Vol.2023-MUS-137 No.44
Vol.2023-SLP-147 No.44
2023/6/24

単旋律撥弦楽器の奏法変化と音韻変化の音響的類似に着目した統計的楽器音合成

小口 純矢[1,a)]　森勢 将雅[1,b)]

概要：エレクトリック・ベースのような撥弦楽器において，アタック区間ではピッキングノイズに由来する非周期成分が，サスティン区間では弦振動に由来する周期成分がそれぞれ支配的である．これは音声における無声子音と有声母音で見られる現象に類似しており，音声の音素表現を撥弦楽器の奏法表現へと応用できる可能性を示唆する．そこで本研究はエレクトリック・ベースの楽器音を対象に，アタック区間を子音，サスティンからディケイまでの区間を母音のように扱うことで，楽器音音響信号に対応する奏法のラベルを設計し，統計的楽器音合成システムを構築した結果を報告する．

1. はじめに

機械学習や信号処理技術の発展によりあらゆる音を高品質に合成することができるようになった．Deep Neural Network（DNN）ベースの音声合成手法は人間と同等の品質を達成し [1]，合成の対象は話し声だけでなく歌声 [2]，そして楽器音 [3] と多様なものになっている．DNN に基づく音響信号の合成は，通常入力ラベル列とそれに対応する音響信号を予測する回帰問題として解かれる．例えば，テキスト音声合成は入力にテキストあるいはその言語情報を入力し，読み上げ音声を出力するよう学習される．その意味で，楽器音は楽譜によって表記される音声言語とみなすことができ，音声合成の枠組みを利用することができる．このアプローチは，ラベルによって音響信号の特徴を条件づけることで，合成音に対する高い制御性を期待できる [4]．

合成器の制御性は入力として与える情報に依存する．表現力豊かな楽器音合成器を構築する上では，Musical Instrument Digital Interface (MIDI) のような楽譜情報だけでなく奏法の変化に関する詳細な情報を含めることが不可欠である．しかし，MIDI の仕様上で奏法を表現するフォーマットは定義されていない [5]．いくつかのソフトウェア音源では音域外のノートを奏法を切り替えるキースイッチとして導入しているが，デベロッパーによって扱える奏法の種類やその割り当ては異なる [6,7]．また，ノートごとに

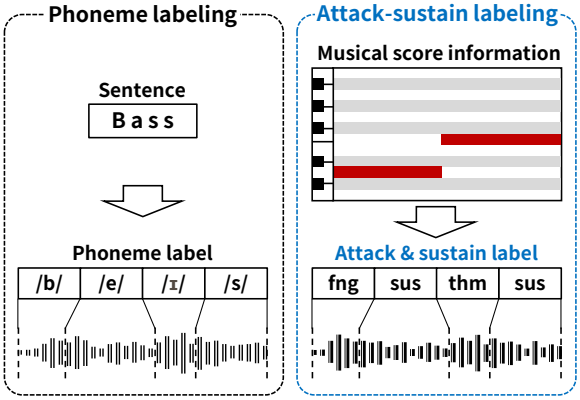

図 1　提案するアタック–サスティン表現の概要図．音声における子音と母音のように，1 つのノートをアタック・サスティン区間に分割し，それぞれに奏法を割り当てる．例えば，フィンガー・ピッキング（fng）で 1 音目，サムピング（thm）で 2 音目を演奏した場合の表現は，いずれもサスティン区間は通常の弦振動（sus）であるため，“fng-sus-thm-sus” と変換される

奏法が異なる場合や 1 つノートに対して複数の奏法が複合的に組み合わさるような場合に制御が困難である．

楽器音の奏法変化にともなう音響的性質の変化を考えたとき，対象をエレクトリック・ベースに代表される単旋律撥弦楽器に限定すれば，アタック部分ではピッキングノイズによる非周期成分が，サステイン部分では弦振動による周期成分が支配的である．これは音声における子音と母音の関係に類似しており，音声合成と同様の枠組みをエレクトリック・ベースの楽器音合成に応用できる可能性がある．

本研究はこの類似に着目し，単旋律撥弦楽器の奏法を音

[1] 明治大学
Meiji University
[a)] korguchi@meiji.ac.jp
[b)] mmorise@meiji.ac.jp

素のように表記するアタック–サスティン表現を用いて，奏法を制御可能な楽器音合成手法を提案する．まず，アタック区間とサスティン区間のどちらに大きな音響的な変化を与えるかによって，奏法を 2 種類に大別する（以後，アタックに変化を与える奏法をアタック奏法，サスティンに与える奏法をサスティン奏法と呼ぶ）．そして，1 つのノートをアタック区間とサスティン区間に分割し，それぞれに対応するアタック，サスティン奏法を割り当てる．これにより，ピックを使ってハーモニクス奏法を演奏するといった，1 つのノートに対して複数の奏法を組み合わせることが可能になる．加えて，奏法がそれぞれ異なる複数の楽器を扱う場合でも，多言語モデルと同様に処理することで 1 つのモデルで複数の楽器を合成するモデルを構築できる．以下では，この奏法表現を用いてエレクトリック・ベースの楽器音合成システムを構築した結果を報告する．

2. 関連研究

深層学習に基づく楽器音合成において，end-to-end モデル [1] のような入力と出力が一貫した枠組みは，そのままの出力が十分に高品質であればユーザによる修正の手間を省くことができる．一方で，ユーザの好みに合わせて音高や長さなど，合成音の性質を細かく制御することができる枠組みを提供することは，動画や音楽制作などプロダクション用途への応用を考える上で重要である．

音声合成モデルとして提案された FastSpeech [8] は length regulator と呼ばれる機構により，音素ごとの埋め込みベクトルが継続長に合わせて複製されてから decoder へ入力されるため音素継続長の制御が可能であるほか，非自己回帰型のネットワークにより高速に推論が行えるため楽音の合成にも用いられる [9]．

Deep Performer は FastSpeech の枠組みを楽器音合成に適用するために，音素の代わりとして楽譜から得られた音高・継続長・タイミング・音量を encoder への埋め込み，音価とオンセットを length regulator に入力する．また，フレームレベルに展開された特徴量に音価におけるフレームの相対的な位置が埋め込まれることで（note-wise positional encoding），ノートの先頭・中間・終端にかけての音色変化をモデル化している．本研究はこの Deep Performer に基づき，奏法の制御を可能な枠組みへと改良を行う．

3. 提案手法：アタック–サスティンモデルによる奏法表現

3.1 奏法ラベルの設計

エレクトリック・ベースのような撥弦楽器は指やピックで弦を弾くことで発音される．まず，弦がピック/指/フレットに衝突して非周期的なノイズが発生する．その後，周期的な弦振動が発生し次第に減衰する．このとき，押弦の仕方（ミュート／ハーモニクスなど）を変えることで応じて弦振動は異なる振る舞いをする [10]．このような発音原理を踏まえると，エレクトリック・ベースの奏法によってもたらされる音響的性質の差異は，アタック区間に現れるものとサスティン区間に現れるものに大別できる．**表 1** に本研究で扱うアタックおよびサスティン奏法とその分類を示す．

表 1 アタックおよびサスティン奏法にそれぞれ対応する奏法の一覧．

アタック奏法	サスティン奏法
Finger, pick, thump, thumb up, pluck, hammer on, pull off	Sustain, mute, harmonics, slide up, slide down

3.2 音響モデル

提案する楽器音合成システムの音響モデルには Deep Performer [9] の枠組みを利用する．そのうち encoder 部の概略図を**図 2** に示す．まず，楽譜から音高・オンセット・音価・ベロシティを抽出し，それぞれ全結合層による埋め込みがなされる．本研究はこれらの特徴に加えて，奏法の時間的変化の情報を，アタック–サスティン表現に変換した上で埋め込む．楽譜情報だけでなく奏法によって条件付けされ，合成音は奏法を反映したものとなることを期待する．その後，encoder によって得られた特徴量系列は， length regulator によってオンセット時刻および音価に合わせたフレーム数の分だけ複製される．そして，再度 positional encoding を経た後に decoder へと渡され，最終的にメルスペクトログラム系列が推定される．音響モデルは推論されたメルスペクトログラムとターゲット音のメルスペクトログラムとの平均二乗誤差を最小化するように学習される．なお，オリジナルの Deep Performer とは異なり学習時と推論時いずれも音価とオンセットは ground truth を直接入力する．また，単旋律楽器であるため，多重音のための polyphonic mixer は組み込まれていない．

3.3 メルスペクトログラムからの波形生成

Decoder から出力された音響特徴量から波形を合成するための手法は，音響特徴量から決定論的に設計された励振源とフィルタを用いて波形を合成するチャネルボコーダ [11] と，音響特徴量と波形との対応を統計的に学習・推論するニューラルボコーダ [12] に大別される．

エレクトリック・ベース音を合成することを考えたとき，その音域は最低音（レギュラーチューニング）で E1（およそ 60 Hz）に及ぶ．チャネルボコーダの多くはフィルタに最小位相特性を仮定しており，低音を合成する際に波形のエネルギーが局所的に集中するため buzzy と形容される音質劣化が顕著になることが予想される．

そこで，本研究は波形生成にニューラルボコーダである

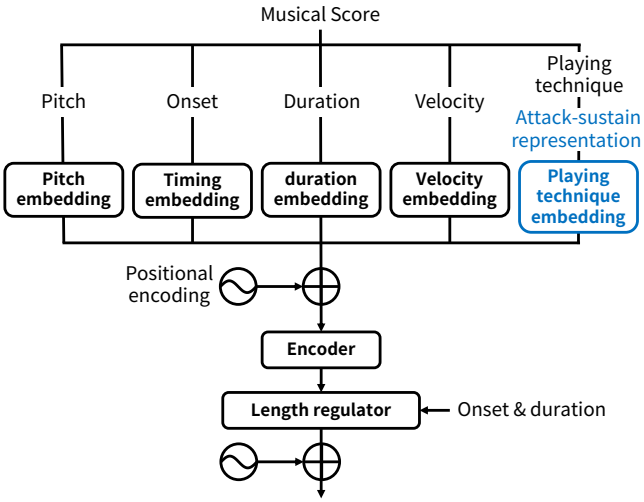

図 2 アタック–サスティン表現を用いた楽器音合成システムの入力から encoder 部．楽譜情報から得られた奏法情報はアタック–サスティンラベルに変換されてから埋め込まれる．

BigVGAN [13] を用いる．学習データの分布から外れた音に対しても頑健な動作することが報告されており，音高や奏法変化が多様なエレクトリック・ベース音においても高品質な波形生成を期待できる．事前実験として，音声のみを使って訓練された学習済みモデルによる分析再合成を行ったが，良好な品質の合成音が得られることを確認している．

4. 楽器音合成システムの実装

4.1 データセット

DNN の学習に使用する楽器音として，新たにエレクトリック・ベース音のデータベースを構築した．それぞれ 60 から 120 beat per minute; BPM で 4 小節の単旋律のベースラインを 180 フレーズ（約 112 分）収録した．収録されたフレーズは奏法リスト（**表 1**）にあるすべての奏法をカバーしており，楽譜情報および各ノートに対応する奏法ラベルは手動で与えられた．演奏者はアマチュア奏者が行い，エレクトリック・ベースは Fender custom shop 1962 Jazz Bass [14]，オーディオインターフェースは RME ADI-2 Pro FS R [15] を使用し，48 kHz サンプリング，24 bit 量子化の wav ファイルとして収録した．

4.2 ラベルと音響信号とのアライメント

スペクトログラムや音響特徴量の時系列から奏法の境界を手動で与える場合，熟練した知識と時間を要するため困難である．そこで，まず既存の合成器から出力された合成音とデータベースの音響信号に対して，dynamic time warping (DTW) を用いて時間的な対応を取ることで，楽譜に忠実な発音時刻からのずれを計算することで境界を求めた [16]．このとき，合成音を Gaussian Mixture Model (GMM) に基づく声質変換の枠組みを用いて合成音をターゲットの音色へ変換してから実施することで DTW の精度を向上させた．音響特徴量には Harvest [17]，D4C [18] を用いてそれぞれ抽出した基本周波数と非周期性指標，16 kHz にダウンサンプリングした波形から抽出した 24 次元メルケプストラムを用いた．フレーム長は基本周波数と非周期性指標の抽出には 2048，メルケプストラムは 1024 に設定し，シフト長はいずれも 5 ms で行った．音色変換に用いた GMM の混合数は 9 とした．Harvest の推定範囲は，レギュラーチューニングにおける最低音付近の 60 Hz を下限に，ハーモニクス奏法における最高音付近の 400 Hz を上限に設定した．

4.3 音響モデルとニューラルボコーダ

実装も Deep Performer [3] と同様に，すべての埋め込み次元は 128 次元であり，エンコーダとデコーダはそれぞれ 3, 6 つの feed-forward Transformer（FFT） ブロックで構成した．

ノート埋め込みに使用した音高・オンセット・音価・ベロシティは，MIDI 上のノート情報から 4 分音符を 24 分割した時間解像度で抽出した．音声は 24 kHz モノラルにダウンサンプリングし，100 次元のメルスペクトログラムを抽出した．このとき，分析窓にはハン窓，窓長 1024 サンプル，シフト長は 256 サンプルで計算を行った．学習の最適化アルゴリズムには Adam [19] を使用し，バッチサイズは 16，100000 ステップの学習を実施した．

BigVGAN は著者の GitHub リポジトリ（`https://github.com/NVIDIA/BigVGAN`）で提供されている学習済みモデル（bigvgan_base_24khz_100band）を fine-tuning する形で学習した．メルスペクトログラムの計算および最適化アルゴリズムは音響モデルと同様である．学習時に切り出すセグメントは 8192 サンプルであり，バッチサイズは 32，10000 ステップの学習を実施した．

5. 結果

5.1 奏法の制御

構築した楽器音合成システムが実際に奏法の制御性を備えているかを確認するため楽器音を合成する実験を行った．実際に合成した楽器音のメルスペクトログラムを**図 3** に示す．上段と下段は音高・タイミング・音価・ベロシティはそれぞれ同一のフレーズであるが，一方は一音のみアタック奏法を他の奏法に置換したものである．具体的には，上段は全てフィンガー・ピッキングで，下段は 5 音目のアタック奏法をサムピングに置換して合成したものである．白線で描かれた四角に囲まれた部分に注目すると，下段のアタック部分の非周期成分がより強く変化していることが確認できる．サムピングは弦を親指で叩く奏法であり，フィンガーピッキングと比べて弦がフレットや指に強く衝突する．合成結果にも奏法変化に伴う妥当な音響的性質の変化

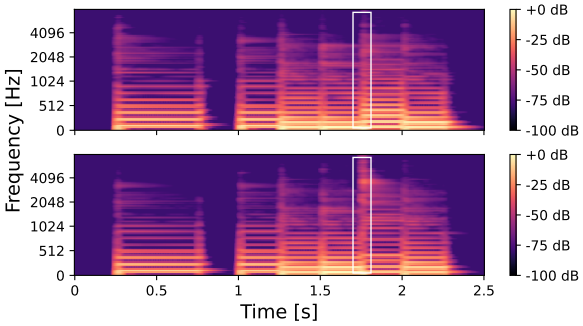

図 3 合成音のメルスペクトログラム．上段は全てフィンガー・ピッキングで，下段は 5 音目のアタック奏法をサムピングに置換して合成したもの．四角で囲まれた部分の音響特性が変化していることが分かる．

が反映されていると考えられる．聴感上も奏法の変化を感じることを確認している．

5.2 議論

提案した楽器音合成システムが奏法を制御できる可能性が示唆された．一方で，1 つのノートにつき 1 種類の奏法を割り当てた場合と比較して，合成音の品質や制御性がどのように異なるのかは未知である．エレクトリック・ベースの演奏に関して，構築したデーターセットに収録されているフレーズに着目すると，指弾きとピック弾きのように奏法が異なっていてもサスティン部分は共通であるものが多く含まれている．このように，ノート毎に全く別の奏法として区別するよりも，音響的な性質の変化に基づいたコンテキストを設計することで，効率的に音響変化を学習でき合成音の品質が向上する可能性がある．実際，日本語テキスト音声合成において，音節単位よりも音素単位での表現を用いたほうが合成音声の品質が向上することが報告されている [20].

6. おわりに

本研究は音声における音韻変化と単旋律撥弦楽器の奏法変化の音響的な類似に着目し，奏法をアタック部分とサスティン部分に分けて音素のように表記するアタック–サスティン表現を提案した．楽器音合成において提案手法を用いて奏法の制御が可能であることを確認した．

今後は，**5.2 節**において挙げた課題の調査や，多重音への改良，撥弦楽器以外の楽器へのアタック–サスティン表現の適用に取り組む予定である．

謝辞 本研究は，JSPS 科研費 JP22J22158 の支援によって行われた．

参考文献

[1] Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerrv-Ryan, R., Saurous, R. A., Agiomvrgiannakis, Y. and Wu, Y.: Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions, *Proc. ICASSP 2018*, pp. 4779–4783 (2018).

[2] Nishimura, M., Hashimoto, K., Oura, K., Nankaku, Y. and Tokuda, K.: Singing voice synthesis based on deep neural networks, *Proc. Interspeech 2016*, pp. 2478–2482 (2016).

[3] Dong, H.-W., Zhou, C., Berg-Kirkpatrick, T. and McAuley, J.: Deep Performer: Score-to-Audio Music Performance Synthesis, *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 951–955 (2022).

[4] Cooper, E., Wang, X. and Yamagishi, J.: Text-to-speech synthesis techniques for MIDI-to-audio synthesis, *Prpc. ISCA SSW 11*, ISCA, pp. 130–135 (2021).

[5] 音楽電子事業協会：MIDI 1.0 企画書，`https://www.midi.org/specifications`.

[6] Toontrack: EZBass, `https://www.toontrack.com/product/ezbass/`.

[7] IK Multimedia: MODO Bass 2, `https://www.ikmultimedia.com/products/modobass2/`.

[8] Ren, Y., Ruan, Y., Tan, X., Qin, T., Zhao, S., Zhao, Z. and Liu, T.-Y.: Fastspeech: Fast, robust and controllable text to speech, *Adv. Neural Inf. Process. Syst.*, Vol. 32 (2019).

[9] Chen, J., Tan, X., Luan, J., Qin, T. and Liu, T.-Y.: HiFiSinger: Towards High-Fidelity Neural Singing Voice Synthesis (2020).

[10] Abeßer, J., Lukashevich, H. and Schuller, G.: Feature-based extraction of plucking and expression styles of the electric bass guitar, *Proc. ICASSP 2010*, pp. 2290–2293 (online), DOI: 10.1109/ICASSP.2010.5495945 (2010).

[11] Gold, B. and Rader, C.: The channel vocoder, *IEEE Trans. Audio Electroacoustics*, Vol. 15, No. 4, pp. 148–161 (1967).

[12] Tamamori, A., Hayashi, T., Kobayashi, K., Takeda, K. and Toda, T.: Speaker-dependent wavenet vocoder, *Proc. Interspeech 2017*, Vol. 2017, pp. 1118–1122 (2017).

[13] Lee, S.-G., Ping, W., Ginsburg, B., Catanzaro, B. and Yoon, S.: BigVGAN: A Universal Neural Vocoder with Large-Scale Training (2022).

[14] Shop, F. C.: .

[15] RME: ADI-2 Pro FS R Black edition, `https://www.rme-audio.de/adi-2-pro-fs-be.html`.

[16] Koguchi, J., Takamichi, S. and Morise, M.: PJS: phoneme-balanced Japanese singing-voice corpus, *Proc. APSIPA ASC 2020*, pp. 487–491 (2020).

[17] Morise, M.: Harvest: A high-performance fundamental frequency estimator from speech signals, *Interspeech 2017*, ISCA (2017).

[18] Morise, M.: D4C, a band-aperiodicity estimator for high-quality speech synthesis, *Speech Communication*, Vol. 84, pp. 57–65 (2016).

[19] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, *Proc. ICLR 2015* (2015).

[20] Fujimoto, T., Hashimoto, K., Oura, K., Nankaku, Y. and Tokuda, K.: Impacts of input linguistic feature representation on Japanese end-to-end speech synthesis, *Proc. ISCA SSW 10* (2019).