Vol.2026-HCI-217 No.10
2026/3/9

Seqtrument：機械学習を応用した自然な楽器音の逐次的生成手段

小平卓実[†1]　西本一志[†1]

概要：コンピュータでアコースティック楽器音を用いた楽曲を制作する際，現実の楽器を再現したソフトウェア音源が用いられることがある．しかし既存の音源を用いた場合，各種のパラメタをかなり緻密に調整しても，実際に人間が演奏した音に比べて不自然な音色になることが多い．これに対して，近年 AI を使った解決が試みられているが，既存研究の多くは楽曲全体や一区切りの旋律など，まとまった範囲で処理しているため，楽曲の先頭から逐次的に楽譜を読み込む一般的な音楽制作ソフト（DAW）上で直接動かすことが難しい．そこで本研究では，AI による自然な音色を，既存のソフトウェア音源と同様に DAW 上で直接生成できるように，楽譜情報から逐次的に楽器音を生成する機械学習モデル Seqtrument を提案する．本稿では，システムの構成とこれを用いて生成したバイオリン音の品質についての実験結果を報告する．

キーワード：音楽創作支援，楽器音生成，逐次処理，機械学習

Seqtrument: A Method for Sequentially Generating Natural Instrument Sounds Using Machine Learning

TAKUMI KOBIRA[†1]　KAZUSHI NISHIMOTO[†1]

Abstract: When creating music using acoustic instruments sounds on a computer, software sound sources that reproduce real instruments sounds are often used. However, when using existing sound sources, even with very precise adjustments to various parameters, the resulting timbre often becomes unnatural compared to sounds actually produced by humans. In recent years, attempts have been made to address this issue using AI, but most existing research processes a single section of music, such as an entire piece or a single phrase, making it difficult to run directly on standard music production software (DAW), which sequentially read sheet music from the beginning of the piece. Therefore, in this study, we propose “Seqtrument”, a machine learning model that sequentially generates instrument sounds from sheet music information, so that natural-sounding AI timbres can be generated directly on a DAW, just like existing software sound sources. This paper reports on the system configuration and experimental results on the quality of violin sounds generated using this system.

Keywords: Music creation support, instrumental sound generation, sequential process, machine learning

1. はじめに

近年，コンピュータの普及や安価なソフトウェア音源の登場により，誰でも手軽に作曲を行えるようになった．ソフトウェア音源が楽器の音色を生成する方法として，楽器の 1 音 1 音を全て録音したサンプラー方式や，楽器の物理的特性を再現する物理モデリング方式といったものが存在する．しかし，これらの手法で生成された楽器音をそのまま単純に使用すると，実際に演奏された音に比べて不自然な音色になることが多い. 自然な音色に近づけるためには，音量や音高の変化といった様々なパラメタを楽曲の進行に合わせて調整する必要がある．しかしながらこのパラメタ調整は非常に複雑かつ微妙であり，手間がかかる困難な作業となるため，誰でも手軽に実行できるというわけにはいかない.

そこでこの課題を解決するために，AI の活用が試みられている．人間の歌声を対象としたものとして，法野らが発表した Sinsy [1]では DNN をベースとすることで，従来の HMM ベースのもの[2]よりも自然な歌声が合成できることを示した．また，Liu らが発表した DiffSinger [3]では拡散生成モデルによってメルスペクトログラムを生成することで，敵対的生成を用いた手法[4]より高い品質が得られることを示した．楽器を対象としたものでは，Hao らが発表した ViolinDiff [5]等があり，拡散ベースのモデルにピッチベンド情報を明示的に組み込むことで，他の拡散ベースの手法[6]より生成品質が上回った．Deep Performer [7]では Transformer ベースのモデルを用い，ピアノ等の多声音を含む演奏音を高品質に生成できることを示した.

これらの既存手法は，楽曲全体や一区切りの旋律など，まとまった範囲を対象としてバッチ的に処理するものがほとんどである．一方，コンピュータ上で作曲を行う際に現在一般的に用いられている音楽制作ソフト（DAW: Digital Audio Workstation）では，楽曲の先頭から逐次的に楽譜や波形を読み込みながら音色の合成やエフェクト処理等を行う. それゆえ，既存の AI による音色合成手法を直接 DAW 上で動かすことができない．そのため，音楽制作者は DAW と DAW 以外の複数のソフトウェアとの間を行き来しながら音楽制作を行う必要があり，作業が煩雑になる.

†1 北陸先端科学技術大学院大学 先端科学技術研究科
Graduate School of Advanced Science and Technology, Japan Advanced Institute of Science and Technology

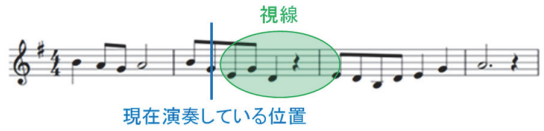

図 1 楽器演奏者の演奏位置と注視箇所の関係
Fig. 1 Relationship between instrumentalists' playing position and point of focus

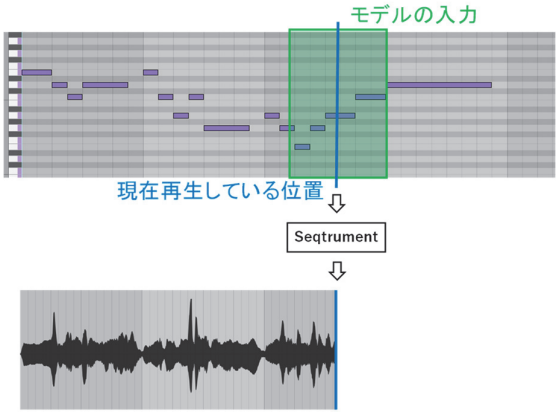

図 2 Seqtrument の動作イメージ．DAW の再生に合わせて上の楽譜情報から下の波形を逐次的に生成する．

Fig. 2 System operation overview of Seqtrument. Synchronized with DAW playback, it sequentially generates the waveform below from the musical score information above.

そこで本研究では，AI による自然な楽器音を従来のソフトウェア音源と同様に DAW 上で直接生成させる手段を構築することにより，作曲者のメロディ探索を支援することを目指している．本稿では，この手段の中核である逐次的に波形を生成する機械学習モデルについて述べ，生成した音色の品質からモデルの有用性を検証する．なお本稿で示す実験は，知識科学倫理審査会議（承認コード：KSEC-F20250111001）を得て実施している．

2. Seqtrument

2.1 システムの概要

多くの楽器演奏者は，図 1 のように実際に演奏している位置よりもわずかに先の楽譜を参照しながら演奏していることが知られている．また，擦弦楽器や管楽器のような単一音の持続中に音量や音高を連続的に変化させられる楽器では，音符の長さや前後のフレーズから，音量の変化やビブラートの有無などをリアルタイムに制御している．このような人間による演奏制御を DAW の逐次的な処理方式の中で再現するための生成モデル Seqtrument を提案する．図 2 に Seqtrument の動作イメージを示す．図 2 の上に示すのは，一般的な DAW における楽譜表示形式（ピアノロール）であり，縦軸が音の高さ，横軸が時間を表している．

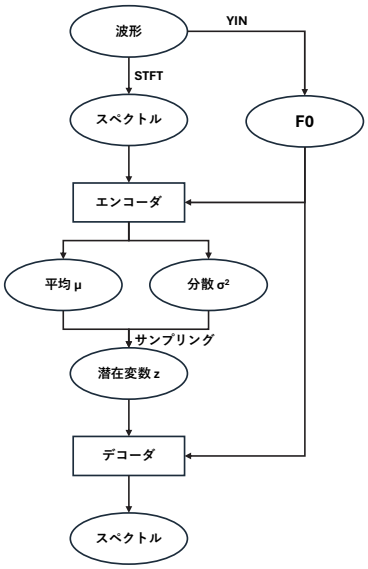

図 3 VAE の構成
Fig. 3 Configuration of VAE

表 1：短時間フーリエ変換の各パラメタ
Table 1. Parameters for short-time Fourier transformation

サンプリング周波数	16kHz
FFT サイズ	2048
フレーム間隔	512
窓関数	hann
周波数軸の次元数	1025

Seqtrument では，DAW の再生位置を基準に前後数秒の楽譜情報を入力として、逐次的に波形（図 2 の下）を生成する．

2.2 F0 を条件とした VAE によるスペクトルの圧縮

図 3 に VAE（Variational Autoencoder）の構成を示す．VAE[8]は，入力層と出力層に同じデータを用い，入力データの次元を下げたあとに再び戻すことで次元の圧縮等を行うオートエンコーダの一種であり，入力データを潜在空間へ確率的に圧縮することで，新しいデータの生成を可能にしたものである．本研究ではまず，表 1 のパラメタを用いた短時間フーリエ変換によって波形データから変換した時間フレームごとの 1025 次元のスペクトルを圧縮するために VAE を用いた．さらに図 3 に示すように，音高情報である基本周波数（F0）をエンコーダとデコーダ両方の入力に明示的に与える，条件付き VAE を構成した．F0 の推定には，時間軸方向にずらした波形と元の波形との差分が小さくなる距離から F0 を推定する自己相関法から誤検出を低減するように改良された Yin アルゴリズム[9]を用いた．

エンコーダは，各時間フレームにおけるスペクトルと F0 を入力とし，潜在変数 z の平均 μ と分散 σ^2 を出力する．デコーダは，平均 μ と分散 σ^2 から式（1）によってサンプリ

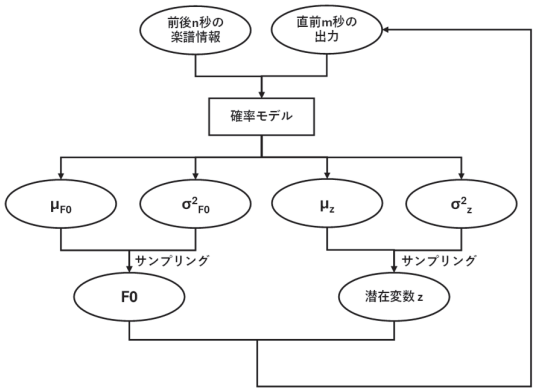

図 4 確率モデルに対する入出力の構成
Fig. 4 Input-output configuration for the probabilistic model

ングした潜在変数 z と F0 を入力とし，元のスペクトルを出力する．

$$\boldsymbol{z = \mu + \sigma \cdot \epsilon,\quad \epsilon \sim N(0,1)} \tag{1}$$

損失関数は，式（2）に示すように，再構成誤差と，潜在変数分布と事前分布との間の KL ダイバージェンス（KL 項）の和として定義した．

$$\boldsymbol{loss = (x - \hat{x})^2 + \beta \frac{1}{2}\left(\mu^2 + \sigma^2 - \log \sigma^2 - 1\right)} \tag{2}$$

ここで，x は教師データ，$\hat{x}$はデコーダの出力，μ，σ^2 はそれぞれエンコーダが出力する潜在変数における平均, 分散，β は KL 項の重みを表す．

最適化アルゴリズムには adam を用い, 学習率は 0.001 に設定，学習は 1000 エポック実行し，バッチサイズは 64 とした．学習を安定させるため KL アニーリングを導入し，KL 項の重み β を最初の 200 エポックかけて 0 から 0.01 まで線形に増加させた．潜在変数が過度に事前分布へ収束することを防ぐため Free-bits を導入し, KL 項が 0.5 を下回らないよう制約を設けた．潜在変数の次元は大きくするほど再構成品質は向上するが，潜在変数側も F0 に関する情報を持つようになり，条件 F0 との分離性が低下する．これらのバランスから，潜在変数の次元数を 8 次元に設定した．

2.3 楽譜情報に基づく F0，潜在変数の確率的予測

図 4 に確率モデルに対する入出力の構成を示す．生成箇所の前後 n 秒の楽譜情報（楽譜上の音高データ）と直前 m 秒の出力をモデルの入力とし, 楽譜情報（楽譜上の音高データ）と実際の演奏データから推定した F0 の差分，及び 2.2 のエンコーダによって圧縮した潜在変数 z の平均を出力する．楽譜情報は MIDI から各時間フレームにおける以下の形式に変換する．

- ピッチ：楽譜上の音の高さ（Hz）を対数化した実数値
- ノート on/off：ノートの on/off に対応したバイナリ値

例として図 5 は図 6 の楽譜を変換したものである．

損失関数には，式（3）に示す Gaussian Negative log likelihood Loss を用いることで，生成箇所における F0，z の確率分布を学習する．

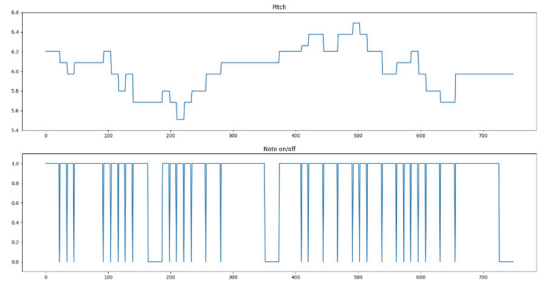

図 5 確率モデルに入力される楽譜情報の形式．上がピッチ，下がノート on/off
Fig. 5 Format of musical score information input to the probabilistic model. Top row: pitch, bottom row: note on/off.

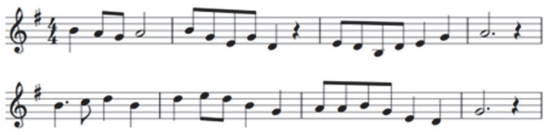

図 6 童謡「七つの子」の楽譜の一部
Fig. 6 Part of the sheet music of the children's song “Nanatsu-no-Ko”

$$\boldsymbol{loss = \frac{1}{2}\left(\log \sigma^2 + \frac{(y - \mu)^2}{\sigma^2}\right)} \tag{3}$$

ここで，y は教師データ，μ，σ^2 はそれぞれモデルが出力するガウス分布における平均，分散を表す．

学習は 200 エポック実行し，バッチサイズは 64 とした．VAE 同様，最適化アルゴリズムには adam を用い，学習率は 0.001 に設定した．

2.4 システムの全体構成

システムの全体構成を図 7 に示す. 推論時は 2.2 のデコーダと 2.3 の確率モデルを図 7 のように繋げて使用する．確率モデルによって，前後の楽譜情報，および直前の出力から F0，潜在変数 z の平均 μ，分散 σ2 を出力し，式（4）によってサンプリングする．

$$\boldsymbol{F0 \sim N(\mu_{F0}, \sigma^2_{F0})}$$
$$\boldsymbol{z \sim N(\mu_z, \sigma^2_z)} \tag{1}$$

サンプリングで得られた，F0，潜在変数 z をデコーダに入力し，出力されたスペクトルから逆短時間フーリエ変換（ISTFT）によって波形へ変換する．

3. 予備実験

モデルの妥当性と改良の方向性を確かめるため，VAE が学習する対象を振幅スペクトルのみに限定し，位相情報は明示的に含まれないモデルで予備実験を実施した．生成された振幅スペクトルから波形を復元する際に Griffin-Lim

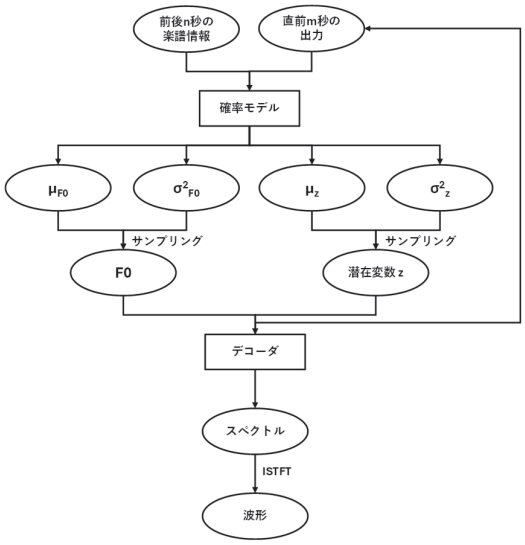

図 7 システムの全体構成

Fig. 7 The entire system setup

アルゴリズム[10]を用いて位相の推定を行った．なお，Griffin-Lim アルゴリズムではスペクトル全体に対して反復的な処理が必要となるため，この予備実験段階では全てが逐次的ではなく，スペクトルから波形に復元する際に一括的な処理が含まれている．

3.1 データセット

筆者らが所属する大学院大学の学生 1 名による「七つの子」や「蛍の光」などの童謡 10 曲分のバイオリン演奏を各 3 セット（約 10 分×3 セット）録音し，ピッチ補正ソフトによって意図しないピッチやタイミングのずれに対して修正を行ったものを元音源とした．さらにこれらの元音源を，DAW のトランスポーズ機能を用いて半音上げたデータを追加し，これらすべてをデータセットとした．このうちの 9 割を学習データとして用い，残りの 1 割をテストデータとした．なお，元音源のサンプリングは 16bit/44.1kHz で行ったが，実験ではこれをモノラル 16bit/16kHz にダウンサンプリングして使用した．

3.2 評価方法

筆者らが所属する大学院大学の学生 7 人と学外の作曲経験者 1 人を被験者として雇用した．各被験者に以下に示す 5 種類の生成演奏音を聴き比べてもらった．

- Ground Truth：ピッチとタイミングの修正を施した実際のバイオリン演奏データ（元音源）
- VAE（予備実験版）：2.2 で説明した VAE のみを用いて，Ground Truth を再構成した生成演奏音
- Seqtrument（予備実験版）：提案システム（ただし位相情報を含まないモデル）による生成演奏音
- Sampling：DAW 上で既存のサンプリング音源を用いた生成演奏音

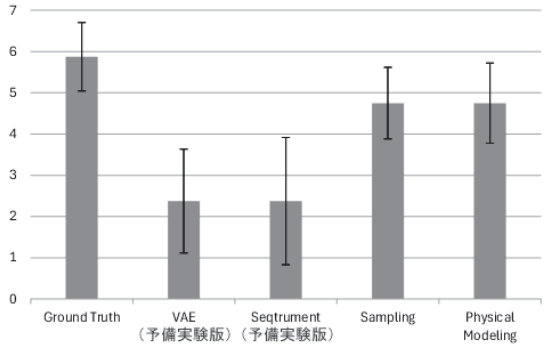

図 8 自然さに関する評価結果

Fig. 8 Evaluation results about naturalness

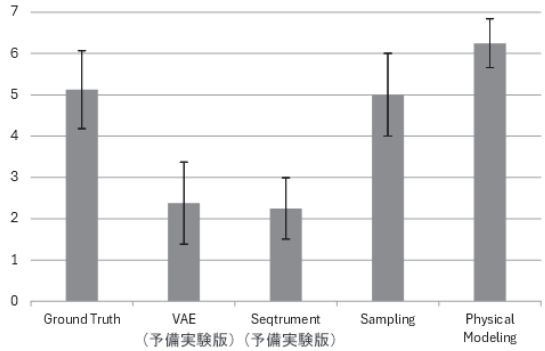

図 9 音質に関する評価結果

Fig. 9 Evaluation results about quality of sounds

- Physical Modeling：DAW 上で既存の物理モデリング音源を用いた生成演奏音

その後，以下の 2 つの項目について 7 段階の評価アンケートを実施した．

- 自然さ：人間が演奏しているように感じますか？
- 音質：ノイズやひずみがなくきれいに聴こえますか？

3.3 結果・考察

図 8，図 9 はそれぞれ自然さ，音質について 8 人の回答結果をまとめたものである．予備実験モデルによる生成演奏音は，自然さ，音質の両面において既存のソフトウェア音源より低い評価となった．一方で，VAE のみを用いて再構成した生成演奏音と Seqtrument による生成演奏音との評価の間に有意差が認められなかったことから，VAE がボトルネックになっていると考えられる．被験者からは，両生成演奏音に対して「音が途切れているように感じる」といった意見が多く得られた．前述の通り，予備実験段階での VAE には位相情報を与えておらず，各フレームは明示的に時系列間の情報を持たない．そのため，復元する際に時間軸方向の再構成精度が十分に確保されなかったことで音の途切れが生じ，各評価の低下につながった可能性がある．

そこで，次章では VAE の精度改善に重点を置き，位相情報を含めたスペクトルの圧縮を試みる．

4. 本実験

4.1 VAE による位相情報を含めたスペクトルの圧縮

予備実験結果を踏まえ，VAE の学習データに位相情報を組み込む．STFT で得られた複素スペクトルから，振幅スペクトルと偏角 θ を求める．この際，θ は 2π 周期のため，0 と 2π は等価な意味を持つが，MSE 損失では$(2\pi)^2$となってしまう．これを解決するため，事前に θ から「sin(θ), cos(θ)」に変換し，F0 を条件として，振幅スペクトルを対数化した対数振幅スペクトル A，sin(θ)，cos(θ)を VAE によって圧縮した．各時間フレームにおける再構成損失（Rec Loss）は以下の式(5)で計算する．位相誤差に振幅で重み付けすることで，聴覚上重要な周波数成分の位相誤差の影響が大きくなるように設計した．

$$\boldsymbol{Rec\ Loss_t} = \frac{\mathbf{1}}{\boldsymbol{K}}\sum_{\boldsymbol{k=0}}^{\boldsymbol{K}} \left|\boldsymbol{A_k} - \widehat{\boldsymbol{A}}_{\boldsymbol{k}}\right| + \boldsymbol{A_i}\left(\left(\sin\boldsymbol{\theta_k} - \sin\widehat{\boldsymbol{\theta}}_{\boldsymbol{k}}\right)^{\mathbf{2}} + \left(\cos\boldsymbol{\theta_k} - \cos\widehat{\boldsymbol{\theta}}_{\boldsymbol{k}}\right)^{\mathbf{2}}\right) \quad (5)$$

ここで，K は FFT ビン数であり，A，θは教師データの対数振幅および偏角，$\hat{A}$，$\hat{\theta}$はモデルの出力を表す．これに式(2)で定義した KL 項を加算したものを損失として定義した．圧縮した潜在変数 z を予測する確率モデルに対する入出力は予備実験と同様である．

4.2 逐次的生成音に対する評価

学習した各モデルを図 7 のように結合し逐次的に生成された音と予備実験で生成した音を作曲経験のある本学学生 1 名に聴き比べてもらい，自然さ，音質について比較を行った．結果として，音質は同様だが本来鳴るべき音とは異なる音高の音が時折紛れ込むようになったため自然さは下がったという評価が得られた．図 10 は，データ拡張分を含む全楽曲中の音符の度数分布である．横軸は MIDI Note Number であり，音高に対応する．調査の結果，使用頻度の少ない音高において異音の混入が顕著であることがわかった．このことから，この問題の原因は VAE によって圧縮する対象が増えたことによって，データセットが少なかったことや偏りがあったことに起因する問題が顕在化したためだと考えられる．また，本システムで生成された音が多重奏を使う場面で使えそうだという感想が得られた．このようになる要因として，楽譜上で同一の音高でもピッチや位相が微妙に異なる音を同一の音として学習した結果として，出力される音にも微妙にピッチや位相がずれた音が重なり，あたかも複数の楽器で演奏した音が重なっているかのように聞こえたためだと推察される．

4.3 リアルタイム生成によるモデルの有用性検証

最終的に本モデルを搭載したソフトウェアを使用する際は，多くの DAW に標準で搭載されているトラック全体の発音タイミングを調節する機能であるトラックディレイに負の値を設定することで，楽譜の先読みを行うことを想定している．そのため，使用状況としてはピアノロール上にマウス等で音符を配置していくステップ入力を前提としている．しかし，DAW 上で使用可能なソフトウェアとして実装する前に，作曲者に自由に即興的な演奏をしてもらう実験を実施した．これは，先述の実験のように演奏される楽曲の内容があらかじめわかっている状況では見つかりづらいような課題やあるいは有用性などを洗い出すことを目的としている．そこで MIDI キーボードからの演奏データをリアルタイムに入力して音を合成し，スピーカから再生するシステムを構築した．この際，モデルの入力に再生箇所より先の楽譜情報が必要なため，約 64～96ms の遅延が生じている．

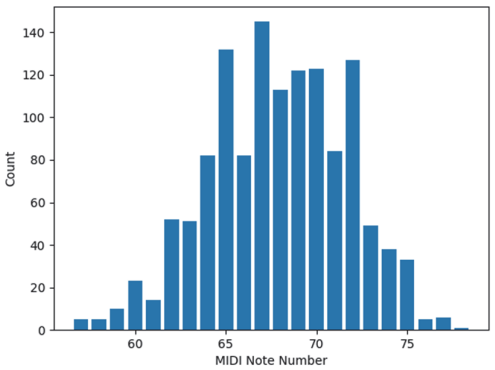

図 10 全楽曲中の音符分布（データ拡張分含む）
Fig. 10 Distribution of notes across all songs (including data extensions)

4.2 と同一の学生 1 名に構築したシステムを使用してもらい，音質・使用感について自由記述形式でのアンケートを実施した．以下はそれらの回答をまとめたものである．

- 音質について
  - 予備実験段階と比較して音質が向上している
  - 異音が混入する
  - 音高により音の大きさが異なる
  - ビブラートの振幅に揺らぎがある
- 使用感について
  - 入力タイミングによる音の欠落がある
  - トリル，和音，ベロシティ調整は不可
  - ラグは少なめ

4.2 の実験とは異なり，予備実験段階より音質が向上したと感じられたことについて，以下の理由が考えられる：

- ✓ 4.2 の実験ではイヤホンを用いて比較を行ったが，今回の実験ではスピーカを用いたこと．
- ✓ 4.2 の実験で用いた既存の童謡について，被験者が楽曲に対するイメージをあらかじめ保有していたため，次に期待する音と実際に出力される音とを比較しながら聴くことが容易であった．これに対し今回の実験では，メロディを手探りしながら即興的に演奏してい

るため具体的な音のイメージが定まっておらず, そのような比較が困難であったため, 音質の劣化が気にならなくなった可能性があったこと.

一方，確率的に生成する過程で入力タイミングによる音の欠落やビブラートの振幅の揺らぎなど，実際の演奏に比べると不自然だと感じられる部分も多かった．また，トリルができないことや音高により大きさが異なるといったものの原因として，学習に使用した楽曲が童謡で構成されておりトリルが表現できるほどの極めて短い音がデータセット中に登場しないことや，楽譜に強弱記号が記載されておらず，元音源の演奏を行ったバイオリン演奏者に対して音量についての明示的な指示を与えていなかったことなどから，データセットに依存した問題だと考えられる．なお，楽譜情報として単音のピッチとノート on/off のみからスペクトルを生成する現在のモデルの仕様上，和音やベロシティの調整は不可なため，これらについては今後の展望とする．

5. おわりに

本研究では DAW 上での使用を想定した逐次的楽器音生成モデルである Seqtrument を提案した．前後の楽譜情報と過去の出力から現時刻におけるスペクトルを予測することで, 逐次的な楽器音の生成が可能であることが示唆された．一方，既存手法と比べて優れた生成品質を達成できなかった．その要因として，学習データが 30 分程度と他の機械学習モデルと比較して少なかった点や，異音等の想定していない音が発生した際の後処理など，アルゴリズムにも改良の余地があったことが考えられる．今後はさらに大規模なデータセットを基にモデルの改良を試み，DAW 上で直接操作可能なソフトウェアとして実装し，創造活動に与える影響を含めた評価実験を検討している．

謝辞 実験にご協力いただいた皆さんに厚くお礼申し上げます．

参考文献

[1] Yukiya Hono, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda：Sinsy: A Deep Neural Network-Based Singing Voice Synthesis System, IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol.29, pp.2803-2815, 2021.
[2] Keijiro Saino, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda : An HMM-based singing voice synthesis system, Ninth International Conference on Spoken Language Processing, pp.1141-1144, 2006
[3] Jinglin Liu, Chengxi Li, Yi Ren, Feiyang Chen, and Zhou Zhao：DiffSinger: Singing Voice Synthesis via Shallow Diffusion Mechanism, Proceedings of the AAAI Conference on Artificial Intelligence, 2022.
[4] Jie Wu, and Jian Luan : Adversarially Trained Multi-Singer Sequence-To-Sequence Singing Synthesizer, INTERSPEECH 2020, pp.1296-1300, 2020
[5] Daewoong Kim, Hao-Wen Dong, and Dasaem Jeong : ViolinDiff: Enhancing Expressive Violin Synthesis with Pitch Bend Conditioning, ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing （ICASSP）, 2025.
[6] Ben Maman, Johannes Zeitler, Meinard Müller, and Amit H. Bermano：Performance Conditioning for Diffusion-Based Multi-Instrument Music Synthesis, ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2024
[7] Hao-Wen Dong, Cong Zhou, Taylor Berg-Kirkpatrick, and Julian McAuley：Deep Performer: Score-to-Audio Music Performance Synthesis, ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing （ICASSP）, 2022.
[8] Diederik P Kingma, and Max Welling：Auto-Encoding Variational Bayes, International Conference on Learning Representations (ICLR), 2014.
[9] Alain de Cheveigné, and Hideki Kawahara : YIN, a fundamental frequency estimator for speech and music, The Journal of the Acoustical Society of America, Vol.111, pp.1917-1930, 2002.
[10] D. Griffin, and Jae Lim : Signal estimation from modified short-time Fourier transform, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol.32, Issue2, pp.236-243, 1984