に編集するパラメータ自動編集システムに関する。

【背景技術】

【0002】

MIDIデータを音符の情報のみで構成すると、機械的で無表情な演奏になってしまう。より自然な演奏、美しい演奏、生々しい演奏、癖のある個性的な演奏等、表情のある多彩な演奏出力を得るためには、様々な音楽的表現や、楽器らしさを制御データとして加え

る必要がある。様々な表現を加えるシステムとしては、楽譜を通して表現を加えるという方法などが考えられる。しかし、表情には上述のように様々なものがある。様々な表情を扱うことができるようになってこそ有意義なシステムと言える。

【0003】

また、表情の種類が多くなると自動的に付加するシステムを案出しないと、どのような表情を付けたら良いのか戸惑うばかりになってしまうため、自動的なシステムの実現のためにも、表情付けをモジュール化して扱うことが好ましい。従って、音楽的なMIDIデータを作成するためには、様々な音楽表現の特徴をルール化したモジュールを用意すると良いことが分かる。

【0004】

従来、供給された演奏データから再生される演奏に各種表情を付加するためのパラメータの値を編集する演奏データ作成装置又はパラメータ編集装置として、次のものが知られている。

【0005】

すなわち、

(1) 供給された演奏データに対して、マニュアル入力により、既に設定されているパラメータの値を修正したり、新たなパラメータを追加したりして、当該演奏データから再生される、機械的で無表情な演奏に表情(自然な演奏、美しい演奏、生々しい演奏等の音楽的表現)を付加できるようにしたもの

(2) 供給された演奏データ中、ユーザが指定した範囲の演奏データに対して演奏表情(たとえば、クレッシェンドやデクレッシェンド効果)を自動的に付加できるようにしたもの
が知られている。

【0006】

しかし、上記した従来の演奏データ作成装置又はパラメータ編集装置のうち(1)のものでは、修正したり追加したりすべきパラメータをユーザが自ら選択し、その値を決定しなければならず、特にユーザが初心者である場合には、好みの表情を付加するためのパラメータを選択し、その表情に最適なパラメータ値を決定することは困難であり、また、一々マニュアルで入力すること自体が煩わしい作業となる。

【0007】

また、上記した従来の演奏データ作成装置又はパラメータ編集装置のうち(2)のものでは、ユーザが指定可能な範囲は曲データ全体に対して局所的な一部範囲であるため、その一部範囲に属する演奏データに基づいて付加される演奏表情は簡単なもの(たとえば、当該部分の音量を線形的に増加または減少させるだけの演奏表情)とならざるを得ず、したがって、ユーザが望む多様な演奏表情を付加するには、演奏表情を付加すべき範囲を順次変更しながら同様の操作を何度も行わなければならず、簡単な操作で多様な演奏表情を付加することはできなかった。さらに、演奏表情を付加すべき範囲はユーザが自ら指定しなければならないため、ユーザは、どのような演奏個所にどのような表情を付加すれば、その曲にとって最適な表情付けとなるのかを知っている必要があり、初心者ユーザにとっては、困難な作業となる。

【発明の開示】

【発明が解決しようとする課題】

【0008】

この発明は、このような事情に鑑み、テンポやタイミング等の時間的变化により表情付けを行う場合の種々のルールや手順に着目し、これをモジュール化した新規な表情付け変換器を提供し、元の無表情な楽音データから音楽性のある多彩な演奏を可能とする演奏データに自動的に変換することができる演奏データ作成システム(パラメータ自動編集システム)を提供することを目的とする。

【0009】

この発明は、また、上述の各点に着目してなされたものであり、ユーザが初心者であっ

10

20

30

40

50

ても簡単な操作で曲に多様な表情付けを行うことが可能な演奏データ作成システム（パラメータ自動編集システム）を提供することを目的とする。

【0010】

この発明は、さらに、テンポやタイミング等の時間的変化パラメータ、音量パラメータの外、ピッチ等の音程パラメータなどの演奏に関する種々の制御パラメータを自動的に編集して、ユーザが初心者であっても、簡単な操作で曲に多種多様な表情付けを行うことが可能な演奏データ作成システム（パラメータ自動編集システム）を提供することを目的とする。

【課題を解決するための手段】

【0011】

この発明は、特に、演奏データの同音色パートに対する音量パラメータ制御によって簡単に同音色パートに効果的な表情付けを行うことができる演奏データ作成装置を提供するものであり、この発明の主たる特徴に従うと、音符情報及び音量を表わす楽音制御情報をそれぞれ含む複数のパートから成る演奏データを供給する演奏データ供給手段〔A B：図2〕と、演奏データから同音色のパート及びそのパート数を検出するパート検出手段〔S 1 4 1〕と、パート数に基づいて、上記同音色のパートの音量が減少するように当該各パートの音量値を算出する音量値算出手段〔S 1 4 2〕と、音量値を表わす楽音制御情報を該当パートに付与する音量値付与手段〔S 1 4 3〕とを具備する演奏データ作成装置が提供される〔実施例（N）：図75に対応〕。なお、括弧書きは、理解の便のために付記した実施例の対応記号・用語乃至参照箇所を示し、以下においても同様である。

【発明の効果】

【0012】

この発明の主たる特徴によると、音符情報及び音量を表わす楽音制御情報をそれぞれ含む複数のパートから成る演奏データ（元演奏データ）が供給されると〔A B：図2〕、演奏データから同音色のパート及びそのパート数が検出される〔S 1 4 1〕。そして、検出されたパート数に基づいて、検出された同音色パートの音量が減少するように当該各パートの音量値が算出され〔S 1 4 2〕、算出された音量値を表わす新たな楽音制御情報が該当パートに付与されて〔S 1 4 3〕、新たな楽音制御情報により表情付けされた演奏データが再生に利用される〔S B：図2〕。

【0013】

従って、この発明によると、供給された演奏データから、同音色のパート及びそのパート数を抽出し、抽出されたパートの音量を、抽出されたパート数に応じた値に減少させるような音量値の楽音制御情報を生成し、演奏データに付与するようにしているので、パートディビジョン（同音色のパートを複数のパートに分けて各パートを複数人数で演奏する演奏方法を模擬する）の際に、各パートの音量を下げるような表情付けを行うことができる。

【0014】

なお、この明細書において、「楽音制御量」とは、時間的楽音制御量や、音楽的楽音制御量、音量的制御量等、演奏上楽音を制御するための変量をいう。また、「楽音制御情報」とは、テンポやゲートタイム、発音開始タイミング等の時間的楽音制御情報（時間パラメータ）や、音程的楽音制御情報（音程パラメータ等）、音量的制御情報（音量パラメータ）など、演奏上楽音を制御するための変量情報をいい、「演奏パラメータ」或いは単に「パラメータ」ということがある。また、この発明による「演奏データ作成装置」は、演奏パラメータの編集という観点からいえば、「パラメータ自動編集装置」ということができる。

【0015】

〔別の特徴〕

また、この明細書に記載された種々の発明の実施に関する別の特徴により次のように構成することができる。

〔1〕演奏データを供給する供給手段と、供給された演奏データの特徴情報を取得する取

10

20

30

40

50

得手段と、所定の特徴情報に対応する情報であって、楽音制御情報を生成するための生成方法を表わす生成方法情報を、記憶する記憶手段と、取得された特徴情報、及び、この特徴情報に対応する生成方法情報から、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例では、図2及び図52, 53が対応〕：ここで、

〔1a〕この演奏データ作成装置は、さらに、付加手段により楽音制御情報が付加された演奏データを出力する出力手段と、出力された演奏データを評価し、評価結果に応じて、生成された楽音制御情報を調節する調節手段とを具備する〔実施例(1)~(24), (27)~(30)が対応〕。

【0016】

別の特徴〔1〕では、予め生成方法情報を記憶しておき、演奏データから得た特徴情報に対応する生成方法情報に基づいて、楽音制御情報を生成し、これを演奏データに付加する。つまり、所定の特徴情報と表情付けのための楽音制御情報との対応関係を表情付けモジュール(表情付けアルゴリズム)におけるルールとして設定され、この表情付けルールを表わす生成方法情報が記憶手段に予め記憶されており、供給された演奏データの特徴情報を取得すると、取得された特徴情報に対応する生成方法情報に基づき、表情付けモジュール(表情付けアルゴリズム)に従って、楽音制御情報(時間パラメータ、音程パラメータ、音量パラメータ等の各種演奏パラメータ)が生成され、演奏データに付加されるので、取得された特徴情報に応じて、ユーザが初心者であっても簡単な操作で曲に多様な表情付けを行うことができ、より音楽的な演奏データを自動的に作成することができる。また

【0017】

〔2〕演奏データを供給する供給手段と、供給された演奏データから、音符の発生時間間隔に相当する特徴情報を抽出する抽出手段と、音符の発生時間間隔に相当する特徴情報に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、抽出された特徴情報及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例(1), (3), (C)が対応〕：ここで、

〔2a〕特徴情報は、所定単位時間あたりの音符数を表わす情報であり、生成方法情報は、所定単位時間あたりの音符数が所定数を越えている場合に、演奏データの再生テンポの値を速くするような楽音制御情報を生成する生成方法を表わす情報であり〔実施例(1)に対応〕、

〔2b〕特徴情報は、音符の発音時間間隔を表わす情報であり、生成方法情報は、音符の発音時間間隔がせまい場合に、演奏データの再生テンポの値を遅くするような楽音制御情報を生成する生成方法を表わす情報であり〔実施例(3)に対応〕、或いは、

〔2c〕特徴情報は、所定単位時間あたりの音符数を表わす情報であり、生成方法情報は、所定単位時間あたりの音符数が所定数を越えている場合に、演奏データの音量を徐々に減少するような楽音制御情報を生成する生成方法を表わす情報である〔実施例(C)に対応〕。

【0018】

別の特徴〔2〕では、供給された演奏データから、音符の発生時間間隔に相当する特徴情報(音符時間情報)を抽出し、この特徴情報に対応する生成方法情報に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているので、音符時間情報(音符密度、2音間隔等)に基づいて多彩な表情のある演奏出力を得ることができる。

【0019】

ここで、別の特徴〔2a〕により、所定単位時間あたりの音符数を表わす音符密度情報(例えば、“1小節中の音符数÷1小節の拍数”)を特徴情報として抽出し、所定単位時間あたりの音符数が所定数を越えている場合には、演奏データの再生テンポの値を速くす

10

20

30

40

50

るように構成すると、音符密度に応じてテンポを変更する表情付け（音数増加によるテンポ加速）を行うことができる。なお、この場合、設定されたテンポ強弱値、音符密度とテーブルから求められるテンポ係数K及び現在の設定テンポ値に基づき算出した区間毎のテンポチェンジ量（現在の設定テンポ値××K）をMIDIデータに付与する構成をとることができる。また、元演奏データをディスプレイ上に表示させ、表示された元演奏データ上に付与されたパラメータ値やその位置を重ねて表示することで、付与結果を確認することもできる。さらに、複数パートから成る演奏データが供給される場合は、選択された所定パートの演奏データから抽出された音符密度情報に基づいて再生テンポを変更するようにしても、或いは、複数パートを総合的に評価した音符密度情報により再生テンポを変更するようにしてもよい。

10

【0020】

また、別の特徴〔2b〕により、音符の発音時間間隔を表わす情報を特徴情報として抽出し、音符の発音時間間隔がせまい場合に、演奏データの再生テンポの値を遅くするように構成すると、音符が細かいときにテンポを低速にする表情付けを行うことができる。

【0021】

さらに、別の特徴〔2c〕により、所定単位時間あたりの音符数を表わす情報を特徴情報として抽出し、所定単位時間あたりの音符数が所定数を越えている場合に、演奏データの音量を徐々に減少するように構成すると、演奏音の細かさに応じて音量を変化させる表情付けを行うことができる。例えば、音符密度が所定値以上である部分を難関部分として検出し、その部分に音量変化を付与する。さらに、難関部分の音程最大値と音程最小値の変化幅に応じて音量を変化させることもできる。この手法は、音程の不安定さを付与する表現付与にも適用可能であり、また、音符密度の高い部分におけるハンマリングオンの頻度を算出し、これに応じて演奏データにハンマリングオンを自動付加することで、音符が細かいとハンマリングオンするような表情を付与するのにも適用することができる。

20

【0022】

〔3〕演奏データを供給する供給手段と、演奏データの進行状態に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、供給された演奏データの進行状態及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例（2）、（B）が対応〕：ここで、

30

〔3a〕生成情報情報は、演奏データの進行に従って、演奏データの再生テンポを徐々に速くするような楽音制御情報を生成する生成方法を表わす情報であり〔実施例（2）に対応〕、或いは、

〔3b〕生成情報情報は、演奏データの進行に従って、音量を徐々に大きくするような楽音制御情報を生成する生成方法を表わす情報である〔実施例（B）に対応〕。

【0023】

別の特徴〔3〕では、供給された演奏データの進行状態に対応する生成方法情報に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているので、演奏データの演奏進行状態の評価に基づいて多彩な表情のある演奏出力を得ることができる。

【0024】

ここで、別の特徴〔3a〕により、演奏データの演奏進行に従って演奏データの再生テンポを徐々に速くするように構成すると、例えば、曲の進行につれて1小節毎に所定倍率でテンポをだんだん早くする等により、曲を通して徐々にテンポを加速する表情付けを行うことができる。この場合、具体的には、区間毎に、テンポ強弱値とテンポ係数Kと現在テンポ値とから区間毎のテンポチェンジ量を算出するのが有効である（テンポ係数Kは、テーブルから区間毎に求められ、曲の進行に従い徐々にテンポが増大する）。また、テンポ強弱値を興奮度と名付けて、「強い」、「中程度」、「弱い」、「変化なし」のような言葉で選択可能として更に有用性を高めることができる。

40

【0025】

また、別の特徴〔3b〕により、演奏データの進行に従って音量パラメータの値を徐

50

々に大きくするように構成すると、曲進行に伴い興奮度を高め盛り上がり具合を演出する表情付けを行うことができる。この場合、具体的には、例えば、曲全体の音量変化傾向を表わすパターンデータに基づいて、音量変化区間毎の変化値を算出し、演奏データの各対応位置に挿入する。この場合、複数トラック構成のシステムでは、各トラックで使用する変化付与パターンは、同一のものであってもよいし、異なるものであってもよい。なお、この手法は、音程パラメータに変化を付与する際にも適用可能であり、例えば、曲の進行に従い、音程パラメータの値を徐々に増大させて興奮度を高めることができる。

【 0 0 2 6 】

〔 4 〕演奏データを供給する供給手段と、供給された演奏データから、音長が所定長以上であり且つトリル又はビブラートが付与されている部分を抽出する抽出手段と、音長が所定長以上であり且つトリル又はビブラートが付与されている部分に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、抽出された部分及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例(4)〕、(G)が対応〕：ここで、

10

〔 4 a 〕生成方法情報は、トリル又はビブラートの経過時間に応じて、演奏データの再生テンポの値を最初はゆっくりで徐々に速くするような楽音制御情報を生成する生成方法を表わす情報であり〔実施例(4)に対応〕、或いは、

〔 4 b 〕生成方法情報は、音量を最初は大きく途中は小さく最後は大きくするような楽音制御情報を生成する生成方法を表わす情報である〔実施例(G)に対応〕。

20

【 0 0 2 7 】

別の特徴〔 4 〕では、供給された演奏データから、音長が所定長以上であり且つトリル又はビブラートが付与されている部分(微小震動音情報)を特徴情報として抽出し、この特徴情報に対応する生成方法情報に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているので、微小震動音情報(ロングトーンのトリル、ビブラート等)に基づいて多彩な表情のある演奏出力を得ることができる。

【 0 0 2 8 】

ここで、別の特徴〔 4 a 〕により、トリル又はビブラートの経過時間に応じて、演奏データの再生テンポの値を最初はゆっくりで徐々に速くするように構成すると、ロングトーンのトリル/ビブラート部分については最初はゆっくりのテンポとする表情付けを行うことができる。この場合、予め用意されたテーブルからトリル又はビブラートの時間経過に応じたテンポ係数Kを求め、その値とテンポ強弱値 および現在テンポ値とに基づいて区間毎のピッチベンド列を生成させるのが有効である。

30

【 0 0 2 9 】

また、別の特徴〔 4 b 〕により、音量を、最初は大きく、途中は小さく、最後は大きくするように構成すると、ロングトーンにトリルまたはビブラートが付与されているときの音量を大 小 大きくとなる変化させる表情付けを行うことができる。なお、この手法は、音程変化の速さを変化させることにも適用可能である。

【 0 0 3 0 】

〔 5 〕演奏データを供給する供給手段と、供給された演奏データから、ピッチベンドが付与されている部分を抽出する抽出手段と、ピッチベンドが付与されている部分に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、抽出された部分及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例(20)〕、(23)〕、(Q)が対応〕：ここで、

40

〔 5 a 〕生成方法情報は、ピッチベンドの変化に基づいて、演奏データの再生テンポの値を徐々に遅くするような楽音制御情報を生成する生成方法を表わす情報であり〔実施例(20)に対応〕、

〔 5 b 〕生成方法情報は、ピッチベンドが深く或いは速くかかっている場合に、演奏データの再生テンポの値を徐々に遅くするような楽音制御情報を生成する生成方法を表わす情

50

報であり〔実施例(23)に対応〕、或いは、
〔5c〕生成方法情報は、ピッチベンドの変化率が大きいほど、音量を小さくするような楽音制御情報を生成する生成方法を表わす情報である〔実施例(Q)に対応〕。

【0031】

別の特徴〔5〕では、供給された演奏データから、ピッチベンドの付与されている部分を特徴情報として抽出し、この特徴情報に対応する生成方法情報に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているため、微小震動音情報(ピッチベンド等)に基づいて多彩な表情のある演奏出力を得ることができる。

【0032】

ここで、別の特徴〔5a〕により、ピッチベンドの変化に基づいて、演奏データの再生テンポの値を徐々に遅くするように構成すると、例えば、検出されたピッチベンドに基づいたテンポチェンジを付与することにより、ピッチベンドをかけるときにテンポを触る表情付けを行うことができる。

10

【0033】

また、別の特徴〔5b〕により、ピッチベンドが深く或いは速くかかっている場合に、演奏データの再生テンポの値を徐々に遅くするように構成すると、ビブラートを深くかけたり、速くかけたりしたときにテンポを遅くするというように、ビブラートの深さや速さに応じてテンポを設定する表情付けを行うことができる。この場合、例えば、ビブラートを検出してその深さと速さからテーブルを引き、テンポ係数Kを算出し、この算出値に基づいたテンポチェンジを元の演奏データに付与するのが有効である。

20

【0034】

さらに、別の特徴〔5c〕により、ピッチベンドの変化率が大きいほど音量を小さくする(例えば、大きく下げる)ように構成すると、ピッチベンド変化時に音量を変化させる表情付けを行うことができる。

【0035】

〔6〕演奏データを供給する供給手段と、供給された演奏データからフレーズの区切り部分を抽出する抽出手段と、フレーズの区切り部分に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、抽出された部分及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例(5)、(F)〕、(f)が対応〕：ここで、

30

〔6a〕生成方法情報は、フレーズの区切り部分において、演奏データの再生テンポの値を遅くするような楽音制御情報を生成する生成方法を表わす情報であり〔実施例(5)に対応〕、

〔6b〕フレーズの区切り部分において、音量を徐々に小さくするような楽音制御情報を生成する生成方法を表わす情報であり〔実施例(F)に対応〕、或いは、

〔6c〕生成方法情報は、フレーズの区切り部分において、フレーズの開始位置にはダンパオンに関する情報を、フレーズの終端位置にはダンパオフに関する情報を挿入するような楽音制御情報を生成する生成方法を表わす情報である〔実施例(f)に対応〕。

【0036】

40

別の特徴〔6〕では、供給された演奏データから抽出したフレーズの区切り部分に対応する生成方法情報に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているため、フレーズの区切り(フレーズ終端部等)において多彩な表情のある演奏出力を得ることができる。

【0037】

ここで、別の特徴〔6a〕により、フレーズの区切り部分において演奏データの再生テンポの値を遅くする(例えば、元の演奏データからフレーズの切れ目位置を検出して、テンポが遅くなるテンポチェンジを元の演奏データに付与する)ように構成すると、フレーズ区間の終わり部分でテンポを遅くしフレーズ経過部でゆっくりとした表情付けを行うことができる。

50

【0038】

また、別の特徴〔6b〕により、フレーズの区切り部分において音量を徐々に小さくするように構成すると、フレーズの終止では音量を抑え、フレーズの終了位置で音量を徐々に小さくしていき、フレーズの終止感を与える表情付けを行うことができる。この場合、終止位置の音量は、フレーズに設定されたテンポに基づいて算出するのが好ましい。なお、フレーズの先頭位置の音符の音量を大きくするようにしてもよい。

【0039】

さらに、別の特徴〔6c〕により、フレーズの区切り部分において、フレーズの開始位置にはダンパオンに関する情報を挿入し、フレーズの終端位置にはダンパオフに関する情報を挿入する（例えば、フレーズを解釈してフレーズの区切りを検出し、フレーズの開始位置ではダンパーオン、フレーズの終端位置でダンパーオフとする）ように構成すると、ピアノのサスティンペダルによる演奏の幅を広げる表情付けを行うことができる。

【0040】

〔7〕演奏データを供給する供給手段と、供給された演奏データから、所定区間毎の平均音高情報又は平滑化された音高情報を算出する算出手段と、平均音高情報又は平滑化された音高情報に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、算出された平均音高情報又は平滑化された音高情報及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例（6）、（M）が対応〕：ここで、

〔7a〕生成方法情報は、平均音高情報又は平滑化された音高情報の音高が高いほど、演奏データの再生テンポの値を速くするような楽音制御情報を生成する生成方法を表わす情報である〔実施例（6）に対応〕。

【0041】

別の特徴〔7〕では、供給された演奏データから、所定区間毎の平均音高情報又は平滑化された音高情報を算出し、この音高情報に対応する生成方法情報に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているので、所定の音高情報に対応して多彩な表情のある演奏出力を得ることができる。この手法により、例えば、区間毎の音程平均値に基づいた音量変化を付与することができ〔実施例（M）を参照〕、また、この手法は音程についても適用可能である。

【0042】

ここで、別の特徴〔7a〕により、平均音高情報又は平滑化された音高情報の音高が高いほど、演奏データの再生テンポの値を速くするように構成すると、例えば、高音ほどテンポを速くするルールを採用し、音域によってテンポを変更する表情付けを行うことができる。具体的には、例えば、平均音高からテーブルを引いてテンポ係数Kを求め、その値とテンポ強弱値及び現在テンポ値とに基づいて区間毎のテンポチェンジを元の演奏データに付与するようにすればよく、また、音高変化をフィルタリングにより滑らかにし、フィルタリング後の音高変化カーブ上の各音高値及びテーブルからテンポ係数Kを求めてもよい。

【0043】

〔8〕演奏データを供給する供給手段と、供給された演奏データから、音高の変化傾向に相当する特徴情報を抽出する抽出手段と、音高の変化傾向に相当する特徴情報に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、抽出された特徴情報及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例（7）、（A）が対応〕：ここで、

〔8a〕特徴情報は、音高の上昇傾向と下降傾向との切替り位置を表わす情報であり、生成方法情報は、音高の上昇傾向と下降傾向との切替り位置において、演奏データの再生テンポの値を遅くするような楽音制御情報を生成する生成方法を表わす情報であり〔実施例（7）に対応〕、

10

20

30

40

50

〔 8 b 〕特徴情報は、音高の上昇傾向と下降傾向との切替り位置を表わす情報であり、生成方法情報は、音高の上昇傾向と下降傾向との切替り位置において、切替り位置の音符の音量にアクセントを付与するような楽音制御情報を生成する生成方法を表わす情報であり〔実施例（ A ）に対応〕、或いは、

〔 8 c 〕特徴情報は、音高の上昇傾向部分を表わす情報であり、生成方法情報は、上昇傾向部分において、音量を徐々に大きくするような楽音制御情報を生成する生成方法を表わす情報である〔実施例（ A ）に対応〕。

【 0 0 4 4 】

別の特徴〔 8 〕では、供給された演奏データから、音高の変化傾向に相当する音高変化傾向情報を特徴情報として抽出し、この音高変化傾向情報に対応する生成方法情報に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているので、音高変化傾向に応じて多彩な表情のある演奏出力を得ることができる。

10

【 0 0 4 5 】

ここで、別の特徴〔 8 a 〕により、音高変化傾向情報は音高の上昇傾向と下降傾向との切替り位置を表わし、音高の上昇傾向と下降傾向との切替り位置において、演奏データの再生テンポの値を遅くするように構成すると、音高の上昇部分から下降部分に切り替わる時に、いったん、テンポが遅くなるようにし、音高の上昇下降切り替わり時点に間をおく表情付けを行うことができる。具体的には、例えば、元の演奏データの音高変化カーブ及びテーブルを用いて、音高上昇部分の絡端近傍でテンポが徐々に遅くなるテンポチェンジを元の演奏データに付与する手法を採ることができる。

20

【 0 0 4 6 】

また、別の特徴〔 8 b 〕により、同様に音高の上昇傾向と下降傾向との切替り位置を表わす音高変化傾向情報を用いて、音高の上昇傾向と下降傾向との切替り位置において、切替り位置の音符の音量にアクセントを付与する（例えば、上昇から下降に変化したときには変化点のノートイベントにアクセントを付与する）ように構成すると、音符上昇部分の最後にアクセントをもたせる表情付けを行うことができる。

【 0 0 4 7 】

また、別の特徴〔 8 c 〕により、供給された演奏データから、音高の上昇傾向部分を表わす音高変化傾向情報を特徴情報として用い、上昇傾向部分において、音量を徐々に大きくするように構成すると、ノートイベント列の音高が上昇傾向にあるときには音量を徐々に増大させる表情付けを行うことができる。具体的には、例えば、演奏データ中の解析区間をユーザが設定し、解析区間からノートイベント列の音高が上昇および下降傾向にある部分区間を検索し（この場合、全体として上昇傾向にある部分を上昇傾向部分とする。）検索された部分区間毎に音高の変化速度を算出し、その結果に応じて、ノートイベントに付与する音量パラメータの変化付与パターンを決定し、決定された変更付与パターンに基づき、部分区間内の音量パラメータ値を変更する手法が採られる。なお、この手法は、音程パラメータに変化を付与する際にも適用可能である。

30

【 0 0 4 8 】

〔 9 〕演奏データを供給する供給手段と、供給された演奏データから、同一又は類似するデータ列が連続して存在する部分を抽出する抽出手段と、同一又は類似するデータ列が連続して存在する部分に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、抽出された特徴情報及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備することを特徴とする演奏データ作成装置〔実施例（ 8 ）,（ D ）に対応〕：ここで、

40

〔 9 a 〕生成方法情報は、連続して存在する同一又は類似するデータ列の後側のデータ列に対して、演奏データの再生テンポの値を変更するような楽音制御情報を生成する生成方法を表わす情報であり〔実施例（ 8 ）に対応〕、或いは、

〔 9 b 〕生成方法情報は、連続して存在する同一又は類似するデータ列の後側のデータ列に対して、音量を類似度に応じて変更するような楽音制御情報を生成する生成方法を表わ

50

す情報である〔実施例（D）に対応〕。

【0049】

別の特徴〔9〕では、供給された演奏データから、同一又は類似するデータ列が連続して存在する部分を特徴情報として抽出し、この特徴情報に対応する生成方法情報に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているのので、同一又は類似するデータ列が連続して存在する場合に多彩に表情付けされた演奏出力を得ることができる。

【0050】

ここで、別の特徴〔9a〕により、連続して存在する同一又は類似するデータ列の後側のデータ列に対して、演奏データの再生テンポの値を変更するように構成すると、同一あるいは類似パターンが連続したときに、2回目のパターンについてはテンポを遅くあるいは速くする表情付けを行うことができる。

10

【0051】

また、別の特徴〔9b〕により、同様に連続して存在する同一又は類似するデータ列の後側のデータ列に対して、音量を類似度に応じて変更するように構成すると、同一あるいは類似パターンが連続して出現する場合に、パターンの類似性に応じて音量パラメータを変化させる表情付けを行うことができる。例えば、類似フレーズが繰り返し出現する場合に、2番目以降の類似フレーズの音量パラメータを、その類似度および出現態様に応じて変化させる。この場合、類似度が高いフレーズが連続して出現する場合には、2番目以降の類似フレーズの音量パラメータ値を、最初の類似フレーズのそれより小さくする。また、類似フレーズが連続せずに繰り返し出現する場合には、最初に出現する類似フレーズのそれに類似した値であって、その類似度に応じた値に変更する。なお、この手法は、音程パラメータに変化を付与する表情付けに適用可能である。

20

【0052】

〔10〕演奏データを供給する供給手段と、供給された演奏データから、類似するデータ列を抽出する抽出手段と、類似するデータ列の相違点に基づいて、演奏データの再生テンポの値を変更するような楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、抽出されたデータ列及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例（9）に対応〕：

30

【0053】

別の特徴〔10〕では、供給された演奏データから、類似するデータ列を抽出し、類似するデータ列の相違点に基づいて、演奏データの再生テンポの値を変更するようにしているので、似たフレーズは同一乃至類似のテンポを与える表現付けを行うことができる。具体的には、例えば、類似フレーズ間の相違点を検出して、その相違点に基づくテンポチェンジを元の演奏データに付与する。

【0054】

〔11〕演奏データを供給する供給手段と、供給された演奏データから、予め登録されている音形を抽出する抽出手段と、音形に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、抽出された音形及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例（10）に対応〕：

40

【0055】

別の特徴〔11〕では、供給された演奏データから、予め登録されている音形を抽出し、この音形に対応する生成方法情報に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているのので、登録音形に対応したテンポを設定する表情付けを行うことができる。具体的には、例えば、予め登録されている音形＝リズムパターン（フレーズ）の部分に対して所定のテンポを設定する。登録するフレーズは音形だけでなく、エクスペリションイベント列などでもよい。なお、テンポ変更だけでなく、音程パラメータの変更についても同様に適用することができる。また、この方法は、実施例で述べるように、楽

50

器の種類に応じたスラー付与を行うこともできる。

【 0 0 5 6 】

〔 1 2 〕演奏データを供給する供給手段と、供給された演奏データから、所定区間毎の平均音量情報又は平滑化された音量情報を算出する算出手段と、平均音量情報又は平滑化された音量情報に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、算出された所定区間毎の平均音量情報又は平滑化された音量情報及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例(1 1)が対応〕：

【 0 0 5 7 】

別の特徴〔 1 2 〕では、供給された演奏データから、所定区間毎の平均音量情報又は平滑化された音量情報を算出し、この音量情報に対応する生成方法情報に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているので、例えば、音量が小さいほどテンポを遅くするといった表情付けを行うことができる。具体的には、例えば、区間毎に平均音量値を求め、予め用意した音量が小さいほどテンポが遅くなるようなテーブルに基づいて、所定のテンポチェンジを元の演奏データに付与する。また、音量変化をフィルタリングした音量変化カーブ上の各音量値を基にしてもよい。

【 0 0 5 8 】

〔 1 3 〕演奏データを供給するデータ供給手段と、緊張感パラメータを入力するパラメータ入力手段と、緊張感パラメータに対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、供給された演奏データ、入力された緊張感パラメータ及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備することを特徴とする演奏データ作成装置〔実施例(1 2)が対応〕：

【 0 0 5 9 】

別の特徴〔 1 3 〕では、パラメータ入力手段を用いて緊張感パラメータを入力し、この緊張感パラメータに対応する生成方法情報に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているので、曲全体の感じ或いはサビなどの特定部の感じに沿って、緊張感というパラメータを入力し、このような緊張感パラメータに従ってテンポを変化する表情付けを行うことができる。このように緊張感でテンポを変化するには、例えば、操作子を用いて緊張感パラメータを設定し、設定された緊張感パラメータ値から対応するテンポ係数Kを生成し、生成されたテンポ係数K及び現在テンポ値に応じたテンポチェンジを元の演奏データに付与する。

【 0 0 6 0 】

〔 1 4 〕演奏データを供給する供給手段と、供給された演奏データから所定の音符群列を抽出する抽出手段と、所定の音符群列に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、抽出された音符群列及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例(1 3)、(1 4)、(E)が対応〕：ここで、

〔 1 4 a 〕生成方法情報は、音符群列の最初の音符長を長めにし、残りの音符長を音符群列の時間長に収まる長さ修正するような楽音制御情報を生成する生成方法を表わす情報であり〔実施例(1 3)に対応〕、或いは、

〔 1 4 b 〕生成方法情報は、音符群列を複数の群列に分割し、各群列の最初の音符長を長めにするような楽音制御情報を生成する生成方法を表わす情報であり〔実施例(1 4)に対応〕、或いは、

〔 1 4 c 〕生成方法情報は、音符群列の最初のタイミングに出現する音符の音量を強調するような楽音制御情報を生成する生成方法を表わす情報である〔実施例(E)に対応〕。

【 0 0 6 1 】

別の特徴〔 1 4 〕では、供給された演奏データから所定の音符群列を抽出し、この音符

10

20

30

40

50

群列に対応する生成方法情報に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているため、まとまった音符群に対して多彩な表情のある演奏出力を得ることができる。

【0062】

ここで、別の特徴〔14a〕により、音符群列の最初の音符長を長めにし、残りの音符長を音符群列の時間長に収まる長さに修正するように構成すると、まとまった音符群に対して所定の時間的表現を付与することができる。例えば、3拍子の4分音符の羅列のような、何らかのブロックでまとめられる音符群の最初の音符を長めにし、使った時間を残りの音符で取り戻すような表現を付与する。このためには、例えば、音符群の群感の強さというパラメータを設定し、このパラメータ値に基づいてテンポチェンジを元の演奏データに付与する。なお、ピアノの左手だけ、もしくは、伴奏パートだけに適用するのも有効である。

10

【0063】

また、別の特徴〔14b〕により、音符群列を複数の群列に分割し、各群列の最初の音符長を長めにするように構成すると、連符中における所定の音を強調する表情付けを行うことができる。つまり、検出した連符を複数の小連符に分割し、分割された小連符に対応したテンポチェンジを演奏データに付与する。この場合、5連符以上の長い連符を、例えば、“2連符+3連符”又は“3連符+2連符”のように、小連符に分割し、小連符の1つ目の音を時間をかけて強調する。さらに、最初の小連符はテンポを遅め、残りの小連符はテンポを早めにするなどの手法も有効である。

20

【0064】

さらに、別の特徴〔14c〕により、音符群列の最初のタイミングに出現する音符の音量を強調するように構成すると、音符群列に対して音量強調による表情付けを行うことができる。例えば、3拍子の演奏データの場合、同一音符長の音符が連続して出現するときには1拍目を強調し、この際、先頭拍位置の音符数に応じて音量変化を補正する。さらに、連符を検出して、その最初の音の音量を大きくすることにより連符らしさを表現することもできる。また、5拍子の場合には、2拍子と3拍子が交互に並んでいると考えて、それぞれの1拍目の音量を強調する。連符についても同様に分解して考えることができる。

【0065】

〔15〕演奏データを供給する供給手段と、供給された演奏データから、同時に複数音が発音される部分を抽出する抽出手段と、同時に複数音が発音される部分に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、抽出された部分及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例(15)、(J)に対応〕：ここで、

30

〔15a〕生成方法情報は、同時に発音される音の数に応じて、演奏データの再生テンポの値を変更するような楽音制御情報を生成する生成方法を表わす情報であり〔実施例(15)に対応〕、或いは、

〔15b〕生成方法情報は、同時に発音される音のそれぞれの重要度を規定し、規定された重要度に応じて、各音の音量を変更するような楽音制御情報を生成する生成方法を表わす情報である〔実施例(J)に対応〕。

40

【0066】

別の特徴〔15〕では、供給された演奏データから、同時に複数音が発音される部分を抽出し、この部分に対応する生成方法情報に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているため、同時に複数音が発音される部分に対して多彩な表情のある演奏出力を得ることができる。

【0067】

ここで、別の特徴〔15a〕により、同時に発音される音の数に応じて、演奏データの再生テンポの値を変更する〔例えば、和音における同時発音数(和音の音符密度)に応じてテンポを遅くする〕ように構成すると、和音構成音の数によってテンポを変更する表情

50

付けを行うことができる。

【 0 0 6 8 】

また、別の特徴〔 1 5 b 〕により、同時に発音される音のそれぞれの重要度を規定し、規定された重要度に応じて、各音の音量を変更するように構成すると、複数音同時発音演奏において音量による効果的な表情付けを行うことができる。例えば、重要度について予めテンプレートで用意しておき、和音の構成音の音量を、重要度の高い音の音量を大きくするきいうように、重要度に応じて変更する。この場合、単に、和音の最低音と最高音の音量のみを、他の構成音の音量より大きくしてもよいし、和音の根音の音量のみを大きくしてもよい。さらに、オクターブユニゾンについても同様に適用することができる。なお、この手法を音程について適用し、和音を自動的に純正調に変えることもできる。

10

【 0 0 6 9 】

〔 1 6 〕演奏データを供給する供給手段と、供給された演奏データから音色情報を抽出する抽出手段と、音色情報に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、抽出された音色情報及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例（ 1 6 ）,（ P ）が対応〕：ここで、

〔 1 6 a 〕生成方法情報は、音色種類又は音色パラメータに応じて、演奏データの再生テンポの値を変更するような楽音制御情報を生成する生成方法を表わす情報であり〔実施例（ 1 6 ）に対応〕、或いは、

20

〔 1 6 b 〕生成方法情報は、予め規定されている音色種類毎の音量に基づいて、音色の変更位置で、音量が変更前の音色に対応する音量から、変更後の音色に対応する音量に滑らかに変更するような楽音制御情報を生成する生成方法を表わす情報である〔実施例（ P ）に対応〕。

【 0 0 7 0 】

別の特徴〔 1 6 〕では、供給された演奏データから音色情報を抽出し、この音色情報に対応する生成方法情報に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているので、演奏データ中の所定音色情報に対して多彩な表情のある演奏出力を得ることができる。

【 0 0 7 1 】

ここで、別の特徴〔 1 6 a 〕により音色種類又は音色パラメータに応じて、演奏データの再生テンポの値を変更するように構成すると、音色が変わる度に自動的にテンポが微妙に変化する表現を得ることができる。

30

【 0 0 7 2 】

また、別の特徴〔 1 6 b 〕により、予め規定されている音色種類毎の音量に基づいて、音色の変更位置で、音量が変更前の音色に対応する音量から、変更後の音色に対応する音量に滑らかに変更するように構成すると、音色変化に応じた音量表現による表情付けを行い、特定の音色による演奏を目立たせることができる。。

【 0 0 7 3 】

〔 1 7 〕演奏データを供給する供給手段と、供給された演奏データから、フィンガリングに関する情報を抽出する抽出手段と、フィンガリングに関する情報に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、抽出されたフィンガリングに関する情報及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例（ 1 7 ）～（ 1 9 ）,（ S ）が対応〕：ここで、

40

〔 1 7 a 〕生成方法情報は、演奏しにくい部分に相当するフィンガリングに関する情報を規定し、当該演奏しにくい部分において、演奏データの再生テンポの値を遅くするような楽音制御情報を生成する生成方法を表わす情報であり〔実施例（ 1 7 ）に対応〕、

〔 1 7 b 〕生成方法情報は、ポジション移動部分に相当するフィンガリングに関する情報を規定し、当該ポジション移動部分において、演奏データの再生テンポの値にゆらぎを与

50

えるような楽音制御情報を生成する生成方法を表わす情報であり〔実施例(18)に対応〕、

〔17c〕生成方法情報は、低いポジションで演奏する部分に相当するフィンガリングに関する情報を規定し、当該低いポジションで演奏する部分において、演奏データの再生テンポの値を遅くするような楽音制御情報を生成する生成方法を表わす情報であり〔実施例(19)に対応〕、

〔17d〕生成方法情報は、演奏しにくい部分に相当するフィンガリングに関する情報を規定し、当該演奏しにくい部分において、音量を小さくするような楽音制御情報を生成する生成方法を表わす情報であり〔実施例(S)に対応〕、或いは、

〔17e〕生成方法情報は、ポジション移動部分に相当するフィンガリングに関する情報を規定し、当該ポジション移動部分において、音程を変更するような楽音制御情報を生成する生成方法を表わす情報である〔実施例(S)に対応〕。

【0074】

別の特徴〔17〕では、供給された演奏データから、フィンガリングに関する情報を抽出し、フィンガリングに関する情報に対応する生成方法情報に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているため、フィンガリングに関して多彩な表情のある演奏出力を得ることができる。

【0075】

ここで、別の特徴〔17a〕により、演奏しにくい部分に相当するフィンガリングに関する情報を規定し、当該演奏しにくい部分において、演奏データの再生テンポの値を遅くするように構成すると、動きにくい指(演奏しにくい指)での演奏のテンポを遅くするように、演奏指に応じてテンポを変更する表情付けを行うことができる。このためには、例えば、フィンガリングの指を検出して、予め用意したテーブルを用いてテンポチェンジを算出する。

【0076】

また、別の特徴〔17b〕により、ポジション移動部分に相当するフィンガリングに関する情報を用い、当該ポジション移動部分において演奏データの再生テンポの値にゆらぎを与えるように構成すると、低いポジションほど速い演奏に向かない傾向を再現するために、低ポジションでテンポを減速する表情付けを行うことができる。

【0077】

また、別の特徴〔17c〕により、低いポジションで演奏する部分に相当するフィンガリングに関する情報を用い、当該低いポジションで演奏する部分において演奏データの再生テンポの値を遅くするように構成すると、フィンガリングのポジションに応じてテンポ係数を設定することによって、人間が演奏した不均一さを表現することができる。

【0078】

さらに、別の特徴〔17d〕により、演奏しにくい部分に相当するフィンガリングに関する情報を用い、当該演奏しにくい部分において音量を小さくするように構成すると、フィンガリングに応じて、演奏操作が困難と考えられる音高の音量を、他の音高の音量よりも相対的に小さくするといった表情付けを行うことができる。

【0079】

さらに、別の特徴〔17e〕により、ポジション移動部分に相当するフィンガリングに関する情報を用い、当該ポジション移動部分において音程を変更するように構成すると、フィンガリングのポジション移動に応じて音程を自動変化させる表情付けを行うことができる。なお、フィンガリングを考え、低いポジションで大きな音量のときには、ノイズ音を加える手法も適用可能である。

【0080】

〔18〕演奏データを供給する供給手段と、供給された演奏データから、特定の楽器奏法に相当する部分を抽出する抽出手段と、特定の楽器奏法に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、抽出された特定の楽器奏法に相当する部分及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成

10

20

30

40

50

された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例(21)、(22)、(24)、(H)、(T)が対応〕：ここで、〔18a〕特定の楽器奏法は、ハンマリングオン奏法又はプリングオフ奏法であり、生成方法情報は、ハンマリングオン奏法又はプリングオフ奏法にあたる部分における、演奏データの再生テンポの値を速くするような楽音制御情報を生成する生成方法を表わす情報であり〔実施例(21)に対応〕、

〔18b〕特定の楽器奏法は、ピアノのサスティンペダル奏法であり、生成方法情報は、ピアノのサスティンペダル奏法にあたる部分における、演奏データの再生テンポの値を遅くするような楽音制御情報を生成する生成方法を表わす情報であり〔実施例(22)に対応〕、

〔18c〕特定の楽器奏法は、ストリングのトリル奏法であり、生成方法情報は、ストリングのトリル奏法にあたる部分における演奏データを複数のパートに分割し、各パート毎に異なる、演奏データの再生テンポの値を設定するような楽音制御情報を生成する生成方法を表わす情報であり〔実施例(24)に対応〕、

〔18d〕特定の楽器奏法は、トリルやドラムのロール奏法であり、生成方法情報は、トリルやドラムのロール奏法にあたる部分の音符の音量を不均一にするような楽音制御情報を生成する生成方法を表わす情報であり〔実施例(H)に対応〕、或いは、

〔18e〕特定の楽器奏法は、弦楽器の弓の返し奏法であり、生成方法情報は、弦楽器の弓の返し奏法にあたる部分の近傍音符の音量を変更するような楽音制御情報を生成する生成方法を表わす情報である〔実施例(T)に対応〕。

【0081】

別の特徴〔18〕では、供給された演奏データから、特定の楽器奏法に相当する部分を抽出し、特定の楽器奏法に対応する生成方法情報に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているので、特定の楽器奏法に対応して多彩な表情のある演奏出力を得ることができる。

【0082】

ここで、別の特徴〔18a〕により、特定の楽器奏法がハンマリングオン奏法又はプリングオフ奏法であり、このハンマリングオン奏法又はプリングオフ奏法にあたる部分において、演奏データの再生テンポの値を速くするように構成すると、ギターの手マリングオン又はプリングオフなどのときにテンポを速くする表情付けを行うことができる。このようにプリングオフ時にテンポを加速するには、ハンマリングオン、プリングオフを検出し、検出結果に基づいたテンポチェンジを演奏データに付与する。

【0083】

また、別の特徴〔18b〕により、特定の楽器奏法がピアノのサスティンペダル奏法であり、ピアノのサスティンペダル奏法にあたる部分において、演奏データの再生テンポの値を遅くするように構成すると、ピアノのサスティンペダル操作に応じて微妙にテンポを遅くするというように、サスティンペダル操作に応じてテンポを設定する表情付けを行うことができる。これには、例えば、サスティンペダルオンを検出し、この検出結果に基づいたテンポチェンジを演奏データに付与する。

【0084】

また、別の特徴〔18c〕により、特定の楽器奏法がストリングのトリル奏法であり、ストリングにおいて微小変動(震動)音を持続するトリル奏法にあたる部分の演奏データを複数のパートに分割し、演奏データの再生テンポを各パート毎に異なる値が設定されるように構成すると、ストリングのトリルについて、複数パートを使いこれらのパートについて微妙にタイミングをずらすようにする表情付けを行うことができる。これには、例えば、トリル部分を検出してその部分を複数パートにコピーし、パート毎にMIDIデータのタイミングが異なるように変更する。この場合、パート毎に異なる音色特性とすることもできる。なお、この手法は、音程パラメータの変化付与にも適用でき、この場合は、分解したそれぞれのパートにおけるトリル音程を微妙にずらすようにすればよい。

【0085】

さらに、別の特徴〔18d〕により、特定の楽器奏法がトリルやドラムのロール奏法であり、トリルやドラムのロール奏法にあたる部分の音符の音量を不均一にするように構成すると、トリルや打楽器のロール演奏部分に、個々の音符の音量を不均一になる値とする表情付けを行うことができる。

【0086】

さらに、別の特徴〔18e〕により、特定の楽器奏法が弦楽器の弓の返し奏法であり、弦楽器の弓の返し奏法にあたる部分の近傍音符の音量を変更するように構成すると、弦楽器の弓の返し位置近傍に音量変化を付与する表情付けを行うことができる。

【0087】

〔19〕演奏データを供給する供給手段と、供給された演奏データから歌詞情報を抽出する抽出手段と、歌詞情報に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、抽出された歌詞情報及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備することを特徴とする演奏データ作成装置〔実施例(27)、(U)が対応〕：ここで、

〔19a〕生成方法情報は、特定の単語に対するテンポ制御値を規定しており、規定した内容に基づいて、演奏データの再生テンポの値を変更するような楽音制御情報を生成する生成方法を表わす情報であり〔実施例(27)に対応〕、或いは、

〔19b〕生成方法情報は、特定の単語に対する音量変化を規定しており、規定した内容に基づいて、音量を変更するような楽音制御情報を生成する生成方法を表わす情報である〔実施例(U)に対応〕。

【0088】

別の特徴〔19〕では、供給された演奏データから歌詞情報を抽出し、歌詞情報に対応する生成方法に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているので、歌詞情報がある場合に多彩な表情のある演奏出力を得ることができる。

【0089】

ここで、別の特徴〔19a〕により、特定の単語に対するテンポ制御値に基づいて、演奏データの再生テンポの値を変更するように構成すると、歌詞に応じたテンポを設定する表情付けを行うことができる。この表情付けには、例えば、所定の単語をテンポ係数と共に予め登録しておき、元の演奏データの歌詞データ中から所定の単語を検出するようにし、当該単語が出現すると、それに基づいたテンポチェンジを演奏データに付与してテンポを変化させる。この場合、明るい単語には速いテンポを設定し、暗い単語や重要単語には遅いテンポを設定する。

【0090】

また、別の特徴〔19b〕により、特定の単語に対する音量変化に基づいて音量を変更するように構成すると、所定の単語に音量変化を付与する表情付けを行うことができる。このような歌詞情報に伴うパラメータの処理手法は、音程変化についても適用することができる。

【0091】

〔20〕演奏データを供給する供給手段と、供給された演奏データから、音源からの出力波形に関する波形情報を取得する取得手段と、音色の種類に応じた音源からの出力波形に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、取得された波形情報及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例(28)に対応〕：

【0092】

別の特徴〔20〕では、供給された演奏データから、音源からの出力波形に関する情報を取得し、音色の種類に応じた音源からの出力波形に対応する生成方法情報に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているので、例えば、実際の出力波形を観測しながら、音色の種類によって音符のオン時刻を異なるように補正するように

10

20

30

40

50

して、音源の出力波形に応じてタイミングを補正する表情付けを行うことができる。

【 0 0 9 3 】

〔 2 1 〕演奏データを供給する供給手段と、供給された演奏データから、演奏記号に関する情報を抽出する抽出手段と、演奏記号に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段と、抽出された演奏記号に関する情報及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例（ 2 9 ）,（ 3 0 ）,（ K ）が対応〕：ここで、

〔 2 1 a 〕演奏記号に関する情報は、ピアノ記号とフォルテ記号であり、生成方法情報は、ピアノ記号が添付された音符の直前の音符にフォルテ記号が添付されている場合に、当該フォルテ記号が添付されている音符の発音長を短めにするような楽音制御情報を生成する生成方法を表わす情報であり〔実施例（ 2 9 ）に対応〕、

〔 2 1 b 〕演奏記号に関する情報は、スタッカート記号であり、生成方法情報は、スタッカート記号が添付された音符の直前の音符の発音長を変更するような楽音制御情報を生成する生成方法を表わす情報であり〔実施例（ 3 0 ）に対応〕、或いは、

〔 2 1 c 〕演奏記号に関する情報は、スタッカート記号であり、生成方法情報は、スタッカート記号が添付された音符の直後の音符の音量を小さくするような楽音制御情報を生成する生成方法を表わす情報である〔実施例（ K ）に対応〕。

【 0 0 9 4 】

別の特徴〔 2 1 〕では、供給された演奏データから、演奏記号に関する情報を抽出し、演奏記号に対応する生成方法情報に基づいて、楽音制御情報を生成し、演奏データに付加するようにしているので、演奏データ中の演奏記号に対応して多彩な表情のある演奏出力を得ることができる。

【 0 0 9 5 】

ここで、別の特徴〔 2 1 a 〕により、演奏記号がピアノ記号とフォルテ記号であり、ピアノ記号が添付された音符の直前の音符にフォルテ記号が添付されている場合に、当該フォルテ記号が添付されている音符の発音長を短めにするように構成すると、ピアノ記号直前のフォルテ記号は短めに設定して発音させる表情付けを行うことができる。

【 0 0 9 6 】

また、別の特徴〔 2 1 b 〕により、演奏記号がスタッカート記号であり、スタッカート記号が添付された音符の直前の音符の発音長を変更するように構成すると、スタッカート直前の音の発音時間を長めにする表情付けを行うことができる。これには、例えば、スタッカートを検出し、その直前の音のゲートタイムを強弱値 に応じて伸長し、その結果に基づいたテンポチェンジを演奏データに付与する。

【 0 0 9 7 】

さらに、別の特徴〔 2 1 c 〕により、同様に演奏記号がスタッカート記号であり、スタッカート記号が添付された音符の直後の音符の音量を小さくするように構成すると、スタッカート音を強調するために、スタッカート発音の直後の音符の音量を小さくする表情付けを行うことができる。このようなスタッカート演奏においては、音量を変化させる度合いをスタッカート音符の音価やテンポに応じて調整することが好ましい。

【 0 0 9 8 】

〔 2 2 〕演奏データを供給する供給手段と、既に供給されている演奏データの所定の特徴情報と楽音制御情報との関係を記憶する記憶手段と、新たに供給された演奏データの特徴情報を抽出する抽出手段と、抽出された特徴情報に基づき、記憶手段に記憶された関係に従って楽音制御情報を生成する生成手段と、生成された楽音制御情報を、新たに供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例（ 2 5 ）が対応〕：

【 0 0 9 9 】

別の特徴〔 2 2 〕では、既に供給されている演奏データの所定の特徴情報と楽音制御情報との関係を記憶しておき、新たに供給された演奏データの特徴を抽出すると、記憶手段

10

20

30

40

50

に記憶された関係に従って、楽音制御情報を生成し、新たに供給された演奏データに付加するようにしているので、学習機能によりテンポを設定する表情付けを行うことができる。この手法では、音高変化とテンポ変化の関係を自動的に予想するシステムを構成し、ある曲の途中までテンポ変化を手入力した後の残りのテンポ変化を学習機能により自動入力させる。例えば、学習ルーチンでテンポ変化を既に付与しているMIDIデータからフレーズとテンポ変化の関係を学習させ、その結果を記憶しておく。そして、未だテンポの付与されていないフレーズのMIDIデータに対し、記憶されている学習結果を基にしたテンポチェンジを演奏データに付与する。

【0100】

〔23〕演奏データに関して所定の特徴情報と楽音制御情報との関係を複数記憶したライブラリと、演奏データを供給する供給手段と、供給された演奏データの特徴情報を抽出する抽出手段と、抽出された特徴情報に基づき、ライブラリを参照することによって楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例(26)が対応〕：

10

【0101】

別の特徴〔23〕では、演奏データに関して所定の特徴情報と楽音制御情報との関係をライブラリに記憶しておき、供給された演奏データの特徴情報を抽出すると、ライブラリを参照することによって、楽音制御情報を生成し、演奏データに付加するようにしている。ライブラリを用いてテンポを設定する表情付けを行うことができる。この手法においては、演奏データの種々の特徴情報に対応して一旦生成したテンポ変化をある時間区間で切り取ってライブラリ化しておき、他の部分にも同様に適応する。例えば、ライブラリ化ルーチンでテンポ変化を既に付与したMIDIデータからテンポチェンジを抽出して相対値に直してライブラリ化しておく。そして、MIDIデータの所定の特徴情報に対応してこのライブラリからテンポチェンジを選択し、選択されたテンポチェンジを時間方向、テンポ値方向に伸縮した上、演奏データに付与する。なお、この手法は、音程パラメータの変化付与にも適用することができる。

20

【0102】

〔24〕演奏データを供給する供給手段と、供給された演奏データの所定の特徴情報に基づき楽音制御情報を生成する生成手段と、生成された楽音制御情報及び供給された演奏データの楽音制御情報を演奏データ全体で対比する対比手段と、この対比結果に基づいて、生成された楽音制御情報を修正する修正手段とを具備する演奏データ作成装置〔実施例(31)、(V)が対応〕：

30

【0103】

別の特徴〔24〕では、供給された演奏データの所定の特徴情報に基づき楽音制御情報を生成し、生成された楽音制御情報及び供給された演奏データの楽音制御情報を演奏データ全体で対比し、この対比結果に基づいて、生成された楽音制御情報を修正するようにしている。演奏データ全体を見通して、バランスのとれた最適な表情のある演奏出力を得ることができる。例えば、1曲を通してテンポ変化させた結果を見て、テンポの平均が元々設定されていたテンポ値になるように、全体のテンポを補正する(全体のテンポを均一に補正したり、テンポ変更回数の多い区間のテンポを優先的に補正したり)ことによって、全体を見通してテンポを決めることができる〔実施例(31)を参照〕。また、演奏データ全体の音量の平均値を算出し、この平均値が所望値になるように、全体の音量にオフセットを加えることもできる〔実施例(V)を参照〕。

40

【0104】

〔25〕演奏データを供給する供給手段と、供給された演奏データから、発音を指示する演奏データであってその発音長が所定長以上の部分を抽出する抽出手段と、発音長が所定長以上の部分に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段であって、この生成方法情報は、発音長が所定長以上の部分の音量を不均一に変更するような楽音制御情報を生成方法を表わすものである記憶手段と、抽出された部分及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音

50

制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例(L)が対応〕：

【0105】

別の特徴〔25〕では、供給された演奏データから、発音を指示する演奏データであってその発音長が所定長以上の部分を抽出し、所定長以上の発音指示データ部分に対応する生成方法情報に基づいて、発音長が所定長以上の部分の音量を不均一に変更するような楽音制御情報を生成し、演奏データに付加するようにしているため、ロングトーンの音量にゆらぎ/ランダム性をつける表情付けを行うことができる。この場合、ゆらぎについては、乱数カウンタと所定の変化付与パターンに基づき決定するのが好ましい。なお、この手法は、音程パラメータについても適用可能である。

10

【0106】

〔26〕演奏データを供給する供給手段と、供給された演奏データからメロディ部分を抽出する抽出手段と、メロディ部分に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段であって、この生成方法情報は、メロディ部分の音量をその他の部分の音量よりも大きくなるように変更するような楽音制御情報を生成する生成方法を表わすものである記憶手段と、抽出されたメロディ部分及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備することを特徴とする演奏データ作成装置〔実施例(R)が対応〕：

【0107】

別の特徴〔26〕では、供給された演奏データからメロディ部分を抽出し、メロディ部分に対応する生成方法情報に基づいて、抽出されたメロディ部分の音量をその他の部分の音量よりも大きくなるように変更するような楽音制御情報を生成し、演奏データに付加するようにしているため、パートに応じた表情付けを行うことができる。つまり、メロディ部分の音量が相対的に他の部分の音量より上がるように変化量を決定し、メロディラインを浮き立たせる。なお、この手法は、音程パラメータについても適用可能であり、メロディ部分や伴奏パートに応じて基本ピッチをかえるようにする。さらに、パートに応じて遠近感を変化させたり、リバーブの深さを変化させたりする等に応用することができる。

20

【0108】

〔27〕演奏データを供給する供給手段と、供給された演奏データから、音量変化が付与された部分を抽出する抽出手段と、音量変化が付与された部分に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段であって、この生成方法情報は、付与された音量変化に相当する音程変化を、音量変化が付与された部分に付与するような楽音制御情報を生成する生成方法を表わすものである記憶手段と、抽出された部分及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例(a)が対応〕：

30

【0109】

別の特徴〔27〕では、供給された演奏データから、音量変化が付与された部分を抽出し、音量変化付与部分に対応する生成方法情報に基づいて、抽出された音量変化に相当する音程変化を、抽出された部分に付与するような楽音制御情報を生成し、演奏データに付加するようにしているため、音量変化(アクセント)が付与された部分に対して、音量変化に相当する音程変化を決定し、アクセント音を微妙に上ずらせる表情付けを行うことができる。

40

【0110】

〔28〕演奏データを供給する供給手段と、供給された演奏データから、ダブルチョーキングが行われている部分を抽出する抽出手段と、ダブルチョーキングが行われている部分に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段であって、この生成方法情報は、ダブルチョーキングが行われている部分の演奏デー

50

タを上音と下音とで2つのパートに分割し、それぞれに異なる音量変化を与えるような楽音制御情報を生成する生成方法を表わすものである記憶手段と、抽出されたダブルチョーキングが行われている部分及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例(b)が対応〕：

【0111】

別の特徴〔28〕では、供給された演奏データから、ダブルチョーキングが行われている部分を抽出し、ダブルチョーキング部分に対応する生成方法情報に基づいて、抽出されたダブルチョーキング部分の演奏データを上音と下音とで2つのパートに分割し、それぞれに異なる音量変化を与えるような楽音制御情報を生成し、演奏データに付加するようにしているため、ダブルチョーキングの際に、ダブルチョーキングの上音と下音とを別々のパートに分離し、わざと、別パートで音量の時間変化タイミングをずらす表情付けを行うことができる。

10

【0112】

〔29〕演奏データを供給する供給手段と、供給された演奏データから、連続してチョーキングが行われている部分を抽出する抽出手段と、連続してチョーキングが行われている部分に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段であって、この生成方法情報は、連続してチョーキングが行われている部分において、毎回のチョーキングに対する演奏データの音程に不均一性を持たせるような楽音制御情報を生成する生成方法を表わすものである記憶手段と、抽出された連続してチョーキングが行われている部分及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例(c)が対応〕：

20

【0113】

別の特徴〔29〕では、供給された演奏データから、連続してチョーキングが行われている部分を抽出し、連続チョーキング部分に対応する生成方法情報に基づいて、抽出された連続チョーキング部分において、毎回のチョーキングに対する演奏データの音程に不均一性を持たせるような楽音制御情報を生成し、演奏データに付加するようにしているため、連続チョーキングの際の微妙な表情付けを行うことができる。つまり、チョーキングが連続して現れる部分において、毎回のチョーキングの音程変化が同じにならないように音程に不均一性を加える。例えば、チョーキングが連続している回数や音程の変化傾向に基づいて、変化付与テンプレートを選択し、付与する音程の変化傾向を決定し、チョーキングのかかった真ん中の音の音程から微妙に高くしたり、低くしたりする。

30

【0114】

〔30〕演奏データを供給する供給手段と、供給された演奏データから、アルペジオ演奏に相当する部分を抽出する抽出手段と、アルペジオ演奏に相当する部分に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段であって、この生成方法情報は、アルペジオ演奏における共通倍音を検出し、検出された倍音を別パートで発音させるようにするよう楽音制御情報を生成する生成方法を表わすものである記憶手段と、抽出されたアルペジオ演奏に相当する部分及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例(d)が対応〕：

40

【0115】

別の特徴〔30〕では、供給された演奏データから、アルペジオ演奏に相当する部分を抽出し、アルペジオ演奏相当部分に対応する生成方法情報に基づいて、抽出されたアルペジオ演奏における共通倍音を検出し、検出された倍音を別パートで発音させるようにするよう楽音制御情報を生成し、演奏データに付加するようにしているため、アルペジオの共通倍音を検出し、この倍音を別パートで小さい音量でならす表情付けを行うことができる。

【0116】

50

〔 3 1 〕演奏データを供給する供給手段と、供給された演奏データから、音色の変更を指示する所定の音楽記号に対応する部分を抽出する抽出手段と、音色の変更を指示する所定の音楽記号に対応する楽音制御情報を生成するための生成方法を表わす生成方法情報を記憶する記憶手段であって、この生成方法情報は、音色の変更を指示する所定の音楽記号に対応する部分で、当該音楽記号に対応する音色に、設定音色を変更するような楽音制御情報を生成する生成方法を表わすものである記憶手段と、抽出された音色の変更を指示する所定の音楽記号に対応する部分及び生成方法情報に基づき、楽音制御情報を生成する生成手段と、生成された楽音制御情報を、供給された演奏データに付加する付加手段とを具備する演奏データ作成装置〔実施例（e）が対応〕：

【 0 1 1 7 】

10

別の特徴〔 3 1 〕では、供給された演奏データから、音色の変更を指示する所定の音楽記号に対応する部分を抽出し、音色変更を指示用音楽記号に対応する生成方法情報に基づいて、抽出された音色変更指示用音楽記号に対応する部分で、当該音楽記号に対応する音色に、設定音色を変更するような楽音制御情報を生成し、演奏データに付加するようにしているので、楽譜シンボルにより音色を選択する表情付けを行うことができる。例えば、“ p i z z . ” が表示されていると、自動的に音色をピチカートストリングに変更し、“ a r c o ” の表示位置で元の擦弦音色に戻すようにする。

【 0 1 1 8 】

〔種々の特徴〕

また、この明細書に記載された種々の発明の実施に関する種々の特徴により、以下の（ 20
1 ）～（ 2 3 ）のように構成することもできる。

（ 1 ）演奏データを入力する手段と、入力される演奏データの特徴情報を取得する手段と、演奏データに関して所定の特徴情報と楽音制御情報との対応をルール化した表情付けモジュールを供給する手段と、取得された特徴情報に基づき、供給された表情付けモジュールのルールに従って楽音制御情報を設定する手段と、設定された楽音制御情報を、入力された演奏データに付加する手段と、楽音制御情報が付加された演奏データを出力する手段とを具備する演奏データ作成装置〔図 2 〕。つまり、（ 1 ）の構成によると、楽音制御の因子となる入力演奏データの特徴に対する楽音制御情報の設定手順をルール化した種々の表情付けモジュールを用意しておき、この表情付けモジュールに基づいて楽音制御情報を設定するようにしているので、より音楽的な演奏データを自動的に作成することができ 30
る。

【 0 1 1 9 】

（ 2 ）演奏データを入力する手段と、入力される演奏データの特徴情報を取得する手段と、取得された特徴情報に基づき、予め定められたルールに従って楽音制御量を設定する手段と、設定される楽音制御量の制御パラメータを調整する手段と、設定された楽音制御量と調整された制御パラメータに基づいて楽音制御情報を決定する手段と、決定された楽音制御情報を、入力された演奏データに付加する手段と、楽音制御情報が付加された演奏データを出力する手段とを具備する演奏データ作成装置、並びに、この演奏データ作成装置において、パラメータ調整手段が、出力された演奏データを評価し、評価結果に応じて再度制御パラメータを調整するもの〔実施例（ 1 ）～（ 2 4 ）,（ 2 7 ）～（ 3 0 ）〕。つ 40
まり、（ 2 ）の構成によると、種々の楽音制御情報の設定に際し、特徴情報に基づきルールに従って設定される楽音制御量（時間的、音程的、音量的等の演奏上の各種楽音制御量）の制御パラメータを調整可能とし、楽音制御量と調整された制御パラメータに基づいて楽音制御情報（時間パラメータ、音程パラメータ、音量パラメータ等の各種演奏パラメータ）を決定するようにしているので、最適の楽音制御情報による表情付けを行うことができる。

【 0 1 2 0 】

（ 3 ）入力される演奏データから音符時間情報を抽出する方法〔実施例（ 1 ）,（ 3 ）〕、入力される演奏データの演奏の進行状態を評価する方法〔実施例（ 2 ）〕、入力される演奏データから微小震動音情報を抽出する方法〔実施例（ 4 ）,（ 2 0 ）,（ 2 3 ）〕、 50

入力される演奏データからフレーズの区切りを認識する方法〔実施例(5)〕、入力される演奏データから所定区間毎の音高情報或いは平滑化した音高情報を算出する方法〔実施例(6)〕、入力される演奏データから音高変化方向転換情報を取得する方法〔実施例(7)〕、入力される演奏データから同一又は類似のパターン等を検出する方法〔実施例(8)、(9)〕、入力される演奏データから、予め登録されている音形を検出する方法〔実施例(10)〕、入力される演奏データから所定区間毎の音量情報或いは平滑化した音量情報を算出する方法〔実施例(11)〕、入力される演奏データの雰囲気情報を取得する方法〔実施例(12)〕、入力される演奏データから所定の音符群列情報を抽出する方法〔実施例(13)、(14)〕、入力される演奏データの和音音数情報を抽出する方法〔実施例(15)〕、入力される演奏データから音色情報を抽出する方法〔実施例(16)〕、入力される演奏データからフィンガリング情報を抽出する方法〔実施例(17)~(19)〕、入力される演奏データから特定の楽器奏法に対応する奏法情報を抽出する方法〔実施例(21)、(22)〕、入力される演奏データからストリングス等の微小変動音情報を抽出する方法〔実施例(24)〕、入力される演奏データの歌詞情報を取得する方法〔実施例(27)〕、入力演奏データから音源の出力波形情報を取得する方法〔実施例(28)〕、入力された演奏データから所定の演奏記号情報を抽出する方法〔実施例(29)、(30)〕などによって取得された特徴情報に基づき、それぞれの特徴と時間的楽音制御内容や音程的楽音制御内容等との対応ルールに従って時間パラメータや音程パラメータ等の楽音制御情報を設定するもの。このように、入力演奏データの特徴として、音符時間情報(音符密度、2音間隔)、演奏の進行状態、微小震動音情報(ロングトーンのトリル/ビブラート、ピッチベンド等)、フレーズの区切り(フレーズ終端部等)、音高情報、音高変化方向転換情報(上下降部等)、同一又は類似のパターン(同パターン連続、類似フレーズ等)、登録音形(フレーズテンプレート等)、音量情報、雰囲気情報(「緊張感」等)、音符群列情報(まとまった音符、長い連符等)、和音音数情報、音色情報、フィンガリング情報(指、ポジション移動、ポジション等)、奏法情報(ギターのプリングオフ、ハンマリングオン、ピアノのサスティンペダル等)、微小変動音情報(複数パートのトリル等)、歌詞情報、所定の演奏記号(強弱記号、スタッカート等)などを取得し、これらの特徴に応じて楽音制御情報を設定することによって、これらの特徴情報に基づく多彩な表情のある演奏出力を得ることができる。

【0121】

(4)演奏データを入力する手段と、既に入力された演奏データの所定の特徴情報と楽音制御情報の関係を記憶する手段と、新たに入力される演奏データの特徴情報を取得する手段と、取得された特徴情報に基づき、記憶された関係に従って楽音制御情報を設定する手段と、設定された楽音制御情報を、新たに入力された演奏データに付加する手段と、楽音制御情報が付加された演奏データを出力する手段とを具備する演奏データ作成装置〔実施例(25)〕、並びに、演奏データに関して所定の特徴情報と楽音制御情報との関係を複数組記録したライブラリと、演奏データを入力する手段と、入力される演奏データの特徴情報を取得する手段と、取得された特徴情報に基づき、ライブラリを参照することによって楽音制御情報を設定する手段と、設定された楽音制御情報を、入力された演奏データに付加する手段と、楽音制御情報が付加された演奏データを出力する手段とを具備する演奏データ作成装置〔実施例(26)〕。このように、既に入力された演奏データの所定の特徴情報と楽音制御情報の関係を記憶しておき、新たに入力される演奏データの特徴情報に基づき、記憶された関係に従う学習結果を利用して楽音制御情報を設定するようしたり、入力演奏データから取得した特徴情報に基づき、演奏データに関して所定の特徴情報と楽音制御情報との関係を複数組記録したライブラリを参照することによって楽音制御情報を設定するよう構成することによって、学習機能やライブラリ化により、表情付与の適応性を向上することができる。

【0122】

(5)演奏データを入力する手段と、入力される演奏データの所定の特徴情報に基づき楽音制御情報を設定する手段と、設定された楽音制御情報及び入力された演奏データの楽音

10

20

30

40

50

制御情報を演奏データ全体で対比する手段と、この対比結果に基づいて、設定された楽音制御情報を修正する手段とを具備する演奏データ作成装置〔実施例(31)〕。つまり、(5)の構成によると、設定された楽音制御情報及び入力された演奏データの楽音制御情報を演奏データ全体で対比した結果に基づいて、設定された楽音制御情報を修正するようにしているので、演奏データを見通して楽音制御情報を最適値に設定することができる。

【0123】

(6)演奏データを供給する供給手段と、供給された演奏データを解析し、複数種類の表情のうちいずれかの表情を付与可能な、演奏データをすべて含む全体区間内の部分区間を抽出する解析手段と、抽出された部分区間に含まれる演奏データに付与すべき表情を複数種類の表情から選択して決定する決定手段と、抽出された部分区間に含まれる演奏データのパラメータを、決定された表情に対応する表情付けアルゴリズムに従って自動的に編集するパラメータ編集手段とを有するパラメータ自動編集装置、並びに、演奏データを供給手段から供給する供給モジュールと、供給された演奏データを解析し、複数種類の表情のうちいずれかの表情を付与可能な、演奏データをすべて含む全体区間内の部分区間を抽出する解析モジュールと、抽出された部分区間に含まれる演奏データに付与すべき表情を複数種類の表情から選択して決定する決定モジュールと、抽出された部分区間に含まれる演奏データのパラメータを、決定された表情に対応する表情付けアルゴリズムに従って自動的に編集するパラメータ編集モジュールとを含む、コンピュータが実現できまプログラムを格納したパラメータ記憶媒体〔図52及び図53〕。ここで、演奏データは、シーケンスデータを想定している。このため、演奏データは時系列に並べることができ、「区間」という概念を、そこに入れることができる。また、「パラメータ」とは、テンポやタイミング等の時間的楽音制御情報や、音程的楽音制御情報、音量的制御情報など、演奏上楽音を制御するための変量情報であり、「演奏パラメータ」ということもあり、以下においても同様である。

10

20

【0124】

(7)演奏データを供給する供給手段と、供給された演奏データから、その音高が上昇傾向にあるデータ領域を抽出する抽出手段と、抽出されたデータ領域に含まれる演奏データの音量又は音程が、当該データ領域の最初に位置する演奏データから最後に位置する演奏データまで、それぞれ、徐々に増大するか又は低くずれるように、当該各演奏データの音量又は音程パラメータ値を編集するパラメータ編集手段とを有するパラメータ自動編集装置〔実施例(A)〕。ここで、上昇傾向とは、単純な上昇音系ではないが、全体を評価した場合には上昇音系と言えるものを含む趣旨であり、以下においても同様である。

30

【0125】

(8)演奏データを供給する供給手段と、供給された演奏データから、その音高が上昇傾向にあるデータ領域および下降傾向にあるデータ領域を抽出し、抽出されたデータ領域から、音高が上昇傾向から下降傾向に変化する変化点の演奏データを含むデータ領域をさらに抽出する抽出手段と、抽出されたデータ領域に含まれる演奏データ中、変化点の演奏データの音量にアクセントが付与されるように、当該音量パラメータ値を編集するパラメータ編集手段とを有するパラメータ自動編集装置〔実施例(A)〕。ここで、下降傾向とは、上昇傾向とは逆に、単純な下降音系ではないが、全体を評価した場合には下降音系と言えるものを含む趣旨であり、以下においても同様である。

40

【0126】

(9)演奏データを供給する供給手段と、供給された演奏データの最初に位置する演奏データから最後に位置する演奏データまでの音量乃至音程パラメータ値の変化傾向を規定する音量乃至音程変化パターンを複数種類記憶する記憶手段と、記憶された複数種類の音量乃至音程変化パターンからいずれかを選択する選択手段と、供給された演奏データの音量乃至音程パラメータ値が選択された音量乃至音程変化パターンによって規定される変化傾向になるように、当該各演奏データの音量乃至音程パラメータ値を編集するパラメータ編集手段とを有するパラメータ自動編集装置〔実施例(B)〕。ここで、変化傾向としては、たとえば、音量乃至音程パラメータ値を曲の進行に従って徐々に増大させるという傾向

50

を挙げることができる。このような変化傾向を用いることにより、曲の進行に従って興奮度を高めて行くことができる。なお、増大特性は、曲の進行に従って、所定の有限値に収束するような特性であることが望ましい。その方が、音源の出力範囲を超えず、また、より自然な表情付けになるからである。これは以下においても同様である。

【0127】

(10) 演奏データを供給する供給手段と、供給された演奏データから、発音を指示する演奏データの出現密度が所定値以上のデータ領域を抽出する抽出手段と、抽出されたデータ領域に含まれる演奏データの音量乃至音程不安定度パラメータ値を、出現密度に応じた値に編集するパラメータ編集手段とを有するパラメータ自動編集装置〔実施例(C)〕。ここで、出現密度とは、単位時間当たりの出現頻度を意味する。そして、出現密度が所定値以上とは、演奏が難関であることを意味するため、出現密度に応じた値とは、通常、元の音量パラメータ値より小さい値、すなわち音量が小さくなる方向への変更を意味し、音程不安定度パラメータ値については大きくなる方向への変更を意味する。これは、以下においても同様である。

10

【0128】

(11) 抽出されたデータ領域に含まれる演奏データに基づいて音程を算出し、算出された音程の最小値と最大値との音程変化幅を算出する算出手段を有し、パラメータ編集手段は、抽出されたデータ領域に含まれる演奏データの音量乃至音程不安定度パラメータ値を、出現密度および算出された音程変化幅に応じた値に編集するパラメータ自動編集装置〔実施例(C)〕。ここで、音程変化幅も、音量パラメータ値乃至音程不安定度パラメータ値を変更する指標になり得、通常、音程乃至音程不安定度変化幅が大きいほど、音量パラメータ値は減少方向に変更し、音程不安定度パラメータ値は増大方向に変更する。これは、以下においても同様である。

20

【0129】

(12) 演奏データを供給する供給手段と、供給された演奏データから、類似フレーズ領域を抽出する抽出手段と、抽出された類似フレーズ領域に含まれる類似フレーズ間の類似度を算出する算出手段と、類似フレーズ領域が連続して出現するときには、2番目以降に出現する類似フレーズの音量パラメータ値を、最初に出現する類似フレーズの音量パラメータ値より小さい値であって、算出された類似度に応じた値に編集する一方、類似フレーズ領域が離散して出現するときには、2番目以降に出現する類似フレーズの音量パラメータ値を、最初に出現する類似フレーズの音量パラメータ値に類似した値であって、算出された類似度に応じた値に編集するパラメータ編集手段とを有するパラメータ自動編集装置〔実施例(D)〕。ここで、類似度は、たとえば、比較するフレーズ間で、(1)演奏データがすべて同一、(2)演奏データの一部のみが相違、(3)演奏データの一部のみが同一、(4)演奏データがすべて相違、の4段階の値とするのが分かりやすいが、この段階は、これより細かくても粗くてもよい。これは以下においても同様である。

30

【0130】

(13) 演奏データを供給する供給手段と、供給された演奏データから、3拍子かつ発音を指示する演奏データの発音長がすべて同一音長である小節長のデータ領域を抽出する抽出手段と、抽出されたデータ領域中、強弱を付与すべき拍位置を決定する決定手段と、抽出されたデータ領域に含まれる演奏データ中、決定手段により強が決定された拍位置の演奏データの音量パラメータ値を増大させ、決定手段により弱が決定された拍位置の演奏データの音量パラメータ値を減少させるように編集するパラメータ編集手段とを有するパラメータ自動編集装置〔実施例(E)〕。ここで、強弱を付与すべき拍位置を決定する基準は、たとえば、演奏データとして選択された曲の作風、時代および作曲家等が挙げられる。これは以下においても同様である。

40

【0131】

(14) 演奏データを供給する供給手段と、供給された演奏データから、フレーズ長のデータ領域を抽出する抽出手段と、抽出されたデータ領域に設定されているテンポ値を算出する算出手段と、抽出されたデータ領域に含まれる最後に位置する演奏データであって、

50

発音を指示する演奏データ、およびその発音長を検出する検出手段と、検出された演奏データの音量が、算出されたテンポ値および検出された発音長に応じた持続時間だけ徐々に減衰するように、当該演奏データの音量パラメータ値を編集するパラメータ編集手段とを有するパラメータ自動編集装置。〔実施例(F)〕

【0132】

(15) 演奏データを供給する供給手段と、供給された演奏データから、発音を指示する演奏データであって、トリルまたはビブラートが付与されかつその発音長が所定長以上のものを抽出する抽出手段と、発音長が所定長以上の演奏データをトリル演奏するときの音量の変化乃至音程変化の速さを規定する音量変化パターン乃至音程変化速さパターンと、発音長が所定長以上の演奏データをビブラート演奏するときの音量の変化乃至音程変化の速さを規定する音量変化パターン乃至音程変化の速さパターンとを、それぞれ発音長に応じた種類記憶する記憶手段と、抽出された演奏データに応じた音量変化パターン乃至音程変化速さパターンを、記憶手段から読み出す読み出し手段と、抽出された演奏データの音量の変化乃至音程変化の速さが、読み出された音量変化パターン乃至音程変化速さパターンに規定された音量の変化乃至音程変化の速さになるように、当該演奏データの音量パラメータ値乃至音程変化の速さパラメータ値を編集するパラメータ編集手段とを有するパラメータ自動編集装置。〔実施例(G)〕

10

【0133】

(16) 演奏データを供給する供給手段と、供給された演奏データから、トリル演奏またはロール演奏が行われるデータ領域を抽出する抽出手段と、抽出されたデータ領域に含まれる演奏データの音量パラメータ値を不均一値に編集するパラメータ編集手段とを有するパラメータ自動編集装置。〔実施例(H)〕

20

【0134】

(17) 演奏データを供給する供給手段と、供給された演奏データから、同時に発音を指示する複数個の演奏データによって構成されるデータ領域を抽出する抽出手段と、強調すべき演奏データの位置を示すパターンを、同時に発音を指示する演奏データの個数およびその音高に応じた種類記憶する記憶手段と、抽出されたデータ領域に含まれる演奏データの個数およびその音高に応じたパターンを、記憶手段から読み出す読み出し手段と、抽出された演奏データ中、読み出されたパターンにより示される位置の演奏データが強調されるように、当該演奏データの音量乃至音程パラメータ値を編集するパラメータ編集手段とを有するパラメータ自動編集装置〔実施例(J)〕。ここで、同時に発音を指示する複数個の演奏データとしては、典型的には、和音を構成する演奏データであるが、これに限らず、たとえば、オクターブユニゾンがある。これは以下においても同様である。

30

【0135】

(18) 演奏データを供給する供給手段と、供給された演奏データから、発音を指示する演奏データであって、スタッカート発音する演奏データの直後のものを抽出する抽出手段と、抽出された演奏データの音量が小さくなるように、当該演奏データの音量パラメータ値を編集するパラメータ編集手段とを有するパラメータ自動編集装置。〔実施例(K)〕

【0136】

(19) 演奏データを供給する供給手段と、供給された演奏データから、発音を指示する演奏データであって、その発音長が所定長以上のものを抽出する抽出手段と、不均一値を出力する出力手段であって、発音開始からの経過時間に応じて、不均一値の変化幅を変化させるものと、抽出された演奏データの音量乃至音程が不均一かつその変化幅が変化しながら当該発音長持続するように、当該演奏データの音量乃至音程パラメータ値を出力手段によって出力される不均一値に編集するパラメータ編集手段とを有するパラメータ自動編集装置〔実施例(L)〕。ここで、抽出される演奏データは、いわゆるロングトーンの演奏データであり、このロングトーンの発音中に、その音量を、不均一かつ徐々に振幅を変えながら変更しているため、ロングトーンにゆらぎを付与していることになる。これは、以下においても同様である。

40

【0137】

50

(20) 演奏データを供給する供給手段と、供給された演奏データから、同音色のパートおよびそのパート数を検出する検出手段と、検出されたパート数に応じて、各パートの演奏音量値を算出する算出手段と、算出された各演奏音量値を、当該各パートの音量パラメータ値として設定する設定手段とを有するパラメータ自動編集装置〔実施例(N)〕。ここでは、パートディビジョン指定がされている楽譜の演奏を再現するときに、従来のパラメータ編集装置では、音量パラメータに対して何の変更も加えないために、その指定がされないときに比べて、音量が増大するという問題を解消するために、「検出されたパート数に応じて、各パートの演奏音量値を算出」している。この事情は、以下においても同様である。

【0138】

(21) 演奏データを供給する供給手段と、供給された演奏データから、ピッチバンドによるピッチの変更が指示された演奏データを抽出する抽出手段と、抽出された演奏データでのピッチバンドの変化傾向を算出し、算出結果に応じて、音量の変化量を決定する決定手段と、抽出された演奏データの音量の変化が、決定された音量の変化になるように、当該演奏データの音量パラメータ値を編集するパラメータ編集手段とを有するパラメータ自動編集装置。〔実施例(Q)〕

【0139】

(22) 演奏データを供給する供給手段と、供給された演奏データから、メロディ部分を抽出する抽出手段と、抽出されたメロディ部分を他のパートと比較して、メロディパートが相対的に他のパートより浮き立つように、演奏パラメータの変化量を決定する決定手段と、抽出されたメロディパートの演奏パラメータ値を、決定された演奏パラメータの変化量に基づいて編集するパラメータ編集手段とを有するパラメータ自動編集装置〔実施例(R)〕。ここでは、演奏データが複数パートからなるものであることを前提としている。しかし、ピアノのように、1つの音色でメロディから伴奏まで担当するような演奏データに対しては、演奏データに含まれるトップノートを抽出して、メロディパートとすれば、この(22)の構成をそのまま適用することができる。

【0140】

(23) 演奏データを供給する供給手段と、供給された演奏データから、歌詞情報を抽出する抽出手段と、抽出された歌詞情報に含まれる単語から、音量乃至音程変化を付与すべき単語を検出する検出手段と、単語に付与すべき音量乃至音程変化のパターンを示す音量乃至音程変化パターンを単語毎に記憶する記憶手段と、検出された単語に応じた音量乃至音程変化パターンを、記憶手段から読み出す読み出し手段と、検出された単語の音量乃至音程の変化が、読み出された音量乃至音程変化パターンによって示される音量乃至音程の変化になるように、当該演奏データの音量乃至音程パラメータ値を編集するパラメータ編集手段とを有するパラメータ自動編集装置〔実施例(U)〕。以上の(6)~(23)の構成をもつパラメータ自動編集装置によれば、供給手段から供給された演奏データが解析され、複数種類の表情のうちいずれかの表情を付与可能な、前記演奏データをすべて含む全体区間内の部分区間が抽出され、該抽出された部分区間に含まれる演奏データに付与すべき表情が前記複数種類の表情から選択されて決定され、前記抽出された部分区間に含まれる演奏データのパラメータが、前記決定された表情に対応する表情付けアルゴリズムに従って自動的に編集されるので、ユーザが初心者であっても簡単な操作で曲に多様な表情付けを行うことが可能となる。

【発明を実施するための最良の形態】

【0141】

以下、図面を参照しつつ、この発明の好適な実施例を説明する。なお、以下の実施例は単なる一例であって、この発明の精神を逸脱しない範囲で種々の変更が可能である。

【0142】

〔ハードウェア構成〕

図1は、この発明の一実施例による演奏データ作成装置(即ち、パラメータ自動編集装置)のハードウェア構成のブロック図を示す。この例では、システムは、中央処理装置(

10

20

30

40

50

CPU) 1、読出専用メモリ (ROM) 2、ランダムアクセスメモリ (RAM) 3、第1及び第2の検出回路4、5、表示回路6、音源回路7、効果回路8、外部記憶装置9等を備え、これらの装置1~9は、バス10を介して互いに接続されており、演奏データ作成処理を行うための演奏データ作成システム、或いは、パラメータ自動編集処理を行うためのパラメータ自動編集システムを構成している。

【0143】

システム全体を制御する(即ち、装置全体の制御を司る)CPU1は、テンポクロックや割り込みクロックの発生等に利用される(即ち、タイマ割り込み処理における割り込み時間や各種時間を計時する)タイマ11を備え、所定のプログラムに従って種々の制御を行い、特に、後述するテンポ変化、音量変化や音程変化等のための種々の処理を中枢的に遂行する。ROM2には、この演奏データ作成(パラメータ自動編集)システムを制御するための所定の制御プログラムが記憶されており、これらの制御プログラムには、基本的な演奏情報処理と共に、この発明による各種テンポ/タイミング、音量変化や音程変化等の変換処理プログラムや各種テーブル、各種データを含ませる(即ち、CPU1が実行する制御プログラムや、各種テーブルデータ等を記憶する)ことができる。RAM3は、これらの処理に際して必要なデータやパラメータを記憶し、また、各種レジスタやフラグ、処理中の各種データを一時記憶する(即ち、演奏データ、各種入力情報および演算結果等を一時的に記憶する)ためのワーク領域として用いられる。

10

【0144】

第1の検出回路4は鍵盤等の演奏操作子を備えた演奏操作装置12に接続され、第2の検出回路5に接続される操作スイッチ装置13は、各種のモード、パラメータ等の設定を行うために、数字/文字キーやつまみ等の操作子を備える。例えば、演奏操作装置12には主に音声情報や文字情報を入力するためのキーボードがあり、操作スイッチ装置13には、ポインティングデバイスであるマウスがあり、第1及び第2の検出回路4、5には、キーボードの各キーの操作状態を検出するキー操作検出回路や、マウスの操作状態を検出するマウス操作検出回路がこれらに対応して設けられる。

20

【0145】

表示回路6は、各種情報等を表示するディスプレイ14〔例えば、大型液晶ディスプレイ(LCD)若しくはCRT(Cathode Ray Tube)ディスプレイ〕や各種インジケータ〔発光ダイオード(LED)等〕を備えており、これらのディスプレイ14やインジケータは、スイッチ装置13の操作パネル上の各種操作子に並置することができる。また、ディスプレイ14には、各種設定画面や各種操作子ボタンを表示させ、各種モードやパラメータ値の設定・表示を行うようにすることもできる。

30

【0146】

DSP等で構成される効果回路8に接続されるサウンドシステム15は、音源回路7及び効果回路8と共に楽音出力部を構成し、この発明による種々の処理中に作成された各種演奏データに基づいて楽音を放音させ、表情付けされた出力演奏データによる演奏を試聴することができる。例えば、音源回路7は、キーボード(12)から入力された演奏データや予め記録された演奏データ等を楽音信号に変換し、効果回路8は音源回路7からの楽音信号に各種効果を付与し、サウンドシステム15は、DAC(Digital-to-Analog Converter)やアンプ、スピーカ等が含まれ、効果回路8からの楽音信号を音響に変換する。

40

【0147】

外部記憶装置9は、ハードディスクドライブ(HDD)、コンパクトディスク・リード・オンリィ・メモリ(CD-ROM)ドライブ、フレキシブルディスクドライブ(FDD)、光磁気(MO)ディスクドライブ、デジタル多目的ディスク(DVD)ドライブ等の記憶装置から成り、各種制御プログラムや各種データを記憶することができる。従って、演奏データの処理に必要な各種プログラムやデータは、ROM2を利用するだけでなく、外部記憶装置9からRAM3内に読み込むことができ、必要に応じて、処理結果をRAM3を介して外部記憶装置9に記録しておくこともできる。例えば、FDDは、記憶媒体であるフレキシブルディスク(FD)をドライブし、HDDは、制御プログラムを含む各

50

種アプリケーションプログラムや各種データ等を記憶するハードディスクをドライブし、CD-ROMドライブは、制御プログラムを含む各種アプリケーションプログラムや各種データ等を記憶するCD-ROMをドライブする。

【0148】

HDD(9)のハードディスクには、上述のように、CPU1が実行する制御プログラムも記憶でき、ROM2に制御プログラムが記憶されていない場合には、このハードディスクに制御プログラムを記憶させておき、それをRAM3に読み込むことにより、ROM2に制御プログラムを記憶している場合と同様の動作をCPU1にさせることができる。このようにすると、制御プログラムの追加やバージョンアップ等が容易に行える。

【0149】

CD-ROMドライブ(9)のCD-ROMから読み出された制御プログラムや各種データは、HDD(9)内のハードディスクにストアされる。これにより、制御プログラムの新規インストールやバージョンアップ等が容易に行える。なお、このCD-ROMドライブ以外にも、外部記憶装置9として、光磁気ディスク(MO)装置等、様々な形態のメディアを利用するための装置を設けるようにしてもよい。

【0150】

この例では、バス10にMIDIインターフェイス(I/F)16が接続され、システムは他のMIDI機器17と通信することができ、外部からのMIDI(Musical Instrument Digital Interface)信号を入力したり、MIDI信号を外部に出力したりする。さらに、バス10には通信インターフェイス18も接続され、通信ネットワーク19を介してサーバコンピュータ20とデータの送受信を行い、サーバコンピュータ20から制御プログラムや各種データを外部記憶装置9にストアすることもできる。また、通信ネットワーク19には、サーバコンピュータ20の外、他のクライアントコンピュータを接続することができる。

【0151】

MIDI I/F16は、専用のものに限らず、RS-232CやUSB(ユニバーサル・シリアル・バス)、IEEE1394(アイトリプルイー1394)等の汎用のインターフェースより構成してもよい。この場合、MIDIメッセージ以外のデータをも同時に送受信してもよい。

【0152】

通信I/F18は、上述のように、たとえばLAN(Local Area Network)やインターネット、電話回線等の通信ネットワーク19に接続されており、該通信ネットワーク19を介して、サーバコンピュータ20に接続される。HDD(9)内のハードディスクに上記各プログラムや各種パラメータが記憶されていない場合には、通信I/F18は、サーバコンピュータ20からプログラムやパラメータをダウンロードするために用いられる。クライアントとなるコンピュータ(本実施の形態では、演奏データ作成装置或いはパラメータ自動編集装置)は、通信I/F18および通信ネットワーク19を介してサーバコンピュータ20へとプログラムやパラメータのダウンロードを要求するコマンドを送信する。サーバコンピュータ20は、このコマンドを受け、要求されたプログラムやパラメータを、通信ネットワーク19を介してコンピュータへと配信し、コンピュータが通信I/F18を介して、これらプログラムやパラメータを受信してHDD(9)内のハードディスクに蓄積することにより、ダウンロードが完了する。

【0153】

この他、外部コンピュータ等との間で直接データのやりとりを行うためのインターフェースを備えてもよい。

【0154】

なお、本実施の形態の演奏データ作成装置或いはパラメータ自動編集装置は、上述の構成から分かるように、汎用的なパーソナルコンピュータ上に構築されたものであるが、これに限らず、この発明を実施できる最小限要素のみから構成した専用装置上に構築してもよい。

10

20

30

40

50

【 0 1 5 5 】

この発明による演奏データ処理（パラメータ自動編集）システムは、電子楽器の形態で実施することができるが、パーソナルコンピュータ（PC）に楽音処理のアプリケーションプログラムを組み込んだ形態で実施することもできる。また、音源回路7は、ハードウェアで構成する必要はなく、ソフトウェア音源で構成することもでき（本実施の形態では、音源回路7は、その名称の通り、すべてハードウェアで構成したが、これに限らず、一部ソフトウェアで構成し、残りの部分をハードウェアで構成してもよいし、また、すべてソフトウェアで構成するようにしてもよい。）、さらに、音源機能を含む楽音出力部の機能を他のMIDI機器17に委ねることもできる。

【 0 1 5 6 】

〔実施形態1〕

まず、具体的な演奏パラメータとして、主として、テンポ変化やゲートタイム、発音開始タイミング等のタイミングシフトなどの時間的楽音制御情報（具体例中には、音程パラメータや音量パラメータ等、他の演奏パラメータを含むものもある。）を用いて表情付けを行う「実施形態1」について説明する。

【 0 1 5 7 】

〔システム機能の概要〕

図2は、この発明の演奏データ処理（パラメータ自動編集）システムによる機能の概要を示す。システムの機能は、元演奏データODを取込む元演奏データ取込みブロックAB、ブロックABから供給される元演奏データODに主としてテンポ変換を用いて時間に関連する表情付けを行う表情付けブロックEB、表情付けブロックEBに対して供給される各種表情に対応した表情付けのルールや手法を記憶した表情付けモジュールEM、及び、表情付けされた表情付け後演奏データEDを音源部等に送り出す表情付け後演奏データ送出ブロックSBから成る。ここで、表情付けモジュールEMは、様々な音楽表現における主としてテンポ変化の特徴をルール化したものであり、このルールに従い、元演奏データODに対して、テンポ変化やゲートタイム、発音開始タイミング等のタイミングシフトを主とする時間的な表情付けを行い、元演奏データODを音楽的な表情付け後演奏データEDに変換するものである。この表情付けモジュールEMには、種々の表情付けモジュールEM1, EM2, ..., EMnが予め用意されており、その中から任意の1乃至複数の表情付けモジュールがユーザにより選択され、表情付けブロックEBへと供給される。

【 0 1 5 8 】

表情付けモジュールEMの機能は、図1の演奏データ処理（パラメータ自動編集）システムにおいて、ROM2の各種テンポ・タイミング変換処理プログラムを作動させることにより、或いは、外部記憶装置9から所望のテンポ・タイミング変換処理プログラムをロードすることにより、実現することができる。このように外部記憶装置9から表情付けモジュールを供給可能としたので、図1の演奏データ処理（パラメータ自動編集）システムに対して、多数の表情付けモジュールのうちユーザが所望する表情付けモジュールのみをインストールしたり、メーカー等から新たに供給された表情付けモジュールを新規インストールしたりすることができる。また、元演奏データODは、システム機能の元演奏データ取込みブロックABにより、鍵盤形操作子装置12、外部記憶装置9、他のシーケンサや演奏機器等のMIDI機器17等から任意に入力することができ、表情付け後演奏データEDは、表情付け後演奏データ送出ブロックSBにより、楽音出力部、外部記憶装置9、他のMIDI機器等に任意に出力することができる。なお、演奏データは、典型的にはMIDIデータである。

【 0 1 5 9 】

つまり、この発明による表情付けモジュールEMは、楽音の時間的制御の因子となる演奏データの特徴を予めルールとして用意している。表情付けブロックEBに元演奏データODが入力されると、元演奏データODから取り出された特徴情報に基づき、ルールに従い時間的制御情報（テンポやタイミング）が設定される。この楽音制御情報は元演奏データODに付加され、表情が付加された演奏データEDとして出力される。時間的楽音制御

10

20

30

40

50

量は、制御パラメータにより調整可

能で、楽音に対して最適の時間的表情付けを行うことができる。さらに、学習機能やライブラリ化により表情付与の範囲を拡大したり、楽音制御情報は、演奏データ全体で対比して修正し、演奏データを見通した設定が可能である。

【0160】

表情付けモジュールEMは、また、楽音の音量や音程等の他の演奏上の制御因子となる演奏データの特徴を予めルールとして用意しておくことができる。従って、表情付けブロックEBに元演奏データODが入力されると、元演奏データODから取り出された特徴情報に基づき、ルールに従って、音量や音程等に变化を与える演奏制御情報（楽音制御情報、演奏パラメータ）を設定する。そして、これらの演奏制御情報は元演奏データODに付加され、表情が付加された演奏データEDとして出力することができる。また、これらの演奏制御情報に基づく楽音制御量も、制御パラメータにより調整可能で、楽音に対して最適の表情付けを行うことができる。さらに、学習機能やライブラリ化により表情付与の範囲を拡大したり、これらの演奏制御情報も、演奏データ全体で対比して修正し、演奏データを見通した設定が可能である。

10

【0161】

〔表情付けモジュールの種々の例〕

以下、この発明の一実施例による種々の表情付けモジュールEM（EM1，EM2，…，EMn）により実行される種々のテンポ変化等のルールや手法を個々の実施例（1）～（31）について説明する。これらの実施例では、モジュールEMにより元のMIDI演奏データODを表情付け後MIDI演奏データEDに変換するものとして説明されるが、この発明は、MIDI等のデータ形式には関係なく適用することができる。なお、演奏データに対してテンポ変化を付与する方法としては、テンポチェンジイベントを挿入する方法と、音符の時刻をずらす方法とがある。以下の説明では、主として、テンポチェンジイベントを挿入する方法を用いているが、音符の時刻をずらす方法に置き換えてもよい。また、逆に、音符の時刻をずらす方法のものをテンポチェンジイベントを挿入する方法に置き換えてもよい。

20

【0162】

（1）「音数が増えるとテンポが速くなる」：

一般に、参照音数が多くなるに従い、演奏が難しくなり、テンポが遅くなると思われ勝ちである。しかしながら、実際の演奏では、このような心理状態からテンポを保とうとする意識が強く表われ、逆に、速くなってしまおうという側面がある。また、音数が多い方が、より活発な演奏局面である場合が多いという傾向がみられる。したがって、参照音数の増大に従い、必然的にテンポを速くして演奏することが、人間的でありまた自然でもある。そこで、このテンポ変化ルールでは、このような表情を再現するために、音符の密度に応じて、例えば、演奏データを小節単位で評価し、1拍当りの音符の数に応じて、テンポを変化させるという手法が採用される。

30

【0163】

例えば、図3に示した譜例において、1小節目では1拍当りの音符数は1つであるが、2小節目では1拍あたりの音符数が2つになっている。それで、この数値〔（1小節中の音符数）/（1小節の拍数）〕を音符密度とし、この音符密度に応じてテンポを変化させる。1小節目は、音符密度が「1」なので、そのままのテンポで4分音符“132”となる。一方、2小節目は、音符密度が「2」となるので、この音符密度からテーブルを通してテンポの値を設定する。このようなテーブルには、例えば、図4に示されるような「音符密度 - テンポ係数（倍率）」テーブルを用いることができる。従って、図3の譜例の2小節目については、音符密度「2」に対して、図4のテーブルから、例えばテンポ係数K = 1.02（2%増速）という結果を得ることができる。そして、このテンポ係数Kを元演奏データのテンポ値に乗じて所望のテンポ値を決定する。

40

【0164】

図5は、表情付けモジュールEMによる、音符の密度に応じてテンポを速める音符密度

50

応答処理の一例を示すフローチャートである。この処理フローの最初のステップA1では、元のMIDIデータODからテンポ決定を行うための対象パートを複数の演奏パートの中から選択し、次のステップA2で、テンポについて強弱の値を設定(ステップA1から最初にステップA2に進んだ当初は、値の初期値として例えば「1」を設定)してステップA3に進む。

【0165】

次のステップA3において、選択したパートを指定区間(例えば、1小節毎の区間)に分割し、次のステップA4で各指定区間のテンポ係数Kを求めることにより、各指定区間での音符密度を評価する。ステップA3～A4での音符数の評価は、指定区間を例えば1小節毎の区間として、小節単位で行う他、拍単位で行っても良いし、テンポの基準となる時間(Tick=最小時間分解能、クロック、ステップタイムともいう)で行っても良い。また、各指定区間のテンポ係数Kは、例えば、上述した図4の「音符密度-テンポ係数」テーブルから求めることができる。

10

【0166】

なお、ステップA4でテンポ係数Kを求める場合、区間音符数が「0」のときは、図4に で表わすように、例外的にテンポ係数Kを「1」とする。一般に、MIDIデータにおいては休符は音符として扱わないので、このままでは、休符ばかりで構成される小節はテンポが遅くなるという結果になってしまう。そのため、評価する区間内に音符がない(休符である)ときには、テンポ係数Kは「1」と評価する例外的処理を行う必要がある。

20

【0167】

ステップA4からステップA5に進むと、求められた各指定区間のテンポ係数K、強弱の値及び現在設定されているテンポの値(例えば、図3の譜例では“132”)に基づいてテンポチェンジ量(例えば、“132”×K)を算出し、このテンポチェンジを各指定区間のMIDIデータに付与する。次のステップA6では、テンポチェンジを付与したMIDIデータを一曲全部又は必要部分だけ再生し、例えば、楽音出力部を介して放音させてこのMIDIデータによる演奏を試聴する。この後、ステップA7に進み、MIDIデータ再生の結果、テンポチェンジを付与したMIDIデータが良好であると判断した場合は、この音符密度応答処理を終了する。一方、再生MIDIデータにテンポ設定上の不具合があると判断した場合は、ステップA2に戻って強弱値を設定し直した上、ステップA3～A7の処理を繰り返す。

30

【0168】

なお、良否判定ステップA7～強弱値設定ステップA2は、所定の判断基準により再生MIDIデータを自動判定し、良好な結果を得る方向に所定値だけ強弱値を自動的に増減する等の方法により、自動化することもできる。また、強弱値は、区間毎に異なる値を設定することができるようにしてもよい。さらに、強弱値を各区間に共通的に設定する方法と区間毎に個別に設定する方法を共用してもよい。例えば、強弱値=1に初期設定した状態で全曲を通してこのテンポ変化ルールを適用してみて、表情を付けたMIDIデータを再生したとき(ステップA6)、これがNGでステップA2に戻ったときに、表情が過剰な各区間については各過剰度に応じて強弱値を「1」未満に再設定し、逆に、不足の各区間には各不足度に応じて「1」を超える値に設定し直すという具合に編集を行うことができる。これらの態様は、実施例(2)以下のテンポ変化/タイミングシフト等に関する諸ルールを適用して演奏パラメータ値付与を実施する場合にも、同様に実施可能である。

40

【0169】

この場合、元の演奏データ(MIDIデータ)ODをディスプレイ14上に表示させ、表示された演奏データOD上に、付与したパラメータ値(テンポチェンジ量等)やその位置を重ねて表示させることにより、ユーザにパラメータ値設定の結果を確認させることができる。また、この確認時には、パラメータ変化付与処理を施したくない位置をユーザが指定できるようにしてもよい。

50

【 0 1 7 0 】

複数の演奏パート（リズム／伴奏、メロディ等）がある場合、一般にはテンポは共通とするので、上述した図5の音符密度応答処理では、テンポを決めるための特定パートをステップA1で選択的に設定しているが、次に説明するように、複数のパートの情報を総合した評価でテンポを決定する方法も採用することができる。なお、テンポイベントを変更するのではなく、音符の時刻をずらすことによって、テンポの変更を再現することにより、複数のパート間で独立なテンポ設定を行うことも可能である。

【 0 1 7 1 】

図6は、表情付けモジュールEMによる、音符の密度に応じてテンポを速める音符密度応答処理の他の例を示すフローチャートであり、この処理は、パートを選択指定せずに全パートの情報を総合した評価でテンポを決定する場合に適用される。この処理フローでは、最初のステップB1でテンポについて強弱の値を設定してステップB2に進む。ステップB2では、各パートの指定区間（例えば、1小節毎の区間）に分割し、次のステップB3で全パートの各指定区間の平均の音符密度を算出し、ステップB4において、算出した音符密度から各指定区間のテンポ係数Kを求め、各指定区間での音符密度を評価する。なお、ステップB2～B4での音符数評価は、既述したのと同様に、小節単位で行う他、拍単位で行っても良いし、テンポの基準となる時間（Tick）で行っても良いこと、音符無しの指定区間でテンポ係数Kを「1」とすることは、もちろんである。

【 0 1 7 2 】

ステップB4からステップB5に進んだ後は、前述したステップA5、A6と同様に、ステップB5において、求められたテンポ係数K、強弱の値及び現在テンポ値に応じたテンポチェンジを各指定区間のMIDIデータに付与し、次のステップB6でこのMIDIデータを再生する。この後、ステップB7にて再生MIDIデータが良好であるとされれば、この音符密度応答処理を終了し、不具合があれば、ステップB1に戻りステップB1～B7の処理を繰り返す。

【 0 1 7 3 】

(2) 「曲を通して、徐々に、じわじわテンポが速くなる」：

ライブ演奏などでは、興奮度が高まることによってどんどんテンポが速くなっていくということは、よく起る現象である。このテンポ変化ルールでは、1曲の演奏が進行して行くに従って興奮度が高まることに着目し、この現象を逆に曲制作に利用する。つまり、曲の進行につれてだんだんとテンポを速くすることによって、興奮度の高い所謂「走る」感覚の表情を出すようにする。この場合、例えば1小節毎に、予め定められた倍率をもって、テンポを速くしていくようにするのが良い。

【 0 1 7 4 】

図7は、この発明の一実施例の表情付けモジュールEMによる、曲進行に従いテンポを速める処理のフローチャートである。この処理フローでは、最初のステップC1でテンポについて強弱の値を設定し、次のステップC2で、元のMIDIデータODを指定区間（例えば、1小節毎の区間）に分割し、ステップC3に進み曲の進行を評価する。ステップC3においては、例えば、図8に示すようなテーブルを用いて、各指定区間のテンポ係数（倍率）Kを求める。このテーブルは、1曲の開始から終了までの間を例えば、小節毎の指定区間に区切り、曲の進行に従い徐々に増大するテンポ係数Kの値を小節毎に予め定めたものである。従って、このようなテーブルを利用することにより、曲の進行に応じて、対応する小節のテンポ係数Kの値を順次取り出すことができる。

【 0 1 7 5 】

次のステップC4では、求められたテンポ係数K、強弱の値及び現在テンポ値に応じたテンポチェンジを各指定区間のMIDIデータに付与し、次のステップC5でこのMIDIデータを再生する。この後、ステップC6にて再生MIDIデータが良好であるとされれば、この進行状態応答処理を終了し、不具合があれば、ステップC1に戻りステップC1～C6の処理を繰り返す。

【 0 1 7 6 】

10

20

30

40

50

なお、ステップC3で利用するテーブルの要件としては、テンポをむやみに速くすることはかえって自然な表情を損なうことになるので、図8に示すように、テンポがある程度の速さに近づくと、テンポ係数Kを或る値（例えば、 $K = 1.2$ ）以上には高くすることができなくなるように、飽和して行き上に凸な曲線になっている特性であることが好ましい。もちろん、テンポ変化は小節単に行う必要はなく、拍単位でもよいし、テンポを決めるための単位時間（Tick）でも良い。

【0177】

また、上述の処理フローにおいて、1曲中の各指定区間毎に決められたテンポ係数Kを用いる際、ステップC1で設定する強弱値として「興奮度」と名付けた可調整パラメータを用いて、小節毎に速くなるテンポ係数Kを調整可能にすると、テンポ決定手順の操作性を向上することができるようになる。この興奮度パラメータは、計算上は、図8のようなテーブルの縦軸（テンポ係数K）を何倍にするかを表わす数値とするが、設定上は、ディスプレイ14の興奮度設定画面等に、興奮度が「強い」、「中程度」、「弱い」、「変化なし」というような言葉を選択可能に表示しておき、実際の処理において、選択された言葉に対応する数値を用いてテンポチェンジを行うようにする。これにより、ユーザには、言葉で設定を行うというインターフェースを提供するので、ユーザーフレンドリーになる。

10

【0178】

(3) 「音符が細かいときはテンポを遅くする」:

このテンポ変化ルールは、(1)のテンポ変化ルールと矛盾するようであるが、音符が細かいときは人は弾けないので、このような表情を与えたい場合には有効である。このためには、単純には、(1)のテンポ変化ルールで用いる図4の「音符密度 - テンポ係数」テーブルとは逆に、テンポ係数Kが音符密度の増大に応じて減少するテーブルを用い、図5や図6の処理フローと同様の手法で、処理することができる。例えば、図2の2小節めのように、音数が多いときは、音符数/拍 = 2をこのテーブルに通すと、例えば、「0.98」（2%遅くする）というような結果になるようにすればよい。

20

【0179】

難しく弾けないことをシミュレートして、実在感を増すようにしたいという場合等には、図9に示すように、低いテンポの間は、連続する2音間の「音符時間間隔」に対して、テンポTを比例させるが、ある一定テンポ T_s 以上からは、テンポTを遅めに变化させるようなテーブルを用いるのが良い。このテンポ閾値 T_s を可調整パラメータとすることによって、下手さのシミュレートを調整することができる。

30

【0180】

(4) 「ロングトーンのトリル/ビブラートは、最初はゆっくりとする」:

異なる音高を微小周期で交互に繰り返し演奏するトリルや、音を微妙に震わせるビブラートは、例えば、1秒以上（テンポ値が“120”での2分音符など）のロングトーンに対しては、最初はゆっくりさせるのが好ましい。そこで、このテンポ変化ルールは、ロングトーンのトリル/ビブラートについて最初はゆっくりとしたテンポを設定するものである。トリルやビブラートは、一般には、ノートイベント（note event）やピッチベンドイベント（pitch bend event）で表現されるが、時間的に均等間隔で生成されるノートイベントやピッチベンドイベントに対して、テンポイベント（tempo event）を变化させることによって、トリルやビブラートの速さを調整することができる。この場合、曲自体のテンポはそのまま、トリルやビブラートのテンポを变化させる。

40

【0181】

図10は、この発明の一実施例の表情付けモジュールEMによるロングトーンのトリル/ビブラートに対する処理の例を示すフローチャートである。この処理フローでは、最初のステップD1でテンポについて強弱の値を設定した後、ステップD2において、元のMIDIデータOD中からトリル且つロングトーンの部分或いはビブラート且つロングトーンの部分を検出する。ステップD2でロングトーンに対してトリル又はビブラートがか

50

かっている部分を検出すると、ステップD3に進んで、例えば、予め用意された図11のようなテンポ変化曲線を用いてトリル又はビブラートの時間経過に応じたテンポ係数Kを求める。

【0182】

次のステップD4では、求められたテンポ係数K、強弱の値及び現在テンポ値に応じてトリル音符列(トリル音符列をピッチベンド列で表現することもできる)又はビブラートのピッチベンド列を生成し、続くステップD5において、生成したトリル音符列又はビブラートピッチベンド列を元のMIDIデータODに付与し、次のステップD6でこのMIDIデータを再生する。この後、ステップD7にて再生MIDIデータが良好であるとされれば、この長音トリル/ビブラート処理を終了し、不具合があれば、ステップD1に戻りステップD1~D7の処理を繰り返す。

10

【0183】

ステップD3で用意されるテンポ変化曲線は、図11に示されるようなものである。この曲線では、時間 t_0 にテンポ変化処理が開始され、テンポ係数Kは、当初値($K_0 = 1$)から減少し時間 t_a でテンポ変化が最も落ち込んで値 K_d となり、その後は徐々に増大して時間 t_b でテンポ変化の目標に到達し目標値 K_t となる。ここで、当初値 K_0 から目標値 K_t への変化量 K_t は、テンポ変化の全体時間 $t_0 \sim t_b$ におけるテンポ変化の幅Kの10~20%とするのがよく、増大時間 $t_a \sim t_b$ は減少時間 $t_0 \sim t_a$ の倍以上あることが好ましい。

【0184】

20

図11のようなテンポ変化曲線に対しては、単にテンポ係数Kの倍率(強弱値)だけでなく、時間(横軸)や他の変化量(縦軸)を変更することができる可調整パラメータを用意し、適切なテンポ変化を選択することができるようにするのが良い。この場合、例えば、全体時間 $t_0 \sim t_b$ 、減少時間 $t_0 \sim t_a$ 、テンポ変化幅K、目標変化量 K_t 等のパラメータを、「強い」、「弱い」、「速い」、「遅い」等の言葉で設定可能とすると好都合である。

【0185】

(5)「フレーズの経過部ではゆっくりめにする」:

このテンポ変化ルールでは、元のMIDIデータODについて、フレーズの区間を解釈し、或いは、ユーザがフレーズの区切りを指定し、そのフレーズ区間の終わりの部分では、当該フレーズが終了し間もなく次フレーズが開始することを表わすために、テンポを遅くするという表情付けを設定によって自動的に行う。図12は、この場合のテンポの変化例を示し、各フレーズ1, 2, ...は、例えば、8小節程度の区間から成る。具体的には、例えば、遅くする部分は、最後の2拍程度とし、遅くするテンポ変化量は95%程度とすることができる。しかしながら、このようなテンポチェンジの効果は、元のテンポにも依るし拍子設定にも依存するので、相当の自由度が必要である。例えば、拍子設定については、4/4拍子であれば2拍とするが、3/4拍子なら1拍とするのである。

30

【0186】

図13は、この発明の一実施例の表情付けモジュールEMによる、フレーズ経過部でテンポをゆっくりめにする処理の例を示すフローチャートである。この処理フローでは、最初のステップE1でテンポについて強弱の値を設定した後、ステップE2において、元のMIDIデータOD中からフレーズの切れ目位置の部分を検出し、さらに、ステップE3に進む。ステップE3では、切れ目部分に対応するフレーズの終端部に、強弱値と現在のテンポ値に応じて図12のように徐々に遅くなるテンポチェンジを付与し、次のステップE4で、切れ目部分に対応する次フレーズの始端には、元のテンポに戻るテンポチェンジを付与する。その後、次のステップE5でこのMIDIデータを再生した後、ステップE6にて再生MIDIデータが良好であるとされれば、このフレーズ終端部処理を終了し、不具合があれば、ステップE1に戻りステップE1~E6の処理を繰り返す。

40

【0187】

(6)「音域によってテンポを変える」:

50

例えば、高い音を出す楽器ほど、速い運指に対応できるし、音の立ち上がりも速いので、高音ほどテンポを速くし、低音ほどテンポを遅くするということには妥当性がある。従って、このテンポ変化手法では、音域によってテンポを変え、高音ほどテンポを速くするルールを表情付けに採用する。このために、音高情報によってテンポ係数 K を決定し、このテンポ係数 K で「音高 - テンポ係数」テーブルをひくことによって、テンポの変化量を決定する。このテーブルについて、音高とテンポ変化の相関関係を可調整パラメータとすればよい。1つ1つの音でこのような処理を行っても良いが、ある指定した区間毎にその音高の平均から処理を決めるということも有効である。ここで、指定区間は、フレーズを単位としても良いし、小節や拍を単位としても良い。

【0188】

10

図14は、この発明の一実施例の表情付けモジュールEMによる、音域によってテンポを変え高音ほどテンポを速くする処理の一例を示すフローチャートである。この処理フローでは、最初のステップF1でテンポについて強弱の値を設定した後、ステップF2において、元のMIDIデータODを指定区間(例えば、1小節毎の区間)に分割し、さらに、ステップF3では、各指定区間の音符の平均音高を算出した上、ステップF4に進む。

【0189】

次のステップF4では、図15に示されるような「音高 - テンポ係数」テーブルを用いて、算出された平均音高からテンポ係数 K を求める。この場合、当該区間の音符数が「0」のときはテンポ係数 K の値は「1」とする。更に次のステップF5にて、求められたテンポ係数 K 、強弱の値及び現在テンポ値に応じたテンポチェンジを各指定区間のMIDIデータに付与し、次のステップF6でこのMIDIデータを再生する。この後、ステップF7にて再生MIDIデータが良好であるとされれば、この音域対応処理を終了し、不具合があれば、ステップF1に戻りステップF1～F7の処理を繰り返す。

20

【0190】

上述の処理フローにおいては、ステップF4で図15のような「音高 - テンポ係数」テーブルを用い、ステップF1で強弱値なる可調整パラメータを設定してテンポ係数 K を調整可能にし、指定区間毎にその音高の平均から処理を決めている。この場合、設定上のパラメータとして「強い」、「中程度」、「弱い」等の所定の文言を用いることによりテンポ決定手順の操作性を向上することができる。また、ステップF1～F4での処理を行うのに、指定区間毎にパラメータ値を個別的に指示することがもできるようにしておくと、効率的な編集を行うことができる。

30

【0191】

例えば、起動当初は、パラメータを「中程度」に設定して $= 1$ に設定しておき、全曲を通して、このルールを適応させて、ステップF5で新しい曲データを生成する。そして、ステップF6でこの曲データを再生してみて、ステップF7でこのルールによる表情に不具合があると、ステップF1に戻ったときに、表情が極端な部分についてはパラメータを「弱い」に設定し直し、逆に、足りない部分については「強い」に設定し直すという具合に編集を行うことができる。このように区間を個別的に指示する場合は、1音毎に「音高 - テンポ係数」テーブル(例えば、図15)を参照してテンポを変化させることが好ましい。

40

【0192】

また、1音毎にテンポを変化させたのでは、不自然な変換になってしまう、という場合には、音高変化フィルタを用い音高の変化を滑らかにしてから「音高 - テンポ係数」テーブルを参照し、テンポを変化させるのが良い手法である。図16は、この発明の一実施例の表情付けモジュールEMによる、音域によってテンポを変え高音ほどテンポを速くする処理の他の例を示すフローチャートである。この処理フローでは、最初のステップG1でテンポについて強弱の値を設定した後、ステップF2にて音高変化をフィルタリングし音高変化カーブを求める。

【0193】

50

次のステップG3では、この音高変化カーブ上の各点の音高値を基にして「音高 - テンポ係数」テーブルを参照し、テンポ係数Kを求める。更に次のステップG4にて、求められたテンポ係数K、強弱の値及び現在テンポ値に応じたテンポチェンジをMIDIデータに付与し、次のステップG5でこのMIDIデータを再生する。この後、ステップG6にて再生MIDIデータが良好であるとされれば、この音域対応処理を終了し、不具合があれば、ステップG1に戻りステップG1～G6の処理を繰り返す。

【0194】

(7)「音高が上昇から下降に変わるときに間をおく」:

(6)のテンポ変化ルールによって、上昇音系はだんだん速く、下降音系はだんだん遅く、というテンポ表現が実現することができるが、さらに、上昇音系から、下降音系に転じるときに、一旦テンポ的に間を開けると、自然なつながりが得られる場合がある。そこで、このテンポ変化手法では、音高が上昇から下降に変わるときに間をおくというルールを採用する。例えば、この間の開け方の度合いをパラメータとして全曲を通してこのルールを適用し、新しい曲データを生成する。そして、再生してみて、このルールによる表情が極端な部分に関しては「弱い」に設定し直し、逆に足りない部分については「強い」に設定し直すという具合に編集を行う。

10

【0195】

図17は、この発明の一実施例の表情付けモジュールEMによる、音高の上下降変化部で間をおく処理の例を示すフローチャートである。この処理フローでは、最初のステップH1でテンポについて強弱の値を設定した後、ステップH2にて音高変化をフィルタリングし音高変化カーブを求める。ここで、フィルタリングは、例えば、図18の下段に示されるように、総体的に音高上昇傾向にある音群中に、音n1から音n2に向かう微小な音高降下があった場合、この降下現象をノイズとして除去する機能を有する。次のステップH3では、求められた音高変化カーブの極性符号〔上昇時は正(+)の値であり、下降時は負(-)の値である。〕の変化を検知することにより、音高が上昇から下降に変化する時間位置を検出する。

20

【0196】

次のステップH5では、例えば、図18の上段に示されるように、MIDIデータの上昇部分の終端近傍の音に強弱値とテンポの値に応じた徐々に遅くなるテンポチェンジを付与してステップH6に進む。ステップH6でこのMIDIデータを再生する。この後、ステップH7にて再生MIDIデータが良好であるとされれば、この音高上下降変化部処理を終了し、不具合があれば、ステップH1に戻りステップH1～H7の処理を繰り返す。

30

【0197】

(8)「同じパターンが連続したとき、2回目は遅く/速く」:

一般に、同じパターンを同じ表情で演奏することは、音楽的でないとして嫌われる。このような演奏を避けるには種々の方法があるが、1回目と2回目でテンポを変える方法もある。このテンポ変化手法では、同一/類似パターンが連続したとき、2回目は遅く或いは速く演奏するルールを採用する。この手法を実行するには、当初は、例えば「2回目は速く」(2回目の強弱値 = $1 > 1$)と設定して全曲を通してこのルールを適用させて、新しい曲データを生成する。そして、この曲データを再生してみて、このルールによる表情が相応しくない場合には、「表情をつけない」つまり「1回目と同じ」($= 1$)や「2回目は遅く」(< 1)、或いは、「2回目は前回より小さめに速く」($1 < < 1$)や「2回目は前回より大きめに速く」(> 1)に設定し直すというような手法で編集を行う。

40

【0198】

なお、2回目は複音で演奏するような場合には、複音の方が重量感が出るため、テンポが遅くなる傾向がある。それと同様に、独奏楽器と伴奏楽器で同じフレーズを演奏するときには、独奏楽器の方が速くなる傾向がある。また、類似のフレーズが連続したときも同様の傾向がある。従って、これらの傾向も、このテンポチェンジ手順におけるルールとし

50

て採用することができる。

【0199】

図19は、この発明の一実施例の表情付けモジュールEMによる、同一又は類似パターンが連続する場合の処理例を示すフローチャートである。この処理フローでは、最初のステップJ1でテンポについて強弱の値を設定した後、ステップJ2にて同一又は類似パターンの繰返しを検出し、次のステップJ3にて、MIDIデータにおける検出パターン毎に強弱値と現在のテンポ値に応じた異なるテンポチェンジを付与してステップJ4に進み、このMIDIデータを再生する。そして、ステップJ5にて再生MIDIデータが良好であるとされれば、この同一/類似パターン連続時の処理を終了し、不具合があれば、ステップJ1に戻りステップJ1～J5の処理を繰り返す。

10

【0200】

(9)「似たフレーズは同一乃至類似のテンポ表現を設定する」:

或るフレーズに微妙なテンポ変化を与えるテンポ表情を設定した後、同じフレーズ又は類似のフレーズが別の場所で再現されている場合には、このテンポ変化ルールにより、最初のテンポ表情(微妙なテンポ変化)をそのままコピーしたり、或いは、似てはいるが微妙に異なるテンポ表現を設定することにより、良い表情を付与することができる。全く同一のフレーズでなく、似ているという場合であれば、最初のテンポ表情と似た(少し異なる)テンポを設定するのが良い。また、1回目と2回目とは、(8)の例に則してフレーズ全体のテンポを変える。

【0201】

図20は、この発明の一実施例の表情付けモジュールEMによる、類似フレーズに類似のテンポ表現を設定する処理の例を示すフローチャートである。この処理フローでは、最初のステップK1でテンポについて強弱の値を設定した後、ステップK2にて類似のフレーズを検出し、次のステップK3にて、検出された類似フレーズ間の相違点を検出する。つまり、両ステップK2, K3では、元のMIDIデータODを順次フレーズ解釈し、解釈されたフレーズを順次記憶しておき、新たに解釈されたフレーズと既に記憶されたフレーズとを対比することにより、類似の判断をする。そして、似たフレーズがあると判断したら、何処が違うのか相違点を求めておくのである。

20

【0202】

次のステップK4では、MIDIデータの検出された各フレーズに、検出された相違点と現在設定されている強弱値と現在のテンポ値に応じた異なるテンポチェンジを付与してステップK5に進み、このMIDIデータを再生する。そして、ステップK6にて再生MIDIデータが良好であるとされれば、この類似フレーズへの類似テンポ表現設定処理を終了し、不具合があれば、ステップK1に戻りステップK1～K6の処理を繰り返す。

30

【0203】

(10)「登録した音形に対してテンポを設定する」:

このテンポ表現手法では、予め登録しておいた音形(例えばリズムパターン)が現われると、この部分に所定のテンポ設定を行う。この手法では、(9)の手法とよく似てはいるが、同じフレーズを特定するためには、フレーズのテンプレートとして音形等を予め登録しておき、MIDIデータの音形をこれと比較することも良い方法なので、この方法を採用する。図21はフレーズテンプレートの例を示す。なお、図21(1), (2)に例示されるように、バイオリン演奏における弓の返しや、管楽器演奏におけるプレス(息つき)に相当する位置で区切り()が存在するが、図21(3)に示すように、区切り位置候補()が複数ある場合には、各候補に重みを付けた上これらをランダム選択するようにしてもよい。

40

【0204】

図22は、この発明の一実施例の表情付けモジュールEMによる、登録した音形に所定テンポを設定する処理の例を示すフローチャートである。この処理フローでは、最初のステップL1でテンポについて強弱の値を設定した後、ステップL2にて、登録されている音形と一致するフレーズを検出し、次のステップL3にて、MIDIデータの検出され

50

たフレーズに登録音形と強弱値と現在のテンポ値に応じた所定のテンポチェンジを付与してステップL4に進み、このMIDIデータを再生する。そして、ステップL5にて再生MIDIデータが良好であるとされれば、この登録音形適用処理を終了し、不具合があれば、ステップL1に戻りステップL1～L5の処理を繰り返す。

【0205】

また、フレーズテンプレートとして音形のみを基準にするのではなく、登録したエクスプレッションイベント(expression event)など、他の種類のイベントを参照して、テンポを評価することも良い方法である。例えば、バイオリンでは、弓の返しに応じたエクスプレッションイベントを生成すると、実感的な表情をつくることができるが、弓の返し際には、テンポが遅くなり過ぎて間があくというように、テンポの乱れが生じ易い。このような状況を自動的に付加するために、弓返しを再現するエクスプレッションイベント(例えば、弓が一瞬止まるので音量が低下する)にテンポの変化を連動させると、より実感的な演奏を再現することができる。同様に、管楽器では、プレス(息つき)感を表現するためにエクスプレッションイベント(息つぎのため音量が低下する)を用いる場合があるが、それも同様で、プレス前後では、どうしても間があいてしまうため、それを表現するテンポ変化を行えば良い。

10

【0206】

この発明の一実施例によれば、図21及び図22で説明した登録音形(テンプレート)を用いた処理は、音程パラメータについても同様に適用し、演奏データ(MIDIデータ等)と、登録されたフレーズテンプレートとの比較に基づいて、所定の音程変化を設定することができる。この場合も、同じフレーズを特定するためには、テンポと同様に、テンプレートとして音形を登録しておくのがよく、また、音形を基準にするのではなく、登録しておいたエクスプレッションイベントなどの他のイベントを参照して音程を評価するのも良い方法である。

20

【0207】

例えば、バイオリンなどでは、弓の返しに応じたエクスプレッションイベントを生成すると、実感的な表情を作ることができるが、弓の返し際には弓の圧力の不均一が発生し音程に変化が生じ易い。このような状況を自動的に付加するために、弓返しを再現するエクスプレッションイベントにテンポの変化を連動させると、より実感的な演奏を再現することができる。

30

【0208】

管楽器では、プレス感を表現するためにエクスプレッションイベントで表現する場合があるが、それも同様で、吹奏圧の変化によって音程が変化する。

【0209】

なお、フレーズやイベント列によって、ポルタメントタイムを登録しておき、自動的に設定するというのも効果的である。また、フレーズやイベント列でなく、音色の選択によって、音程変化やポルタメントタイムを自動的に設定することも効果的である。

【0210】

この発明の一実施例によると、楽器によって音程の表現の方法を変更するもう一つの例として、楽器の種類によってスラーの処理方法を変更するということがある。次に例示するように、同じスラーの記号が指定されたとしても、楽器の種類によってその再現方法を変更することが望ましい。

40

【0211】

ピアノのような鍵盤楽器においては、図23(上側)に示すように、連続する2音w1, w2に対してスラーをかける場合、連続音に時間的な重なりがあってもよく、とにかく音を長くすることが好ましいので、例えば、図23(下側)に示すようなスラー処理を行う。

【0212】

一方、クラリネットやトランペットのような管楽器では、スラーの場合、連続する音に時間的な重なりがあると不自然だし、連続する後ろ側の音はアタック音も小さい方がよい

50

。これを実現するためには、MIDIデータ等の演奏データにおいて、図24(上側)の連続する2音w1, w2に対し、図24(下側)に示すように、1つの長い音符として生成し、その途中で、ピッチベンドイベント(pitch bend event)などのデータによって音程を変化させる処理方法を用いるのが良い。

【0213】

(11)「テンポを音量で変化させる」:

音高で変化させるのと同様に、音量でテンポを変化させることも有効な表情付けの方法であるので、このテンポ変化ルールではテンポを音量で変化させる。音量が小さいほどテンポが遅くなるということは、演奏する側には起りがちであるので、このルールに採用する。もちろん、逆に設定できるようにすることもできる。それ以外の手順進行上の処理については、(6)のテンポ変化ルールと同様にして実行することができる。

10

【0214】

図25は、この発明の一実施例の表情付けモジュールEMによる、音量に応じてテンポを変化する処理の一例を示すフローチャートである。この処理フローでは、最初のステップM1でテンポについて強弱の値を設定した後、ステップM2にて、元のMIDIデータODを指定区間(例えば、1小節毎の区間)に分割し、さらにステップM3で各指定区間の音符の平均音量を算出した上、ステップM4に進む。ステップM4では、例えば、音量が小さいほどテンポが遅くなるように予め設定されている「音量-テンポ係数」テーブルを用いて、算出された平均音量からテンポ係数を求める。この場合、当該区間の音符数が「0」のときはテンポ係数値は「1」とする。次のステップM5にて、求められたテンポ係数、強弱の値及び現在テンポ値に応じたテンポチェンジを各指定区間のMIDIデータに付与し、次のステップM6でこのMIDIデータを再生する。この後、ステップM7にて再生MIDIデータが良好であるとされれば、この音量応答処理を終了し、不具合があれば、ステップM1に戻りステップM1~M7の処理を繰り返す。

20

【0215】

図26は、音量に応じてテンポを変化する処理の他の例を示すフローチャートであり、最初のステップN1でテンポについて強弱の値を設定した後、ステップN2にて音量変化をフィルタリングし音量変化カーブを求め、次のステップN3にて、この音量変化カーブ上の各点の音量値を基にして「音量-テンポ係数」テーブルを参照し、テンポ係数を求める。更に、ステップN4において、テンポ係数、強弱の値及び現在テンポ値に応じたテンポチェンジをMIDIデータに付与し、次のステップN5でこのMIDIデータを再生する。この後、ステップN6にて再生MIDIデータが良好であるとされれば、この音量応答処理を終了し、不具合があれば、ステップN1に戻りステップN1~N6の処理を繰り返す。

30

【0216】

(12)「緊張感という言葉でテンポを変化させる」:

テンポは演奏の緊張感と関係が深い。例えば、緊張感が高まると、1つ1つの音を大切に演奏するという気持ちから、テンポが遅くなる傾向がある。このことから、「緊張感」というパラメータによって、テンポを変化させることはインターフェースとして有効である。そこで、このテンポ変化ルールでは、曲全体の感じ或いは「サビ」等の特定部の感じに沿って、「緊張感」という言葉で表わされるパラメータに従ってテンポを変化させる。このパラメータは、例えば、図27に示すように、操作スイッチ装置13上に「緊張感」つまみ13kを設け、つまみ13kを移動操作することにより、任意値に設定可能とすることができる。この「緊張感」つまみ13kによるテンポ設定は、緊張感が「高い」及び「低い」位置に夫々対応する上下最大値に対し、テンポ変化を「+10%」及び「-10%」程度とする。

40

【0217】

図28は、この発明の一実施例の表情付けモジュールEMによる、緊張感に応じてテンポを変化する処理の例を示すフローチャートである。この処理フローの最初のステップP1で「緊張感」つまみ13kを操作して緊張感パラメータ値を設定した後、ステップP

50

2で、つまみ13kの緊張感パラメータ値から、対応するテンポ係数を生成する。次のステップP3では、生成されたテンポ係数及び現在テンポ値に応じたテンポチェンジをMIDIデータに付与し、次のステップP4でこのMIDIデータを再生する。この後、ステップP5にて再生MIDIデータが良好であるとされれば、この緊張感応答処理を終了し、不具合があれば、ステップP1に戻りステップP1～P5の処理を繰り返す。

【0218】

(13)「3拍子の4分音符の羅列では1拍めに長い時間をとる」:

3拍子の4分音符の羅列に限らず、何らかのブロックでまとめられる音符群では、最初の音符を長めにし、使った時間を残りの音符で取り戻すという表現が有効な場合がある。そこで、このテンポ変化ルールでは、まとまった音符群に対して所定の時間的表現を付与する。例えば、「音符群の群感の強さ」というようなパラメータを用意し、まず、全曲を通してこのルールを適応させて、新しい曲データを生成する。そして、再生してみて、このルールによる表情が極端な部分に関しては「弱い」に設定し直し、逆に足りない部分については、「強い」に設定し直すという具合に編集を行う。

10

【0219】

図29は、この発明の一実施例の表情付けモジュールEMによる、3拍子の4分音符の羅列のようなまとまった音符群に対してテンポやタイミングを変化する処理の例を示すフローチャートである。この処理フローでは、最初のステップQ1で音符群の群感の強弱を表わすパラメータ値を設定した後、ステップQ2で所定の音符群を検出し、次のステップQ3にて、検出した音符群に対応したMIDIデータに、群感パラメータ値と現在のテンポ値に応じたテンポチェンジを付与するか、或いは、群感パラメータ値に応じて音符のタイミングを変更する。そして、ステップQ4でこのMIDIデータを再生した後、ステップQ5にて再生MIDIデータが良好であるとされれば、この音符群対応処理を終了し、不具合があれば、ステップQ1に戻りステップQ1～Q5の処理を繰り返す。

20

【0220】

ここで、ステップQ2での音符群検出については、音符を群として取り扱う手法は、拍を単位としても良いし、楽譜を用いている場合なら、音符の旗のつながりを利用することも良い方法である。また、当然、連符は強い群感を示すことになる。また、ステップQ3については、前にも述べたことであるが、時間をかける手法としては、テンポを遅くするばかりでなく、音符の時刻(タイミング)をずらすということも有効な方法である。このテンポ変化ルールでは、音符の時刻をずらす方が扱い易いであろう。

30

【0221】

なお、ピアノの左手だけ、若しくは、伴奏パートだけに適応するのも有効である。例えば、ピアノでワルツ系の曲を弾く場合には、「ブン」-「チャッ」-「チャ」の「ブン」にたくさん時間をかけて、残りは軽く弾くということをテンポ的に変化させることによって実現できる。

【0222】

(14)「5連符以上の連符では、2+3のように簡単な数字の組合せに変換し、その中で1つめの音に時間をかけ強調する」:

(13)のルールでは、連符を音符群として扱うことを述べたが、5連符以上の音符群では、音列をハッキリとさせた演奏とするためには、2連符+3連符のようにもっと小さい数の連符に分解すると良い。このように分解した連符に対しては、(13)のルールによる音符群対応処理を行えば良いが、分解された連符のうち、最初のを強く設定することが好ましい。そこで、このテンポ変化ルールにおいては、5連符以上の長い連符では、例えば、2連符+3連符又は3連符+2連符のように、小連符に分割して簡単な数字の組合せに変換し、各小連符中で1つ目の音に時間をかけて強調するだけでなく、長い連符を小連符の群とみて、最初の小連符はテンポを遅め、残りの小連符はテンポを速めるようにし、より有効な表現付けを行う。

40

【0223】

図30は、この発明の一実施例の表情付けモジュールEMによる、5連符以上の連符の

50

ように長い連符に対してテンポを変化する処理の例を示すフローチャートである。この処理フローの最初のステップR1で強弱の値を設定した後、ステップR2で所定の連符群を検出し、次のステップR3にて、検出した連符を複数の小連符に分割する。さらに、ステップR4では、分割された小連符に対応したMIDIデータについて、強弱値と現在のテンポ値に応じたテンポチェンジを付与するか、或いは、強弱値に応じて音符のタイミングを変更する。そして、ステップR5でこのMIDIデータを再生した後、ステップR6にて再生MIDIデータが良好であるとされれば、この音符群対応処理を終了し、不具合があれば、ステップR1に戻りステップR1～R6の処理を繰り返す。

【0224】

(15)「和音では、その数によってテンポを遅くする」:

10

和音を演奏する場合は、同時発音数が多くなるほど、それらの音を強調するために、時間をたっぷり使おうとしてテンポを遅くする傾向がある。そこで、このテンポ変化ルールでは、和音における同時発音数(和音の音符密度)に応じてテンポを遅くするというような所定の時間的表現を付与する。例えば、「和音の強調」というようなパラメータを用意し、全曲を通してこのルールを適応させて、新しい曲データを生成する。そして、再生してみて、このルールによる表情が極端な部分に関しては「弱い」に設定し直し、逆に足りない部分については「強い」に設定し直すという具合に編集を行う。

【0225】

図31は、この発明の一実施例の表情付けモジュールEMによる、和音の同時発音数に応じてテンポを変化する処理の例を示すフローチャートである。この処理フローの最初のステップS01で強弱の値を設定した後、ステップS02において、和音を検出し、その和音の音符密度を評価して同時発音数を検出する。次のステップS03では、同時発音数と強弱値及び現在のテンポ値に応じたテンポチェンジを算出し、算出したテンポチェンジを、検出した和音に対応したMIDIデータに付与する。そして、ステップS04でこのMIDIデータを再生し、次のステップS05にて再生MIDIデータが良好であるとされれば、この和音同時発音数応答処理を終了し、不具合があれば、ステップS01に戻りステップS01～S05の処理を繰り返す。

20

【0226】

(16)「音色に連動してテンポ係数を設定する」:

このテンポ変化ルールによると、音色毎に対応するテンポ係数値をもつ「音色-テンポ係数」テーブルを予め用意しておき、音色が変わる度に、このテーブルにより自動的にテンポが微妙に変化する表現を得ることができる。アンサンブルの演奏に対して各パート毎にテンポを変えるとかえって不自然であるので、このルールは、ソロの場合又はアンサンブルのメロディパートに適用可能である。このルールでは、音色そのものを基にしてテンポを変化させるのではなく、例えば、EGアタックタイム(envelope generator attack time)のような音色のパラメータで、テンポを変化させる(例えば、ピアノのようにアタックが速い場合はテンポを速くし、バイオリンのようにアタックが遅い場合にはテンポを遅くする等)方が、むしろ、軽い処理でテンポの微妙な変化を実現することができる。

30

【0227】

また、単にテンポを変化させるのではなく、テンポの拍よりも前ノリに音符をずらすということも、音色によっては効果的である。つまり、低音を早く発音すると和音が安定すると言われており、同じ楽器であっても、音高によってずらし方を異ならせてもよい。例えば、ベースギターやバスドラムなどの低音楽器系では、このようなずらし処理が有効になる。そこで、このような処理を有効にする音色を予め設定しておくこと音色を選択するだけで、自動的にノリまで調整できる仕組みを提供することができる。この場合は、ソロやメロディパートに限定する必要はない。

40

【0228】

図32は、この発明の一実施例の表情付けモジュールEMによる、音色や音色パラメータに連動してテンポ係数を設定する処理の例を示すフローチャートである。この処理フロ

50

一の最初のステップT1で強弱の値を設定した後、ステップT2で音色の種類又は所定の音色パラメータを検出し、次のステップT3では、予め用意した「音色種類/音色パラメータ-テンポ係数」テーブルを用いて、検出した音色種類又は所定音色パラメータと強弱値及び現在のテンポ値に応じたテンポチェンジを求め、求めたテンポチェンジをMIDIデータに付与する。或いは、強弱値に応じて音符のタイミングを変更する。そして、ステップT4でこのMIDIデータを再生し、次のステップT5にて再生MIDIデータが良好であるとされれば、この和音同時発音数応答処理を終了し、不具合があれば、ステップT1に戻りステップT1~T5の処理を繰り返す。

【0229】

(17)「フィンガリングを考え、指に応じてテンポ係数を変える」:

10

人間の指は、動き易い指と動き難い指があり、例えば、薬指や小指は、一般的に動きにくいとされている。これは演奏の習熟によって改善される問題ではあるが、動きにくい指ほどテンポが遅くなる。従って、このテンポ変化ルールでは、フィンガリングの指に応じてテンポ係数を設定することによって、人間が演奏した不均一さを表現することができる。

【0230】

図33は、この発明の一実施例の表情付けモジュールEMによる、フィンガリングの指に応じてテンポ係数を設定する処理の例を示すフローチャートである。この処理フローの最初のステップU1で強弱の値を設定した後、ステップU2で、曲データ中の指データからフィンガリングの指を検出し、次のステップU3では、予め用意した「指-テンポ係数」テーブルを用いて、検出したフィンガリング指と強弱値及び現在のテンポ値に応じたテンポチェンジを算出し、算出したテンポチェンジをMIDIデータに付与する。そして、ステップU4でこのMIDIデータを再生し、次のステップU5にて再生MIDIデータが良好であるとされれば、このフィンガリング指応答処理を終了し、不具合があればステップU1に戻りステップU1~U5の処理を繰り返す。

20

【0231】

(18)「フィンガリングを考えポジション移動があればテンポを遅くする」:

フィンガリングにはポジション移動が伴い、ポジション移動があるときとないときでは、当然、演奏のし易さが異なる。特に、音高間隔の広いポジション移動では、さらに演奏の難度が上がる。そこで、このテンポ変化ルールでは、ポジションの有無も含めたポジション移動量に応じてテンポ係数を設定することによって、ポジション移動に伴うテンポのゆらぎを再現し、人間が演奏した不均一さを表現することができる。

30

【0232】

図34は、この発明の一実施例の表情付けモジュールEMによる、フィンガリングのポジション移動に応じてテンポ係数を設定する処理の例を示すフローチャートである。この処理フローの最初のステップV1で強弱の値を設定した後、ステップV2, V3でフィンガリング及びポジションの移動を順次検出し、次のステップV4において、予め用意した「ポジション移動量-テンポ係数」テーブルを用いて、検出したポジション移動と強弱値及び現在のテンポ値に応じたテンポチェンジを算出し、算出したテンポチェンジをMIDIデータに付与する。そして、ステップV5でこのMIDIデータを再生し、次のステップV6にて再生MIDIデータが良好であるとされれば、このフィンガリングポジション移動応答処理を終了し、不具合があればステップV1に戻りステップV1~V6の処理を繰り返す。

40

【0233】

(19)「フィンガリングを考え、低いポジションならテンポを遅くする」:

鍵盤楽器では低いほど鍵盤が重く感じるかも知れないし、弦楽器だと低いポジションほど指を広く開かなければならないので、低いポジションほど速い演奏には向かない傾向がある。また、太い弦は、強く押さないといけないので、速い演奏にはむかない。そこで、このテンポ変化ルールでは、フィンガリングのポジションに応じてテンポ係数を設定することによって、人間が演奏した不均一さを表現することができる。

50

【 0 2 3 4 】

図 3 5 は、この発明の一実施例の表情付けモジュール E M による、フィンガリングのポジションに応じてテンポ係数を設定する処理の例を示すフローチャートである。この処理フローの最初のステップ W 1 で強弱の値 を設定した後、ステップ W 2 , W 3 でフィンガリング及びポジションを順次検出し、次のステップ W 4 において、予め用意した「ポジション - テンポ係数」テーブルを用いて、検出したポジションと強弱値 及び現在のテンポ値に応じたテンポチェンジを算出し、算出したテンポチェンジを M I D I データに付与する。そして、ステップ W 5 でこの M I D I データを再生し、次のステップ W 6 にて再生 M I D I データが良好であるとされれば、このフィンガリングポジション応答処理を終了し、不具合があればステップ W 1 に戻りステップ W 1 ~ W 6 の処理を繰り返す。

10

【 0 2 3 5 】

(2 0) 「ピッチベンドをかけるときは、自動的にテンポも触る」：

フィンガリング評価の一種でもあるが、ピッチベンドをかけるということは、ポジション移動（スライドやグリッサンド）を伴うことが多いので、ピッチベンド等がかかっているとところを目印にしてテンポを遅くするということが有効な場合がある。そこで、このテンポ変化ルールにより、ピッチベンドをかけるときは、ベンドをかけるるとテンポが遅くなるようなテンポ変化を自動的に与えるようにする。

【 0 2 3 6 】

図 3 6 は、この発明の一実施例の表情付けモジュール E M による、ピッチベンドに応じてテンポ設定を行う処理の例を示すフローチャートである。この処理フローの最初のステップ X 1 で強弱の値 を設定した後、ステップ X 2 でピッチベンドを検出し、次のステップ X 3 では、検出したピッチベンドと強弱値 及び現在のテンポ値に応じたテンポチェンジを算出し、算出したテンポチェンジを M I D I データに付与する。そして、ステップ X 4 でこの M I D I データを再生し、次のステップ X 5 にて再生 M I D I データが良好であるとされれば、このピッチベンド対応処理を終了し、不具合があればステップ X 1 に戻りステップ X 1 ~ X 5 の処理を繰り返す。

20

【 0 2 3 7 】

(2 1) 「ギターのプリングオフ等のときにはテンポを速くする」：

(2 0) のピッチベンドとは逆に、ギターのハンマリングオン (h a m m e r i n g o n) 、プリングオフ (p u l l i n g o f f) 等のときには、新たに弦をはじかない分、テンポが速くなる傾向がある。そこで、このテンポ変化ルールでは、プリングオフ等のときにはテンポを速くする。このためには、例えば、ピッチベンド検出部分で、ハンマリングオン、プリングオフ等を検出すれば良く、この検出には、例えば、ピッチベンドで急に半音以上ピッチを変えたとか、楽譜記号でハンマリングオンなどを記入してあるとかを利用する。

30

【 0 2 3 8 】

図 3 7 は、この発明の一実施例の表情付けモジュール E M による、ギターのプリングオフ等のときにはテンポを速くする処理の例を示すフローチャートである。この処理フローの最初のステップ Y 1 で強弱の値 を設定した後、ステップ Y 2 でハンマリングオン、プリングオフ等を検出し、次のステップ Y 3 では、検出したハンマリングオン、プリングオフ等と強弱値 及び現在のテンポ値に応じたテンポチェンジを算出し、算出したテンポチェンジを M I D I データに付与する。そして、ステップ Y 4 でこの M I D I データを再生し、次のステップ Y 5 にて再生 M I D I データが良好であるとされれば、このプリングオフ等応答処理を終了し、不具合があればステップ Y 1 に戻りステップ Y 1 ~ Y 5 の処理を繰り返す。

40

【 0 2 3 9 】

(2 2) 「ピアノのサスティンペダル操作に応じてテンポ係数を設定する」：

ピアノでは、サスティンペダルを踏むと響きを豊かになるが、この場合、速く弾くと音の分離が悪くなるので、微妙にテンポを遅くする傾向がある。そこで、このテンポ変化ルールでは、ピアノのサスティンペダル操作に応じて所定のテンポ係数を設定する。この

50

ためには、ピッチバンド検出部分で、サスティンペダルオン等を検出すれば良い。

【0240】

図38は、この発明の一実施例の表情付けモジュールEMによる、ピアノのサスティンペダル操作に応じてテンポ係数を設定する処理の例を示すフローチャートである。この処理フローの最初のステップZ1で強弱の値を設定した後、ステップZ2でサスティンペダルオンを検出し、次のステップZ3では、検出したサスティンペダルオンと強弱値及びテンポデータに応じたテンポチェンジを算出し、算出したテンポチェンジをMIDIデータに付与する。そして、ステップZ4でこのMIDIデータを再生し、次のステップZ5にて再生MIDIデータが良好であるとされれば、このサスティンペダル応答処理を終了し、不具合があればステップZ1に戻りステップZ1～Z5の処理を繰り返す。

10

【0241】

(23)「ビブラートの深さや速さに応じてテンポ係数を設定する」:

ビブラートを深くかけるということは、たっぷりとした響きを出したいときであるから、テンポも遅めになる傾向がある。また、ビブラートを速くかけるときには、緊張感のある音を出そうとしているときなので、1つ1つの音を確かめるように演奏するためにテンポも遅めになる。逆に、ビブラートを遅くかけるときには、落ち着いた響きを出そうとしているときなので、これもテンポは遅めになる傾向にある。

【0242】

そこで、このテンポ変化ルールでは、図39に示すような「ビブラート深さ - テンポ係数」テーブル及び図40に示すような「ビブラート速さ - テンポ係数」テーブルを用いて、ビブラートの深さや速さに応じてテンポ係数Kを設定する。つまり、上述したことから、ビブラートとテンポ変化の関係は、ビブラートの深さについては、図39のように、深いほどテンポが遅くなるのに対して、ビブラートの速さについては、図40のように、速くても遅くてもテンポが遅めになるという中ぶくらの曲線になっているのが好ましいのである。なお、図39及び図40の各テーブルにおいて、ビブラートの深さ及び速さの値が「0」の場合は、例外的にテンポ係数Kを「1」とする。

20

【0243】

実施に当たっては、バンド検出部分で、ビブラートを検出し、さらに、各テーブルにおけるビブラートの深さや速さのパラメータの設定を検索すれば良い。もし、ビブラートのパラメータがないときには、出力波形からビブラートの深さや速さの情報を得ても良い。

30

【0244】

図41は、この発明の一実施例の表情付けモジュールEMによる、ビブラートの深さや速さに応じてテンポ係数を設定する処理の例を示すフローチャートである。この処理フローの最初のステップAa1で強弱の値を設定した後、ステップAa2, Aa3でビブラート及びビブラートの深さ及び速さを順次検出する。次のステップAa4では、図39に示すような「ビブラート深さ - テンポ係数」テーブル及び図40に示すような「ビブラート速さ - テンポ係数」テーブルを用いて、検出したビブラートの深さ及び速さに対応するテンポ係数Kを求め、このテンポ係数Kと強弱値及び現在のテンポ値に応じたテンポチェンジを算出し、算出したテンポチェンジをMIDIデータに付与する。そして、ステップAa5でこのMIDIデータを再生し、次のステップAa6にて再生MIDIデータが良好であるとされれば、このビブラート速さ/深さ応答処理を終了し、不具合があればステップAa1に戻りステップAa1～Aa6の処理を繰り返す。

40

【0245】

(24)「ストリングスのトリルは、複数のパートを使い微妙にタイミングをずらす」:

一般に、ストリングス等のトリルは、図42(1)に示される「楽譜上の音符」に対して、図42(2)に示す「演奏上の楽譜」のように、1つの長い音符を複数の短い音符に分解することで実現する。しかし、実際のストリングスパートでは複数の奏者が演奏するわけであって、ストリングスのトリルに対しぴったり同期をとるとおかしいので、全演奏者が同じタイミングでトリルの細かい音符を演奏するとは考え難い。そこで、このタイミング変更ルールによって、複数のパートを使いこれらのパートについて微妙にタイミング

50

をずらすのである。つまり、このような同期演奏状態を回避し、自然な演奏をMIDI音源のストリングスで再現するのに、複数のパートを使うと共にそれぞれのパートでのトリルのタイミングを微妙にずらせることによって有効な結果を得ることができる。

【0246】

なお、複数のパートは微妙に異なる音色（例えば、バイオリンとビオラ等）であることが好ましい。また、トリルのタイミングを微妙にずらす方法としては乱数を用いて、トリルの細かい音符のオン時刻やデュレーション（duration）を変更するのが効果的である。

【0247】

図43は、この発明の一実施例の表情付けモジュールEMによる、ストリングスのトリルに
10 応じて複数パートに分け且つ各パートのタイミングをずらす処理の例を示すフローチャートである。この処理フローの最初のステップBb1で強弱の値を設定した後、ステップBb2でトリル部分を検出し、次のステップBb3でトリル部分を複数パートにコピーする。この場合、パート毎に微妙に音色を変更するのが好ましい。その後、ステップBb4では、トリル部分のMIDIデータのタイミングを、強弱値に応じてパート毎に異なるように変更する。そして、ステップBb5でこのMIDIデータを再生し、次のステップBb6にて再生MIDIデータが良好であるとされれば、このストリングストリルの
20 タイミング処理を終了し、不具合があればステップBb1に戻りステップBb1～Bb6の処理を繰り返す。

【0248】

この発明の一実施例によれば、図42及び図43で説明したトリルの複数パート処理の
20 手法は、音程パラメータの変化付与に適用することができる。つまり、ストリングスのトリルはぴったりと同期をとるとおかしいので、複数のパートに分割し、微妙に音程をずらす表情付けを行う。これをMIDI音源のストリングスで再現するには、ステップBb1で音程のずらし幅について強弱の値を設定し、ステップBb2、Bb3で図43と同様の処理をして複数パートに分解（分割）した後、ステップBb4において、分解したそれぞれのパートにおけるトリルの音程を微妙にずらせることによって有効な結果を得ることができる。この場合、ずらせるパラメータには、音程だけでなく、タイミングを含ませるのが好ましい。

【0249】

(25)「学習機能によりテンポを設定する」:

このテンポ付与手法は、テンポ設定に学習機能を持たせ、音高の変化とテンポの変化の
30 関係などを自動的に予想するシステムを構成し、ユーザによるテンポ変化ルールを自動構築を可能にするものである。一例としては、ある曲の途中までテンポ変化を手入力した後、残りのテンポ変化は、学習機能により自動入力する。この場合、残りの曲データが入力されるとフレーズ解釈を行い、既に処理したフレーズと比較し、同じフレーズには、処理済みフレーズのテンポ変化と同じテンポ変化を与え、新規フレーズについてはテンポ変化をランダムにするなどというように処理する。また、他の例としては、ある曲のテンポ変化を分析しておき、他の曲を解釈し、分析済の曲との対比結果に応じて、対応する分析済テンポ変化を付与する。
40

【0250】

図44は、この発明の一実施例による表情付けモジュールEMの学習機能に基づく処理の例を示すフローチャートである。この処理では、学習ルーチンCc5において、テンポ変化を既に付与したMIDIデータからフレーズとテンポ変化の関係を学習し、学習結果をRAM3内の所定エリアに格納しておく。

【0251】

次に、テンポ変化を付与するに当たっては、ステップCc1で強弱の値を設定した後、
50 ステップCc2で、未だテンポ変化を付与していないフレーズのMIDIデータに対して、学習結果を基にして強弱値と現在のテンポ値に応じたテンポチェンジを付与する。そして、ステップCc3でこのMIDIデータを再生し、次のステップCc4にて再生MIDI

D Iデータが良好であるとされれば、この学習によるテンポ付与処理を終了し、不具合があればステップC c 1に戻りステップC c 1 ~ C c 4の処理を繰り返す。

【0252】

(26)「ライブラリを用いてテンポ変化を設定する」:

演奏データの種々の特徴情報に対応して一旦生成したテンポ変化をある時間区間で切り取ってライブラリ化しておく、他の部分にも同様に適応できる場合がある。このテンポ付与手法は、特徴情報に対するテンポ変化をライブラリ化することによって、使い易いテンポチェンジ手法を提供するものである。ライブラリ化に際しては、ライブラリに名前を付けて保存すると、尚一層使い易くなる。また、相対テンポ値で記憶したり、時間変化方向又はテンポ値変化方向に伸縮可能とするのがよい。

10

【0253】

図45は、この発明の一実施例による表情付けモジュールE Mの、ライブラリを用いてテンポ変化を設定する処理の例を示すローチャートである。この処理では、ライブラリ化ルーチンD d Lにおいて、テンポ変化を既に付与したM I D Iデータからテンポチェンジを抽出し、抽出したテンポチェンジを相対値に変換してライブラリ化して外部記憶装置9の所定エリアに保存しておく。

【0254】

次に、テンポ変化を付与するに当っては、ステップD d 1において、M I D Iデータの所定の特徴情報に対応してライブラリからテンポチェンジを選択し、選択されたテンポチェンジを時間方向及び/又はテンポ値方向に所定倍率で伸縮する。次のステップD d 2では、選択され伸縮されたテンポチェンジを現在のテンポ値に応じて絶対値に変換し、変換されたテンポチェンジをM I D Iデータに付与する。そして、ステップD d 3でこのM I D Iデータを再生し、次のステップD d 4にて再生M I D Iデータが良好であるとされれば、このライブラリ使用によるテンポ付与処理を終了する。一方、不具合があれば、ステップD d 1に戻って倍率を変更するか、或いは、他のテンポチェンジを選択した上、ステップD d 2 ~ D d 4の処理を繰り返す。

20

【0255】

この発明の一実施例によれば、図45で説明したライブラリを用いた処理の手法は、音程パラメータの変化付与に適用し音程変化をライブラリ化することができる。つまり、図45と同様の処理によって一旦生成した音程変化を、ある時間区間で切り取りライブラリ化すると、他の部分についても、同様に適応することができる場合がある。この場合も、ライブラリに名前をつけて保存すると、なお一層使い易くなることはいうまでもない。

30

【0256】

(27)「歌詞(詩)に応じてテンポ係数を設定する」:

歌を伴う曲の場合は、同じメロディーでも、歌詞の内容によってテンポ感が変わる場合がある。そこで、このテンポ変化手法では、ある単語をテンポ係数とともに予め登録しておき、その単語が出現するとテンポを変化させるというような手順を用いて、歌詞(詩)に応じてテンポ係数を設定する。テンポ係数を予め設定するに当っては、明るい単語についてはテンポが速いとするのに対し、暗い単語についてはテンポが遅いとし、重要単語に対しては遅いテンポとして、進遅の程度に対応する係数値を登録しておけばよい。また、単語の部分のみをテンポ変化させたり、或いは、単語を含む区間全体をテンポ変化させたり、テンポ変化の対象部分を指定可能としてもよい。

40

【0257】

図46は、この発明の一実施例による表情付けモジュールE Mの、歌詞(詩)に応じてテンポ係数を設定する処理の例を示すローチャートである。この処理では、最初のステップE e 1にて強弱の値を設定した後、ステップE e 2で、元のM I D IデータO Dの歌詞データ中から所定の単語を検出する。次のステップE e 3では、検出された単語に対応して設定されたテンポ係数と設定している強弱値及び現在のテンポ値に応じたテンポチェンジを、M I D Iデータに付与する。そして、ステップE e 4でこのM I D Iデータを再生し、次のステップE e 5にて再生M I D Iデータが良好であるとされれば、この歌詞

50

応答処理を終了し、不具合があればステップ E e 1 に戻りステップ E e 1 ~ E e 5 の処理を繰り返す。

【 0 2 5 8 】

(2 8) 「音源の出力波形に応じてタイミングを補正する」：

同じ音源の音でも、音色の種類によって聴感上の発音タイミングが異なる。そこで、このタイミング変化ルールでは、実際の出力波形を観測しながら音符のオン時刻を補正する仕組みを用いて、音源の出力波形に応じてタイミングを補正する。図 4 7 は、音源出力波形と補正タイミングの関係を表わす図である。同図において、或る音色の音源の出力波形 T W が、タイミング t s で発音を開始して立ち上がり、タイミング t m で最大音量値 (1 . 0 0) に到達するものとする。例えば、音源出力波形 T W の音量値が最大値の 8 0 % に到達する時点 t c を、本来の発音開始タイミングとなるように、前にずらすことによって、聴感上の発音タイミングに合わせることができる。また、この補正タイミング t c は、最大音量値の 8 0 % に限らず、音源種類等の事情に応じて妥当な値に選定され、例えば、1 0 0 % (タイミング t m) でもよい。

10

【 0 2 5 9 】

具体的手順としては、例えば、曲データ制作の場合なら、一度再生しながら、タイミングの補正データを生成していく。以降は、生成した補正データによってタイミング補正したものを再生すれば、音色によるタイミングのずれを補正し、更には、音源によるタイミングのずれまで補正することができることになる。一方、リアルタイム演奏 (再生) の場合なら、最初の一音は発音が始まってからしか補正できないが、2 音目以降は、補正量を予測しながら、また、予測値を学習によって修正しながら、タイミングの補正を行っていけば良い。さらに、曲データの再生前に、全音色、全音高の出力波形を観測しておけば、再生しなくても最初の一音から補正が可能となる。

20

【 0 2 6 0 】

図 4 8 は、この発明の一実施例による表情付けモジュール E M の、音源の出力波形に応じてタイミングを補正する処理の例を示すローチャートである。この処理では、最初のステップ F f 1 にて強弱の値 を設定した後、ステップ F f 2 では、音源の出力波形 T W を観測する。次のステップ F f 3 では、この観測結果に応じてタイミング補正量 (t s - t c) を算出し、さらに、ステップ F f 4 にて、算出された補正量と強弱の値 に応じて、M I D I データのタイミングを補正する。そして、ステップ F f 5 でこの M I D I データを再生し、次のステップ F f 6 にて再生 M I D I データが良好であるとされれば、この出力波形対応処理を終了し、不具合があればステップ F f 1 に戻りステップ F f 1 ~ F f 6 の処理を繰り返す。

30

【 0 2 6 1 】

(2 9) 「ピアノ (p) の直前のフォルテ (f) は短かめにする」：

強弱記号のフォルテ (f) が続いた後ピアノ (p) が現われる場合、ピアノ (p) の直前のフォルテ (f) の音を短かめにする。ピアノ (p) の音が聞き取り易くなる。このテンポ変化ルールでは、このような強弱記号を伴う音のテンポ変化を制御する。

【 0 2 6 2 】

図 4 9 は、この発明の一実施例による表情付けモジュール E M の、強弱記号に対する処理の例を示すローチャートである。この処理では、最初のステップ G g 1 にて強弱の値を設定した後、ステップ G g 2 では、元の M I D I データ O D 中からフォルテ (f) が続いた後のピアノ (p) を検出する。次のステップ G g 3 では、検出したピアノ (p) の直前にあるフォルテ (f) 音のゲートタイムを強弱値 に応じて短縮する。そして、ステップ G g 4 で、この短縮処理を施した M I D I データを再生し、次のステップ G g 5 にて再生 M I D I データが良好であるとされれば、この強弱記号対応処理を終了し、不具合があればステップ G g 1 に戻りステップ G g 1 ~ G g 5 の処理を繰り返す。

40

【 0 2 6 3 】

(3 0) 「スタッカートの直前の音は長めにする」：

スタッカート (s t a c c a t o) の直前の音は長めにする。スタッカートらしく切

50

れのよい音に聞こえる。このテンポ変化ルールでは、このスタッカートを伴う音のテンポ変化を制御する。

【0264】

図50は、この発明の一実施例による表情付けモジュールEMの、スタッカートに対する処理の例を示すフローチャートである。この処理では、最初のステップHh1にて強弱の値を設定した後、ステップHh2では、元のMIDIデータOD中からスタッカートを検出する。次のステップHh3では、検出したスタッカートの直前の音のゲートタイムを強弱値に応じて伸長する。そして、ステップHh4で、この伸長処理を施したMIDIデータを再生し、次のステップHh5にて再生MIDIデータが良好であるとされれば、このスタッカート対応処理を終了し、不具合があればステップHh1に戻りステップHh1~Hh5の処理を繰り返す。

10

【0265】

(31)「テンポは全体を見通してから決める」:

これまで説明してきたように、テンポ変化させる要因は様々である。1曲を通して様々なテンポ変化を加えた結果、変化前のテンポと比較して大きく異なるものになっていることがあるかも知れない。このテンポ決定手法では、そのような事態に備えて、1曲を通してテンポ変化させた結果を見て、その結果の平均が元々設定されていたテンポの値になるように(平均をとる方法は、必要に応じて任意の方法を採用することができる。)、全体的なテンポ補正を行う。この全体的なテンポ補正をするには、例えば、全体のテンポを均一に補正したり、或いは、全般的な均一補正ではなく、テンポ変更回数の多い区間のテンポを優先的に補正したりする。なお、元のテンポと平均テンポとの差が所定の範囲内であればテンポを補正しないというような許容量を、予め選定しておいたり、或いは、ユーザにより選定可能としてもよい。

20

【0266】

図51は、この発明の一実施例による表情付けモジュールEMの全体的な見直し処理の例を示すフローチャートである。この処理の最初のステップJj1では、これまで説明してきたテンポ変化ルール(1)~(30)から必要に応じて採用された所定ルールに基づいて、個々のテンポ変化を元のMIDIデータODに付与する。そして、ステップJj2において、個々のテンポ変化が付与されたMIDIデータの全体のテンポが元の平均テンポと一致するか又は近似したものとなるように、或いは、総演奏時間が元の総演奏時間と一致するか又は近傍のものとなるように、テンポを全体的に修正する。

30

【0267】

その後、ステップJj3に進んで、所望のテンポ又は総演奏時間が得られたかを、自動計算やMIDIデータの1曲全部の再生試聴等により、検証して、良好であると判定されればこの全体的見直し処理を終了する。一方、不具合があれば、ステップJj4に進んで個々のテンポ変化処理を行うか否かが打診される。そして、個々のテンポ変化処理を行う必要がある判断した場合(YES)には、ステップJi1に戻ってステップJj1~Jj2の処理を再度行い、そうでない場合(NO)にはステップJj2の処理を再度行う。

【0268】

この発明の一実施例によれば、図51で説明した全体的見直し処理の手法は、音程パラメータの変化付与に適用し、全体を見通してから音程を再度調整することができる。様々な音程変化を加えて表情付けを行っていくことにより、最終的に曲全体が或る値の音程でシフトされる結果となる場合がある(例えば、ピッチベンドデータによる)。このような場合には、全体を見通してから、マスターチューニングやピッチシフト、ファインチューニングという別のパラメータにまとめた方が扱いやすいことがある。

40

【0269】

〔実施形態2〕

次に、具体的な演奏パラメータとして主に音量パラメータを中心にして(具体例中には、音程パラメータ等の他の演奏パラメータを含むものもある)表情付けを行う「実施形態2」について説明する。図52及び図53は、図2に示されるシステムにおける表情付け

50

モジュールの機能の概要について、パラメータ自動編集の観点から、CPU 1の実行する処理フローとして説明するフローチャートである。まず、図52及び図53を参照してその概要を説明し、次に、図54～図84を参照して「実施形態2」の具体例(A)～(V)につき詳細に説明するものとする。

【0270】

本実施の形態のパラメータ自動編集装置は、主として、次の制御処理を実行する。すなわち、

〔1〕供給された演奏データを解析し、その解析結果に基づいて変化付与を行うパラメータの選択およびその内容の決定を行い、当該演奏データにパラメータ変化を付与するパラメータ変化付与処理

〔2〕上記〔1〕の処理により表情付けされた演奏データを再生する再生処理

〔3〕上記〔1〕の処理により表情付けされた演奏データを確認する確認処理である。

【0271】

ここで、演奏データとしては、本実施の形態では、たとえばSMF(Standard MIDI File)のフォーマットで作成されたものを想定しているが、これと異なったフォーマットのものを採用してもよい。また、〔3〕の確認処理における演奏データの確認とは、具体的には、演奏データを何らかの形態で表示装置(ディスプレイ)14上に表示させ、その表示された演奏データ上に上記〔1〕の処理で付与したパラメータ値やその位置を重ねて表示させることにより、ユーザに〔1〕の処理結果を確認させることをいう。そして、この確認処理中で、〔1〕のパラメータ変化付与処理を施したくない位置をユーザが指定できるようにしてもよい。

【0272】

なお、上記〔2〕及び〔3〕の再生及び確認処理として、「実施形態1」の各実施例で説明したように、音量や音程等の所定の演奏パラメータの強弱値を設定し(例えば、ステップA2)、当該演奏パラメータを付与した演奏データ(MIDIデータ等)を再生し、パラメータ付与後演奏データの良否判断(例えば、ステップA7)の結果に応じて強弱値を設定し直すという手法を採用し得ることは、いうまでもない。

【0273】

図52は、本実施の形態のパラメータ自動編集装置、特にCPU1が実行するメインルーチンの手順を示すフローチャートである。

【0274】

同図において、まず、RAM3のクリアや、最初に処理(上記〔1〕～〔3〕のいずれかの処理)すべき演奏データの選択等を行う初期化処理を行う(ステップS1)。

【0275】

次に、ユーザの選択した演奏データを、たとえば各種記憶媒体(前記ハードディスク、FD、CD-ROM等)や外部(前記他のMIDI機器17およびサーバコンピュータ20等)から検索して、たとえばRAM3の所定位置に確保された演奏データ記憶領域に格納する(ステップS2)。

【0276】

次に、この演奏データに対して、上記〔1〕のパラメータ変化付与処理(その具体的な処理手順は、図53を用いて後述する)を実行する(ステップS3)。なお、図53のパラメータ変化付与処理は、後述する複数(本実施の形態では、19個)のパラメータ変化付与処理全てに共通した処理を抽出して構成したものである。したがって、ステップS3では、より具体的には、パラメータ変化付与処理1～19のうち、ユーザが選択した少なくとも1つ以上を実行している。

【0277】

さらに、このパラメータ変化付与処理以外のその他処理を実行する(ステップS4)。その他処理では、上記〔2〕および〔3〕の処理が行われるが、それに加えて、たとえば演奏データの新規作成処理を行うようにしてもよい。

10

20

30

40

50

【0278】

そして、ユーザが本メインルーチンを終了させる操作を行ったか否かを判別し（ステップS5）、その操作を行ったときには、本メインルーチンを終了する一方、その操作を行わないときには、前記ステップS2に戻って、ステップS2～S5の処理を繰り返し行う。

【0279】

図53は、ステップS3のパラメータ変化付与処理の手順を示すフローチャートである。

【0280】

同図において、まず、前記演奏データ記憶領域に格納されている演奏データを解析する（ステップS11）。演奏データの解析とは、具体的には、演奏データ中、パラメータの値を変化させた方がよい個所を抽出することをいう。

10

【0281】

次に、抽出された個所のそれぞれに対して、その個所に最適なパラメータ値（変化値）を決定する（ステップS12）。

【0282】

そして、決定された各パラメータ値を、演奏データ中の当該位置に付与した（ステップS13）後に、本パラメータ変化付与処理を終了する。

【0283】

なお、図52及び図53により説明した処理は、音量パラメータのほか、先に説明したテンポやタイミング等の時間的制御情報（時間パラメータ）に変化を付与したり、音程等の他のパラメータに変化を付与するのに用いられることは、いうまでもない。

20

【0284】

この発明は、このパラメータ変化付与処理に特徴があり、以下、その具体的な処理方法を詳細に説明する。

【0285】

（A）音符が上昇している部分での処理

図54は、音量パラメータ変化付与処理1の手順を示すフローチャートであり、本音量パラメータ変化付与処理1では、演奏データの音量パラメータをその音高の変化に応じて変化させるという表情付けを行う。具体的には、次のアルゴリズムに従って、音量パラメータを変化させる。すなわち、

30

（1）ノートイベント列の音高（ノートナンバ）が上昇傾向にあるときには、音量を徐々に増大させる

（2）ノートイベント列の音高が上昇傾向から下降傾向に変化したときには、その変化点のノートイベントにアクセントを付与する

ここで、ノートイベントは、通常、ノートオンイベントおよびノートオフイベントの両者を含む意味で用いられるが、本実施の形態では、ノートオフイベントを考慮していないので、ノートイベントは、ノートオンイベントのみの意味で用いることにする。

【0286】

図54において、まず、前記演奏データ記憶領域に格納されている演奏データ（以下、この演奏データを「選択演奏データ」という）中、音量変化を付与したい区間をユーザが指定すると、その指定区間を、RAM3のワークエリアに格納して、解析区間とする（ステップS21）。なお、区間は指定せずに、演奏データの全区間を解析区間としてもよいことは言うまでもない。ここで、演奏データはシーケンスデータであるため、各演奏データは時間軸に対して広がりをもっている。このため、区間という概念を、そこに持ち込むことができる。

40

【0287】

次に、解析区間からノートイベント列の音高が上昇傾向および下降傾向にある部分区間を検索して切り出す（ステップS22）。

【0288】

50

図55は、音高が上昇傾向にあるノートイベント列の一例を示す図であり、(a)は、単純な上昇音系の一例を示し、(b)は、単純な上昇音系ではないが、全体としては上昇音系と言える一例を示している。すなわち、本実施の形態では、図55(a)に示すノートイベント列の区間を、音高が上昇傾向にある部分区間として切り出すだけでなく、図55(b)に示すノートイベント列の区間も、音高が上昇傾向にある部分区間として切り出すようにしている。以下、この部分区間を検索する方法について説明する。

【0289】

まず、選択演奏データからノートイベントのみを抽出し、そのノートイベントに含まれるノートナンバを時系列に並べることにより、時系列のノートナンバ列を生成する。このノートナンバ列に対して、ハイパスフィルタ(HPF)処理を施すことにより、音程の変化傾向を算出する。なお、このHPF処理のみでは、その算出値の変化が時間的に細かすぎる場合には、さらに、その算出値(データ列)に対してローパスフィルタ(LPF)処理を施し、その変化を時間的に滑らかにする。ノートナンバ列に対して、このようなフィルタ処理を施すことにより、図55(b)のような全体としては上昇音系と言える音系を、図55(a)のような単純な上昇音系と同様に検索できるようになる。

10

【0290】

図56は、このようにして算出された音程の変化傾向を表わす時系列データの一例を示す図であり、縦軸は、その正側が音程上昇傾向を示し、その負側が音程下降傾向を示し、横軸は時刻を示している。すなわち、同図において、解析区間 $t_0 - t_7$ のうち、区間 $t_0 - t_1$ 、 $t_2 - t_3$ 、 $t_4 - t_5$ 、 $t_6 - t_7$ の各区間を、音高が上昇傾向にある部分区間として切り出す。

20

【0291】

また、音高が下降傾向にある部分区間も、音高が上昇傾向にある部分区間を切り出す過程で、簡単に切り出すことができる。すなわち、図56中、時系列データが負の値を採る区間 $t_1 - t_2$ 、 $t_3 - t_4$ 、 $t_5 - t_6$ の各区間が、音高が下降傾向にある部分区間に相当するので、この各区間を音高が下降傾向にある部分区間として切り出せばよい。

【0292】

なお、切り出し区間の長さは、小節長や拍長の整数倍であってもよいし、任意の長さであってもよい。そして、図56に示すように、音高が上昇傾向にある部分区間が1解析区間内に複数個あるときには、そのすべての部分区間を切り出すようにする。

30

【0293】

図54に戻り、切り出された部分区間毎に、当該区間に属するノートナンバ列の音高の変化速度を算出し、この算出結果に応じて、当該区間に属するノートイベント列の各ノートイベントに付与すべき音量パラメータの変化付与パターンを決定する(ステップS23)。ここで、音高の変化速度とは、音高変化の傾き、すなわち、単位時間当たりの音高変化量を意味する。そして、変化付与パターンとは、予め作成されて、たとえばROM2に記憶されている音量パラメータの変化傾向を表わすテンプレートのことをいう。このテンプレートは、具体的には、元の音量パラメータ値を置き換えるデータまたは元の音量パラメータ値を修正するデータのいずれかを時系列データとしたものである。そして、本実施の形態では、音量パラメータの変化傾向とは、音高が上昇傾向にある部分区間では、音量パラメータの値を徐々に増大させて行くという変化傾向をいい、この増大率を各テンプレート毎に変化させ、音高の変化速度が大きい部分区間に対しては、増大率が大きく設定されているテンプレートが選択される。

40

【0294】

なお、変化付与パターンとして、上記テンプレートを用いる代わりに、所定の演算式により算出するようにしてもよい。

【0295】

次に、切り出された部分区間毎に、当該区間に属するノートイベント列の各ノートイベントに付与すべき音量パラメータ値を、前記決定された変化付与パターンに基づいて変更する(ステップS24)。なお、変更すべき音量パラメータが存在しない場合には、その

50

音量パラメータを追加すればよい。ここで、音量パラメータとして、具体的に、ノートイベントに含まれるベロシティを採るようにすれば、音量パラメータ値の変更とは、ベロシティ値の変更を意味する。この他、エクスプレッションデータを当該区間内の各ノートイベントに付随させて挿入するようにしてもよい。

【0296】

次に、ステップS22で切り出された部分区間を時系列に並べて、音高が上昇傾向から下降傾向に変化する時点、すなわち、図56において、フィルタ処理後の時系列データの符号が正から負に変化する時点(時刻 t_1 , t_3 , t_5)を検出し、その時点の音符、すなわち切り出された部分区間の先頭位置または最後尾の音符(ノートイベント)にアクセントを示す音量データを付与する(ステップS25)。

10

【0297】

そして、本音量パラメータ変化付与処理1を終了させる操作をユーザが行ったか否かを判別し、行ったときには、本音量パラメータ変化付与処理1を終了する一方、行わないときには、前記ステップS21に戻って、ステップS21~S26の処理を繰り返し行う。

【0298】

この発明の一実施例によれば、図54~図56を用いて説明した「音量パラメータ変化付与処理1」の手法は、音程パラメータに変化を付与するのに適用可能である。つまり、音高上昇系ではだんだん音程が低くずれて所謂「上がり切らない」演奏状況が生じるが、これに対応して、演奏データ(MIDIデータ等)から音高が上昇している部分を検索し(ステップS22)、HPF(場合によってはLPFも)を用いて音高上昇の速さを評価し(ステップS23)、所定の音程付与パターン等に基づいて音程変化を算出し(ステップS24)、これを演奏データに付与する(ステップS25)ことにより、音符の音高が上昇している部分では、keyよりも細かい単位で音程を徐々に低くするという表情付けを実現することができる。

20

【0299】

このように、演奏データからHPF等を用いて音高上昇の速さを検出し、検出された値に従った音程変化を行うと、激しい変化で上昇する場合はより大きく音程が低くずれ、また、逆に、激しい変化で下降する場合には、より大きく音程が高くずれるということが実現できる。しかしながら、場合によっては、下降であろうが、上昇であろうが、変化が大きいほど音程が低くずれるようにしたい場合もある。その場合には、HPFにより検出される絶対値に応じて音程変化を算出する構成にすればよい。

30

【0300】

なお、このように音高上昇系において音程を低くずらす表情付け処理は、鍵盤楽器など、音程がずれる要素の少ないものでは、この機能を使わない方が自然である。従って、演奏データを解析時(例えば、ステップS22)に音色も判断して楽器種別を確認しておく必要がある。

【0301】

この表情付け処理のシステムを利用すると、演奏音域の変化を観測することができるので、演奏音域に大きな変化があったときには、ポルタメント(フレットのない弦楽器の場合)やスライド(フレットのある弦楽器の場合)を加えるという自動表現も可能になる。

40

【0302】

(B) 曲の進行に伴う盛り具合を表わすための処理

図57は、音量パラメータ変化付与処理2の手順を示すフローチャートであり、本音量パラメータ変化付与処理2では、演奏データの音量パラメータの値を徐々に増大させることにより、興奮度を高めるといった表情付けを行う。

【0303】

同図において、まず、選択演奏データの全時間長を算出する(ステップS31)。

【0304】

次に、変化付与パターンを選択する(ステップS32)。変化付与パターンとは、曲全体の音量変化傾向を表わすパターンデータをいい、たとえば、ROM2にテーブルデータ

50

の形式で予め複数種類のパターンデータを用意しておき、その中から、ユーザの選択に応じてまたは自動的に、いずれか1つを選択する。図58は、変化付与パターンの一例を示す図であり、縦軸は音量係数を示し、横軸は曲の進行方向を示している。同図に示すように、音量係数、すなわち音量データ(エクスペリションデータまたはボリュームデータ)に重み付けを行うための係数は、曲の進行に従って、増大している。しかし、音量データを無限大に増大させることは、音源回路7のハードウェア上の制約と曲への自然な表情付けを損なうため、図58に示すように、音量係数は、曲の進行に従って所定の有限値に収束するような特性を備えることが望ましい。

【0305】

次に、ステップS31で算出された全時間長データと、ステップS32で選択された変化付与パターンとに基づいて、音量変化区間毎の変化値を算出する(ステップS33)。これは、曲の長さは、演奏データに応じて任意に異なるのに対して、変化付与パターンは、所定の固定長を基準にしているため、選択演奏データが決まると、それに従って変化付与パターンの全長を伸縮させ、選択演奏データの全時間長に合わせる(スケーリング)必要があるからである。このようにして、その全長が選択演奏データの全時間長に合わされた変化付与パターンから、各音量変化区間毎(たとえば、小節毎であるが、これに限らず、演奏フレーズ毎でもよい)の変化値を決定する。すなわち本実施の形態では、変化付与パターンはテーブルデータの形式をなしているため、上記スケーリング後のテーブルデータから、各音量変化区間にそれぞれ対応する位置のデータを読み出す。なお、テーブルデータは、上述したように、音量係数値から構成されているので、読み出された各音量係数値を、当該各音量変化区間毎の音量データと乗算すること(重み付けを行うこと)により、変化値に変換する必要がある。

【0306】

最後に、上記算出された音量変化区間毎の変化値のそれぞれを、選択演奏データの対応する位置に挿入する(ステップS34)。具体的には、算出された小節毎の変化値(音量係数値により重み付けされた音量データ)を、選択演奏データ中、当該各小節の先頭位置に記録(挿入)する。

【0307】

なお、選択演奏データが複数のトラックにより構成されているときには、該各トラックそれぞれに対して、同一の変化付与パターンを使用するようにしてもよいし、または、それぞれ異なった変化付与パターンを使用するようにしてもよい。さらに、変化付与パターンをユーザが作成または編集することができるような機能を設けるようにしてもよい。具体的には、変化付与パターンが図58に示すテーブルデータである場合には、倍率を示す数値をユーザが選択すると、その倍率で各データ値が一律に増減し、テーブルデータの全体的な特性は変化しないようにする。このとき、数値に代えて、「強い」、「中程度」、「弱い」、「変化なし」というような言葉を選択するようにし、ユーザがこのいずれかの言葉を選択すると、CPU1がこの選択された言葉を倍率に換算し、その倍率で各データ値を一律に増減させるようにしてもよい。さらに、所定範囲毎に異なった倍率でデータ値を増加減できるようにしてもよい。

【0308】

この発明の一実施例によれば、図57及び図58を用いて説明した「音量パラメータ変化付与処理2」は、音量パラメータだけでなく、音程パラメータの変化を付与するのに適用可能であり、音程について、相応する効果を得ることができる。つまり、演奏データ(MIDIデータ等)の曲の長さを評価し(ステップS31)、変化付与パターン(音程変化テーブル)のパラメータを決定し(ステップS32)、これに基づいて音程変化を算出し(ステップS33)、表情付けを行う(ステップS34)ことによって、曲の進行に従い、音程パラメータの値を徐々に増大させてだんだん音程を高くし、興奮度を高めるといふ表情付けを行うことができる。

【0309】

ここで、変化付与パターンとして、図59に示すように、図58のテーブルと同様に係

10

20

30

40

50

数値が増大する「曲の進行 - 音程係数」テーブル（縦軸を「音程係数」としたもの）を用いるのが好ましい。この場合、例えば、1小節毎に、図59の「曲の進行 - 音程係数」テーブルに従った倍率で音程を高くしていくと良い。このような「曲の進行 - 音程係数」テーブルとしては、音程の目標値を設定してその値を正規化した音程係数変化カーブに乗算して実際の変化カーブを得るように構成するのが望ましい。なお、この音程パラメータ変化付与の効果は、どんな場合にでも有効な機能ではないので、オフにできることが必要である。

【0310】

(C) 演奏音の細かさに応じた処理

図60は、音量パラメータ変化付与処理3の手順を示すフローチャートであり、本音量パラメータ変化付与処理3では、連続演奏の細かさに応じて音量を変化させるという表情付けを行う。

10

【0311】

同図において、まず、選択演奏データ中から音符密度が所定値以上の部分領域、すなわち難関部分を検出する（ステップS41）。音符密度とは、単位時間当たりの音符数を意味する。そして、部分領域は、たとえば、1小節単位、半小節単位、所定の音符長以下の音符が存在する場所近傍のいずれかで検出する。

【0312】

例えば、譜面の一例を示す図3において、音符密度を1拍当たりの音符数とし、所定値を8分音符2個以上とし、部分領域を1小節単位とすると、ステップS41では、同図の第2小節が部分領域として検出される。

20

【0313】

次に、難関部分（検出部分）の音程変化幅を算出する（ステップS42）。音程変化幅とは、音程の最大値と最小値との変化幅を意味する。そして、難関部分の音程変化幅を算出するようにしたのは、演奏時の音量は、難関部分であるかどうかに加えて、音程変化幅の大小によっても、変化するのが自然だからである。

【0314】

次に、難関部分毎に前記音符密度と音程変化幅に応じた変化付与パターンを決定する（ステップS43）。変化付与パターンは、音符密度および音程変化幅の各値に応じた数のパターンがテンプレートとして予め用意されており、ステップS43では、その複数のテンプレートから、前記算出された音符密度および音程変化幅に応じて、1つのテンプレートを選択する。

30

【0315】

なお、これに限らず、たとえば図61に示すように、音符密度と音量（前述した音量係数）との関係を示すテーブルデータを用意し、算出された音量密度に対しては、テーブルデータを検索することにより、対応する音量係数値を求め、この音量係数値を、算出された音程変化幅に応じて修正するようにしてもよい。さらに、テーブルデータも用いずに、所定の演算アルゴリズムによって、演算のみで、音量係数を算出するようにしてもよい。

【0316】

次に、上記決定された変化付与パターンに基づいて音量パラメータを修正する（ステップS44）。ここで、音量パラメータとして、具体的に、ノートイベントに含まれるベロシティを採るようになれば、音量パラメータの修正とは、上記難関部分に属する演奏データ中の各ベロシティを変化付与パターンに基づいて修正することを意味する。そして、変化付与パターンを構成するデータとして、音量係数を採れば、元のベロシティ値に音量係数値を乗算することで、音量パラメータを修正する。なお、音符密度と演奏難易度が共に高い部分では、通常は音量が小さくなる傾向があるが、その傾向を心得た奏者なら、その傾向を補正しようとして、逆に音量が大きくなることもあり得ることを考慮した変化付与が行えるようにしてもよい。（難関部分では必ず音量を下げるとは限らない）。

40

【0317】

この発明の一実施例によれば、図3、図60及び図61を用いて説明した「音量パラメ

50

「一タ変化付与処理 3」の手法は、「音程の不安定さ」を付与する表情付けに適用可能である。すなわち、特に、弦楽器や管楽器などでは、音符が細かいと弾きにくくなるので音程が不安定になるという演奏態様に対応して、音符の密度が細かい部分を検出し（ステップ S 4 1）、音程変化の幅を評価し（ステップ S 4 2）、この評価から音程の不安定さを算出して（ステップ S 4 3）音程パラメータを修正する（ステップ S 4 4）により、演奏の細かさに応じて音程の不安定さを増大させるという表情付けを行うことができる。

【 0 3 1 8 】

例えば、図 3 に示す譜面例において、1 拍当りの音符数は 1 小節目は 1 つであり 2 小節目は 2 つであるが、この数値を係数として音程の不安定さを変化させるのである。このためには、「音符時間間隔 - 音程の不安定さ」テーブルを用意しておき、図 3 の譜面例でい

10

【 0 3 1 9 】

つまり、難しく弾けない結果音程が不安定になることをシミュレートして、実在感を増す表情付けを行うには、図 6 2 のように、ある一定の音符密度以上から音程の不安定さが大きくなるように変化するような「音符時間間隔 - 音程の不安定さ」テーブルを変化付与パターンとして用い、音程の不安定さに関するパラメータについて図 6 0 の処理フローを適用すれば良い。また、このテーブルの閾値をパラメータとすることによって、下手さ

20

【 0 3 2 0 】

図 6 0 の処理フローの適用に当り、ステップ S 4 1 での音符密度の評価は、小節目や拍を単位として、音符数を数え、テンポを評価することによって、単位時間あたりの音符数を数えると良い。また、ステップ S 4 2 での音程変化の幅の評価とは、その単位時間あたりの音符の *key* 単位での音程の最大値と最小値を調べることである。生演奏では、最大値と最小値の幅が広いほど、激しい運動になるので、*key* より細かい単位の音程が不安定気味になる。従って、音符の密度の評価に加えて、音程の幅が広いほど音程が不安定になる度合いを大きくすると自然な表情になる。

【 0 3 2 1 】

なお、鍵盤楽器の場合、ミスタッチはあるものの、基本的に音程が不安定になることはないので、このシステムは、弦楽器（特にフレットの無いタイプ）や管楽器に適應するべきである。

30

【 0 3 2 2 】

また、音符密度が高くなると、例えば、撥弦楽器の場合、はじき損ないが起こりやすくなる。その場合、プリングオフ奏法やハンマリングオン奏法と同じような効果となってしまう。そのような事例に倣うと、音符密度によって、ある確率で自動的にプリングオフやハンマリングオン奏法を挿入するということが、自然な演奏の再現に寄与することになる。そこで、この発明の一実施例によれば、演奏データ（MIDI データ等）から音符密度が高い部分を検出し（ステップ S 4 1）、この部分におけるハンマリングオン（プリングオフ）の頻度を算出し（ステップ S 4 2 に対応）、これに応じて演奏データにハンマリング

40

【 0 3 2 3 】

（D）同一ノ類似フレーズの演奏に伴う処理

図 6 3 は、音量パラメータ変化付与処理 4 の手順を示すフローチャートであり、本音量パラメータ変化付与処理 4 では、類似フレーズが繰り返し出現する選択演奏データの場合に、2 番目以降の類似フレーズの音量パラメータを、その類似度および出現態様に応じて変化させるという表情付けを行う。具体的には、次のアルゴリズムに従って、音量パラメータを変化させる。すなわち、

50

(1) 類似度が高いフレーズが連続して出現する場合には、2番目以降の類似フレーズの音量パラメータの値を、最初の類似フレーズのそれより小さくする

(2) 類似フレーズが連続せずに繰り返し出現する場合には、2番目以降の類似フレーズの音量パラメータの値を、最初に出現する類似フレーズのそれに類似した値であって、その類似度に応じた値に変更する。

【0324】

図63において、まず、選択演奏データをフレーズ単位に分割する(ステップS51)。フレーズ単位とは、本実施の形態では、たとえば1小節長であるが、これに限らず、複数小節長であってもよいし、他の単位長であってもよい。また、フレーズとして、イントロ部、フィルイン部およびメイン部等の、いわゆる演奏フレーズを採用してもよい。

10

【0325】

次に、上記分割された各フレーズに含まれる演奏データ(具体的には、ノートイベントおよびデルタタイム等)を、該各フレーズ間で比較することにより、類似フレーズを検出する(ステップS52)。

【0326】

そして、このようにして検出された類似フレーズ毎に類似度を算出する(ステップS53)。ここで、類似度は、たとえば、比較するフレーズ間で、(1)演奏データがすべて同一、(2)演奏データの一部のみが相違、(3)演奏データの一部のみが同一、(4)演奏データがすべて相違、の4段階の値とするが、この段階は、これより細かくても粗くてもよい。なお、ステップS53では、その前のステップS52で類似フレーズを検出しているため、類似度が上記(4)ということはありません。

20

【0327】

次に、類似フレーズが連続しているか否かを判別し(ステップS54)、類似フレーズが連続して出現しているときには、上記算出された類似度から、変化付与すべきフレーズおよび該フレーズに適用すべき変化付与パターンを決定し(ステップS55)、この決定された変化付与パターンに基づいて、決定されたフレーズに含まれる演奏データの音量パラメータを修正する(ステップS56)。

【0328】

ステップS55において、変化付与すべきフレーズとは、連続して出現する類似フレーズ中、2番目以降に出現するフレーズをいい、変化付与パターンとは、最初に出現するフレーズに含まれる各演奏データの音量パラメータ(具体的には、ノートイベント中のベロシティ)の値に重み付けするための音量係数値(比率)のパターンをいう。そして、音量係数値は、類似度に応じて変更される。たとえば、類似度が上記(1)の場合には、音量係数値を“0.8”とし、類似度が上記(2)の場合には、音量係数値を“0.9”とし、類似度が上記(3)の場合には、音量係数値を“1.0”とするようにである。なお、類似度に加えて、出現する順位も考慮して、音量係数値を決定するようにしてもよい。たとえば、類似度が上記(1)である場合に、2番目に出現する類似フレーズに対しては、音量係数値を“0.8”とし、3番目に出現する類似フレーズに対しては、音量係数値を“0.7”とするようにである。

30

【0329】

ステップS56では、このようにして決定された音量係数を、最初に出現するフレーズ中の各ベロシティ値に乗算し、その各乗算結果で、2番目以降の類似フレーズ中の対応するベロシティ値を置き換えることにより、音量パラメータを修正する。

40

【0330】

一方、ステップS54で、類似フレーズが連続して出現していないときには、フレーズ毎の類似度に基づいた割合で、類似フレーズ間での音量変化を一致させる(ステップS57)。具体的には、類似フレーズ間の類似度が上記(1)の場合には、最初に出現するフレーズに含まれる各演奏データの音量パラメータ(具体的には、ノートイベント中のベロシティ)で、2番目以降に出現するフレーズに含まれる各演奏データの音量パラメータを置き換え、類似フレーズ間の類似度が上記(2)の場合には、演奏データが同一部分に対

50

しては、最初に出現するフレーズ内の対応する部分の音量パラメータで、2番目以降に出現するフレーズ内の対応する部分の音量パラメータを置き換えるとともに、演奏データが相違する部分に対しては、置き換えられた部分に適應するように、その音量パラメータを編集（ペロシティ値の比率を調節）する。

【0331】

この発明の一実施例によれば、図63で説明した「音量パラメータ変化付与処理4」の手法は、そのまま、演奏データの音程パラメータに変化を付与する表情付けに適用可能である。つまり、似たフレーズは似た音程表現をもつことから、図63の処理フローを音程パラメータの修正に適用することによって、同じフレーズが連続して繰り返されるのであれば、別の場所で再現される場合には、最初の音程の表情をそのままコピーするのが良く、また、全く同じフレーズでなく「似ている」という程度であれば、最初の音程の表情と似た音程変化を設定するのが良い。

10

【0332】**(E) 同音符の連続発音に伴う処理**

図64は、音量パラメータ変化付与処理5の手順を示すフローチャートであり、本音量パラメータ変化付与処理5では、3拍子の演奏データの場合に、同一音符長の音符が連続して出現するときには1拍目を強調するという表情付けを行う。

【0333】

同図において、まず、3拍子で小節内がすべて同一音符長である小節を検出する（ステップS61）。同一音符長とは、たとえば4分音符をいう。

20

【0334】

次に、検出された小節毎に先頭位置の音符数を検出する（ステップS62）。先頭位置の音符数とは、先頭の拍に属する音符の数（先頭拍タイミングで同時に発音される音符の数）を意味する。

【0335】

このように同時に発音される音符の数を検出したのは、和音イベントが含まれている演奏データに対して、本音量パラメータ変化付与処理5を単純に適用すると、和音イベントを含む小節は、ステップS61により無条件に検出されるので、1小節中に和音イベントが多く含まれる場合には、単純に音量が大きくなってしまいうため、この場合にも単純に音量が大きくならないように、ステップS62でまず、検出された各小節毎に、和音イベントを構成する音符数を検出し、次のステップS63で、音量変化を補正するという処理を追加するようにしている。

30

【0336】

次に、検出された音符数に基づいて、検出された小節毎の変化値を算出する（ステップS63）。変化値とは、本実施の形態では、音量係数値を示し、この音量係数値は、元のペロシティ値に対する変化の割合を示すものである。なお、本実施の形態では、変化値の算出に、先頭の拍に属する音符の数を考慮しているが、その音符数を考慮しないようにしてもよい。

【0337】

そして、算出された変化値に基づいて、検出されたそれぞれの小節における先頭音符の音量パラメータを修正する（ステップS64）。音量パラメータは、本実施の形態では、ノートイベントのペロシティであり、その修正は、元のペロシティ値を増大させる方向への修正である。なお、先頭音符（ノートイベント）は複数個存在する場合もあり、その場合には、各音符のペロシティ値を同様に修正する。もちろん、各ペロシティ値それぞれに異なる比率で重み付けするための重み付け係数を設定し、増大後のペロシティ値を、この重み付け係数で重み付けして各ペロシティ値として設定するようにしてもよい。

40

【0338】

このように、本音量パラメータ変化付与処理5では、先頭拍の音符のみが強調される。しかし、ウィンナワルツでは、1拍目は標準、2拍目を強く、3拍目を弱く、という表情付けが望ましく、このような表情付けも、上記ステップS62～S64の処理を少し変更

50

するだけで可能になる。また、作風、時代や作曲家に応じて、強弱すべき拍位置を変更するようにしてもよい。

【0339】

さらに、連符（3連符や5連符等）を検出し、その最初の音の音量を大きくすると、連符らしさが表現できる。これも、本音量パラメータ変化付与処理5を少し変更することにより、実現できる。具体的には、ステップS61の処理は行わず、ステップS62で、連符を検出するようにし、ステップS63で、検出された連符の先頭音の変化量を決定し（たとえば、予め設定されている値としたり、現在の音量を所定割合だけ増加させたりする等）、ステップS64で、決定された変化量に基づいて音量パラメータを変更することで、連符の最初の音の音量を大きくすることができる。

10

【0340】

また、たとえば、5連符は2+3のように、簡単な数字の組み合わせに変換し、アクセントを付けるようにしてもよい。

【0341】

一般の人に分かりやすい拍子は、「2拍子」、「3拍子」および「4拍子」であり、5拍子以上の拍子では、それよりも細かい単位に分割した方がイメージをつかみやすい。たとえば、5拍子では、「1, 2, 3, 4, 5」と数えるのではなく、「1, 2, 1, 2, 3」または「1, 2, 3, 1, 2」のように、2拍子+3拍子に分解するとイメージし易いし、6拍子では、「1, 2, 1, 2, 1, 2」または「1, 2, 3, 1, 2, 3」のように、2拍子+2拍子+2拍子または3拍子+3拍子のように分解することが分かりやすい。

20

【0342】

この拍子感を具体的に音に出して表現するために、たとえば、5拍子の場合には、2拍子と3拍子が交互に並んでいるとして、それぞれの1拍目の音量を強調することによって実現することができる。同様に、5連符、6連符、それ以上なども、2連符や3連符に分解して、それぞれの最初の音の音量を強調すると、拍子感が分かりやすくなる。しかし、5拍子はあくまで5拍子であるため、2+3と分解したとしても元々の拍子の先頭の音符と、分割して先頭になった音符とは区別した方がよく、元々の拍の先頭の音符をより強調した方がよい。

【0343】

(F) フレーズの終止感に対応する処理

図65は、音量パラメータ変化付与処理6の手順を示すフローチャートであり、本音量パラメータ変化付与処理6では、フレーズの終止では、音量を抑えるという表情付けを行う。

30

【0344】

同図において、まず、前記ステップS51と同様にして、選択演奏データをフレーズ単位に分割する（ステップS71）。

【0345】

次に、分割されたフレーズ毎に、当該フレーズに設定されているテンポに基づいて終止音量を算出する（ステップS72）。ここで、終止音量は、当該フレーズの終了位置の音符の音量を小さくするものではなく、その音符が発音されてからその音量を徐々に小さくするものである。そして、その発音から消音に至るまでの時間を、当該フレーズに設定されているテンポおよびその音符長に応じて決定している。なお、音符の音量を徐々に小さくする制御を行うためには、その音符が持続音を発生させるものである必要があるが、その音符が減衰音を発生させるものである場合には、その音符の音量を単純に小さくしたり、その音符の前のいくつかの音符の音量を徐々に小さくしていきようにしたりするよう終止音量を算出する。もちろん、その音符が持続音を発生させるものである場合にも、このような終止音量を算出するようにしてもよい。

40

【0346】

そして、前記分割されたフレーズ毎に、当該フレーズの終了位置の音量（具体的には、

50

終了位置のノートイベント中のベロシティ値)を、前記算出された終止音量に基づいて修正する(ステップS73)。

【0347】

なお、本音量パラメータ変化付与処理6では、このように、フレーズの終了位置の音量を抑えるようにしたが、逆にフレーズの先頭位置の音符(ノートイベント)を検出して、その音符の音量(ベロシティ値)を大きくするようにしても、効果的な場合がある。選択演奏データが低音リズム楽器用のものである場合には、特にそうである。

【0348】

(G)ロングトーンのトリル/ビブラートに対する処理

図66は、音量パラメータ変化付与処理7の手順を示すフローチャートであり、本音量パラメータ変化付与処理7では、ロングトーンに対してトリルまたはビブラートが付与されているときには、その音量を、最初大きく、途中小さく、最後大きくなるような表情付けを行う。

【0349】

同図において、まず、トリルまたはビブラート傾向のピッチデータ列が付与されている音符で、音符長が所定長以上のもの(ロングトーン)を検出する(ステップS81)。ここで、所定長とは、たとえば2分音符長をいう。もちろん、これは例示に過ぎず、他の音符長であってもよい。

【0350】

次に、検出された音符毎に、適応する変化付与パターンを決定する(ステップS82)。図67は、この変化付与パターンの一例を示す図であり、縦軸は音量係数を示し、横軸は時間を示している。同図中、実線で示す曲線は音量変化曲線を示し、矢印Aは音量変化の幅を示し、矢印Bは音量変化の全体時間を示し、矢印Cは音量変化の目標値を示し、矢印Dは音量変化が最も落ち込むまでの時間を示している。これらのパラメータの値を、「強い」、「弱い」、「速い」、「遅い」等の言葉で設定するとよい。このような変化付与パターンが、トリル用およびビブラート用それぞれ複数種類(たとえば、音符長毎に)予め用意され、その中から、検出された音符毎に最適な変化付与パターンが自動的に決定される。

【0351】

そして、このようにして決定された変化付与パターンに基づいて、トリルまたはビブラート演奏時の音量パラメータ(ベロシティ)を修正する(ステップS83)。

【0352】

この発明の一実施例によれば、図66及び図67で説明した「音量パラメータ変化付与処理7」の手法は、「音程変化の速さ」を変化させるのに適用可能である。つまり、「音程変化の速さ」の変化付与パターンとして、例えば、図67のパターンに類似した図68に示すような変化曲線パターンを用意しておき、演奏データ(MIDIデータ等)からトリル(ビブラート)且つロングトーンの部分を検出し(ステップS81)、図68のような変化曲線パターンから音程変化を算出する(ステップS82, S83)ことによって、ロングトーンのトリル(ビブラート)について最初はゆっくりした音程変化を与える表情付けを行うことができる。

【0353】

ここで、図68における記号「A」は速さ変化の幅を表わし、記号「B」はトリル(ビブラート)の全体時間を表わし、記号「C」は速さ変化の変曲点までの時間を表わす。変化付与パターンの決定(ステップS82)に際しては、図68の変化曲線パターンの時間や変化量を変更できるパラメータ(A~C等)により、適切な変化を選択できるようにすると良い。また、これらのパラメータは、既に説明したものと同様に、「強い」、「弱い」、「速い」、「遅い」等の言葉で設定することができるように構成するのが良い。

【0354】

なお、トリルやビブラートでは、音程変化の1サイクル毎の音程が不均一である方がより自然になるので、不均一を生む機能も含めると良い。また、ソリストティック(ソロ演奏

10

20

30

40

50

的)なトリルを再現するためには、トリルの上側の音程を高めに変化させると良い。そのために「ソリスティック」と設定することによって、音程変化を高めを設定するようなインターフェースが良い。より自然な表情を自動的に再現するためには、ビブラートの深さや速さは、音程相関、音量相関、duration(発音長)相関を与えると良い。

【0355】

(H)トリル、打楽器のロール演奏に伴う処理

図69は、音量パラメータ変化付与処理8の手順を示すフローチャートであり、本音量パラメータ変化付与処理8では、トリルや打楽器のロール演奏時に自然な表情付けを行う。

【0356】

同図において、まず、選択演奏データ中、トリルまたは打楽器のロール演奏に対応する部分区間を検出する(ステップS91)。

【0357】

次に、検出された区間毎に、当該区間内の個々の音符の音量の変更値を決定する(ステップS92)。ここで、変更値としては、トリルや打楽器のロール演奏時に自然となる値、具体的には、個々の音符の音量(ベロシティ値)が不均一になる値を決定する。このような変更値を決定する方法としては、たとえば、単純にランダム値を発生させて不均一値とする方法、不均一値の流れを記録した変化付与パターン(テンプレート)を予め用意しておき、これに基づいて決定する方法等が挙げられる。

【0358】

(J)複数音同時発音演奏に伴う処理

図70は、音量パラメータ変化付与処理9の手順を示すフローチャートであり、本音量パラメータ変化付与処理9では、和音が含まれている演奏データの場合に、和音に対してその発音をすっきりしたものにするという表情付けを行う。

【0359】

同図において、まず、選択演奏データ中、複数音が同時に発音される部分、すなわち和音部分を検出する(ステップS101)。

【0360】

次に、検出された和音毎に、構成音の重要度に対応させた変化値を決定する(ステップS102)。ここで、変化値は、当該和音の各構成音(ノートイベント)の音量パラメータ(具体的には、各ノートイベント中のベロシティ)の値を変化させる値(たとえば音量係数値)である。そして、本実施の形態では、構成音の重要度に応じて、対応する変化値を異ならせている。これは、和音の各構成音を同じ音量で演奏すると、騒がしいだけになる場合があるので、和音の構成音のうち主要なものだけ、音量を大きめに設定して、すっきりとした和音演奏をしようとしているからである。主要なものであるか否か、すなわち重要度の判断基準は、たとえば、和音の主音を最重要とし、属音を次に重要なものとするという基準を挙げることができる。また、和音毎の重要度の検出は、予め用意された和音重要度テンプレート、具体的には、和音毎に、その和音の各構成音のそれぞれに優先順位(たとえば、優先順位が高いほど重要度が高いとする)を付けた情報を記録したものを用いて行う。変化値の決定方法としては、たとえば、重要度毎に予め決定されている値に決定する方法や、現在の音量値を重要度に応じた割合で増減させる方法が挙げられる。

【0361】

そして、ステップS102で決定された変化値に基づいて音量パラメータを修正する(ステップS103)。

【0362】

なお、本音量パラメータ変化付与処理9では、重要な音符の音量を大きくするようにしたが、逆に、重要でない音符の音量を小さくするようにしてもよい。

【0363】

また、重要度という概念を用いずに、単に、和音の最低音と最高音の音量のみを他の構成音の音量より大きくするようにしてもよいし、和音の根音の音量のみを他の構成音の音

10

20

30

40

50

量より大きくするようにしてもよい。

【0364】

なお、和音ではないが、ギターでよく演奏される分散和音というものがあるが、この分散和音の主音を強調すると、調性感が出てくっきりとした演奏になる。この場合の処理も、本音量パラメータ変化付与処理9の一部処理を変更するだけで、簡単に実現できる。

【0365】

また、オクターブユニゾンで演奏する場合に、すべてのパートを同じ音量で演奏すると、各音色が混ざってしまい、ユニゾンらしさが表現できない場合がある。そのため、選択演奏データ中に、オクターブユニゾンを検出したときに、上のオクターブから順に音量を大きくする、または、小さくする、という音量変化を与えると効果的なことがあり、この音量変化も、本音量パラメータ変化付与処理9の一部処理を変更するだけで、簡単に実現できる。なお、オクターブユニゾンに限らず、同じオクターブのユニゾンに対しても、音色に依存した音量調整を行うようにしてもよいし、3度や5度、その他のユニゾンでも同様な処理を行うようにしてもよい。

10

【0366】

この発明の一実施例によれば、図70により説明した「音量パラメータ変化付与処理9」の手法は、音程についても適用可能であり、和音を自動的に純正調に変えるという表情付けを行うことができる。フレットのない弦楽器や管楽器の音程的な特徴の1つに、純正調の和音を出すことができる点が挙げられるが、これに対応して、先ず、和音で発音されている部分を検索し(ステップS101に相当)、検索した和音の種類を評価し、評価した和音に対応して純正調の場合の音程変化を算出する(ステップS102に相当)。ここで、音程変化算出には、各和音の音程テーブルのようなものを参照することが簡単である。そして、この音程変化に基づいて音程パラメータを修正し音程を変化させて(ステップS103に相当)純正調を発音させる表情付けを行うのである。

20

【0367】

(K) スタッカート演奏に伴う処理

図71は、音量パラメータ変化付与処理10の手順を示すフローチャートであり、本音量パラメータ変化付与処理10では、スタッカートが含まれている演奏データの場合に、生のスタッカート演奏を模倣した表情付けを行う。

【0368】

同図において、まず、選択演奏データ中、スタッカート発音する音符の直後の音符を検出する(ステップS111)。

30

【0369】

次に、検出された音符毎に、変化量を決定する(ステップS112)。生のスタッカート演奏では、スタッカートは必ずしも音の長さを短くするだけでなく、その音を強調をもしている。スタッカートの音を強調する1つの方法としては、その音符の直後にある音符の音量を小さくすることである。ステップS112では、この音量制御を行っている。したがって、ここでの変化量の決定とは、検出された音符(ノートイベント)の音量パラメータ(具体的には、ノートイベント中のベロシティ)の値を減少方向に変化させる量を決定することを意味している。

40

【0370】

そして、ステップS111で検出された音符の音量パラメータを、ステップS112で決定された変化量に基づいて修正する(ステップS113)。

【0371】

なお、一般にスタッカート表現は、その音符の音価やテンポによって変化の度合いが異なるので、スタッカート発音する音符の直後の音符の音量を変化させる度合いも、スタッカート発音する音符の音価やテンポに応じて調整することが好ましい。また、スタッカート直後の音符を検出する際、間に休符相当の時間があり、これを検出対象から除外する場合には、ステップS111で、「スタッカート発音する音符の所定時間内の直後の音符」を検出するようにすればよい。

50

【 0 3 7 2 】

(L) ゆらぎ / ランダム性を伴う処理

図 7 2 は、音量パラメータ変化付与処理 1 1 の手順を示すフローチャートであり、本音量パラメータ変化付与処理 1 1 では、ロングトーンに対して、その音量にゆらぎをつけるという表情付けを行う。

【 0 3 7 3 】

同図において、まず、前記ステップ S 8 1 と同様にして、所定発音長以上の音符 (ロングトーン) を検出する (ステップ S 1 2 1) 。

【 0 3 7 4 】

次に、検出された音符毎に、乱数カウンタと変化付与パターンを用いて、変化量を決定する (ステップ S 1 2 2) 。ロングトーンを生演奏すると、ゆらぎがだんだん増加する傾向がある。ステップ S 1 2 2 では、この現象をシミュレートするための変化量を決定する。すなわち、乱数カウンタから出力される乱数の振幅を、予め用意された変化付与パターン、具体的には、図 7 3 に示すような、乱数カウンタのカウント値に応じて、乱数カウンタが発生する乱数の振幅を変化させるテーブルデータに基づいて変更し、変更後の乱数カウンタから出力された乱数値を上記変化量に決定する。これにより、ロングトーンが発音を開始してからその時間の経過に従って、出力される乱数の振幅が増大するため、ゆらぎをだんだん増加させるような変化量を決定することができる。なお、乱数の振幅は、カウンタ値が増大するに従って際限なく大きくすると、これに応じて決定された変化量は不自然になるので、図 7 3 に示すように、乱数の振幅は、カウンタ値が増大するに従って所定値に収束して行くことが望ましい。

10

20

【 0 3 7 5 】

そして、ステップ S 1 2 1 で検出された音符 (ノートイベント) の音量パラメータ (具体的には、ノートイベント中のベロシティ) の値を、ステップ S 1 2 2 で決定された変化量に基づいて修正する (ステップ S 1 2 3) 。

【 0 3 7 6 】

なお、本音量パラメータ変化付与処理 1 1 では、乱数カウンタを使用したか、これに代えて、ゆらぎテーブルを使用してもよい。さらに、変化付与パターンとして生楽器の演奏データから採取した不均一なパラメータ変化のデータを使用することも効果的である。

【 0 3 7 7 】

選択演奏データ中、音量記号として f (フォルテ) が付加されている区間内に、ロングトーンがある場合に、そのロングトーンを同じ音量で演奏し続けると、だんだん存在感がなくなり、fらしくなくなってくる。したがって、そのような場合には、そのロングトーンの音量を、徐々に大きくすることが効果的である。この音量制御も、本音量パラメータ変化付与処理 1 1 の一部を変更することにより、実現できる。具体的には、ステップ S 1 2 1 の処理の最初に、f 記号の検出処理を追加し、ステップ S 1 2 2 の処理では、乱数カウンタに代えて通常のカウンタを用い、ステップ S 1 2 3 の処理は、そのまま用いるようにする。なお、管楽器の場合、あまり長いロングトーンのときには、その音量がだんだん落ちてくる。この音量制御は、上記 f 記号の検出処理を削除するとともに、上記通常カウンタを減算カウンタに変更して、音量が徐々に下降するように変化させればよい。

30

40

【 0 3 7 8 】

この発明の一実施例によれば、図 7 2 及び図 7 3 により説明した「音量パラメータ変化付与処理 1 1」の手法は、音程についても適用可能であり、ロングトーン (長い音符) に対して音程にゆらぎという表情付けを行うことができる。フレットのない弦楽器や、管楽器などの人間の演奏では、変化のない音程での演奏は不可能であり、或る程度の音程のゆらぎがあるはずである。そこで、このようなゆらぎを実現するために、楽譜データから強弱記号 (f) のあるロングトーンを検出し (ステップ S 1 2 1) 、音程変化を算出して (ステップ S 1 2 2) 音程パラメータを変化させる (ステップ S 1 2 3) ことにより、音量の場合と同様に、「音程」を乱数で変動させるのである。

【 0 3 7 9 】

50

この場合、この乱数には、単純乱数を用いて勝手に音程を変化させるのではなく、乱数をバンドパスフィルタ（BPF）で帯域制限を加えた方が効果が出る場合がある。また、乱数によって、音程を変化させるのではなく、テーブルによって変化させても良い。特に、生楽器の音程変化を記録したテーブルを用いると、その効果は自然なものになる。

【0380】

また、生演奏のロングトーンでは、音長の増大に伴ってだんだん音程のゆらぎが増大する傾向があるので、音量ゆらぎの場合と同様に、新しい音符でリセットされるカウンタの出力により、図73のような特性の乱数テーブルを指示することにより、ロングトーンの最中に徐々に音程ゆらぎが大きくなるシステムを構成するのが好ましい。なお、このような音程ゆらぎは、鍵盤楽器などのように、音程がずれる要素の少ないものでは、この機能を使わない方が自然である。従って、演奏データ（MIDIデータ等）の解析時（ステップS121）に音色を判断して楽器種類を確認するように構成する必要がある。

10

【0381】

さらに、この発明の一実施例によれば、上述した音程のゆらぎ付け処理とは別に、音程のランダム性を増加する表情付けを行うこともできる。例えば、音量記号がf（フォルテ）などの区間でロングトーンがある場合に、変化のない音程で演奏し続けると、だんだん存在感がなくなり、fらしくなくなってくる。従って、このような場合には、徐々に音程の不安定さを大きくすることが効果的である。つまり、ロングトーンでは、だんだん音程のランダム性を増加させる。また、音程のランダム性は、大きさを音色に連動させたり、メロディと伴奏（バック）で変化させたりすることが好ましい。

20

【0382】

このような処理を行うには、まず、強弱記号検出ステップで、強弱記号がf（又はf以上）であるかを検出して、その範囲内にあるロングトーンの音符を検出し、次に、その音符の長さを評価して、音程の不安定さを徐々に大きくするような表情を加える。ここで、音符の長さを評価するのは、音符の時間内に有効に音程の不安定さを変化させたいため、つまり、音程の不安定さを無制限に大きくしないためである。従って、音符の長さに応じて、音程の不安定さの変化のカーブも調整する必要がある。

【0383】

なお、逆に、強弱記号がp（ピアノ）の場合でもこのような不安定現象が起こり得るので、上述した強弱記号検出ステップにおいて、強弱記号が中程度でないものを検出する構成としてこれに対応して音程の不安定さを与えるようにも良い。また、強弱記号は一切無視する方法でも良い。

30

【0384】

また、音色によっては、ランダム性が大きいと不自然なものもあるので、音色によって、ランダム性の大きさを制御する必要がある。特に、ピアノのような鍵盤楽器ではランダム性があると不自然であり、管楽器においても、クラリネットなどは不自然になる。さらに、伴奏パートまで必要以上にランダム性を持たせると、騒々しい伴奏になってしまうので、メロディパートと伴奏パートではランダム性を加減した方が良い。

【0385】

（M）音域に応じたパラメータ処理

40

図74は、音量パラメータ変化付与処理12の手順を示すフローチャートであり、本音量パラメータ変化付与処理12では、音程平均に応じた音量表現による表情付けを行う。

【0386】

同図において、まず、前記ステップS51と同様にして、選択演奏データをフレーズ単位に分割する（ステップS131）。

【0387】

次に、分割されたフレーズ毎に音程の平均値を算出し、この平均値に基づいて変化付与パターンを決定する（ステップS132）。変化付与パターンは、予め音程の平均値毎にテーブルデータの形式で記憶され、ステップS132では、この中から1つを決定する。

【0388】

50

そして、分割されたフレーズ毎に、決定された変化付与パターンに基づいて音量パラメータ（具体的には、エクスペリションデータ）を修正する（ステップS 1 3 3）。

【0389】

この発明の一実施例によれば、図74により説明した「音量パラメータ変化付与処理12」の手法は、音程についても適用可能であり、音域によって音程表現を変えるという表情付けを行うことができる。音程（key）情報によって、音程（keyよりも細かい単位での）係数を決定し、音程係数によって、あるテーブルをひくことによって、音程の変化量を決定する。しかし、1つ1つの音にこのような表情を加えたのでは、従来のキースケールと同じであるが、ある指定した区間（フレーズ、小節或いは拍等）毎に、その音程の平均から処理を決めるということも有効である。

10

【0390】

（N）パート演奏に伴う処理

図75は、音量パラメータ変化付与処理13の手順を示すフローチャートであり、本音量パラメータ変化付与処理13では、パートディビジョン（Part Division）のときには各パートの音量は下げるといった表情付けを行う。

【0391】

この表情付けの処理を説明する前に、この発明の趣旨を説明する。オーケストラスコアにおいて、ストリングスの1つのパートを、さらに2つ以上のパートに分けて演奏する、“Division”という指定がある。第1バイオリンのパートを10人で演奏しているとして、Division記号により、2つのパートに分かれて演奏する場合には、5人ずつに分かれるというのが一般的である。したがって、単純に考えると、Divisionパートの音量は10人分から5人分に半減するはずである。しかし、従来のDTM（desk top music）では、オーケストラスコアでDivision指定があつたとしても、同一音源パートで、ポリフォニックに演奏させる程度の曲作りが一般的であつた。それでは、各音符の音量は一定で、同時に鳴る音符数が倍増するので、音量が増える結果となってしまう。熟練したユーザは、この点に気づき、Division指定があつた場合には、音量を減少させるような設定をマニュアルで行ってきた。本音量パラメータ変化付与処理13は、これを自動的に行おうとするものである。

20

【0392】

図75において、まず、選択演奏データから、同音色のパートおよびそのパート数を検出する（ステップS 1 4 1）。ここで、同音色には、類似音色を含めるようにしてもよい。なお、同音色という条件のみで検出すると、第1バイオリンと第2バイオリンがDivisionとして扱われてしまう場合がある。このような場合には、自動検出でなく、手動でパートを選択するようにすればよい。

30

【0393】

次に、検出されたパート数に基づいて、各パートの演奏音量値を算出する（ステップS 1 4 2）。具体的には、同音色（同Division）で演奏される複数パートのうちの最大音量値を検出し、その最大音量値をパート数で除算して算出された値を各パートの演奏音量値に決定する。そして、演奏音量とは、それぞれのパート演奏の全体音量を調節する値（具体的には、パート毎のボリュームデータまたはエクスペリションデータ）をいう。

40

【0394】

そして、算出された音量値を各パートに記録する（ステップS 1 4 3）。具体的には、算出された音量値を各パートの初期音量として記録する。

【0395】

なお、複数パートが並行して再生されている部分（演奏区間）のそれぞれについて、上記方法で音量制御することが望ましい。その場合には、算出された音量値を対応する演奏区間においてのみ有効にする。

【0396】

（P）音色変化に連動した処理

図76は、音量パラメータ変化付与処理14の手順を示すフローチャートであり、本音

50

量パラメータ変化付与処理 14 では、音色変化に応じた音量表現による表情付けを行う。

【0397】

同図において、まず、音色の変更位置と変更内容を検出する（ステップ S151）。すなわち、選択演奏データ中の解析対象パート内での、音色変更がある位置と、その音色変更の内容（変化前の音色と変化後の音色）を、変更位置毎に検出する。

【0398】

次に、変化内容に対応した音量の変化量を決定する（ステップ S152）。具体的には、音色毎に最適な音量値が、たとえばテーブルデータの形式で予め決定されており、変更前の音色と変更後の音色のそれぞれの音量値をテーブルデータから読み出し、その音量に基づいて、音色変更位置において、音量が滑らかに変更されるような音量パラメータ（エクスプレッションデータ）を算出して決定する。

10

【0399】

そして、ステップ S152 で決定された変化量に基づいて、音量パラメータを修正する（ステップ S153）。具体的には、上記算出された音量パラメータを変更位置毎に記録する。

【0400】

このようにして、本音量パラメータ変化付与処理 14 では、特定の音色による演奏を目立たせることができる。

【0401】

（Q）ピッチベンド変化に伴う処理

20

図 77 は、音量パラメータ変化付与処理 15 の手順を示すフローチャートであり、本音量パラメータ変化付与処理 15 では、ピッチベンド時には音量が変化するという表情付けを行う。

【0402】

同図において、まず、ピッチベンドによるピッチの変更位置を検出する（ステップ S161）。具体的には、選択演奏データ中の解析対象パート内での、ピッチ変更がある位置と、そのピッチ変更の内容（ピッチベンドデータ）を、変更位置毎に検出する。

【0403】

次に、検出されたピッチの変更位置毎に、ピッチベンドの変化傾向を算出し、その算出結果に応じて、音量の変化量を決定する（ステップ S162）。ここで、ピッチベンドの変化傾向は、検出されたピッチベンドデータに対して、ハイパスフィルタ（HPF）処理を施すことにより、検出する。必要に応じて、この処理結果にローパスフィルタ（LPF）処理を施し、その変化を滑らかにする。そして、フィルタ処理が施されたピッチベンドデータの絶対値を用いて、音量パラメータ（ピッチの変更位置におけるノートイベントのベロシティ）の値の変化量（減少率）を算出する。すなわち、ピッチベンドの変化率が大きいほど、音量を大きく下げるような変化量を算出して決定する。これにより、自然なピッチベンド感を表現できる。

30

【0404】

そして、ステップ S162 で決定された変化量に応じて音量パラメータを修正する（ステップ S163）。なお、修正する音量パラメータは、具体的には上述のように、ピッチの変更位置におけるノートイベントのベロシティであるが、これに限らず、ステップ S161 で検出対象となる演奏データが、弦楽器における 1 弦の演奏に対応するものである場合には、音量パラメータとして、エクスプレッションデータを使用してもよい。

40

【0405】

（R）パートに応じた処理

図 78 は、音量パラメータ変化付与処理 16 の手順を示すフローチャートであり、本音量パラメータ変化付与処理 16 では、マルチパートからなる演奏データの場合に、そのメロディラインを浮き立たせるという表情付けを行う。

【0406】

同図において、まず、メロディパートを検出する（ステップ S171）。

50

【0407】

次に、メロディパートの音量を他のパート（伴奏パート）の音量と比較して、メロディパートの音量が相対的に他のパートの音量より上がるように、変化量を決定する（ステップS172）。

【0408】

そして、決定された変化量に基づいてメロディパートの音量パラメータ（エクスプレッションデータ）を修正する（ステップS173）。

【0409】

なお、メロディパートの音量を上げようとしても、前記音源回路7の出力可能範囲を超えてしまっても上げることができない場合には、他のパートの音量を一律に下げることにより、相対的にメロディパートの音量を上げるようにすればよい。

10

【0410】

また、ピアノのように、1つの音色でメロディから伴奏まで担当するような演奏データに対して、そのメロディラインを浮き立たせるためには、演奏データに含まれるトップノートを抽出して、メロディパートとし、残りを伴奏パートとして、メロディラインと伴奏とを分離し、その後、本音量パラメータ変化付与処理16のステップS172およびS173の処理を行えばよい。

【0411】

この発明の一実施例によれば、図78により説明した「音量パラメータ変化付与処理16」の手法は、音程についても適用可能であり、メロディパートや伴奏パートに応じて基本ピッチを変えるという表情付けを行うことができる。つまり、マルチパートの中でメロディを浮き立たせる1つの方法として、メロディパートの遠近感を近づけ伴奏パートの遠近感を遠ざける方法があるが、演奏データ（MIDIデータ等）からメロディパートを検出し（ステップS171）ピッチ変化を算出し（ステップS172）てピッチ変化を付与する（ステップS173）ことにより、ピッチの変化でこの方法を実現するのである。

20

【0412】

ここで、ピッチ変化をPitchBendEventで実現する場合には、メロディパートのピッチを上げようとしても、PitchBendの最大値を超えてしまうと、上げることができない。その場合には、その他全てのパートのPitchBendを一律に下げることによって、相対的にメロディのピッチを上げるように構成することが好ましい。

30

【0413】

また、ピアノの楽譜のように、1つの音色でメロディから伴奏まで担当するような楽譜については、楽譜が入力されると、そのトップノートを抽出してメロディパートとし、残りを伴奏パートというように分離した後、メロディパートのピッチをわずかに上げてメロディを強調して演奏させると、メロディパートが浮き出た自動編曲演奏装置を構成することができる。

【0414】

さらに、この発明の一実施例によれば、図78により説明した「音量パラメータ変化付与処理16」の手法は、音量や音程だけでなく、他の演奏パラメータについても適用可能であり、一般的に、メロディパートや伴奏パートに応じて演奏パラメータを変えるという表情付けを行うことができる。つまり、マルチパートの中でメロディを浮き立たせる1つの方法として、メロディパートの遠近感を近づけ伴奏パートの遠近感を遠ざける方法に対して、演奏データ（MIDIデータ等）からメロディパートを検出し（ステップS171）遠近感を算出し（ステップS172）、これに応じて演奏パラメータの変化を付与する（ステップS173）ことにより、これを実現することができる。

40

【0415】

ここで、遠近感の設定方法には音量や音程以外にいろいろ考えられるが、例えば、リバーブの深さを変更する方法がある。リバーブが深いほど遠くに感じる。また、LPF、H

50

P Fが設定できるなら、このようなフィルタを用いて低域を強調したすれば近い感じが得られ、逆に、低域を弱くすると遠い感じになる。これは、LPFやHPFでなく、イコライザ（等価器）を用いても同様に実現することができる。また、リバーブなどと組み合わせると、より一層効果が大きくなる。

【0416】

（S）フィンガリングに応じたパラメータ処理

図79は、音量パラメータ変化付与処理17の手順を示すフローチャートであり、本音量パラメータ変化付与処理17では、フィンガリングに応じた音量変化により生々しい演奏を再現するという表情付けを行う。

【0417】

同図において、まず、選択演奏データ中、演奏操作が困難と考えられる音高を検出する（ステップS181）。演奏操作が困難と考えられる音高とは、具体的には、たとえばピアノ演奏やギター演奏における小指で演奏する音高や、ピアノ演奏における特に速いアルペジオに相当する音高を言う。もちろん、他の判定基準を用いてもよい。

【0418】

次に、検出された音高の音量が他の音高の音量よりも相対的に小さくなるような変化値を決定する（ステップS182）。

【0419】

そして、決定された変化値に応じて音量パラメータ（具体的には、検出された音高に対応するノートイベント中のベロシティ）を修正する（ステップS183）。

【0420】

この発明の一実施例においては、この音量パラメータ変化付与処理17に対し、音色について、フィンガリングを考え指に応じて音程のランダム性を変える表情付けを行うことができる。例えば、フレットのない弦楽器の生演奏に倣うと、フィンガリングに応じて、高くなりがちな音と低くなりがちな音があり、特に、速いフィンガリングの時にはこのようなことが起こりやすい。従って、このようにフィンガリングを考えた音程の変化を与えることによって、より生々しい演奏を再現することができる。また、木管楽器でも、押さえる穴の組み合わせによって、高くなりがちな音と低くなりがちな音がある。金管楽器でも管の長さのポジションによって、同様であるので、このような表情付けは、弦楽器に限られたことではない。

【0421】

このようなフィンガリングを考えて音程のランダム性を変えるには、演奏データ（MIDIデータ等）からフィンガリングを判断し（ステップS181に相当）、このフィンガリングに応じた音程変化を算出し（ステップS182に相当）、算出された音程変化に基づいた音程パラメータを付与することによって、フィンガリングに応じて音程を自動変化させる処理を行う。

【0422】

例えば、チェロの場合、図80のような楽譜に対して、第1ポジションといわれるフィンガリングを行うとしたときは、「ミ」 1（人差し指）、「ファ」 2（中指）、「ソ」 4（小指）という具合のフィンガリングになる。この場合、人間の指の構造上、人差し指、中指、薬指及び小指は等間隔に開かず、中指と薬指との間が他より狭くなる。その結果、図80の楽譜では、「ファ」の音程が高めにずれやすい傾向を待つ。熟練した演奏者はこのような現象を知っているので、訓練によって補正することが可能であるが、熟練度によって、この音程のずれが顕著になる。逆に、薬指の場合には、音程が下がる傾向があるので、そのような傾向を登録することによって、奏者の下手さを再現することができる。

【0423】

この処理は下手さの再現であるので、あまり度合いが強いと好ましい結果は得られない。しかし、適度な下手さは、自然な演奏形態として表現されるので、この下手さ加減をパラメータによって調整することができるように構成するのが良い。

10

20

30

40

50

【0424】

また、フィンガリングのポジションが移動する場合には、ポジション移動に応じた音程変化の音加わることがあるので、フィンガリングを判断した後、このフィンガリングのポジション移動の大きさを評価してポルタメント（又はスライド）を付加し（ステップS182に相当）、これに基づいた音程パラメータを付与することによって、ポジション移動の評価に応じて音程を自動変化させる処理を行うのがよい。このようにポジション移動を評価することによってポジション移動量に応じた音程変化を発生させる際、例えば、フレットのない弦楽器の場合には、ポルタメント的に連続的な音程変化を加えるのが好ましく、また、フレットのある弦楽器の場合には、フレットの音程に対応した階段的变化の音程変化を加えるのが好ましい。

10

【0425】

なお、フィンガリング判断によって開放弦と評価された場合には、ビブラートを禁止すると、より自然な表情を再現することができる。また、管楽器の場合、フィンガリングでレジスターチューブの開閉をまたぐ際には、滑らかなスラー演奏が難しくなる。そこで、滑らかなスラーの許可、不許可も、フィンガリングの評価によって行うと、実感的な演奏を再現することができる。

【0426】

さらに、この発明の一実施例によれば、音量パラメータ変化付与処理17の手法を応用して、フィンガリングを考え、低いポジションで大きなベロシティ（Velocity）のときに、「びびり」（ノイズ音）や「チョッパー」を加える表情付けを行うことができる。例えば、第1ステップで演奏データ（MIDIデータ等）からフィンガリングを判断し（ステップS181に相当）、第2ステップでは、低いポジションで大きなベロシティを検出してこれに応じたノイズを算出し（ステップS182に相当）、第3ステップにおいて、これを演奏データに付加する（ステップS183に相当）ことにより、フィンガリングに応じてノイズを自動的に付加するシステムを構成することができる。

20

【0427】

弦楽器の生演奏に倣うと、フレットの有無に関わらず、低いポジションで強くはじくと弦が指板に当たる音が出やすくなる。そのようなフィンガリングを考えたノイズ付加を行うことによって、より生々しい演奏を再現することができる。また、木管楽器でも、押さえる穴の組み合わせによって、高くなりがちな音と低くなりがちな音があり、金管楽器でも、管の長さのポジションによって、同様であって、このような演奏形態は、弦楽器に限られたことではない。

30

【0428】

また、音色がベースギターの場合、ノイズを加える代わりに、音色をチョッパーベースに切り替えるという方法も効果的である。この場合には、第2ステップ（ステップS182に相当）で低いポジション且つ大きなベロシティの検出に応じて音色を一時的に変更するフィンガリング応答自動音色変更システムを構成すればよい。なお、フィンガリング判断で開放弦と評価された場合には、開放弦用の音色に変更することも効果的である。

【0429】

（T）弓の返しに伴う処理

40

図81は、音量パラメータ変化付与処理18の手順を示すフローチャートであり、本音量パラメータ変化付与処理18では、弓を用いて演奏する弦楽器の演奏データの場合に、弦楽器の弓の返しに応じた音量変化により運弓感を再現するという表情付けを行う。このように、本音量パラメータ変化付与処理18では、弓の返しに応じた音量変化を付与するため、選択演奏データが弓を用いて演奏する弦楽器用のものでなければならない。したがって、本音量パラメータ変化付与処理に入る前に、選択演奏データがこの種のものであるか否かをチェックし、この種のものであるときにのみ、以下の処理に移行するようにする。

【0430】

同図において、まず、元の音量値（ベロシティ値）から弓の移動速度を算出し、その値

50

を用いて弓の折り返し位置を検出する(ステップS191)。具体的には、算出した移動速度に基づいて弓の軌跡を算出し、この軌跡と弓の長さを示すパラメータとを比較することにより、弓の折り返し位置を検出し、この時刻を求める。なお、実際には、弓の圧力も考慮する必要があるが、本実施の形態では、単純化のため省略している。もちろん、この弓の圧力を考慮するようにしてもよい。

【0431】

次に、検出位置毎に、弓の折り返し位置近傍の演奏データを読み出し、変化付与パターンを用いて変化量を決定する(ステップS192)。図82は、この変化付与パターン(テーブルデータ)の一例を示す図であり、縦軸は音量変化値を示し、横軸は時間を示している。同図に示すように、音量変化値を示す曲線は、弓の折り返し位置(図中央位置)でその音量変化値が最小となっている。なお、生演奏の場合には、弓返しを気づかせない方が、よりよい表情付けになる場合もあるので、図82のテーブルデータを用いる際には、その縦軸をスケールリングして、表現の効果を調整することが望ましい。

10

【0432】

そして、決定された変化量に応じて音量パラメータ(ベロシティ)の値を修正する(ステップS193)。

【0433】

なお、単純に音量から弓の速さを算出したのでは、好ましくない時間に弓の折り返しが発生する場合があるので、弓の速さは音量とは無関係に調整できるようにした方が好ましい。

20

【0434】

(U) 対応歌詞情報に伴うパラメータ処理

図83は、音量パラメータ変化付与処理19の手順を示すフローチャートであり、本音量パラメータ変化付与処理19では、歌詞を伴う演奏データの場合に、歌詞の内容に応じて抑揚を変更するという表情付けを行う。

【0435】

同図において、まず、解析演奏データ(選択演奏データ)内に歌詞情報があるか否かを判別し(ステップS201)、歌詞情報がないときには、本音量パラメータ変化付与処理19を直ちに終了する一方、歌詞情報があるときには次のステップS202に進む。

【0436】

ステップS202では、歌詞情報内の単語から、音量変化を付与すべき単語を検出する。この検出方法としては、たとえば、音量変化を付与することが好ましい単語のリストが予め作成され、そのリストと歌詞情報内の単語とを照らし合わせることにより、音量変化を付与すべき単語を検出する方法が挙げられる。

30

【0437】

次に、検出された単語毎に使用する音量変化パターンを読み出して、音量の変化値を決定する(ステップS203)。具体的には、上記リスト内の単語毎に、どのような音量変化を付与すべきかを記載した音量変化パターン(テーブル)が用意され、ステップS202では、この音量変化パターンを読み出して、音量の変化値を決定する。音量変化パターンには、たとえば、単語が発音されている間の音量変化曲線が記録され、この曲線の時間軸(縦軸が音量変化値を表わし、横軸が時間を表わすものとする)を伸縮することにより、音量の変化値を決定する。なお、音量の変化値の決定には、単語位置に対応した音符の元の音量も考慮することが望ましい。

40

【0438】

そして、決定された変化値に応じて音量パラメータ(エクスペッションデータ)を修正する(ステップS204)。

【0439】

この発明の一実施例によれば、図83で説明した音量パラメータ変化付与処理19の手法は、音程変化の付与するのに適用し、作曲システムにおいて、詩を入力すると、単語の抑揚データ、音節データから、微妙に音程を変化させるように構成することができる。歌

50

を伴う曲の場合、同じメロディーでも、歌詞の内容によって抑揚を変えた方が自然な場合があり、抑揚をピッチの微妙な変化によって実現することも良い方法である。そこで、ある単語を音節単位に音程係数とともに登録しておき、演奏データの中にその単語が出現すると、音程を（微妙に）変化させるように構成するのである。ここで、単語に従った音程変化とは、単語単位で音程を変更するのではなく、単語の音節に対応する音符の時間内での音程変化を意味している。

【0440】

(V) 種々の実施の形態

なお、上記音量パラメータ変化付与処理1～19の処理によって、選択演奏データに各種音量変化を伴う表現処理を施すと、選択演奏データ全体に亘って非常に大きな音量が付与されていたり、逆に、非常に小さな音量が付与されていたりする場合があり得る。これを防止するためには、選択演奏データ全体の音量の平均値を算出し、この平均値が所望の値になるように、選択演奏データ全体の音量に対してオフセットを加えるようにすればよい。そのときに、ある一部の音量が音源回路7が出力可能な最大値や最小値を超えることがあり得るので、選択演奏データ全体の音量の平均値を算出するときには、選択演奏データ全体を通しての最大値と最小値を求めておくことよい。そして、オフセットを加える場合には、最大値にオフセットを加えた結果が音源回路7の最大値を超えなければ問題ないし、超えるときには、超えない最大の値をオフセット値として全体に加えるという処理を行えばよい。また、最大値がある一瞬だけしか音源回路7の最大値を超えないときには、その値だけを音源回路7の最大値でクリップする方が好ましい場合もあるので、最大値を超えている時間が、選択演奏データ全体に亘って再生したときの時間の何%まで許すかという許容時間（閾値）を予め決めておくようにしてもよい。最小値についても同様である。

【0441】

また、選択演奏データに想定されている楽器の種類に応じて、選択演奏データに適用可能な表現付けの種類を規定するようにしてもよい。たとえば、ピアノの音色では、一旦発音した後に、その音符をクレッシェンドすることは非現実的である。もっとも非現実的な効果を狙うのであれば、それでもよいが、ピアノ音をシミュレートする場合には、ある音符に対して、その発音後に音量変化させることは禁止した方が好ましい。そのためには、音色毎に音量制御用のフラグを設けるようにすればよい。このフラグにより、たとえば、上記減衰音の場合には、発音後の音量制御を禁止し、持続音の場合には、発音後の音量制御を許可するというように、音量制御を行うか否かを決定する。図84は、発音後の音量制御を規定するためのテーブルデータの一例を示す図であり、上記音量制御用のフラグには、このテーブルデータから読み出された値が設定される。さらに、このテーブルデータを用いて、発音後に音量制御が不許可とされる音色に対して、そのようなデータが既に設定されている場合には、当該ノートオンからノートオフまでの範囲内に含まれる音量制御データをすべて消去する、という掃除機能を設け、不自然さを自動的に解消するようにしてもよい。

【0442】

さらに、本実施の形態（音量パラメータ変化付与処理1～19）では、この発明を音量パラメータの制御にのみ適用するようにしているが、これに限らず、この発明は、音量パラメータ以外のパラメータ、たとえば、ピッチ、効果（リバーブ、コーラス、パンニング等）、音色等に関するパラメータの制御に対しても有効である。

【0443】

また、オーケストラスコアを演奏データにして、音源回路7で再生するような場合に、パート数が多数になると、個々のパートに個別な音量制御を行うことがあまり有効でない場合もある。そのような場合、従来は、1つのパート用に作成した音量制御用データ（特に、エクスプレッションデータやピッチベンドデータ）をコピーして、他のパートに適用させていた。しかし、それでは同じデータが多数複製されるという無駄が生じるので、複数のパートをグループ化して、1つの音量制御用データで複数のパートを制御する仕組みとした方がよい。

10

20

30

40

50

【 0 4 4 4 】

〔実施態様 3〕

この発明による更に別の実施例 (a) ~ (e) を、「実施態様 3」として、以下に説明する。

【 0 4 4 5 】

(a) 「アクセントの音は微妙に音をうわずらせる」：

アクセントを表現するためには、一般には、音量を強調することを主体にしているが、生の演奏表現に倣って更に音程の変化を伴うような表情付けを行うと、より自然な表情を実現することができる。また、アクセントに限らず、一般に音量を変化させると言われる演奏表現では、実際には、音程の変化を伴っていることが多い。そのために、演奏記号に音量変化の処理を対応させるとともに、同時に音程変化の処理も対応させることが望ましい。そこで、この発明の一実施例においては、音符データからアクセントを検出すると、音量及び音程の時間変化を算出し、これらの変化に基づいて表情付けを行うようにしている。

10

【 0 4 4 6 】

図 8 5 は、この発明の一実施例によるアクセント音について微妙に音をうわずらせる処理の例を示すフローチャートである。まず、ステップ K k 1 において、選択した演奏データ中から音量変化を付与したい区間を指定し、続くステップ K k 2 では、指定された区間から音量変化 (アクセント) が付与されている部分を検出し、例えば、図 8 6 に示すように、この部分の音量及び音程の時間変化を算出する。なお、このように区間を指定しないで、演奏データ全般を解析区間としてもよい。

20

【 0 4 4 7 】

さらに、ステップ K k 3 では、検出した部分毎に、音量変化に相当する音程変化傾向を決定する。この音程変化傾向の決定は、音量変化傾向を用いて演算により決定する方法や、音量変化傾向に相当する予め用意された音程変化の変化付与パターンを読み出して決定する方法などがある。そして、ステップ K k 4 において、決定された内容に基づいて検出部分の音程パラメータを変更し、ステップ K k 5 に進んで、終了操作があったと判断されるとこの処理を終了し、そうでなければ、ステップ K k 2 に戻り、ステップ K k 2 ~ K k 5 の処理を繰り返す。

【 0 4 4 8 】

なお、このようなアクセント音に対する処理は、鍵盤楽器などのように、音程がずれる要素の少ないものでは、この機能を使わない方が自然である。また、アクセントに限らず、拍子の頭の音符や、連符の頭の音符に対して、自動的にアクセント的な処理を施すことも有効である。さらに、5 連符、7 連符のように大きな数の連符に対しては、2 連符 + 3 連符とか、2 連符 + 2 連符 + 3 連符などのように小さい数の連符に分解して、本来の頭の音符には普通のアクセント処理を施し、分解された頭の音符にはやや弱いアクセント処理を施すと、拍感が出やすくなるので有効である。

30

【 0 4 4 9 】

音程を変化させる際に、音量も同時に変化させるということに関して、ビブラートやポルタメントの際、音程に動きがあるときには、音量をやや下げると、より自然な表現が可能になる。これを実現するには、音符データからビブラートなど音程変化を行う表現記号を検索する処理を行い、検索された表現を伴う音符の音程変化の大きさを算出し、これに基づいて音量を下げるようにすればよい。ここで、この音程変化の算出には、H P F による抽出を用いるのが好ましいが、H P F の出力の変化が激しすぎる場合には、H P F の後段に L P F を配置することが望ましい。

40

【 0 4 5 0 】

(b) 「ダブルチョーキングでは、音程変化を平行にしない / 勝手にチャンネルを分ける」：

ギターの奏法において、2 本の弦を同時に発音し而も同時にチョーキング奏法を行った場合には、2 つの音程間隔が平行になるように変化させることは、生楽器では不可能であ

50

る。この発明の一実施例においては、このような事実に倣って、ダブルチョーキングの際には、わざと音程変化のタイミングをずらすことによって、自然な演奏表現を再現させることができる。例えば、ダブルチョーキングがあった場合、図 87 に示すように、ダブルチョーキングの上音と下音で、音量の時間変化タイミングをずらすのである。このように「タイミングをずらす方法」は、同じ時間変化ではあるが変化開始の時刻をずらすようにしても良いし、図 87 のように時間変化の形状そのものが異なっても良い。

【0451】

図 88 は、この発明の一実施例によるダブルチョーキングでは平行にしない処理の例を示すフローチャートである。この処理がスタートすると、ステップ L11 において、選択された演奏データ中からダブルチョーキング部を検出する。ここで、ダブルチョーキング部が複数検出された場合には、検出された各部毎にステップ L12 以下の表情付与処理を行う。また、検出された複数部から、表情付与処理行いたい部分を選択できるようにしてもよい。

10

【0452】

次のステップ L12 においては、まず、ダブルチョーキングにおける上下の二音に異なる音程（又は音量）変化を付与するために、上下各音を別々のパートに分離して記憶し、次に、各パートに音程（音量）変化傾向を決定する。この変化傾向については、図 87 に示すように時間変化タイミングがずらされるが、図 87 のような変化傾向を、予め変化傾向パターンとして記憶しておいてもよいし、演算で求めるようにしてもよい。このようにして、変化傾向パターンが決定されると、ステップ L13 にて、決定された変化傾向に基づいてダブルチョーキング部の音程（音量）パラメータを変更し、この処理を終了する。

20

【0453】

なお、「弦の性質」というパラメータを設定し、このパラメータをチョーキングのカーブに対応づけると、実感的になる。また、目標となる音程自体を本来の音程差からずらしておいた方が、より自然な感じになる。つまり、図 87 の例においては、上音と下音との音程間隔は 5 度であるが、音程が変化している最中は 5 度からずれることになるが、音程変化が終了した時点でも 5 度からずれた状態にしておくこと、より自然な感じが得られる。

【0454】

また、上述した「タイミングをずらす方法」を自動的に実現するためには、2本の弦に対応する音程変化を独立に行う必要があるため、2つの M I D I トラックに自動的に分離させる必要がある。このため、この発明の一実施例においてはダブルチョーキングに対して、勝手にチャンネルを分けるように構成している。つまり、演奏データ（M I D I データ等）からダブルチョーキングを検出すると、表情付けがなされた複数パートの演奏データを生成させ、複数ダブルチョーキングを複数パートに自動的に割り振るのである（ステップ L12 参照）。

30

【0455】

（c）「連続したチョーキングでは、音程変化を不均一にする」：

例えば、図 89 に示す連続したチョーキングの譜例では、本来は、チョーキングが 4 回繰り返しており、全て同じ音程変化を行うべきであるが、実際の演奏に倣うと、必ずしも同じ音程変化になっているわけではなく、むしろ、毎回微妙に音程が異なる場合が多く、その方が自然な演奏に聞こえる。そこで、この発明の一実施例においては、このような事実に倣って、チョーキングの指定が連続して現れる部分を検索し、毎回のチョーキングの音程変化が同じにならないように、音程に不均一性を加えることによって、自然な演奏表現を再現させることができる。このような表現付けは、図 89 の例では、チョーキングのかかった真ん中の音「レ」の音程から微妙に高くなったり、低くなったり変化を与えることによって実現される。

40

【0456】

図 90 は、この発明の一実施例による連続チョーキングでは音程変化を不均一にする処理の例を示すフローチャートである。この処理では、まず、ステップ Mm1 において、選択された演奏データ中から連続してチョーキングが行われている部分を検出する。ここで

50

、連続チョーキング部が複数検出された場合には、検出された各部毎にステップM m 2以下の表情付与処理を行う。また、検出された複数部から、表情付与処理行いたい部分を選択できるようにしてもよい。

【0457】

次のステップM m 2では、チョーキングが連続している回数や音程の変化傾向に基づいて、変化付与テンプレートを選択したり、所定の演算を行ったりすることで、付与する音程の変化傾向を決定する。そして、ステップM m 3において、決定された変化傾向に基づいて、検出されたチョーキング連続部の音程パラメータを変更し、この処理を終了する。

【0458】

(d) 「アルペジオでは、倍音を別チャンネルで鳴らす」：

10

例えば、図91(1)に示すアルペジオの譜例に示されるような音形のアルペジオを演奏したとすると、生楽器の場合には、各音の共通の倍音である図91(2)の「ソ」の音がずっと響くという現象が起こる。そこで、この発明の一実施例では、演奏データ(MIDIデータ等)からアルペジオの共通倍音を検出し、この倍音の音を別の音源パートに割り振って別パートにて小さな音量で鳴らすことにより、このように倍音が鳴り続けている現象をシミュレートすることができる。

【0459】

図92は、この発明の一実施例によるアルペジオでは倍音を別チャンネルで鳴らす処理の例を示すフローチャートである。この処理では、まず、ステップN n 1において、選択された演奏データ中からアルペジオ部を検出する。ここで、アルペジオ部が複数検出された場合には、検出された各部毎にステップN n 2以下の表情付与処理を行う。また、検出された複数部から、表情付与処理行いたい部分を選択できるようにしてもよい。ステップN n 1に続くステップN n 2で各音の共通の倍音を抽出した後、ステップN n 3にて、抽出した倍音を選択演奏データの新規パートで発音させるよう記憶し、この処理を終了する。

20

【0460】

(e) 「楽譜のシンボルによって音色を選択する」：

楽譜のシンボルと音色との対応を考察すると、例えば、バイオリン等の擦楽器の楽譜において、図93に示すarco(弓で弦を擦るアルコ奏法)とpizz.(pizzicato:弦をはじくピチカート奏法)の切り替え譜例の場合、第1小節は運弓(擦弦)による演奏が指示され、第2小節は撥弦(弦をはじくこと)による演奏(ピチカートストリング)が指示され、そして、第1小節は再び運弓(擦弦)による演奏が指示されていることになる。ここで、楽譜上に「pizz.」と表示されていると、自動的に音色をピチカートストリングに変更すると便利であり、また、「arco」と表示されておれば、再び、擦弦音色に戻すようにすると便利である。例えば、現状のGMシステム(MIDI音源標準化規格)の音色では、ピチカート音色は1つしかなく、擦弦音色は多くの種類が用意されている。従って、ピチカートストリングに変更したときには、「arco」が表示されると、どの音色に復帰するかを記憶する手段を用意しておくべきである。

30

【0461】

そこで、この発明の一実施例によると、演奏データ(MIDIデータ等)から、ピチカート奏法を示すデータ(「pizz.」記号)を検索し、このデータが検出されると現状の音色を保持した上ピチカート用の音色を設定して表情付けを行うことにより、ピチカート記号に応じて自動的に音色を変更するように構成することができる。なお、GMシステムのようにピチカート音色が1に限るといことがない場合には、ピチカート記号に対応する音色を別途登録しておくことが好ましい。

40

【0462】

また、このような楽譜シンボルと音色の対応関係は、上述の「pizz.」に限った話ではなく、ベース音色を一時的にチョッパーベース音色に変更する場合にも、この仕組みをそのまま使うことができる。その他、ストリングスパートを一時的にソロ音色にする場合もあるし、コルレーニョなど特殊奏法の音色にする場合など、様々の態様がある。さら

50

に、似た機能として、ピアノのペダル記号をコントロールチェンジのダンパオンに対応して自動設定することなども有効である。従って、この発明の一実施例では、楽譜のシンボルによってこれに対応する所定の音色を選択するシステムが提供される。

【0463】

図94は、この発明の一実施例による楽譜のシンボルにตอบสนองして音色を選択する処理の例を示すフローチャートである。この処理がスタートすると、まず、ステップPp1において、選択された演奏データ中から音色の変更を指示する所定の音楽記号(それに相当するデータ)Aを検出する。ここで、複数部が検出された場合には、検出された各部毎にステップPp2以下の表情付与処理を行う。また、検出された複数部から、表情付与処理を行いたい部分を選択できるようにしてもよい。

10

【0464】

次のステップPp2では、ステップPp1で検出された音色の変更指示を行う音楽記号のそれぞれに対応して、音色を基に戻すための音楽記号Bを検出する。そして、続くステップPp3において、ステップPp1, Pp2で検出された音楽記号A, Bの位置にあわせて、対応する音色の変更データを挿入する。例えば、変更を指示する音楽記号Aの位置から、それに対応して音色を復帰させる音楽記号Bの位置までの間には、変更後の音色を表わす音色変更イベント(プログラムチェンジデータおよびバンクセレクトデータ)を挿入し、音楽記号Bの位置以降は復帰後の音色を挿入する。ここで、変更後の音色は、音楽記号毎に予め決められているものを使用し、復帰後の音色については、変更前の音色を保持しておき、それを挿入する。

20

【0465】

(f)「ピアノではサスティンペダルの場所を勝手に計算する」:

ピアノ楽譜において、ペダル記号が付いているものであれば、その記号の位置にてコントロールチェンジによるダンパー制御を行えばよいが、ペダル記号が省略されている楽譜データもある。そこで、この発明の一実施例によれば、このような楽譜に対応するため、フレーズを解釈して自動的にダンパ制御位置を算出することにより、表現の幅を広げることができる。例えば、演奏データ(MIDIデータ等)からフレーズを解釈してフレーズの区切りを検出し、フレーズの開始位置ではダンパオン(サスティンペダルオン)とし、フレーズ終端位置ではダンパオフ(サスティンペダルオフ)とするように、演奏データに対して自動的にダンパ制御を行う。

30

【0466】

(g)「スコア譜を入力すると、自動的に楽器をアサインする」:

この発明の一実施例においては、楽譜編集装置において、スコア譜の段数を設定するだけで、これを典型的なスコア譜と比較し、音源の音色を自動的に設定するように構成される。また、このようにスコア譜入力に応じて自動的に楽器音色をアサインさせた後、さらに、音符を入力すると、その旋律が、設定した音色らしい旋律か否を学習結果と比較しながら、確からしさを評価するように構成される。

【0467】

〔種々の実施態様〕

なお、上述した実施の形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムまたは装置に供給し、そのシステムまたは装置のコンピュータ(またはCPUやMPU)が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、この発明の目的が達成されることは言うまでもない。

40

【0468】

この場合、記憶媒体から読出されたプログラムコード自体がこの発明の新規な機能を実現することになり、そのプログラムコードを記憶した記憶媒体はこの発明を構成することになる。

【0469】

プログラムコードを供給するための記憶媒体としては、たとえば、前記フレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気

50

テープ、不揮発性のメモ리카ード、ROM2などを用いることができる。また、他のMIDI機器17や通信ネットワーク19を介してサーバコンピュータ20からプログラムコードが供給されるようにしてもよい。

【0470】

また、コンピュータが読出したプログラムコードを実行することにより、上述した実施の形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOSなどが実際の処理の一部または全部を行い、その処理によって上述した実施の形態の機能が実現される場合も含まれることは言うまでもない。

【0471】

さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPU1などが実際の処理の一部または全部を行い、その処理によって上述した実施の形態の機能が実現される場合も含まれることは言うまでもない。

【0472】

〔実施例の特徴のまとめ〕

以上、この発明に関して種々の実施例をあげて説明したが、これらの実施例の基本的な特徴をまとめると、以下のとおりである：

所定の特徴情報と表情付けのための楽音制御情報との対応関係を表情付けモジュール（表情付けアルゴリズム）におけるルールとして設定し、この表情付けルールを表わす生成方法情報が記憶手段に予め記憶しておき、供給された演奏データの特徴情報を取得すると、取得された特徴情報に対応する生成方法情報に基づき、表情付けモジュール（表情付けアルゴリズム）に従って、楽音制御情報（時間パラメータ、音程パラメータ、音量パラメータ等の各種演奏パラメータ）を生成して演奏データに付加することにより、取得された特徴情報に応じて、ユーザが初心者であっても簡単な操作で曲に多様な表情付けを行うことができ、より音楽的な演奏データを自動的に作成することができる。さらに、楽音制御情報が付加されて出力される演奏データを評価し、評価結果に応じて、楽音制御情報を調節するようにしているので、最適の楽音制御情報による表情付けを行うことができる。

【0473】

また、演奏データの特徴には、音符時間情報としての音符密度や2音間隔、演奏の進行状態、微小震動音情報としてのロングトーンのトリル、ピッチベンド、ビブラート等、フレーズの区切り（フレーズ終端部）、音高情報、音高変化方向転換情報（音高上下降変化部）、同一又は類似のパターンの連続や類似フレーズ、登録音形（フレーズテンプレート）、音量情報、雰囲気情報としての「緊張感」、音符群列情報（まとまった音符、長い連符）、和音音数情報、音色情報、フィンガリング情報としての指、ポジション移動、ポジション等、奏法情報としてのギターのプリングオフ、ハンマリングオン、ピアノのサステインペダル等、微小変動音情報としてのトリル、歌詞情報、強弱記号やスタッカート等の演奏記号、等々があり、これら種々の特徴に従って、多彩且つ多様な表情のある演奏出力を得ることができる。

【0474】

また、既に供給されている演奏データの所定の特徴情報と楽音制御情報との関係を記憶しておき、新たに供給された演奏データの特徴を抽出すると、記憶手段に記憶された関係に従って、楽音制御情報を生成し、新たに供給された演奏データに付加することにより、学習機能によりテンポを設定する表情付けを行うことができる。また、演奏データに関して所定の特徴情報と楽音制御情報との関係をライブラリに記憶しておき、供給された演奏データの特徴情報を抽出すると、ライブラリを参照することによって、楽音制御情報を生成し、演奏データに付加することにより、ライブラリを用いてテンポを設定する表情付けを行うこともできる。

【0475】

また、供給された演奏データの所定の特徴情報に基づき楽音制御情報を生成し、生成さ

10

20

30

40

50

れた楽音制御情報及び供給された演奏データの楽音制御情報を演奏データ全体で対比し、この対比結果に基づいて、生成された楽音制御情報を修正することにより、演奏データ全体を見直して、演奏データを見通した上、バランスのとれた最適な楽音制御情報を設定することができる。

【0476】

さらに、供給される演奏データから、発音長、同音色パート、メロディパート、音量変化（アクセント）、ダブルチョーキング、連続して行われるチョーキング、アルペジオ演奏、音色変更／復帰指示用音楽記号情報などの特徴を抽出し、これらの特徴情報に基づいて、音量パラメータ、音程パラメータ、別パートでの倍音発音、音色データなどを編集することにより、さらに、多彩且つ多様な表情のある演奏出力を得ることができる。

10

【図面の簡単な説明】

【0477】

【図1】図1は、この発明の一実施例による演奏データ作成システムのハードウェア構成を示すブロック図である。

【図2】図2は、この発明の演奏データ作成システムによる機能の概要を示す機能的ブロック図である。

【図3】図3は、この発明の実施例（1）において使用される譜例を示す図である。

【図4】図4は、この発明の実施例（1）において使用される「音符密度 - テンポ係数」テーブルの例を示す図である。

【図5】図5は、この発明の実施例（1）による音符密度応答処理の一例を示すフローチャートである。

20

【図6】図6は、この発明の実施例（1）による音符密度応答処理の他の例を示すフローチャートである。

【図7】図7は、この発明の実施例（2）による曲の進行状態にตอบสนองする処理を示すフローチャートである。

【図8】図8は、この発明の実施例（2）における曲の進行に従ったテンポ変化の例を示す図である。

【図9】図9は、この発明の実施例（3）における音符時間間隔とテンポの関係を示す図である。

【図10】図10は、この発明の実施例（4）によるロングトーンのトリル／ビブラートに対する処理を示すフローチャートである。

30

【図11】図11は、この発明の実施例（4）におけるロングトーンのトリル／ビブラートに対するテンポ変化曲線を示す図である。

【図12】図12は、この発明の実施例（5）におけるフレーズ解釈によるテンポ変化の例を示す図である。

【図13】図13は、この発明の実施例（5）によるフレーズ終端部への処理を示すフローチャートである。

【図14】図14は、この発明の実施例（6）による音域に対応した処理の一例を示すフローチャートである。

【図15】図15は、この発明の実施例（6）において使用される「音高 - テンポ係数」

40

テーブルの例を示す図である。

【図16】図16は、この発明の実施例（6）による音域に対応した処理の他の例を示すフローチャートである。

【図17】図17は、この発明の実施例（7）による音高が上昇から下降に移行する際の処理を示すフローチャートである。

【図18】図18は、この発明の実施例（7）における音高の上下降とテンポ変化の関係を示す図である。

【図19】図19は、この発明の実施例（8）による同じパターンが連続した際の処理を示すフローチャートである。

【図20】図20は、この発明の実施例（9）による同一／類似フレーズへの同一／類似

50

テンポ表現を付与する処理を示すフローチャートである。

【図 2 1】図 2 1 は、この発明の実施例 (1 0) におけるフレーズテンプレートの例を示す図である。

【図 2 2】図 2 2 は、この発明の実施例 (1 0) による登録音形を用いた処理を示すフローチャートである。

【図 2 3】図 2 3 は、この発明の一実施例による鍵盤楽器におけるスラーの処理を説明するための図である。

【図 2 4】図 2 4 は、この発明の一実施例による管楽器におけるスラーの処理を説明するための図である。

【図 2 5】図 2 5 は、この発明の実施例 (1 1) によるテンポを音量で変化させる処理の一例を示すフローチャートである。 10

【図 2 6】図 2 6 は、この発明の実施例 (1 1) によるテンポを音量で変化させる処理の他の例を示すフローチャートである。

【図 2 7】図 2 7 は、この発明の実施例 (1 2) において使用される「緊張感」つまみの例を示す図である。

【図 2 8】図 2 8 は、この発明の実施例 (1 2) による「緊張感」に基づく処理を示すフローチャートである。

【図 2 9】図 2 9 は、この発明の実施例 (1 3) によるまとまった音符群に対する処理を示すフローチャートである。

【図 3 0】図 3 0 は、この発明の実施例 (1 4) による長い連符に対する処理を示すフローチャートである。 20

【図 3 1】図 3 1 は、この発明の実施例 (1 5) による和音の音数に応じた処理を示すフローチャートである。

【図 3 2】図 3 2 は、この発明の実施例 (1 6) による音色に連動する処理を示すフローチャートである。

【図 3 3】図 3 3 は、この発明の実施例 (1 7) によるフィンガリングの指に応じた処理を示すフローチャートである。

【図 3 4】図 3 4 は、この発明の実施例 (1 8) によるフィンガリングのポジション移動に応じた処理を示すフローチャートである。

【図 3 5】図 3 5 は、この発明の実施例 (1 9) によるフィンガリングのポジションの高低に応じた処理を示すフローチャートである。 30

【図 3 6】図 3 6 は、この発明の実施例 (2 0) によるピッチベンドをかける際の処理を示すフローチャートである。

【図 3 7】図 3 7 は、この発明の実施例 (2 1) によるギターのプリングオフに対応する処理を示すフローチャートである。

【図 3 8】図 3 8 は、この発明の実施例 (2 2) によるピアノのサスティンペダル操作に対応する処理を示すフローチャートである。

【図 3 9】図 3 9 は、この発明の実施例 (2 3) における「ビブラート深さ - テンポ係数」テーブルの例を示す図である。

【図 4 0】図 4 0 は、この発明の実施例 (2 3) における「ビブラート速さ - テンポ係数」テーブルの例を示す図である。 40

【図 4 1】図 4 1 は、この発明の実施例 (2 3) によるビブラートの深さ及び速さに応じた処理を示すフローチャートである。

【図 4 2】図 4 2 は、この発明の実施例 (2 4) におけるストリングスのトリルの再現例を示す図である。

【図 4 3】図 4 3 は、この発明の実施例 (2 4) によるストリングスのトリルに対する複数パートでの処理を示すフローチャートである。

【図 4 4】図 4 4 は、この発明の実施例 (2 5) における学習機能に基づく処理を示すフローチャートである。

【図 4 5】図 4 5 は、この発明の実施例 (2 6) によるライブラリ化による処理を示すフ 50

ローチャートである。

【図46】図46は、この発明の実施例(27)による歌詞に基づく処理を示すフローチャートである。

【図47】図47は、この発明の実施例(28)における音源出力波形と補正タイミングの関係を表わす図である。

【図48】図48は、この発明の実施例(28)による音源出力波形に応じた処理を示すフローチャートである。

【図49】図49は、この発明の実施例(29)による強弱記号列に対応した処理を示すフローチャートである。

【図50】図50は、この発明の実施例(30)によるスタッカートに対応した処理を示すフローチャートである。

10

【図51】図51は、この発明の実施例(31)による全体的見直し処理を示すフローチャートである。

【図52】図52は、図1のパラメータ自動編集装置、特にCPUが実行するメインルーチンの手順を示すフローチャートである。

【図53】図53は、図2のパラメータ変化付与処理の手順を示すフローチャートである。

【図54】図54は、演奏データの音量パラメータをその音高の変化に応じて変化させるという表情付けを行う音量パラメータ変化付与処理1の手順を示すフローチャートである。

20

【図55】図55は、音高が上昇傾向にあるノートイベント列の一例を示す図である。

【図56】図56は、フィルタ処理が施された後の時系列データの一例を示す図である。

【図57】図57は、演奏データの音量パラメータの値を徐々に増大させることにより、興奮度を高めるといった表情付けを行う音量パラメータ変化付与処理2の手順を示すフローチャートである。

【図58】図58は、変化付与パターンの一例を示す図である。

【図59】図59は、この発明の一実施例における音符密度と音程の不安定さの関係の一例を示す図である。

【図60】図60は、連続演奏の細かさに応じて音量を変化させるという表情付けを行う音量パラメータ変化付与処理3の手順を示すフローチャートである。

30

【図61】図61は、音符密度と音量との関係を示すテーブルデータの一例を示す図である。

【図62】図62は、この発明の一実施例において音符密度と音程の不安定さとの関係を示すテーブルデータの一例を示す図である。

【図63】図63は、類似フレーズが繰り返し出現する選択演奏データの場合に、2番目以降の類似フレーズの音量パラメータを、その類似度および出現態様に依りて変化させるという表情付けを行う音量パラメータ変化付与処理4の手順を示すフローチャートである。

【図64】図64は、3拍子の演奏データの場合に、同一音符長の音符が連続して出現するときには1拍目を強調するという表情付けを行う音量パラメータ変化付与処理5の手順を示すフローチャートである。

40

【図65】図65は、フレーズの終止では、音量を抑えるという表情付けを行う音量パラメータ変化付与処理6の手順を示すフローチャートである。

【図66】図66は、ロングトーンに対してトリルまたはビブラートが付与されているときには、その音量を、最初大きく、途中小さく、最後大きくなるような表情付けを行う音量パラメータ変化付与処理7の手順を示すフローチャートである。

【図67】図67は、変化付与パターンの一例を示す図である。

【図68】図68は、曲の進行経過に対するトリル/ビブラート速さ変化の関係の一例を示す図である。

【図69】図69は、トリルや打楽器のロール演奏時に自然な表情付けを行う音量パラメ

50

ータ変化付与処理 8 の手順を示すフローチャートである。

【図 7 0】図 7 0 は、和音が含まれている演奏データの場合に、和音に対してその発音をすっきりしたものにするという表情付けを行う音量パラメータ変化付与処理 9 の手順を示すフローチャートである。

【図 7 1】図 7 1 は、スタッカートが含まれている演奏データの場合に、生のスタッカート演奏を模倣した表情付けを行う音量パラメータ変化付与処理 1 0 の手順を示すフローチャートである。

【図 7 2】図 7 2 は、ロングトーンに対して、その音量にゆらぎをつけるという表情付けを行う音量パラメータ変化付与処理 1 1 の手順を示すフローチャートである。

【図 7 3】図 7 3 は、乱数カウンタのカウント値に応じて、乱数カウンタが発生する乱数の振幅を変化させるテーブルデータの一例を示す図である。 10

【図 7 4】図 7 4 は、音程平均に応じた音量表現による表情付けを行う音量パラメータ変化付与処理 1 2 の手順を示すフローチャートである。

【図 7 5】図 7 5 は、パートディビジョンのときには各パートの音量は下げるといった表情付けを行う音量パラメータ変化付与処理 1 3 の手順を示すフローチャートである。

【図 7 6】図 7 6 は、音色変化に応じた音量表現による表情付けを行う音量パラメータ変化付与処理 1 4 の手順を示すフローチャートである。

【図 7 7】図 7 7 は、ピッチベンド時には音量が変化するという表情付けを行う音量パラメータ変化付与処理 1 5 の手順を示すフローチャートである。

【図 7 8】図 7 8 は、マルチパートからなる演奏データの場合に、そのメロディラインを浮き立たせるといった表情付けを行う音量パラメータ変化付与処理 1 6 の手順を示すフローチャートである。 20

【図 7 9】図 7 9 は、フィンガリングに応じた音量変化により生々しい演奏を再現するという表情付けを行う音量パラメータ変化付与処理 1 7 の手順を示すフローチャートである。

【図 8 0】図 8 0 は、チェロの運指の例を示す図である。

【図 8 1】図 8 1 は、弓を用いて演奏する弦楽器の演奏データの場合に、弦楽器の弓の返しに応じた音量変化により運弓感を再現するという表情付けを行う音量パラメータ変化付与処理 1 8 の手順を示すフローチャートである。

【図 8 2】図 8 2 は、変化付与パターンの一例を示す図である。 30

【図 8 3】図 8 3 は、歌詞を伴う演奏データの場合に、歌詞の内容に応じて抑揚を変更するという表情付けを行う音量パラメータ変化付与処理 1 9 の手順を示すフローチャートである。

【図 8 4】図 8 4 は、発音後の音量制御を規定するためのテーブルデータの一例を示す図である。

【図 8 5】図 8 5 は、この発明の実施例 (a) によるアクセント音に対する処理を示すフローチャートである。

【図 8 6】図 8 6 は、この発明の実施例 (a) における音量及び音程の時間変化の関係を表わす図である。

【図 8 7】図 8 7 は、この発明の実施例 (b) におけるダブルチョーキングの音程の時間変化の関係を表わす図である。 40

【図 8 8】図 8 8 は、この発明の実施例 (b) によるダブルチョーキングでの処理を示すフローチャートである。

【図 8 9】図 8 9 は、この発明の実施例 (c) における連続したチョーキングの例を示す図である。

【図 9 0】図 9 0 は、この発明の実施例 (c) による連続したチョーキングでの処理を示すフローチャートである。

【図 9 1】図 9 1 は、この発明の実施例 (d) におけるアルペジオと共通の倍音の例を示す図である。

【図 9 2】図 9 2 は、この発明の実施例 (d) によるアルペジオでの処理を示すフローチャート 50

ャートである。

【図93】図93は、この発明の実施例(e)における「arco」と「pizz.」の切替えの例を示す図である。

【図94】図94は、この発明の実施例(e)による楽譜のシンボルによるの音色選択処理を示すフローチャートである。

【符号の説明】

【0478】

1 CPU(解析手段、決定手段、パラメータ編集手段、抽出手段、算出手段、検出手段、設定手段)、

2 ROM(記憶手段)、

9 FD(記憶手段)を駆動するFDD、HDD、CD-ROMD等の外部記憶装置(供給手段、読み出し手段)、

16 MIDIインターフェース(供給手段)、

18 通信インターフェース(供給手段)、

OD 元演奏データ、

ED 表情付け後の演奏データ、

Ko 当初テンポ値、

to テンポ変化開始時点、

ta 最下テンポ値Kdに到達する時点、

tb 目標テンポ値Ktに到達する時点、

K テンポ変化幅、

Kt テンポ目標変化量、

n1, n2 ノイズ的音高変化を与える音、

13k 「緊張感」つまみ、

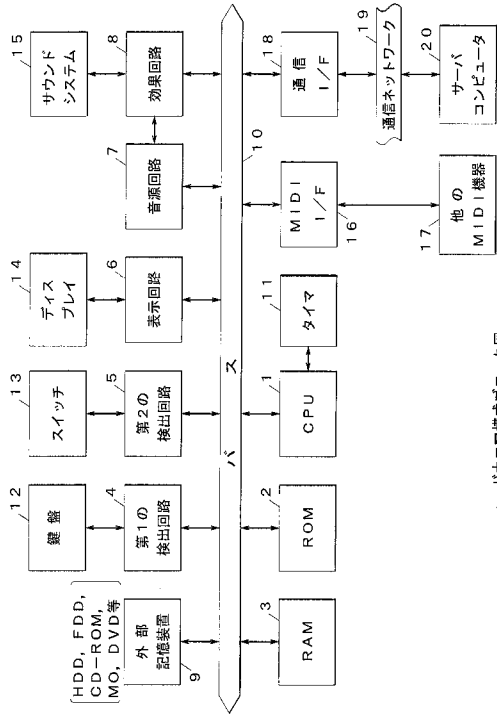
TW 時点tsで立ち上がり時点tmで最高音量値に到達する音源出力波形、

tc 補正タイミングを与える80%音量値到達時点。

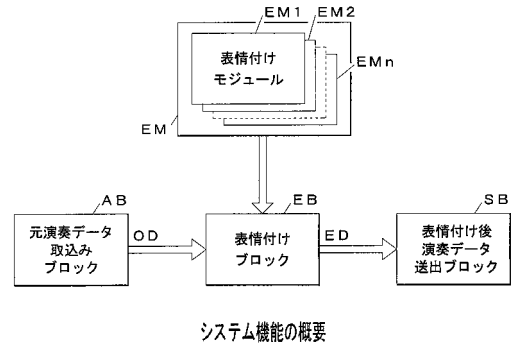
10

20

【 図 1 】



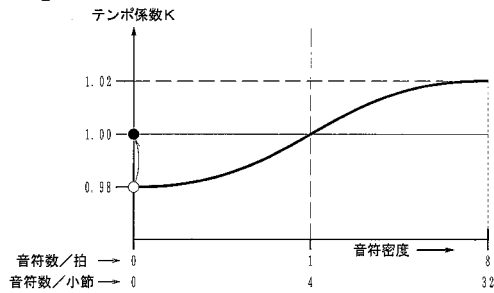
【 図 2 】



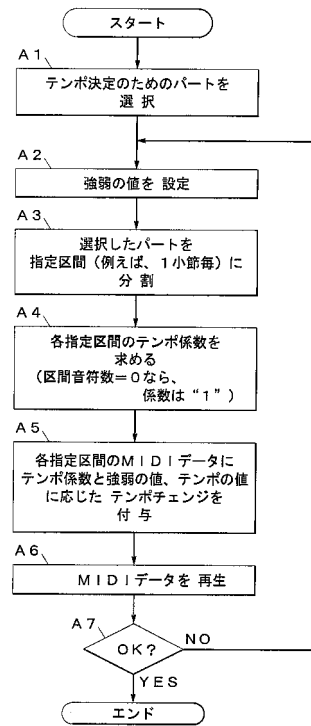
【 図 3 】



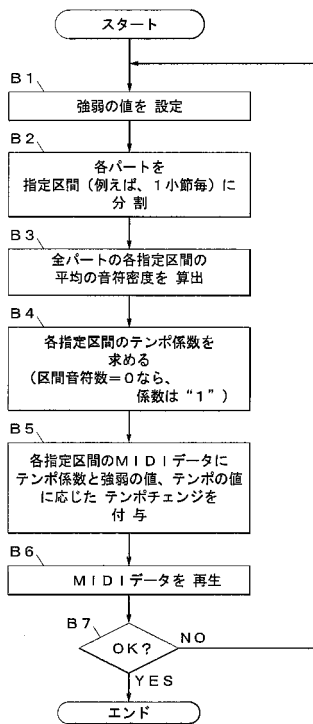
【 図 4 】



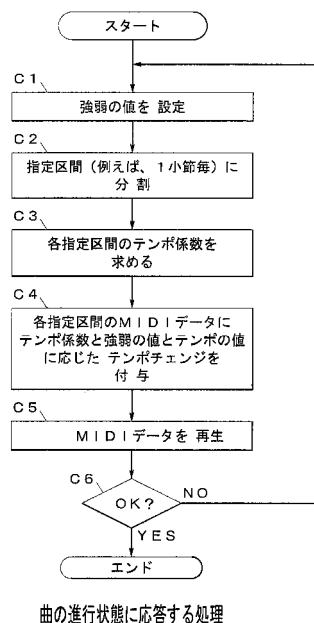
【 図 5 】



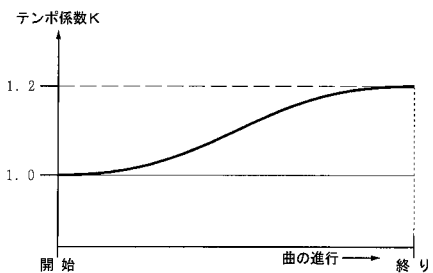
【 図 6 】



【 図 7 】

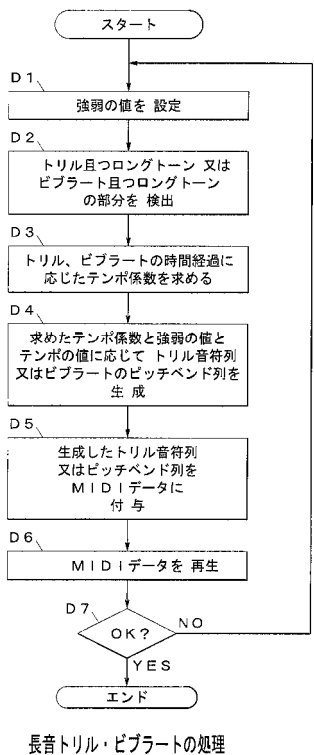


【 図 8 】

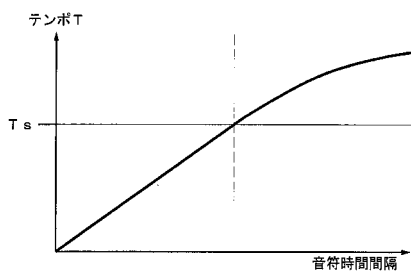


曲の進行に従ったテンポ変化の例

【 図 10 】

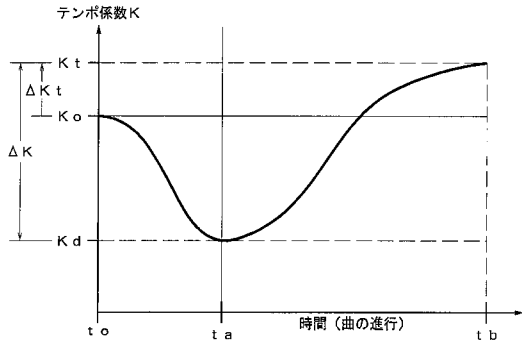


【 図 9 】



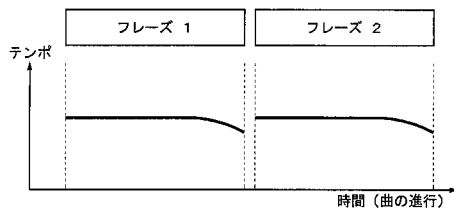
音符時間間隔とテンポの関係の例

【 図 1 1 】



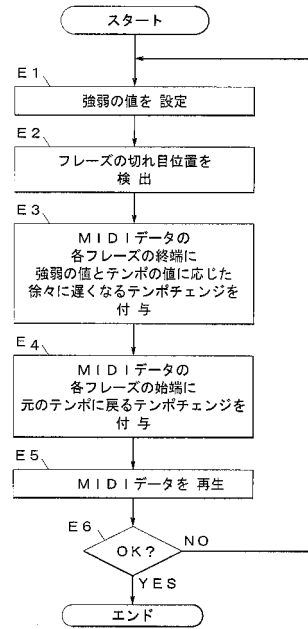
長音トリル/ピラートに対するテンポ変化曲線

【 図 1 2 】



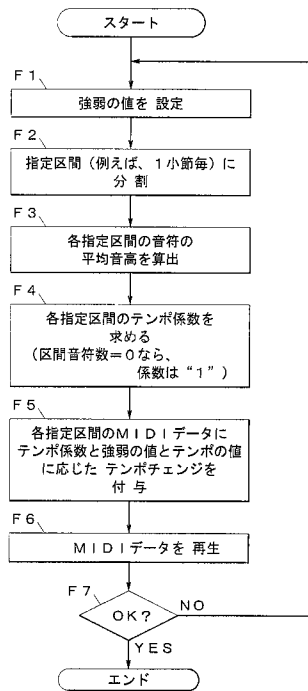
フレーズ解釈によるテンポ変化の例

【 図 1 3 】



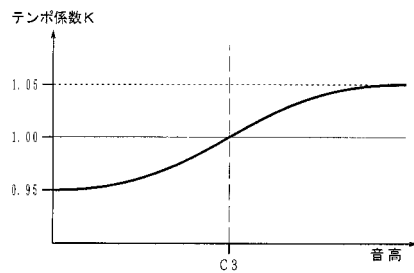
フレーズ終端部への処理

【 図 1 4 】



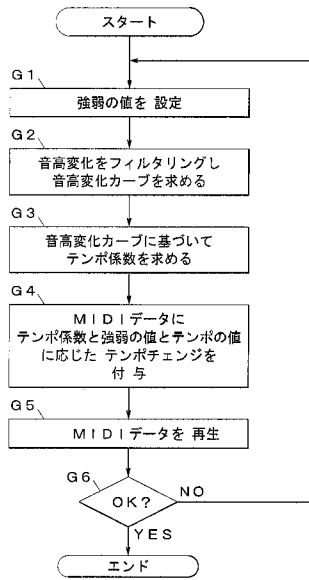
音域に対応した処理 [1]

【 図 1 5 】



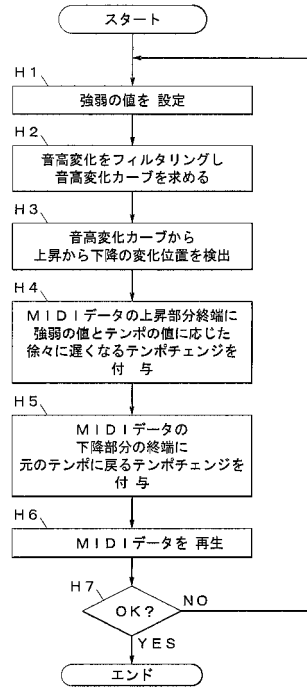
音高-テンポ係数テーブルの例

【 図 1 6 】



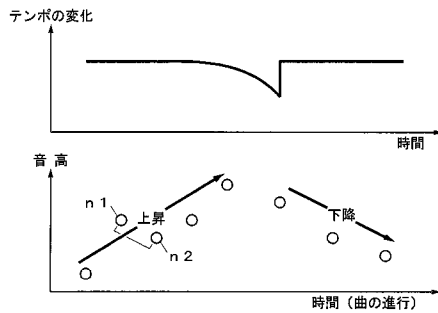
音域に対応した処理 [2]

【 図 1 7 】



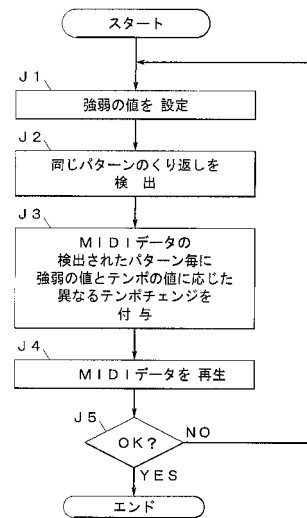
音高上下降部での処理

【 図 1 8 】



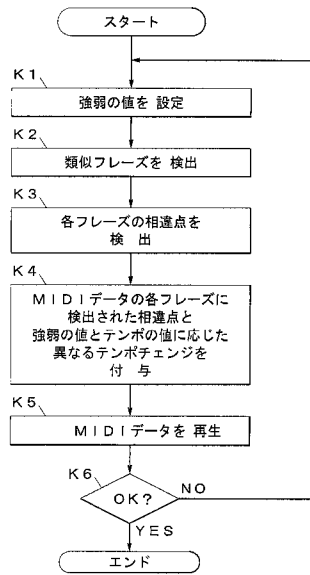
音高の上下降とテンポ変化

【 図 1 9 】



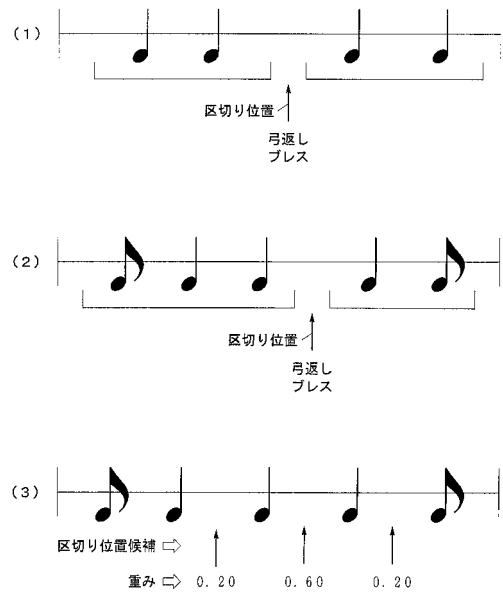
同パターン連続時の処理

【 図 2 0 】



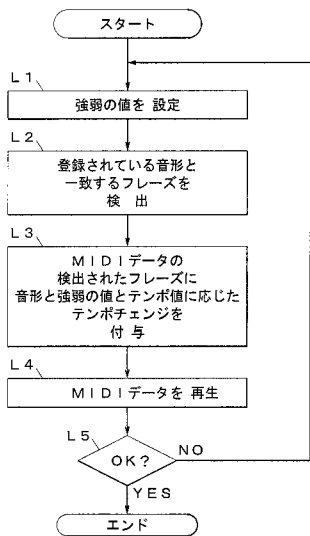
類似フレーズへの類似テンポ設定処理

【 図 2 1 】



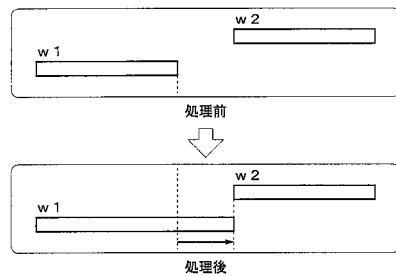
フレーズテンプレートの例

【 図 2 2 】



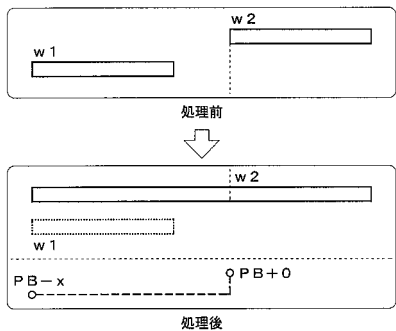
登録音形を用いた処理

【 図 2 3 】



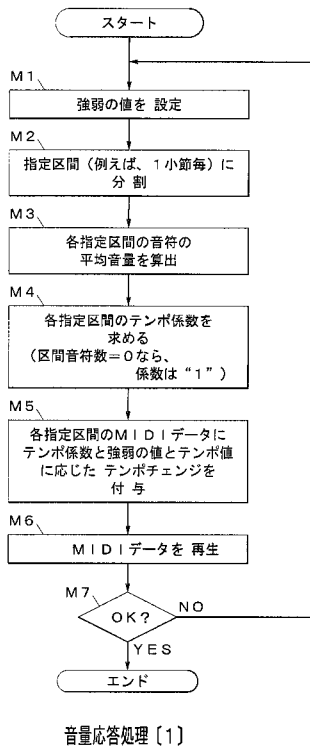
ピアノのスラーの例

【 図 2 4 】

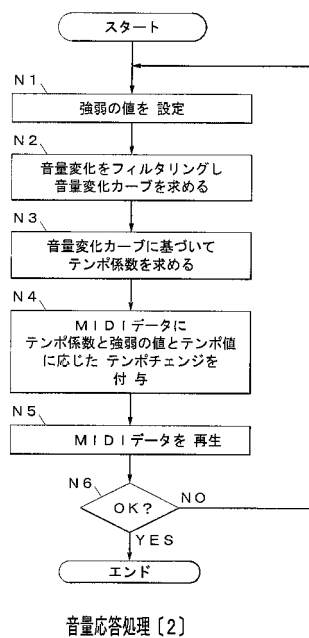


管楽器のスラーの例

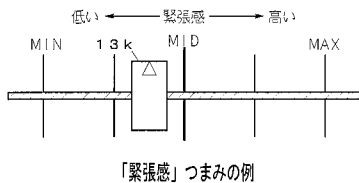
【 図 2 5 】



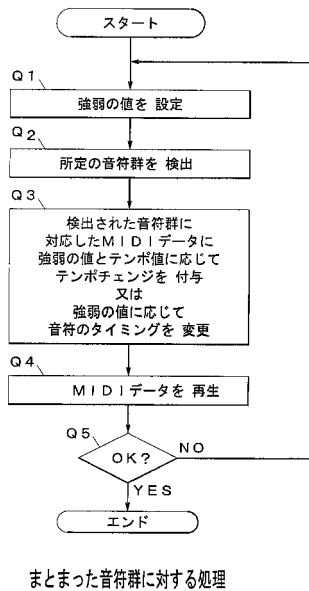
【 図 2 6 】



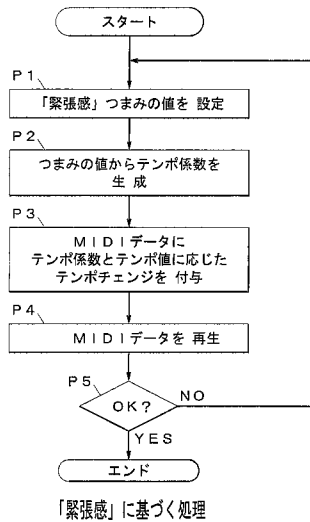
【 図 2 7 】



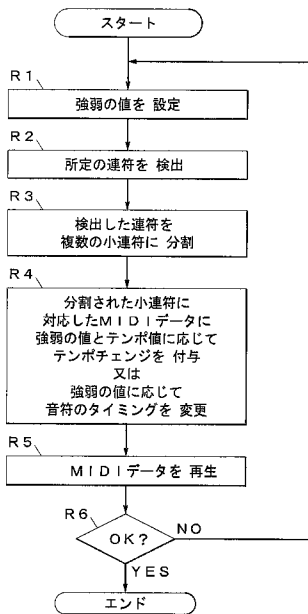
【 図 2 9 】



【 図 2 8 】

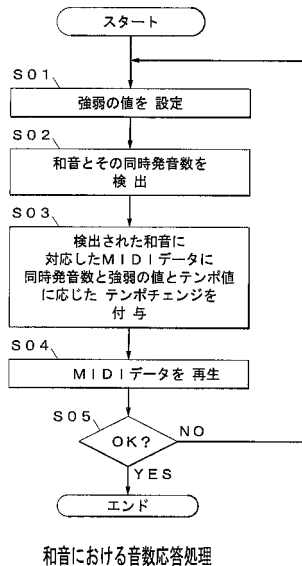


【 図 3 0 】



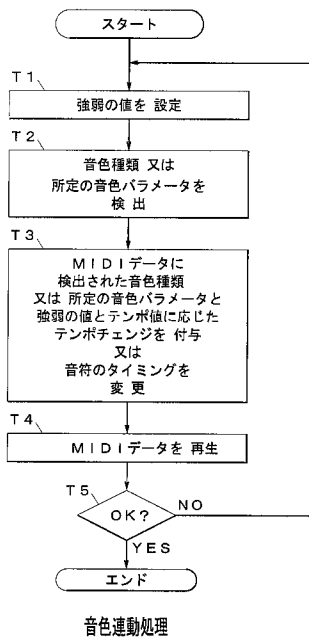
長い連符に対する処理

【 図 3 1 】



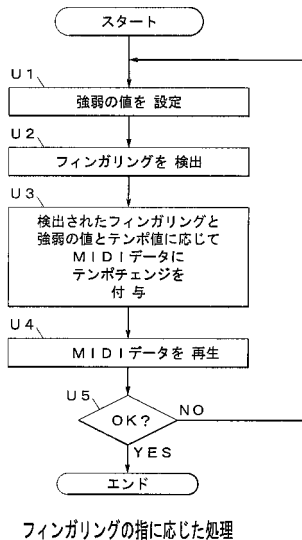
和音における音数応答処理

【 図 3 2 】



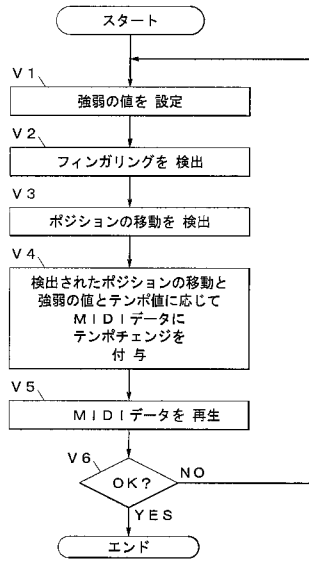
音色連動処理

【 図 3 3 】



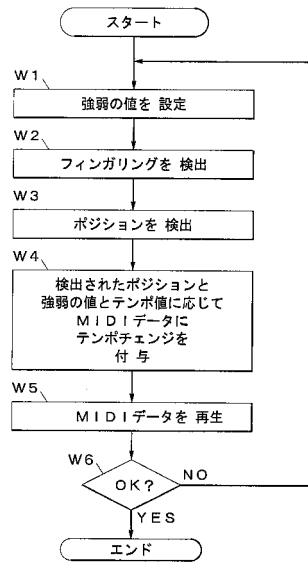
フィンガリングの指に応じた処理

【 図 3 4 】



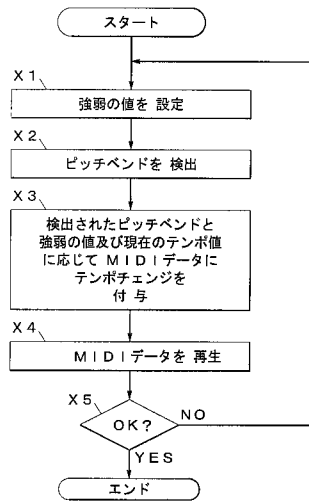
フィンガリングポジション移動応答処理

【 図 3 5 】



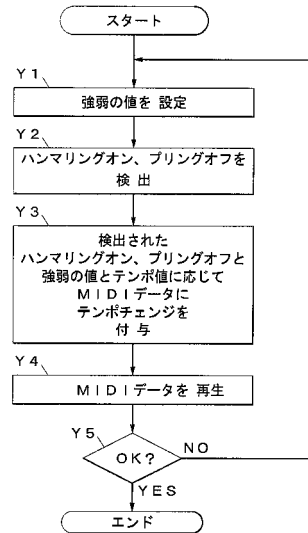
フィンガリングポジション応答処理

【 図 3 6 】



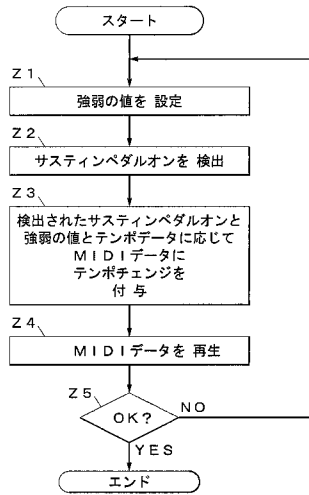
ピッチベンド対応処理

【 図 3 7 】



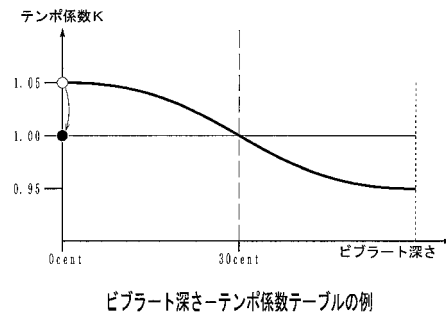
ブリングオフ等応答処理

【 図 3 8 】



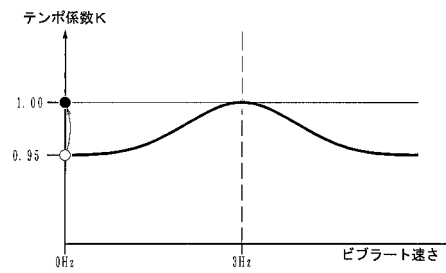
サスティンペダル応答処理

【 図 3 9 】



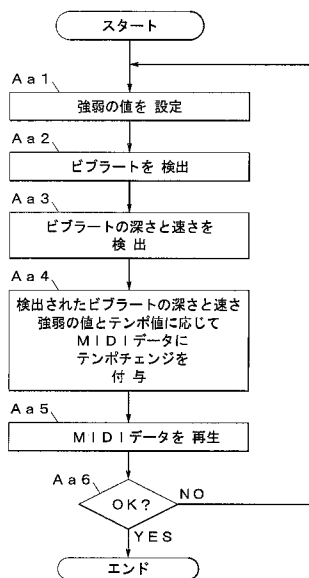
ビブラート深さ-テンポ係数テーブルの例

【 図 4 0 】



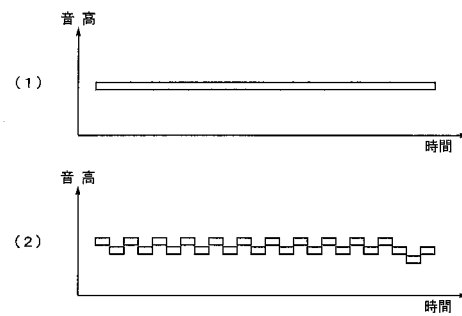
ビブラート速さ-テンポ係数テーブルの例

【 図 4 1 】



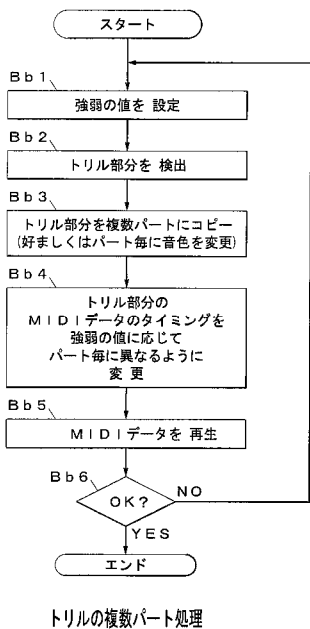
ビブラート深さ/速さ応答処理

【 図 4 2 】

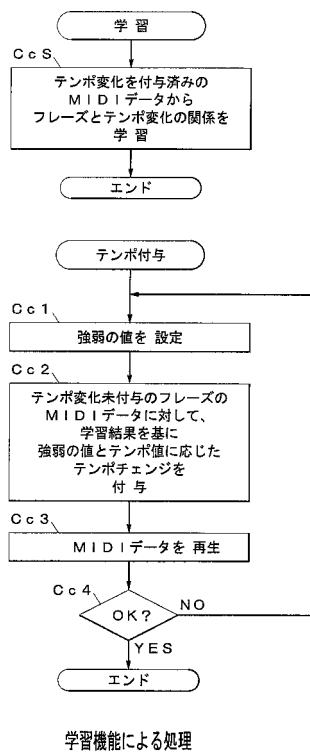


トリルの再現例

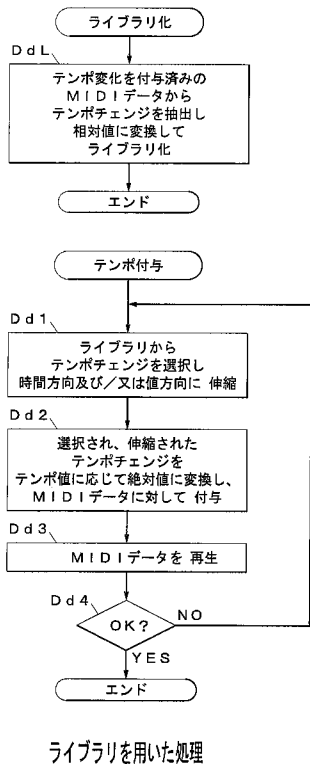
【 図 4 3 】



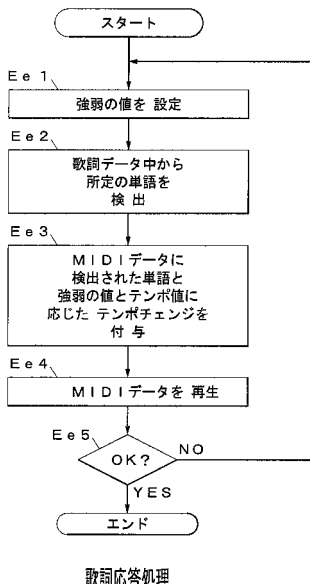
【 図 4 4 】



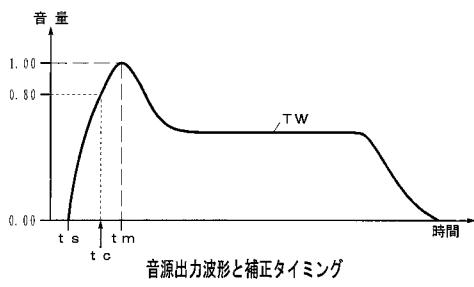
【 図 4 5 】



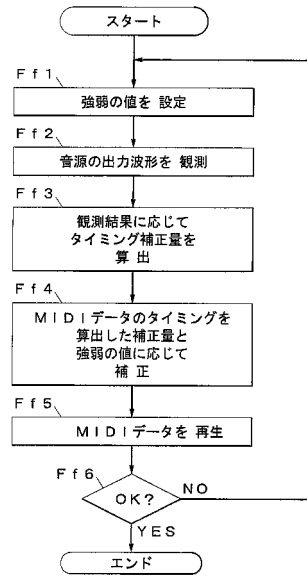
【 図 4 6 】



【 図 4 7 】

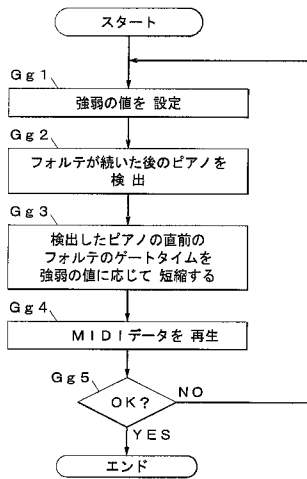


【 図 4 8 】



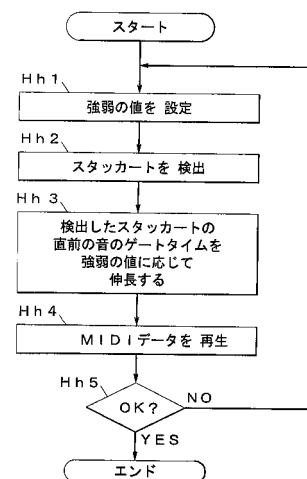
音源出力波形に対応した処理

【 図 4 9 】



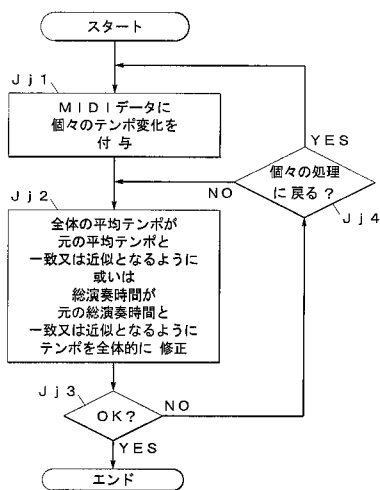
強弱記号に対応した処理

【 図 5 0 】



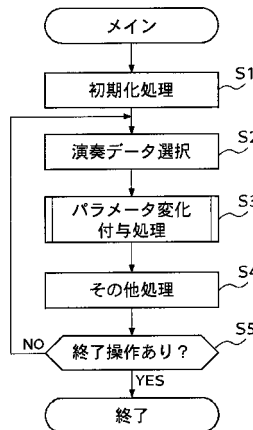
スタッカートに対応した処理

【 図 5 1 】

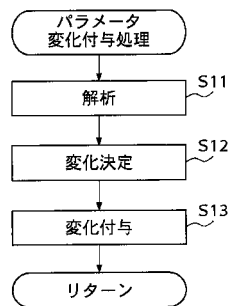


整体的見直し処理

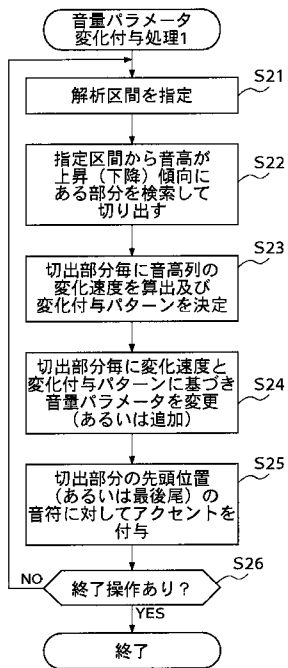
【 図 5 2 】



【 図 5 3 】



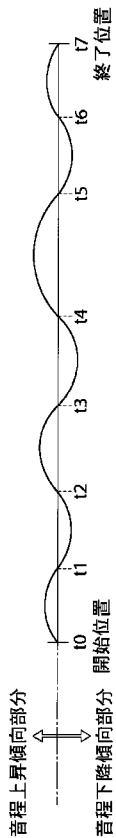
【 図 5 4 】



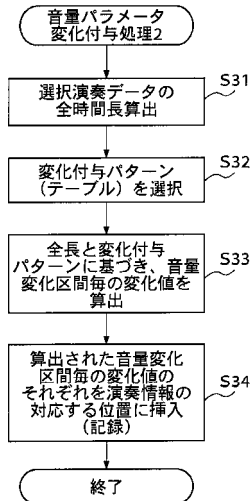
【 図 5 5 】



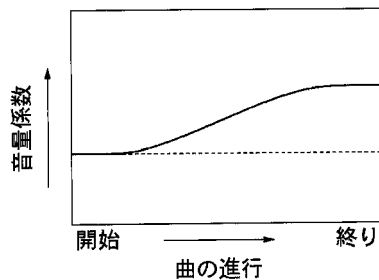
【 図 5 6 】



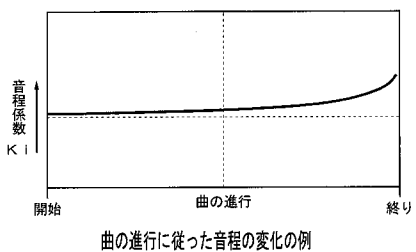
【 図 5 7 】



【 図 5 8 】

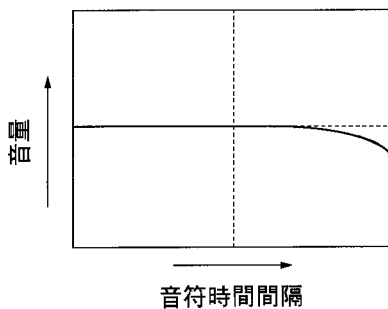


【 図 5 9 】

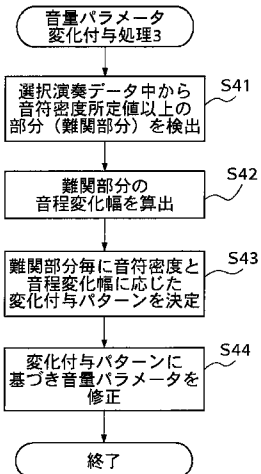


曲の進行に従った音程の変化の例

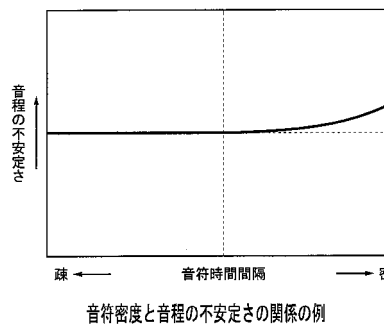
【 図 6 1 】



【 図 6 0 】

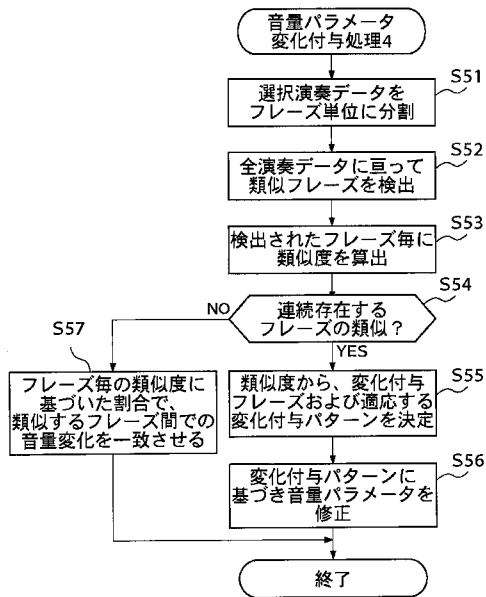


【 図 6 2 】

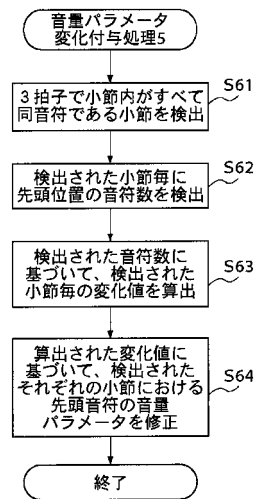


音符密度と音程の不安定さの関係の例

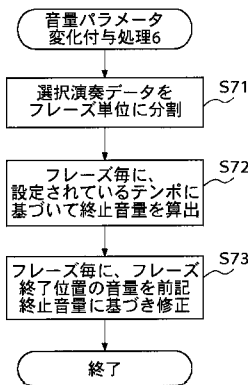
【 図 6 3 】



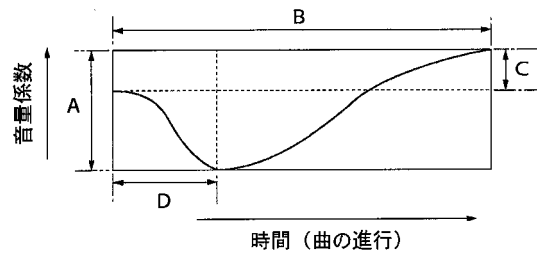
【 図 6 4 】



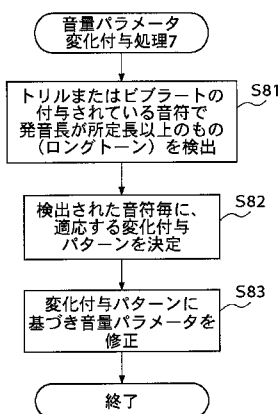
【 図 6 5 】



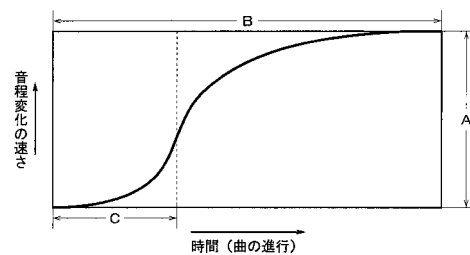
【 図 6 7 】



【 図 6 6 】

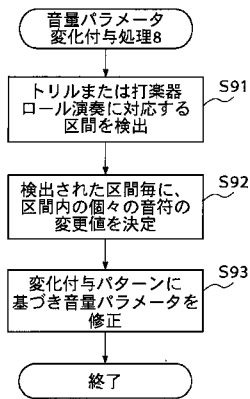


【 図 6 8 】

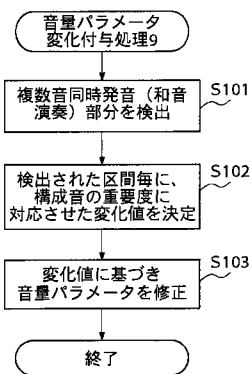


トリル、ビブラートの速さの変化の例

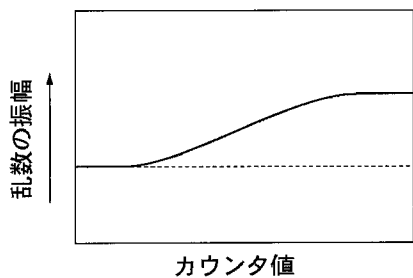
【図 69】



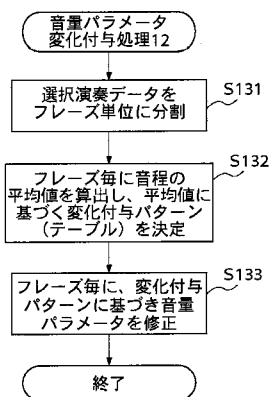
【図 70】



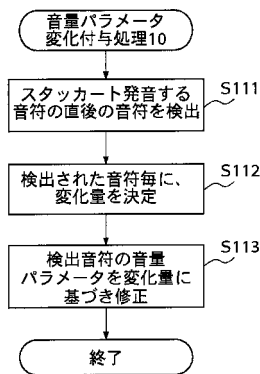
【図 73】



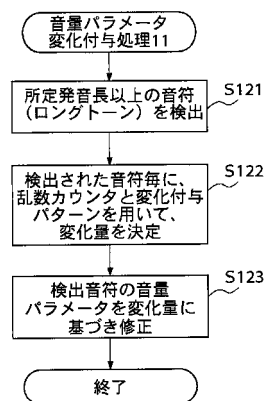
【図 74】



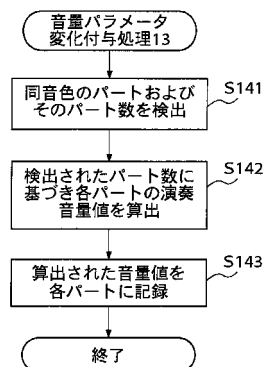
【図 71】



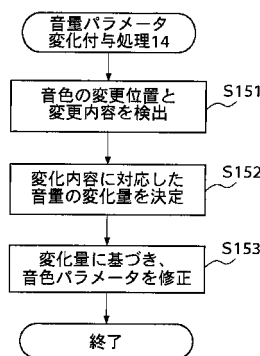
【図 72】



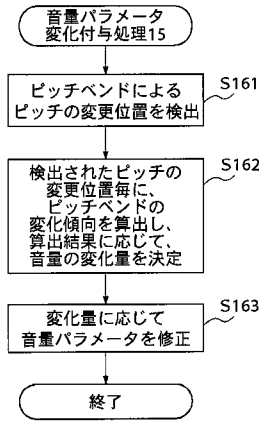
【図 75】



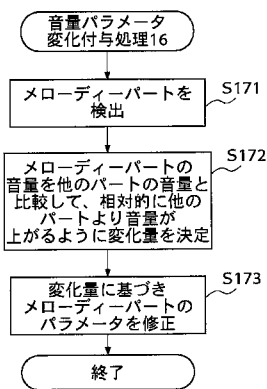
【図 76】



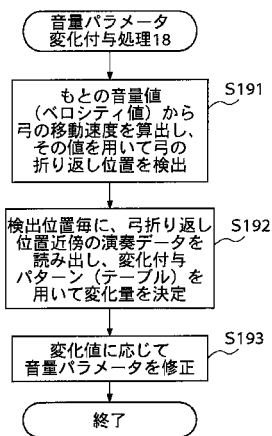
【 図 7 7 】



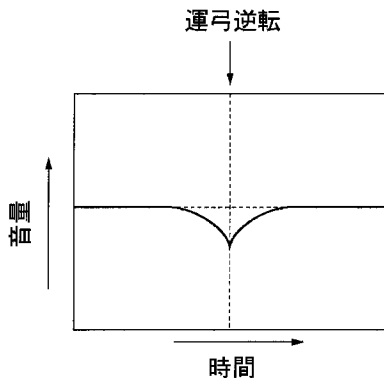
【 図 7 8 】



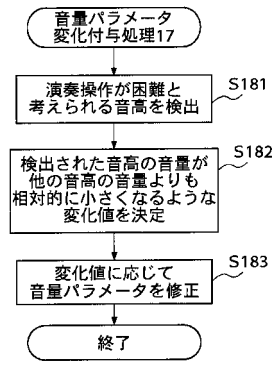
【 図 8 1 】



【 図 8 2 】



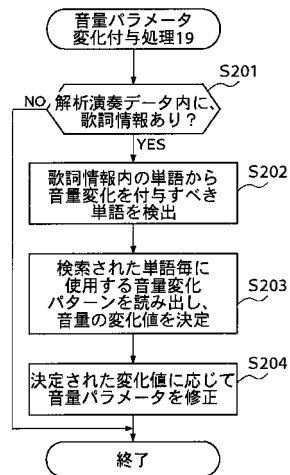
【 図 7 9 】



【 図 8 0 】



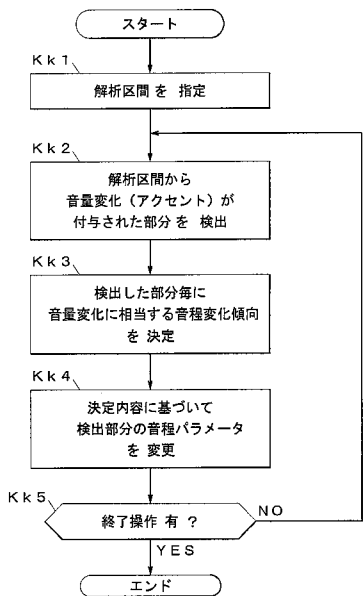
【 図 8 3 】



【 図 8 4 】

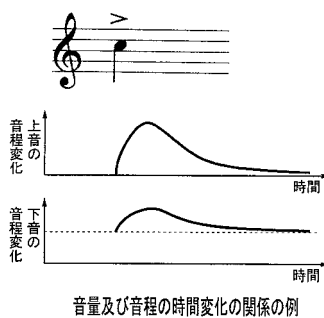
音色	発音後制御
ピアノ	不許可
ギター	不許可
バイオリン	許可
フルート	許可

【 図 8 5 】



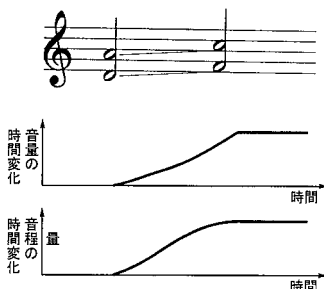
アクセント音に対する処理

【 図 8 6 】



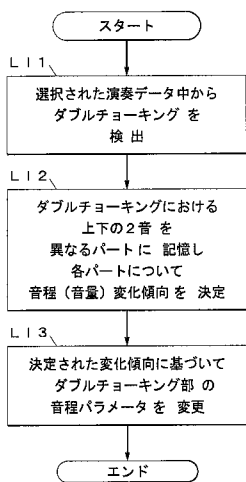
音量及び音程の時間変化の関係の例

【 図 8 7 】



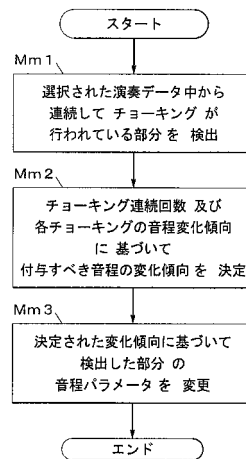
ダブルチョーキングの音程の時間変化の例

【 図 8 8 】



ダブルチョーキングでの処理

【 図 9 0 】



連続したチョーキングでの処理

【 図 8 9 】



連続したチョーキングの例

【 図 9 1 】

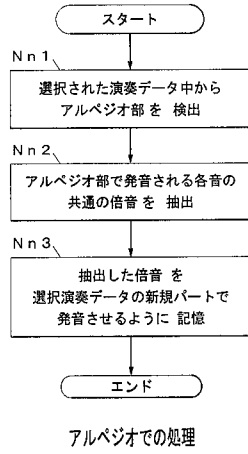


(1) アルペジオの例

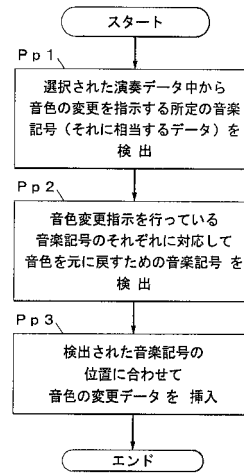
アルペジオと共通の倍音

(2) 共通の倍音

【 図 9 2 】



【 図 9 4 】



楽譜のシンボルによる音色選択処理

【 図 9 3 】



フロントページの続き

- (72)発明者 北山 徹
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 岩本 俊幸
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 鈴木 則夫
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 彌富 あかね
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 山内 明
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06-337690(JP,A)
特開昭61-200786(JP,A)
特開平07-056574(JP,A)

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