Vol.2014-DCC-8 No.14
Vol.2014-MUS-105 No.14
2014/11/21

# 実時間音響入力を備えたマルチエージェント協調演奏システム

渡辺和泉[†1]　中村亮介[†1]　金丸悠理[†1]
後藤敏行[†1]　田村直良[†1]　島田広[†2]

筆者らは，MIDI シーケンサを備え，独立に演奏を模擬する複数の演奏エージェントで構成されたマルチエージェント協調演奏システムの開発を進めている．本発表では，入力された歌声や楽器音などの音響信号を認識し，テンポ，演奏位置，音高のズレなどの演奏情報を解析して，MIDI 信号として出力する手法について報告する．さらに，開発した音響信号モジュール，MusicXML から装飾記号を埋め込んだ拡張 MIDI 生成モジュール，演奏エージェントモジュールを結合させた協調演奏システムの構成について述べる．これにより，テンポだけでなく，移調などによる音高変化にも追随する協調演奏が可能であることを確認した．

# Automated Music Performance System by Real-time Acoustic Input Based on Multiple Agent Simulation

IZUMI WATANABE[†1]　RYOSUKE NAKAMURA[†1]　YURI KANAMARU[†1]
TOSHIYUKI GOTOH[†1]　NAOYOSI TAMURA[†1]　HIROSHI SHIMADA[†2]

We have developed an automated musical performance system based on multiple agent simulations, which enable us to play music with multiple agents and real users. In this paper, we describe a method which recognizes the inputted sound signals of human voices and musical instrument sounds, and analyzes the performance information, in order to generate the MIDI sequences with a pitch bend. The configuration of prototype system, which consists of acoustic and music analysis, expanded MIDI transform, and musical performance agent modules, is also discussed.

## 1. はじめに

筆者らは，オーケストラ演奏における奏者同士の関わり合いをマルチエージェントによりモデル化することで，ユーザと計算機内部の複数の演奏者エージェントが相互に演奏情報を交換しながら実時間で協調して演奏する仮想空間オーケストラ・システムの開発を進めている[1]．これまで，このシステムではユーザの演奏情報の入力手段として，MIDI コントローラを用いていた．本研究では，マイクから得た歌声や楽器音などの音響信号を入力とすることによって，MIDI 楽器の種類による制約を無くし，さまざまな楽器への対応を目指している．これまで，音響信号に対する楽譜追跡やテンポ追従に関する研究[2-5]は多いが，音高の外れや移調などに対応できる方法は少ない．本稿では，ユーザが異なる音高で演奏した場合でも楽譜追跡を行い，音高と音高のずれをピッチベンドとして分離して解析することが可能な音響信号入力法，および，これを用いたマルチエージェント協調演奏システムの構成について述べる．

## 2. 仮想空間オーケストラ概要

図 1 は仮想空間オーケストラの全体構想を示したものである．仮想空間オーケストラでは，最初に電子楽譜のデファクトスタンダードとなっている MusicXML から，各種の

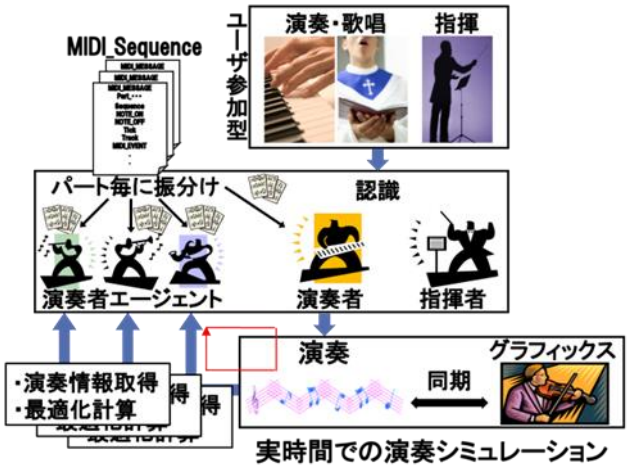


図 1　仮想空間オーケストラの全体構想

装飾記号の情報を埋め込んだ拡張 MIDI シーケンスをあらかじめ自動的に生成する．それを楽器パートごとにシーケンスに分割した上で，それぞれのパートを演奏する演奏者エージェントに振り分ける．各演奏者エージェントはそれぞれ独立した MIDI シーケンサを備えており，ピッチ，テンポ，演奏位置など演奏情報を独立に設定しながら演奏ができる．また，パートを担当する任意のエージェントを実ユーザ（ユーザ演奏入力）に置き換えることによって，複数の実ユーザの参加も可能になる．

各エージェントは共有メモリを介して他のエージェント（実ユーザも含む）の演奏情報を参照するとともに，自

†1　横浜国立大学大学院環境情報研究院／学府
Graduate School of Environment and Information Sciences, Yokohama National University
†2　横浜国立大学教育人間科学部
Faculty of Education and Human Sciences, Yokohama National University

Vol.2014-DCC-8 No.14
Vol.2014-MUS-105 No.14
2014/11/21

分自身の演奏情報を記録する．各エージェントは自分と他のエージェントの演奏情報に基づいて，各自が独立に最適になるように演奏を制御し，MIDI シンセサイザを通して演奏を行う．この「演奏情報取得→最適化計算→情報記録」という最適化の過程を繰り返すことによって演奏全体の最適化を図ることになる．図 1 の中の，指揮動作の自動認識インターフェイスや，演奏と同期したグラフィックス提示部については，今後の計画となっている．また，同図のなかで音響信号入力モジュールは，実ユーザによる音響信号から MIDI 信号に変換し，共有メモリを経由して各エージェントに情報を伝えるようになっている．

図 2 にモジュール間の楽譜・演奏情報の流れを示す．楽譜情報としては，MusicXML で記述されたスコア譜を用いる．これをパート譜に分割して，装飾記号の情報を埋め込んだ拡張 MIDI シーケンスを生成する．一方，音響信号入力部では，オーディオインタフェース経由で音響信号を入力し，MusicXML を用いて楽譜追跡を行い MIDI シーケンスを生成する．この際，原楽譜と音響信号に音高の外れがある場合には，ノートオン信号に加えて，ピッチベンド信号を付加する．これらの MIDI 信号を入力として，各エージェントが実時間で動作しながら演奏を進めることになる．

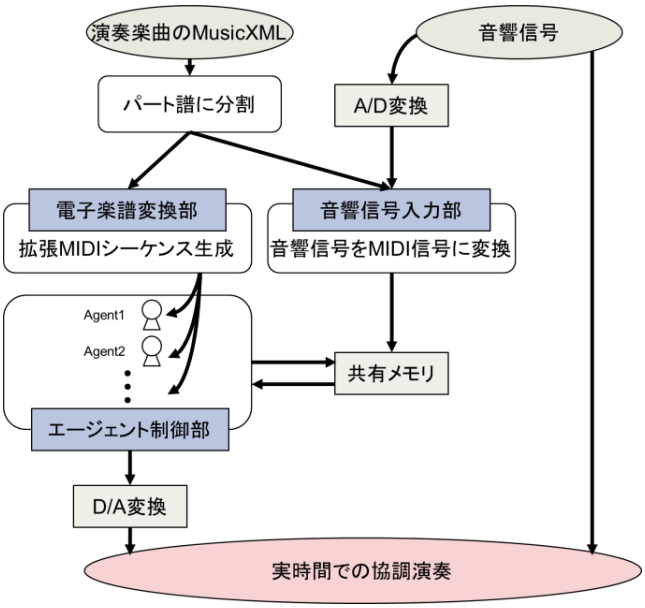


図 2　演奏情報の処理の流れ

## 3. 音響信号入力モジュール

### 3.1 処理の概要

音響信号入力モジュールでは，マイクで入力された音響信号に対して，楽譜上の音符から音符への遷移タイミング（以下，遷移時刻と呼ぶ）を実時間で推定することによって，楽譜追跡を行う．本研究では，音高の外れや移調などに対応することを目標とするが，音高の外れを許容した場合に，弾き直しにともなう楽譜位置の同定は困難になる．そこで，今回の研究では音高外れの検出を行い，検出された場合にはその小節から弾き直しを促すこととした．

遷移時刻は，楽譜情報の各音符の音価から生成する遷移確率分布（楽譜情報由来の遷移確率）と入力の音響信号をもとに解析した音高及び音量の変化量から算出した遷移確率（音響情報由来の遷移確率）を重ね合わせることによって推定する．遷移時刻と判定されるとノートオン信号が共有メモリに送られ，それぞれの演奏者エージェントに実ユーザの演奏が次の音符へ推移したことを知らせる．一方，演奏誤りなどの突発的な音高の違いは無視しつつ，キー変更などの定常的な音高の違いを認識して，その音高に合わせたベンド信号を一定間隔で送信することで，演奏者エージェントの楽譜追跡には音高的な調和に不整合が生じないように工夫している（図 3）．

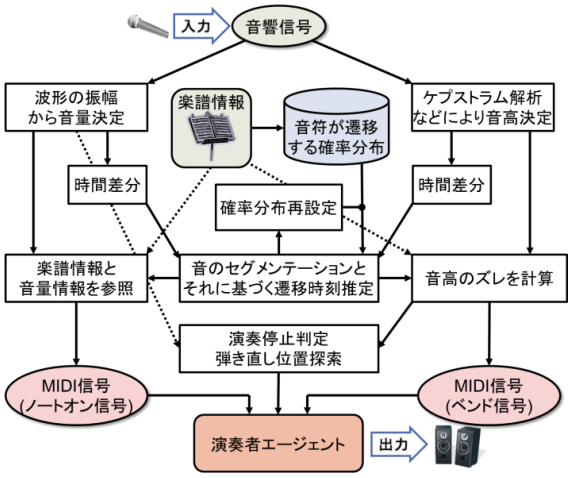


図 3　音響信号入力モジュールの処理の流れ

### 3.2 遷移時刻の推定

$n$番目の音符を演奏している時，遷移時刻$T_n$は楽譜情報由来の遷移確率$P_{score_n}(t)$と音響情報由来の遷移確率$P_{audio_n}(t)$を掛け合わせた確率$P_{next_n}(t)$が最大となる時刻$t$と仮定する（図 4）．ここで，$t$は$n$番目の音符が発音されてからの時刻とする．楽譜情報から得られた原曲のテンポに対する相対的なテンポを$r_n$（なお$r_1 = 1.0$とする），テンポ推定に用いる重み係数を$a$，原曲テンポで演奏した場合の音長を$\tau_n$，実際のテンポから推測される音長を$\hat{\tau}_n$（なお$\hat{\tau}_1 = \tau_1$とする），音高の変化量によって決定される遷移確率を$P_{pitch}(t)$，音量の変化量によって決定される遷移確率を$P_{volume}(t)$，$P_{audio}(t)$算出の際の$P_{pitch}(t)$に対する重み係数を$\alpha$とすると，以下のように定式化できる．

$$\hat{\tau}_n = r_n \tau_n \tag{1}$$

$$r_n = a \cdot r_{n-1} + (1-a) \cdot T_{n-1}/\tau_{n-1} \tag{2}$$

$$P_{score_n}(t) = \mathcal{N}(\log(t); \hat{\tau}_n, \Sigma) \tag{3}$$

$$P_{audio_n}(t) = \alpha \cdot P_{pitch}(t) + (1-\alpha) \cdot P_{volume}(t) \tag{4}$$

$$P_{next_n}(t) = P_{score_n}(t) \cdot P_{audio_n}(t) \tag{5}$$

ここで，$P_{score_n}(t)$は式(3)の通り，対数正規分布でモデル化できるものと仮定する．一般に，フェルマータなどの有無に関わらず演奏中の音符を引き延ばして演奏しても演奏が破綻しにくい．一方，早めに次の音符を演奏する場合は音楽的に不自然になる．このことから，確率変数である時間の対数を取った正規分布でより自然にモデル化できるものと考えた．

以上の式に基づいて$T_n$を求めるが，ノートオン信号を送信する時刻は$\hat{\tau}_n$とし$n+1$番目の音符を1/5を演奏した時点で$T_n$を推定する．

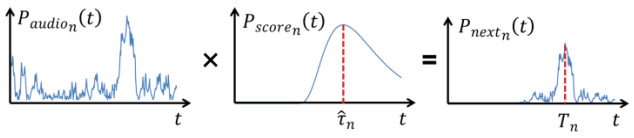


図 4　遷移確率掛け合わせのイメージ

### 3.3 ノートオン信号の生成

前節の通り，$\hat{\tau}_n$になった時に$n+1$番目の音符のノートオン信号を送信するが，ノートオン信号に含まれる音高を示すノートナンバは楽譜情報を参照し，音量を示すベロシティは直近 10 サンプルの音響信号の振幅の平均を参照する．このように音高については演奏された音高に関わらず楽譜上の音高がノートナンバとなるため，演奏者エージェントによる演奏位置の把握が容易になる．

### 3.4 演奏誤りとキー変更の判別

演奏誤りとキー変更はどちらも楽譜上の音高とは違う音を演奏することになるが，楽譜と実際に演奏された音高の差の時間的な分散に注目すると，演奏誤りは分散が大きく，キー変更は分散が小さい値となる．本手法では，頻度が多い音符や，音価の長い音符を楽譜情報から注目して判定することで，楽譜追跡の揺らぎに基づく不安定性を回避している. また, エージェントに対するキー変更の情報は，ピッチベンド信号を利用するが，各エージェントは実ユーザによるベンド信号に応じて，それぞれの担当パートの音高を変化させることになる．

### 3.5 弾き直し位置の探索

ユーザが誤って演奏を行った上で演奏を中止した場合，演奏誤りが検出された小節から演奏を始めることを促すこととした．ここでは，楽譜上で音符がある演奏位置において，入力の音響信号が閾値以下の場合に演奏を中断する．また，演奏に誤りがないにも関わらず演奏を中断した場合には，その小節から演奏を始めるか任意の別の小節から始めるかの選択ができる．

## 4. 複数エージェントを用いた演奏制御

### 4.1 エージェントの楽譜情報の扱い

本システムでは，演奏者エージェントが演奏する情報は電子楽譜の実質標準である MusicXML から変換した MIDI シーケンスを用いる．一般に，MIDI 信号では，演奏情報は特定のピッチ，ベロシティ，演奏位置(ティック)で音を鳴らすノートオンと，音を消すノートオフの組み合わせで物理的な時刻を用いて音楽を表している．一方，トリル，トレモロ，モンデルトなどのように，打鍵回数や打鍵時間などが演奏者の裁量に任される奏法も多い．これらは，一般に楽譜上では装飾記号や装飾音符などとして表されており，MIDI 信号の生成時には固定値に設定されることになる．このため，実ユーザの演奏と MIDI 信号の記述が合わないために楽譜追跡が破綻する可能性が出てくる．この問題に対して，本システムでは MIDI シーケンスを利用しながらも，装飾記号をメタメッセージの中に埋め込むことよって対応している．これによって，MIDI の長所を利用しつつ，システム側で演奏中に装飾記号の存在を知ることが可能になる（図 5）．本研究では，この拡張 MIDI ファイル形式を MusicXML から自動生成するプログラムの開発も行っている[1].

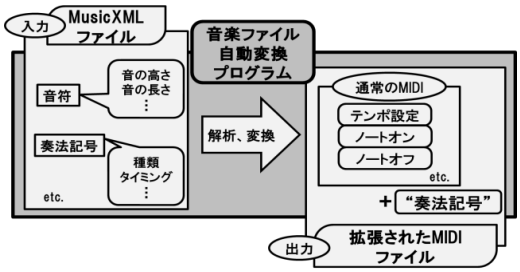


図 5　MusicXML 変換プログラム

### 4.2 エージェントの演奏位置決定方法

MIDI シーケンスに対する弾き直しや弾き飛ばしに頑健な楽譜追跡法として，確率モデルに基づく楽譜追跡法が提案されている[6]．この手法は，直前の演奏位置から次の音符に遷移する確率（状態遷移確率）と，特定の音の高さの音符が出力される確率（出力確率）からなる確率モデルをもとに，各音符の最尤値を求めることによって演奏位置を決定する．さらに，確率モデルの遷移確率を装飾記号に応じて切り替えることで，装飾音や装飾記号などに係わる楽譜追跡の問題に対応する手法が提案されている[7].

また，筆者らは，確率モデルの切り替えに，前節で述べた拡張型 MIDI シーケンスを用いる方法を開発している[1]. ここでは，トリルなどの装飾記号が埋め込まれた MIDI 情報を参照して，確率モデルの自己遷移確率を高めることによって，同一音符への推定が可能になる．また，装飾記号が終了した場合には，同様に装飾記号の終了情報に基づいて遷移確率を戻すことによって通常演奏モードに戻る．

## 5. システム動作

本システムは，音響信号入力モジュール，拡張 MIDI 生成モジュール，演奏エージェントモジュールの 3 つの処理パートで構成されている. 音響信号入力モジュールは, C++

Vol.2014-DCC-8 No.14
Vol.2014-MUS-105 No.14
2014/11/21

で開発されており，オーディオインタフェース経由で音響信号を入力し，MIDI シーケンスを実時間で生成する．拡張 MIDI 生成モジュールは JAVA で開発され，MusicXML から装飾記号や装飾音の情報を埋め込んだ拡張 MIDI シーケンスをあらかじめ生成する（図 1）．また，演奏エージェントモジュールは JAVA で開発されており，上記の 2 種類の MIDI シーケンスを入力して，それぞれのパートが協調して演奏するようになっている．各モジュールは MIDI ケーブルや仮想 MIDI ケーブル経由で接続することで，複数の PC での協調動作も可能となっている．

従来の自動伴奏システムの多くが伴奏全体を一体として制御するのに対して，本システムでは伴奏を楽器パート毎に分割し，それぞれのエージェントによって独立に処理が行われる．これによって，また，エージェントを MIDI キーボードや音響入力モジュールと切り替えることで，複数のユーザの参加が可能になる（図 8）．また，ユーザのキーの変更に対して，自動的に伴奏を追従させることも可能である．

図 6　システムの外観

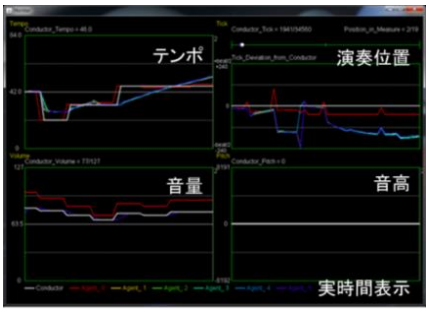


図 7　パラメータの実時間表示

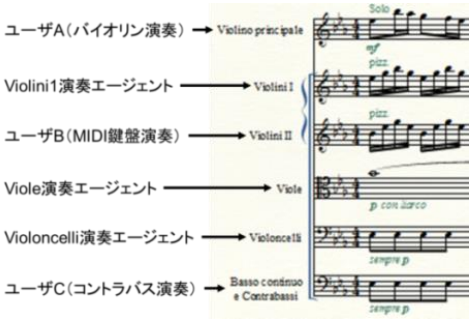


図 8　ユーザとエージェントの合奏例

## 6. 評価方法

現在，音響入力部の評価として，部分的に移調を行ったパート譜を用いて，別の PC で再生した MIDI 演奏を入力として，試作システムの動作確認を進めており，ほぼ安定な動作を確認した．システムの定量的な評価方法の検討や，実楽器に対する追従性の評価および改良については，今後継続して取組む予定である．

## 7. まとめ

本稿では，音響信号や MIDI デバイスの情報をもとに，実演奏者と PC 内の複数のエージェントが演奏情報を交換しながら協調演奏を行う仮想空間オーケストラ・システムについて報告した．また，音響信号をもとに演奏誤りと移調にともなうキー変更を判別しながら楽譜に追従し，テンポ，音高，音高外れを出力する方法について述べた．試作したシステムの定量的な評価や実際に演奏される楽器に対する追従性の検討が今後の課題である．

**謝辞**　横浜国大音楽教育課程の皆さんには実験や試用評価などでご協力頂いた．また，木更津高専の斎藤康之先生には楽譜追跡の技術動向についてご教示頂いた．関係各位に心から感謝する．

## 参考文献

[1] 中村亮介他：“複数の実ユーザとの共演が可能なマルチエージェント協調演奏システム”，映像情報メディア学会技術報告, Vol.38, No.9, pp.57-60(2014)
[2] Cont, A.: ANTESCOFO: “Anticipatory Synchronization and Control of Interactive Parameters in Computer Music”, Proc. ICMC, (2008).
[3] 中村友彦他：“音楽演奏の誤りや反復に頑健な音響入力自動伴奏”，日本音響学会 2012 年秋季研究発表会，pp. 931-934, (2012).
[4] 酒向慎司他：“Ryry：弾き飛ばし・弾き直しを含む演奏に追従する音響信号による自動伴奏システム”，情報処理学会技術報告, 2012-MUS-96, No.13(2012)
[5] 前澤陽他:“結合動的モデルに基づく音響信号アライメント”，情報処理学会技術報告, 2014-MUS-104, No.13(2014)
[6] 中村栄太他：“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)
[7] 中村栄太他：“多声 MIDI 演奏の楽譜追跡における演奏の即興性のモデル化と自動伴奏への応用”，情報処理学会技術報告, 2012-MUS-96, No.14(2012)