

持続音楽器の演奏表情付けシステムにおける ビブラート表現力向上の検討

福知山公立大学 情報学部情報学科

32245024 勝川千聖

指導教員 橋田光代 准教授

提出日 2026 年 1 月 31 日

改訂日 2026 年 2 月 7 日

— 目次 —

| | | |
|----------|---------------------------|----------|
| 1 | はじめに | 1 |
| 2 | 定義と歴史 | 1 |
| 2.1 | ヴィブラートの定義 | 1 |
| 2.2 | ヴィブラートの歴史 | 1 |
| 2.2.1 | 弦楽器 | |
| 2.2.2 | 管楽器 | |
| 2.2.3 | 声楽 | |
| 2.2.4 | 電子音楽 | |
| 3 | 電子音楽について | 2 |
| 3.1 | 仕組み | 2 |
| 3.1.1 | 電子楽器 | |
| 3.1.2 | 歌声合成システム | |
| 3.2 | ヴィブラートの再現を試みる目的 | 2 |
| 4 | ヴィブラートシステムの検討 | 2 |
| 4.1 | 目的 | 2 |
| 4.2 | 使用アプリケーション | 3 |
| 4.3 | 実装 | 3 |
| 4.3.1 | システム全体構成 | |
| 4.3.2 | ヴィブラートのモデル化 | |
| 4.3.3 | Python および Max 8 による実装 | |
| 4.4 | 結果 | 4 |
| 4.5 | 展望 | 4 |
| 5 | 結論 | 4 |

— 図目次 —

| | | |
|---|-------------------------------|---|
| 1 | システム全体構成 | 3 |
| 2 | 赤とんぼ楽譜 | 3 |
| 3 | 実行時の Python ターミナル画面 | 4 |
| 4 | Max 8 構成 | 4 |
| 5 | ヴィブラート波形の可視化例 | 4 |

1. はじめに

近年、コンピュータによる音楽生成や演奏支援に関する技術は非常に発展している。プログラミングや信号処理を用いた表現など様々な観点、方法で演奏表現の再現が試みられている。しかしながら、機械演奏における音楽的表現には、依然として人間の演奏に比べ限定的である。

筆者は大学においてプログラミングおよび情報処理技術を学ぶ一方で、管楽器演奏の経験があり、これら二つの知識を合わせた音楽表現システムの構築に強く関心を持っている。本研究は、演奏経験に根ざした表現を、コンピュータ上でどのように扱い、拡張できるかを探求することを目的とする。

特に管楽器における音楽表現は、音高や音量だけでなく、息の流れや身体捜査による連続的かつ時間的に変化する表現に依存している。その代表的な要素の一つがヴィブラートである。ヴィブラートは、ピアノや打楽器のような楽器では本質的に実現が難しく、管楽器や弦楽器、声といった持続音を出すことができる楽器で、音楽表現の幅を拡張する技法として用いられてきた。

一方で、ヴィブラートは広く用いられている表現技法であるにもかかわらず、その実態や生成メカニズムについては演奏教育の場においても必ずしも明確に説明されているとは限らない。筆者自身も、管楽器演奏においてヴィブラートの取得を試みた際、「揺らす」「かける」といった感覚的な指導を受けることは出来たものの、どのような身体動作や音響的变化によってヴィブラートが生じているのかについては十分に理解をすることは出来なかった。耳で聞こえる音と自身の感覚を頼りに習得を進めていた。ヴィブラートは、演奏者の感覚的理解にゆだねられる部分が大きく、ジャンルや楽器、演奏者によってヴィブラートの方法が異なる。そのため、コンピュータ上で管楽器特有のヴィブラート表現を再現、制御するための統一的なモデルは十分に確立されていない。

本研究では、楽器演奏におけるヴィブラート表現に着目し、その実態を整理分析するとともに、演奏者の立場から理解可能で、かつコンピュータ上で扱いやすい表現モデルの構築を目指す。

2. 定義と歴史

2.1 ヴィブラートの定義

ニューグロブ音楽辞典¹⁾において、ヴィブラート (vibrato) とは、音の高さ (音高) を中心とした周期的な変動によって生じる効果であり、主として声楽および多くの旋律楽器において用いられる表現技法であると定義されている。この変動は、一定の中心音高を基準として上下に揺れる形で生じ、音程の変化幅、および変動速度によって特徴づけられる。

また同辞典では、ヴィブラートは単なる装飾音ではなく、音色・音量・響きの豊かさに影響を与える総合的な表現現象であるとされており、特に 20 世紀以降の演奏監修においては旋律音を持続させる基本的な音響要素として位置づけられていることが指摘されている。

また、ヴィブラートはしばしばトレモロ (tremolo) と混同されるが、両者は本来異なる概念である。ヴィブラートが主として音高の変動を指すのに対し、トレモロは音量 (振幅) の周期的な変動を指す用語であり、ニューグロブ音楽辞典でも両者は明確に区別されている。ただし、実際の演奏においては音高変動と音量変動が同時に生じる場合も多く、聴覚的には複合的な効果として知覚されることが指摘されている。

ヴィブラートの生成メカニズムは楽器や発声方法によって異なる。声楽では生体振動および呼気圧の周期的変動が関与し、弦楽器では指による弦長の連続的变化、管楽器ではあごや唇、息の制御による共鳴条件の変化が主な要因となる。このように物理積生成方法は多様であるにもかかわらず、聴覚的には共通した「ヴィブラート」として認識される点は、ヴィブラートが単なる物理現象ではなく、音楽的・近く概念であることを示している。

以上のように、ヴィブラートとは、音高を中心とした周期的かつ微小な変動によって音に生命感と表現的深みを与える音楽的技法であると言える。

2.2 ヴィブラートの歴史

弦楽器、管楽器、声楽、電子音楽におけるヴィブラートの歴史について述べる。弦楽器、管楽器、声楽はニューグロブ音楽辞典²⁾を参考にした。電子音楽は、文献³⁾を主に参考にした。その他は文中に引用を示す。

2.2.1 弦楽器

弓奏弦楽器に関しては少なくともバロック初期から 20 世紀初頭のクライスラー (1875～1962) の世代に至るまで、演奏の中で常用されていたとされる。この時代におけるヴィブラートの程度はかなり連続的なものから、要所のみに用いられるが目立つものまでさまざまであった。このような弦楽器におけるヴィブラート観は、時代や流派、演奏美学によってその評価や使用頻度が異なっていた。例えば、ほとんどすべてのバロックや古典派、およびロマン派音楽においての適切なヴィブラートの用法は、控えめで連続的なヴィブラート (左手による色彩感付与と考えられる) と断続的で目立ったヴィブラート (装飾と考えられる) に分け、両者を混ぜ組合せ用いることであり、後期ロマン派の作品や 20 世紀の作品は、より極端なヴィブラートが適切であると書かれている。また考え方としても、表現性増強のための特殊な装飾法とみなす考え方、音楽に心地よい陰影を与えるために連続的に使用するべきものという 2 つの考え方があり、19 世紀には通常、ヴィブラートは表現性増強のための装飾

とみなされた。

2.2.2 管楽器

管楽器におけるヴィブラートの歴史に関して、ニューグロブ音楽辞典には書いていなかった。しかし、ヴィブラートのかけ方は弦楽器とは異なるが、演奏の中でどのように使われてきたかは弦楽器とほぼ等しいと考えることもできる。

2.2.3 声楽

声楽においてのヴィブラートは正式なヴィブラートの構成要素である音程の変動を是認しないが、実際はトレモロである声の強さに同じような変動を与えることは多少なりとも勧めておりヴィブラートと誤称されている。正式のヴィブラートと正式のトレモロ(声楽のヴィブラート)は両者ともに、中世の歌唱論者によってほとんど容認出来ない効果として記述されている。

2.2.4 電子音楽

20 世紀初頭から電氣的に新たな音を作り出す試みが始まり、様々な電子楽器がつくられるようになった。「テルミン」や「オンド・マルトノ」といった楽器もこの時代につくられた。二つの発振回路を作り、2つの振動のうなりによって音声がつくられる。そして、人体とアンテナの位置関係によって決まる静電気量の変化によって音の高さや音量を変化させるという仕組みになっている。このような装置は電氣的に連続的な音高操作が可能となり、独特のグリッサンド、ヴィブラートを表現することができるようになった。

第二次世界大戦後、戦争がもたらしたテクノロジーの急速な発展の恩恵を受けて、電氣的に合成・加工・編集することで、自由なサウンド構築可能な電子音を使用した電子音楽がつくられるようになった。20 世紀中頃になると音合成ソフトウェアなどが開発されるようになる。コンピュータがデジタル音源を直接コントロールする技術が登場した。この技術がデジタル・シンセサイザー全盛期の序章となった。シンセサイザーの発展により、LFO（低周波発振器）を用いた意図的なピッチ変調が一般化し、ヴィブラートは電子楽器上で設計可能な音響パラメータとなった。最後に、デジタル歌声合成、なかでも VOCALOID 系²⁾は 2003 年の商用化以降に「音の高さ（ピッチ）」や「各倍音のレベル（音色）」をパラメータとして精密に扱えるようになり、ヴィブラートを周期・深さ・位相・高さなどのパラメータで表現し用いるようになった。

3. 電子音楽について

電子音楽とは、電子技術を用いて生成・加工された音を素材として構成された音楽である。

3.1 仕組み

3.1.1 電子楽器

電子楽器の基本的な音生成は電気信号の合成・加工によって行われる。発振器によって生成された基本波形に対

し、フィルタやアンプなどの信号処理を施すことで音色を形成する。電子楽器の一つである「シンセサイザ」は、最初に「オシレータ」という発振器で基本波形である、正弦波、三角波、矩形波などを生成し、それをフィルタやエンベロープで変化させることにより音色を合成する。この手法を減算合成といわれる。

減算合成型のシンセサイザにおけるヴィブラートは、低周波発振器 (LFO) を用いたピッチ変調として実装されている。具体的には深さと速さといったパラメータで表現強弱を制御する。すなわち、電子楽器ではピッチ変動を数値化した制御信号でヴィブラートが再現されている。

3.1.2 歌声合成システム

VOCALOID に代表される歌声合成システムは、人間の歌声を対象とした音声情報処理技術に基づいて音を生成する。VOCALOID は、ヤマハが開発した歌声合成技術およびその応用ソフトウェアの事である。音符と歌詞を入力するエディタ、実際の歌声から取り出した音声素片の集まりである歌声ライブラリ、音声素片を接続・変換する合成エンジンの 3 つで構成させる。歌声素片は、事前に録音された歌声や音声から切り出し加工して作成され、周波数領域で接続・補間・変換することで自然な歌声に近づけている。

歌声合成一般においてヴィブラートは、音高の周期的な変動として実装される歌唱表現の要素である。その制御には速さと深さという 2 つのパラメータが用いられている。

3.2 ヴィブラートの再現を試みる目的

電子楽器と歌声合成システムは、ともにコンピュータをもちいた音生成システムであるが、その開発の目的の本質は異なる。

田中氏の電子音楽史の研究 [3] では、電子楽器は新たな音響表現の創出を目的として発展してきたことが述べられている。そこでは、音そのものを設計対象とし、既存の楽器音や人声に必ずしも依拠しない音楽表現が追及されてきた。あくまで、ヴィブラートは音色や音響表現を設計・拡張するための機能として位置づけられる。

一方で、歌声合成システムは、人間の歌唱行使を機械的に再現することを主としており、音響的自然さや歌唱らしさが重視される。したがって、ヴィブラートも音色の変化の一種ではなく歌唱表現の一部として扱われる。

4. ヴィブラートシステムの検討

4.1 目的

ヴィブラートに関して調べてきた。ヴィブラートは演奏者の感覚に依存する表現技法であると私は考える。熟練した演奏者は他者が求めるヴィブラートの在り方を感覚的に把握できる一方で、初心者にとってはその判断が困難であることが多い。指導の場においては、声や実演によって具体例が示されることもあるが、その時ごとに揺れ方が異な

るため、学習者が同一条件のヴィブラートを繰り返し聴取・比較することは難しい。

そこで本研究では、ヴィブラートを機械音声として簡単に制御・再現できるシステムを作成することで常に同一条件のヴィブラートを提示し、どのようなヴィブラートが求められているかを客観的に理解できる環境を構築することできるようにする。このようなシステムにより、感覚的に語られがちなヴィブラートを可視化、可聴化し、学習者が具体的なイメージを持って練習に取り組むことが可能になると考える。

一般的に、上級者は意図的かつ持続的にヴィブラートをかけることができるが、初心者は常に安定したヴィブラートをかけることが難しく、十分な練習を行わなければ理想とされる揺れに到達できない。本研究では特に、「初心者が音楽的に適切な箇所では、どのようなヴィブラートをかければよいのか」を直感的かつ簡潔に理解できるシステム構築を目指す。

なお、本システムの設計には、筆者が演奏経験を有するフルートにおけるヴィブラート練習の知見を一部取り入れている。そのため、本研究で扱うヴィブラートのモデルには、フルート演奏における実践的な感覚に基づく要素が含まれる可能性がある。

4.2 使用アプリケーション

本研究で使用したアプリケーションは、Python および Max 8 である。Python は汎用的なプログラミング言語であり、主に演奏情報及び制御データの生成・処理を担っている。Max 8 は音楽・音響分野で広く用いられているビジュアルプログラミング言語であり、Python からのデータを受信して音声生成およびリアルタイムな音響処理を行う役割を果たしている。両者はデータ通信によって連携し、Python 側で設定したパラメータをもとに、Max 8 上でヴィブラートを含む音響表現を制御する構成とした。

この構成により、ヴィブラートの揺れ幅や周期などを数値的に調整し、それを音として再現することが可能となり、同一条件のヴィブラートを繰り返し提示できるシステムを実現する。

4.3 実装

4.3.1 システム全体構成

本システムの処理は、大きく Python 側の処理と Max 8 側の処理に分けられる。システムの全体構成は図 1 に示す通りである。

まず、Python プログラムにおいてユーザーが MIDI ファイルを選択すると、当該ファイルが読み込まれ、楽曲の拍子情報が解析・表示される。次に、ユーザーは「何拍以上の長さを持つ音にヴィブラートをかけるか」という条件を指定し、その条件を満たすノートが抽出される。該当するノート

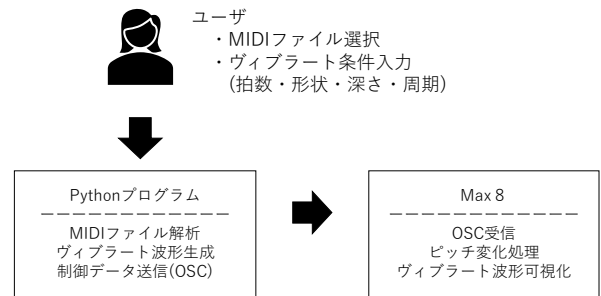


図 1: システム全体構成



図 2: 赤とんぼ楽譜

数がユーザーに提示され、条件が適切でない場合には再選択を促すようにした。その後、抽出されたノートの中から、ヴィブラートを付加するノートを選択し、各ノートに対してヴィブラートの形状を指定する。指定されたヴィブラート情報は Python 側で時間変化する数値データとして生成され、OSC 通信を用いて Max 8 に逐次送信される。Max 8 では受信した情報に基づき音声を合成し再生を行う。

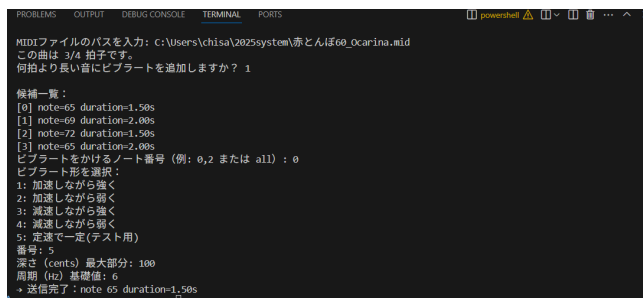
4.3.2 ヴィブラートのモデル化

本研究では、ヴィブラートを音高の周期的変動として定義し、その変動量をセント（cents）単位で行う。ヴィブラートは正弦波による周期変動を基本とし、周期（Hz）および深さ（音高変動幅）という 2 つのパラメータによってモデル化した。

さらに、演奏中のヴィブラートが時間とともに変化する点を考慮し、周期および深さが継続時間に応じて変化するようにした。初心者が感覚的に理解しやすいよう、ヴィブラートの変化は以下の 5 パターンに限定した。

- (1) 加速しながら強く
- (2) 加速しながら弱く
- (3) 減速しながら強く
- (4) 減速しながら弱く
- (5) 定速で一定（テスト用）

これらは、「加速」「減速」「強く」「弱く」といった演奏上の感覚的表現を、周期および深さの時間変化として数値的に定義したものである。各時刻における音高変動量は、周期と深さを反映した正弦関数によって算出される。周期はヴィブラートの細かさ、深さは音の変化の大きさを示している。



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
MIDIファイルのパスを入力: C:\Users\chisa\2025system\赤とんぼ60_ocarina.mid
この曲は 3/4 拍子です。
何拍より長い音にビブラートを追加しますか? 1

編集一覧:
[0] note=65 duration=1.50s
[1] note=69 duration=2.00s
[2] note=72 duration=1.50s
[3] note=65 duration=2.00s
ビブラートをかけるノート番号 (例: 0,2 または all) : 0
ビブラート形を選択:
1: 加速しながら強く
2: 加速しながら弱く
3: 減速しながら強く
4: 減速しながら弱く
5: 定速で一定 (テスト用)
番号: 5
深さ (cents) 最大部分: 100
周期 (Hz) 基礎値: 6
+ 送信完了: note 65 duration=1.50s
```

図 3: 実行時の Python ターミナル画面

4.3.3 Python および Max 8 による実装

Python

Python プログラムでは、まず MIDI ファイルを読み込み、楽曲に含まれる拍子およびテンポ情報を取得する。次に、各ノートの開始時間及び、終了時刻を解析し、ノートの長さを算出する。この際、MIDI メッセージの tick 情報をテンポおよび ticks per beat に基づいて実時間（秒）に変換することで、時系列を保持したノート情報を得ている。

得られたノート群の中から、指定した拍数以上の長さを持つノートを抽出し、ユーザがヴィブラートを適用する対象ノートを選択できるようにした。これにより、短い音符への不要なヴィブラートの付加を避けることができる。

ヴィブラートの制御信号は、正弦波を基本形とし、その周期および振幅を時間的に変化させることで生成する。具体的には、加速・減速および強弱の変化を組み合わせた複数のビブラート形状を定義し、ユーザが選択した形状に応じて、時間経過に伴い周期および振幅が変化するように設計した。生成されたビブラート量は cents 単位で表され、OSC メッセージとして逐次 Max8 に送信される。

このように Python 側では、音響合成そのものは行わず、あくまで「音高変動を記述する制御信号の生成」に役割を限定している。

Max 8

Max 8 側では、Python から送信された OSC メッセージを UDP 経由で受信する。受信したメッセージは、ノート情報と音高変動量に分歧して処理され、ビブラート制御信号は cents 表記から周波数比へ変換される。

4.4 結果

本研究において構築したヴィブラートシステムでは、Python プログラムから OSC を用いて送信された音高変動量は Max 8 で受信され、可視化することができた。一定周期の制限波形に加え、周期や振幅が時間的に変化する複数のヴィブラートの形状が波形表示錠で確認できた。

4.5 展望

今後の展望として、まず可視化されたヴィブラート波形と聴覚上の印象との対応関係を評価することが挙げられる。

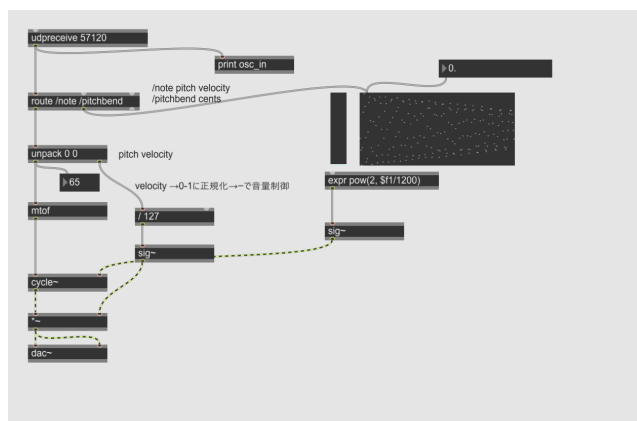


図 4: Max 8 構成

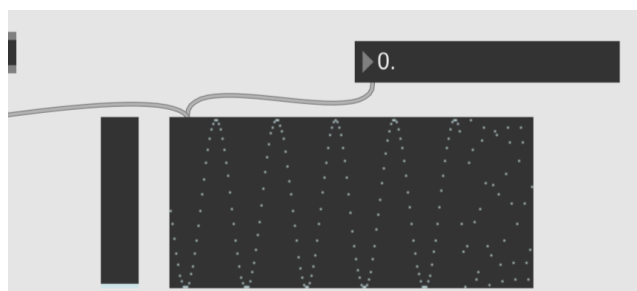


図 5: ヴィブラート波形の可視化例

また、生成されたヴィブラート制御信号をユーザ自身が直接操作できるよう、波形を視覚的に編集しながら感覚的にヴィブラートの形状を調整できる機能の実装も今後の課題である。これにより、ヴィブラートを数値やプログラムとしてでなく形状として理解することが可能になると期待出来る。

さらに、専用のユーザインタフェースを整備することで、プログラミングの知識を持たないユーザでも容易に本システムを利用できる環境を構築したい。将来的には波形操作と音響的な効果を同時に体験できるインタラクティブなヴィブラートシステムへと発展させることを目指す。

5. 結論

本研究では、ヴィブラートという音楽的表現について、ニュエグロブ音楽辞典を中心とした定義の整理を行った。ヴィブラートは、音高を中心都市ら周期的な微笑変動によって生じる興亜であり、単なる装飾音にとどまらず、音色や響き、表現の豊かさに関与する総合的な音楽現象であることを確認した。また、その物理的生成方法は、声楽・弦楽器・管楽器などによって異なるにもかかわらず、聴覚的には共通の表現として認識される点から、ヴィブラートが物理現象のみならず音楽的概念として成立していることが考えられる。

次にヴィブラートの歴史を整理するとともに、電子音楽および音声合成における音生成の仕組みと、そこでのヴィブラートの扱われ方について考察を行った。ヴィブラートの

使用頻度や美学的な評価は時代や様式によって大きく変化してきた。特に弦楽器において装飾的な用法と連続的な用法が併存してきたこと、声楽においては理論上の否定と実践上の容認という乖離が存在していたことが確認された。また、20 世紀以降の電子音楽の発展によりヴィブラートは演奏者の身体運動に依存するものから、意図的に設計・制御可能な音響パラメータへと変化していった。

さらに、電子楽器および歌声合成システムの仕組みを整理し、それぞれにおけるヴィブラートの位置づけを比較した。電子楽器においては、今回あげた例では LFO によるピッチ変調としてヴィブラートが実装され、音響表現を設計するための機能の一つとして扱われていた。一方、歌声合成システムでは、人間の歌唱表現の再現を目的とするため、ヴィブラートは歌唱らしさを構成する重要な表現要素として位置づけられ、周期や深さといったパラメータによって精密に制御されている。この違いは、両者の開発目的の本質的差異を反映したものである。

以上の知見をふまえ、本研究では初心者を対象としたヴィブラートシステムの作成を行った。本システムはヴィブラートを感覚や経験で取得するのではなく、楽曲中の特定の部分にヴィブラートをかけ同じ形のヴィブラートを何度も聞き自分が求める理想のヴィブラートを何度も聞くことができる特徴がある。これにより、初心者でもどのようなヴィブラートをかければよいかわかりやすくなりヴィブラートの取得を行うことができるようになると思う。

本研究では、ヴィブラートを歴史的・音響的・情報処理的観点から横断的に捉え、その理解を教育的応用へと結びつけた。今後は、実際の学習者による使用評価や、個人差を考慮したパラメータ調整機能の導入などを通して、より実践的なヴィブラートシステムへと発展させることが課題である。

謝辞 本研究を進めるにあたり、指導教員の橋田光代准教授には大変お世話になりました。また、研究室 4 年生の皆さんをはじめ、多くの方々から貴重な助言や支援をいただきました。深く感謝申し上げます。

参考文献