持続音楽器の演奏表情付けシステムにおける
ヴィ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 30 日

— 目次 ———

— 図目次 ———

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，息の流れや身体操作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のにもかかわらず，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変化によってヴィブラートが生じている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

ニューグローブ音楽辞典 [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，ニューグロー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変化，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，音楽的知覚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，声楽はニューグローブ音楽辞典 [1] を参考にした．電子音楽は，莱氏の研究 [2] と田中氏の研究 [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，ニューグロー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，弦楽器と同じように教会や宮廷音楽の中で発展してきた歴史もあるため，演奏の中でどのように使われてきたかは弦楽器とある程度等しいと考えることもできるだろう．

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 世紀初頭から電気的に新たな音を作りだす試みが始まり，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 [4]．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気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，電気的に合成・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[5] 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・高さの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 [4] [6] 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 [7] 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[5] 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 [8] 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] によれば，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 [9]．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査し，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筆者は考える．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．また, ヴィブラートを波形化するもの [10] はあるが，特定の部分に特定の波形を再現し実際の音として再現されることは少ない．

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をできるようにする．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，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．

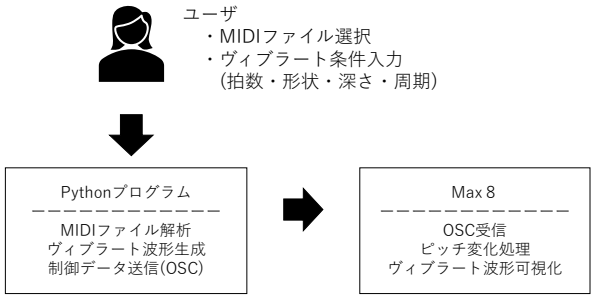

図 1: システム全体構成

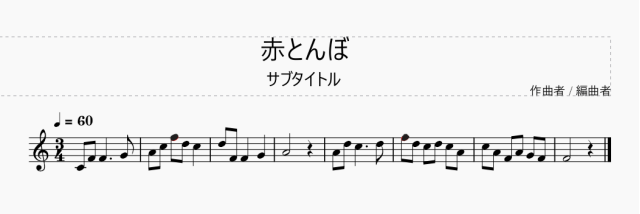

図 2: 赤とんぼ楽譜

まず，Python プログラムにおいてユーザが 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．該当するノート数がユーザに提示され，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，リアルタイムにデータを送受信するための通信プロトコル 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．

今回研究に使用した楽譜は**図 2** に示す「赤とんぼ」である．筆者がフルートでヴィブラートを練習する際に実際に使用していた曲である。比較的ゆっくりであり，曲中のフレーズがわかりやすいため練習に使用していた．今回，ヴィブラートをかける場所を明確にしたかったため，システムのテスト用として使用した．

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．

先行研究 [9] では，実際の歌唱におけるヴィブラートは周期および深さが時間的に変動する非定常な現象であることが示されている．本研究では，これらの複雑な揺らぎをそのまま再現するのではなく，初心者が理解しやすいように，周期と深さの変化を段階的かつ単純な形に抽象化したモデルを採用した．

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 メッセージが持つ tick は内部的な時間の最小単位であり，人間が直接理解できる実時間ではない．そのため，テンポ（1 拍が何マイクロ秒で演奏されるか）と ticks per beat（1 拍を何 tick で分割して表現しているか）に基づいて tick を秒へ変換することで，時系列を保持したノートの開始時刻と持続時間を実時間（秒）として算出し，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．システムを起動した際の Python ターミナル画面を**図 3** に示す．図 3 では、1 拍より長い音を抽出するよう入力したため、4 つのノートが抽出されている．抽出された 4 つのノートを楽譜上に示すと**図 4** の通りである．次に，ユーザは抽出されたノートの中からヴィブラートを適用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．今回は 0 のノート（楽譜上では一つ目の赤のファの音）のみを選択し，5 の「定速で一定」を適用した．ヴィブラートの形状を指定後, 深さの最大値および周期の最大値を入力することで，制御信号の生成が開始される．今回は変化をわかりやすくするため深さを 100 セント，周期を 8Hz に設定した．周期は [11], [12] を元に一般的に 5〜7Hz 程度であることを考慮し，やや速めに設定した．

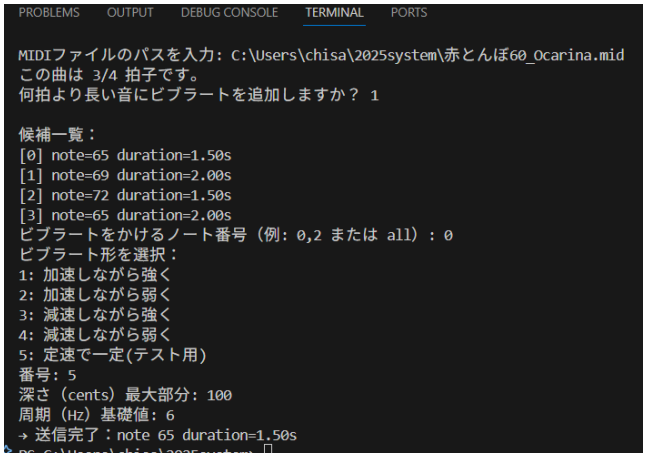

図 3: 実行時の Python ターミナル画面

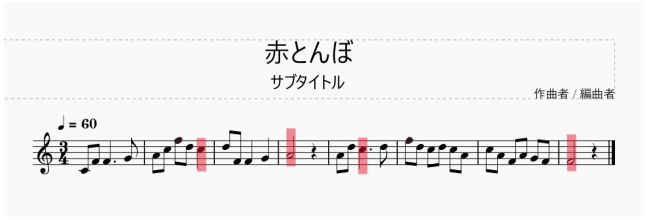

図 4: テスト時の抽出ノート

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

Max8 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．UDP は確認応答を行わない軽量な通信方式であるため低遅延で高頻度のデータ受信に適しており，ビブラートのような大量の時系列制御信号をリアルタイムで扱うことができる．受信したメッセージはノート情報と音高変動量に分岐して処理され，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で確認できた．確認できた波形は**図 6**，**図 7** に示す通りである．横軸が時間，縦軸が音高変動量を示している．なお，波形は右側から左側に

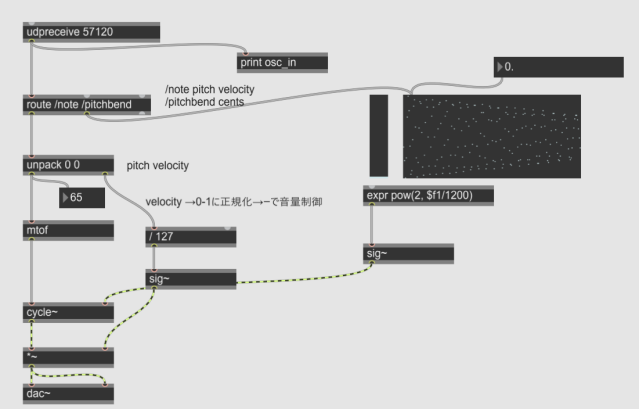

図 5: Max 8 構成

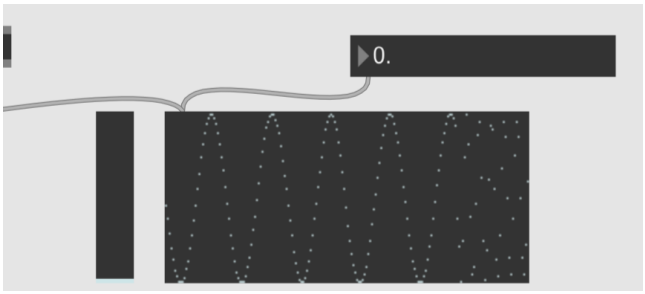

図 6: ヴィブラート波形の可視化例（定速で一定（テスト用））

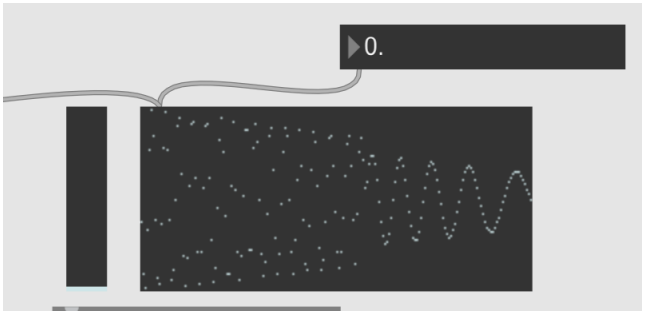

図 7: ヴィブラート波形の可視化例（加速しながら強く））

時間が進んでいる．

4.5 評価

本システムの評価は，主に以下の 2 点に焦点を当てて行う．一つ目はヴィブラート波形を生成するまでの操作性である．二つ目は生成されたヴィブラート波形が意図したとおりに表示できたかである．システムの操作性に関しては，Python ターミナル上でのノート選択，ヴィブラート形状の指定まで一連の流れが直感的に理解できるかを確認した．また，生成されたヴィブラート波形に関しては，Python 側で指定した形状と Max 8 側で可視化された波形が一致しているかを確認した．

評価はゼミ生に行ってもらった．評価者は吹奏楽経験のあり楽譜が読めるがヴィブラートの経験は少ない者 2 名である．システムの起動,MIDI ファイルの選択まで筆者が行った．その後は,Python の指示により進めてもらい, 筆者は質問された際のみ説明を行った．

評価の結果, 以下の意見が得られた．

まず, ビブラートの深さおよび周期に関する説明がコンソール内に明示されていると, 操作内容の理解がより容易になると指摘された. また,Max 上で表示される波形表示について, 縦軸が音高、横軸が時間を示していることを口頭説明だけでなく, 文字情報として併記することで, 視覚的理解が向上するとの意見があった. さらに, ビブラートを付与する音符の選択において,「all」を選択した場合, 本実験条件では一度の波形選択で全ての音符（4 音）に適用されると誤解される可能性がある点が指摘された. 加えて, 現在は対象となる音符がテキストで表示されているが, 楽譜表示ソフト（MuseScore 等）を併用し, 楽譜を視覚的に確認しながら操作できると, より直感的な操作が可能になるとの意見が得られた. 一方で, ビブラートの生成および表示については, 意図したとおりに動作しており, 表現結果は適切であったとの評価が得られた．

これらの意見と筆者自身が説明を行う際に改善の余地があると感じた点を踏まえ，以下の改善点を挙げる．

・ビブラートの深さや周期などの専門用語について, 事前知識のないユーザにも理解できるよう, 用語説明をシステム内（コンソール表示やマニュアル）に明示する必要がある．

・操作中に複数のウィンドウを同時に使用する構成となっているため, 操作の各段階において「どのウィンドウを参照すべきか」が分かりにくいという課題が確認された．

・ビブラートを付与する音符の選択において,「all」を選択した際の適用範囲が直感的に理解しにくく, 誤解を招く可能性があるため, 表示方法や説明の改善が必要である．

・対象となる音符がテキスト情報のみで提示されているため, 楽譜表示ソフト（MuseScore 等）と連携し, 楽譜を確認しながら操作できるインタフェースが望まれる．

・波形表示図の周囲に, 各要素の意味を示す注釈や説明文がないため, 視覚的に理解しにくいとの指摘があった．

図 8 と**図 9** は改善点をもとに修正したシステムの Python ターミナル画面と Max 8 の波形図周辺画面である．用語や選択範囲等の説明の追加を行い，ユーザが操作内容を理解しやすさの改善や視覚的理解の向上を図った．図 8 では, ビブラートの深さや周期などの専門用語について, 用語説明をシステム内に明示するようにした. また, 例を挙げることで, 初心者にも理解しやすいよう工夫した．例の数値は [11], [12] を参考にした. 図 9 では, 波形表示図の周囲に, 各要素の意味を示す注釈や説明文を追加した．

なお，本システムの評価は限定的な条件下で行われたため，今後はより多様なユーザを対象とした評価を実施し，システムの汎用性や操作性を検証する必要がある．

```
PS C:\Users\chisa\2025system> python syuuseivibsys.py
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

候補一覧：
[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
選択したノートの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)
適用したい形の番号を入力してください: 5
ヴィブラートのパラメータを設定してください。
例）深さ=50、周期=6
深さ：ヴィブラートの音高の変化幅
   例）深さ=20 → ±20 （半音の1/5）
       深さ=50 → ±50 （半音の1/2）
       深さ=100 → ±100 （半音1つ分）
周期：ヴィブラートの速さ
    例）4 Hz → 比較的遅いビブラート
        6 Hz → 一般的なビブラート
        8 Hz → 比較的速いビブラート

深さ（cents）最大部分: １００
周期（Hz）基礎値: ６
ビブラートを送信中
max8で図示されますので確認してください。
→ 送信完了：note 65 duration=1.50s
```

図 8: 修正した Python ターミナル画面

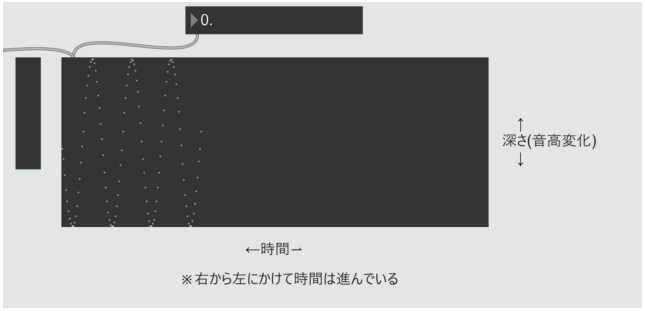

図 9: 修正した Max ８波形図周辺画面

4.6 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．

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できることが可能になると期待出来る．

さらに，対象音符の提示方法を改善し，テキスト表示に加えて楽譜表示ソフト（MuseScore 等）と連携することで，楽譜を確認しながら操作できる構成とすることも重要な課題である．これにより，ユーザは音符とヴィブラート処理の対応関係を直感的に把握できるようになると考えられる．

加えて，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. おわりに

本研究では，ヴィブラートという音楽的表現について，ニューグロー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，音高を中心とした周期的な微小変動によって生じる現象であり，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考える．

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

[1] : ニューグローヴ世界音楽大事典.
[2] 莱孝之: 音楽創作とコンピュータ音楽，オペレーションズ・リサーチ学会誌，Vol. 54, No. 9, pp. 563–569 (2009 年 9 月). https://orsj.org/wp-content/corsj/or54-9/or54_9_563.pdf.
[3] 田中範康: エレクトロニクス技術、特に電子音が音楽作品に 及ぼした影響について - 作曲家の立場から -，名古屋芸術大学研究紀要， Vol. 第 37 巻, pp. 197–212 (2016). https://x.gd/DnFM6.
[4] 鴨川仁, 山本訓久, 番田清美, 稲崎弘次, 藤原博伸, 荒川悦雄, 織原義明, 中村真帆, 佐藤良衛, 阪井陸真, 高崎瑞希: 電子楽器「テルミン」による音楽・理科・ものづくり教育, 東京学芸大学紀要. 芸術・スポーツ科学系， Vol. 65, pp. 15–23 (2013). https://u-gakugei.repo.nii.ac.jp/records/31407.
[5] 剣持 秀紀: 歌声合成技術 VOCALOID™ と新しい音楽，*JAS Journal Vol.54 No.2*, No. 3 月号. https://www.jas-audio.or.jp/journal-pdf/2014/03/201403_007-012.pdf.
[6] 清水祐樹: 電子楽器「マグテルミン」の製作と電磁気実験への活用，物理教育， Vol. 56, No. 4, pp. 268–271 (2008).
[7] 堂前幸司，秋田純一，小松孝徳: 音色の印象語に基づく減算合成型シンセサイザの パラメータ操作手法の検討，HAI シンポジウム 2010，pp. 1–6 (2010). https://hai-conference.net/proceedings/HAI2010/html/paper/paper-3d-1.html.
[8] 田中瑞穂，竹川佳成，平田圭二: 合成音声歌唱におけるヴィブラートのパラメータ自動推定，研究報告音楽情報科学（MUS）， Vol. 2020-MUS-126, No. 14, pp. 1–8 (2020). https://ipsj.ixsq.nii.ac.jp/records/203127.
[9] 中野倫靖，後藤真孝，平賀譲: 楽譜情報を用いない歌唱力自動評価手法，情報処理学会論文誌， Vol. 48, No. 1, pp. 227–236 (2007). https://ipsj.ixsq.nii.ac.jp/records/10100.
[10] 宮崎嵩大，森勢将雅: 音声波形からのヴィブラートパラメータ推定の高精度化と評価，研究報告音楽情報科学（MUS）， Vol. 2019-MUS-123, No. 12, pp. 1–7 (2019). https://ipsj.ixsq.nii.ac.jp/records/197797.
[11] Seashore, C. E.: *Psychology of Music*, McGraw-Hill (1938).
[12] Sundberg, J.: *The Science of the Singing Voice*, Northern Illinois Univ Pr, Dekalb, Ill (1987).