

# 遠隔地間におけるインターネットを用いた MIDI 制御

榊原 圭吾<sup>†</sup> 皆月 昭則<sup>†</sup>

釧路公立大学<sup>†</sup>

## 1. はじめに

2019 年に発生した新型コロナウイルス感染症が世界的な猛威を振るうなかで、「オンライン会議」や「オンライン授業」など、それまで対面で行われてきた日常生活が「オンライン」に置き換えられていった。

その一方で、音楽教室やライブなど、音楽の世界では大々的な普及は見られなかった。これには様々な理由が考えられるが、その一つとして「Web 会議システム」の音質を上げることができる。これらのシステムの音質は、通信時の圧縮などの処理やマイクの性能などの要因により低下し、音楽には適しているとは言えない。

そこで音声の代わりに電子楽器の通信規格である MIDI を用いることで、実質的に高い音質で送受信することを可能にし、かつ MIDI データであれば受信後に作曲などへ応用することが可能である。以上のことから本研究では、リアルタイムに MIDI データを送受信できるソフトウェアを開発した。

## 2. 関連研究

後藤ら<sup>[1]</sup>は、MIDI と LAN を融合し MIDI を拡張した独自の通信プロトコルを設計したうえで、コンピュータなどの支援を受けつつ MIDI 機器を用いた合奏が可能な分散協調システムを提案した。これは CG などへの拡張性に優れ、鍵盤が弾けない人へのアプローチも考慮されているが、使用している機器や通信規格などが旧式化しており、現在の環境では使用が困難である。

本研究では現在主流になっているインターネットや Web の技術を用いた。一方で MIDI 制御を用いた音楽作業への支援を主眼にするため、CG などへの拡張は行わず、独自のプロトコルは設計せず MIDI で対応した。

## 3. MIDI 送受信システム

### 3-1. システムの概要

本研究では JavaScript を開発言語とし、MIDI をリアルタイムに送受信できるソフトウェアを開発した。このソフトウェアでは MIDI デバイス

の呼び出しや制御に Web MIDI API<sup>[2]</sup>、通信に SkyWay<sup>[3]</sup>を用いた。なおシステム開発の過程において、SkyWay 公式の開発ドキュメント<sup>[4]</sup>および JavaScript SDK リファレンス<sup>[5]</sup>を参考に、Web MIDI API を組み合わせた。

ドロップダウンで MIDI Input と MIDI Output を選択し、それぞれ指定されたデバイスを接続する。その後に任意の Room Name を入力し Join ボタンを押すと ID が発行される。通信する相手も同じ Room Name を入力し Join ボタンを押すと双方に相手の ID と通信するメディアが表示され、それを指定することで通信される。



図1 ソフトウェア画面

### 3-2. SkyWay

SkyWay とは NTT コミュニケーションズ株式会社が提供しているソフトウェア開発キット (SDK) である。通信に WebRTC<sup>[6]</sup>の技術を用い、アプリケーションに対しビデオ通話や音声通話、データ通信を組み込むことができる。WebRTC とは Google によってオープンソース化された API であるが、これを容易に取り扱うことができることから SkyWay を採用することにした。なお 2023 年 1 月 31 日より新しいバージョンが公開されており、本研究ではこちらを採用した。

SkyWay で通信できるメディア (Stream) は、動画 (VideoStream)、音声 (AudioStream)、データ (DataStream) の 3 種類である。DataStream では様々なデータを取り扱うことができ、本研究ではこの Stream で MIDI データを送受信した。

SkyWay では、コンピュータ同士を直接接続する P2P と、サーバを介してコンピュータを接続する SFU の 2 種類が選択できる。本研究では遅延時間を少なくするため、P2P を採用した。

MIDI control using the Internet between remote locations

<sup>†</sup> Keigo Sakakibara, <sup>†</sup> Akinori Minaduki

<sup>†</sup> Kushiro Public University Minaduki lab

### 3-3. システムの数値的目標

本研究が開発したシステムでは、送受信に係る遅延時間の目標値 0.5 秒以下を達成した。これは多くの DAW(DTM で用いるソフトウェア)の初期値が BPM120 前後であることから、その 1 拍以下とするものである。

## 4. 検証

### 4-1. 検証方法

このシステムを用いて MIDI データの送受信を行い、発生した遅延を計測した。使用機材は表 1 の通りである。

電子楽器	送信側	YAMAHA DX7
	受信側	YAMAHA MODX6
PC	送信側	dynabook X30L-J (Win10Home) CPU: Intel Core i5-1135G7 メモリ: 8GB
	受信側	Lenovo ThinkPad L15 (Win10Pro) CPU: Intel Core i5-10210U メモリ: 12GB
MIDI-USBケーブル	送信側/受信側	Roland UM-ONE mk2
オーディオミキサー	-	YAMAHA AG06
音声編集ソフトウェア	-	Audacity 3.0.2

表 1 計測における使用機材

同一ネットワーク内(Wi-Fi)において、実際に電子楽器を演奏した。送信側の電子楽器から送出された MIDI データを、本研究で開発したソフトウェアを経由して受信側の電子楽器へ送信した。このとき送信側・受信側双方から出力される音声信号をオーディオミキサーにてミキシングし、録音した。その後、送信側と受信側のタイミングを音声編集ソフトウェアにて計測した。計測の仕組みは図 2 の通りである。

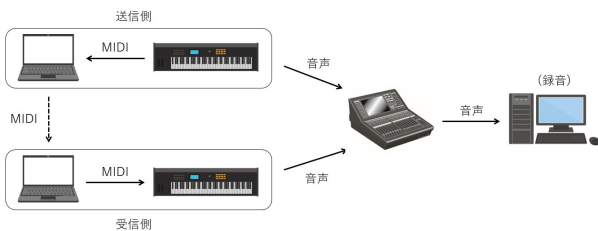


図 2 計測の仕組み

### 4-2. 結果

MIDI の送受信を 30 回試行した結果、図 3 のようになった。

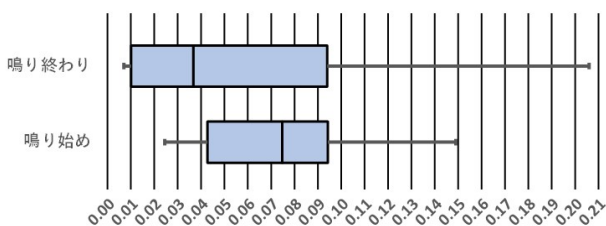


図 3 遅延時間のグラフ

鳴り始め・鳴り終わりの遅延時間はいずれも目標値の 0.5 秒を下回った。その一方で鳴り始めの最大値が 0.1490 秒で最小値が 0.0245 秒、鳴り終わりの最大値が 0.2060 秒で最小値が 0.0070 秒と、遅延時間に差があることが判明した。また鳴り始めと鳴り終わりで遅延時間に差があることが多かった。したがって送信側と受信側で発音時間が異なる場合があることが露見した。

## 5. おわりに

本研究では MIDI をインターネット経由で遠隔送受信することに成功した。また遅延時間についても目標値 0.5 秒を下回ることができた。

その一方で本研究には課題もある。P2P の特性上、接続するコンピュータが増えるとデータ処理に時間がかかる。本研究では 1 対 1 のみの検証を行ったが、大人数での送受信も想定される。さらに 3-2 節でも述べたように、遅延時間に振れがあること、送信側と受信側で発音時間が異なる場合が存在することが検証により明らかになった。今後多くの電子楽器や通信環境などでの検証を続け、更なる改良を加えていく。

## 参考文献

- [1] 後藤真孝, 橋本裕司 “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 音楽情報科学研究会 研究報告 1993
- [2] mdn web docs\_ Web MIDI API [https://developer.mozilla.org/ja/docs/Web/API/Web\\_MIDI\\_API](https://developer.mozilla.org/ja/docs/Web/API/Web_MIDI_API) 2023. 12. 21
- [3] NTT コミュニケーションズ「ニューノーマルにおける新しいビジネスの形「リモート〇〇」を支える「SkyWay」とは」 <https://www.ntt.com/bizon/newnormal-skyway.html> 2023. 12. 21
- [4] SkyWay 開発ドキュメント JavaScript SDK <https://skyway.ntt.com/ja/docs/user-guide/javascript-sdk/> 2023. 12. 21
- [5] SkyWay JavaScript SDK リファレンス <https://javascript-sdk.api-reference.skyway.ntt.com/core/index.html> 2023. 12. 21
- [6] WebRTC ホーム <https://webrtc.org/?hl=ja> 2023. 12. 21