

ローカル・ラグ制御機能と同期機能を持つ 音響サーバの性能評価

Experimental Evaluation of Acoustic Server Having Local-lag Control and Synchronization Functions

岩原正典† 竹森 幸輝† 前田 佳奈†
Masanori Iwahara Kouki Takemori Kana Maeda
片桐 滋† 大崎 美穂†
Shigeru Katagiri Miho Ohsaki

1. はじめに

広帯域ネットワークの普及に伴い、大容量の音声・映像信号の双方向伝送を利用した、遠隔地間におけるコラボレーション支援の試みが行われるようになってきた[1]。しかしながら、ネットワークや計算処理に不可避免的に伴う遅延は、特に同期を必要とするコラボレーション作業、同時作業の阻害要因となり、その解決が求められている。この問題の軽減を目指し、我々は、通信経路上の遅延（ラグ）の存在を前提としたローカル・ラグ法[2]に着目し、特に音メディアを用いた同時作業（例えば遠隔合奏など）を支援する手法の開発を進めてきた[3][4][5]。ローカル・ラグ法とは、発信側のメディア信号に受信側に生じる遅延と同程度の遅延を追加することで、利用者間のメディア知覚における時間的なずれの軽減を図る手法である[2]。

入江らは、2 地点間の通信に伴う遅延を自動的に計測し、その結果に基づいて遅延量を音信号に付加することができる通信用音響サーバを構築し、それを用いた遠隔合奏の実験を通し、通信遅延が大きい環境においてローカル・ラグ法が演奏の質の低下を回復する効果を持つことを示している[3]。しかし、そこで構築された音響サーバは、3 地点以上の多地点間接続には対応しておらず、またサーバどうしの時刻同期処理やサーバ内計算に伴う処理時間の算定なども十分ではなく、得られた遅延量制御の精度には限界があった。この問題を踏まえ、竹森、前田らは、多地点接続機能を持ち、かつ正確なサーバ間時刻同期機能をも備えた新しい音響サーバを開発し、その基本的な性能評価結果を報告している[4][5]。

[4]では、通信処理時間などに関する測定データが示され、さらに[5]においては、新しい音響サーバを用いた遠隔合奏およびローカル・ラグに関する実験も可能であることが示されている。しかし、そこで行われた測定の時間はやや短く、また、サーバが扱う音データのバッファサイズ（以下、データサイズと呼ぶ）等の適切な選択に関する吟味も必ずしも十分になされていない。長時間の安定な同時作業支援に耐え得る音響サーバの開発のためには、より詳細なサーバ性能の調査と、必要に応じた一層の改良が必要である。本稿は、こうした観点に立ち、[4]で構築した音響サーバの通信処理の時間的な安定性の分析に焦点を合わせた、長時間（1 時間程度）測定と様々なデータサイズ値の設定を伴う体系的な性能調査を行い、その結果を報告するものである。

2. ローカル・ラグを用いた遠隔合奏支援

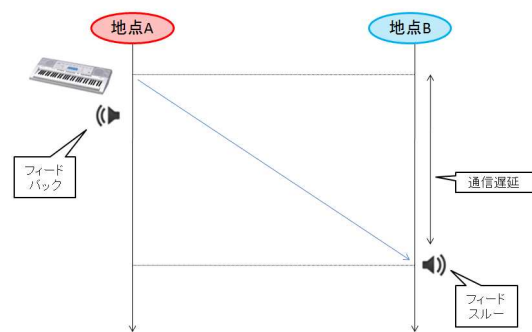


図1 遠隔合奏に伴う音データの流れ。

本研究では、遠隔同時作業の実験課題として遠隔合奏を採用している。その遠隔合奏に伴う音データの流れを、図1に図解する。図では地点Aと地点Bの2地点を接続した合奏が想定されている。図中の縦軸は時間軸である。

地点Aで演奏者が演奏した音データがネットワークに入力されている。地点Aの演奏者が聞く音をフィードバック音、地点Bの演奏者に届けられる音をフィードスルー音とする。このとき、その地点Aの入力音データは地点Bに通信遅延分だけ遅れて届く。地点Bの演奏者は、この遅れた音に合わせて演奏を試み、その演奏音は演奏段階で既に地点Aの演奏音に遅れてしまう。この遅れた地点Bの演奏音は、さらに新しい通信遅延を伴って地点Aに届けられ、地点Aの演奏者はその遅れた地点Bの演奏に合わせてしようと、自らもまた演奏を遅らせようとする。こうしたことが繰り返され、結果的に、両地点の演奏者が聞く相手の演奏音のずれは増大し、やがて演奏は破たんしてしまう。

この問題に対し、ローカル・ラグ法は、演奏地点のそれぞれにおいて、演奏者が聴く演奏音から遅延を取り除くことで演奏の破たんを防ぐことを試みる。ローカル・ラグ法を用いる遠隔合奏に伴う音データの流れを図2に図解する。ローカル・ラグとは図に示すように、通信遅延と同量の遅延をフィードバック音に加え、演奏者がともに同時に同じ音を聴こえるようにするものである。演奏者は自身の演奏音をローカル・ラグ・フィードバック音として遅れて聴くことになるが、合奏者同士の聴取のレベルにおいて同期をとることで合奏の同時性を確保することが目指されている。なおここで、簡単化のために、

†同志社大学大学院 工学研究科

Graduate School of Engineering, Doshisha University

演奏音は音響サーバやネットワーク機器類を経由してのみ出力される，すなわち気導音は聴取できない状況を仮定している．歌唱や伝統的なアコースティック楽器の演奏に伴う遅延のないフィードバック音の処理については，[3]で試みられている拡張ローカル・ラグ法などによる対処を想定している．

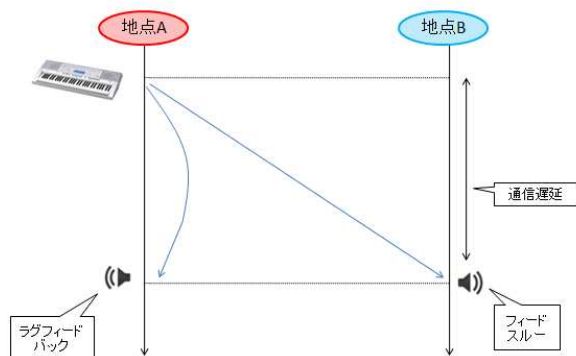


図2 ローカル・ラグを適用した遠隔合奏に伴う音データの流れ。

3. 遠隔合奏支援システム

3.1 遠隔合奏支援システムの構成要素

調査対象である音響サーバを含む遠隔合奏支援システムは，同期制御サーバと音響サーバから構成される．同期制御サーバは，時刻同期機能と，音響サーバ同士を接続させる機能などを持つ．一方，音響サーバは，音声入出力機能と，共通タイムスタンプを伴うファイル書き出し機能（音データ書き出し機能）などを持つ．同期制御サーバは接続地点のいずれかに1台あればよい．これに対し，音響サーバは演奏を行う各地点にそれぞれ1台必要である．いずれのサーバもLinuxをOSとする計算機上で動作する．

音響サーバと同期制御サーバの双方を実装した計算機の基本的諸元を以下に示す．

- ・OS: Fedora14
- ・カーネル: 2.6.35.6
- ・サウンドドライバ: ALSA
(I/O ライブラリである PortAudio を介して使用)
- ・サンプリングレート: 44.1kHz
- ・量子化ビット: 16bit
- ・チャンネル数: 1 (モノラル)

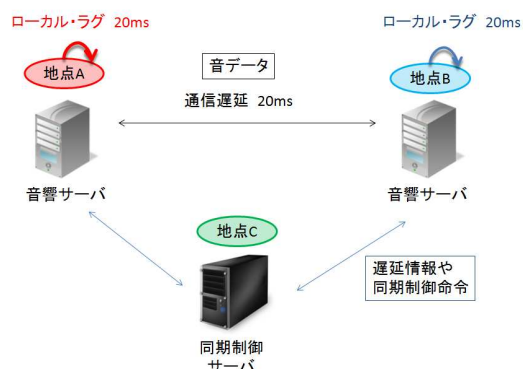


図3 2地点間におけるシステムの構成例。

3.2 システムの構成例と処理の流れ

3.2.1 2地点間におけるシステムの構成例

2地点間におけるシステムの構成例を以下の図3に示す．例えば通信遅延が20msである場合，地点Aから入力された音は地点Bに20ms遅れて出力される．ここにローカル・ラグ法を適用し，地点Aと地点Bの演奏者が同時に同じ音を聴取できるようにするためには，地点Aのフィードバック音に20msの遅延をかけることになる．

2地点間における同期制御サーバと音響サーバの主な計算処理を以下に示す．

- ①同期制御サーバは自身が所持している設定ファイルから音響サーバの情報を取得する．
 - ②同期制御サーバは音響サーバからの接続を受け付け，自身と音響サーバを接続する．
 - ③同期制御サーバは，音響サーバと時刻同期を行い，各音響サーバのタイマの値を同期させる．タイマについては3.4節で詳しく述べる．
 - ④同期制御サーバが各音響サーバ同士を接続させる．
 - ⑤音響サーバ同士が遅延の計測を行う．
- 以上の処理を経て，合奏が出来る状態が確立される．

3.2.2 3地点間におけるシステムの構成例

3地点間におけるシステムの構成例を以下の図4に示す．

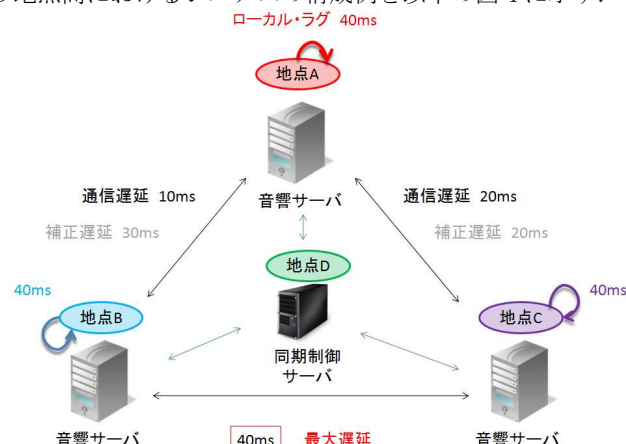


図4 3地点間におけるシステムの構成例。

3地点間の場合も，全地点の演奏者が同時に同じ音を聴取できるようにするという考え方にに基づき，全ての地点の中の最大遅延を基準に音データの遅延設定を行って支援システムを構成する．図の例では地点A-地点B間の通信遅延が10ms，地点A-地点C間の通信遅延が20ms，地点B-地点C間の通信遅延が40msであり，最大遅延は地点B-地点C間の40msである．地点Aに着目した場合，地点Aで聴こえる音は，自身のフィードバック音と，地点Bからのフィードスルー音，地点Cからのフィードスルー音の3つである．まず自身のフィードバック音に関しては，遅延が0msのため，地点Aにおけるラグ・フィードバックの遅延量は最大遅延に合わせて40msに設定する．続いて地点Bからのフィードスルー音に関しては，元々10msの通信遅延を伴って地点Aに届くため，最大遅延の40msに合わせるため30ms分の補正遅延を追加してから出力する．最後に地点Cからのフィードスルー音に関しては，元々20msの通信遅延を伴って地点Aに届くため，20ms分の補正遅延を付加してから出力する．これにより自身の

フィードバック音と、地点 B からのフィードスルー音、地点 C からのフィードスルー音のそれぞれが 40ms 遅れて同期的に聴こえるという状況を作り出すことができる。地点 B と地点 C においても同様の処理を行うことにより、全地点で 40ms の遅延は伴うものの、同時に同じ音を聴取できる状況を作り出すことができる。

この考え方はより多数の地点から構成される場合にも同様に適用でき、最大遅延と補正遅延を求めることによって、3 地点間以上におけるローカル・ラグ制御が可能になる。

3 地点間における同期制御サーバと音響サーバの主な処理を以下に示す。

- ① 同期制御サーバは自身が所持している設定ファイルから音響サーバの情報を取得する。
- ② 同期制御サーバは音響サーバからの接続を受け付け、自身と音響サーバを接続する。
- ③ 同期制御サーバは、音響サーバと時刻同期を行い、各音響サーバのタイマの値を同期させる。
- ④ 同期制御サーバが各音響サーバ同士を接続させる。
- ⑤ 音響サーバ同士が遅延の計測を行い、最大遅延を各音響サーバに配信する。
- ⑥ 各地点の音響サーバは受信した最大遅延を基に補正遅延を算出し合奏が出来る状態を生成する。

3.3 音データのフロー

3 地点間での音響サーバ内の音データのフローを以下の図 5 に示す。図では、内部の処理機能が記述されている音

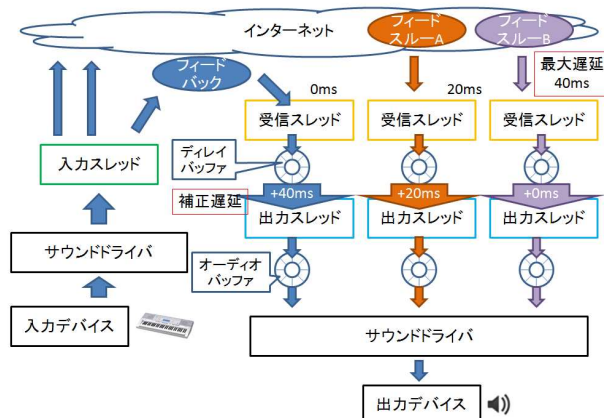


図 5 音響サーバ内の音データフロー。

響サーバがインターネットを経由して A（フィードスルー音の送信元）と B（フィードスルー音 B の送信元）の 2 台の音響サーバと接続されていることが想定されている。図中の主な処理は以下のように要約される。

◆**入力スレッド**：入力デバイスから音データを取得し、接続している全ての音響サーバに音データを送信するスレッド。

◆**受信スレッド**：対応する送信元から送られてくる音データを受信し、受信時刻を記録してディレイバッファに格納するスレッド。音データのチャンネル毎に 1 つ用意される。

◆**ディレイバッファ**：受信時刻が付加された音データを格納するリングバッファ。補正遅延量に基づいてこのバッファから音データを取り出すタイミングを調整することで、ローカル・ラグ法のラグ付加機能を実現する。

◆**出力スレッド**：取り出し時刻を求めてディレイバッファから音データを取り出し、オーディオバッファに転送するスレッド。受信スレッド毎に 1 つ用意される。

◆**オーディオバッファ**：サウンドドライバに渡すデータを格納するリングバッファ。音データはこのバッファから PortAudio のコールバック関数を使用してサウンドデバイスに渡される。

図 5 では、まず音データが入力されると、入力スレッドが音データを入力デバイスから受け取り各音響サーバに送信する。ここでは 3 地点接続を想定しているため、自身を含む 3 つの地点にデータが送信されるが、フィードバックの転送に関してはインターネットを介せずにサーバ内に閉じて行われる、受信スレッドは音データを受信すると自サーバの時刻情報を用いて、受信時刻を取得し音データのヘッダに付加し、ディレイバッファに格納する。出力スレッドは取り出し時刻に従ってディレイバッファから音データを取り出し、オーディオバッファに転送する。取り出し時刻は自サーバ内の現在時刻からその音に付加する補正遅延時間を引くことによって算出する。この時刻がデータに付与された受信時刻を超えていれば、バッファから音データを取り出す。オーディオバッファに格納された音データは PortAudio のコールバック関数によってサウンドドライバに渡され、ミキシングされた後、1 チャンネルの音として出力デバイスに転送される。ディレイバッファ、オーディオバッファの 2 つのバッファを用いた処理機構を採用した理由であるが、ディレイバッファで処理されるデータには時間情報が付加されており、そのデータをサウンドドライバに渡した場合、サウンドドライバで正常な処理を行うことが出来なくなる。そのため、時間情報が付加されたデータを出力スレッドで取り出し時刻に従って取り出し、オーディオバッファに渡すことで、サウンドドライバで正常に処理が行うことが出来る機構を取っている。

3.4 独自タイマ

本遠隔合奏支援システムは、サーバ間の同期処理の基準となる独自のタイマを、システムコールである clock_nanosleep 関数を用いて実現している。タイマ専用のスレッドを用意し、その中で clock_nanosleep 関数を用いて 0.1ms スリープさせたあと、int 型の数値をカウントアップする処理を無限ループさせる。この処理により、タイマは原理的には 0.1ms 間隔でそのカウンタを更新する。このタイマは同期制御サーバと音響サーバの両方に実装されている。

リアルタイムスケジューリングを適用し、タイマ機能を実現するスレッドに対して優先度を最も高く割り当てた場合、このタイマは、原理的な理想値 0.1ms に近い 0.11ms の間隔で動作することを確認している。

3.5 時刻同期

3.4 節に紹介したタイマを使用して、時刻同期機能を実現する。具体的には、同期制御サーバが自身のタイマのカウント数を各音響サーバに配信し、その値に従って各音響サーバが自身のタイマのカウント数を補正することによって各音響サーバのタイマの同期を確保する。同期制御サーバから各音響サーバにタイマのカウント数を配信する際に、同期制御サーバと各音響サーバ間に通信遅延が発生するため、その遅延を考慮した上でタイマのカウント数を配信する。すべての音響サーバのタイマが同期するまでの手順を以下に示す。

- ① 接続を予定していた全てのサーバが接続を済ませたことを確認し時刻同期が開始される。
- ② 同期制御サーバが、各音響サーバとの通信遅延を計測する。
- ③ 同期制御サーバがリセット値として現在のタイマのカ

ウント数に通信遅延をタイマ換算した値を付加し、各音響サーバに送信する。

- ④各音響サーバは同期制御サーバから取得したリセット値を自身のタイマの値に反映させる。

以上の手順で各音響サーバのタイマが同期される。タイマ同期後、同期制御サーバが指定時刻に録音ファイルをオープンするという命令を各音響サーバに送信すれば、共通のタイムスタンプを持つ録音ファイルが作成できる。

4. 性能評価実験

4.1 遠隔合奏支援システムのフィードバック音精度計測

初めに、音響サーバに入力された音データをフィードバック音として直接出力する際に音響サーバ内の計算処理が引き起こす遅延の量と特性を、データサイズを変化させて調査した。この処理は、音響サーバの動作の基本に位置するものである。ここで、データサイズとはバッファに対して読み書きする short 型の配列の長さである。

4.1.1 実験環境と実験手順

実験では2台のコンピュータを使用し、1台は同期制御サーバとし、もう1台は音響サーバとした。以下に2台のコンピュータの諸元を示す。

コンピュータ A (音響サーバ)

- ・CPU: Intel(R) Core(TM)2 CPU 6400 2.13GHz

コンピュータ B (同期制御サーバ)

- ・CPU: Intel(R) Core(TM)2 Quad CPU Q6600 2.40GHz

構築した実験環境を以下の図6に示す。

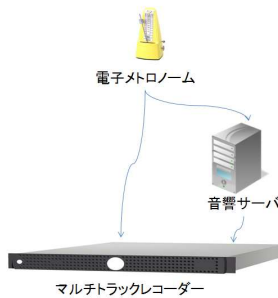


図6 実験環境。

実験では、120BPMの電子メトロノームを音源とし、電子メトロノームから直接出力される音と、音響サーバAを経由して出力される音とを、マルチトラックレコーダーに入力・録音し、それら両者の音の時間的ずれを計測することで遅延の特性を調査した。構築した環境でシステムを動作させ、データサイズを16から32, 64, 128, 256, 512, 1024と変化させて実験を行った。1回の計測は60分間行い、実験で得られた2つの録音ファイルの0分(ファイルの1音目)から60分までのデータに対し、5分ごとにメトロノームから直接出力される音データと音響サーバAから出力される音データのずれを計測した。

4.1.2 実験結果と考察

2音データのずれの計測結果を以下の図7から図13に示す。図の縦軸は、メトロノームから直接出力される音と音響サーバAから出力される音データのずれである。図の横軸は5分ごとの計測時点を示している。

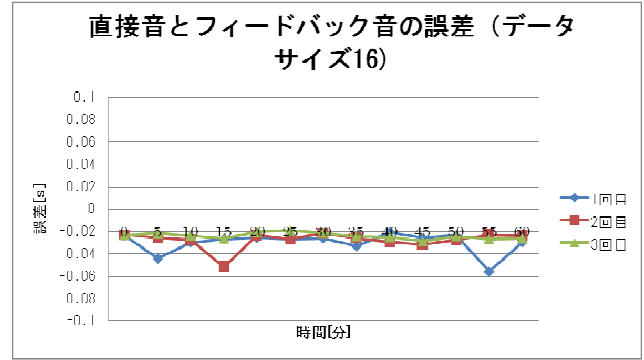


図7 フィードバック音に伴う誤差(データサイズ16)。

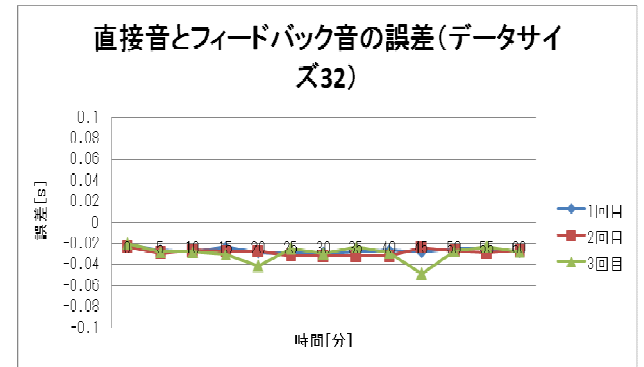


図8 フィードバック音に伴う誤差(データサイズ32)。

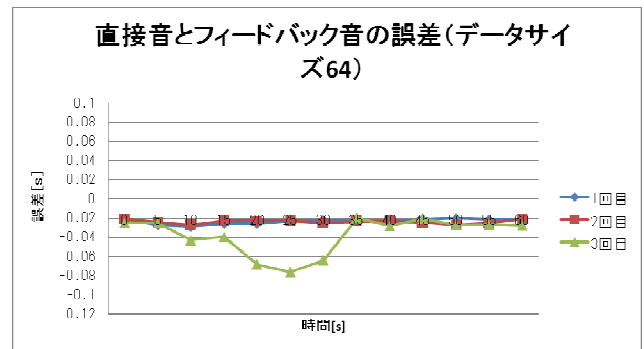


図9 フィードバック音に伴う誤差(データサイズ64)。

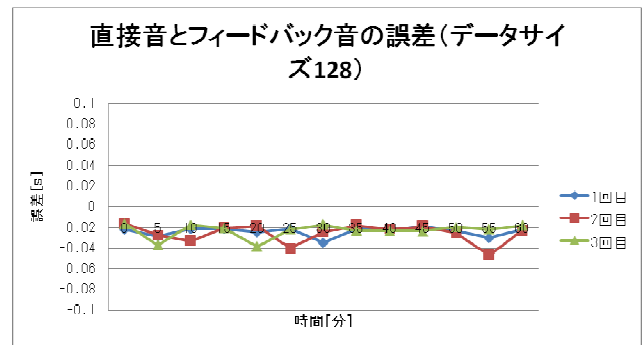


図10 フィードバック音に伴う誤差(データサイズ128)。

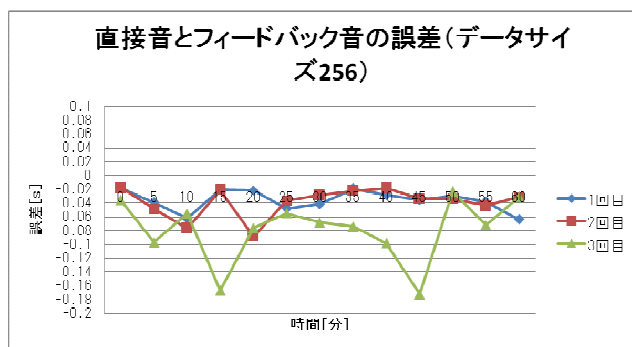


図 11 フィードバック音に伴う誤差(データサイズ 256)。

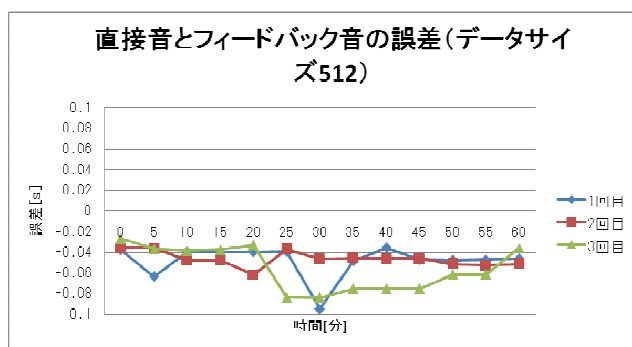


図 12 フィードバック音に伴う誤差(データサイズ 512)。

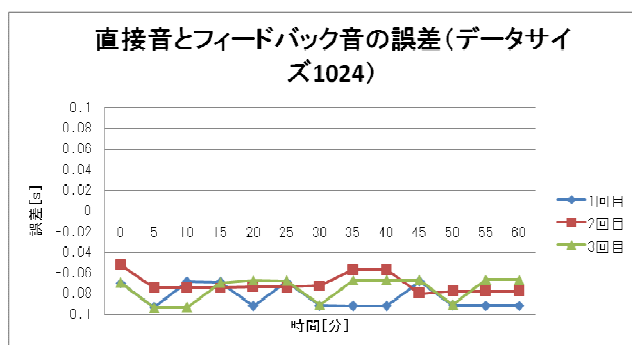


図 13 フィードバック音に伴う誤差(データサイズ 1024)。

計測結果からデータサイズが大きい場合と比較して、データサイズが小さい場合に本システムはより安定して動作することが分かった。これはバッファにデータを格納する、またはバッファからデータを取得する際に、データサイズ単位で処理を行っていることから、データサイズが大きい場合 1 つのデータを格納できなかった場合、または 1 つのデータを取得出来なかった場合のずれが大きくなっていくためであると考えられる。例えば、データサイズ 16 の音データのフレーム長は約 0.36ms であるのに対しデータサイズ 1024 の音データのフレーム長は約 23.2ms である。従って、データの格納あるいは取得に失敗した場合のずれは、後者は前者のおよそ 70 倍のずれを最小単位とするずれを引き起こす可能性があるためと考えることができる。

データサイズが比較的小さい場合の結果から、それらの条件においては、サーバ内の計算処理（ほとんどが入出力デバイスやメモリに対するデータの読み書き）には平均的に 20ms 程度の時間がかかっていることがわかる。

また、遅延（ずれ）の変動は、不規則に発生していることもわかる。サーバ内におけるデータの処理には、複数のスレッドに加えて、入出力デバイス用のドライバが、複雑に関わっている。上記のずれに伴う不規則さは、これらの処理モジュールの（データ読取等の）繰返し動作

の途中になんらかの原因による不規則な遅滞等が生じるためと考えられる。

4.2 スレッドの優先度に対する書き込みデータサイズの計測

スレッド動作は OS の管理下であり、当然ながらその遅滞等の発生には動作の優先度設定が関係し得る。この点を精査するため、スレッドの優先度を変化させ、この変化がバッファへの音データの格納とバッファからの取得にかかわる遅延に及ぼす影響を調査した。ここでの遅延は、バッファに書き込まれるデータ量（データサイズを単位長としたときのバッファに書き込まれている単位数）によって計測した。

4.2.1 実験環境と実験手順

実験では以下の図 14 に示すように、

- ①ディレイバッファに音データを格納する時刻、
 - ②ディレイバッファから音データを取得する時刻、
 - ③オーディオバッファにデータを格納する時刻、
 - ④オーディオバッファからデータを取得する時刻
- の計 4 つの時刻におけるそれぞれのバッファの書き込みデータサイズ（上記の書き込まれたデータの単位数）を計測した。

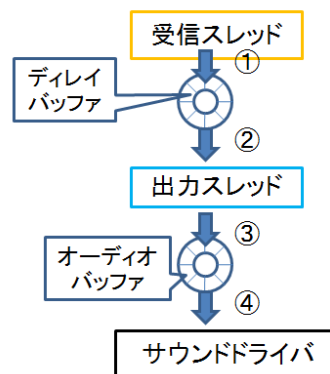


図 14 書き込みデータサイズを表示させる場所。

データサイズは、特にフィードバック音に伴う誤差が大きかったデータサイズである 1024 とした。スレッドの優先度の設定は、

- I. タイムスレッドの優先度を最も高く設定した場合と、
- II. タイムスレッドの優先度を最も低く設定した場合、
- III. 受信スレッドの優先度を最も高く設定した場合、
- IV. 受信スレッドと出力スレッドの優先度を最も高く設定した場合

の 4 種類を調査した。計測は 60 分間行った。

4.2.2 実験結果と考察

スレッドの優先度を下記の 4 通りに設定した場合の、計測開始後 15 分間の計測結果を以下の図 15 から図 30 に示す。図中、縦軸の書き込みデータサイズは、バッファ中に書き込まれたデータサイズであり、横軸はサーバ計算機自身が持つシステム時刻である。書き込みデータサイズはバッファに音データを格納する時に 1 増加し、バッファから音データを取得する時に 1 減少するため、システムが正常に動作している時の①、②、③、④の時刻での理想の書き込みデータサイズはそれぞれ 1, 0, 1, 0 である。

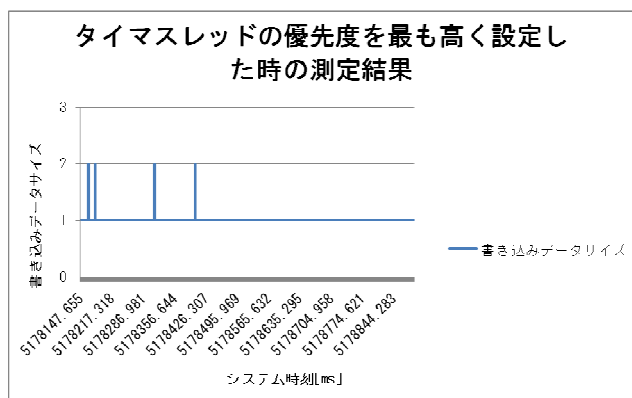


図 15 ディレイバッファに音データを格納する時刻での書き込みデータサイズ.

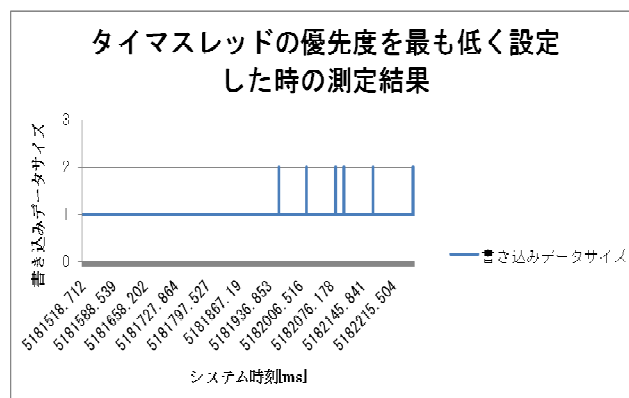


図 19 ディレイバッファに音データを格納する時刻での書き込みデータサイズ.

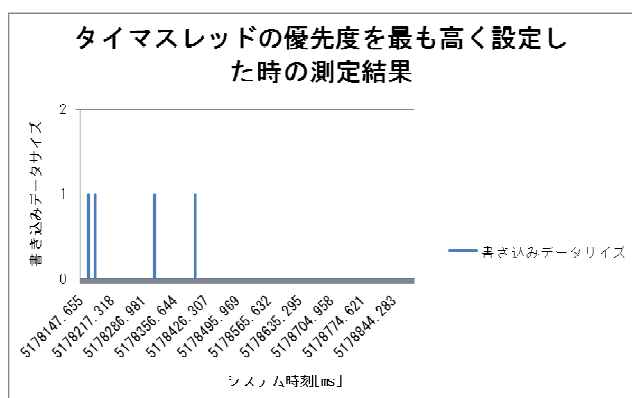


図 16 ディレイバッファから音データを取り出す時刻での書き込みデータサイズ.

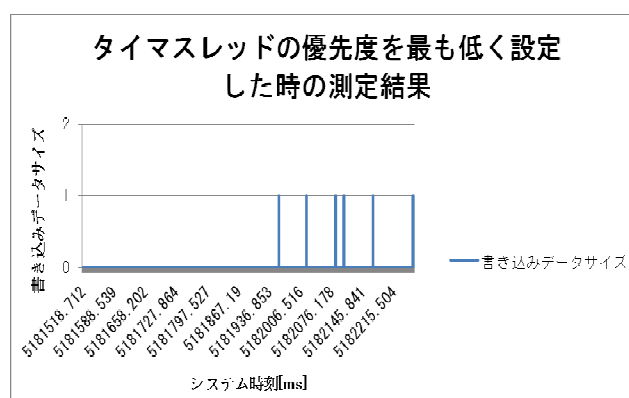


図 20 ディレイバッファから音データを取り出す時刻での書き込みデータサイズ.

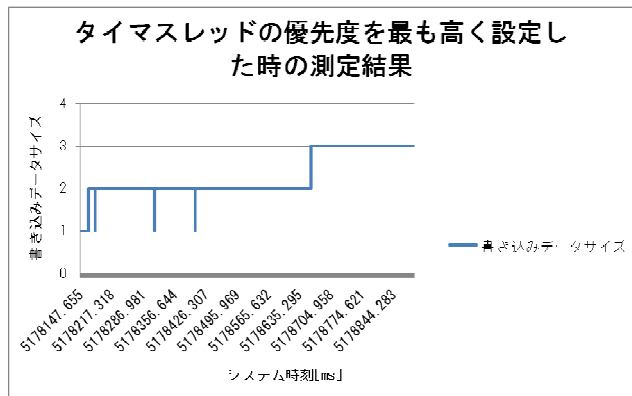


図 17 オーディオバッファに音データを格納する時刻での書き込みデータサイズ.

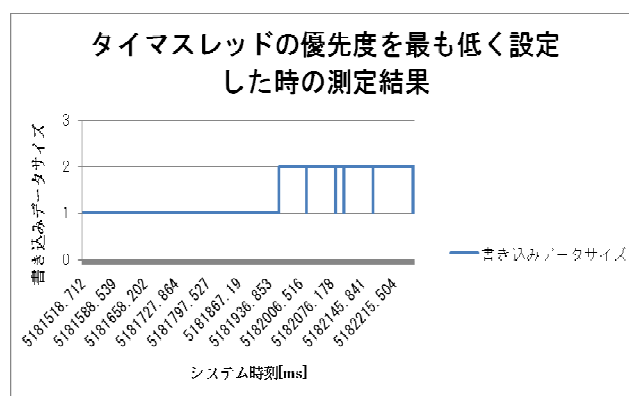


図 21 オーディオバッファに音データを格納する時刻での書き込みデータサイズ.

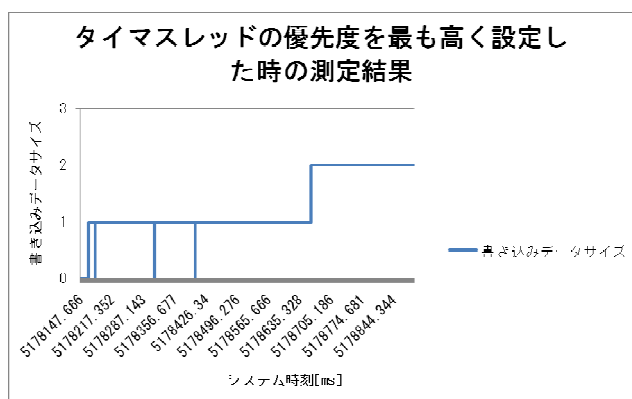


図 18 オーディオバッファから音データを取り出す時刻での書き込みデータサイズ.

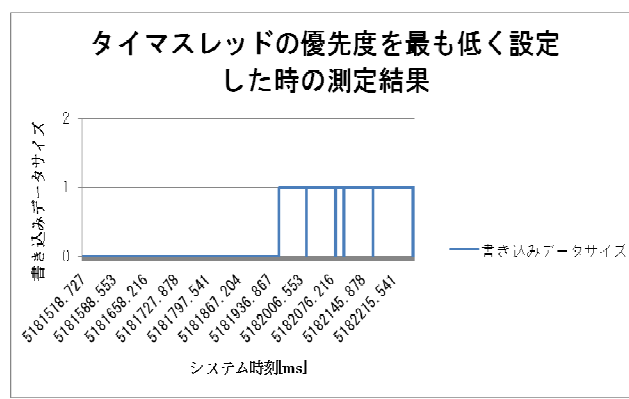


図 22 オーディオバッファから音データを取り出す時刻での書き込みデータサイズ.

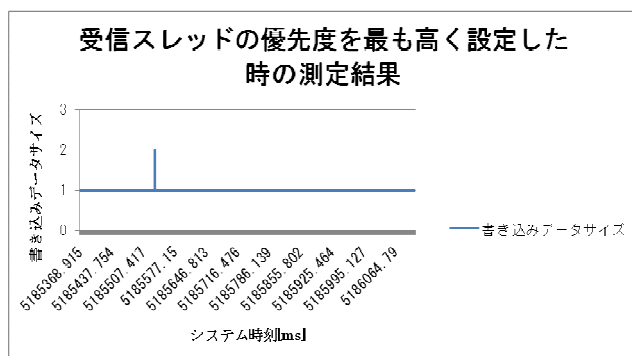


図 23 ディレイバッファに音データを格納する時刻での書き込みデータサイズ。

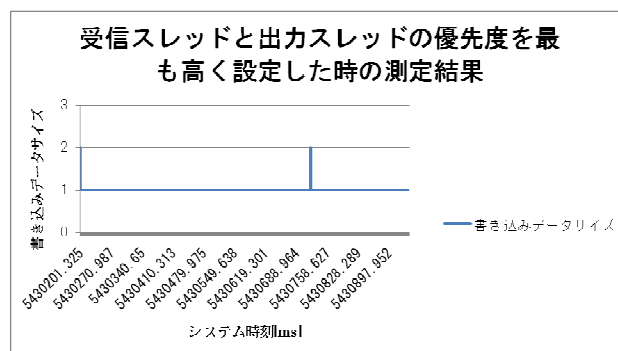


図 27 ディレイバッファに音データを格納する時刻での書き込みデータサイズ。

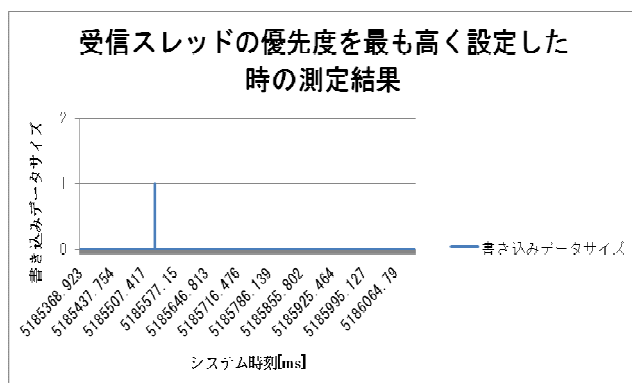


図 24 ディレイバッファから音データを取り出す時刻での書き込みデータサイズ。

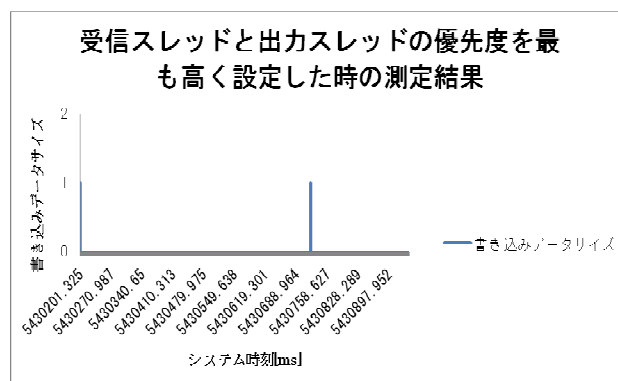


図 28 ディレイバッファから音データを取り出す時刻での書き込みデータサイズ。

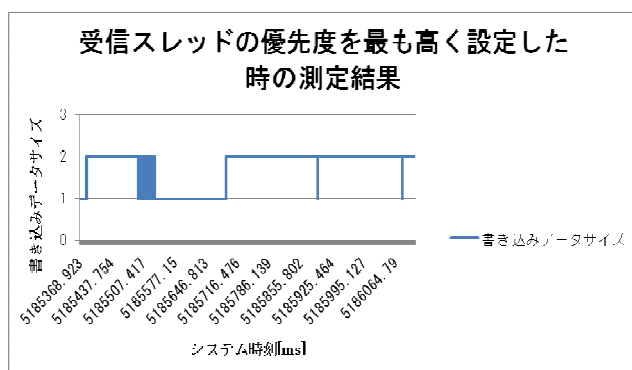


図 25 オーディオバッファに音データを格納する時刻での書き込みデータサイズ。

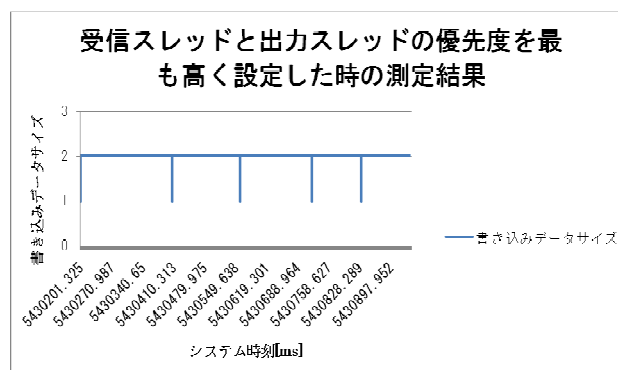


図 29 オーディオバッファに音データを格納する時刻での書き込みデータサイズ。

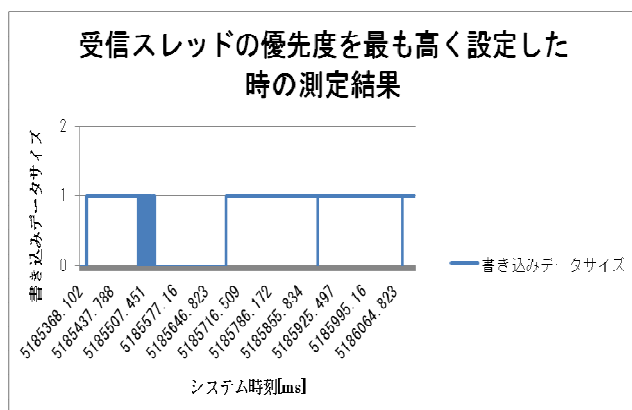


図 26 オーディオバッファから音データを取り出す時刻での書き込みデータサイズ。

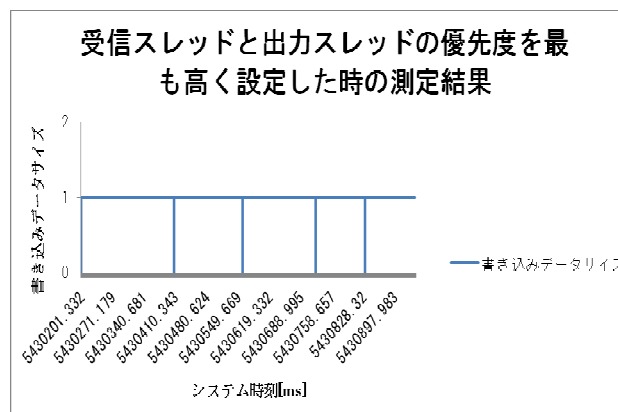


図 30 オーディオバッファから音データを取り出す時刻での書き込みデータサイズ。

実験結果から、オーディオバッファにデータを格納・取得する場合と比べ、ディレイバッファにデータを格納・取得する場合における書き込みデータサイズの変動が小さかった。これより、ずれが発生するタイミングは、オーディオバッファでデータの受け渡しを行っているタイミングに強く関係付けられていることが推測される。

また、オーディオバッファにデータを格納・取得する際は、受信スレッドの優先度を高く設定した場合が最も安定していることが分かった。

60 分計測した結果、タイマスレッドの優先度を高くした場合にオーディオバッファの書き込みデータサイズが最大 4 に到達したが、それ以外では最大が 2 であった。この要因としてタイマスレッドは 0.1ms という非常に短い頻度で呼び出されるため PortAudio のコールバック関数のスケジュールを奪ってしまっていることが影響しているのではないかと推測する。

5. まとめ

新しく開発した音響サーバのデータ処理時間の安定性や精度を、1 時間の動作時間に渡って計測し、データ出力等の処理に伴う時間的ずれの程度を明らかにし、またそのずれを生じる原因の解明を試みた。その結果、サーバ内でバッファに対して読み書きするデータのブロックの大きさであるデータサイズは小さい方が、データ出力時のずれが小さく、また安定性も増す（ずれの大きさの変動が小さい）ことが確認された。

またスレッドの優先度設定を変更した計測を通して、ずれが、サウンドドライバにデータを引き渡す直前で用いるオーディオバッファに対するデータの読み書きの不安定性に起因して発生している可能性が大きいことを明らかにした。この不安定性を軽減するため、オーディオバッファに対するデータ読み書きを行うスレッドの優先度設定を高くすることで、この不安定さが相当程度軽減されることもまた明らかにした。

謝辞 本研究を進めるにあたり多くの貴重なご意見を賜りました山口毅氏に心より感謝致します。

参考文献

- [1] Kenji Hirata, Toshihiro Takada, Shigemori Aoyagi, Yoshinari Shirai, Naomi Yamashita, and Junji Yamato, The t-Room: Toward the Future Phone, NTT Technical Review, Vol4, No12, pp26-33 (2006).
- [2] Dane Stuckel and Carl Gutwin; “The Effect of locallag on Tightly-Coupled Groupware: Computer Supported Cooperative Work”, pp. 447-456 (2008).
- [3] 入江洋介, 青柳滋己, 高田敏弘, 平田圭二, 梶克彦, 片桐滋, 大崎美穂; “t-Room のための遠隔合奏支援システムの構築”, 情報処理学会グループウェアとネットワークサービス研究会研究報告, Vol. 2009-GN-73, No. 23 (2009).
- [4] 竹森幸輝, 前田佳奈, 岩原正典, 片桐滋, 大崎美穂: “ローカル・ラグ制御機能とログ同期機能を持つ音響サーバの開発”, オーディオビジュアル複合情報処理研究会 (2012. 02).
- [5] 前田佳奈, 竹森幸輝, 岩原正典, 片桐滋, 大崎美穂: “ローカル・ラグ機能を持つ音響サーバを用いた遠隔合奏の評価”, オーディオビジュアル複合情報処理研究会 (2012. 02).