2003－MUS－53　(10)
2003／12／21

テンポ変化を考慮した楽音間で同期の取れない音場における演奏者間プロトコル

大部　由香[*1]　　米倉 達広[*1]

あらまし　本研究は、ネットワークを介したアンサンブル演奏を目的としている。ネットワークを介したアンサンブル演奏には高いリアルタイム性が求められるが、ネットワーク遅延により演奏に遅延が発生する。この場合、相手演奏者との間の同期が乱れ、自分のリズムが崩されるため、ネットワーク遅延は大きな問題となる。このような問題を解決するため、本研究では Mutual Anticipated Session (M.A.S.)という演奏形態を提案している。これは、相手の演奏の発音時刻を任意の長さだけ遅らせ、アンサンブル演奏を行う方法である。これまで、M.A.S.システムでは演奏のテンポは一定であることを前提として発音時刻の調節を行ってきた。今回は、テンポ変化に対応する機能の追加を行ったので報告する。

A Protocol for a Musical Session Considering the Change of the Tempo in an Asynchronous Sound Field

Yuka Obu[*1]　　Tatsuhiro Yonekura[*1]

Abstruct　Recently, two-way application systems on the Internet have become increasingly popular. When a musical session is performed via the network, it is necessary to interact in real time; however, there is the problem of delay, and the time lag between the musical notes may become an impediment. For this problem, we propose a new protocol for a musical session called Mutual Anticipated Session (M.A.S.), which is a type of ensemble that controls the timing of the sounds.

1.　はじめに

近年、Internet を利用したチャットやオンラインゲームのような双方向型のアプリケーションシステムが増加している。これらは、高いリアルタイム性は求められないため、Internet のような遅延が大きい環境であっても対話性の点での問題は少ない。しかし、アンサンブル演奏のように高いリアルタイム性が求められる用途においては、ネットワーク遅延により演奏に遅延が発生すると、相手演奏者との間の同期が乱れ、自分のリズムが崩されるため、ネットワーク遅延は大きな問題となる。

このような問題を解決するために、Open Remote GIG[1]、GDSM (Global Delayed Session Music)[2]といった手法が提案されてきた。Open Remote GIG は RMCP[3][4][5]というプロトコルを用い、数小節を単位として相手の演奏を 1 単位分遅らせたものと自分の演奏でアンサンブル演奏を行う。GDSM は 2 小節を単位として、相手の 2 小節前の演奏とアンサンブル演奏を行う手法である。どちらの手法も即興演奏を楽しむことは可能であるが、譜面が用意された楽曲のアンサンブル演奏には使用されていない。また、どちらの手法においても、演奏テンポは一定であることを前提としているため、演奏者がより自由な演奏を行うことが困難である。

また、Ethernet 上に存在する端末で生成された音の同期に関する研究もなされている[6]。

一方、ネットワーク遅延問題に関する他の解決策として、ネットワークの高速化、データの圧縮などの研究が行われている[7]。しかし、これらはアンサンブル演奏を行う際に生じる問題の本質的な解決策とはならない。

そこで本研究では、Mutual Anticipated Session (M.A.S.)という演奏形態を提案してきた[8][9]。これは、演奏の発音時刻を調節することによって、ネットワーク遅延やその変動による影響を最小限に抑制し、発音時刻の同期を取りアンサンブル演奏を行う方法である。これまで、M.A.S.はテンポ一定で演奏がなされることを前提としてきた。しかしながら、演奏者によるテンポ変化には対応していなかった。今回、曲のテンポ変化へ対応する機能を付加したので紹介する。

2.　同期の取れない音場

2.1　楽音間に生じる時間的ずれによる問題

地理的に離れた地点間でアンサンブル演奏を行うと、演奏された音は遅延の影響を受ける。発音タイミングのずれが 50ms 程度である場合、演奏者は遅延に

*1: 茨城大学大学院　理工学研究科
*1: Graduate School of Science and Engineering, Ibaraki University

対応して演奏を行うことは可能である。しかし、ずれが 300ms 以上である場合、これはテンポ 120bpm の曲においておよそ 8 分音符 1 つ分の長さに相当するため、リズムの同期に乱れが生じる。さらに、遅延を伴った演奏に合わせてアンサンブル演奏を行うと、その演奏が相手側でさらに遅延を伴って発音されるため、遅延が増幅し、演奏を続けることが困難となる。

2.2 同期の取れない音場

本研究では、上記のように楽音間にずれが生じる音場を“同期の取れない音場”と呼ぶ。同期の取れない音場においてアンサンブル演奏を行う場合、演奏者のリズムが乱されないようにするため、発音タイミングを調節する必要がある。

3. Mutual Anticipated Session

相手の演奏が遅れて聞こえると演奏が難しくなる。しかし、ネットワークを介してアンサンブル演奏を行う場合、遅延をすべて回避することは不可能である。そこで、本研究では、同期の取れない音場における新しい演奏形態である Mutual Anticipated Session (M.A.S.) を提案する。M.A.S.は以下に述べるように、曲をカノンの形にし、先行演奏でアンサンブル演奏を行う演奏形態である。

3.1 先行演奏と演奏遅延量

相手の演奏を演奏情報のネットワーク遅延分より早く発音することは不可能であるが、遅延分よりも遅らせて発音させることは可能である。したがって、Fig.1 のように相手の演奏を遅延分よりも更に任意の長さだけ遅らせ、人間にとってより演奏しやすいと思われる遅延量に調節して演奏することで、遠隔地間のアンサンブル演奏を実現する。この演奏方法では、相手の演奏に対して自分が先行して演奏を行うことから、この方法を“先行演奏”(Precedent Musical Performance) と呼ぶ。また、先行する演奏と後に続く演奏との時間間隔を“演奏遅延量”(Precedent Time)と呼ぶ。

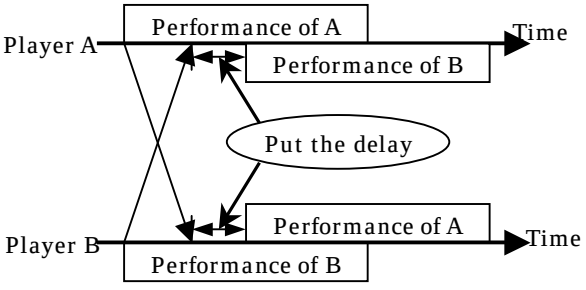

Fig. 1 Precedent Musical Performance

先行演奏を行う際、問題となるのが適切な演奏遅延量である。予備実験でのアンケート結果から、4 分の 4 拍子の曲の場合、演奏遅延量が 2 拍または 4 拍の時にアンサンブル演奏が行い易いことがわかった[8]。

4. M.A.S.の実現

以下のように、M.A.S.を行うシステムを実現した。

4.1 サーバ・クライアント

本システムは、LAN や WAN を介して演奏情報の送受信を行う。本システムは、サーバ・クライアント型を採用する。複数の演奏クライアントがサーバに接続することで、同じサーバに接続しているすべての演奏クライアントとのアンサンブル演奏を行うことが可能である。また、視聴クライアントや記録クライアントを接続することで、演奏者以外がアンサンブル演奏をほぼリアルタイムに視聴したり、記録したりできる（Fig.2）。

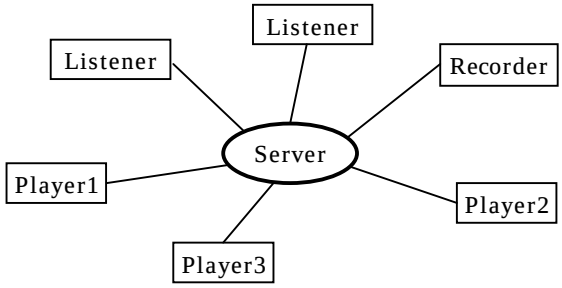

Fig. 2 Server and Client

サーバは、演奏開始のタイミングと演奏中の音の発音タイミングの調節を行う。各クライアントは、サーバを介して情報のやり取りを行う。

視聴・記録クライアントで聞こえる演奏は、全員の演奏が演奏遅延量分だけ遅れた演奏となる。つまり、演奏者は先行演奏を行っているが、視聴・記録クライアント側では、先行演奏の形にはならず、全員の演奏が遅延のない状態で演奏したかのように聞こえる。

通信には、TCP と UDP を採用する。TCP は高い信頼性が得られるが、処理時間が長くリアルタイムに演奏情報を送受信するには適さない。一方 UDP は、パケットロスによる情報損失等の危険性があるが、処理時間は早い。そこで、本システムでは、演奏開始やテンポ情報等の重要なデータは TCP を用いて確実に送信し、遅延の影響を減らす必要のある演奏情報には UDP を用いる（Fig.3）。演奏情報の送受信に UDP を用いることで、パケットロスによる演奏情報の欠落が発生すると考えられる。しかし、演奏をミスして音が抜けても演奏を続けることは可能であることから、パケットロスによる演奏情報の欠落で音が鳴らなくても、演奏を行ううえでは問題はないと思われる。ただし、現在

の本システムでは、演奏情報に MIDI を用いている。MIDI では、1 つの音を NOTE ON と NOTE OFF という 2 つのメッセージを用いて表しているため、NOTE OFF メッセージが欠落すると音が鳴り止まなくなる。そこで、NOTE OFF メッセージが欠落しても音が鳴り止むようにする必要がある。

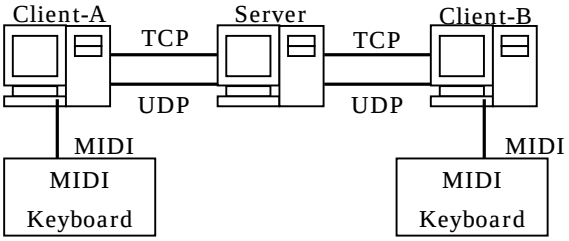

Fig. 3 Construction of the system

NOTE OFF メッセージの欠落に対処するため、本システムでは、EXTEND パケットを導入する。演奏クライアントは NOTE ON を送信後、1 拍ごとに EXTEND パケットを送信する。1 拍後に EXTEND が届かなかった場合、受信側では NOTE OFF を受信していなくても音を止める。この処理を行うことで、音が鳴り止まなくなることを防ぐことができる。

4.2 時間管理

先行演奏を実現するためには、正確に演奏を遅らせる必要がある。そのため、時間の管理が重要となる。

ネットワーク遅延には変動がある。その変動が演奏遅延量に影響を及ぼすことを防ぐために、本システムでは前もってクライアントに演奏情報を送信しておき、クライアント側で発音時刻まで待って音を鳴らす。

本システムでは、発音時刻の管理のために、各クライアントにタイマーを持たせている。入力された演奏情報の時刻や、受信した演奏情報の発音時刻は、このタイマーの値を用いている。

全体の流れは、以下のようになっている。ここでは、演奏クライアントが二つの場合で説明する。(Fig.4)

最初にサーバから演奏開始のパケットが各クライアントに向けて送信される。クライアントは演奏開始パケットを受け取ると、タイマーをスタートさせ、演奏が開始される。Client-A 側で演奏された音の場合、Client-A 側では即発音され、同時に Client-A のタイマー値（Client-A での発音時刻）を付加した演奏データがサーバへ送信される。サーバは、受信した演奏情報のタイマー値に演奏遅延量分を加算して、Client-B に対して送信する。Client-B はサーバから演奏情報を受け取ると、発音する時刻になるまで待ってから発音する。以上のような処理を行うことで、Client-A、Client-B それぞれにおいて、正確に 2 拍または 1 小節分遅らせることが可能になる。

ネットワーク遅延等により、タイマーが開始される時刻が同じであるとは限らない。しかし、演奏遅延量を D、Client-A からサーバまでの通信時間を t1、サーバから client-B までの通信時間を t2、タイマーの開始時刻のずれを g とすると、D>t1+t2+g であれば、問題はない。

Fig.4 に具体的な例を示す。Fig.4 は、Client-A での演奏時刻が 500ms、演奏遅延量が 1000ms の場合である。

5. テンポ変化への対応

演奏者が自由に演奏を行った場合、テンポに変化が生じる。しかしながら、これまでの M.A.S.システムでは、演奏遅延量が常に一定であるため、演奏テンポが変わるとリズム同期に乱れが生じる。したがって、M.A.S.を用いてアンサンブル演奏を行う場合、演奏者がより自由な演奏を行えるようにするには、テンポの揺れに応じた演奏遅延量を求めることが重要となる。

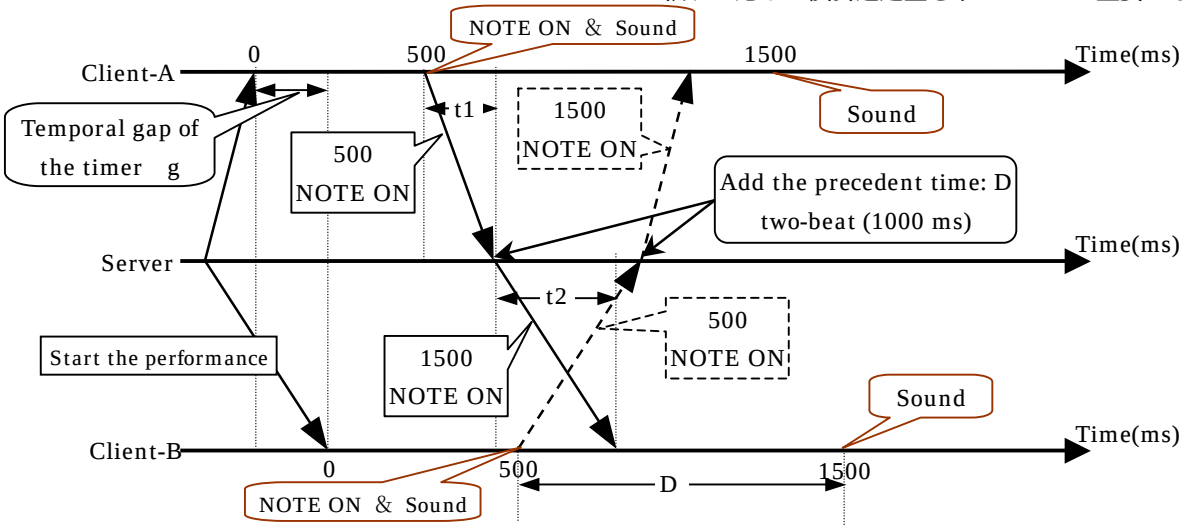

Fig. 4: Flow of the process

しかしながら、これまでの議論とは異なり、演奏者の相互の関係は非対称となる。以下では、具体的な説明を行う。

Fig.5 は、テンポ変化が生じた場合のイメージ図である。この場合、Client-A 側でのみ先行演奏を行い、Client-B での演奏を聴きながら演奏を進める。一方、Client-B では Client-A で行われた演奏に合わせて演奏が行われる。Client-A でテンポ変化が生じたとする。このとき、Client-A から Client-B へ演奏データを送信する場合は、発音時刻に一定の演奏遅延量（図中の Delay）を加算することで、Client-A 側の演奏情報をテンポ変化の情報を含めて伝えることができる。しかしながら、Client-B から Client-A へのデータ送信では、Client-A でテンポ変化が生じている。したがって、テンポ変化に応じた演奏遅延量（図中の Delay'1 と Dlay'2）を算出し、Client-A でのテンポに同期させて音を発音させる。

テンポ変化への対応はサーバ上で行う。テンポ変化へ対応するための処理には、初期化時の発音時刻の変更処理（静的変更）と、演奏時の変更処理（動的変更）の２段階がある。以下で、それを説明する。

5.1 楽譜における発音時刻変更

サーバは、前もって演奏される曲の楽譜情報を持つ。この楽譜情報には、演奏テンポ、音程とその音の発音時刻が含まれている。

まず、演奏開始前に楽譜情報の発音時刻の変更を行う（静的変更）。サーバでは、演奏開始前にテンポの設定が行われる。サーバは、設定されたテンポで曲を演奏した場合における、それぞれの発音時刻を計算しておく。この情報を用いて楽譜の再構成を行う。

以降の処理は、新しい楽譜に基づいて行うものとする。

元の楽譜情報から得られるテンポを *Tempo*、その楽譜における演奏間隔（発音時刻の差）を *Interval* とする。また、設定されたテンポを *Tempo'*、求めたい発音時刻を T_{n+1}、それより一つ前のそれを T_n とすると、T_{n+1} は以下の式（1）で求められる。

$$T_{n+1} = T_n + Interval * \frac{Tempo}{Tempo'} \qquad (1)$$

5.2 演奏遅延量の変更

演奏中、サーバは、クライアントから演奏データを受信後、受信したデータの発音時刻と、演奏開始前に計算した発音時刻とを比較し、テンポ変化を察知する。そして、そのデータをもとに、テンポ変化に応じた演奏遅延量を算出する（動的変更）。

サーバが持っている新たな楽譜の発音時刻（予定されていた発音時刻）と受信したデータの発音時刻（実際の発音時刻）を比較し、二つの発音時刻にずれが生じた場合、テンポ変化の生起とみなす。

テンポ変化が生起した場合、これまでの実際の発音時刻から、次の音と最新の音との演奏間隔を推定し、これと予定されている音の演奏間隔との比を用いて、新しい演奏遅延量を決定する。

最新の連続する過去 3 音の発音時刻を t_{n-2}, t_{n-1}, t_n とすると、演奏間隔（*Interval*）と演奏間隔の変化量（*IntervalChange*）はそれぞれ式 (2)、(3) で表される。

$$Interval = t_n - t_{n-1} \qquad (2)$$

$$IntervalChange = (t_n - t_{n-1}) - (t_{n-1} - t_{n-2}) \qquad (3)$$

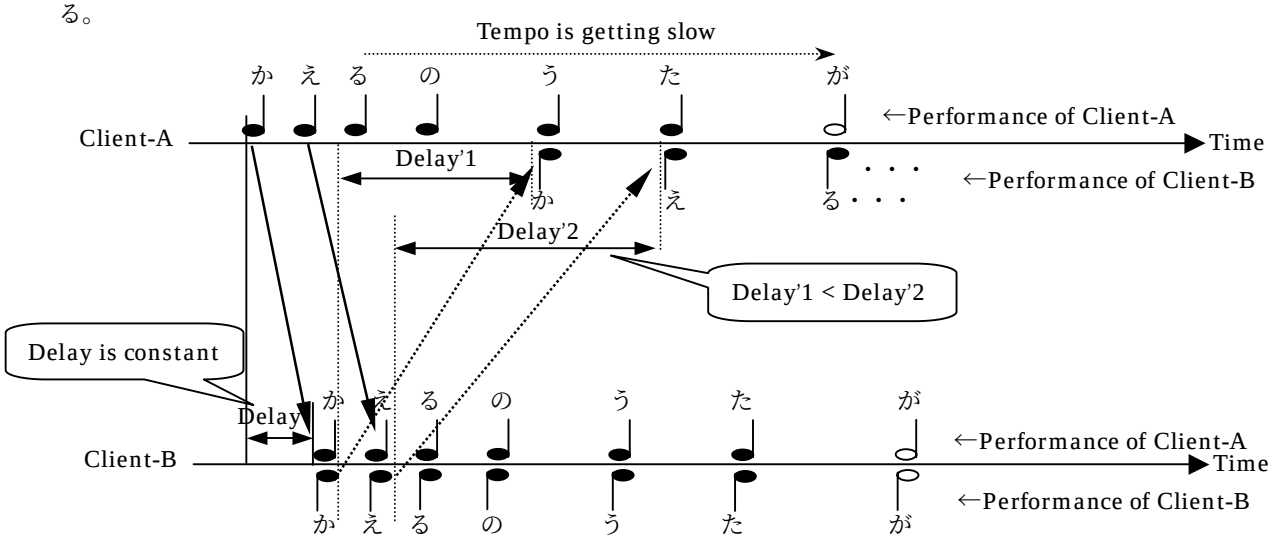

Fig.5:Image of the Change of the Tempo

予期される次の発音時刻を $\tilde{t}_{n+1}$ とし、$\tilde{T}$ を $\tilde{t}_{n+1}-t_n$ の推定値とすると、$\tilde{T}$ は式（2）に（3）を加えたものと考えられる（式（4））。

$$\tilde{T} = 2t_n - 3t_{n-1} + t_{n-2} \qquad (4)$$

予定されている発音時刻における演奏間隔を $T(=t_{n+1}-t_n)$、実際の発音時刻における推定値を $\tilde{T}$、テンポ変化前の演奏遅延量を *Delay*、テンポ変化後の新しい演奏遅延量を *Delay'* とすると、*Delay'* は式（5）で求めることができる。

$$Delay' = Delay * \frac{\tilde{T}}{T} \qquad (5)$$

サーバは、上式によって得られる演奏遅延量を受信したデータ内の発音時刻に加算し、推定発音時刻を求め、データを送信する。

5.3 動作確認

上記のようなテンポ変化機能が所望の動作をするかどうかを確認するため、動作実験を行った。

実験は、二人の演奏者 A,B による演奏で、A がテンポを変化させ、B は A の演奏に合わせて演奏するという方法をとった。B 側では、A の演奏と B 自身の演奏を聴くことができる。B からは、B が聴いていた音が A に送信される。A 側では、B の演奏とそのときに B 側で発音されていた A の演奏を、遅延を伴って聴く。今回用いた曲はきらきら星で、テンポ 120bpm で演奏を開始し、演奏中の任意の場所で、テンポを速めたり遅くしたりする演奏を２回行った。

実験の結果、サーバがテンポ変化に応じて演奏遅延量を算出していることが確認できた。Fig.6,7 に、演奏遅延量の変化のグラフを示す。また、演奏遅延量の精度を調べるため、A 側における、A の演奏の発音時刻と、そのときに A が聴いていた B と A の演奏の発音時刻を比較した。発音時刻の誤差の平均値を Table.1 に示す。

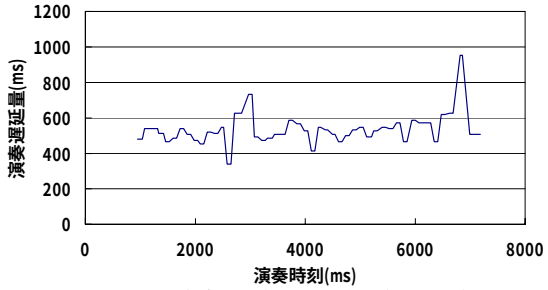

Fig.6:演奏遅延量の変化（１回目）

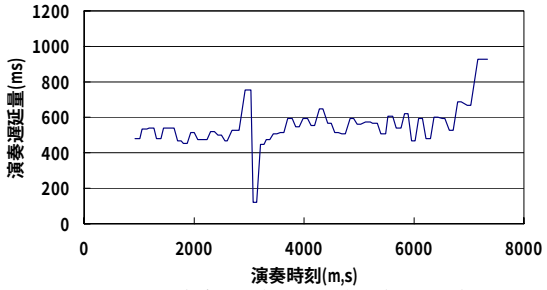

Fig.7:演奏遅延量の変化（２回目）

Table.1 において、発音時刻の誤差がすべて負になっている。これは、サーバで調整が行われたデータの発音時刻が、A の演奏の時刻よりも早くなっていたことを示す。今回の実験では、テンポ 120bpm の曲では、１拍の長さは、500ms となる。このことから、今回の結果は、一回目ではテンポ 120bpm の曲における半拍、二回目は約１拍程度の発音時刻の差があったことがわかる。

Table.1:発音時刻の誤差の平均(ms)

	一回目	二回目
教師側での教師の演奏と教師の演奏	-259.813	-437.063
教師側での生徒の演奏と教師の演奏	-273.433	-449.931

また、今回行った実験では、A のテンポ変化を B へ伝えることは可能であったが、A 側で B 側での演奏を発音させると、自分の演奏がわかりにくくなるという問題が見られた。今後は、A 側で A 自身の演奏が行いやすくなるような方策が必要であると思われる。さらに、演奏者 A のテンポが緩から急へ変化する場合、推定発音時刻が逆転するという現象が起きてしまう。この問題に対する方策も必要である。ただし、急激なテンポの変化が起こらない限り、本提案は有効であるといえる。

6. まとめ

本論文では、ネットワークを介したアンサンブル演奏のための演奏形態、Mutual Anticipated Session (M.A.S.)を紹介し、M.A.S.システム.へ追加したテンポ変化への対応方法について報告した。

今後は、この新しいシステムを用いた応用例として、遠隔音楽教室システムを考える予定である。

謝辞

本研究は文部科学省科研費（課題番号 14580442）ならびに本学 SVBL「複雑系環境制御マネージメント」

プロジェクトによっている。ここに謝意を表する。

参考文献

[1] 後藤,根山;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:情報処理学会論文誌, Vol.43, No.2, pp.299-309, (2002)

[2] 長島; GDS(global delayed session) Music の拡張モデルについて:情報科学技術フォーラム 2002,(2002).

[3] 後藤, 橋本; MIDI 制御のための分散協調システムー遠隔地間の合奏を目指してー:情報処理学会音楽情報科学研究会研究報告,93-MUS-4-1,Vol.93, No.109, pp.1-8,(1993)

[4] 後藤,根山,他; RMCPーRemote Media Control Protocolー 時間管理機能の拡張と遅延を考慮した遠隔地間の合奏:情報処理学会音楽情報科学研究会研究報告, 97-MUS-21-3, Vol.97, No.67, pp.13-20, (1997)

[5] Goto, M., et al;RMCP Remote Music Control Protocol –Design and Applications-:Proc. of the 1997 ICMC, ICMA,pp446-449,(2997)

[6] Fober, D.;Real-time Midi data flow on Ethernet and the software architecture of MidiShare: Proc of the 1999 ICMC, ICMA, pp.311-313,(1999)

[7] Fober, D., et al; Real Time Musical Events Streaming over Internet: Proc of the International Conference WEB Delivering of Music IEEE2001, pp.147-154,(2001)

[8] 大部,吉田,米倉; 楽音間で同期の取れない音場における演奏者間プロトコル:ヒューマンインタフェース学会研究報告,Vol.5, No.3, pp.15-18, (2003)

[9] Obu, Kato, Yonekura; M.A.S.:A Protocol for a Musical Session in a Sound Field where Synchronization between Musical Notes is not guaranteed; Proc. of the 2003 ICMC, ICMA, pp.309-313, (2003)

[10] NIST Net URL;http://www.antd.nist.gov/tools/nistnet