

# Web 技術に基づいた遠隔共有型ルーパーの提案

渋川 大樹<sup>1,a)</sup> 横山 昌平<sup>1,b)</sup>

**概要：**2020 年初頭から、COVID-19 の影響により、リモートで行う活動が盛んになった。その結果、音楽分野でもインターネットを介した活動が行われている。また、音楽分野では楽曲などの製作時、DTM の使用が主流となっており、MIDI データなどコンピュータ上やインターネット上で扱いやすいデータ形式のものが使用されるようになった。データとして扱いやすい状況にあるにもかかわらず、音楽をデータベースで扱っている研究は少なく、そのなかで演奏そのものを扱うような演奏者のための研究はさらに少ない。そこで本研究は関係データベースを用いた遠隔地間での演奏支援の手法を提案する。本研究では、複数人が利用できるルーパーを Web アプリケーションとして作成した。このシステムにおいて遅延の観点から検討する。

## 1. はじめに

2020 年初頭から、COVID-19 の影響によりさまざまな経済活動が停滞することになった。その結果 IT 化が進み、インターネットを介した活動が活発に行われるようになった。音楽の分野においても活動に多くの制限がかけられ、ライブ活動や楽曲制作など従来のコミュニケーションを介して活動することが難しい状況にある。このような状況においても、オンラインライブや遠隔地にいる人同士とリアルタイムで合奏ができるアプリケーションである SYNCROOM<sup>\*1</sup>のように、インターネットを介した新しい表現の形としての活動が行われている。

また、シンセサイザーなどの音楽機材の規格が製品ごとに異なるため、機器間での接続が困難であったが、MIDI (Musical Instrument Digital Interface) 規格の登場により、それが解決されるようになった。そして DTM (Desktop Music) の登場により、コンピュータで音楽を扱いやすくなった。現在ではこの DTM による制作が主流であり、楽譜に音符を書き込んだり、磁気テープに録音したりすることが少なくなった。この流行により、多くの種類の DAW (Digital Audio Workstation) ソフトが販売されるようになり、音楽制作の敷居が低くなった。DAW ソフトにおいては、MIDI 規格に則って制作されたデータを扱うことが多い。この要因として、この MIDI データはデータサイズが小さく、かつ音楽情報の細部を容易に変更することが可能であることが挙げられる。MIDI データは音声情報のような

波形データではなく、どのような音が発せられたかという音源や楽器へのメッセージであり、演奏情報のみをもつため、音そのものをもたない。そのため、機器間で共有や転送がしやすいという特徴があり、インターネット上で広がった。また、コンピュータの性能の向上により、近年の DAW ソフトではオーディオも扱うことが可能になった。

さらに、Web 技術の発達により、MIDI やオーディオを Web ページ上で扱うことが可能になった。現在、W3C Audio Working Group によって提唱された Web MIDI API[1] や Web Audio API[2] によって、音楽を扱った Web アプリケーションを簡単に作成し、これを公開することで多くのユーザが利用できる状態にある。

また、5G (第 5 世代移動通信システム) の登場により、高速、大容量、低遅延かつ、多数接続が可能な状況にある。今までリアルタイムで扱うことができなかった高精細な映像音声のライブ配信や、複数の医療機器に同時に接続することなどが可能になる。

我々はこのような状況に着目し、Web 上で音楽データを扱うことで、複数人で遠隔演奏するシステムを構築する研究を進めている。このシステムを構築するためには、インターネット越しに扱うことが必要であり、さらに音楽データへのアクセスに対し、複数人が可能であり、かつ同期的で一貫性を保てるのが条件となる。そのため、データの整合性が特に必要とされる e コマースや金融などの分野においてもシステムの核となっており、高度なトランザクション処理能力を有している関係データベースに着目した。関係データベースを用いることで、遠隔のサーバで音楽データを管理することができ、Web 上でそのデータにアクセスすることが可能になる。

<sup>1</sup> 東京都立大学

Tokyo Metropolitan University

a) shibukawa-taiki@ed.tmu.ac.jp

b) shohei@tmu.ac.jp

\*1 <https://syncroom.yamaha.com/>

関係データベースによって、データの保存・取得が容易になることを活用するためにも、我々はルーパーという既存のエフェクターに着目した。ルーパーは録音・再生を繰り返しながら、任意の小節で半永久的にループ再生する。それと同時に、演奏フレーズが重ね合わさり、一つの合奏曲を作成することが可能になる。これらを踏まえて、この録音・再生をデータの保存・取得に置き換えることが可能と推測した。

我々はこれまでに音楽データの一つである MIDI データに着目し、このデータを使用した 1 人の演奏者が 1 人で合奏するシステム [3] を作成し、関係データベースが演奏データを扱うことが可能か検証した。本研究では、このシステムを拡張し、複数人かつオーディオにも対応したシステムを作成した。

また、本研究ではインターネットを介して音楽を扱う。そのため、演奏に対して遅延が発生する。特に複数人がかかわる合奏において、大きな遅延があるとリズムやタイミングがずれる。その結果、演奏に影響が及ぶため演奏が困難になる。西堀ら [4] の検証によると 50 ms 以下でなら正しい演奏が可能であり、30 ms 以下なら遅延そのものを認識しなくなるとある。このことから本論文ではシステムの説明をするとともに、遅延に関する検証結果から本システムの有効性を検証する。

本論文の構成は以下の通りである。2 章では関連研究について述べる。3 章では提案手法について述べ、作成したシステムの仕組みや、Web アプリケーションに関して説明する。4 章では遅延に関する評価実験を述べ、本システムの有効性を検証する。5 章では本研究のまとめと今後の課題、展望を述べる。

## 2. 関連研究

### 2.1 遠隔地から音楽を扱う研究

インターネットの普及にともなって、遠隔地から音楽を扱う研究 [5], [6], [7], [8], [9], [10] が行われるようになった。インターネットを利用する場合、データを送受信するために遅延が発生する。また、音楽セッションなど音楽がかかわるものには低遅延であることが望ましい。

後藤ら [6] は RMCP という音楽用通信プロトコルを設計した。これは計算機ネットワークと MIDI を融合した分散協調システムであり、コネクションレス型のプロトコルである。このプロトコルでは時刻同期機能とタイムスタンプを用いた時間管理により、低遅延を実現し、音楽情報をリアルタイムで扱うことを実現した。また、このように特別な通信プロトコルを利用することで低遅延を実現するだけでなく、遅延を利用して音楽セッションを行う方法もある。

後藤ら [7] は Open RemoteGIG という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を開発した。このシステムでは、MIDI を用いた同一のコード進行の繰り返

しを一定のテンポで演奏することを前提とする。さらに、不特定多数のユーザが参加できるようにするために、スター状のネットワーク接続によって演奏情報を一つの場所に集約し、コード進行の 1 周期の整数倍遅延して送り返すことでセッションを可能としている。これにより遅延をなくすのではなく、遅延を利用するという新しい方向性を示した。

また、これまでの遠隔セッションに対する先行研究では、演奏のテンポが一定であることが前提とされていたが、大部ら [8] は遅延を利用して、曲の途中でテンポ変化が起きたとしても同期して演奏ができるシステムを開発した。M.A.S.[11], [12] という演奏の発音時刻を調整し、インターネットによる遅延を抑制する演奏形態を改良した。このシステムではあらかじめ楽譜情報を持ち、予定される発音時刻と実際に演奏された発音時刻を比較することでテンポ変化を発見する。テンポ変化を発見した場合、これまで実際に演奏された発音時刻から、最新の発音時刻と次の発音時刻の演奏間隔を推測し、予定される発音時刻との比によって同期させている。しかしながら、急激なテンポ変化には対応しきれていない。

さらに、Web を利用して遠隔で音楽を扱った研究もあり、野原ら [10] は WebRTC を用いた音楽表現の指導を配慮した遠隔指導支援システムを制作した。WebRTC の機能によって、ブラウザ間の直接通信が可能になり、リアルタイム送受信を行え、カメラやマイクが利用可能になった。この結果、DAW の遠隔指導において、指導者がリアルタイム化された音響を聴くことで、対面と同様に問題把握をしながら指導可能になった。しかしながら、リアルタイム音楽セッションを行うには不十分という結果も得られた。

これらのような先行研究を踏まえ、我々はルーパーの特性を利用し、遅延に対しては 1 ループ遅らせることで同期させる。そして、データは MIDI だけでなく、オーディオにも対応し、関係データベースに集約することでデータの記憶、選択が可能になる。そのため、個々が演奏したいものを選択しながら、リアルタイム音楽セッションと同じように聴くことができる。また、1 ループという長さは変化しないため、そのループ内の変化に対しては演奏者の能力次第で対処可能と考えられる。

### 2.2 音楽とデータベースを扱った研究

Web にとってデータベースは特別なものであり、そのデータベースと音楽を合わせた研究もある。Maxwell ら [13], [14] は音楽情報検索に利用可能な音楽情報データベース・検索システムを提案した。これは、シンプルな 6 部構成の構文でクエリを受け付けており、データベースに格納された音楽形式のエンコードされた知識を利用して、さまざまなタイプの音楽情報を返すことが可能である。そして、アルゴリズム作曲システムの設計者がツールとして利用できるように開発を試みている。

橋田ら [15] はピアニストの演奏解釈を記述した演奏表情データベースの提案をしている。このデータは、音楽構造記述に焦点を当て、音符の継続時間や発音時間のずれなど、音楽表現に関する詳細なデータを提供し、コンピュータが可読できる。そのため、著者は音楽演奏に関する研究にこのデータベースを活用できると考えている。

音楽とデータベースを扱った研究は、音楽専用のデータベースを使用したものが多く、演奏に利用するシステムではないことが多い。我々は汎用的なデータベースを用いて、演奏に直接利用する。

## 2.3 音楽システムに関する研究

音楽システムに関する研究は数多く存在する [16], [17], [18], [19], [20], [21], [22]。佐竹ら [17] は過去の演奏履歴を活用して合奏を行うシステムを提案した。ルーパーの機能を拡張したもので、Emacs 拡張の入力の繰り返しを自動化する Dynamic Macro を利用し、過去の演奏を繰り返すことで、その演奏に対して、さらに演奏を重ねられるものとなっている。そのため、演奏中や演奏終了後にいつでも繰り返し再生を行い、合奏を開始することが可能である。

金杉ら [18] は、演奏者の身体や楽曲に合わせて、鍵の大きさを変更することが可能なソフトウェア鍵盤楽器による演奏支援を提案した。このシステムでは手動と自動の2種類のシステムを作成した。手動では演奏者の意図を直接伝えられるため、演奏がしやすくなった。一方、自動では変化が乏しく、演奏のしやすさも変化がなかった。しかしながら、結果として、鍵の大きさの変更が、演奏のしやすさの向上につながると確認できた。

音楽システムには演奏者が演奏の幅を拡張するものや、演奏をしやすくなるよう手助けするものがある。我々の研究ではルーパーを題材とするが、ルーパーそのものを拡張したものも存在する [20]。さらに、5G 環境が扱えるようになり、この環境を背景とした音楽システム研究 [21], [22] も出現した。

本研究では遠隔でかつ、不特定多数で演奏可能にすることで演奏の幅を拡張する。また、5G 環境でなくとも問題なく使用できるか検証するために、本研究室の Wi-Fi 環境にて遅延検証を行う。

## 3. 提案手法

本研究では Web 技術に基づいた遠隔共有型ルーパー作成の手法として、関係データベースを中核にした Web アプリケーションを作成した。本システムでは Web MIDI API と Web Audio API を用いることで、MIDI とオーディオの両方を扱うことが可能になるようにした。また、データの通信方法としては WebSocket 通信を用いた。本章では利用した技術であるルーパー・MIDI・オーディオの説明、システムの仕組み、関係データベースの設定に関して述べる。

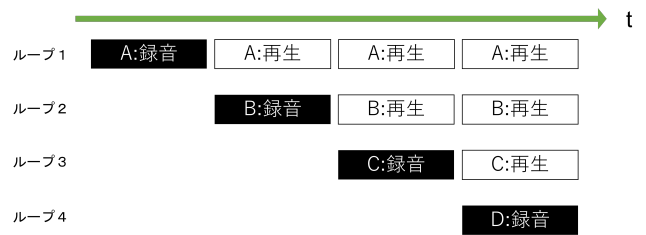


図 1 既存のルーパーの説明図

| チャンネルメッセージ  |                 |            | システムメッセージ |
|-------------|-----------------|------------|-----------|
| 9 0         | 3 C             | 3 C        | 0         |
| 音を鳴らす チャンネル | 音階 (ド: 261.6Hz) | 音の強さ (127) | 時刻 (msec) |
| 8 0         | 3 C             | 0 0        | 5 0 0     |
| 音を止める チャンネル | 音階 (ド: 261.6Hz) | 音の強さ (0)   | 時刻 (msec) |

図 2 MIDI データの例

## 3.1 利用技術

### 3.1.1 ルーパー

ここではルーパーについて説明する。ルーパーとは任意の長さの記録した演奏フレーズを重ねながら、半永久的にループ再生するエフェクターのことをいう。ギター機器や DJ 機器などで使用され、1 人で音楽セッションをすることが可能になる。また、演奏を重ねながら録音するイメージとしては図 1 のようになる。始めに A を録音し、A の録音が終わると、A が再生されると同時に B を録音する。同じように B の録音が終わると A, B が再生され、C を録音する。ルーパーはこの繰り返しを行うことで、多重録音をしながら演奏することが可能になる。

しかしながら、既存のルーパーは 1 人で扱うことが前提とされているため、複数人で同時に使用することができない。また、記録は基本ルーパー本体にされるため、遠隔での使用にも対応できない。本研究では Web 技術を用いることでこの部分を克服した。

### 3.1.2 MIDI

ここでは MIDI の規格について説明する。MIDI は複数の機器の間で、演奏の情報を電子データとして転送・共有するために規格化された世界共通のものである。MIDI データは演奏の情報を伝えるメッセージであり、シンセサイザーなどの音源にどのような演奏をするかを指示するものである。そのため、MIDI データそのものには音がなく、シンセサイザーなどの音源に情報を伝えることで音を鳴らす。また、音そのものをもたないためデータ量が少ない。

図 2 に実際の MIDI データであるドの音が鳴らされた場合、および音を停止させた場合の例を示す。MIDI データは大きく 2 種類に分けられ、チャンネルメッセージとシステムメッセージからなる。チャンネルメッセージは演奏情報そのものであり、システムのなかで個別に機器や音色をコントロールし、16 のチャンネルの一つを主に指定して送信





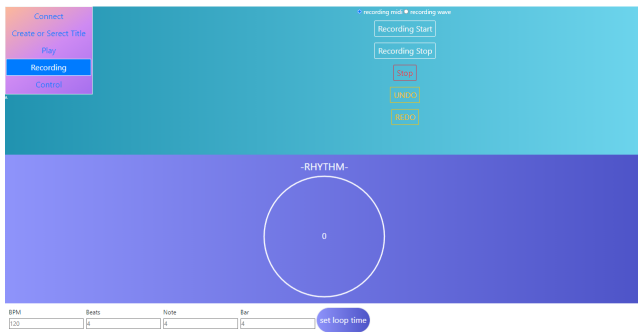


図 6 作成した Web アプリケーション

表 1 MIDI を管理するテーブルの構造

| Field      | Type                     | Null | Key | Default | Extra    |
|------------|--------------------------|------|-----|---------|----------|
| midId      | int                      | NO   | PRI | NULL    | auto_inc |
| personalId | int                      | YES  | MUL | NULL    |          |
| titleId    | int                      | YES  | MUL | NULL    |          |
| noteOnOff  | int                      | YES  |     | NULL    |          |
| noteNumber | int                      | YES  |     | NULL    |          |
| velocity   | int                      | YES  |     | NULL    |          |
| time       | int                      | YES  |     | NULL    |          |
| soundOnOff | enum('on','off','other') | YES  |     | on      |          |

なく、データを取得するかしないかで制御している。さらに、1 ループが残りにどれくらいかを判断する円形のインジケータを付けた。既存のルーパでは弾きながら操作し、初めの録音が 1 ループ分の長さになるが、Web アプリケーションなので、フットペダルなどで制御がしにくい。そのため、演奏前に 1 ループの長さや、リズムを決定できるようにした。

### 3.3 関係データベースの仕組み

ここでは関係データベースの仕組みについて説明する。本システムでは個人の識別を行うテーブル、曲名の識別を行うテーブル、MIDI を管理するテーブル、オーディオを管理するテーブル、ループを管理するテーブルの五つを用意した。そのなかで演奏に直接影響する MIDI を管理するテーブル、オーディオを管理するテーブル、ループを管理するテーブルの三つについて説明する。

表 1 に MIDI を管理するテーブルの構造を示す。MIDI を管理するテーブルの構造としてカラムを八つ用意した。midId をこのテーブルの主キーとし、personalId と titleId は、それぞれ、個人の識別を行うテーブル、曲名の識別を行うテーブルの外部キーとしてある。二つの外部キーにより、個々のデータが誰がどの曲を演奏したデータなのかを判別する。noteOnOff, noteNumber, velocity は音を出すために必要な情報で、それぞれ、どのチャンネルの音を鳴らす・止めるの情報、音階の情報、音の強さの情報を格納する。time には正しい時刻の演奏をするために MIDI データが作成された時刻を格納する。soundOnOff では取得すべきデータなのかを判別する。

表 2 にオーディオを管理するテーブルの構造を示す。

表 2 オーディオを管理するテーブルの構造

| Field         | Type                     | Null | Key | Default | Extra    |
|---------------|--------------------------|------|-----|---------|----------|
| audioBufferId | int                      | NO   | PRI | NULL    | auto_inc |
| personalId    | int                      | YES  | MUL | NULL    |          |
| titleId       | int                      | YES  | MUL | NULL    |          |
| audioBuffer   | mediumblob               | YES  |     | NULL    |          |
| soundOnOff    | enum('on','off','other') | YES  |     | on      |          |

表 3 ループを管理するテーブルの構造

| Field         | Type | Null | Key | Default | Extra    |
|---------------|------|------|-----|---------|----------|
| loopId        | int  | NO   | PRI | NULL    | auto_inc |
| personalId    | int  | YES  | MUL | NULL    |          |
| midId         | int  | YES  | MUL | NULL    |          |
| audioBufferId | int  | YES  | MUL | NULL    |          |
| loopNumber    | int  | YES  |     | NULL    |          |
| loopCount     | int  | YES  |     | NULL    |          |

オーディオを管理するテーブルの構造としてはカラムを五つ用意した。audioBufferId をこのテーブルの主キーとし、personalId と titleId は、MIDI を管理するテーブルと同様に外部キーとしてある。また、soundOnOff も MIDI を管理するテーブルと同様である。audioBuffer には録音して得られた情報をバイナリデータとして格納する。

表 3 にループを管理するテーブルの構造を示す。ループを管理するテーブルの構造としてはカラムを六つ用意した。loopId をこのテーブルの主キーとし、personalId, titleId, midId, audioBufferId はそれぞれ対応するテーブルの外部キーとしてある。loopNumber ではループ数を格納し、loopCount では UNDO により同ループが複数回行われた際の区別としてある。

これらのテーブルを組み合わせる必要のある情報を取り出す。また、これらの制御はサーバから SQL 操作にて行う。

## 4. 遅延計測

本システムではインターネットを介して音楽を扱う。そのため、遅延が発生し、演奏にずれが生じる可能性が高くなる。演奏にずれが生じてしまうと複数人で同時に音楽セッションをすることができなくなる。そこで本システムにおいて遅延がどの程度発生するか計測を行った。以下に計測内容を示す。

計測 1: 異なるコンピュータ間で MIDI データが伝達されるまでの計測

計測 2: 関係データベースから MIDI データを取り出して演奏が再生されるまでの計測

計測 3: 関係データベースからオーディオデータを取り出して演奏が再生されるまでの計測

また、サーバは仮想専用サーバであるさくら VPS\*5 に設置しての計測となる。計測としては有効数字を 2 桁とする。

\*5 <https://vps.sakura.ad.jp/>

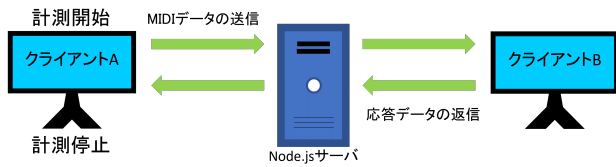


図 7 計測 1 の計測方法

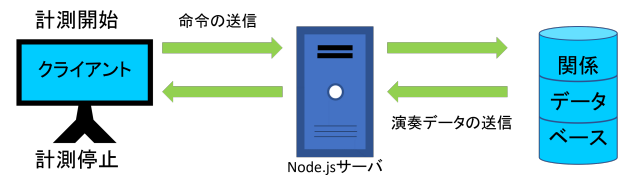


図 9 計測 2 と計測 3 の計測方法

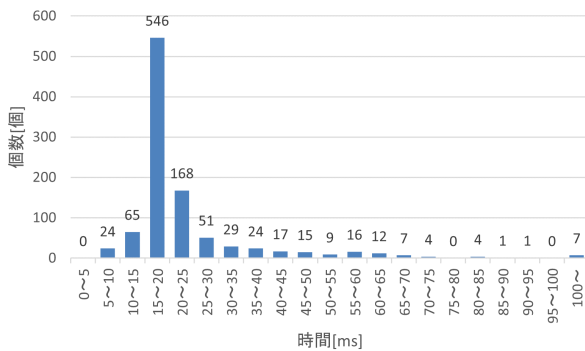


図 8 計測 1 の結果

#### 4.1 計測 1

図 7 に計測 1 の計測方法を示す。本システムにおける WebSocket 通信による遅延を計測するために、本システムの最小のデータである MIDI データを用いて検証する。

まず、クライアント A とクライアント B は別コンピュータで、それぞれで Web アプリケーションを立ち上げる。そして、一方が一つの MIDI データをサーバを経由して送信し、もう一方はデータを受信したらその応答を返す。これにより、2 地点間の往復時間を計測する。実際に計測すべきなのは往路時間のみだが、往復時間のほうが計測する時間が長くなる。したがって、往復時間が十分に短ければ、往路時間も短くなる。そのため、この計測方法で十分な応答速度を有しているか判断することができる。

図 8 に 1,000 回分の MIDI データの遅延計測を行った結果を示す。図 8 が示すように、15 ms から 20 ms の範囲の遅延が 546 個と最も多くなった。計測の平均値は 23.9 ms、中央値は 18.2 ms、標準偏差は 21.6 ms となった。また、最大値は 297.4 ms、最小値は 5.3 ms である。標準偏差や最大値と最小値の差から、遅延にはばらつきがあることがわかる。この原因として、インターネット回線の速度の揺らぎによって引き起こされたと考えられる。また、今回計測した結果として、93.9%が 50 ms 以下となった。さらに、計測した結果は往復の時間であるため、真の結果としては遅延がさらに小さくなる。したがって、安定したインターネット回線のなかでなら、MIDI データにおいてはリアルタイムでの遠隔音楽セッションを行うことは可能であると考えられる。

#### 4.2 計測 2

図 9 に計測 2 と計測 3 の計測方法を示す。関係データ

表 4 計測 2 の結果

| 総データ数  | 平均値 [ms] | 中央値 [ms] | 標準偏差 [ms] | 最大値 [ms] | 最小値 [ms] |
|--------|----------|----------|-----------|----------|----------|
| 10     | 42.6     | 42.7     | 10.3      | 89.2     | 35.8     |
| 100    | 47.6     | 41.6     | 20.9      | 176.4    | 38.6     |
| 1,000  | 60.5     | 47.2     | 63.4      | 501.2    | 44.4     |
| 5,000  | 94.0     | 108.4    | 23.5      | 214.5    | 83.2     |
| 10,000 | 132.7    | 142.8    | 8.17      | 160.9    | 124.2    |

ベースから MIDI データを取り出し、演奏が再生されるまでの計測では、演奏開始の命令を出してから演奏が始まるまでの計測をする。関係データベースから取り出す MIDI データの個数を 10 個、100 個、1,000 個、5,000 個、10,000 個とし、これらを一括で取り出した場合の結果を表 4 に示す。示す値としては 50 回計測した平均値、中央値、標準偏差、最大値、最小値である。

平均値、中央値ベースだとデータ数が増加するにつれて遅延が大きくなることがわかる。二つの値だけを考慮すると、データ数が 10 個、100 個程度の場合ではリアルタイムな演奏に活用できる。しかしながら、実際には数あるデータのなかから抜き出すため、遅延は計測した値より増加すると考える。また、標準偏差や最大値と最小値の差からは、どのデータ数においても遅延にばらつきがあることがわかる。この原因として、計測 1 の場合と同じように、インターネットの回線速度に揺らぎがあったと考えられる。したがって、リアルタイムな演奏に活用するのは困難であると考えられる。

#### 4.3 計測 3

関係データベースからオーディオデータを取り出し、演奏が再生されるまでの計測では、演奏開始の命令を出してから演奏が始まるまでの時間から、音を合成するのにかかる時間を省いた時間を計測をする。関係データベースから取り出すオーディオデータの個数を 1 個から 10 個までを一括で取り出した場合の結果の分布を図 10 に示す。また、表 5 に平均値、中央値、標準偏差、最大値、最小値を示す。データとしては 6 秒でサンプリングレートは 48,000 であるものを用いて、各データ数 50 回の計測を行う。

MIDI データの場合と同じように、平均値、中央値ベースだとデータ数が増加するにつれて遅延が大きくなることがわかる。また、標準偏差のみを MIDI データの場合と比較すると遅延のばらつきは小さい。しかしながら、最大値と最小値の差を考えると遅延のばらつきはある。この原因として、計測 2 の場合と比較してインターネットの回線がより安定していただいただけであると考えられる。また、6 秒 1 個の

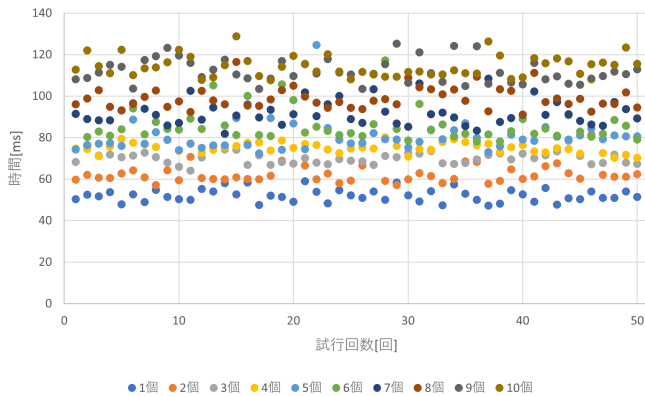


図 10 計測 3 の結果の分布

表 5 計測 3 の結果

| 総データ数 | 平均値 [ms] | 中央値 [ms] | 標準偏差 [ms] | 最大値 [ms] | 最小値 [ms] |
|-------|----------|----------|-----------|----------|----------|
| 1     | 52.1     | 51.9     | 3.0       | 59.0     | 47.2     |
| 2     | 62.5     | 61.2     | 4.3       | 81.4     | 57.1     |
| 3     | 70.2     | 70.3     | 2.8       | 78.4     | 64.1     |
| 4     | 75.8     | 75.1     | 5.1       | 107.8    | 70.3     |
| 5     | 80.4     | 79.0     | 7.6       | 124.7    | 72.5     |
| 6     | 86.0     | 83.5     | 7.7       | 117.2    | 78.1     |
| 7     | 92.0     | 91.0     | 5.8       | 108.5    | 81.9     |
| 8     | 99.0     | 97.8     | 5.1       | 116.4    | 91.1     |
| 9     | 112.0    | 110.9    | 5.7       | 125.3    | 103.5    |
| 10    | 114.4    | 113.6    | 5.0       | 128.9    | 107.6    |

オーディオデータでさえも、50 ms 以下でデータを取得できたのは 26%であった。さらに、実際に演奏されるまでにかかる時間は音を合成する時間も含まれるため、インターネットの回線が安定していたとしても、リアルタイムな演奏に活用するのは困難であると考えられる。

また、60 秒のデータを 1 個取り出すときの計測も行った。その結果、平均値は 227.1 ms、中央値は 226.6 ms、標準偏差は 9.5 ms、最大値は 276.4 ms、最小値は 211.9 ms となった。6 秒のデータ 10 個と比較すると 60 秒のデータのほうが遅延が大きい。したがって、録音する時間が長くなればなるほど遅延が大きくなると考えられる。また、音を合成する時間はデータの長さによって変化し、6 秒のデータ 2 個の合成の平均値は 6.6 ms、中央値は 6.5 ms、標準偏差 0.8 ms であり、60 秒のデータ 2 個の合成の平均は 37.2 ms、中央値は 37.1 ms、標準偏差 4.0 ms である。この合成にかかる時間はマシンパワーに依存する。

さらにクライアント環境を本学に整備されたローカル 5G 環境<sup>\*6</sup>で扱う検証を行った。結果として、6 秒 1 個のデータに対し、平均値は 160.7 ms、中央値は 163.3 ms、標準偏差は 12.7 ms、最大値は 182.2 ms、最小値は 119.2 ms となった。サーバをさくら VPS 上に設置した影響やルータでの処理などが結果に影響を与えたと考えられるため、一概には比較できない。

<sup>\*6</sup> <https://www.metro.tokyo.lg.jp/tosei/hodohappyo/press/2021/10/05/02.html>

#### 4.4 考察

ここでは三つの計測パターンについての考察を行う。計測 1 からは WebSocket 通信における本システムで扱う最小データである 1 個の MIDI データを送信する場合の遅延について検証した。その結果、1 個の MIDI データを送信する程度のデータ量であれば、リアルタイムで遠隔音楽セッションすることができると考えられる。

しかしながら、計測 2 からもわかるように、関係データベースから MIDI データを取得する過程があると、遅延は増大する。10 個や 100 個程度の少ないデータ数でなら、リアルタイムで活用することができる場合があるが、実際の音としてはデータ量の半分となり、5 音や 50 音程度となる。さらに、リバーブやディレイといったエフェクトを MIDI として使用すると実際の音はさらに少なくなる。そのため、実際にリアルタイムで活用することは困難であると考えられる。

また、計測 3 では関係データベースからオーディオデータを取得する場合であるが、6 秒 1 個のオーディオデータでさえも、平均値、中央値は 50 ms を超える。そのため、計測 2 と同じように実際にリアルタイムで活用することは困難であると考えられる。また、6 秒 10 個のデータと 60 秒 1 個のデータを比較した場合、60 秒 1 個のデータのほうが遅延が大きくなるという結果が得られた。これにより、オーディオデータにおいて、分割したものを伝送したほうが良いと考える。しかしながら、分割することによって処理も増加すると考えられるため、さらなる検証が必要である。

遅延に対応するために本システムでは、関係データベースから取得するデータは 1 ループ分遅延させる処理を行った。三つの計測パターンの結果から、1 ループ分遅延させる処理は演奏にずれを起こさないためにも十分に有効であると考えられる。

#### 5. おわりに

本研究では、Web 技術に基づいた遠隔共有型ルーパーを Web アプリケーションとして作成した。このシステムでは、演奏データを関係データベースに集約することでデータの共有を図った。また、遠隔演奏では遅延が発生して演奏にずれが起きる。本システムでは、ルーパーの特徴である演奏がループすることを活かし、関係データベースから取得したデータを 1 ループ分遅らせて同期させることで対処した。さらに、MIDI、オーディオの両方に対応し、同時に演奏を再生させることができた。

しかしながら、本システムは問題点があり、三つの計測からも判断できるように、インターネットの状態に遅延が大きく左右することがわかった。リアルタイム性を向上させるためにもインターネットの状態に左右されない安定したシステムが必要と考える。

また、本研究ではクライアント環境を本学に整備されたローカル 5G 環境で扱う実験も行った。しかしながら、本シ

システムではサーバをさくら VPS 上に設置した影響やルーターでの処理などにより、予測通りの結果が得られなかった。そのため、今後は全システムをローカル 5G 環境に設置し、システムの検証を行うとともに、5G 環境で音楽を扱うことの有効性を検証していく。

## 参考文献

- [1] W3C: Web MIDI API W3C Working Draft 17 March 2015, W3C (online), available from (<https://www.w3.org/TR/webmidi/>) (accessed 2022-05-17).
- [2] W3C: Web Audio API W3C Recommendation, 17 June 2021, W3C (online), available from (<https://www.w3.org/TR/webaudio/>) (accessed 2022-05-17).
- [3] 渋川大樹, 横山昌平: 関係データベースおよび MIDI データを用いた演奏支援システム, *DEIM Forum 2021 D24-3*, pp. 1-7 (2021).
- [4] 西堀佑, 多田幸生, 曽根卓朗: 遅延のある演奏系での遅延の認知に関する実験とその考察, 情報処理学会研究報告音楽情報科学 (MUS), Vol. 2003, No. 127 (2003-MUS-053), pp. 37-42 (2003).
- [5] 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 (MUS), Vol. 1993, No. 109 (1993-MUS-004), pp. 1-8 (1993).
- [6] 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).
- [7] 後藤真孝, 根山亮: 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).
- [8] 大部由香, 米倉達広: 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, 情報処理学会研究報告音楽情報科学 (MUS), Vol. 2003, No. 127 (2003-MUS-053), pp. 43-48 (2003).
- [9] 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, 研究報告電子化知的財産・社会基盤 (EIP), Vol. 2009, No. 23, pp. 1-8 (2009).
- [10] 野原祐一, 辻靖彦: WebRTC を用いた DAW 用遠隔指導支援システムの開発, 情報処理学会研究報告音楽情報科学 (MUS), Vol. 2017, No. 15, pp. 1-6 (2017).
- [11] 大部由香, 吉田雅弘, 米倉達広: 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, 電子情報通信学会技術研究報告. MVE, マルチメディア・仮想環境基礎, Vol. 103, No. 106, pp. 15-18 (2003).
- [12] Obu, Y., Kato, T. and Yonekura, T.: MAS: A Protocol for a Musical Session in a Sound Field where Synchronization between Musical Notes is not guaranteed., *ICMC* (2003).
- [13] Maxwell, J. B. and Eigenfeldt, A.: A music database and query system for recombinant composition., *ISMIR*, pp. 75-80 (2008).
- [14] Maxwell, J. B. and Eigenfeldt, A.: The MusicDB: a music database query system for recombinance-based composition in Max/MSP, *ICMC* (2008).
- [15] 橋田光代, 兼口敦音, 中村栄太, 古屋晋一, 小川容子, 片寄晴弘: ピアニストの演奏解釈を記述した演奏表情データベースの構築, 情報処理学会研究報告音楽情報科学 (MUS), Vol. 2017, No. 23, pp. 1-6 (2017).
- [16] 横山裕, 岩井将行: モバイル端末のブラウザ間で実現する即興的音楽演奏の教育支援アプリケーション, 情報処理学会研究報告音楽情報科学 (MUS), Vol. 2017, No. 22, pp. 1-6 (2017).
- [17] 佐竹紘明, 増井俊之: Re: Piano: 演奏履歴を活用する楽器演奏支援システム, エンタテインメントコンピューティングシンポジウム 2017 論文集, Vol. 2017, pp. 192-195 (2017).
- [18] 金杉季実果, 宮下芳明: 演奏者と楽曲に合わせて変形可能なソフトウェア鍵盤楽器による演奏支援の検討, エンタテインメントコンピューティングシンポジウム 2019 論文集, Vol. 2019, pp. 113-118 (2019).
- [19] 文俊明, 堀内靖雄, 黒岩真吾: 伴奏者の演奏表現を考慮した伴奏システム, 情報処理学会研究報告エンタテインメントコンピューティング (EC), Vol. 2019, No. 22 (2019-EC-51), pp. 1-6 (2019).
- [20] Marchini, M., Pachet, F. and Carré, B.: Rethinking reflexive looper for structured pop music., *NIME*, pp. 139-144 (2017).
- [21] Fu, W.: Random network calculation under the background of 5G network in remote piano music video teaching application, *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-15 (2021).
- [22] Carot, A., Sardis, F., Dohler, M., Saunders, S., Uniyal, N. and Cornock, R.: Creation of a hyper-realistic remote music session with professional musicians and public audiences using 5G commodity hardware, *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, IEEE Computer Society, pp. 1-6 (2020).