

Open RemoteGIG : 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

後藤 真孝[†] 根山 亮^{††}

本論文では、Internet のような広域ネットワークを経由して、不特定多数のユーザが遠隔地間でジャムセッションできるシステム Open RemoteGIG を提案する。本研究は、物理的に 1 カ所に集まらなくても、音楽的なインタラクションを楽しむことができるセッションを実現することを目的とする。従来、遅延の小さいネットワークを介した MIDI 中継は実現されていたが、遅延の大きい Internet 等を介した双方向の MIDI 中継によって、調性とリズムのある音楽を合奏することはできなかった。そこで本研究では、不可避な遅延を積極的に利用した、調性とリズムのある音楽のための新たな形態の遠隔セッションシステム Open RemoteGIG を実現した。Open RemoteGIG では、同一のコード進行（12 小節のブルース進行等）の繰返しを、テンポ一定で演奏することを前提として、遠隔地にいる複数の演奏者が Internet 経由のセッションを行う。演奏者は、お互いの演奏をコード進行の 1 周期の整数倍の時間だけ遅れて聞き合いながら、即興演奏する。コード進行は繰り返すため、遅延した演奏は再び同じコードとなり調和する。さらに、Open RemoteGIG は、演奏相手の発見環境や打合せ用のチャット機能も提供する。MIDI 中継は音楽用通信プロトコル RMCP に基づいて実装され、実際に遠隔地間で提案したセッションをできることが確認された。

Open RemoteGIG: An Open-to-the-public Distributed Session System Overcoming Network Latency

MASATAKA GOTO[†] and RYO NEYAMA^{††}

This paper describes a distributed session system called *Open RemoteGIG* which enables widely distributed musicians to join a worldwide jam session by using the Internet. Our goal is to have musicians participating in a truly interactive jam session without getting together physically. Although MIDI transmission via small-latency networks has been tried, performing a distributed jam session with tonality and rhythm has been considered difficult when we use a long-latency public network like the Internet. We therefore developed an innovative distributed session system for music with tonality and rhythm, *Open RemoteGIG*, which turns the long inevitable network latency to its advantage. In *Open RemoteGIG*, distributed musicians can interact over the Internet under the assumption that the tempo is constant and the chord progression is repetitive, like that of 12-bar blues. Each musician can improvise while listening to the others' performances that are intentionally delayed for multiples of the constant period of the repetitive chord progression; the delayed performance can fit the chords because the progression is repetitive. *Open RemoteGIG* also supports a musician-finding and group-chatting environment. The MIDI transmission was implemented by using a music network protocol called *RMCP* and we confirmed that the proposed distributed session was achieved.

1. はじめに

遠隔地間の人間同士のコミュニケーションは、Internet のような広域ネットワークやネットワークコン

ピューティングの普及にともない、専用回線や専用装置を用いずに実現できる環境が整いつつある。しかし、音楽による演奏者同士のコミュニケーションであるジャムセッションや合奏は、依然として物理的に 1 カ所に集まらなければ実施が困難な状況にある。遠隔地間でジャムセッションや合奏を行うことができれば、物理的に異なる場所にいる相手と、距離の制約を超えて音楽的なインタラクションを楽しむことができる。さらに、実世界では困難な、面識のない複数の演奏者と国境を超えたセッションも可能となる。本研究は、

[†] 独立行政法人産業技術総合研究所（旧電子技術総合研究所）
National Institute of Advanced Industrial Science and
Technology (former Electrotechnical Laboratory)

^{††} 早稲田大学理工学部
School of Science and Engineering, Waseda University
現在、IBM
Presently with IBM

このような遠隔セッションを可能にするシステムを実現することで、新たな形態の遠隔地間の合奏や、音楽における新たなインタラクションを探求することを目的とする。

従来、調性とリズムのある音楽を遠隔地間でセッションすることは、Internetのような遅延時間の長いネットワーク経由では困難と考えられていた。ISDNやInternet2、衛星中継等を用いて、MIDIあるいは演奏音を遠隔地間で中継した事例はあるが^{1)~3)}、遅延時間が小さいことを前提としており、長距離になると同時性が失われて実用上問題があることが指摘されていた¹⁾。そのため、不特定多数が利用するInternetのようなネットワークでは、実用的には企業等によって片方向のMIDI中継しか試みられていなかった。遠隔地にいる複数のユーザに、演奏をリアルタイムに自動生成するプロセスを制御させるアプローチ^{4),5)}や、楽音生成プロセスのパラメータ時変化を共同作曲(指定)させるアプローチ⁶⁾も提案されたが、制約が大きく、演奏者が通常の楽器(MIDIコントローラ)を用いてセッションに加わることはできなかった。音響信号のストリームに複数のユーザが各自の演奏を順次混合していくシステム⁷⁾もあるが、演奏中のユーザ間のインタラクションは得られなかった。このように、Internetを経由した遠隔地間で、調性とリズムのある音楽の合奏(双方向のMIDI中継)を可能にするシステムは、実現されていなかった。

このような遠隔セッションを可能にするために、我々は文献8)、9)において、RemoteGIGを提案した。RemoteGIGは、Internet等で不可避な遅延を積極的に利用した、新たな形態の遠隔地間のセッションである。RemoteGIGでは、同一のコード進行(12小節のブルース進行等)の繰返しを、テンポ一定で演奏することを前提とし、遠隔地にいる2人の演奏者が即興でネットワーク経由のセッションを行う。演奏者はお互いの演奏を、コード進行のちょうど1周期分の時間だけ遅れて聞き合うことでインタラクションを行う。コード進行は繰返すため、演奏が1周期分遅れれば再び同じコードとなり調和することが期待できる。

本論文では、このRemoteGIGを不特定多数のユーザが演奏できるように拡張した遠隔セッションシステムOpen RemoteGIGを提案する。RemoteGIGは知人同士でしか実施できず、運用場面が限定されていたが、Open RemoteGIGによって面識のないユーザ間のセッションを達成することで、より現実的な場面に適用可能とする。Open RemoteGIGでは、遠隔地にいる全ユーザの演奏情報が、Internet等を経由した

スター状のネットワーク接続によって、すべて中央に集められる。そして、自分以外の演奏情報が、コード進行の1周期の整数倍の時間だけ遅延して送り返されてくることで、全ユーザが互いの演奏を聞き合いながらセッションできる。Open RemoteGIGではさらに、演奏相手を発見するための環境や、演奏内容の打合せ等を可能にするチャット機能も提供する。なお、ユーザは演奏を行わずに、他人の演奏を観客として鑑賞してもよい。これを活用すれば、Internet上でのOpen RemoteGIGの公開ライブパフォーマンスも実施できる。

以下、2章において通信プロトコルRMCPを用いたRemoteGIGの実現方法を述べ、東京とニューヨーク間でInternetを介して実施した実験コンサートの結果を報告する。次に、3章において本研究が提案するOpen RemoteGIGの機能を紹介し、システムの構成と具体的な実装方法を説明する。本実装では、演奏情報の中継機能はC言語で実装されているが、ユーザはWWWブラウザを用いてOpen RemoteGIGのWWWページにアクセスすることで、遠隔セッションに参加することができる。最後に、4章においてRemoteGIGやOpen RemoteGIGによる遠隔セッションの意義を議論し、5章でまとめと今後の研究の方向性を述べる。

2. 遅延を考慮した遠隔セッション：RemoteGIG

RemoteGIGは、遠隔地間で遅延が大きい状況を考慮した、新たな演奏のモデルに基づくセッションである。演奏者が1カ所に集まってセッションを行うときの伝統的なモデルは、他者の演奏音を小さい遅延時間で聞くことができ、それに反応して演奏する自分の演奏音も小さい遅延時間で相手に届くことを前提としている。しかし、光速でさえ地球を半周するのに約66 msecもかかってしまうことから分かるように、遠隔地間で通信する際には、ネットワークの遅延時間を回避することは不可能である。したがって、遠隔地間のセッションのためには、伝統的なモデルに代わる演奏のモデルが必要となる。

遠隔地にいる2人の演奏者がセッションする際のRemoteGIGのモデルを、図1に示す。本モデルは、同一のコード進行の繰返しを、その1周期分の時間(D)を一定にして演奏することを前提とする。そし

将来的に、地球と月にいる演奏者がセッションをする場合には、光速でも地球から月へ届くのに約1.28 secかかる。

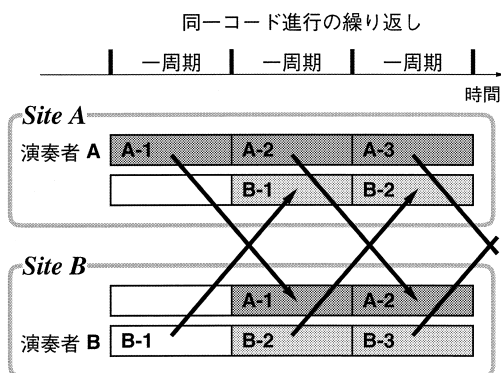


図 1 RemoteGIG のモデル

Fig. 1 RemoteGIG session model.

て、演奏者はお互いの演奏を、ちょうど周期 D 分の時間だけ遅れて聞き合い、それに反応して即興演奏する。たとえば、図 1 において、演奏者 A の A-1 の区間の演奏音は、演奏者 B のもとでは B-2 の区間で鳴る。コード進行は繰り返すため、A-1 のある演奏音が鳴っているときのコードは、その音が伝送されて B-2 で鳴るときのコードと一致して調和する。このように、RemoteGIG のモデルは遅延がコード進行 1 周期分の時間よりも短ければよく、遠隔地間のセッションに適している。

2.1 実現方法

RemoteGIG を実現するには、大別して、音響信号を直接中継する方法と、MIDI 信号のようなシンボル化した演奏情報を中継する方法の 2 種類が考えられる。音響信号の中継は、使用楽器の制約がない反面、MIDI 信号に比べて非常に伝送量が多く、受信側で演奏情報を再利用・加工しにくいという欠点がある。そこで本研究では、MIDI 信号を双方向に中継することで RemoteGIG を実現する。

一定の遅延時間で双方向の MIDI 中継を行うには、ネットワーク経由で MIDI 信号を伝送し、適切な時刻となるまでバッファリングして再生する必要がある。本研究では、このために、我々が設計した音楽用通信プロトコル RMCP (Remote Music Control Protocol^{8)~12)}を用いた。RMCP では、演奏情報は RMCP パケットという単位で伝送され、MIDI note on/off 等の各 MIDI メッセージは、1 つの RMCP パケットの中に含まれて送信される。RMCP は、

- タイムスタンプを用いたパケットの時間管理
- 信頼性を確保した遠隔地間の双方向パケット中継の 2 つの機能を提供できることから、今回の目的に適している。さらに RMCP は、運用するネットワークの性質・信頼性に依りて下位レイヤを使い分けることで、

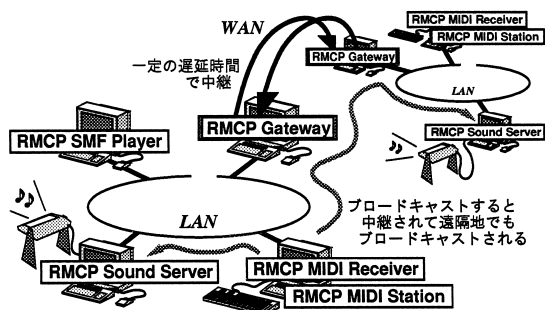


図 2 RMCP による遠隔地間のパケット中継

Fig. 2 RMCP-based packet relay between remote sites.

分散音楽情報処理に適した効率が良い実用的な情報共有を達成している。具体的には、Ethernet 等の LAN (Local Area Network) 内の通信には UDP/IP を用いて、ブロードキャストによるオーバーヘッドの小さい情報共有を実現する一方、Internet 等の WAN (Wide Area Network) 経由の通信には TCP/IP を用いることで、信頼性を確保した情報共有を可能にしている。

2.2 RMCP による RemoteGIG の実装

RMCP によって RemoteGIG を実現する際の、プロセス構成の例を図 2 に示す。RemoteGIG は、MIDI メッセージを MIDI 機器から受信するための RMCP MIDI Receiver、計算機のキーボードとマウスを用いて演奏するための RMCP MIDI Station、受信パケット中の MIDI メッセージを MIDI 機器へ送信するための RMCP Sound Server、WAN で結ばれた 2 つの LAN 間で双方向に RMCP パケットを中継するための RMCP Gateway、標準 MIDI ファイル (SMF: Standard MIDI File) を再生するための RMCP SMF Player によって実現される。

演奏者が MIDI 楽器等を弾いて生成した MIDI メッセージが、どのように伝送されて RemoteGIG が実施されるかを、LAN と WAN の場合を分けて説明する。

• LAN 上での MIDI メッセージの伝送

演奏者は、RMCP MIDI Receiver か RMCP MIDI Station を用いて演奏する。これらがブロードキャストしたパケットを RMCP Sound Server が受信すると、パケットに含まれる MIDI メッセージを MIDI 楽器へ送り、演奏音として出力する。

詳しい記述は文献 8), 9) に譲るが、これらのパケットを RMCP Display Server が同時に受信して視覚化したり、RMCP Packet Recorder が受信して記録したりできる。これは、RMCP がブロードキャストによって LAN 上で通信する利点の 1 つである。

● WAN を経由した MIDI メッセージの伝送

図2のように、双方向のパケット中継をする WAN の両端で RMCP Gateway を実行し、TCP/IP のコネクションを確立する。各 RMCP Gateway は、自分の LAN 上にブロードキャストされた RMCP パケットを、他方の RMCP Gateway へ送信する。逆に、他方からのパケットを受信すると、遅延時間が一定となるようにタイムスタンプ(パケットが処理されるべき時刻を表す)を書き換えて、自分の LAN 上にブロードキャストする。これにより、一方の演奏者が RMCP MIDI Receiver 等を用いて演奏した MIDI メッセージは、RMCP Gateway によって中継されて他方の LAN の RMCP Sound Server にも届き、演奏音として出力される。

以上のように、自分の演奏音が、自分と相手のもとにある MIDI 楽器によって出力されることで、遅延付きの遠隔地間のセッションが行える。なお、コード進行の周期を遅延時間と同一に保つためには、コードとリズムが分かるような伴奏(ベースとドラムス等)を SMF に記録しておき、RMCP SMF Player を用いていずれかの LAN 上で再生するとよい。この伴奏も RMCP Gateway によって中継され、両方で共有される。

2.3 実 験

RemoteGIG の有効性を実証するために、日米インターカレッジ・コンピュータ音楽フェスティバル(第23回音楽情報科学研究会)^{3),14)}の一環として、日本時間 1997 年 12 月 14 日(日)午前 11 時(ニューヨークでは 13 日(土)午後 9 時)より、コンサート形式での公開実験“Internet RemoteGIG”を実施した。東京の NTT ICC(InterCommunication Center)とニューヨークのコロンビア大学との間で、Internet を経由して RemoteGIG を行った。日本側の会場となった NTT ICC では、Rick Bassett が MIDI キーボードで演奏した。一方、USA 側の会場となったコロンビア大学では、2 曲演奏した中で、1 曲目を Brad Garton が MIDI キーボードで演奏し、2 曲目を Perry Cook が自作の MIDI コントローラーを用いて演奏した。

それぞれの会場では SGI O2 (IRIX-6.3) を 1 台占有使用し、RMCP Gateway、RMCP Sound Server、RMCP MIDI Receiver を動作させた。そして、同種

の MIDI 音源(Roland SC-55mkII)により演奏音を出力した。また、会場にいる聴衆が、日米のどちらの演奏者が弾いた音かを容易に確認できるように、画面上の鍵盤の色を変えて MIDI 情報を視覚化する RMCP Display Server を動作させた。さらに、ビデオカメラで撮影した動画を Internet を経由して中継し、演奏会場の風景をお互いに見ることを可能にした。

RMCP Gateway による中継用遅延時間は、両会場間のネットワークの遅延時間(約 0.25 秒)に比べて十分大きい値である 30.72 秒に設定した。そして、これがコード進行 1 周期となるようなテンポで、伴奏用のベースとドラムス等が含まれた SMF を再生した。具体的には、1 曲目(後藤真孝作曲)は 4/4 拍子 12 小節のブルース進行の曲であったため、テンポ 93.75 M.M.(4 分音符 = 640 msec)で演奏した。2 曲目(Rick Bassett 作曲)は 4/4 拍子 16 小節のコード進行を持っていたため、テンポ 125 M.M.(4 分音符 = 480 msec)で演奏した。

実験の結果、知覚できるテンポの揺らぎやパケットロスは発生せず、実際に遠隔地間で RemoteGIG が行えることを確認した。

2.4 考 察

RemoteGIG の実験を通じて得られた考察を以下にまとめる。

● 演奏の形態

RemoteGIG では、演奏開始時、ソロ演奏時、演奏終了時において、1 カ所に集まって行うセッションとは異なる演奏形態が求められた。その一例を図 3 に示す。まず演奏開始時には、B-1 の演奏が演奏者 A に届くまでに 2 コーラス(コード進行 2 周期分)かかってしまう。その A-1 と A-2 の期間、演奏者 A は 1 人で演奏をする必要があった。

次にソロ演奏に関しては、1 カ所のセッションだと数コーラス交代でソロを弾いても問題は起きないが、RemoteGIG では問題になる。たとえば演奏者 A が A-1 と A-2 のソロ演奏の後に A-3 で演奏者 B のバック



図 3 RemoteGIG の演奏形態の一例 Fig. 3 An example of RemoteGIG performance.

RMCP Gateway の実装方法は、文献 9) に述べられている。中継の無限ループを回避するために、パケットヘッダの中継禁止フラグをセットしてからブロードキャストする。調性とリズムのある音楽を Internet を経由して遠隔地間でセッションしたコンサートとしては、我々の知る限り世界で初めての試みであった。

グに移るとすると、本来は演奏者 B は B-1 でソロを弾いていなければならない、演奏者 B のもとでは A-1 のソロと B-1 のソロが重なってしまう問題が生じる。そこで今回は、2 小節交代あるいは 4 小節交代で、ソロとバックングを繰り返し演奏する形式を考案した。たとえば、16 小節のコード進行を 4 小節ずつに分割して、「A のソロを 4 小節 → B のソロを 4 小節 → A のソロを 4 小節 → B のソロを 4 小節」という形式で演奏した。さらに、演奏の進行にともない、交代する小節数を短くしていくような形式も効果的であった。演奏終了時には、伴奏の終了とともに演奏を終えるための工夫が必要であった。そこで、図 3 の S-6 と S-8 の伴奏の位置にフィルイン等の合図を入れておき、演奏者 B は S-6 を聞くと演奏を終了し、演奏者 A は S-8 を聞くと演奏を終了することとした。

以上をふまえた RemoteGIG の拡張案としては、SMF を事前に送っておいて、両者の演奏開始時刻を合わせる機能を実装することがあげられる。これにより、図 3 の演奏者 B の演奏開始を 1 コーラス早めることができる(演奏者 A のもとでは A-2 と B-1 が重なる)。そして、両者が奇数番目のコーラスにソロを演奏し、偶数番目のコーラスにバックングを演奏することで、1 コーラスごとに交代する演奏形式も可能となる。また、RMCP Gateway に MIDI 中継を途中で中止する機能を実装すれば、MIDI 中継を S-8 の直後にやめることができ、両者が伴奏の最後まで演奏できるようになる。

- 音楽以外のコミュニケーション手段の必要性
演奏前後の準備等でお互いの状況を的確に伝達するためには、演奏音以外のコミュニケーション手段も不可欠であった。そこで今回は、UNIX の talk コマンドによりテキストベースでリアルタイムチャットを行った。このような手段の確保は、コンサート形式で実施する場合必ず考慮する必要がある。

- ネットワークトラフィックの問題
コンサート中は問題が起きなかったが、Internet の混雑状況によっては、演奏中の遅延時間がコード進行 1 周期よりも突然長くなってしまったことがあった。コンサート前に早稲田大学とコロンビア大学間でリハーサルを繰り返した際には、日本時間の夕方等に、ほとんど TCP/IP パケットが通らない状況も発生した。

3. 不特定多数のユーザが参加できる遠隔セッション : Open RemoteGIG

Open RemoteGIG は、多数の人が自由に RemoteGIG に参加することを可能にする遠隔セッションシステムである。これは、基本的には 2 章で述べた Re-

moteGIG に基づいているが、RemoteGIG をそのまま不特定多数のユーザに対して運用しようとする、以下のような問題が生じてしまう。

- (1) 演奏者を増やすためには、すべての演奏者の LAN 間で、RMCP Gateway による全対全結合のコネクションを確立しなければならなかった。
- (2) 遅延時間等はコネクションを確立した後に変更できなかったため、コネクションを張ったままで様々な曲を演奏するには、コード進行の周期を合わせなければならないという制約があった。
- (3) 事前に演奏相手を定め、両者の LAN 間でコネクションを確立しておく必要があった。そのため、面識のない相手と演奏することはできなかった。
- (4) 伴奏に用いる SMF の情報は、事前に内容が確定しているにもかかわらず、演奏中に伝送していた。これは汎用性を持たせるためには有効な設計であったが、Open RemoteGIG のように多数の演奏者が参加する場合には、伝送容量を圧迫してしまう点が問題となる。
- (5) 2.4 節でも指摘したように、音楽以外のコミュニケーション手段は提供していなかった。しかも、演奏者が 3 人以上の場合は、多対多のコミュニケーション手段が必要となる。

そこで本研究では、これらの問題を以下のように解決することで、Open RemoteGIG を実現する。

- (1) スター状のネットワーク接続
Open RemoteGIG では、演奏中継用のセンタを用意する。そして、各ユーザは自分の計算機から、Internet 等の WAN 経路でセンタに接続(コネクションを確立)するだけで、他の複数のユーザの演奏音を同時に聞きながら即興演奏ができる。4 人の演奏者がセッションする際の Open RemoteGIG のモデルを図 4 に示す。このようなシステムでは時間軸方向の制御が重要であるが、Open RemoteGIG では全ユーザ間

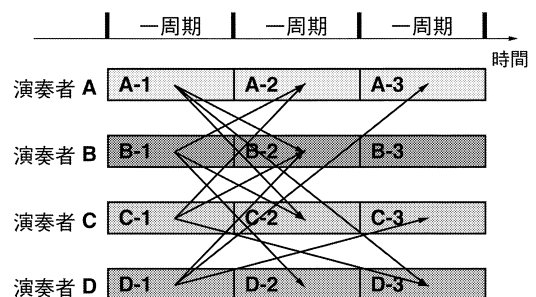


図 4 Open RemoteGIG のモデル
Fig. 4 Open RemoteGIG session model.

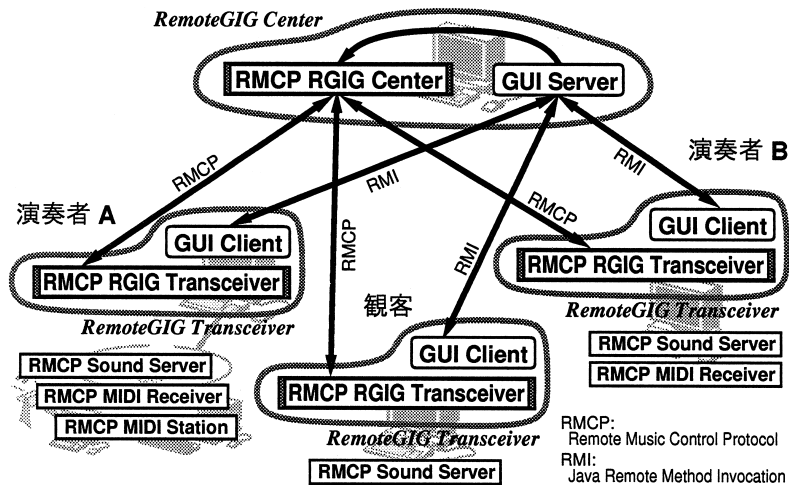


図 5 Open RemoteGIG のシステム構成

Fig. 5 System configuration of Open RemoteGIG.

で共通の仮想時刻（たとえばセンタの時刻）を設定し、仮想時刻上で図 4 のような演奏が行えるようにする。

- (2) 中継用遅延時間を動的に変更できる演奏中継 RemoteGIG と同様に同一のコード進行を繰り返し演奏することを前提とし、コード進行 1 周期の整数倍の遅延時間で演奏情報（MIDI メッセージ）を中継する。その際、コネクション確立後の任意の時点で遅延時間を変更できるようにすることで、様々なコード進行の異なる周期に対応可能とする。また中継用遅延時間は、ネットワークの遅延時間よりも十分長くなければならないという前提を考慮しながら、コード進行の何周期分とすべきかを決定する。
- (3) 演奏相手の発見環境
センタに接続している他のユーザを見つけ、一緒に演奏を開始できる環境を用意することで、面識のない相手との遠隔セッションを可能にする。ただし、各ユーザの演奏が異なる音色（MIDI プログラム番号）で正しく再生されるために、ユーザに対して MIDI チャンネルを排他的に割り当てるよう管理する（現在の実装では、1~9 チャンネルを割り当てている）。そのため、同時に演奏できる人数は割当て MIDI チャンネル数に制限される。なお、全ユーザのもとでプログラム番号と音色の対応の一貫性を保つために、MIDI 音源として GM（General MIDI）音源を用いることを前提とする。

(4) SMF の事前伝送

コード進行 1 周期分が記録された SMF を演奏前にダウンロードしておき、各ユーザのもとで繰り返しループさせて再生する。その際、ユーザ同士の即興演奏が時間軸上で調和するために、SMF の再生開始時刻が全ユーザのもとで同一の仮想時刻となるように調整する。なお、SMF が使用する MIDI チャンネルは、上記の割当て用 MIDI チャンネル以外を使用しなければならない（現在の実装では、伴奏が 10~16 チャンネルを使用する）。

- (5) テキストベースのリアルタイムグループチャット演奏音以外のコミュニケーション手段として、任意のユーザ間の多対多のチャットを実現し、演奏内容の打合せ（たとえば、ソロを何小節ごとに交代するか）等を可能にする。

本研究ではさらに、Open RemoteGIG の機能として以下の 2 つを提案する。

- 仮想演奏スタジオの切替え（入退室）機能
複数の仮想演奏スタジオ（GIG チャンネルと呼ぶ）を用意し、異なるユーザグループが独立に RemoteGIG セッションを行うことを可能にする。GIG チャンネルごとに異なる伴奏の SMF を用意しておけば、ユーザは GIG チャンネルを切り替えることで、好みの伴奏に合わせて即興演奏ができる。前述の MIDI チャンネルの割当ては、この GIG チ

GIG チャンネル 1 はフュージョン風、GIG チャンネル 2 はジャズ風、GIG チャンネル 3 はロック風、といったように、様々なチャンネルの伴奏をセンタ側で用意しておく効果的である。

ネルごとに独立に行う。

- 観客の立場での演奏の鑑賞機能

演奏に参加せずに、他のユーザの演奏音だけを聞く機能(鑑賞モード)を提供する。これにより、即興演奏のできないユーザでも観客の立場で Open RemoteGIG を利用できる。あるいは、すべての MIDI チャンネルが割当て済みのときに、MIDI チャンネルが空くまでは観客として待機してもよい。この機能を活用すれば、大勢の観客に対し、Open RemoteGIG の公開ライブパフォーマンスを Internet 上で実施できる。

3.1 システム構成

Open RemoteGIG のシステムは、演奏中継用のセンタである RemoteGIG Center と、各ユーザがセンタに接続するための RemoteGIG Transceiver で構成される。システムの構成図を図 5 に示す。Open RemoteGIG の機能は、演奏中継機能と GUI・コミュニケーション機能に二分でき、前者は RemoteGIG と同様に RMCP を用いて実現し、後者は WWW ブラウザ上で動作するように Java Applet として実現する。実際の MIDI 楽器との入出力には、2.2 節で述べたように RMCP Sound Server, RMCP MIDI Receiver 等を用いる。

3.1.1 RemoteGIG Transceiver

RemoteGIG Transceiver は、ユーザの演奏情報をネットワークを経由して送受信するための RMCP RGIG Transceiver と、ユーザからの指示を伝達したり他のユーザの属性を表示したりする GUI Client から成る。ただし、図 5 に示したように、両者は直接通信せず、つねに RemoteGIG Center を経由する。

- RMCP RGIG Transceiver

演奏中継に関しては RMCP Gateway と同等の機能を持ち、後述する RMCP RGIG Center と TCP/IP を用いて RMCP パケットを送受信する。その際、RMCP RGIG Center へ送信する前に、パケットのタイムスタンプを実時刻(ローカル時刻)から仮想時刻に変換する。また、RMCP RGIG Center から受信したパケットのタイムスタンプを仮想時刻から実時刻に変換し、中継用遅延時間を加えた後に LAN 上にブロードキャストする。

さらに、RMCP RGIG Center から以下のような指

示を受信し、それに応じた処理を行う。

- 「伴奏用 SMF のダウンロード」

RMCP RGIG Center との接続時に、全 GIG チャンネル分の SMF (テンポ、コード進行 1 周期の長さ、開始仮想時刻も含む)を受信する。

- 「GIG チャンネルの指定」

指定チャンネルの伴奏用 SMF に切り替え、そのコード進行の周期とネットワークの遅延時間に基づいて、新たな中継用の遅延時間を決定する。

- 「SMF の演奏開始/演奏停止」

演奏開始を受信すると、開始仮想時刻とコード進行の周期から次回の伴奏開始実時刻を計算し、演奏停止を受信するまで無限に繰り返しながら SMF を再生する。

- 「鑑賞モード/演奏モード(割当て MIDI チャンネルの指定付き)の変更」

RMCP RGIG Center への MIDI メッセージの送信に関して、鑑賞モードのときはいっさい送信せず、演奏モードのときは指定 MIDI チャンネルのみ送信する。

- 「MIDI のオールノートオフ/プログラム番号変更の送信命令」

当該 MIDI メッセージをブロードキャストする。

- GUI Client

ユーザ用の GUI は、演奏管理ウィンドウ(図 6)とコミュニケーションウィンドウ(図 7)で構成される。ユーザは、GUI Client を起動して GUI Server に接続した後に、コミュニケーションウィンドウで演奏相手を見つけて希望の GIG チャンネルに切り替え、演奏管理ウィンドウで演奏状態を変更して演奏を開始する。

ユーザは参加時に、コミュニケーションウィンドウ(図 7)の左下で、名前を入力して login する。すると、右の“All Users”のサブウィンドウに、全参加ユーザの一覧が表示され、任意のユーザ(複数も可)を指定してチャット可能となる。グループチャットを簡便に行うために、新たなサブウィンドウ(“Chat Group:番号”)を作成し、選択したユーザを登録しておく機能もある。チャット相手を切り替えるには、このサブウィンドウをクリックするだけでよい。そして、“GIG Channel”のメニューから GIG チャンネル番号を選ぶと、“GIG Channel:番号”のサブウィンドウが現れ、そこで演奏/鑑賞しているユーザの一覧が表示される。その後、演奏管理ウィンドウの操作に移れるようになる。

演奏管理ウィンドウ(図 6)では、SMF の演奏開始

異なる RMCP サーバ番号やソケットポート番号を用いれば、同一 LAN 上で複数の RMCP RGIG Transceiver を独立に運用できる。

これは、文献 9) の RMCP Gateway による内部時計の時刻差の補正と同等の処理である。



図6 GUI Clientの演奏管理ウィンドウ
Fig. 6 Manager window of GUI Client.

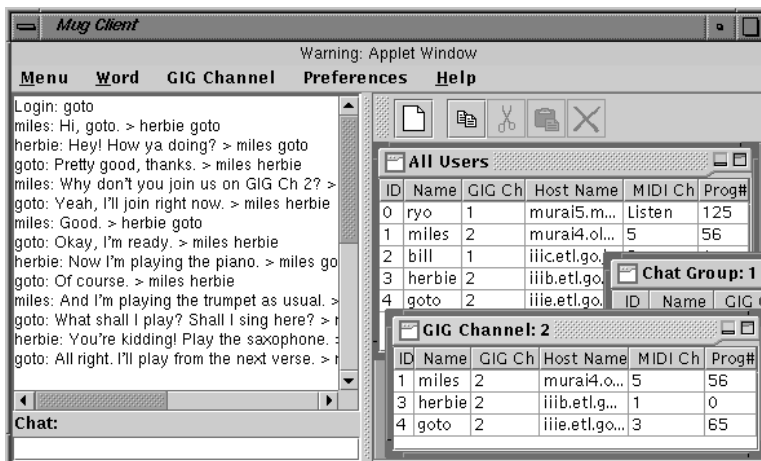


図7 GUI Clientのコミュニケーションウィンドウ
Fig. 7 Communication window of GUI Client.

(“Start”)・停止 (“Stop”), 鑑賞モード (“Listen”) と演奏モード (“Play”) の切替え, 割当て希望 MIDI チャンネルの指定を指示する. 他にも, MIDI プログラム番号の変更 (音色切替え), MIDI オールノートオフの送信を指示できる.

3.1.2 RemoteGIG Center

RemoteGIG Center は, 演奏情報を全ユーザのもとへ中継する RMCP RGIG Center と, 全ユーザの属性を管理・中継する GUI Server から成る. ユーザの属性の変更や RMCP RGIG Transceiver への動作指示は, 図 5 のように GUI Server から RMCP RGIG Center へ一方方向に通知される. ユーザの属性には, ユーザ ID, GIG チャンネル, 鑑賞モード/演奏モード, MIDI チャンネル, MIDI プログラム番号があり, 動作指示に

は, 演奏開始/停止命令, MIDI オールノートオフ送信命令がある.

- RMCP RGIG Center

全 RMCP RGIG Transceiver からの接続を受け付け, GIG チャンネルごとに分けて管理する. RMCP RGIG Transceiver の接続時に伴奏用 SMF を送信した後は, 受信したユーザの演奏 (RMCP パケット) を, それ以外の全ユーザへ送信する処理を行う. また, GUI Server からユーザ属性の変更や動作指示を通知されると, 対応する処理をしたり, 当該 RMCP RGIG Transceiver へ送信したりする.

- GUI Server

全 GUI Client のユーザの属性管理と変更の通知, チャットの中継を行う. GUI Client の操作による属

性変更や動作指示は GUI Server へ集められ、RMCP RGIG Center へ通知される。同時に、画面表示を更新するために全 GUI Client にも通知される。RMCP RGIG Transceiver が RMCP RGIG Center に接続されていることを確認する処理 (IP アドレスに基づいて同定する) や、MIDI チャンネルの割当ての排他処理も行う。

3.2 実装

RMCP RGIG Center と RMCP RGIG Transceiver は C 言語で実装し、両者間の通信には RemoteGIG と同様に TCP/IP 上の RMCP を用いた。RMCP RGIG Transceiver は UDP/IP 上の RMCP により、RMCP Sound Server、RMCP MIDI Receiver 等と通信する。

一方、GUI Server と GUI Client は Java 言語で実装した。GUI Client は Java Applet として実装され、WWW ブラウザ (Netscape Communicator 等) や appletviewer 上で動作する。HTTP (Hyper Text Transfer Protocol) によってダウンロードした GUI Client は、RMI (Java Remote Method Invocation) を利用して GUI Server との通信を行う。GUI の実装には、JFC (Java Foundation Classes) を利用した。また、GUI Server から RMCP RGIG Center へのユーザ属性変更・動作指示の通知には、JNI (Java Native Interface) を用いた。

3.3 実験結果

Open RemoteGIG によって実際に遠隔地間でセッションが実施できることを確認するために、電子技術総合研究所 (電総研) と早稲田大学 (早大) 間で Internet を経由した遠隔セッションを試みた。Open RemoteGIG を運用する OS には、SGI IRIX および SUN Solaris を用いた。RemoteGIG Center を電総研あるいは早大に設置し、電総研と早大において複数の RemoteGIG Transceiver を動作させて実験した。伴奏用 SMF には、2.3 節の実験の際の SMF 等を使用し、1 周期の時間は 30 秒前後で SMF ごとに異なる値を設定した。

実験の結果、遠隔地間で全ユーザの演奏が適切に中継され、チャット等の GUI に関連した機能も設計どおり動作することを確認した。特に、GUI やチャット等を利用して、他のユーザの状況を把握しながら遠隔セッションをできる点が有効であった。

4. 遠隔地間のセッションに関する議論

RemoteGIG や Open RemoteGIG で実現した遠隔地間のセッションは、様々な意義を持ち、新たな展開も期待できる。以下、主要な 2 つの観点から議論する。

4.1 距離の制約を超えた新たな形態のセッション
実現した遠隔地間のセッションは、まず第 1 に、異なる場所にいる演奏者同士が調性とリズムのある音楽を即興演奏することを可能にした点で意義がある。従来は、そのようなセッションは実施困難だと考えられていた。

第 2 に、このセッションのモデルは、従来の演奏者が 1 か所に集まるセッションのモデルとは大きく異なるため、音楽演奏に対して新たな観点を与える点で意義がある。従来の音楽では、演奏者は全員ほぼ同時に同じ演奏を聞くものという暗黙の前提があったが、この遠隔セッションのモデルでは、全員が同じ演奏を聞くことがなく相手の反応もすぐには分からないという状況下での、従来なかった音楽的なインタラクションの方法論が求められる。

たとえば、RemoteGIG で当初達成されるのは、各自の音が調和するレベルでの演奏だと考えられる。お互いに、相手の過去の演奏に影響は受けているものの、インタラクションの範囲は限られている。その先にある方向性の 1 つとして考えられるのは、自分と相手の過去の演奏の記憶も考慮した高度なインタラクションである。コード進行の 1 周期分だけ遅れて聞き合うとき、自分が聞いている相手の演奏は、自分の 2 周期前の演奏への反応である。そこで、どういった反応が得られたかを記憶に基づいて判断して次の演奏内容を決定できれば、より緊密なインタラクションを行える。将棋や囲碁の棋士が対局中の過去の指し手を記憶しているように、演奏者がセッション中の過去の演奏を記憶し、それに基づいて即興演奏できれば、このような新たなインタラクションも達成できる可能性がある。

4.2 不特定多数のユーザによる音楽演奏

Open RemoteGIG では、不特定多数のユーザによる音楽的なインタラクションを、遅延時間を管理することによって実現した点で意義がある。すでに、不特定多数のユーザが、テキストベースのチャット・掲示板やゲーム等に参加することを可能にするシステムは広く普及している。しかし、そのほとんどは、リアルタイム性があまり要求されない (同時性が要求されず、遅延時間が小さいけれど不定な) 情報共有か、掲示板のような完全に非リアルタイムな (遅延時間が非常に大きく不定な) 情報共有に基づいている。それらを直接音楽に適用しようとしても、前者の情報共有では各ユーザの演奏音がずれて問題があり、後者の情報共有では調性とリズムのある音楽を即興演奏できない。Open RemoteGIG では、遅延時間を意図的に長く一定にし、伝送内容に応じて管理することで、そのよう

な音楽的なインタラクションを可能にした。

5. おわりに

本論文では、1997年12月のRemoteGIGの公開実験の結果と考察を述べ、それをふまえたうえで、面識のない多数の人がInternet経由でRemoteGIGの遠隔セッションに参加できるシステムOpen RemoteGIGを提案した。Open RemoteGIGでは、複数のユーザ間の演奏中継を実現しただけでなく、仮想演奏スタジオ(GIGチャンネル)の切替えや、鑑賞モードにより公開ライブパフォーマンスを行うことも可能にした。また、演奏相手の発見環境やグループチャット機能により、演奏音以外のコミュニケーション手段も提供した。RemoteGIGおよびOpen RemoteGIGは、将来普及するであろう遠隔セッションの1つの新しい方向性を示すものであるといえる。

謝辞 村岡洋一教授には、本研究を様々な面からご支援いただいた。また菊地淑晃氏からは、本研究に対し有益なご示唆をいただいた。1997年12月のRemoteGIGの公開実験においては、Rick Bassett氏、Brad Garton氏、Perry Cook氏に演奏者として協力していただき、菜孝之氏にはコンサートの機会を与えていただいた。また、松田周氏をはじめとする国立音楽大学の方々にはNTT ICCでの準備に協力していただき、興梠正克氏には動画像の伝送で協力していただいた。本研究に協力していただいたすべての方々に、心から感謝の意を表する。

参考文献

- 1) 高山 明, 千葉雅之, 三木恵造, 首藤一彦: 遠隔同時演奏システム, 日本音響学会音楽音響研究会 MA89-10, Vol.8, No.3, pp.23-28 (1989).
- 2) Nielsen, O.: MIDI and Audio via ISDN, *Proc. 1994 International Computer Music Conference*, pp.451-454 (1994).
- 3) Helmuth, M.: Sound Exchange and Performance on Internet2, *Proc. 2000 International Computer Music Conference*, pp.121-124 (2000).
- 4) Burk, P.L.: Jammin' on the Web—A new Client/Server Architecture for Multi-User Musical Performance, *Proc. 2000 International Computer Music Conference*, pp.117-120 (2000).
- 5) Pazel, D.P., Abrams, S., Fuhrer, R.M., Oppenheim, D.V. and Wright, J.: A Distributed Interactive Music Application Using Harmonic Constraint, *Proc. 2000 International Computer Music Conference*, pp.113-

116 (2000).

- 6) Jordà, S.: Faust Music On Line: An Approach to Real-Time Collective Composition on the Internet, *Leonardo Music Journal*, Vol.9, pp.5-12 (1999).
- 7) Giuli, D., Pirri, F. and Bussotti, P.: Orchestral: A Distributed Platform for Virtual Musical Groups and Music Distance Learning over the Internet in Java Technology, *Proc. IEEE International Conference on Multimedia Computing and Systems*, Vol.2, pp.987-988 (1999).
- 8) Goto, M., Neyama, R. and Muraoka, Y.: RMCP: Remote Music Control Protocol—Design and Applications, *Proc. 1997 International Computer Music Conference*, pp.446-449 (1997).
- 9) 後藤真孝, 根山 亮, 村岡洋一: RMCP: 遠隔音楽制御用プロトコルを中心とした音楽情報処理, 情報処理学会論文誌, Vol.40, No.3, pp.1335-1345 (1999).
- 10) 後藤真孝: MIDI 制御のための分散協調システム, jus 設立 10 周年記念 UNIX 国際シンポジウム論文集, pp.161-171 (1992).
- 11) 後藤真孝: 分散協調インタラクティブシステム—音と CG による遠隔地間のインタラクション, NICOGRAPH'93 CG 教育シンポジウムプロシーディングス, pp.44-49 (1993).
- 12) 後藤真孝, 橋本裕司: MIDI 制御のための分散協調システム—遠隔地間の合奏を目指して, 情報処理学会研究報告音楽情報科学 93-MUS-4-1, Vol.93, No.109, pp.1-8 (1993).
- 13) Rai, T. and Bassett, R.: Program book of U.S.A./Japan InterCollege Computer Music Festival: Computer Music Concerts and Lectures (1998).
- 14) Hashimoto, S., Moon, B., DuBois, R.L. and Castle, H.: Regional news, *ARRAY: Communications of the ICMA*, Vol.18, No.1, pp.2-4 (1998).

(平成 13 年 6 月 6 日受付)

(平成 13 年 12 月 18 日採録)



後藤 真孝 (正会員)

1993年早稲田大学理工学部電子通信学科卒業。1998年同大学大学院博士後期課程修了。同年、電子技術総合研究所(2001年に独立行政法人産業技術総合研究所に改組)に入所し、現在に至る。2000年より科学技術振興事業団さきがけ研究21研究員を兼任。博士(工学)。音楽情報処理、音声言語情報処理等に興味を持つ。1992年jst設立10周年記念UNIX国際シンポジウム論文賞、1993年NICOGRAPH'93CG教育シンポジウム最優秀賞、1997年情報処理学会山下記念研究賞、1999年平成10年電気関係学会関西支部連合大会奨励賞、2000年WISS2000論文賞・発表賞、2001年日本音響学会第18回粟屋潔学術奨励賞・第5回ポスター賞各受賞。電子情報通信学会、日本音響学会、日本ソフトウェア科学会、日本音楽知覚認知学会、IEEE、ICMA、ISCA各会員。



根山 亮 (正会員)

1997年早稲田大学理工学部情報学科卒業。1999年同大学大学院修士課程修了。同年、日本アイ・ビー・エム(株)東京基礎研究所に入所し、現在に至る。音楽情報処理、分散コンピューティング、Webサービス、SOAP等に興味を持つ。1999年情報処理学会大会奨励賞受賞。著書『Building Web Services with Java — Making Sense of XML, SOAP, WSDL and UDDI』(SAMS Publishing)。日本ソフトウェア科学会会員。