

# 演奏者に独自のコード進行とテンポを許す セッションシステム

柏崎 紘一, 山本 欧

東京電機大学大学院工学研究科  
〒101-0054 東京都千代田区神田錦町 2-2

{kashi, ou}@ps.d.dendai.ac.jp

あらまし : 本研究では, 広域ネットワークを経由し, 遠隔地間で音楽セッションを行うシステムを提案する. ネットワークの遅延を考慮した従来のセッション・システムでは, 奏者は相手の演奏を一定の周期遅らせて聴きながら演奏を行う. これらのシステムでは, 演奏の調和を得るために, 奏者はお互い共通のコード進行やテンポのもとで演奏を行う. これに対し, 本研究では, 各奏者が独自のコード進行とテンポで演奏しつつセッションが可能なシステムを提案する. それぞれの奏者の演奏は相手側のコードやテンポに合わせるような情報に変換され相手側に送信される. そのため, お互いは全く別の曲を演奏しているのにも関わらず, 相手が自分の演奏に合わせて演奏しているように感じる. 本システムによって, 奏者は曲やテンポの制約から解放され, 自由な形でネットワーク上のセッションに参加することができる.

## A Session System allowing individual chord progress and tempo

Kouichi Kashiwazaki, Ou Yamamoto

Tokyo Denki University  
2-2 Nishiki-chou, Chiyoda-ku, Tokyo, 101-0054

{kashi, ou}@ps.d.dendai.ac.jp

Abstract : In this paper, we propose a system which provides a musical session environment on the Internet. So far several musical session systems assuming network delay have been proposed. Those systems naturally assume that all players play a tune in concert. Our system, however, allows players to take their own tempo and chord progress. That is, in our system, each player can play not only the same tune, but also different tune individually. Since each player's performance data is converted into a normal form by the system and then sent to network, the normal form data can be converted into matched performance data for each receiver-side performance. In our system, players can freely participate in a session performance on the Internet.

# 1 はじめに

従来、音楽におけるセッションは、物理的に同一の時間と空間とを共有して成り立っていた。しかし、広域ネットワークの普及とその高速化によって、音楽の分野においてもネットワーク上でセッションを行う試みが多数なされるようになった。しかし、その場合に問題となるのが、ネットワークの遅延である。この遅延により、お互いの演奏が遅れて届き、それぞれの奏者が同時刻に同じ音を聴きながら演奏することが不可能となる。

そこで近年、ネットワークの遅延を考慮したセッションシステムの研究がなされている [1] [2] [3]。これらは、お互いの演奏を一定の周期遅らせる形で聴きあい、それに合わせて自分の演奏を行うシステムである。これらのシステムでは、遅れて届くお互いの演奏を音楽的に調和させるために、各奏者は同一のコード進行とテンポで演奏を行っていた。

これに対し、本研究では、各奏者に独自のコード進行とテンポで演奏することを許すシステムを提案する。本システムでは、各奏者の演奏データは、コード上における調性を示す中間形式に変換されて送信される。そして受信側において、受信者の演奏のコード進行とテンポに合うように再変換された上で再生される。すなわち、互いに別の曲を演奏していても、相手が自分の演奏に合わせて演奏しているように感じられる。本システムによって、奏者は曲やテンポの制約から解放され、自由な形でネットワーク上のセッションに参加することができる。

以下、2章で本システムの詳細を述べ、3章で評価について述べる。最後に4章で結論と今後の展望について述べる。

## 2 システムの詳細

本システムの構成を図1に示す。本システムは、プログラミング環境 Max/MSP [4] を用いて、Macintosh 上に実装した。

演奏者は MIDI 楽器を用い、それぞれ独自のコード進行とテンポの下で演奏を行う。その演奏情報は、変換テーブルによってコード上の調性を示す中間形式に変換される。その後、演奏された拍のタイムスタンプがつけられ、UDP を用いた OpenSoundControl (以下 OSC) プロトコル [5] によって相手

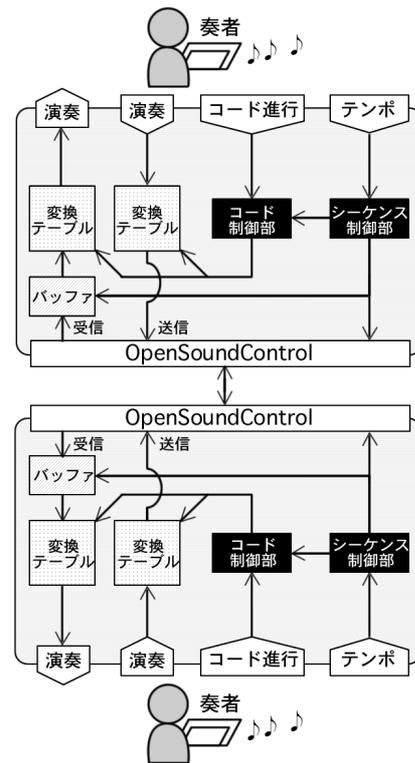


図 1: 本システムの構成

側に送信される。

受信側では、受信した演奏情報を一旦バッファに貯え、タイムスタンプを元に読み出す。そして、受信側のコード進行に調和するように変換され、実際の演奏として再生される。このときの演奏の読み出されるテンポは、送信側のテンポと受信側の演奏のテンポを比較した上で、受信側と同期のとれるようなテンポで再生される。

以下、本章では演奏情報の変換とテンポの同期アルゴリズムについて解説する。

### 2.1 演奏情報の変換

本システムでは、各奏者に独自のコード進行の下での演奏を許している。そのため、演奏情報をそのまま送信したのでは、自分自身のコード進行では調性のあった演奏でも、相手側では調性がずれてしまう。つまり、本システムにおいて重要となるのは、どの音が演奏されたかという情報ではなく、その音がコード上でどのような調性を持っているのかという情報である。

key : C Major ダイアトニック・コード

図 2: ダイアトニック・コードと Mode Scale

そこで、本システムでは、各奏者の演奏情報を、その演奏が行われたコードの Mode Scale における位置の情報に変換する。Mode Scale とは、図 2 に示すように、ある key の中で調性の合った音のコード(ダイアトニック・コード)の主音を中心としたスケールのことである。ダイアトニック・コードごとに、音の配列が異なる 7 種類の Mode Scale がある。Mode Scale 上の音には、表 1 に示すように、コードに対してどのように協和するのかが決まっている。

表 1: Mode Scale と中間形式の関係

協和の度合い	Mode Scale 上の位置	中間形式
完全協和	コードの主音である第 1 音	C
不完全協和	Mode Scale 上の第 2 音	D
完全協和	コードの構成音である第 3 音	E
不完全協和	Mode Scale 上の第 4 音	F
完全協和	コードの構成音である第 5 音	G
不完全協和	Mode Scale 上の第 6 音	A
完全協和	コードの構成音である第 7 音	B
不協和	Mode Scale 外の音	C#, D# F#, G# A#

本システムでは、各奏者の演奏から Mode Scale 内の位置を求め、それらを中間形式として C Ionian Scale の音名に変換する。上述のように、Mode Scale は 7 種類あるので、それぞれに対応した変換テーブルを用意してある。この変換テーブルの考え方は、西本による音機能固定マッピング楽器 [6] の考えを参考にした。

変換の手順を図 3 に示す。ユーザーの演奏は MIDI のノートナンバーから、現在のコード上の主音に対する距離に置き換えられる。そして、上述の変換テーブルによって中間形式に変換され、相手側に送信される。送信された中間形式は、相手側のコードの変換テーブルを経て、そのコードの主音を中心とした音階に置き直される。そして、実際の演奏として相手側で再生される。

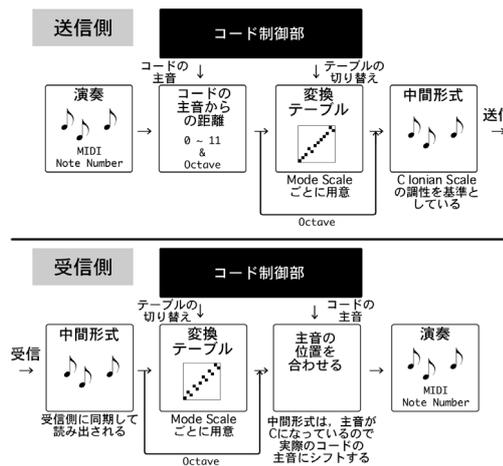


図 3: 演奏情報の中間形式への変換手順

例として、送信側の key が C Major でコードが Dm7、受信側の key が E Minor でコードが F#m7 であるとする。送信側の Mode Scale は D Dorian Scale で、受信側は F# Locrian Scale である。この時の変換テーブルおよび変換の様子を図 4 に示す。

今、送信側が D (コードの主音), E (Mode Scale 上で主音の次に高い音), F (Mode Scale 外の音) の 3 つの音を演奏したとする。これらの音は、図 4 に示す変換テーブルによって、それぞれ C, D, D# という中間形式の音に変換され送信される。受信側では C, D, D# が受信側の変換テーブルによ

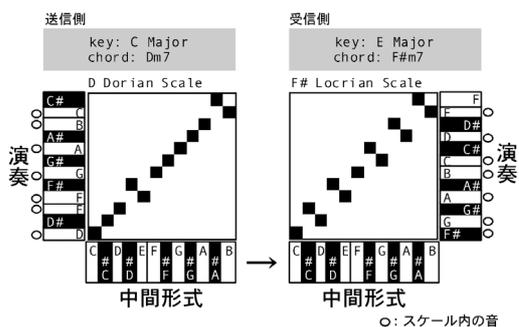


図 4: 演奏の変換の例

て F# (受信側のコードの主音), G (Mode Scale 上で主音の次に高い音), A# (Mode Scale 外の音) に変換される。

## 2.2 テンポの同期

この節では異なるテンポの同期のためのアルゴリズムについて述べる。

各奏者にテンポの変更を許した場合、受信した演奏の再生を強制的に受信者側のテンポに合わせてしまうと、演奏情報が追いつかない場合が生じてしまう。例を図 5 に示す。図 5 において、奏者 B は奏者 A の 2 倍のテンポで演奏を行っている。奏者 B が受信した奏者 A の演奏は、奏者 B のテンポに合わせて再生されているため、時刻 t1 において奏者 B が A4 部分の演奏を必要としているのに、A4 部分はまだ演奏されていない。

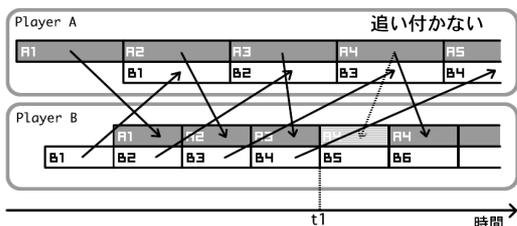


図 5: テンポが追いつかない場合

そこで本システムでは、受信した演奏を強制的に受信側のテンポに合わせて再生するのではなく、受信側の演奏と一定周期で同期可能なテンポで再生する方法をとっている。図 6 に、一定周期で同期のとれるテンポの組み合わせを示す。例えば、受信側が

BPM=120, 送信側が BPM=90 のテンポで演奏している場合、両者の比は 4:3 であるので、受信側が 4 小節演奏する間に送信側は 3 小節演奏され、互いに次の小節の先頭で同期することになる。両者の比が図 6 の比に正確に一致しない場合、図 5 に示したようなテンポが追いつかない状態にならないような比を選択する。

例として、受信側が BPM=124, 送信側が BPM=85 のテンポで演奏している場合、両者の比を 4:3 としてしまうと、送信側の演奏は受信側では  $124 \times (3/4) = 93\text{BPM}$  で再生されてしまい、受信側に追いつかなくなってしまう。したがって、この場合は BPM=85 以下のテンポで演奏可能な比である 3:2 が選択され、送信側の演奏は受信側において  $124 \times (2/3) = 82.67\text{BPM}$  で再生される。

この同期アルゴリズムでは、互いのテンポが等しい場合を除き、1 小節毎に同期することはない。しかし、複数小節ごとに同期する場合でも、お互いの演奏が一種のポリリズムを形成するので、不自然な演奏となることは避けられる。

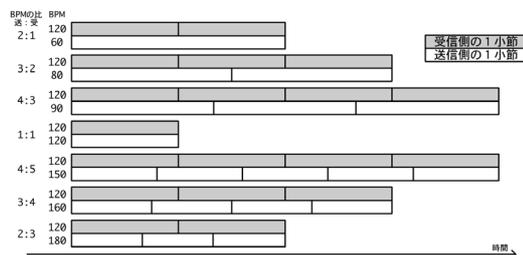


図 6: 同期のとれるテンポの比

## 2.3 ユーザ・インターフェイス

本システムのユーザ・インターフェイス画面を図 7 に示す。まずユーザは、画面中の数字アイコンによって伴奏を設定する。あらかじめ、それぞれのアイコンに伴奏として演奏する音を割り当てておき、1 小節分の演奏場所を示すフィールドの中で演奏するタイミングを設定する。アイコンに割り当てる音は、上述の中間形式の情報であるので、伴奏は 1 小節分の設定であっても、実際に演奏される音はコード進行に従って変化していく。伴奏の設定後、その伴奏に合わせて、MIDI 楽器による演奏を行う。伴奏と MIDI 楽器の入力で演奏される音源と音色は、

各奏者が任意に設定できる。

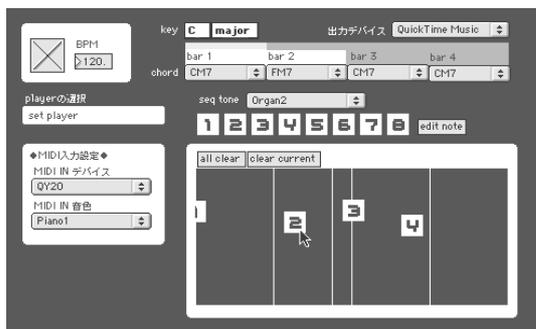


図 7: 実際のインターフェイス

### 3 評価

本システムを使用したセッション演奏の実験を、ADSL 回線で接続された埼玉県蕨市と東京都府中市間の 2 地点で行った。演奏実験に先立ち、ネットワークの遅延時間を測定した。これは、両地点で本システムを起動し、演奏データの往復所要時間を測定することで行った。演奏データを 20 回送信した場合の測定値の平均は 152.5ms であった。

本システムによりセッション演奏をする場合、まず両地点でシステムを起動し、お互いの IP アドレスを設定する。接続を確認した後、伴奏のシーケンスを走らせると、相手からの演奏が受信可能な状態になる。本実験では、両者が伴奏シーケンスを走らせた状態で次のことを行った。

まず、両奏者のテンポは同一とし、各奏者が互いに異なるコード進行上でコードの構成音を演奏した。演奏のリズムは、相手側で認識しやすく 4 拍子のリズムで演奏を行った。その結果、正常に 4 拍子で各奏者のコードの構成音が再生された。次に、各奏者がテンポを任意に変更しながら演奏を行った。その結果、システムが拍の頭を合わせる間に少しリズムが崩れたが、その後は正常に再生された。

本実験により判明した問題点は次の 2 点である。

- 相手の演奏動作を視覚的に確認しながらの演奏ができない。これは、相手の音のみを聴いて演奏することに不慣れた奏者にとって演奏の妨げとなる。このため、相手の演奏情報を簡単な

アニメーション等で可視化し、受信者が相手の演奏を視覚的に確認しつつ演奏できるような機能を付加することで改善可能である。

- 本システムで通信に使用した OSC プロトコルは UDP による通信である。このため、高速な通信が可能である反面、確実な通信ができる保証がない。本実験では比較的高速な回線であったため、データの取りこぼしは確認されなかったが、遅延の大きい環境下では演奏データが正常に届かない可能性がある。このため、今後は TCP を使用した通信も考慮し、転送速度と確実な通信とのトレードオフを検討する必要がある。

### 4 結論

本研究では、各奏者が独自のコード進行とテンポで演奏しながら、ネットワークを介してセッション演奏可能なシステムを提案・実装した。実験による評価の結果、各奏者は同一曲だけでなく、互いに異なる曲を演奏しながら、ネットワーク上でセッション演奏できることが確認された。

今後の課題としては、評価により判明したものの意外に、次の 2 点が挙げられる。

- 現在のところ、本システムは 2 人の奏者によるセッション演奏しか許していない。今後、不特定多数のユーザーが参加できるようにシステムを拡張する必要がある。
- 演奏データを中間形式に変換するテーブルは、現時点では不完全であり、Mode Scale 内のコード構成音以外の音については、適切な変換がなされない。これらの音にはテンション音やアヴォイド音などの種類があり、コード上でそれぞれの役割がある。今後、これらの音を考慮した変換テーブルを作成し、より調和のとれたセッション演奏が可能となるようにする必要がある。

### 参考文献

- [1] Masataka Goto, Ryo Neyama, and Yoichi Muraoka: RMCP - Remote Music Control

- Protocol –Design and Applications–, Proceedings of the 1997 International Computer Music Conference, pp.446-449 (1997)
- [2] 後藤真孝, 根山亮: OpenRemoteGIG - 遅延を考慮した不特定多数による遠隔セッションシステム、情報処理学会論文誌 Vol.43, No.2, p299-309 (2002)
- [3] 長島洋一: GDS (Global Delayed Session) Music - ネットワーク遅延を伴う音楽セッションシステム, 第46回音楽情報科学研究会研究報告 (2002)
- [4] Max/MSP©1900-2002 Cycling'74/IRCAM : Miller Puckete, David Zicarelli
- [5] Matthew Wright, Adrian Freed: Open Sound Control - A New Protocol for Communicating with Sound Synthesizers, Proceedings of the 1997 International Computer Music Conference, pp.101-104 (1997)
- [6] 西本一志、渡邊洋、馬田一郎、間瀬健二、中津良平: 創造的音楽表現を可能とする音楽演奏支援手法の検討 - 音機能固定マッピング楽器の提案、情報処理学会論文誌 Vol.39, No.5, p1556-1567 (1998)