Vol.2014-HCI-157 No.17
Vol.2014-GN-91 No.17
Vol.2014-EC-31 No.17
2014/3/14

自由なテンポで演奏した複数の録音データから楽曲を生成するシステム

川名勇気[†1]　宮下芳明[†2†3]

本研究では，1 つの楽曲を制作する際に，各パートの奏者がテンポを自由に演奏した録音データを統合して，合奏を行っているような楽曲を生成できるシステムを提案する．複数の録音データのテンポを同期させるための「マーク付け」と，生成する楽曲のテンポとなる「指揮データ」を選択することで，他の録音データがこの指揮データに合わせて合奏しているかのような音楽を生成することができる．本システムによって，現在の録音方法で用いられている，ガイドリズムのクリック音や既存の演奏のテンポに合わせて演奏を行うといった方法をとる必要がなくなる．そのため，奏者はテンポに合わせることから解放され，自由に演奏を行うことができるので演奏表現が十分に発揮される．また，楽曲の制作者は，DAW 上での複雑な作業から解放される．これまでは，テンポが既に決まっている楽曲を 1 つしか作ることができなかったが，本システムでは録音データの数だけ指揮データを選ぶことができるため，テンポによって生まれる音楽表現について，楽曲の創造の幅を広げることが可能となっている．

A System Which Generates one Synchronized Music from Multiple Recorded Data Performed at Free Tempo

YUKI KAWANA[†1]　HOMEI MIYASHITA[†2†3]

In this paper, we propose a system which can generate a synchronized music from multiple recorded data performed at free tempo. This system has two actual function. One is “mark up” to synchronize a tempo of multiple performance data. The other is “selection conductor data” to set the standard tempo of the generated music. Recorded data except conductor data synchronize this. This system enables players to record musical performances without listening to a tempo of the clicking emitted by rhythm machines. As a result, players are freed to keep pace with this tempo, then exert own performance expression. Music creators are freed to complicate works on Digital AudioWorkstation. We have could make only one music fixed the tempo from the basis. We can select conductor data from the number of recorded data by using this system. Therefore the generated music has wide dimensions of creation about expression.

1. はじめに

楽曲の制作方法は，1 つの曲を基にして楽器奏者や歌手が各自のパートの演奏を行って録音をし，全員の録音をミキシングして 1 つの音楽を作り上げることが一般的である．近年はコンピュータの普及によって，これまで音楽制作の専門家のみが扱えた DAW（Digital Audio Workstation）システムにも，Singer Song Writer[1]のような初心者でも簡単に扱えるものが多数開発されるようになった．これにより，楽器演奏経験の少ない人や，音楽制作をする上で必要となる知識を持たない人でも容易に楽曲を作ることが可能になってきた．

そのため Web 上には，複数パートによる合奏曲や既存の演奏に合わせて自分が演奏を行った後に重ね合わせた作品などが数多く公開されるようになった．複数パートによる合奏曲はコンサートやライブをそのまま録音しているものや，演奏者が自分のパートをそれぞれの自宅で録音して後で誰かがミキシングしているもの，中には自分で複数の楽器を担当して多重録音を行っているものもある．既存の演奏に合わせて自分が演奏を行った後に重ね合わせた作品は，既存の曲をヘッドフォンなどで聞いて自分のパートを演奏して録音を行っている．

コンサートやライブは例外的であるが，録音をする際にはミキシングをした際に演奏者によって，テンポが乖離してしまうことを防ぐための措置が必要となる．そのため，現在の録音方法はドンカマチック[2]に代表されるようなリズムマシンから出されるガイドリズムのクリック音のテンポや，既存の演奏のテンポに合わせて演奏を行うといった方法がとられている．しかし実際には，楽器演奏経験の多い人や発せられるテンポに合わせることに慣れている人でない限り，自分の演奏表現を維持したまま合わせて演奏することは難しい．しかし，だからといって奏者が自由なテンポで演奏を行った場合には，DAW 上でのミキシング時に複雑な作業を要すため，音楽制作の知識が浅い人にとっては不可能に近いものとなってしまう．

西本らは楽器デザインについて次のような考え方を提唱している[3].

従来は好むと好まざるとにかかわらず「はじめに楽器ありき」であり，作曲者や演奏者は与えられた楽器の表現と

†1 明治大学理工学部情報科学科
Department of Computer Science, Meiji University
†2 明治大学総合数理学部先端メディアサイエンス学科
Department of Frontier Media Science, School of Interdisciplinary Mathematical Sciences, Meiji University
†3 独立行政法人科学技術振興機構, CREST
JST, CREST

Vol.2014-HCI-157 No.17
Vol.2014-GN-91 No.17
Vol.2014-EC-31 No.17
2014/3/14

操作系の枠の中で新たな楽曲や演奏を創作するのが常であった.（中略）現在はそういった制約をやむをえないこととして受け入れ，楽器ならではの味わいというプラスの側面にのみ目を向け，その制約を大前提として表現の範囲を規定している．しかし，人が求める表現は，楽器からの制約によって規定される範囲にとどまるはずがない．楽器デザインの考え方を「はじめに表現ありき」という枠組みへパラダイム転換することが必要である．つまり，作曲者や演奏者による音楽的表現の創造の幅を楽器の制約によって規定せず，逆に作曲者や演奏者が求める表現に応じて楽器を（再）デザインするべきである.

この考え方は楽器デザインだけではなく，楽曲の制作方法にも通じるものがある．これまでは，基準となるテンポに自分の演奏を合わせなければならないという制約から，楽器演奏経験の浅い人はテンポを合わせることが優先となり自分の演奏表現を犠牲にしたり，あるいは経験の豊富な人でも「ここの部分はもう少しリタルダンドを効かせて演奏したい」という要望に応えたりすることはできなかった．そして作られるものは，はじめからテンポが決まった 1 つの楽曲のみであった.

そこで本研究では，1 つの楽曲を制作する際に，各パートの奏者がテンポを自由に演奏した録音データを統合して，合奏を行っているような楽曲を生成できるシステムを提案する．録音データはそのままではテンポが異なるため，「マーク付け」によってテンポを同期させる．マーク付けを行った後は，システムのユーザが最も気に入った録音データを「指揮データ」として選択することで，他の録音データがこの指揮データに合わせて演奏をする．このシステムによって，奏者はこれまでのテンポに合わせることから解放され，自由に演奏を行うことができるので演奏表現は十分に発揮される．また，制作者は DAW 上での複雑な作業から解放される．これまでは，テンポが既に決まっている楽曲を 1 つしか作ることができなかったが，本システムでは録音データの数だけ指揮データを選ぶことができるため，テンポによって生まれる音楽表現について，楽曲の創造の幅を広げることが可能となっている.

2. 音楽表現

音楽表現は，演奏によって音楽に表情が付けられ，それが外面的，感性的に表れたものである．音楽表情については，意味の定義は人によって異なるが，表情付けに関する研究は 1980 年代に遡る[4][5]．1990 年代は，GTTM[6]や IRM[7]など認知的音楽理論の利用や，学習システム[8][9]，事例ベース推論によるアプローチ[10]も見られる．近年では，渡辺らが次のように述べている[11].

人間が実際にある種の楽器を演奏する際は，一定のテンポからは若干ずれが生じており，また音程・音量も微妙に変化する．こうしたずれはグルーヴ感とも呼ばれ，このグルーヴ感がリズムに表情を与え機械的でない自然な音楽を生み出している.

その上で，ドラムの打点時刻と音量を物理的特徴パラメータとして，ドラム演奏のグルーヴ感の解析を目的とした研究を行っている．奥平らは，既存の演奏や演奏モデルを利用することで，簡単な打拍操作で演奏表現感覚を味わったり，名演奏家を指揮したりするような感覚が味わえるシステムを提案，開発した[12][13]．指揮システム iFP では，演奏表情の決定は拍打時のテンポ，音量，拍内表情（時間，音量）の各要素に対して行われると述べている.

大島らの容易な MIDI シーケンスデータを作成するためのシステムの研究では，音楽表情を担う要素として，音符単位での速さと音の強弱が音楽表情を担う要素と述べている[14]．これらは，音楽表情のパラメータとして，「打点時刻」，「拍打時のテンポ」，「速さ」，「アゴーギク」，「音量」などを挙げている．これらは全て拍をどの速さで打つかに着目していると言え，「テンポ」とまとめることができる．そこで，本研究では音楽表情のパラメータを「テンポ」とした.

3. 録音データ

3.1 音楽演奏を扱うデータの選定と課題

音楽演奏を扱うデータは MIDI と波形をディジタル化した音響信号に大別される．本システムで扱う録音データは，後者の音響信号を指す．データの対象が MIDI の場合，後述の理由からシステムへの実装面では楽であるが，「生」の楽器演奏や歌手の演奏を用いることができないため，生成された音楽が機械的になってしまう．そこで，楽器や歌声を録音したものや，気に入った CD の楽曲などそのまま利用できるよう音響信号を対象とした.

しかし，データの対象を音響信号にすることでテンポを変更する際に課題が発生する．MIDI の場合には，発音時刻と音高をセットとした情報であるため，発音時刻の情報のみを変更すれば音高を維持したままテンポを変えることができる．一方で，音響信号の場合には，テンポを変えると波形の周波数が変化し，結果として音高が変わってしまうという課題が生じてしまう.

3.2 課題の解決方法

3.1 節で挙げた音響信号でのテンポ変化時の問題を解決するには，フェーズボコーダ[15][16][17][18]や，同期波形重畳法（SOLA: Synchronized Overlap-add method）[19][20]を用いたタイムストレッチと呼ばれる手法を用いる．タイムストレッチは，音響信号の音高をそのままに，テンポだけを変更することができる手法である．フェーズボコーダは信号の長さを変更するために，時間軸から周波数軸へのフーリエ変換による「分析」，周波数軸上でのリサンプリングによる「加工」，周波数軸から時間軸への逆フーリエ変換による「合成」という手順を踏むために処理の実行に時間

Vol.2014-HCI-157 No.17
Vol.2014-GN-91 No.17
Vol.2014-EC-31 No.17
2014/3/14

がかかる．SOLA はフーリエ変換を用いず時間軸で動作するため，処理の実行が速い．

図 1 に元の音響信号（上）のテンポを 2.0 倍（再生時間を 0.5 倍）に速くした例を示す．SOLA（中）は，テンポを 2.0 倍にするために，元の音響信号を一定の長さの区間に分割し（青・赤・青・…），これを 1 つおきに配置する（青・青・青）．これによって区間の境界では連続性が失われるが，全体としては，おおよその周波数を保ったままテンポを 2.0 倍にすることができる．単純なリサンプリング（下）は，周波数も 2.0 倍となるため，元の音響信号と比べ 1 オクターブ高くなってしまう．テンポを元の音響信号から速くする場合には，区間の大きさを変えることによって，自由なテンポに設定することができる．

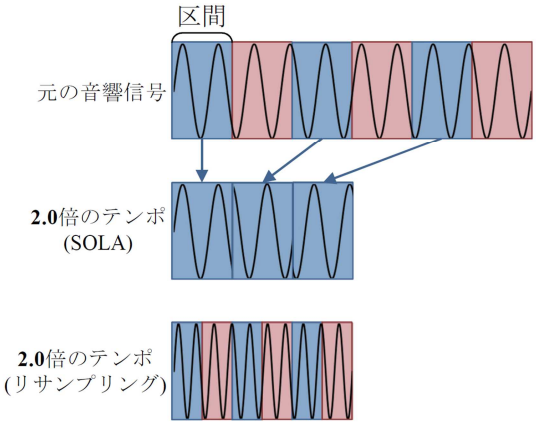

図 1　テンポを 2.0 倍にした際の音響信号

図 2 に元の音響信号（上）のテンポを 0.5 倍（再生時間を 2.0 倍）に遅くした例を示す．SOLA（中）は，テンポを 0.5 倍にするために，元の音響信号に対して，半区間ずつずらしながら区間を設定していく．これを半区間のずれをなくし，1 つずつ配置することで，全体として，おおよその周波数を保ったままテンポを 0.5 倍にすることができる．単純なリサンプリング（下）は，周波数も 0.5 倍となるため，元の音響信号と比べ 1 オクターブ低くなってしまう．テンポを元の音響信号から遅くする場合には，ずらす量を変えることでテンポを自由に設定することができる．

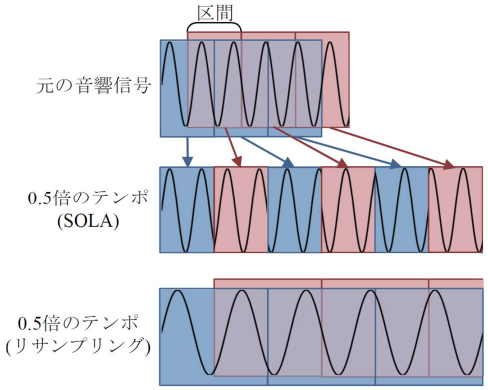

図 2　テンポを 0.5 倍にした際の音響信号

4. 提案手法

システムのインタフェースを図 3 に示す．録音データを読み込ませるための「トラック」，トラックを追加するための「トラック追加ボタン」，楽曲を生成して再生するための「再生ボタン」を実装している．システムで楽曲を作る際の手順は次の通りである．

(1)トラック追加ボタンで録音データを新しいトラックに読み込ませる（読み込みが成功すると録音データのスペクトログラムが作成される）

(2) (1)を繰り返して，全ての録音データを各トラックに読み込ませる

(3)テンポを合わせるための「マーク付け」を各トラックに対して全て行う

(4)楽曲の基準となるテンポを持つ録音データを「指揮データ」として選択する

(5)再生ボタンを押すことで，マーク区間内のテンポが指揮データに合わせて再生し，音楽を生成・再生する

図 3　システムのインタフェース

4.1 マーク付けとマークの削除・移動

「マーク付け」は複数の録音データのテンポを合わせるために必要な作業である．テンポを合わせて聞きたい箇所に番号で振り分けたマークを左クリックして付けることができる（図 4 上）．

また録音データを再生しているときに，スペースキーを押すことで，再生位置に新しいマークを付けることもできる．マークを付ける場所は拍や小節の頭のようなリズムが作り出される位置をはじめとして，ユーザの任意の位置に付ける．マークを付けた位置がユーザの意図していた箇所かどうかをフィードバックするために，その位置から 5 秒間録音データが再生されるようになっている．マークを誤って付けてしまった場合には，消したいマークの上で右クリックをすることで「マークの削除」をすることができる（図 4 下）．

Vol.2014-HCI-157 No.17
Vol.2014-GN-91 No.17
Vol.2014-EC-31 No.17
2014/3/14

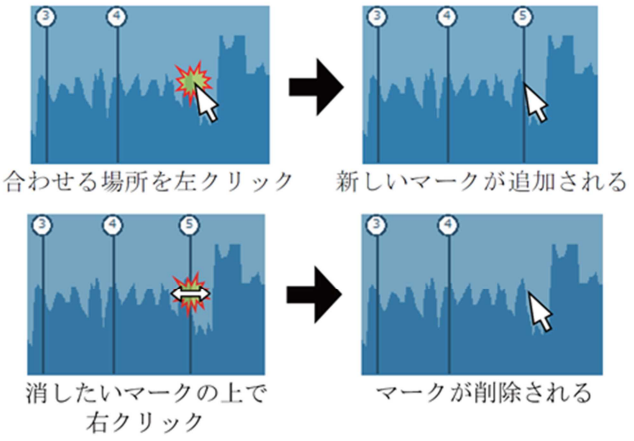

図 4　マーク付けと削除
（上：マーク付の手順　下：マーク削除の手順）

また既に存在するマークの位置を修正する際には，マークを左右にドラッグさせて移動させることでマーク位置の微調整が可能である．これが「マークの移動」である（図 5 上）．マーク付けやマークの削除でマークとマークの間に新しいマークを付けたり，削除したりした際には，左のマークから小さい順に，自動でマークの番号が振り直されるようになっている．マークの移動の場合も同様で，移動しようとしたマークが，前後のマーク位置を超えてしまった場合には，番号の振り直しが行われる（図 5 下）．

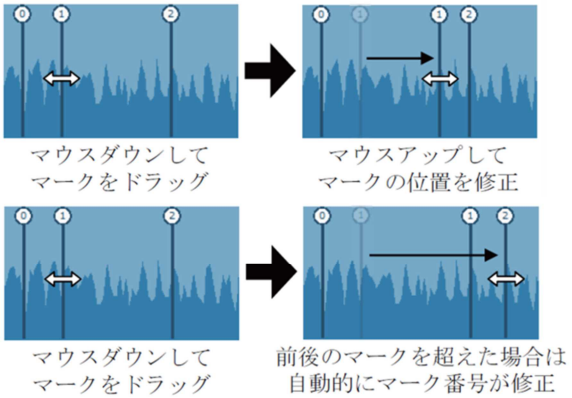

図 5　マーク移動と番号修正
（上：マークの移動　下：マーク番号の振り直し）

4.2　指揮データと録音データ

ユーザは，複数の録音データを各々のトラックに読み込ませ，マーク付けを行った後は，テンポの基準となるトラックの選択を行う．ユーザは適切であったり，好みであったりする録音データが読み込まれたトラックを 1 つ「指揮データ」としてトラックの左部をクリックする．指揮データのトラックは左部に指揮棒マークが付く．指揮データ以外の録音データが読み込まれたトラックは，元の演奏からテンポを変更し，指揮データのマークのタイミングに合わせて演奏を再生することになる．指揮データの候補数は，録音データの数と等しい．言わば，録音データの数だけ指揮者がいることになる．

例えば，図 3 は録音データ（トラック）が 2 つであるので，指揮データの候補は 1 番目の青色のトラックと，2 番目の緑色のトラックの 2 つがある（図 4 は，指揮データは指揮棒マークが付いている 2 番目の緑色のトラックであり，マーク付けは行っていない状態）．

4.3　その他の機能

本システムが，他の録音データが指揮データのテンポに合わせて楽曲を生成・再生すること以外に持っている機能は次の通りである．

(1)トラックに読み込んだ録音データを再生するプレビュー機能

(2)楽曲の再生位置を制御するシークバーと音量を制御する音量調整機能

(3)読み込んだ録音データやマーク付け位置をプロジェクトとして開く・保存する機能システム構成

5.　システム構成

本システムは，トラックへの録音データの読み込みや，マーク付けなどを行う「インタフェース部」と，録音データの音響信号に対して 3 章で紹介したタイムストレッチを行う「音響信号処理部」の 2 つの実行ファイルに分かれている（図 6）．インタフェース部の実行ファイルが裏側で音響信号処理部の実行ファイルを呼び出し，録音データを読み込ませる．そして，音響信号処理部がその録音データを加工して出力し，インタフェース部が加工されたファイルを受け取りシステムに反映させる形をとる．

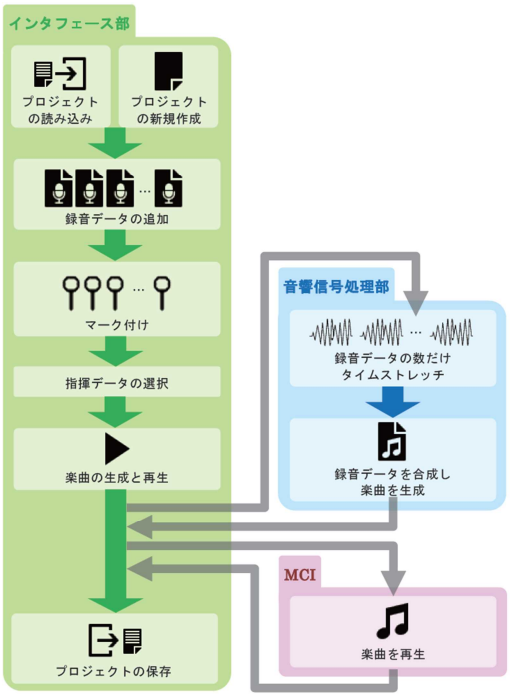

図 6　システム構成図

Vol.2014-HCI-157 No.17
Vol.2014-GN-91 No.17
Vol.2014-EC-31 No.17
2014/3/14

5.1 インタフェース部

グラフィカルユーザインタフェースによって，ユーザに本研究の提案手法を提供する部分である．直接音響信号の処理は行わない．実装は HSP（Hot Soup Processor）で行った．使用する録音データのトラックへの読み込み，マーク付け，録音データや生成した楽曲の再生，プロジェクトの読み込みと保存をユーザに提供する．音の再生には MCI（Windows Media Control Interface）を用いる．

5.2 音響信号処理部

コンソールアプリケーションであり，ユーザが直接使用するものではない．実装は 3 章で述べた SoX を本システム用に，スペクトログラムの作成機能，タイムストレッチ機能，同時再生機能を残したものである．インタフェース部に呼び出されて起動する．

5.3 テンポを合わせるための処理

テンポを合わせるために，以下の式 6.2 を用いる．n 番目のトラックの k 番目の音響信号に対するマーク位置（秒）を $M_{n,k}$ とおく．各マーク区間（秒）を $I_{n,k}$ とおくと，

$$I_{n,k} = M_{n,k} - M_{n,k-1} \quad (6.1)$$

と表せる．ただし，最初のマーク（$k = 0$）のときは $M_{n,k-1}$ を 0, 最後のマークのときは $M_{n,}$を全体の録音データの長さとする．指揮データのトラックを a とし，a のテンポを，全てのマーク区間において 1:0 とすると，指揮データ以外のトラックにおいて，元のテンポとの比率 $T_{n,k}$ は，

$$Tn,k = In,k / Ia,k \quad (n \neq a) \quad (6.2)$$

と表せる. 各変数とシステムとの対応は, 図 7 に示した. インタフェース部のマーク付けの位置を $M_{n,k}$ として, 式 6.2 より，$T_{n,k}$ を算出しパラメータとして，インタフェース部から音響信号処理部を呼び出せば，タイムストレッチの調整を行うことができる．

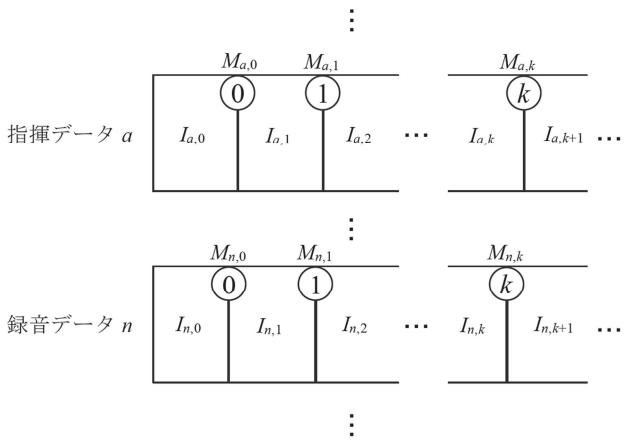

図 7　各変数とシステム上の対応

6. 制作例と考察

自由なテンポで演奏を行ってもらった録音データを用意し，第一著者が本システムを使用して楽曲を作成した際のシステムの使用感と考察，生成されてできた音楽に対する考察，マーク付けにかかる時間をまとめる．

6.1 制作例 1 — 器楽曲

6.1.1 準備

楽器 2 つによる楽曲の制作を行った．楽曲として，クリスティアン・ペツォールト（Christian Petzold，1677-1733）作曲による《メヌエットト長調》を用いた（図 8）．この楽曲を用いた理由は，第 1 に曲の長さが 1-2 分と，システムの試用には適切な長さであったこと，第 2 にクラシック音楽としては誰もが知っている有名な曲であるためシステムで生成された音楽が楽曲として成立しているか確かめやすいという点からである．

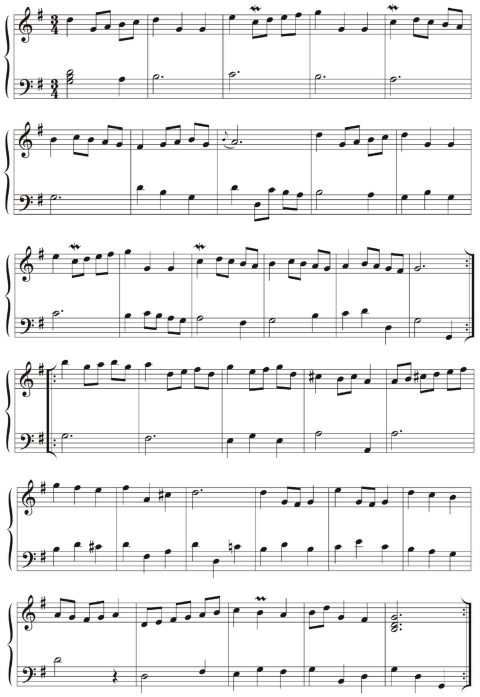

図 8　制作例 1 で用いた楽曲の譜例

録音データは，楽器を A = 415（ヘルツ）で調律し，旋律楽器としてリコーダーを，伴奏楽器としてヴィオラ・ダ・ガンバを使用した（図 8 の上段をリコーダーが，下段をヴィオラ・ダ・ガンバが演奏した）．楽器によって奏者は異なり，各自の自宅で演奏を録音したものを使用した．その際に，繰り返しは行わないこと，テンポ等を自由に設定して良いことを伝えた．録音は各自で行ってもらい，これらを WAVE 形式にしたものを録音データとして使用した．

録音データだけをみると，楽曲全体を通しての平均的な BPM は，リコーダーが約 150，ヴィオラ・ダ・ガンバが約 110 であり，リコーダーがヴィオラ・ダ・ガンバよりも全体的に速かった．そのため，リコーダーは全体的に快活な演奏表現で独自の細かな装飾が多かったのに対し，ヴィオラ･ダ･ガンバはしっとりとした落ち着いた演奏であった．録音データは 2 つとも最後にリタルダンドをかけていたが，リコーダーは最後から 2 小節前でかかっていたのに対し，

Vol.2014-HCI-157 No.17
Vol.2014-GN-91 No.17
Vol.2014-EC-31 No.17
2014/3/14

ヴィオラ･ダ･ガンバは最後から 1 小節前でかかっていた．システム上に，この 2 つのトラックを読み込ませてマーク付けを行った（図 9）．マークは各小節の頭と，曲の最後（32 小節目の 3 拍目の終わり）に付けたため，1 つの録音データに 33 個のマークを使用した．マーク付けにかかる時間は，全ての録音データをシステムに読み込み終わってから，第一著者が最終的にマーク付けの位置に対して満足し，楽曲を生成（再生）するまでの時間である．制作中にマーク付けの位置が正しいかを確認するために，楽曲をいったん生成し再生して確認する時間などは含めていない．

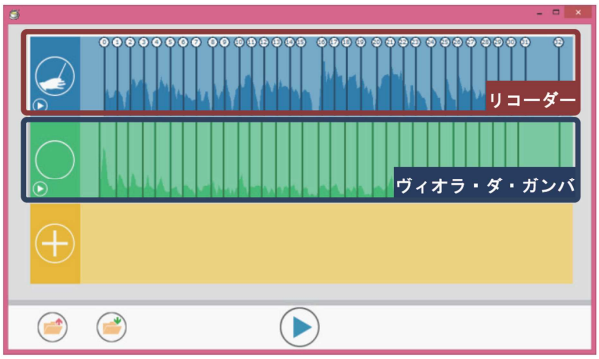

図 9　制作例 1 におけるシステム図

6.1.2 結果

リコーダーを指揮データとして選んだ場合には，ヴィオラ・ダ・ガンバが違和感なくリコーダーに合わせて合奏がなされているように聴くことができた．ヴィオラ・ダ・ガンバを指揮データとして選んだ場合には，リコーダーがヴィオラ･ダ･ガンバに合わせることはできていた．しかし，全体的にリコーダーにブツブツとノイズが入る箇所があり，リコーダーは速くなったり，遅くなったりと急にテンポの変化をしているところが見受けられ，リコーダーが指揮データの時と比べると違和感がある合奏曲となった．

マーク付けにかかった時間は，367.4 秒であった．今回は，2 つの録音データであったため，1 つの録音データにかかる平均時間は 367:4=2 = 183:7 秒である．また，マークの数は 33 個であるため，1 つのマークを付けるのにかかる平均時間は 183:7=33 = 5:6 秒である（表 1）．

表 1 制作例 1 でマーク付けにかかる時間（秒）

	全体	1 録音データあたり	1 マークあたり
時間（秒）	367.4	183.7	5.6

6.1.3 考察

指揮データをヴィオラ・ダ・ガンバにした場合に，リコーダーの演奏にノイズが発生する理由は，音響信号処理部で用いる SoX が備えるタイムストレッチ手法の限界であると考えられる．この症状を無くすには，別のタイムストレッチ手法を使う必要がある．この制作例から，テンポの遅い録音データをテンポの速い指揮データに合わせる場合は違和感がないが，テンポの速い録音データをテンポの遅い指揮データに合わせる場合は違和感が生じると言えると考えられる．しかし，他の制作例でも同様のことが生じるかを確かめるため，次の 6.2 節でテンポの速い演奏と遅い演奏から楽曲の制作を試みた．

6.2 制作例 2 ― 既存の録音から生成した楽曲

6.2.1 準備

6.1 節の器楽曲と同様であるが，今度は CD に収められた既存の演奏を使用する．それぞれの録音データはヴァイオリン 2 つ，チェロ 1 つ，チェンバロ 1 台により，既に合奏が行われている．制作例 1 は録音データが 1 つの楽器のみであった点が異なる．楽曲は，アントニオ・ヴィヴァルディ（Antonio Vivaldi，1678-1741）作曲による《トリオ・ソナタ“ ラ・フォリア”Op. 1 No. 12 RV 63 ニ短調》を用いる．この曲は変奏曲となっており，冒頭主題が提示された後，その後第 19 変奏まで行われる．今回はテンポが遅い主題（BMP は約 72）と，テンポが速い第 18 変奏（BPM は約 120）を録音データとして用いる．

この制作例は，6.1 節で生じた，「速いテンポの演奏データを遅いテンポの指揮データに合わせると違和感が生じる」ことを確かめるために行った．この曲は 3 拍子と感じられたため，拍子の 1 拍目にマーク付けを行った．マークの数は 1 つの録音データに対して 16 個であった．マーク付けの結果を図 10 に示す．

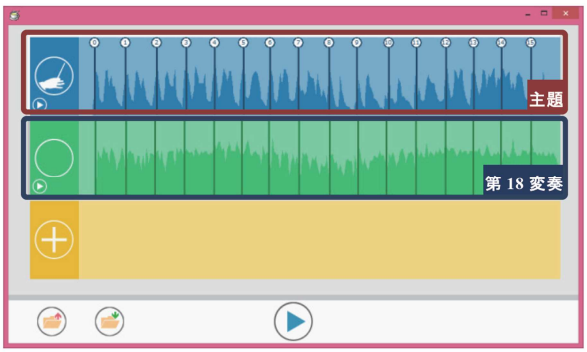

図 1o　制作例 2 におけるシステム図

7.2.2 結果

主題を指揮データとして選んだ場合には，第 18 変奏がテンポを合わせることはできていたが，「速いテンポの演奏データを遅いテンポの指揮データに合わせると違和感が生じる」原因となるノイズ問題，テンポの緩急の差が激しく感じられるという問題が大きく露呈した．第 18 変奏を指揮データとして選び，主題がテンポを合わせる場合には，1 つの楽曲として違和感なく聴くことができた．

マーク付けにかかった時間は，162.0 秒であった．6.1 節と同様に，1 つの録音データにかかるマーク付けの平均時間，1 つのマーク付けにかかる平均時間を次の表 2 にまとめた．

Vol.2014-HCI-157 No.17
Vol.2014-GN-91 No.17
Vol.2014-EC-31 No.17
2014/3/14

表 2 制作例 2 でマーク付けにかかる時間（秒）

	全体	1 録音データあたり	1 マークあたり
時間（秒）	162.0	81.0	5.1

表 3 制作例 2 でマーク付けにかかる時間（秒）

	全体	1 録音データあたり	1 マークあたり
時間（秒）	299.9	150.0	10.0

6.2.2 考察

この制作例からも，本システムを用いた場合には総じて，テンポの遅い録音データをテンポの速い指揮データに合わせる場合は違和感がないが，テンポの速い録音データをテンポの遅い指揮データに合わせる場合は違和感が生じると考えることができる．

楽曲を制作する上では，録音データが 1 つの楽器で録音したものであろうが，元々複数人で録音したものであろうが，演奏している楽曲のモチーフが同じであれば，生成される楽曲の整合性は保たれると考えられる．

6.3 制作例 3 — 歌曲

6.3.1 準備

録音データの中に歌声を含めた楽曲を制作する．CD に収められた既存の演奏を使用する．楽曲は，作曲者不明の《グリーンスリーブス》を用いる．録音データは 2 つ使用する．1 つは，歌声とギターとチェンバロによる演奏で BPM は約 125，もう 1 つは，フルートとハープによる演奏で BPM は約 120 である．元々は 4 小節毎にマーク付けを行ったが，それだけではアウフタクトの位置などでずれる箇所があった．そのためずれる位置にもマーク付けを行った．マークの数は 1 つの録音データに対して 15 個であった．マーク付けの結果を図 11 に示す．

図 11　制作例 3 におけるシステム図

6.3.2 結果

どちらを指揮データとしても，生成された楽曲に違和感はなかった．また，歌声を使用した場合であっても，歌詞の一部が切れるといったような問題は見受けられなかった．

マーク付けにかかった時間は，299.9 秒であった．1 つの録音データにかかるマーク付けの平均時間，1 つのマーク付けにかかる平均時間を次の表 3 にまとめた．

6.3.3 考察

今回は 2 つの録音データにテンポの差がないため，ノイズの問題や，テンポの緩急の差が激しく感じられるという問題が生じず，どちらを指揮データにしても生成される楽曲に違和感がなかったと考えられる．

6.4 制作例を通しての使用感と考察

マーク付けの量でテンポの同期の具合が良くなる・悪くなるといった違いはないと感じられた．マーク数を多くすればするほど，同期がしっかり行われるように感じられるが，実際は多くすると同期がうまくいかず，2 小節以上の大きなフレーズの方が自然な楽曲ができる傾向にあることがわかった．

本システムでは，遅いテンポを速くする場合には違和感がないため，難しいフレーズをゆっくり演奏しておいて，後で本システムを用いて技巧的な部分をヴィルトゥオーゾ（名人芸）のように聴かせることができるのではないかと考える．

7. 議論点と今後

7.1 生成された音楽の価値

生成された音楽は，本来ユーザが意図した音楽となるのか，聴きたいと思っていた演奏になるのか，あるいは，そもそも聴くに耐え得る演奏になっているのかなど多くの疑問点が生じる．そもそも音楽における美的価値観とは一定ではなく，人によって異なるため簡単には評価ができない．本システムによって，聴者が違和感なく楽曲を聴くことができるか，音楽がユーザの想像していたような音楽になったか，などを調査する必要があると考えられる．

7.2 楽曲を生成するシステムとしての今後

本システムでは，指揮データを途中で変更することができないが，実際には「ここの部分はこの録音データのテンポ，ここの部分はまた別の録音データのテンポに切り替えたい」という欲求が生じると考えられる．そのため指揮データを遷移できるようにも改良していきたいと考えている．

8. 関連研究

楽器の演奏者の立場をとると，自分の演奏に合わせて合奏を行う自動伴奏システムが挙げられる．このシステムは，実時間で楽器を演奏する人のテンポに合わせてコンピュータが伴奏を奏でるというものである．人間と機械が共演できる後藤らの仮想ジャズセッションシステム[28]や，人間と機械が音楽演奏の主導権をインタラクティブに交換でき

Vol.2014-HCI-157 No.17
Vol.2014-GN-91 No.17
Vol.2014-EC-31 No.17
2014/3/14

る Suzuki らのシステム[29]がある．楽器を演奏する技術が必要であるこれらの問題点を解決するために，音楽演奏経験のないユーザに対して支援を行う笠原らのシステム[30]もある．

既に用意された複数の楽曲を，波形編集などの音楽制作知識を必要としないで重ね合わせを行うシステムとしてはマッシュアップインタフェースが挙げられる[31][32][33]．ただし，これらは異なる楽曲の一部分を合わせて演奏するのみであって，同一曲を合わせたり，楽曲全体を合わせたりするためのシステムではない点が本システムと異なる．

同一楽曲を対象としたものでは，都築らの合唱制作支援インタフェースが挙げられる[34]．これは，Web 上で公開されている「1 つの曲を様々な歌手が歌った歌声」を複数重ねわせて作られる合唱作品を制作し鑑賞するためのシステムである．歌声の出現時刻を設定するだけで合唱が作成できる点が手軽であるが，このシステムでは本システムが可能としている旋律楽器と伴奏楽器の合奏といったように元となるデータが異なる旋律である場合には制作が難しい．

謝辞　本研究は、科学技術振興機構(JST)の戦略的創造研究推進事業(CREST)「コンテンツ共生社会のための類似度を可知化する情報環境の実現」の支援を受けています。

参考文献

1)　株式会社インターネット．Singer Song Writer, http://www.ssw.co.jp/products/ssw/，2014.
2)　株式会社コルグ．ドンカマチック，http://www.korg.co.jp/SoundMakeup/Museum/Doncamatic/，2014.
3)　西本一志，大島千佳．音楽と創造性，知能と情報（日本知能情報ファジィ学会誌），第 17 巻，第 2 号，pp.156-163，2005.
4)　Lars Fryden, Johan Sundberg．Performance Rules for Melodies. Origin, Functions, Purposes，International Computer Music Association, pp.221-224，1984.
5)　Manfred Clynes．A Composing Program Incorporating Microstructure, International Computer Music Association, pp.225-232, 1984.
6)　Fred Lerdahl, Ray Jackendoff. A generative theory of tonal music, MIT Press，1983.
7)　Eugene Narmour．The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model，the University of Chicago Press，1991.
8)　Gerhard Widmer．Learning expressive performance: The structure-level approach，Journal of New Music Research，Vol.25，No.2，pp.179-205，1996.
9)　石川修，片寄晴弘，井口征士．重回帰分析のイタレーションによる演奏ルールの抽出と解析, 情報処理学会論文誌，第 43 巻，第 2 号，pp.268-276，2002.
10)　Josep Lluis Arcos，Ramon Lopez de Mantaras，Xavier Serra. SaxEx: A Case-Based Reasoning System for Generating Expressive Musical Performances, Journal of New Music Research, Vol.27, No.3, 1998.
11)　渡邊哲朗，近山一．ドラム演奏のグルーヴ感の解析，情報処理学会研究報告，第 9 号，pp.27-32，2006.
12)　奥平啓太，片寄晴弘．演奏表情テンプレートを利用したピアノ演奏システム:sfp，情報処理学会論文誌，第 44 巻，第 11 号，pp.2728-2736，2003.
13)　奥平啓太, 片寄晴弘, 橋田光代. 音楽演奏インタフェース iFP : 演奏表情のリアルタイム操作とビジュアライゼーション，情報処理学会研究報告，第 82 号，pp.13-18，2003.
14)　大島千佳，西本一志，宮川洋平，白崎隆史．音楽表情を担う要素と音高の分割入力による容易な MIDI シーケンスデータ作成システム, 情報処理学会論文誌, 第 44 巻, 第 7 号, pp.1778-1790, 2003.
15)　J. L. Flanagan，R. M. Golden．Phase vocoder，Bell System Technical Journal，Vol.45，No.9，pp.1493-1509，1966.
16)　Michael R. Portnoff．Implementation of the Digital Phase Vocoder Using the Fast Fourier Transform，IEEE Trans. of Acoustics, Speech, and Signal Processing，Vol.ASSP-24，No.3，pp.243-248，1976.
17)　D. Griffin，J. Lim．Signal Estimation from Modified Short-Time Fourier Transform，IEEE Transactions on Acoustics, Speech and Signal Processing，Vol.32，No.2，pp.236-243，1984.
18)　Mark Dolson．The Phase Vocoder: A Tutorial, Computer Music Journal，Vol.10，No.4，1986.
19)　L. Rabiner, R. Shafer. Digital Processing of Speech Signals, 1978.
20)　David Malah．Time-domain algorithms for harmonic bandwidth reduction and time scaling of speech signals，IEEE Transactions on Acoustics, Speech, and Signal Processing，Vol.ASSP-27，No.2，pp.121-133，1979.
21)　A. de Cheveigne，H. Kawahara．YIN, a fundamental frequency estimator for speech and music, The Journal of the Acoustical Society of America，Vol.111，No.4，pp.1917-1930，2002.
22)　P. McLeod, G. Wyvill. A smarter way to find pitch, In Proceedings of the International Computer Music Conference，pp.138-141，2005.
23)　SoX - Sound eXchange, http://sox.sourceforge.net/Main/HomePage，2014.
24)　Olli Parviainen．SoundTouch Audio Processing Library, http://www.surina.net/soundtouch/，2014.
25)　Un4seen Developments - 2MIDI / BASS / MID2XM / MO3 / XM-EXE / XMPlay, http://www.un4seen.com/，2014.
26)　DIRAC Time Stretching Pitch Shifting, http://www.dspdimension.com/technologylicensing/dirac/，2014.
27)　Jazz Mack Smith，Chris Cannam．Rubber Band Audio Time Stretcher Library, http://breakfastquay.com/rubberband/，2014.
28)　後藤真孝，日高伊佐夫，松本英明，黒田洋介，村岡洋一．仮想ジャズセッションシステム: VirJa Session，情報処理学会論文誌，第 40 巻，第 4 号，pp.1910-1921，1999.
29)　Kenji Suzuki，Yoichiro Taki，Hisaki Konagaya，Pitoyo Hartono, Shuji Hashimoto．Machine istening for Autonomous Musical Performance Systems，Proc.2002 International Computer Music Conference，pp.61-64，2002.
30)　笠原俊一，三枝亮，橋本周司．連想型自己組織化マップを用いたリズム演奏支援システム，情報処理学会論文誌，第 48 巻，第 12 号，pp.3649-3657，2007.
31)　M. Davies，et al．AutoMashUpper: An Automatic Multi-Song Mashup System，Proc. ISMIR 2013，2013.
32)　N. Tokui．Massh!: AWeb-based Collective Music Mashup System, Proc. DIMEA 2008，2008.
33)　宮島靖．Music Mosaic Generator: 高精度時系列メタデータを利用した音楽リミックスシステム，WISS 2007 論文集，2007.
34)　都築圭太, 中野倫靖, 後藤真孝, 山田武志, 牧野昭二. Unisoner: 同一楽曲を歌った異なる歌声を重ね合わせる合唱制作支援インタフェース，WISS 2013 論文集，2013.