

演奏者の楽譜の休止区間における自動伴奏のテンポ制御

長野 亜美^{1,†1} 齋藤 康之^{1,a)} 中村 栄太² 嵯峨山 茂樹³

概要: 楽譜に従って、人の演奏に追従して自動的に伴奏パートを演奏するシステム Eurydice (ユリデイス) は、演奏の生成過程を確率モデルでモデル化することにより、速度変化や弾き間違いがあっても柔軟に追従できる自動伴奏システムである。Eurydice では演奏者による入力に基づいてテンポ制御するが、楽譜上で演奏者パートの休止区間では伴奏パートのテンポ制御ができない。本研究では、そのような区間で、鍵のタップ動作による伴奏のテンポ制御について検討した。まず、タップ動作を行う鍵を決定するために、鍵間の距離に関するミスタッチ率を評価実験で確認した。前処理として、Eurydice に使用する standard MIDI File (SMF) ヘタップ動作に必要な音符を挿入し、挿入した音符をタップすることでテンポ制御と曲の進行を行った。楽譜には存在しない音符であるため音は出力しない。これらを実装し、動作を確認した。

キーワード: 自動伴奏, テンポ制御, MIDI ファイル

Tempo control for automatic accompaniment while player rests in musical score

NAGANO AMI^{1,†1} SAITO YASUYUKI^{1,a)} NAKAMURA EITA² SAGAYAMA SHIGEKI³

Abstract: Automatic music accompaniment system synchronizes with human player's performance. Eurydice is one of such systems that model the generation process of the performance with a probabilistic model, which can therefore follow flexibly the performance in presence of tempo changes, mistakes, skips and repeats. As the tempo control is made by following the player's input, it is a problem while the player's part has a long rest in the score. In this study, we discuss tempo control of the accompaniment by key tapping. To determine a key for the tap operation, wrong key rate was experimentally evaluated across various distances between keys. Then, as a preprocessing, we inserted necessary notes for the tap operation into the standard MIDI file (SMF) for Eurydice, so that Eurydice controls tempo and proceeds forward when the player tapped the inserted notes. The inserted notes cause no sounds, because they do not exist in the original musical score. We implemented these new features and confirmed a successful operation.

Keywords: automatic accompaniment, tempo control, MIDI file

1. はじめに

楽譜に従って、人の演奏に追従して自動的に伴奏パート

を演奏するシステムを、ここでは自動伴奏システムと呼ぶ [1], [2]. 自動伴奏システムは、システム側が演奏者の指示に合わせることで成り立っている。

しかし、人の演奏は楽譜通りでなく、不確定なものである。たとえば、テンポの揺らぎや音高の間違い、余分な音の挿入、必要な音の欠落、音量の変化、弾き直しなどの要因が含まれる [3]. これらの要因を踏まえ、Eurydice (ユリデイス) [4] は演奏の生成過程を確率モデルで記述することにより、柔軟に追従することが可能である。このような技術は、ピアノの片手での練習やソロパートの練習の際に使

¹ 木更津工業高等専門学校 情報工学科
NIT,Kisarazu College, Kisarazu, Chiba 2921-0041, Japan
² 京都大学 大学院情報学研究科 知能情報学専攻
Kyoto University, Sakyo-ku, Kyoto 606-8501, Japan
³ 明治大学 総合数理学部 先端メディアサイエンス学科
Meiji University, Nakano-ku, Tokyo 164-8525, Japan
^{†1} 現在, 千葉大学
Presently with Chiba University
^{a)} saito@j.kisarazu.ac.jp

用できるほか、協奏曲や連弾、パートが複数ある曲で奏者数を低減し演奏を可能にする効果も期待される。また、ログの取得（MIDI 録音）が可能であり、その取得したデータを利用して新たな演奏形態や新しい作曲方法などの可能性を与えることができる。

このシステムに残されている問題として、(1) 演奏者が休みの間、システムへテンポ指示ができないことや、(2) 伴奏演奏の表現が必ずしも演奏者の意図に沿わないことがある、という点が挙げられる。そこで本研究では、鍵のタップ動作による伴奏のテンポ制御について検討した。

2. 自動伴奏システム Eurydice

2.1 システム概要

Eurydice は、楽譜を参照しつつ、演奏者の演奏に追従する自動伴奏システムである。演奏者のテンポに合わせて伴奏を演奏する。また、弾き間違い（ミスタッチによる余分な音の挿入、音の脱落、音高誤り）や弾き飛ばし（反復練習や小節の省略）があったとしても演奏者がどこを演奏しているのかを瞬時に認識し、演奏者に合わせてリアルタイムで伴奏を演奏する。その他、音量変化も感知して伴奏の音量を自動的に調節する。Eurydice の使用例を以下に示す。

1) ピアノの片手の練習

Eurydice が左手パートを担当し、演奏者は右手のメロディパートだけに集中して練習できる。左右の担当を入れ替えることで、演奏者は伴奏パートを練習できる。

2) ピアノ連弾の練習

Eurydice がセコンド（低音パート）を担当することで、演奏者がプリモ（高音パート）を練習できる。もちろん役割を入れ替えることも可能である。

3) ピアノ協奏曲、ピアノアンサンブル

Eurydice がオーケストラを担当し、演奏者がピアノ独奏を演奏できる。ピアノ協奏曲を演奏するためには、オーケストラを編曲したピアノ伴奏と演奏することがあるが、Eurydice を使うことで、フルオーケストラをバックにピアノ協奏曲を楽しめる。

2.2 ソフトウェア概要

Eurydice は楽譜追跡と伴奏生成から成る（図 1）。楽譜追跡では人間の演奏を隠れマルコフモデル（hidden Markov model, HMM）で表現し、演奏位置は Viterbi アルゴリズムで推定する [7]。伴奏生成ではテンポ推定を行い、次の音符のオンセット時刻を予測し、伴奏の MIDI 再生を同期して行う。

楽譜情報は SMF format 1 を使用し、MIDI 信号を入力することで動作する。そのため、電子ピアノや MIDI ドラムなどの MIDI 対応機器で使用できる。なお、演奏者が次の音符を演奏するのを待ち続ける「waiting モード」と、任意のジャンプを許容しつつも演奏を待たずに楽曲を順次進

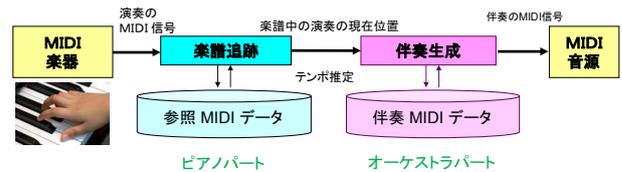


図 1 楽譜追跡と伴奏生成の関係

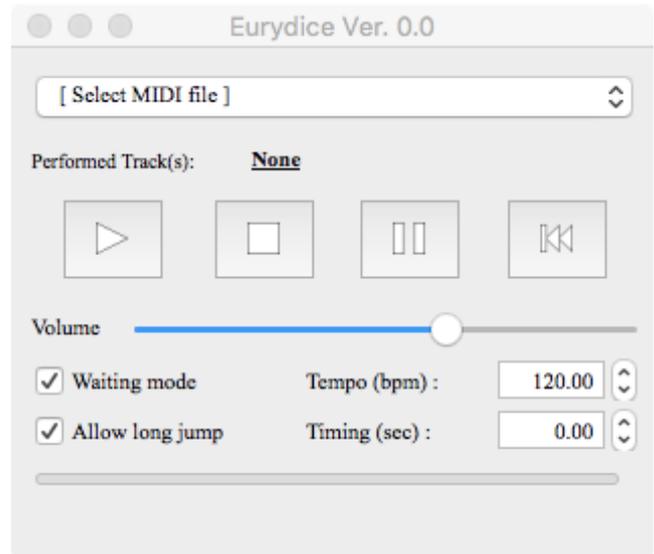


図 2 Eurydice メインウィンドウ

行する「non-waiting モード」を選択できる。

クロスプラットフォームの Qt [8] を用いて実装しており、現在、Eurydice は Windows 版と Mac OS X 版がリリースされている。Eurydice のメインウィンドウを図 2 に示す。

3. 演奏者休止区間におけるテンポ制御に関する問題点

楽譜上で演奏者が休みで伴奏が主体の場合の楽譜例を図 3 に示す。以後、このような楽譜上の区間を「演奏者休止区間」と呼ぶ。現在の Eurydice では、演奏者休止区間においては、その直前のテンポを継承して伴奏を演奏する。そのため、急激なテンポ変化への対応ができない。これに対し、演奏者休止区間の間に Eurydice にテンポを指示しようとする、Eurydice の強力な楽譜追跡能力によって不適切な伴奏が演奏されることになる。これは、テンポを制御する目的で演奏しても、演奏者休止区間では対応する音符が存在せず、その演奏音を楽譜中から探索し、その音が出現する楽譜位置または音高誤りとして意図しない楽譜位置にジャンプするからである。そのため、現状のままでは演奏者休止区間において楽器を用いた伴奏のテンポ制御はできない。

これらは、伴奏演奏の表現が必ずしも演奏者の意図に沿わなくなる原因となる。そこで、本研究ではこれらの問題



図 3 楽譜上で演奏者が休止時（演奏者休止区間）の楽譜例

を解決する。

4. 演奏者休止区間でのテンポ制御方法

4.1 頭部動作によるテンポ制御

演奏に直接関係しない、頭部動作によるテンポ制御方法を考察する。ジェスチャー認識による自動譜めくりの研究では、頷き動作による制御を行っている [9]。この手法を取り入れ、1回の頷きを1拍とし、4回の頷きで4拍1小節 ($\frac{4}{4}$ 拍子) として扱うことが考えられる。

しかし、この手法ではテンポの速い曲の指示が困難である。また、現状では頷き動作を正確に捉えるために意図的な遅延を入れているため、リアルタイム性が損なわれる。

4.2 タップ動作によるテンポ制御

鍵盤のタップ動作によりテンポを制御する方法を考察する。演奏に使用しないキーに順次進行するという役割を与え、打鍵しても音を出さないようにできれば、そのキーを打鍵することは、いわば指揮者の指揮の振りに対応づけられる。タップ動作は演奏時の動作と差異がないため、演奏者が行う動作と共に、順次進行やテンポ推定の制御は今まで通り行えるという利点がある。そこで本研究では、テンポ制御方法はキーのタップ動作によるものを採用する。以後、テンポ制御に用いる、上記の「演奏に使用しないキー」を「タップキー」と呼ぶことにする。

指示するテンポが一定の場合、演奏者休止区間にタップ動作を継続することは演奏者の負担となる。この問題を解決するには、ある程度タップを行って、テンポが一定になった場合にタップ動作を行わなくても直前のテンポを引き継ぐようにすればよいと考えられる。

テンポ制御の処理の流れをまとめると以下の項目に分けられる。

- 1) タップキーが押されたか判定を行う。
- 2) タップキーの場合、音を出さない。
- 3) 順次進行及びテンポ推定を行う。

項目 2 の「タップキーの場合、音を出さない」という部分では演奏しているピアノ音は OFF*1、そして MIDI

*1 アコースティックピアノにおいても、たとえばヤマハのサイレントピアノシリーズを用いたり、ピアノ消音ユニットをアップライトピアノに追加装着すれば演奏音は消せるが [10]、現状の Eurydice では入力 MIDI 信号であることを考慮すると、本研

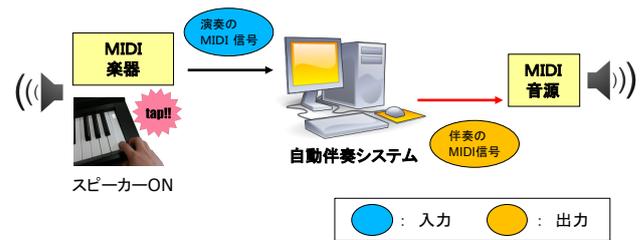


図 4 MIDI Through をしない場合の自動伴奏システムの入出力

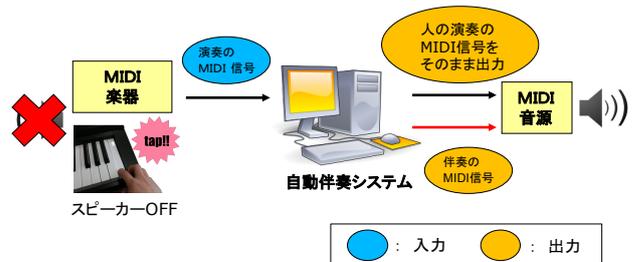


図 5 タップ動作によるテンポ制御を用いる場合における自動伴奏システムの入出力（通常演奏時）

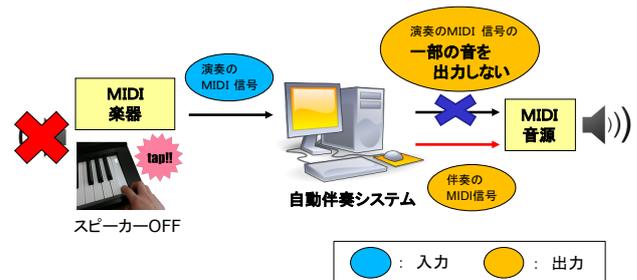


図 6 タップ動作によるテンポ制御を用いる場合における自動伴奏システムの入出力（タップ動作時）

Through にしておき、タップ動作のときはタップキーの音を発音しないようベロシティを 0 に設定を行う。MIDI Through をしない場合の自動伴奏システムの入出力を図 4 に示す。また、タップ動作によるテンポ制御を用いる場合における通常演奏時の自動伴奏システムの入出力を図 5 に、タップ動作時を図 6 に示す。

項目 1 と項目 2 の処理を実装した。項目 3 の順次進行とテンポ推定は、従来の制御を用いることができる。

5. タップキーの決定方法の検討

キーのタップ動作で行うテンポ制御において、テンポを指示するために用いるタップキーは、曲中で用いない音高のキーとする。演奏者が任意でタップキーを決定してもよいのだが、わかりやすく、ミスタッチとなる頻度が低くなるキーであることが求められる。このことを鑑みると、タップキーとして用いることのできるキー（タップキーの候補）のうちで一番高い音か低い音とすればよいと考えられる。また、実装も容易である。しかし、この方法では、演奏においては電子ピアノを用いることが最適である。

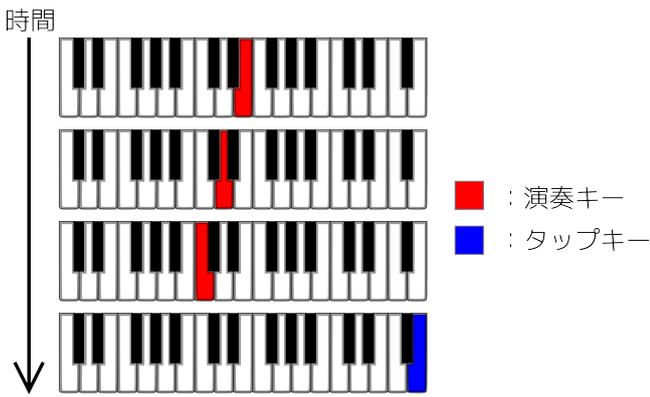


図 7 運指に無理のあるタップキーの例

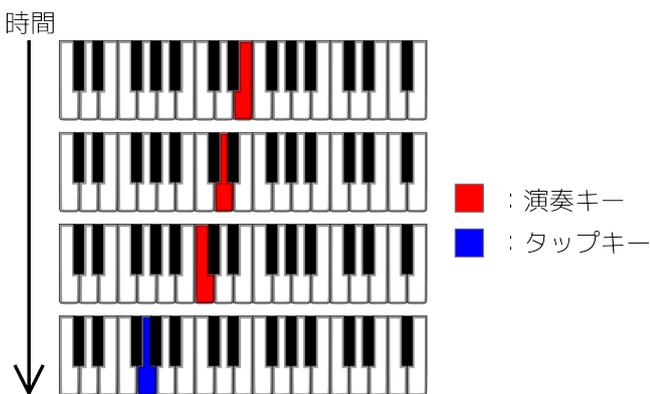


図 8 運指に無理のないタップキーの例

奏キーからタップキーまで手の移動距離が遠すぎることで、演奏の流れに沿っていない動作、無理のある運指(図7)になる可能性がある、という欠点がある。そのため、ミスタッチの可能性が増加する可能性がある。

これらの問題を解決するには、まず、タップキーの候補のうち、最後に演奏した位置から近いものを自動的に導出しておく。次に、ピアノ運指の自動決定[11]を適用し、タップキーの候補のうち、無理のない運指(図8)によるキーを選択すれば、演奏の流れを妨げずに演奏の流れに沿ったタップキーを決定できると考えられる。

6. 利き手とリズムノリに関する関係

左右どちらの手の方がリズムを刻みやすいか(リズムにのるか)について、利き手の方が適していると仮説を立て、以下のような指示を被験者に与えて評価実験を行った。

- 片手で、自由なテンポで一定のリズムを刻む。
- 壁や机、自分の部位など、どこをタップしても構わない。

被験者には「片手で」としか伝えてないため、被験者が一瞬のうちにどちらの手を使い出すかという点と利き手の関係について注目した。

被験者の利き手と実際に用いた手の結果を表1に示す。

表 1 利き手とリズムの刻みに用いた手

利き手(被験者数)	実際にリズムを刻んだ手	
	右手	左手
右手(17名)	17名	0名
左手(3名)	0名	3名



図 9 実験用楽譜(インターバルなし)



図 10 実験用楽譜(インターバルあり)

全員が利き手でリズムを刻んでおり、仮説が裏付けられる結果が得られた。このことから、利き手で押しやすいタップキーを用いることが適切といえる。したがって、たとえば、右利きの人に対しては、中央のC4よりも右側であることが望ましい。

7. キー間の距離に関するミスタッチ率

演奏キーからタップ動作を行うキーまでの手の移動距離が遠すぎることで、ミスタッチの可能性が増加するかの検証を行うため、全音単位のキー間の距離に関するミスタッチ率について評価実験を行った。

7.1 実験方法

演奏を開始するキー(基準キー)を設定し、その基準キーと一番近い距離から徐々に離れたキーを交互にタップする。そして、逆にその基準キーと一番離れた距離から徐々に近いキーを交互にタップする。テンポはBPM=100固定で、白鍵のみを使用する。休符が一切ない「インターバルなし」と4拍中2拍休符の「インターバルあり」を各1回ずつ片手で行う。「インターバルなし」の楽譜の一部を図9に、「インターバルあり」の楽譜の一部を図10に示す。なお、対象のキーを打鍵したとしても、他のキーを余分に打鍵していた場合はミスとして扱うこととする。

被験者はピアノ経験者3名、ピアノ未経験者10名の合計13名であった。今回被験者が全員右利きであったため、6章の結果を鑑みて、右手のみ実験を行った。被験者には説明とトレーニングを行ってから実験を開始する。実験方法を図11に、音階差とキーの対応を図12に示す。

7.2 実験結果

ミスタッチの様子を図13に、基準キーと対象キーの距

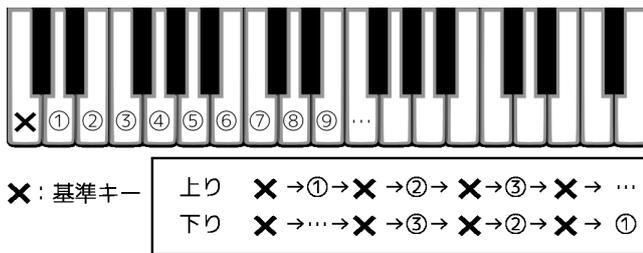


図 11 キー間の距離に関するミスタッチ率の実験方法

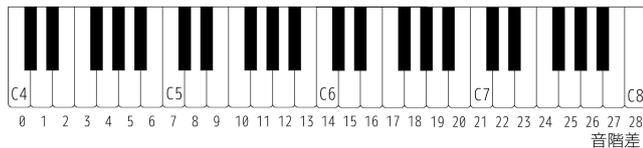


図 12 音階差とキーの対応. これは利き手が右手の被験者に対する実験で用いた範囲を示しており, 中央の C4 より右側の鍵だけを考慮している.

離 (音高差) に対するミスタッチ率を図 14 に示す. 音階差 15 以下はミスタッチ率 0 % の部分が多いことから, この部分のミスタッチは演奏者の不注意から起こっていると考えられる. 音階差 15 以降からミスが増加していることがわかる. その中で一番離れているキーのミスタッチ率が低い. これは, 距離は離れているものの, キーボードの端の部分であるため狙いが定めやすいためであると考えられる. 以上の結果から, 現段階ではキーボードの端をタップキーと定める.

8. 演奏者休止区間におけるテンポ制御の実装

Eurydice は演奏者の演奏を追従するために, 演奏者の演奏する MIDI データを参照している. 演奏者休止区間は, 実際の楽譜上では休符であるが, MIDI データでは休符という直接的なデータは存在しない. そのため, タップキーを打鍵したという特定条件下で処理しようとしても, 実際の楽譜上での休符区間は, そのタップ動作に対応づける音符 (休符) データがないので, 演奏位置推定を行えず, 演奏者の意図した通りの順次進行が行えない. そこで, 音符データのない休符小節に音符を挿入する. ここでは, Eurydice で使用する standard MIDI File (SMF) に前処理を施すように実装した.

SMF の解析には, C++ ライブラリの Midifile [12] を用いた. 実装項目を以下に示す.

【前処理】

- 1) タップ動作で使用するタップキーを決定する.
- 2) 1 小節以上の休止区間に音符を挿入する.
音高 : タップキー
音の長さ : 単位音符長
(オリジナル SMF からタップ動作 SMF を生成)

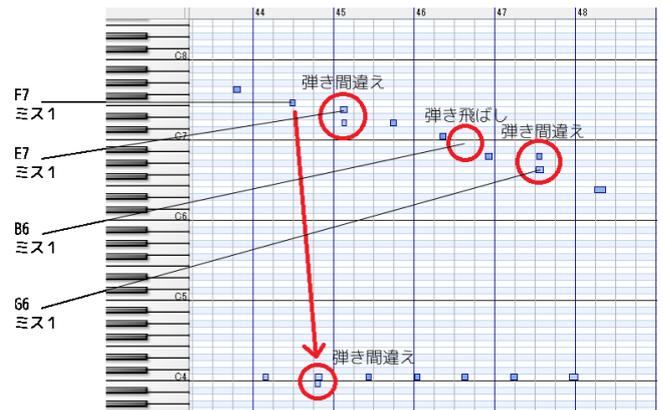


図 13 ミスタッチの様子

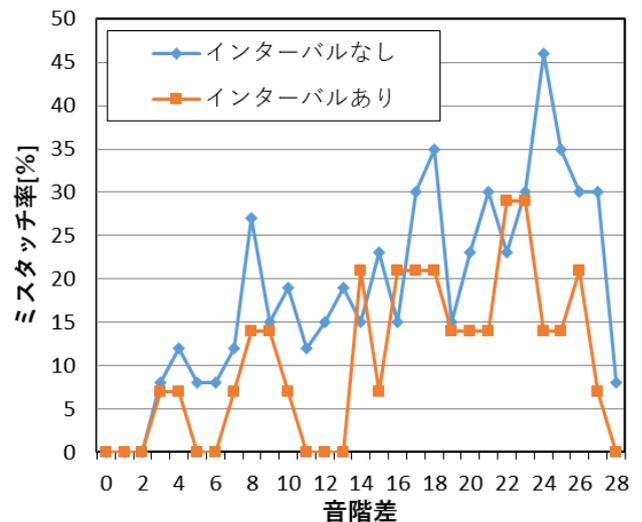


図 14 基準キーと対象キーの距離 (音高差) に対するミスタッチ率

【Eurydice での処理】

- 1) タップ動作 SMF を読み込む.
- 2) タップキーが押されたか判定を行う.
- 3) タップキーの場合, 音を出力しない.
- 4) 順次進行及びテンポ推定を行う.

ここで, 単位音符長とは, 拍子の分母に相当する音符の長さである. たとえば, $\frac{4}{4}$ 拍子の場合には 4 分音符, $\frac{6}{8}$ 拍子の場合には 8 分音符である.

休符が 1 小節以上ある SMF を譜面化したものを図 15 に, その SMF に対して前処理を行った結果を図 16 に示す. 楽譜上で演奏者休止区間において自動伴奏のテンポをタップ動作で制御できた. また, Eurydice を non-waiting モードに設定しておくことで, 演奏者はタップし続けなくてよいことを確認した.

9. まとめ

本研究では, 楽譜上で演奏者が休止時 (演奏者休止区間) のテンポ制御方法を考察し, 自動伴奏システム Eurydice への実装を進めた. 演奏に使用しないキーのうちでテンポ



図 15 オリジナル SMF

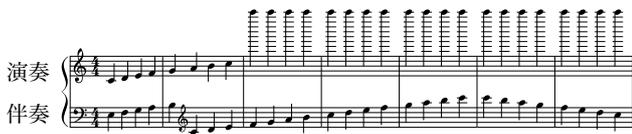


図 16 前処理を行った結果 (タップ動作用 SMF)

指示のために用いる「タップキー」の選択のための予備実験を実施し、基準キーからの音高差に対するミスタッチ率を求めた。その結果から今回は最右端のキーをタップキーと定め、standard MIDI file (SMF) を解析して 1 小節以上の演奏者休止区間を自動的に検出し、タップキーの音高を埋め込んだ。そして、Eurydice にタップキーによるテンポ制御の仕組みを実装し、動作を確認した。

今後、演奏者休止区間でのテンポ制御方法の実装を改良する。現在、SMF への前処理を別システムで実装しているが、サブシステムとして Eurydice 内に組み込む。そして、5 章で述べたように、タップキーの候補のうち、無理のない運指によるキーを選択する方法について検討していく。

謝辞 本研究の一部は、日本学術振興会の科学研究費補助金 26240025 による。

参考文献

- [1] Dannenberg, R.B.: An on-line algorithm for real-time accompaniment, *Proc. ICMC*, pp.193-198 (1984).
- [2] Vercoe, B.: The synthetic performer in the context of live performance, *Proc. ICMC*, pp.199-200 (1984).
- [3] 武田 晴登, 西本 卓也, 嵯峨山 茂樹: HMM による MIDI 演奏の楽譜追跡と自動伴奏, 情報処理学会研究報告, MUS, pp.109-116 (2006).
- [4] 東京大学 大学院 情報理工学系研究科 システム情報学専攻 / 創造情報学専攻工学部 計数工学科 システム情報学 第一研究室 [嵯峨山/守谷/亀岡/(小野) 研究室]: Eurydice (自動伴奏システム), 入手先 <<http://hil.t.u-tokyo.ac.jp/software/Eurydice/>>
- [5] Goebel W.: Melody lead in piano performance: Expressive device or artifact?, *J. Acoustical Society of America*, Vol.110, No.1, pp.563-72 (2001).
- [6] 武田 晴登, 西本 卓也, 嵯峨山 茂樹: テンポ曲線と隠れマルコフモデルを用いた多声音楽 MIDI 演奏のリズムとテンポの同時推定, 処理学会論文誌, Vol.48, No.1, pp.237-247 (2007).
- [7] 中村 栄太, 武田 晴登, 山本 龍一, 齋藤 康之, 酒向 慎司, 嵯峨山 茂樹: 任意箇所への弾き直し・弾き飛ばしを含む演奏に追従可能な楽譜追跡と自動伴奏, 情報処理学会論文誌, Vol.54, No.4, pp.1338-1349 (2013).
- [8] Qt 公式ページ, 入手先 <<http://qt-project.org/>>
- [9] Jibiki, H., Shimizu, T., Saito, Y., Nakamura E. and Sagayama, S.: A study of automatic page turning of musical scores by detecting player's nods, *Proc. ICAI PB1-08*, pp.272-275 (2015).
- [10] ヤマハ: サイレントピアノについて,

- [11] 入手先 <<http://jp.yamaha.com/products/musical-instruments/keyboards/about/silent/>>
東京大学 嵯峨山研究室: 隠れマルコフモデル (HMM) に基づくピアノ運指の自動決定, 入手先 <<http://hil.t.u-tokyo.ac.jp/research/introduction/PianoFingering/japanese.html>>
- [12] Midifile 公式ページ: C++ library for parsing standard MIDI files, 入手先 <<http://midifile.sapp.org/>>