

# 独奏認識誤りに頑健な音響入力伴奏システム

足立亜里紗<sup>†1</sup> 堀内靖雄<sup>†1</sup> 黒岩眞吾<sup>†1</sup>

**概要:** 伴奏システムとは人間の伴奏者のように独奏者に協調した演奏を計算機で実現しようとするシステムである。我々はより人間らしい伴奏システムを実現するために伴奏制御のモデル化を行ってきた。しかしながら、独奏が音響入力の場合、計算機の認識誤りによってモデル適用時に用いる演奏履歴が正しく取得できなかったとき、不適切な伴奏出力を行ってしまうことがあった。そこで本研究では独奏者の演奏位置の推定誤りが存在しても妥当なテンポ決定を可能にする頑健な伴奏制御手法を提案することを目的とする。人間の演奏者による評価実験を行った結果、その有効性が示された。

**キーワード:** 伴奏システム, 伴奏制御モデル, 音響入力, 相互作用, 演奏認識誤り

## 1. はじめに

音楽情報処理の一分野として研究されている伴奏システムとは独奏パートと伴奏パートの楽譜情報が事前に計算機に与えられている状態で、人間の独奏者の演奏に協調した伴奏を人間ではなく計算機で行うシステムである。伴奏システムの登場以前は人間と計算機の合奏において計算機の演奏をリアルタイムで動的に変化させることができなかったため、人間の独奏者が計算機に合わせて演奏する必要があった。これに対し、伴奏システムは人間の独奏者の演奏に合わせた伴奏を行うシステムである[1,2]。

人間の独奏者の演奏にはテンポの揺らぎがあるため、一定のテンポで伴奏を行うのでは独奏者との合奏は成立しない。また、人間の独奏者の演奏時のテンポ変化は演奏表現の一部なので、独奏者が予め録音されたテンポ一定の伴奏に合わせることは、その表現を制限することになる。そこで伴奏システムは一定のテンポで伴奏を出力するのではなく、人間の伴奏者と同じように独奏者の演奏を聴き取って解釈し、それに同期するように伴奏のテンポを変化させる必要がある。そこで我々はこれまで人間同士の合奏を分析し、合奏における伴奏者の演奏制御のモデル化を行ってきた[3]。その際、人間の演奏者は拍を基準として演奏を行っているとは仮定し、拍ごとの独奏者と伴奏者のふるまいを分析し、人間の伴奏者を模倣するモデルを構築した[4,5]。

本研究で開発している伴奏システムの処理の概要を図1に示す。まず伴奏システムは独奏者の演奏を逐次取得する。独奏者の演奏は音響情報として楽器用の集音マイクから取得される。次に取得した音響情報を解析し、事前に与えられている独奏の楽譜情報と照合し、現在独奏者が演奏している楽譜上の位置を推定する(独奏認識)。独奏者の楽譜上の演奏位置を推定したら、その位置情報を利用して、独奏者の演奏に対して適切と考えられる伴奏のテンポを計算し(伴奏制御)、それによって事前に与えられている伴奏の楽譜を演奏する。伴奏の演奏には MIDI (Musical Instrument

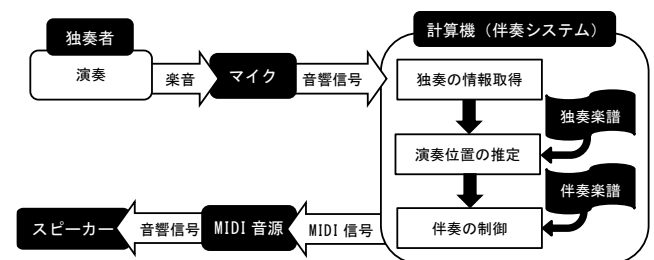


図1 伴奏システムの処理の概要

Digital Interface) と呼ばれる音色・音高・音の大きさなどの演奏情報をデジタル情報で伝達する規格を用いており、伴奏の演奏情報はリアルタイムで決定されるテンポに従って、MIDI信号としてMIDI音源に送信され、音響信号に変換された後、スピーカーから出力される。

伴奏システムと合奏する際、人間の独奏者は自分勝手に演奏することはできず、少なからず伴奏の演奏に影響を受けてしまう。このことは古くから指摘されており[6]、上述の伴奏制御モデルもこのような相互作用を考慮したものとなっている。この伴奏制御モデルを伴奏システム上に実装することにより、独奏者のテンポの揺らぎに対して伴奏システムが反応できるようになるのに加え、伴奏システムが人間らしい伴奏の応答を行うことにより、人間の独奏者にとっても合奏しやすいものとなることが期待される。一方、この伴奏制御モデルを適用するためには独奏者の演奏を正確に認識する必要があり、独奏者の演奏タイミングに認識誤りがあった場合、不自然な伴奏制御を行ってしまう。我々が開発している伴奏システムは音響入力であるため、認識誤りが発生することがある。そこで本研究では人間の独奏認識に誤りがあっても伴奏制御モデルが不自然なテンポ変化を行わない頑健な自動伴奏システムを提案・実装することを目的とする。

<sup>†1</sup> 千葉大学

## 2. 関連研究

伴奏システムに関連した研究では、主に独奏認識あるいは楽譜追跡、および伴奏制御の提案を行っている研究が多く存在する[7-11]. また、誤りへの対応を具体的に記述している研究も存在し、関連研究[12]では誤りや弾き直しおよび弾き飛ばしがあった際に楽譜上のあらゆる場所で探索をし、実時間で楽譜追跡を行う. この研究は演奏者が何度も弾き直しや弾き飛ばしを行う練習での場面を想定しているため、独奏者の演奏がジャンプした際に、独奏者の演奏している楽譜位置を推定し、短時間で独奏者と同じ位置に伴奏を移動させる手法を提案している. 研究[13]では独奏者の演奏において計算機による誤りまたは人間の弾き飛ばしによって音が脱落した時の伴奏制御での対処法を作曲者に委ねる手法を提案している. このとき作曲者はあらかじめ弾き飛ばされた独奏の音に対応した伴奏のフレーズを出力するか否か、および独奏者の音との時間的なタイミングを指示することができる.

## 3. 独奏認識に頑健な伴奏制御手法

### 3.1 独奏認識

本研究の独奏認識では音響特徴量から独奏者の演奏した音の開始時刻を認識し、それを楽譜情報とマッチングし、楽譜位置を推定している. ここで利用している音響特徴量は、独奏者の演奏の基本周波数、対数パワー、自己相関関数(パワーで正規化されたもの)の時間差分2乗和である. 独奏者の音の開始時刻は上記の音響特徴量から得られる音高、音量、音色の時間変化に基づいて推定している. この独奏認識の結果から伴奏制御モデルに使用する独奏者の演奏時刻を取得している.

### 3.2 伴奏制御手法

先行研究[4,5]では人間同士の演奏を分析し、拍単位での伴奏制御方法を提案した. これは拍ごとに独奏者に協調するように伴奏のテンポを計算して伴奏を出力するという方法である. 例えば拍子が4/4の楽曲の場合、1小節につき4回テンポの計算をすることになる. 従来の伴奏システムのほとんどは独奏者と伴奏者の演奏イベントごとにテンポを調整するものであったが、本研究では人間の伴奏者は全ての音符ごとに制御を行っているのではなく、拍を基準として自分の演奏を行いながら、独奏者と協調するようにテンポを制御していると仮定し、拍単位の伴奏制御モデルを提案した. その際、テンポの遅い曲の場合には半拍単位で制御を行うこともある. 逆にテンポの速い曲の場合には2拍あるいは3拍をひとまとまりにして拍と考え制御を行うこともある.

テンポ計算には独奏と伴奏の演奏位置がそれぞれ拍に到達したときの時刻(演奏時刻)から計算されるパラメー

タが使用される. パラメータは  $n$  番目の拍における独奏と伴奏のずれ  $d_n[\text{ms}]$  と伴奏の拍時間長変化  $c_n[\text{ms}]$  の2種類であり、それぞれ次式(1), (2)で定義される.

$$d_n = T_n - t_n \quad (1)$$

$$c_n = l_n - l_{n-1} \quad (2)$$

式(1)中の  $T_n$ ,  $t_n$  はそれぞれ独奏, 伴奏の  $n$  番目の拍の演奏時刻を表す. また, 式(2)中の  $l_n$  は伴奏の  $n$  番目の拍を演奏する際の直前一拍分の時間長を表し, これは一般的に用いられるテンポの単位 bpm (beats per minute) の逆数に相当する.  $l_n$  は伴奏の  $n$  番目の拍と  $n-1$  番目の拍の演奏時刻の差として次式(3)で定義される.

$$l_n = t_n - t_{n-1} \quad (3)$$

以上のパラメータを用いて直前の拍の独奏と伴奏の演奏時刻が揃った時点で次拍の伴奏の時間長変化  $c_n$  を次式(4)で計算する.

$$c_n = \sum_{i=1}^p \alpha_i d_{n-i} + \sum_{j=1}^q \beta_j c_{n-j} \quad (4)$$

式(4)における係数  $\alpha_i$ ,  $\beta_j$  の値は人間同士の合奏データに対して重回帰分析を適用し, 結果として求められる回帰係数を用いている. すなわち, 独奏と伴奏のずれと伴奏のテンポ変化の過去の履歴から次の伴奏のテンポ変化量を決めるモデルである. テンポ計算の際の各パラメータの関係を図2に示す.

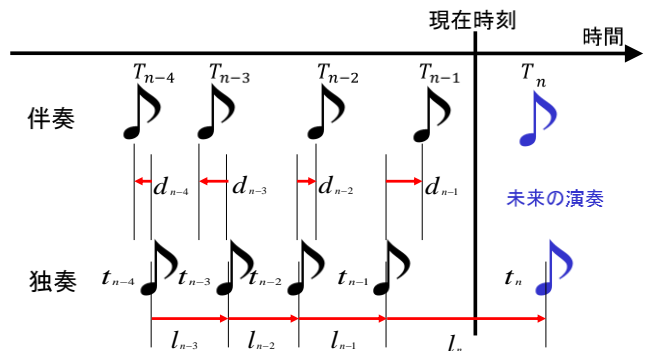


図 2 テンポ計算用パラメータの関係

### 3.3 実装上の問題点

3.2 節の伴奏制御モデルは過去の独奏と伴奏の演奏時刻の履歴をもとに伴奏テンポを決定しているが, 独奏の演奏時刻に誤りが生じてしまうと, 式(4)のずれのパラメータ  $d_n$  が実際のずれと異なる値となり, 急加速や急減速など不自然なテンポ変化をしてしまうことがある.

独奏認識で起こり得る誤りは置換, 脱落, 挿入の三通りに分けることが出来る. 楽器による音響入力の場合, 音高の認識はかなり正確に行えるため, 音高が変化する場面では独奏認識の精度が高いが, 同じ音高の音が連続する場面では音響特徴量の変化から先行する音の終了と後続する音

の開始を見分けることは容易ではない。ここで、本研究では楽譜上に存在する音を認識できなかった未検出と楽譜上に存在しない音を認識してしまった過検出の二種類の誤りへの対応方法を検討する。

未検出が発生した際は、伴奏テンポを計算するために必要な独奏の演奏時刻が揃わない。これにより独奏の演奏が止まっている、または減速したとシステムは解釈し、テンポが不自然に遅くなることもある。一方で過検出が起きた場合には予測よりも早く独奏が演奏（加速）したとシステムは解釈し、独奏とのずれを減らすために伴奏テンポを急加速させてしまうことがある。このように音の開始を検出するときの未検出や過検出により、伴奏制御モデルが適切に動作しない問題がある。未検出に対しては、その後の独奏者の演奏が正しく入力されれば、演奏ミスと同様の処理によって対応することができるが過検出に対しては対応が困難である。独奏者の音の認識において、未検出と過検出の発生はトレードオフの関係にあり、認識の閾値の設定により調整することができる。そこで従来は過検出を抑制し、未検出を増やすような閾値を設定して対処していた。

### 3.4 未検出と過検出への対応（提案手法）

伴奏システムはリアルタイムシステムであるため、独奏認識の結果が正しいのか誤りなのかをその時点で正確に判断することは容易ではない。また、伴奏制御モデルは独奏認識に対して瞬時に反応していく必要がある。従来手法では過検出エラーの発生を抑制して未検出エラーを増やして対処していたが、未検出の音符は伴奏制御に利用することができないため、独奏者の演奏に適切な反応を行えなくなってしまう。そこで本研究では未検出エラーを減らすと同時に過検出エラーに対して適切な制御を行えるようにすることで、独奏者のテンポ変化に適切に対応できると考える。また、独奏者は自分の演奏だけでなく伴奏者の演奏も聞きながら演奏しているため、独奏者の楽譜上の位置は伴奏者の楽譜上の演奏位置から大きくずれることはないと考えられる。そこで認識誤りの検出やその対処に伴奏システムの演奏時刻を利用する手法を検討する。

本研究では独奏認識における閾値を調整することにより未検出エラーの発生を減少させる。しかしながら、同じ音が連続するような場面では、完全に未検出エラーを防ぐことはできない。そこで独奏者のある音符がその予測時刻（独奏の過去の履歴から予測される時刻）から 300 ミリ秒以上経過しても検出されなかった場合、未検出エラーとみなし強制的に状態遷移を行う。このような未検出時には独奏の演奏時刻を取得できないが、独奏は伴奏と同時に演奏されたと仮定し、同拍の伴奏の演奏時刻を独奏者の演奏時刻として伴奏制御を行う。なお、300 ミリ秒という猶予時間は比較的長いですが、その間、伴奏システムは安定したテンポで演奏を継続しているため、それほど大きな問題とはな

らない。また、その間に独奏者の次の音符が入力されれば、伴奏制御モデルを適用することができる。この処理は独奏者がある音符の演奏をしなかった演奏ミスに対応する手法と同様の手法となっている。

本研究では独奏認識において未検出エラーの発生を減少させたことにより、過検出エラーが増加することとなる。その場合、ビブラートなどの微小な音響特徴量の変化も検出してしまふ。予想される時刻付近で検出が複数発生したときに、どの検出が正解の独奏の音符で、どの検出が過検出にあたるのかをリアルタイムで判断することは困難である。特に、同じ音高の音が複数続く場面では過検出の影響によって伴奏の演奏が急加速することが起こりうる。そこで本研究では過検出の発生も考慮に入れ、全ての検出時刻を利用して伴奏制御を行うが、その際に時間的な信頼度（後述）に基づいて伴奏制御量を調整することにより、過検出が存在した場合でも不自然なテンポ変動を防ぐこととした。過検出を含めた全ての検出時刻を利用する場合、独奏のある音符に対して複数回の検出を対応させる必要があるが、2回目以降の検出に対してはその都度、演奏開始時刻を更新して伴奏テンポを再計算する。その際、信頼度に応じた重み付けをした上でテンポ調整を行う。

上述の信頼度は伴奏の楽譜時刻と独奏の追跡候補の楽譜上の時刻のずれ  $u[s]$  により決定される（楽譜上の時刻は初期テンポで演奏された場合の実時間で表現しているため、単位は秒となる）。信頼度の計算に独奏と伴奏の楽譜位置のずれを用いるのは、人間の独奏者は通常は伴奏に対して、大きく逸脱した演奏はせず、伴奏者が演奏している楽譜位置付近を演奏していると仮定しているからである。もし独奏者が大きく逸脱した演奏を行う場合は独奏者の演奏意図を事前に学習するリハーサルにて対応できると考える[14]。

伴奏の楽譜時刻とのずれの信頼度  $r$  ( $0 \leq r \leq 1$ ) は式 (5) で与える。

$$r = \begin{cases} \frac{1}{0.04}(u + 0.3)^2, & -0.3 < u < -0.1 \\ 1, & -0.1 \leq u \leq 0.2 \\ \frac{1}{0.09}(u - 0.5)^2, & 0.2 < u < 0.5 \\ 0, & u \leq -0.3, u > 0.5 \end{cases} \quad (5)$$

$u$  が正の場合は検出した独奏の楽譜時刻が伴奏の楽譜時刻より遅い、また  $u$  が負の場合は楽譜位置候補の楽譜時刻が伴奏の楽譜時刻より早いことを示す。式(5)から  $u$  が  $-100$  ミリ秒から  $+200$  ミリ秒の間は妥当なタイミングであるとして、信頼度は 1 としている。 $r$  を  $u$  の正負により非対称に設定しているのは、 $u$  が負で独奏が先行したときに伴奏テンポは加速することとなるが、伴奏システムにおいては、意図的ではない加速傾向を防ぐことが重要であるからである。これは合奏においては先に演奏した方が合奏全体のテンポ変化に強く影響してしまう傾向があり（先

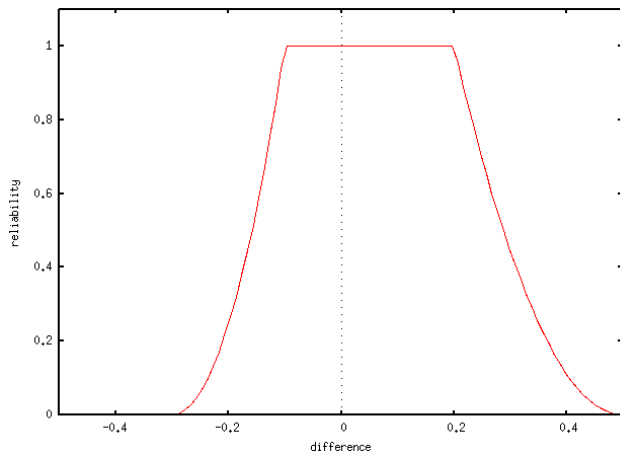


図 3 信頼度曲線 (式(5)) のグラフ  
 (横軸: ずれ  $u$ 、縦軸: 信頼度  $r$ )

に演奏した人にテンポ決定の優先権が生じてしまう), 不適切に早いタイミングで演奏してしまうと演奏相手はそのテンポに追従せざるを得なくなってしまい, 結果として演奏相手や合奏全体に悪影響を与えてしまう. 逆に相手よりも遅いタイミングで演奏した場合, 相手は当該拍をすでに通過してしまっているため, その後, 緩やかに相手とのテンポ調整を行うことができる. このことから, 合奏で伴奏システムがテンポ変化をする際には, 相手に先行する場合は慎重に行う必要があり, 上記のテンポ制御の信頼度はその知見を反映させたものとなっている.

過検出の音符に対して通常の制御モデルでテンポ計算を行うと伴奏の楽譜位置とのずれが大きくなっていくことから, 過度な加速や減速をする方向に伴奏のテンポが変化してしまうと考えられる. そこで, このときの制御量を上述の信頼度に応じて減らすことで急なテンポ変化の抑制ができると考える. 式(4)に対して, 式(5)の信頼度により, ずれのパラメータ  $d_n$  に重み付けをした式を(6)に示す.

$$c_n = \sum_{i=1}^p \alpha_i r_{n-i} d_{n-i} + \sum_{j=1}^q \beta_j c_{n-j} \quad (6)$$

以上の方法を用いることによって, 未検出や過検出に対応し, これらの誤りによる急激な伴奏のテンポ変化が伴奏制御量の調整によって抑制できると考えられる.

## 4. 評価実験

### 4.1 実験方法

第3節で述べた誤りに対する提案手法を検証する評価実験について述べる. 実験は3名のフルート演奏者の協力の下で行い, 誤りに対する提案手法を実装したシステムと誤りに対処していない従来システムをユーザに使い比べてもらい, 結果を比較した.

演奏曲目としてメンデルスゾーン作曲『歌の翼』による

表 1 各システムの評価

システム	評価点	評価点 (棄権除)	途中棄権	未検出	過検出
従来	5.08	6.52	11	33	0
提案	7.11	7.28	1	16	111

幻想曲), エルガー作曲「愛の挨拶」, マリ作曲「金婚式」の3つの楽曲をわずかに編曲したものを用意し, 各曲の1部分を抜粋したものを実験に使用した. また, 各曲に「テンポ一定」「徐々に加速」「徐々に減速」が全て含まれるようにテンポ指示を行った. なお, 本実験では, 「突然加速」「突然減速」などの急激なテンポ変化は対象としない.

提案システムは Intel® Core™ i7-4500U CPU @ 1.80GHz 2.40GHz (8.00GB RAM), Microsoft Windows 7 Professional 2009 (SP1) が動作するコンピュータ上で Microsoft Visual C++.net 2013 を用いて実装した. コンピュータ以外に MIDI 音源として Edirol SD-20, 独奏の音響情報を取得するマイク用のオーディオインターフェースとして Edirol UA-20 を用いた. 実験において独奏者はスピーカーから流れる伴奏の音と自分の生の演奏音を聞きながら演奏を行った. なお, 演奏開始の合図としては, 先行研究[15]で実装されたプレスによる合図を用いている. また, 本実験では式(6)において  $p = q = 1$  とした伴奏制御モデルを用いた.

実験では提案システム, 従来システムの計2種類のシステムを用意し, 各曲は4回ずつ演奏を行った. すなわち実験全体で1人あたり計4回×2(システム2種類)×3(収録曲3種類) = 24回の比較実験を行った.

実験参加者には伴奏の演奏が自分の演奏に合わせやすかったかを10段階で回答してもらった(10が最も合わせやすい). なお, 評価対象はテンポ指定区間を含めて演奏全体とし, 伴奏システムの演奏全体の良し悪しを10段階評価してもらった. なお, 伴奏が独奏に対して大きくずれて演奏の続行が不可能となってしまった場合, 独奏者の判断で演奏を途中棄権することを許した.

### 4.2 実験結果と考察

表1に実験結果を示す. 「評価点」は全36回の主観評価値の平均を表し, 「途中棄権」は独奏者の意思で演奏を棄権した回数を表す. 「未検出」と「過検出」は独奏認識において未検出と過検出それぞれの誤りが起こった回数を示す. また, 「評価点 (棄権除)」は途中棄権の主観評価値を除いた評価点の平均を示す. 表から提案システムの評価点は従来システムに対して向上したことが分かる. これは途中棄権と未検出の回数の減少に起因すると考えられる. なお, 提案システムでも未検出は存在しているが, 従来手法と同様, 誤りへの対応により伴奏制御には大きな影響を与えなかったと考える. 一方で提案手法により過検出が増えている

るが、提案手法の信頼度に応じた伴奏制御量の調整が適切に機能したため、評価点が向上したと考えられる。提案システムでは途中棄権が一回起きているが、これは曲を始める際の独奏者のブレスのタイミングが不自然であったため、伴奏システムが正しく独奏者のテンポを推定できず、伴奏システムの演奏が独奏者と食い違い、結果として伴奏のテンポが急加速してしまったことが原因として考えられる。これはブレス検出の問題であり、提案手法に起因するものではない。また、演奏者の意見として加速時に伴奏の反応が鈍かったとの指摘があったが、これは式(5)の信頼度で加速の度合いを減らしていることに起因すると考えられる。

以上から、本研究で提案した未検出と過検出への対応は独奏の独奏認識に誤りが起こった際の伴奏制御に有効であり、それをシステムに実装することで安定した伴奏制御を可能にし、ユーザ評価を向上させることができたと言える。

## 5. おわりに

本研究では独奏の認識誤りに頑健な伴奏システムを実現するため、独奏認識の未検出と過検出それぞれに対応した手法を提案し、伴奏システムに実装した。評価実験の結果、未検出と過検出が起きた時の伴奏テンポへの影響を軽減したことで、従来のシステムよりも独奏認識の誤りがあったときのユーザ評価を向上させることができた。

今後は伴奏システムと人間の独奏者との合奏を録音するなどして音の立ち上がり時刻を計測し、実装したシステムについての定量的な評価や信頼度および伴奏制御量の妥当性の検討を行いたいと考えている。また、より多くの合奏データを集めて分析して伴奏者の個人差や共通点を考慮したモデルを考え、様々な独奏者や楽曲の特性に対応した伴奏システムを目指したい。

**謝辞** 本研究は JSPS 科研費 26240025 の補助を受けています。

## 参考文献

- [1] Roger B. Dannenberg: An On-Line Algorithm for Real-Time Accompaniment, Proceedings of International Computer Music Conference, pp. 193-198 (1984).
- [2] Barry Vercoe: The synthetic performer in the context of live performance, Proceedings of International Computer Music Conference, pp. 199-200 (1984).
- [3] 堀内靖雄, 坂本圭司, 市川熹, “合奏時の人間の演奏制御の分析・推定,” 情報処理学会論文誌 45(3), 690-697, 2004-03-15.
- [4] 和田静花, 堀内靖雄, 黒岩眞吾, “名演奏家の合奏録音における伴奏者の演奏タイミング制御の分析,” 情報処理学会研究報告. [音楽情報科学],

- 2013-MUS-100(9), pp. 1-6, 2013.
- [5] Shizuka Wada, Yasuo Horiuchi, Shingo Kuroiwa, "Tempo Prediction Model for Accompaniment System" Proceedings of ICMC & SMC 2014 Joint Conference, Sep.2014.
- [6] 澤田秀之, 磯貝昌幸, 橋本周司, 大照完, “音楽演奏における人間と機械の協調動作について,” 全国大会講演論文集 第44回(応用), 389-390, 1992-02-24.
- [7] C. Raphael: Music Plus One and Machine Learning Machine Learning, Proceedings of the Twenty-Seventh International Conference (ICML 2010).
- [8] 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, Vol. 2006, No. 90, pp. 109-116 (2006).
- [9] 前澤陽, “自動合奏のための演奏タイミング結合モデル,” 研究報告音楽情報科学 (MUS), 2016-MUS-112, pp. 1-8, 2016.
- [10] Arshia Cont. A Coupled Duration-Focused Architecture for Real-Time Music-to-Score Alignment. IEEE Transactions on Pattern Analysis and Machine Intelligence, Institute of Electrical and Electronics Engineers, 2010, 32, pp.974-987.
- [11] Guangyu Xia, Yun Wang, Roger Dannenberg, Geoffrey Gordon. “Spectral Learning for Expressive Interactive Ensemble Performance”, 16th International Society for Music Information Retrieval Conference, 2015.
- [12] 中村友彦, 中村栄太, 嵯峨山茂樹, “誤り・任意の弾き直し・弾き飛ばしを含む演奏音響信号への高速な楽譜追跡,” 情報処理学会研究報告. [音楽情報科学], 2013-MUS-99(40), pp. 1-5, 2013.
- [13] Arshia Cont, Jos'e Echeveste, Jean-Louis Giavitto, Florent Jacquemard. Correct Automatic Accompaniment Despite Machine Listening or Human Errors in Antescofo. ICMC 2012 – International Computer Music Conference, Sep 2012, Ljubljana, Slovenia. 2012.
- [14] 山田彩歌, 堀内靖雄, 黒岩眞吾, “伴奏システムにおける独奏者の演奏意図学習のためのリハーサル,” 情報処理学会研究報告[音楽情報科学], 2015-MUS-106(7) pp1-6, Mar 2015.
- [15] 堀内靖雄, 西田昌史, 市川熹, “ブレスによる合図を検出する伴奏システム,” 情報処理学会論文誌, Vol.50, No.3, pp.1079-1089, 2009.3.