

複数の人間と協調する演奏システム

堀内 靖雄 藤井 敦 田中 穂積

本論文では「複数の人間と協調する演奏システム」について述べる。計算機が複数の人間と協調して自然なアンサンブル演奏を行なうためには、すべての演奏者の演奏を考慮してシステム自身のテンポを制御する必要がある。しかしメンバー全員の演奏をすべて等しく考慮するよりは、システムはアンサンブルで重要な演奏者ほど、良く合わせる方が良い。そこで本論文ではまず、アンサンブルにおける各演奏者の重要さを表わす尺度として「独立度」の考えを提案し、その意味付けと設定方法について述べる。次に独立度に応じてシステムの演奏を制御する手法として「アンサンブル時間」の考えを提案する。実際の演奏において、システムはアンサンブルをリードしている演奏者（独立度の高い演奏者）ほど良くテンポを合わせ、またシステムがアンサンブルの中心となるような（システムの独立度が高い）状況では自主的で表情豊かな演奏を行なうことができる。実験的なシステムにより、独立度の変化に応じて人間の演奏者に対する追従法

の変化が確認された。

1 はじめに

音楽におけるコンピュータの利用には長い歴史があり、様々な研究が行なわれている [16] [17]。このような音楽とコンピュータとの関係の中で、最も重要な問題の一つは、ライブ音楽演奏におけるコンピュータの利用、特にコンピュータがライブ演奏のメンバーとして人間と協調して一緒に演奏を行なうというものである。従来は処理速度の問題のため、コンピュータによるリアルタイム演奏は困難であった。そのためコンピュータによる演奏は通常、テープに多重録音され、実際の演奏時にそのテープを再生することにより実現されていた。このような手法では、もし人間の演奏家がコンピュータと一緒に演奏しようとしても、コンピュータはあらかじめプログラムされた演奏しかできないため、人間の演奏家がコンピュータの演奏にテンポなどを合わせて演奏しなけりなかつた。

しかし近年のコンピュータの計算速度の飛躍的な向上により、コンピュータが人間のライブ演奏家に合わせてリアルタイムでコンピュータ自身の演奏を制御することが可能となった。このような研究の一つとして、伴奏システム ([7] [13] etc.) が挙げられる。伴奏システムは1人の人間の演奏にテンポを合わせて伴奏を行なうシステムである。我々はこの伴奏システムを複数の人間と協調して演奏するアンサンブル・システムへと拡張する。アンサンブル・システムは一緒に演奏している複数の人間の演奏家の演奏すべてをその重要度に応じて考慮し、人間の演奏者と協調した自然な演奏を行なうことを目的と

A Performance System Cooperating with Several Human Players.

Yasuo Horiuchi, 東京工業大学理工学研究科情報工学専攻, Department of Computer Science, Tokyo Institute of Technology (現千葉大学工学部情報工学科, Department of Information and Computer Sciences, Chiba University).

Atsushi Fujii, 東京工業大学情報理工学研究科計算工学専攻, Department of Computer Science, Tokyo Institute of Technology.

Hozumi Tanaka, 東京工業大学情報理工学研究科計算工学専攻, Department of Computer Science, Tokyo Institute of Technology.

コンピュータソフトウェア, Vol.12, No.5 (1995), pp.63-71.
[論文] 1994年2月25日受付。

する。また、システムがアンサンブルの主導権を握るべき状況では、自主的で表情豊かな演奏を行なうことにより、他の演奏者をリードした演奏を行なえるようにする。

本論文ではまず第2章で従来の伴奏システムについて述べた後、我々の開発した「自主性を持つ伴奏システム」[9]について述べる。第3章以降でこの伴奏システムから「複数の人間と協調する演奏システム」への拡張について述べる。

2 伴奏システム

伴奏システムとは、コンピュータが1人の人間の演奏にテンポを合わせて伴奏を行なうシステムであり、独奏者と互いに影響を及ぼし合いながらコンピュータ自身の演奏を行なうため、リアルタイム・インタラクティブ・システムの一つであると考えられる。このような適応型の伴奏システムに関しては、すでにいくつかの研究[2][5][8][10][11] etc.)が行なわれている^{†1}。

伴奏システムは独奏・伴奏パートの楽譜があらかじめ与えられており、演奏時に伴奏システムは、独奏者の演奏を聴きながら、その楽譜情報を用いて、独奏者の楽譜上の位置を認識し、その演奏に合わせてようにシステム自身の演奏テンポを制御しながらリアルタイムで伴奏を出力する。楽譜をあらかじめ与えておかないインタラクティブ音楽システムとして即興やリアルタイム作曲を行なうものもあるが[6][12]など、それらはビート・トラッキング、即興、作曲などが主なテーマとなり、処理の内容も伴奏システムとは大きく異なるのでここでは対象としない。

従来の伴奏システムの研究では、コンピュータが独奏者のテンポにできるだけ合わせることを目標としていた。しかし実際には人間の独奏者もコンピュータの演奏(伴奏)に影響を受けており、両者による演奏は互いに影響を及ぼし合っている。これまでの研究は、独奏者からコンピュータへの情報を用いて伴奏を制御するだけであったため、コンピュータから独奏者への影響はほとんど考慮されていなかった。すなわち、従来の伴奏システムによるアンサンブルには双方向のインタラクションがなく、いわば一方通行による伴奏制御であったといえる。しか

し自然なアンサンブルを実現するためには、独奏者と伴奏者のインタラクションを考えることが重要である。

最も簡単な例として、例えば独奏者が長い音符を演奏しているときや休みのときには、テンポを主に決定するのは伴奏システムであり、独奏者は伴奏システムの演奏に合わせて演奏を続けるであろう。ところが従来の伴奏システムでは、このように独奏者から得られる情報が少ないときやまったくないときには、合わせるべき対象がないため、伴奏システムは自分の楽譜情報に従って演奏を行なうしか方法がなく、通常は一定のテンポによる機械的で不自然な演奏になってしまうことが多かった。しかし、このような状況では人間同士によるアンサンブルと同様、伴奏システムがアンサンブルの主導権を握り、人間の独奏者をリードし、人間の独奏者が伴奏システムに合わせるべきであると考えられる。

以上のことから、自然なアンサンブルを実現するためには、伴奏システムが独奏者に単に合わせるだけでは不十分であることが分かる。そこで我々は伴奏システムの自主性を考慮することにより、自然なアンサンブルの実現を目標とした伴奏システムを開発した[9]。このシステムは人間の独奏者がアンサンブルの主導権を握る状況(独奏者がメロディを演奏しているようなとき)では、従来の研究と同様、伴奏システムが独奏者によく合わせて追従し、逆に、伴奏システムがアンサンブルの主導権を握る状況(伴奏パートがメロディであるようなとき)では、伴奏システムが自主的な演奏を行なうことにより、独奏者をリードすることができた。

この伴奏システムを評価するため、人間の独奏者と3通りの伴奏システム(人間の伴奏者、従来の研究に基づく伴奏システム、我々の開発した伴奏システム)による演奏を36人の被験者に聴き比べてもらう聴取実験を行なった。この実験では、被験者に上記の3種類の演奏を録音したテープを十分な回数聴かせた後、4つの評価項目についてその3種類の演奏に順序付けをさせた。評定結果の検定を行なったところ、我々の伴奏システムは従来のシステムよりも良い評価を受け、また人間の伴奏者と比較しても遜色がないという結果を得た[9]。

^{†1} これらの詳細については文献[18]を参照されたい。

3 複数の人間と協調する演奏システム

伴奏システムは1人の人間の演奏者に合わせる演奏システムであるが、複数の人間と一緒に演奏するシステムへの拡張を考えた場合、新たに問題が生じる。その問題とは、複数の人間に合わせようとした場合、人間同士の間にすでに若干(人間にとってはそれほど大きく感じられない程度)のずれが生じているということである。すなわち、合わせるべき相手が複数存在し、それぞれが若干ずれているので、システムは自分の演奏を決定するのに、これら競合した複数の人間の演奏に対して、どのようにして合わせるかを考えなくてはならない。

この問題の最もナイブな解決方法は何らかの方法で複数の人間の誰か1人を選択し、その人間の演奏だけに合わせるというものである。これは従来の伴奏システムをそのまま利用すればよい。しかし、そのような方法では、選択した人間が長い音符を演奏しているときや休符の場合には合わせるべき相手を失ってしまい、他の演奏者とずれてしまうことになる。明らかに複数の人間と協調して演奏を行なうためには、すべての演奏者の演奏を考慮すべきである。

次に考えられる方法は、従来の(Pianoなどの)ポリフォニック楽器の伴奏システム[4]を利用する方法であるが、これはもともと1人の演奏者に合わせることを前提としているので、和音が入力されると最初の音符が入力された時点とその和音の演奏時刻としてしまう。このシステムに複数の人間からの演奏を入力すると、コンピュータはすべての演奏者の中で最初に演奏した演奏者に合わせてしまうことになる。常に最初に演奏した演奏者に合わせるような方法では、コンピュータの自主的なリズム感が存在しないため、まわりの変化に過敏に反応する不自然な演奏になってしまう。

Bairdらは複数の人間の演奏を、入力された時刻、マッチングの確からしさ、旋律の重要性などによって重み付けしてコンピュータの演奏を制御する考えを提案しているが、具体的な方法の呈示にはまだ至っていない[1][3]。また、この方法ではコンピュータが人間の演奏者をリードすることができないが、音楽的な状況によっては、コンピュータがメロディを演奏するなど、他の演奏者をリードすべき状況も生じる。このような状況では

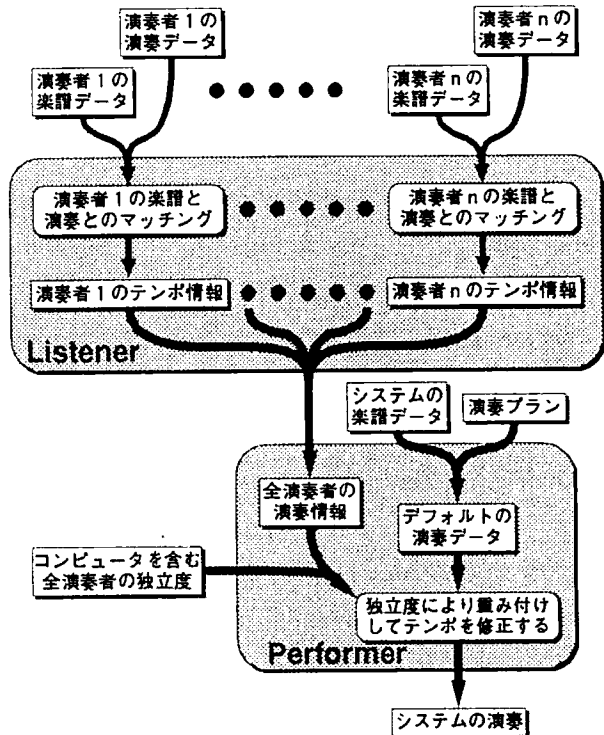


図1 アンサンブル・システムの構成

我々の伴奏システム[9]と同様、コンピュータが自主的に表情豊かな演奏を行ない、他の演奏者をリードすることが望ましい。そこで我々は、上述の伴奏システム[9]を拡張し、複数の人間とのアンサンブルにおいて、すべての演奏者の演奏を状況に応じて考慮した自然な演奏を行ない、また状況によってはコンピュータが人間の演奏者をリードしながら、自主的で表情豊かな演奏を行なう演奏システム(アンサンブル・システム)を提案する。

3.1 アンサンブル・システムの構成

アンサンブル・システムの構成を図1に示す。人間の演奏者とアンサンブルを行なうとき、各演奏者の演奏はMIDI信号としてシステムに入力され、システムはこの演奏に合わせてリアルタイムでコンピュータの演奏(MIDI信号)を出力する。この図はn人の人間と一緒に演奏を行なう場合を示しているが、nは1でもよい(そのときの処理は伴奏システムと同じである)。nの上限は原理的にはいくつでもよいが、現時点ではMIDIの制約から16としている。また、本システムは演奏者が基本的に楽譜通りの演奏を行なうことを仮定しているため(当然、演奏ミスなどは考慮に入れられており、許容できる)、本システムが主対象として扱う音楽は西洋クラシッ

ク音楽とするが、バロック音楽に見られるような装飾などはジャズの即興演奏と同様に現時点では対象外とする。

アンサンブル・システムは Listener と Performer の 2 つのモジュールからなっている。Listener は、すべての人間の演奏者の楽譜をあらかじめ与えられており、その楽譜情報を用いて、各演奏者の演奏が楽譜上のどの位置にあるかを追跡する。Performer は、システムの演奏すべき楽譜と後述する演奏プラン・独立度を与えられており、Listener から出力される各演奏者の演奏位置情報に基づいてコンピュータ自身の演奏テンポを制御し、実際の演奏を出力する。

3.2 Listener

Listener は各演奏者ごとに独立して、楽譜を参照しながら、その演奏者の演奏を追跡し、その演奏者が現在、楽譜のどの位置を演奏しているかという情報を Performer に出力する。各演奏者ごとの追跡アルゴリズムは以前の伴奏システム [9] と同じであり、各追跡プロセスは仮想的に並列に処理される。

各演奏者の演奏 (MIDI 信号) は、1 つの MIDI 信号にマージされ Listener に入力される (このとき、各演奏者の演奏は MIDI のチャンネル番号で識別可能である)。ある演奏者が音符を演奏したとき、MIDI のノート・オン信号が Listener に入力される。ノート・オン信号は、チャンネル番号、その音符の音高情報を持っている。また Listener はその音符が入力された時刻を記憶しておく。Listener はチャンネル番号から、どの演奏者の演奏かを識別し、その演奏者の楽譜情報と入力された音符の音高情報・演奏時刻を利用して、その演奏者の演奏を追跡し、その演奏者が現在、楽譜上のどの位置を演奏しているかを認識する。

しかし人間の演奏は、楽譜通りに一定のテンポで演奏されることはなく、また演奏ミスをすることもありうる。そこで、人間の演奏者のテンポの揺らぎに応じて、対応する楽譜の時間軸を伸縮したり、演奏ミスを許容して人間の演奏を追跡するため、音声認識などで用いられている DP マッチング・アルゴリズムをリアルタイムに適用する。

DP マッチングにおいて、ある演奏者が演奏ミスを犯したり、大きなテンポ変化を行なったときには、その演

奏者の演奏と楽譜とのマッチングが一意に決定できず、マッチングの候補が複数存在する状況が生じることがある。その場合、Listener は可能な複数の解釈 (仮説) と、その解釈の妥当性を表わすスコア (得点) を持ちながら、その演奏者の演奏を追跡する。すなわちある時点での解釈は複数存在し、各解釈ごとにユニークなスコアを持っており、スコアが大きいほどその解釈が妥当であるということを意味する。また、複数ある解釈の中でスコアがあらかじめ決められている閾値よりも小さくなった解釈は妥当でないと考え、破棄される。ある演奏者の解釈が複数考えられるときは、スコアの最も高い解釈がその演奏者の現在の演奏であると考え、追跡における妥当な解釈は通常 1 つであり、演奏ミスなどが発生したときに複数に分岐するが、その後、正しい入力が続くと誤った解釈は破棄されるか、正しい解釈にマージ (同じ楽譜位置にある解釈は 1 つの解釈に統合する) されることにより、1 つの解釈に収束する。

Listener は初期状態として各演奏者ごとに 1 つずつの解釈 (その演奏者が楽譜の最初の位置にいる) を持ち、それらのスコアを最大値にセットする。その後、各演奏者の演奏イベント (ノート・オン信号) を受け取るごとに DP マッチングを行ない、その演奏者のすべての解釈とそのスコアを更新しながら追跡を続ける。もし、ある解釈のもとで、入力された演奏イベントが楽譜から予測された次に来るべき正しい音符であり、予測された時刻付近で入力された場合、その音符が楽譜とマッチしたと考え、その解釈のスコアを最大値を越えない範囲で増加する。

演奏ミスや大きなテンポ変化のために、ある解釈におけるマッチングの可能性が複数存在するときは、それぞれのマッチングに対して、複数の新しい解釈に分岐し、それらの解釈を持ちながら追跡する。そのときミスや時間逸脱の度合いに応じてペナルティを与えるため、もとの解釈のスコアに 1 より小さいペナルティを乗じる。

ペナルティはまず、音符の間違いに対して間違いペナルティ $C_{miss} (< 1)$ を乗じ、さらに、その音符が入力された時刻が予測された時刻より離れるにつれて大きなペナルティを課すように時間ペナルティ $C_{time}(\Delta t) (< 1)$ を乗じる。Dannenbergh らは時間的なペナルティは考慮に入れていないが [4]、Vercoc は時間的なペナルティの

導入を提案している [14]。そのため我々のアルゴリズムは Vercoe のものに近い。

ここで C_{miss} は間違いの種類によって異なる定数である。具体的には、楽譜にある音符の音程を間違えて演奏した場合、楽譜にない音符を余分に演奏した場合、楽譜にある音符を省略した場合のそれぞれに対して別々のペナルティ (定数) を用いている。

C_{time} は予測された時刻と実際に演奏された時刻との間の時間のずれ Δt の関数であり、 Δt が大きいほど C_{time} は小さくなる。このとき、予測された時刻とは、その演奏者の過去のテンポ履歴から重み付け平均によって推測する。すなわち、ある演奏者に対して複数の解釈が存在する場合、各解釈ごとに過去の履歴が異なるので、その推定テンポも解釈ごとに異なる値を取り、次の予測時刻も解釈ごとに異なってくる。この予測時刻とマッチングの対象となっている実際の演奏イベントの時刻とのずれを Δt とし、 $C_{time}(\Delta t)$ を計算する。 C_{miss} の値、及び $C_{time}(\Delta t)$ 関数は追跡の精度に影響を与えるため、その最適値を現在調査中である。

Listener は各演奏者ごとにスコアが最も高い解釈を現在のその演奏者の演奏であると考え、その解釈に従って 1 拍ごとにその演奏位置情報を Performer へ出力する。通常はこのように楽譜の 1 拍ごととしているが、テンポの遅い曲の場合は、楽譜上の 1 拍を 2 分割したり、逆に速い曲の場合には、楽譜上の 2 拍をまとめて 1 拍としている。オーケストラの指揮者が 4 拍子を 8 拍で指揮したり、2 拍で指揮したりする状況を考えて欲しい。出力されるテンポ情報は演奏者の識別子 (MIDI 信号のチャンネル番号)、その演奏者の最もスコアの高い解釈に基づく楽譜上の位置 (拍時間)、演奏時刻 (実時間)、その解釈のスコアの 4 つからなる。

3.3 Performer

Performer は、あらかじめ与えられたコンピュータ・パートの楽譜を演奏するが、人間の演奏者と協調するため、Listener から入力される各演奏者の演奏位置情報を用いてコンピュータの演奏テンポを制御する。そのとき複数の人間すべてと協調して自然な演奏を行なえるようにするため、本研究では「演奏プラン」「独立度」「アンサンブル時間」の 3 つの考えを提案する。

3.3.1 演奏プラン

人間は演奏を行なう際に、自分の意志に基づく独自の演奏プランに従って演奏しており [15]、また、アンサンブルの場合には、この独自の演奏プランを動的に修正し、他の人間と協調しながら演奏を行なっていると考えられる。そこでアンサンブル・システムも人間のように表情豊かに演奏しながら、自然な演奏を行なえるようにするため、システム内部にあらかじめ簡単な演奏プランを与えておく。これはいわばコンピュータの静的な演奏意志に相当するものである。

本システムに与えている演奏プランは、楽譜に書かれていない表情付け情報としてテンポの揺らぎ、音量変化、ペダル情報などである。このときテンポはグローバルなテンポとローカルなテンポの 2 段階で表現されている。グローバルなテンポとは、比較的長い時間スパンにまたがるテンポを表わしている。ローカルなテンポとはリズム的な揺らぎで、例えば十六分音符が 4 つ並んでいるようなときに、最初の十六分音符を少しゆっくり、残りを少し速くというような演奏を表現することができ、これはグローバルなテンポからの揺らぎを意味する。

3.3.2 独立度

人間同士のアンサンブル演奏では、互いに協調しながら演奏を行なっている。そのとき、すべての演奏者が対等に演奏を行なっている訳ではなく、その中にはアンサンブル全体をリードするような演奏者、その演奏者に合わせようとしている演奏者というように、各演奏者ごとにアンサンブルにおける重要性があると考えられる。この重要性は一曲を通じて常に一定ではなく、状況によって変化する。例えば弦楽四重奏の場合、通常は第一 Violin がリードするが、Viola や Cello がメロディとなり、他の演奏者をリードすることもある。このような人間同士のアンサンブルにコンピュータが入って一緒に演奏する場合、コンピュータも人間同様、すべての演奏者の重要性を考慮しながら、協調した演奏を行なうことが望ましい。そこでシステム内部に各演奏者のアンサンブルにおける重要性を表わす尺度として「独立度」を与える。コンピュータ自身もアンサンブルのメンバーなので独立度を与えておく。

独立度は各演奏者ごとに異なる値を取り、楽譜の位置ごとに変化する。独立度は正の整数値であり、数値が高

いほどその演奏者のアンサンブル全体のテンポに及ぼす影響が大きいと考える。すなわち、コンピュータの独立度が他の演奏者よりも高いという状況は、コンピュータは他の演奏者の演奏にあまり影響を受けずに自主的で表情豊かな演奏を行ない、他の演奏者をリードすべきであるということを意味している。逆に、コンピュータの独立度が他の演奏者に比べて低いときは、コンピュータは独立度の高い演奏者に合わせるべきであるということを表わしている。しかし、そのような状況でも、独立度の高い演奏者の演奏にピッタリと合わせて演奏するのでは、コンピュータの自主的なリズム感がなくなるため、不自然な演奏になってしまう。そこでコンピュータは急激にその演奏者に合わせるのではなく、自主的なリズム感を維持しながら、徐々にその演奏者に合わせてゆく。

しかし状況によってはアンサンブルのメンバーが楽譜上で同じ時刻に書かれている、ある特定の音符を狙い込んでタイミングを合わせて一緒に演奏する場合がある。そのような場合を特に「(楽譜上の) 縦の線を揃えるべきところ」とし、コンピュータの独立度として、特別な意味を持つ値 0 を与える。コンピュータの独立度が 0 のときは、他の人間と「同時」にコンピュータのパートの音符を出力するようにする。ここでいう「同時」とは、その位置の音符を誰かが最初に演奏した直後に出力するということであり、若干の遅れが生じるが、人間にはほぼ同時に聴こえる範囲を考える。

逆にコンピュータ以外のパート(人間の演奏パート)の独立度がすべて 0 の状況は、コンピュータが他の人間の演奏に影響を受けず、演奏プランに従って表情豊かに自主的な演奏を行なうべきであるということの意味する。ただし、この場合にも、そのときのテンポから突然逸脱してコンピュータの演奏プラン通りに演奏するのではなく、徐々に演奏プランのテンポに近付ける。

現在のシステムでは、独立度は楽譜の各パート間の関係から以下の規則を順に適用しながら人間が決定し、あらかじめシステムに与えている。しかし独立度の与え方は重要で、以下の規則だけから単純には決定できないので、実際には何度か演奏を行ない、修正を繰り返しながら決定している。

以下、規則について譜例を挙げながら説明する。譜例は J. Brahms 作曲の「ピアノ三重奏曲第 1 番ロ長調作

品 8」の第 1 楽章から抜き出している。この例ではコンピュータが Piano パートを演奏すると仮定する。

1. ある演奏者が長い休みのときはその演奏者の独立度を 0 にする。譜例 1 では Violin パートが休みであり、このようなときに Violin の独立度を 0 にする。
2. アンサンブルのメンバーが縦の線を揃えるべきところ(同時に演奏すべきところ)ではその瞬間のコンピュータの独立度を 0 にする。譜例 2 の 2 小節目の 1 拍目と 3 拍目の二分音符は縦の線を揃えたいので、コンピュータの独立度を 0 にする。ここでは、その前のアウフタクトの四分音符が一般にやや長目に演奏されることが多く、そのとき、テンポ変化が起こるため、次の二分音符のタイミングの予測が困難である。このような状況のときにこの規則を適用する。

コンピュータの独立度を 0 にする頻度を高くすると縦の線はよく揃うようになるが、これはコンピュータの演奏を楽譜上の先の方へ飛ばしたり、演奏を止めて待ったりするので、あまり多用すると不自然な演奏になってしまう。逆にそのような状況が少ないと、合わせるべきところをピッタリと合わせる事が困難になる。それらを考慮に入れ、適当な箇所に人間が設定する必要がある。

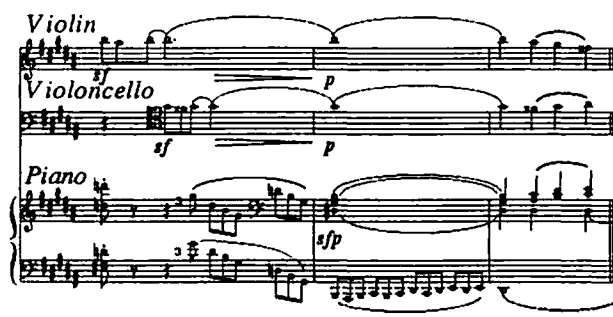
3. 一般的にコンピュータの独立度は人間のものよりも若干低めにする(通常は人間の演奏者に合わせようとする)。例えば譜例 1 では Piano を 40, Cello を 60 のように設定する。
4. メロディを演奏しているパートの独立度は他の演奏者の独立度よりも高くする。譜例 1 の 2 小節目のアウフタクトから Cello がメロディとなるので、Cello の独立度を若干高くし、Piano の独立度を若干下げると、
5. ある演奏者が長い音符を演奏しているときはその演奏者の独立度を低くする。譜例 3 の 2 小節目は Violin も Cello も長い音符を演奏しているので、両者の独立度を下げる。
6. ある演奏者のパートが他のパートに比べて細かい動きをしているとき(弦楽器のきざみのような)は、その演奏者がアンサンブル全体のテンポに大きな影響を与えていると考えられるため、その演奏者の独



譜例 1 Violin が休みで Cello がメロディ。



譜例 2 縦の線を揃えるところ。



譜例 3 Violin と Cello は長い音符を演奏しており、Piano は細かい音符を演奏している。

立度を若干高めにする。譜例 3 の 2 小節目の Piano の左手 (楽譜の下側) は細かい動きをしており、アンサンブルのテンポを決定しているため、この部分の Piano の独立度を上げる。

システムはこの独立度に応じてすべての演奏者の演奏を考慮し、適当な演奏を行なう。すなわち、システムは独立度の高い演奏者のテンポには比較的好く合わせるが、他の演奏者の演奏もその独立度に応じて考慮し、まったく無視することはない。また、システムの独立度が他の演奏者よりも高い状況では、システムは自主的で表情豊かな演奏を行なうことにより、他の演奏者をリードする。

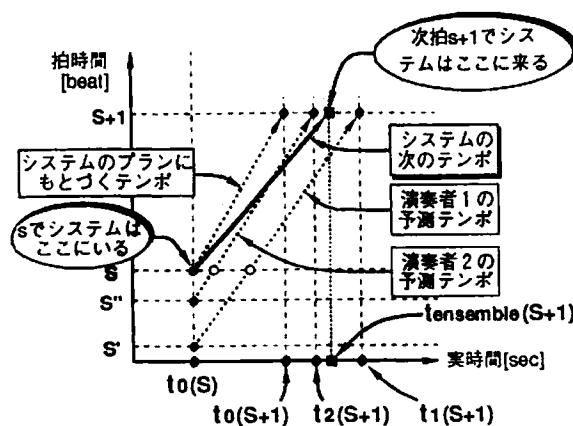


図 2 コンピュータのテンポ制御

人間の演奏者が 2 人の場合、傾きがテンポを表わしており、傾きが急なほどテンポが速く、緩やかなほどテンポが遅い。

3.3.3 アンサンブル時間

「アンサンブル時間」とは全演奏者 (コンピュータを含む) の演奏時刻を独立度により重み付けした平均 (重心) の時刻であり、直感的にはアンサンブル全体の演奏時刻と考えられる。ここで各演奏者の演奏時刻とは実際に演奏された時刻を実時間で表わしたものであり、秒を単位とする。これに対して拍時刻とは楽譜上の位置 (仮想時間) を表わしており、拍を単位とする。ある拍時刻 s における人間の各演奏者の s での演奏時刻を $t_i(s)$ ($i = 1..n$ ここで n は人間の演奏者の数), システムの s での演奏時刻を $t_0(s)$, s での各演奏者の独立度を $Indep_i(s)$ ($i = 1..n$), システムの独立度を $Indep_0(s)$ とするとき, s におけるアンサンブル時間 $t_{ensemble}(s)$ を以下のように定義する。

$$t_{ensemble}(s) = \frac{\sum_{i=0}^n \frac{Indep_i(s)}{\sum_{j=0}^n Indep_j(s)} t_i(s)}{\sum_{j=0}^n Indep_j(s)} \quad (1)$$

この式の意味するところは、アンサンブル時間は独立度の高い演奏者の s での演奏時刻に近付くが、それ以外の演奏者の演奏からも独立度に応じて影響を受けるということである。また、コンピュータの独立度が他の演奏者よりも高ければ、アンサンブル時間はコンピュータ自身の演奏時刻に近付くことになる。

3.3.4 コンピュータの演奏制御

コンピュータの演奏制御について、図 2 を用いて説明する。この図では簡単のため、人間の演奏者を 2 人としている。この図は実時刻 $t_0(s)$ にコンピュータが拍時刻 s を演奏し、演奏者 1 は s' 、演奏者 2 は s'' を演奏している状況を表わしている。コンピュータは演奏を続けるため、次のテンポ、すなわち拍時刻 $s+1$ を演奏すべき実時刻を計算する必要があるが、その時刻として拍時刻 $s+1$ におけるアンサンブル時間 $t_{ensemble}(s+1)$ を適用する。 $t_{ensemble}(s+1)$ は以下の式で表わされる。

$$t_{ensemble}(s+1) = \sum_{i=0}^n \frac{Indep_i(s+1)}{\sum_{j=0}^n Indep_j(s+1)} t_i(s+1) \quad (2)$$

このアンサンブル時間を計算するため、全演奏者（コンピュータを含む）の $s+1$ での（予測）演奏時刻 $t_i(s+1)$ ($i=0..n$) が必要となる。 $s+1$ での人間の演奏者の演奏時刻 $t_i(s+1)$ ($i=1..n$) は、過去数拍の演奏履歴から予測する。このとき現在に近いものほど重み付けをする。コンピュータの $s+1$ での演奏時刻 $t_0(s+1)$ も過去数拍の演奏履歴から予測できるが、そのとき、演奏プランのグローバル・テンポの影響を考慮するため、式 (3) により計算する。

$$t_0(s+1) = t_0(s) + \delta \frac{1}{V_{plan}(s)} + (1-\delta) \frac{1}{V_{estimate}(s)} \quad (3)$$

ここで $V_{plan}(s)$ は、拍 s から $s+1$ までのシステムの演奏プランのグローバル・テンポ（単位は [beat/sec]）を表わしており、 $V_{estimate}(s)$ は過去数拍の演奏履歴から重み付け平均で推定したテンポである。 δ はどの程度演奏プランの影響を受けるかを表わす定数で、現在 1/2 としている。

このように計算したアンサンブル時間 $t_{ensemble}(s+1)$ を用いて、コンピュータの拍 s から $s+1$ までの演奏テンポ $V_0(s)$ は以下のように計算される。

$$V_0(s) = \frac{1}{t_{ensemble}(s+1) - t_0(s)} \quad (4)$$

基本的には上述の考えでコンピュータは演奏を

続けるが、いくつか例外がある。もし、式 (4) で $t_{ensemble}(s+1) < t_0(s)$ となり、計算されたテンポが負になるときは、コンピュータは各演奏者の演奏が追い付いてきて、そのテンポが正になるまで演奏を待つ。これは人間の演奏者がコンピュータよりも大きく遅れた状況を意味するが、通常はすぐに各演奏者の演奏が入力され、コンピュータは演奏を再開することができる。

システム以外の演奏者の独立度がすべて 0 のときは、上記の式に従って演奏を行なうことにより、演奏プランに従って自主的な演奏を行なうことができる。

式 (2) の例外的な処理として、拍時刻 s におけるコンピュータの独立度が 0 のとき（コンピュータが人間の演奏者に縦の線を合わせる必要があるとき）、コンピュータは人間の演奏者の（独立度が 0 でない）誰かが最初に s の音符を演奏した直後にその音符を演奏する。このときの処理は、コンピュータの演奏が他の演奏者よりも遅れている場合には、その拍時間を s にジャンプし、逆に、コンピュータが他の演奏者よりも進んでいる場合には、誰かが追い付いて、その音符を演奏するまで待つことにより実現する。

基本的には上述の処理でシステムのテンポ修正を行なうが、人間の演奏者が演奏ミスをした場合の処理についても述べる必要がある。3.2 節で述べたように、各演奏者の演奏の信頼度は Listener から Performer へ送られるスコア情報により認識できる。例えば、ある演奏者が演奏ミスをしたとき、その演奏者のスコアはミスの度合いに応じて低くなっている。そのとき、演奏ミスをした演奏者の演奏テンポはあまり信頼がおけないと考えられるので、このような場合、システムがその演奏者の不確実な演奏に影響を受けてテンポを変化させるよりは、その演奏者から受ける影響を小さくし、他の演奏者（ミスをしていない）に合わせた方がよい。そこで、スコアの低い演奏者の独立度をそのスコアに応じて低くする。独立度を変化させる以外は上記とまったく同様の処理を行なうことにより、システムはスコアの低い演奏者の演奏の影響を抑え、他の演奏者に合わせた演奏やシステムの演奏プランに従った演奏を行なうことが可能となる。その後、演奏ミスをした演奏者が正しい演奏を継続すれば、その演奏者のスコアは高くなり、結果として独立度も高くなり、再びシステムの演奏に影響を与える

ようになる。

3.4 アンサンブル・システムの実装

これらの考えに基づく実験的なアンサンブル・システムを Listener と Performer に相当する 2 台の Macintosh 上に実装した。このシステムは 2 人の人間とアンサンブルを行なうことができ、独立度の変化に応じて人間の演奏者に対する追従法の変化が確認された。

4 おわりに

本論文では人間との自然なアンサンブルを実現するため、以前我々が提案した伴奏システムを複数の人間の演奏者と協調して演奏を行なうアンサンブル・システムへと拡張した。アンサンブル・システムは演奏プラン (コンピュータの演奏意志) を持ち、状況に従って変化する全演奏者 (コンピュータを含む) の独立度に応じて、すべての人間の演奏を考慮しながら、その演奏と協調するようにコンピュータ自身の演奏を制御する (アンサンブル時間の考えに基づく) ことが可能である。

今後の研究課題として、本論文ではテンポ制御についてしか述べなかったが、ダイナミクス (音量) などの演奏ニュアンスも状況に応じて変化させることが考えられる。そのときにも独立度の考えにより、複数の人間とシステムとの音量のバランスを考慮した制御を行なうことを考えている。また、現在は人間が与えている演奏プランや独立度を楽譜から自動 / 半自動で作成したり、あるいはリハーサルを行なうことにより学習することが考えられる。

参考文献

- [1] Baird, B. : The Artificially Intelligent Computer Performer and Parallel Processing. *Proceedings of International Computer Music Conference*, 1991, pp. 340-343.
- [2] Baird, B., Blevins, D., and Zahler, N. : The Artificially Intelligent Computer Performer on the Macintosh II and a Pattern Matching Algorithm for Real-time Interactive Performance, *Proceedings of International Computer Music Conference*, 1989, pp. 13-16.
- [3] Baird, B., Blevins, D., and Zahler, N. : Artificial Intelligence and Music: Implementing an Interactive Computer Performer, *Computer Music Journal*, Vol. 17, No. 2 (1993), pp. 73-79. (Summer 1993).
- [4] Bloch, J. J. and Dannenberg, R. B. : Real-Time Computer Accompaniment of Keyboard Performances, *Proceedings of International Computer Music Conference*, 1985, pp. 279-289.
- [5] Dannenberg, R. B. : Real-Time Scheduling and Computer Accompaniment, *Current Directions in Computer Music Research* (Mathews, M. V. and Pierce, J. R. (eds.)), System Development Foundation Benchmark Series, No. 2, MIT Press, 1989, chapter 18, pp. 225-262.
- [6] Dannenberg, R. B. and Mont-Reynaud, B. : Following an Improvisation in Real-Time, *Proceedings of International Computer Music Conference*, 1987, pp. 241-248.
- [7] Dannenberg, R. B. : An On-Line Algorithm for Real-Time Accompaniment, *Proceedings of International Computer Music Conference*, 1984, pp. 193-198.
- [8] Dannenberg, R. B. and Mukaino, H. : New Techniques for Enhanced Quality of Computer Accompaniment, *Proceedings of International Computer Music Conference*, 1988, pp. 243-249.
- [9] Horiuchi, Y. and Tanaka, H. : A Computer Accompaniment System With Independence, *Proceedings of International Computer Music Conference*, September 1993, pp. 418-420. at Waseda University.
- [10] Inoue, W., Hashimoto, S., and Ohteru, S. : A Computer Music System for Human Singing, *Proceedings of International Computer Music Conference*, 1993, pp. 150-153.
- [11] Katayose, H., Kanamori, T., Kamei, K., Nagashima, Y., Sato, K., Inokuchi, S., and Simura, S. : Virtual Performer, *Proceedings of International Computer Music Conference*, 1993, pp. 138-145.
- [12] Rowe, R. : *Interactive Music Systems: Machine Listening and Composing*, The MIT Press, 1993.
- [13] Vercoe, B. : The Synthetic Performer in the Context of Live Performance, *Proceedings of International Computer Music Conference*, 1984, pp. 199-200.
- [14] Vercoe, B. and Puckette, M. : Synthetic Rehearsal: Training the Synthetic Performer, *Proceedings of International Computer Music Conference*, 1985, pp. 275-278.
- [15] Sloboda, J. A. : 演奏, 音楽の心理学 (Deutsch, D. (ed.)), 西村書店, 1987, chapter 16, pp. 587-609.
- [16] 田口友康ほか : 音楽情報処理の技術的基盤, 平成 4 年度文部省科学研究費総合研究 (B) 「音楽情報科学に関する総合的研究」 (課題番号 04352030, 研究代表者: 田口友康), 1993.
- [17] 平賀譲 : コンピュータ音楽 (音楽情報処理), コンピュータソフトウェア, Vol. 11, No. 1 (1994), pp. 49-56.
- [18] 堀内靖雄, 橋本周司 : 伴奏システム, 情報処理, Vol. 35, No. 9 (1994), pp. 815-821.