

# 楽譜上のフレーズ指定に基づく 演奏表情のパラメータ制御システム

福知山公立大学 情報学部情報学科

32245043 白波瀬桃花

指導教員 橋田光代 準教授

提出日 2026年1月30日

改訂日 2026年2月7日

1	はじめに	1
2	関連研究	1
2.1	既存の音楽制作環境	1
2.1.1	DAW (Digital Audio Workstation)	
2.1.2	楽譜作成ソフトウェア	
2.1.3	高品質な AI 演奏生成ツール	
2.2	保科理論：音楽構造とエネルギー運動	2
2.2.1	音楽構造の階層モデル：グループとフレーズ	
2.2.2	頂点（重心）	
2.3	Mixtract：演奏表現のデザイン環境	3
2.4	フレーズ頂点の推定手法	3
2.4.1	グループと頂点の推定	
2.4.2	スラー境界に基づく頂点推定	
3	演奏表現支援システムの設計	4
3.1	設計の基本方針	4
3.2	演奏制御パラメータの設計	4
3.2.1	Expression (CC2) による強弱制御	
3.2.2	Onset による発音時刻の制御	
3.3	発想標語プリセットの作成	5
3.4	頂点推定ルールの選定	5
4	システムの実装	5
4.1	システム概要	5
4.2	開発環境と使用したライブラリ	6
4.3	実装の詳細	6
4.3.1	楽譜データと演奏データの対応付け	
4.3.2	楽譜表示と表情指示の反映	
4.3.3	頂点候補の計算	
4.3.4	MIDI データの加工処理	
4.4	UI と操作手順	7
5	評価	8
5.1	適用事例	8
5.1.1	同一フレーズに対するパラメータ変化の検証	
5.1.2	楽曲全体を通じた演奏デザインの実践	
5.2	頂点推定（和声情報の有無による影響）	10
5.3	演奏音源に対する表情付けの定性的な再構築	10
6	評価	10
6.1	発想標語と演奏パラメータの対応付け	10
6.2	頂点推定機能の評価と課題	10
6.3	持続音楽器を考慮した階層的フレーズ表現	11

## — 図目次 ——

1	DAW の画面例 (Cubase)	1
2	楽譜ソフトウェアの画面例 (Dorico)	2
3	AI 演奏生成ツールの画面例 (ACE Studio)	2
4	保科理論におけるフレーズ構造 (文献 [1] より抜粋)	3
5	演奏表現デザイン環境「Mixtract」(文献 [2] より抜粋)	3
6	システムの全体構成と処理の流れ	6
7	システム画面 (左: 操作パネル, 右: 楽譜表示エリア)	7
8	フレーズ範囲と頂点候補の表示	8
9	パラメータ調整パネル (左部分)	8
10	プリセットを楽譜に反映	8
11	対象フレーズ (15 小節～18 小節)	8
12	対象フレーズの範囲指定と頂点設定	8
13	適用前の CC2 値の推移	8
14	Cantabile 適用後の CC2 値の推移	8
15	Maestoso 適用後の CC2 値の推移	9
16	Con brio 適用後の CC2 値の推移	9
17	発想標語による Onset (発音時刻) の比較	9
18	「G 線上のアリア」に対する演奏表現の適用	10

## — 表目次 ——

1	実装した発想標語プリセットとパラメータ設定値	5
2	本システムで採用した頂点推定ルールと評価ポイント	5

## 1. はじめに

音楽の演奏において、聴き手に豊かな感動を届けるためには、楽譜に記された音符や記号を正確に再現するだけでは不十分である。演奏者自身が楽曲を深く解釈し、強弱（ダイナミクス）やテンポ（アゴーギク）に繊細な変化を加えることで、聴き手に感動を与える表現が生まれる。この演奏表現を探求する過程において、近年のデジタルツールは強力な支援となりうるが、既存のツールには演奏者の視点から見ていくつかの課題が存在する。

現在の音楽制作において主流である DAW (Digital Audio Workstation) [3] は、ピアノロール画面を用いて音量やタイミングを数値レベルで緻密に編集できる強力なツールである。しかし、そのインターフェースはトラックベースの音響制作に最適化されており、普段から五線譜に慣れ親しんだ吹奏楽やオーケストラの演奏者が、楽譜全体の構造やフレーズのまとまり、和声の流れといった音楽的文脈を直感的に読み取りながら表現を編集するには向きである。

一方で、Dorico[4] や Sibelius[5] に代表される楽譜ソフトウェアは、近年の製品では高品質なサンプリング音源を用いた再生も可能となっている。しかし、主な目的はあくまで正確な記譜にある。そのため、「Cantabile (歌うように)」や「Dolce (甘く)」といった演奏指示（発想標語）が再生音に与える影響は、ソフトウェアにあらかじめ定義された標準的な解釈が適用されることが多い。もし演奏者が、フレーズの頂点の位置や抑揚のカーブなどを自身の解釈に合わせ、フレーズの頂点や抑揚のカーブを微調整しようとしても、その操作は直感的な五線譜上では完結しない。結果として、ピアノロール画面やグラフエディタを用いたパラメータの編集作業が求められ、フレーズに対して複数の表現を比較・検討するといった創造的な試行錯誤を支援する機能は十分とはいえない。

このように、演奏者が慣れ親しんだ「五線譜」というインターフェースの上で、DAW のように柔軟かつ具体的な音響編集を直感的に行えるツールは不足しており、両者の間には明確なギャップが存在する。筆者自身も吹奏楽の演奏経験の中で、楽譜に記された演奏指示の解釈に迷い、それが具体的にどのような音の変化として表現されるのかを試行錯誤する際に、適切な支援ツールが存在しないことに課題を感じてきた。

そこで本研究では、楽譜ソフトウェアの持つ優れた視認性（五線譜インターフェース）と、DAW の持つ柔軟な音響編集能力の長所を組み合わせることを目的とし、Web ブラウザ上で動作する「楽譜上のフレーズに基づく演奏表情のパラメータ制御システム」を構築した。

本研究の主な対象は、DAW よりもパート譜での練習を中心となる管楽器奏者や、演奏表現を深く探求したいと考えている学生・アマチュア奏者である。本システムは、抽



図 1: DAW の画面例 (Cubase)

象的な発想標語が音量やテンポといった物理的な演奏パラメータとしてどう表現されるのかを可視化・聴取可能にすることで、演奏者が自身の音楽的解釈を具体的かつ主体的に構築するプロセスを支援するものである。

## 2. 関連研究

本章では、既存の音楽制作環境や楽譜ソフトウェア、および演奏表情付けに関する先行研究について整理し、それらの課題と本研究の位置づけについて述べる。

### 2.1 既存の音楽制作環境

#### 2.1.1 DAW (Digital Audio Workstation)

PC を用いた音楽制作において、最も一般的かつ高機能なツールとして DAW (Digital Audio Workstation) [3] が挙げられる。代表的なソフトウェアには、Cubase[6] や Logic Pro[7] などがある。これらのソフトウェアでは、MIDI データの音の高さ、長さ、強弱 (Velocity)，演奏の表情や抑揚に関する Control Change (CC) などのパラメータを数値レベルで緻密に編集することが可能である。

しかし、その編集インターフェースの多くは「ピアノロール」と呼ばれる、縦軸に音高、横軸に時間をとったグラフ形式が採用されている（図 1）。これは、音響的な編集作業には適しているが、五線譜に慣れ親しんだ演奏者にとっては、フレーズのまとまりや和声の流れといった音楽的な文脈を直感的に把握しにくいという課題がある。また、クリッションドやデクリッションドといった抑揚をつけるためには、オートメーションレーンと呼ばれる領域にマウスで曲線を書き込む操作が必要となり、演奏表現の探求においては直感的とは言い難い。

#### 2.1.2 楽譜作成ソフトウェア

五線譜を用いた記譜に特化したツールとして、Mus-eScore[8] や Dorico[4], Sibelius[5] などが広く普及している。これらのソフトウェアは、綺麗な楽譜を作成することを主な目的としているが、演奏表現の再生機能に関しては、本研究の目的である「演奏者が主体的に表現を探求する」という観点から見ると以下の課題がある。

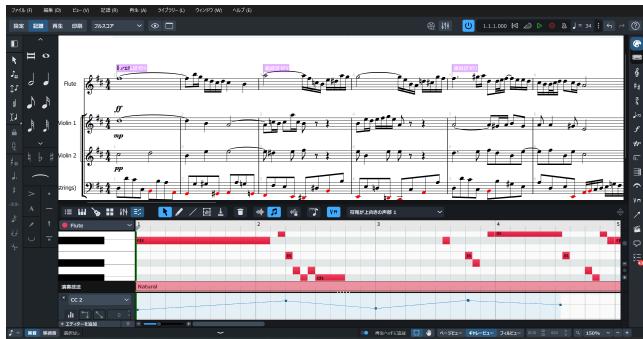


図 2: 楽譜ソフトウェアの画面例 (Dorico)

まず挙げられるのが、楽譜上の発想標語が十分に音に反映されない点である。例えば、最新の MuseScore 4においては、高品質な音源 (MuseSounds) が搭載されているものの、楽譜上に「Cantabile」などの発想標語をテキストとして配置しても、それはあくまで視覚的な文字情報として扱われ、再生される音には変化がないことが多い。音量やテンポに変化を与えるためには、クレッシェンドなどの強弱記号を 1 つずつ配置するか、個々の音符の設定画面を開いて数値を手動で入力する必要があり、直感的に様々な表現を試すことは難しい。

また、高機能なソフトであっても、操作が複雑で五線譜の良さが活かされていない場合が多い。

Dorico などのソフトウェアでは、楽譜の下部に「キー エディタ」と呼ばれるピアノロール画面を表示し、そこで DAW と同様にマウスで曲線を書き込むことで詳細な表現付けが可能となっている(図 2)。しかし、これは実質的に五線譜のインターフェースから離れた操作であり、DAW における描画作業と本質的に変わらない。

さらに、処理の中身が分かりにくく、細かい指定ができない点も課題である。Sibelius には、「Espressivo 機能」や「Rubato 機能」といった再生機能が存在し、これらを使うことで人間らしい揺らぎを加えることができる。しかし、これらの機能は楽曲全体に対して一律に適用されるため、「この特定のフレーズだけを溜めて演奏したい」といった細かい意図を反映させることは難しい。また、それらの設定によって具体的に音量が上がったのか、テンポが遅くなつたのかといった中身がユーザには見えないため、演奏者がパラメータと聴こえ方の関係を学ぶことは困難である。

このように、五線譜という見やすいインターフェースを維持したまま、フレーズ単位で直感的かつ中身の見えるパラメータ制御を行える環境は、既存の楽譜ソフトウェアでは実現されていない。

### 2.1.3 高品質な AI 演奏生成ツール

近年では、AI を用いて楽譜情報から人間らしい演奏を自動生成する技術も実用化されている。

Note Performer[9] は、Sibelius や Dorico などの楽譜作成ソフトウェアに組み込んで使用する AI 演奏エンジンであ



図 3: AI 演奏生成ツールの画面例 (ACE Studio)

る。これは、楽譜上の音符や基本的な記号を読み取り、前後の文脈を考慮して、音の立ち上がりや減衰、拍の強弱 (Velocity) などを自動的に補正する機能を持つ。これにより、ユーザが細かい調整を行わなくても機械的ではない自然な演奏を生成できる点は大きな利点である。しかし、その処理プロセスはブラックボックス化されており、ユーザが特定のフレーズだけを極端に強調したいなどの発想標語を用いた独自の解釈を反映させようとしても、AI の自動補正が優先され、意図通りの変化が得られない場合がある。

ACE Studio[10] は、AI を用いた歌声合成技術を楽器の演奏生成に応用した音楽制作プラットフォームである。ACE Studio の AI バイオリンは、MIDI データを入力するだけで、AI が最適なアーティキュレーション (奏法) や感情表現を自動的に付加し、高品質でリアルな演奏を生成することができる。例えば、MIDI ノートの長さを短くするだけでスタッカート気味の奏法に変化するなど、AI による解釈の精度は非常に高い。

しかし、これらはあくまで「自動化」による品質向上に重点が置かれている。現状では、Expression などの細かいニュアンスを手動で制御する機能は限定的であり、演奏者が意図的に「このフレーズの頂点をズラしたい」といった繊細な表現の試行錯誤を行うことは困難である。

また、その操作インターフェースは図 3 のようなピアノロール形式であり、五線譜を用いて音楽構造を捉えながら表現を探求する本研究のアプローチとは、対象とするユーザ層や利用目的が異なるといえる。

これらの技術は最終的な成果物のクオリティを効率よく高める点では非常に優れている。しかし、生成プロセスがブラックボックス化されているため、演奏者がどのようなパラメータ操作が音の変化に繋がるのかを理解することは難しい。そのため、ユーザは AI が生成した演奏を受動的に受け入れる形での利用に留まりやすい。

## 2.2 保科理論：音楽構造とエネルギー運動

本研究では、システムにおける演奏表現の考え方として、保科洋による「保科理論」[1] を参考にした。保科理論は、楽譜上の音符の並びを、物理的なエネルギーの動きに例えて解説したものである。以下に、本システムの設計において特に重要となる「音楽構造の階層化」と「頂点（重心）

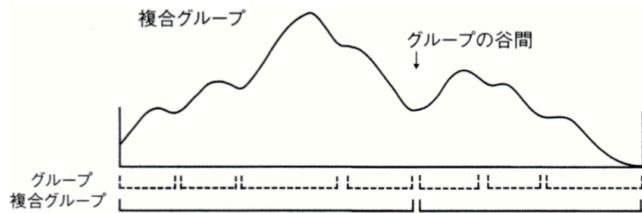


図 4: 保科理論におけるフレーズ構造（文献 [1] より抜粋）

の概念」について述べる。

### 2.2.1 音楽構造の階層モデル：グループとフレーズ

保科理論では、楽譜上の音符の羅列を、エネルギーのまとまりとして階層的に捉える（図 4）。その最小単位がグループである。グループとは、複数の音を 1 つの意味のあるまとまりとして捉えたものであり、言語における「単語」に相当する。楽譜上の音符を適切にグルーピング（分節化）することで、演奏に意味が生まれる。

さらに、複数のグループが連なることで、より大きなまとまりである「フレーズ」が形成される。これは言語における「文章」に相当する。楽曲は、図 4 のように、小さなグループが集まって大きなフレーズを作るという階層構造になっている。

### 2.2.2 頂点（重心）

保科理論では、「独立したひとつのグループには、必ず 1 つの頂点が存在する」と定義している。頂点（重心）とは、グループの中で最もエネルギーが集中する音符のことを指す。演奏表現は、この頂点を中心として以下のような山型のカーブを描くことで生成される。

- (1) アナクルーズ グループの開始点から頂点に向かって、エネルギーを徐々に高めていく過程（クレッシャンド）
- (2) 頂点 エネルギーが最大化する点
- (3) デジナス 頂点を越えた後、エネルギーを減衰させていく過程（ディミヌエンド）

どの音が頂点となるかは、音符の並び（旋律線）、和声進行、拍節（強拍・弱拍）、アーティキュレーションなどの音楽的要素によって決定される。本研究では、「グループごとに 1 つの頂点を持ち、そこに向かって強弱が変化する」という原則を、システムの制御モデル（Expression 値の線形補間）として採用している。

### 2.3 Mixtract：演奏表現のデザイン環境

橋田らは、前節で述べた保科理論の考え方を計算機上で実現し、ユーザが演奏表現を自由に設計できるシステム「Mixtract」を提案している [2]。

図 5 に Mixtract の概要を示す。

このシステムの特徴は、AI が勝手に全ての演奏を決めてしまうのではなく、ユーザ自身が「どこがフレーズか」「どこが頂点か」を決定し、それに基づいて演奏を生成する点にある。

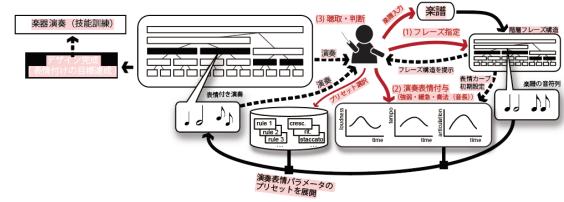


図 5: 演奏表現デザイン環境「Mixtract」（文献 [2] より抜粋）

具体的な処理の流れは、システムが認知的音楽理論である GTTM[11] を用いて、楽曲のフレーズ構造（階層構造）を自動解析して提示する。同時に、保科理論に基づきフレーズ内での頂点（重心）を自動推定する。Mixtract では、各音符が持つエネルギー値を算出し、それを音量やテンポなどの表現要素に変換することで、保科理論における山型の表現を実現している。

ユーザは提示された構造や頂点を確認し、必要に応じて自分の解釈に合わせてフレーズの範囲や階層を修正する。決定されたフレーズに対し、「テンポ（速さ）」「ダイナミクス（強弱）」などの表情カーブを割り当てることで、抑揚ある演奏が生成される。ユーザは生成された演奏を聴取り、納得がいかなければ再度設定を調整する。この「設定→聴取→修正」というサイクルを繰り返すことで、ユーザの頭の中にあるイメージを具体的な演奏として再現することができる。

ユーザが主体となって試行錯誤するという設計思想は、本研究が目指すシステムにおいて非常に重要な指針となっている。しかし、Mixtract における強弱の制御は、主に Velocity (打鍵の強さ) を対象として実装されている。これはピアノなどの打鍵楽器には有効であるが、管楽器のように音を伸ばしている最中に音量を膨らませること（Expression 制御）が重要な楽器に対しては、その表現力を十分に活かしきれないという課題がある。本研究では、Mixtract のユーザ主導のデザインおよびエネルギー値（頂点）に基づく表現生成という方向性を継承しつつ、制御対象を Velocity ではなく Expression (CC2) に置き換えることで、管楽器に適した演奏表現システムを構築する。

### 2.4 フレーズ頂点の推定手法

保科理論を計算機システム上で実現するためには、システムが楽譜の中から自動的に頂点（重心）を特定する、あるいはユーザの頂点決定を支援する仕組みが必要となる。これに対し、橋田らは聴取実験に基づいたグループと頂点の推定モデル [12] や、スラー境界情報に基づく手法 [13] を提案している。

#### 2.4.1 グループと頂点の推定

橋田らのグループと頂点の推定に関する研究 [12] では、ベートーヴェンのピアノソナタ「悲愴」などを題材に、人

間の演奏家がどこを頂点として捉えているかを分析し、頂点となりやすい音符の条件をルール化している。提案されたモデルは、以下の音楽的特徴を持つ音符に対し、頂点らしさを表すポイント（スコア）を加算していく「投票（Voting）方式」を採用している。

- (1) 音高 フレーズ内で相対的に音高が高い音は、エネルギーが高く頂点になりやすい。
- (2) 音価 周囲の音符よりも長い音価を持つ音は、強調される傾向にある。
- (3) 跳躍進行 音程が大きく跳躍した（特に上行跳躍した）到達音は、強いエネルギーを持つ。
- (4) 和声的緊張感 非和声音（倚音や掛留音）や、和声進行上で緊張度の高い音（ドミナントなど）は頂点となる可能性が高い。
- (5) 特定の音型 「ジェットコースター音型」と呼ばれる、下行してから上行するような旋律の変曲点が頂点となる場合がある。

#### 2.4.2 スラー境界に基づく頂点推定

橋田らのスラー境界に基づく頂点推定の研究 [13] では、単純な音高や音価の比較だけでなく、特定の音型パターンがグルーピングや頂点位置に与える影響についても論じられている。その代表例が「バウンド分割」である。

バウンド分割とは、1拍以下の短い音価に分割された音群の後に、長い音価の音が続くパターンのことを指す。一般的に、短い音符の連続の後に現れる長い音符はエネルギーを受け止めやすく、頂点になりやすい性質がある。しかし、短い音群が拍節の強拍（小節の頭など）に現れる場合は、その短い音群自体にアクセントが生じ、頂点となる場合がある。先行研究では、このような「バウンド分割」や前述の「ジェットコースター音型（下行後の上行）」といった特徴的なパターンを検出することで、頂点の判定精度を向上させている。

これらの先行研究は、和声分析やスラー情報を含む多角的な情報を用いることで、高い精度（約 80 % 以上）での頂点推定を実現している。しかし、これらのモデルをそのまま管楽器のパート譜（单旋律）に適用しようとした場合、伴奏パートの和声情報が欠落していたり、計算が複雑になりすぎたりするという課題がある。これらの背景から本研究では、グルーピング（フレーズ範囲の決定）はユーザの音楽的解釈を尊重し直感的な操作に委ねる一方、範囲内の頂点推定においては、单旋律からでも判断可能な「バウンド分割」などの音型ルールを限定的に採用するアプローチをとる。

### 3. 演奏表現支援システムの設計

本章では、第 2 章で述べた課題を解決し、管楽器奏者が五線譜上で直感的に演奏表現を探求できるシステムの設計方針について述べる。

#### 3.1 設計の基本方針

本システムは、以下の 3 点を基本方針として設計する。

##### (1) 五線譜インターフェースでの操作

DAW のようなピアノロールではなく、演奏者が日常的に使用する五線譜を操作の基盤とする。また、楽譜作成ソフトウェアのような見やすさを保つつつ、演奏表現の編集に特化したシンプルなインターフェースを提供する。

##### (2) フレーズ単位でのパラメータ制御

音符 1 つ 1 つを編集するのではなく、音楽的な意味のまとまりである「フレーズ」を選択し、それに対して一括で表現を適用する方式をとる。これにより、音楽的な文脈を保ったままの試行錯誤が可能となる。

##### (3) 聴き比べによるフィードバック

選択した発想標語やパラメータ設定が、実際の音にどのように反映されるかを即座に聴取できる環境を構築する。数値の変化だけではなく、聴覚的な変化を確認することで、演奏者の理解を深める。

#### 3.2 演奏制御パラメータの設計

本システムでは、演奏表現を構成する要素として、強弱と時間の 2 点に着目する。特に管楽器や弦楽器のような持続音を制御できる楽器の特性を考慮し、以下のパラメータを制御対象とする。

##### 3.2.1 Expression (CC2) による強弱制御

ピアノなどの打鍵楽器では、音の強さは打鍵の瞬間の速度 (Velocity) で決定され、発音後に音量を変化させることはできない。しかし、管楽器は息の圧力（ブレスコントロール）によって、音の持続中に音量を自由に増減させることができる。実際、浅野による先行研究では、この持続音の特性に着目し、フレーズの頂点を基準として Expression 値を線形補間する手法が提案されている [14]。第 2 章で述べた保科理論における頂点に向かうクレッセンドや、頂点を越えた後のディミヌエンドを表現するためには、Velocity だけでは不十分である。MIDI のコントロールチェンジの 2 番（以下、CC2 とする）は、拍よりさらに細かい Tick 単位で音量の制御が可能である。

そのため、本システムでは CC2 を用いて、フレーズ内の音量を連続的に変化させる（線形補間する）手法を採用する。

##### 3.2.2 Onset による発音時刻の制御

発想標語には、強弱だけでなく速度の変化も含まれる。例えば、「Maestoso（堂々と）」であれば、単に音が大きいだけでなく、1 音 1 音を十分に保って演奏するような、わずかな遅れ（溜め）が生じることが一般的である。本システムでは、BPM（楽曲全体のテンポ）を変更するのではなく、各音符の発音時刻（Onset）をミリ秒単位で前後にずらすことで、局所的なテンポの揺らぎを表現するものとする。

表 1: 実装した発想標語プリセットとパラメータ設定値

発想標語	意味	Base CC2	Peak CC2	Onset (ms)
Cantabile	歌うように	+15	+35	+20
Dolce	甘く、柔らかく	-25	+10	+15
Maestoso	莊厳に、堂々と	+20	+50	+40
Appassionato	情熱的に	+25	+60	-30
Con brio	生き生きと	+15	+40	-40
Leggiero	軽く、軽快に	-20	+5	-30
Tranquillo	静かに、穏やかに	-35	+5	+30
Risoluto	決然と、きっぱりと	+20	+45	0
Sostenuto	音を十分に保って	+10	+20	+50
Marcato	はっきりと	+15	+65	0

### 3.3 発想標語プリセットの作成

ユーザがパラメータを 1 から設定する負担を軽減するため、代表的な発想標語に対応するパラメータの組み合わせをプリセットとして実装した。

プリセットの値 (CC2 の増減量, Onset の遅延量) の決定にあたっては、楽器経験のある学生 4 名の協力を得て予備調査を行った。調査では、被験者に本システムを使用させ、各発想標語のイメージに合致するパラメータ値を初期値として採用した。これにより、個人の主観に偏り過ぎない、ある程度客観的な妥当性のあるプリセットを提供することを目指した。

本システムでは、「Cantabile (歌うように)」といった旋律的な表現から、「Appassionato (情熱的に)」といった激しい表現、「Tranquillo (静かに)」といった穏やかな表現まで、多様な音楽的性格を比較できるよう、代表的な 10 種類の発想標語を選定し、プロトタイプとして実装した。これらのプリセットの値の決定にあたっては、上記の調査結果に基づき表 1 に示す値を定義した。

各パラメータの役割と設定の意図は以下の通りである。

**Base CC2 と Peak CC2 (強弱の制御) :** これらは、元の演奏データに含まれる CC2 値の平均値を基本とし、それに対する増減量を表す。

- **Base CC2 :** フレーズの開始点から終了点までの音量レベルを決定する。
- **Peak CC2 :** フレーズの頂点における音量レベルを決定する。

例えば、「Appassionato (情熱的に) や「Maestoso (堂々と)」は、Peak の値を Base よりも大幅に高く設定することで、頂点に向かう急激なクレッセンドを生み出し、力強い音楽表現を実現している。また、「Dolce (甘く)」や「Tranquillo (穏やかに)」は、Base と Peak の差を小さく、全体的に低い値に設定することで、起伏の少ない平穏な強弱変化としている。

**Onset (時間的な揺らぎ) :** 各音符の発音タイミングを、楽譜上の正規の位置からミリ秒単位でずらすパラメー

表 2: 本システムで採用した頂点推定ルールと評価ポイント

分類	対象音	条件	評価ポイント		
			先行音	対象音	後続音
音価	隣接する 2 音の第 1 音	後続音より短い 後続音より長い	-	0	1
	同一音価の連続する音群の第一音	-	-	1	0
音高	隣接する 2 音の第 1 音	後続音より低い 後続音より高い	-	0	1
	同一音価が連続する音群の第 2 音以降	-	発音順音符数	-	-
進行	進行到達音 (隣接 4 音の第 3 音)	1) 上行-上行-下行	0	1	0
		2) 下行-上行-下行	2	1	0
		3) 上行-下行-上行	1	2	1
		4) 下行-下行-上行	0	2	1
		5) 1)2) C 基本音長 0.25 秒以下	1	0	0
	開始音 終了音	-	-	1	-
グループ	最長音	バウンド分割を含む バウンド分割を含まない	-	1	-
	最高音	音節の中央値より長く 0.25 秒以上 その他	-	2	-
	最大の上行跳躍音	音節の中央値より長く 0.25 秒以上 その他	0	2	-

タである。これにより、音楽表現におけるアゴーギク(速度の緩急)を再現している。「Maestoso (堂々と)」のように重厚感が求められる標語には正の値(遅延)を与えて溜めを作り、「Con brio (生き生きと)」のように前進するエネルギーが必要な標語には負の値(前倒し)を与えて疾走感を表現している。

### 3.4 頂点推定ルールの選定

フレーズ内の頂点を決定する支援機能として、自動推定機能を設計する。第 2 章で述べた先行研究 [12] のモデルを参考にしつつ、本システムでは伴奏や和声情報を持たない単旋律のパート譜でも機能するよう、以下の 4 つの観点からルールを選定した。詳細なルールと配点を表 2 に示す。

- (1) **音価**: 長い音符は強調されやすい。
- (2) **音高**: 高い音符はエネルギーが高い。
- (3) **進行**: 跳躍進行の到達音や、特定の音型(バウンド分割など)に着目する。
- (4) **位置**: フレーズの開始・終了音は頂点になりにくい。

## 4. システムの実装

本章では、第 3 章の設計に基づき実装した「楽譜上のフレーズに基づく演奏表情のパラメータ制御システム」の詳細について述べる。

### 4.1 システム概要

本システムは、フロントエンドに HTML, CSS, JavaScript、バックエンドに Python (Flask[15]) を用いて、ローカル環境で動作する Web アプリケーションとして構築した。システム全体の処理の流れを図 6 に示す。

ユーザは、Web ブラウザから MusicXML (楽譜データ) と MIDI (演奏データ) をアップロードする。システムはこれらを解析し、ブラウザ上に楽譜を表示するためのデータと、音符と時間を対応付けるマッピングデータを生成する。ユーザは楽譜上でフレーズと頂点を指定し、発想標語プリセットを選択する。サーバー側では、その指示に基づ

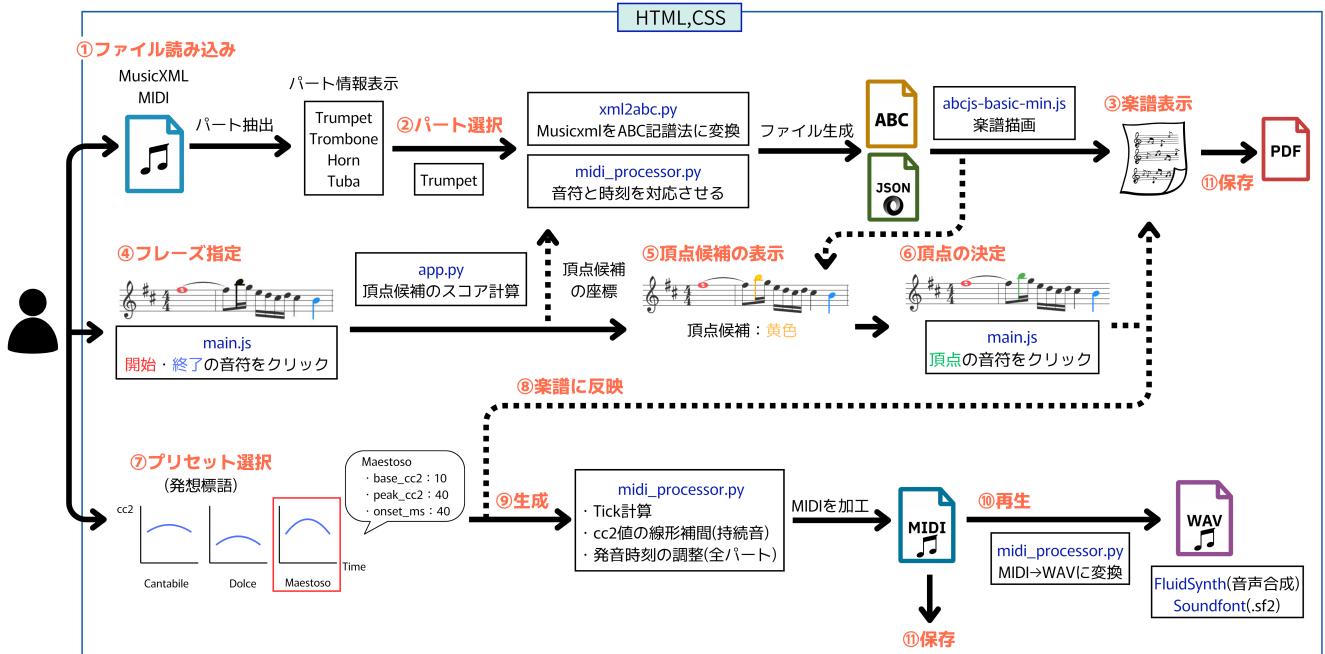


図 6: システムの全体構成と処理の流れ

いて MIDI データを加工し、FluidSynth[16] を用いて WAV 形式の音源を生成する。また、加工された MIDI データをユーザのローカル環境に保存する。

#### 4.2 開発環境と使用したライブラリ

本システムの実装に使用した主な言語・ライブラリは以下の通りである。

- 言語 : Python 3.12, JavaScript, HTML5, CSS3
- Web フレームワーク : Flask
- 楽譜解析・描画 :
  - music21[17] : MusicXML の解析、パート抽出を行う。
  - xml2abc[18] : MusicXML を ABC 記譜法に変換する。
  - abcjs[19] : ブラウザ上で ABC 記譜法を表示し、クリック操作に対応した楽譜表示を実現する。
- MIDI 处理・音声合成 :
  - mido[20] : MIDI メッセージの解析・生成・編集を行う。
  - FluidSynth : 加工した MIDI データを高品質な SoundFont を用いて WAV 音声に変換する。

#### 4.3 実装の詳細

##### 4.3.1 楽譜データと演奏データの対応付け

本システムにおける最大の技術的課題の 1 つは、楽譜上の音符 (MusicXML) と演奏データの音符 (MIDI) を正確に対応付けることである。MusicXML は小節や拍といった音楽的な単位で時間を管理するのに対し、MIDI は Tick (時間分解能) という絶対的な数値で管理するため、両者の時間軸の定義が異なる。

そこで本システムでは、サーバ側での解析処理 (app.py, midi\_processor.py) において、MusicXML のテンポ情報をもとに各音符の発音時刻を Tick に換算し、MIDI イベントと対応させる処理を実装した。具体的には、音符の index, Pitch, 発音時刻 (ms および Tick) を記録した音符対応表 (NoteMap) を JSON 形式で生成する。これにより、次項で述べるユーザ操作と内部処理の連携を実現している。

##### 4.3.2 楽譜表示と表情指示の反映

ユーザが五線譜上でフレーズを指定し、演奏表情 (発想標語) を指示するためには、ブラウザ上の GUI とシステム内部のデータ構造が同期している必要がある。本システムでは、五線譜の描画に abcjs ライブラリを採用し、以下の仕組みを実装した。

###### (1) 操作位置の特定

abcjs は、MusicXML から変換された ABC 記譜データを SVG 形式で描画する際、各音符の要素 (DOM) に対してクリックイベントを検知する機能を持つ。ユーザが画面上の任意の音符をクリックすると、システムはその音符の index 番号と Tick を取得する。この index と Tick 情報から前項で生成した NoteMap を参照することで、クリックされた箇所が「楽曲全体の何 Tick 目にあるのか」を即座に特定する仕組みを構築した。

###### (2) 表情指示の可視化

ユーザがフレーズ範囲と頂点を決定し、プリセットから発想標語 (例: Cantabile (歌うように)) を選択すると、システムはその指示を楽譜上に視覚的に反映させる。具体的には、SVG 要素を操作し、選択されたフレーズ範囲に緑色

のハイライトを、強調表示した音符（開始：赤、終了：青、頂点：緑（候補は黄色））を描画する。また、フレーズの開始位置上部に発想標語のテキストを追記する。これにより、ユーザはどの範囲に・どのような演奏表情の指示を与えたかを一目で把握することが可能となる。

### (3) 操作履歴の管理

これらの指示内容は、操作履歴としてリスト構造で管理される。各履歴には、「フレーズ範囲」「選択された頂点」「適用されたプリセットのパラメータ」が記録されており、これに基づき MIDI 加工が行われる。また、この履歴構造を持つことで、操作の取り消し (Undo) や、やり直し (Redo) といった試行錯誤を支援する機能を実現している。

#### 4.3.3 頂点候補の計算

サーバ側の `app.py` では、ユーザが指定したフレーズ範囲内の全ての音符に対し、第 3 章で定めた頂点推定ルール（表 2）に基づき、スコア計算を行う。各ルール（音価の長さ、跳躍の有無など）に合致するごとに所定のポイントを加算し、最終的にスコアが最も高い音符を「頂点候補」として抽出する。抽出された候補はフロントエンドに送信され、画面上の楽譜において該当する音符が強調表示される。

#### 4.3.4 MIDI データの加工処理

確定したフレーズ範囲と頂点情報、および選択されたプリセットに基づき、MIDI データを加工する。

##### (1) Expression (CC2) の線形補間

本システムでは、フレーズの頂点に向かって音量を増加させ、頂点通過後は音量を減少させることで、自然な抑揚を表現する。現在の時刻 (Tick) を  $t$ 、フレーズの開始時刻を  $t_{start}$ 、頂点の時刻を  $t_{peak}$ 、終了時刻を  $t_{end}$  とする。また、それぞれの時点での CC2 目標値（音量）を  $V_{base}$ （開始・終了値）、 $V_{peak}$ （頂点値）とする。

このとき、時刻  $t$  における CC2 値  $E(t)$  は、以下の区分線形関数として定義される。

$$E(t) = \begin{cases} V_{base} + (V_{peak} - V_{base}) \cdot \frac{t - t_{start}}{t_{peak} - t_{start}} & (t_{start} \leq t \leq t_{peak}) \\ V_{peak} + (V_{base} - V_{peak}) \cdot \frac{t - t_{peak}}{t_{end} - t_{peak}} & (t_{peak} < t \leq t_{end}) \end{cases} \quad (1)$$

ここで、目標値  $V_{base}$  および  $V_{peak}$  は、元の演奏データから算出されたフレーズ範囲の平均 CC2 値  $\overline{CC_{orig}}$  と、選択された発想標語プリセットのパラメータ ( $P_{base}, P_{peak}$ ) を用いて次の式のように決定される。

$$\begin{aligned} V_{base} &= \overline{CC_{orig}} + P_{base} \\ V_{peak} &= \overline{CC_{orig}} + P_{peak} \end{aligned} \quad (2)$$

なお、算出された  $E(t)$  は MIDI 規格に基づき、 $0 \leq E(t) \leq 127$  の範囲に制限される。この計算を 1 Tick 単位で行うこ



図 7: システム画面（左：操作パネル、右：楽譜表示エリア）

とで、滑らかな強弱変化を実現している。

##### (2) Onset の調整

各音符の発音時刻 (Onset) に対し、プリセットのパラメータ (`onset(ms)`) に応じたオフセットを加算する。本システムでは、単純に一律の遅延を与えるのではなく、フレーズの進行（経過拍数）に比例して変化量を増大させる線形モデルを採用している。

元の発音時刻 (Tick) を  $t$ 、補正後の発音時刻を  $t'$  とし、フレーズ開始点からの経過拍数を  $\Delta B(t)$  とするとき、制御式は以下のように定義される。

$$t' = t + \alpha \cdot P_{onset} \cdot \Delta B(t) \quad (3)$$

ここで、 $P_{onset}$  は 1 拍あたりの変化量 (`onset(ms)` に対応) を表し、 $\alpha$  はミリ秒から Tick への変換係数である。この式により、フレーズが進むにつれて遅延量（または前倒し量）が蓄積されるため、 $P_{onset}$  が正の場合は「重厚感（溜め）」、負の場合は「切迫感（走り）」といった効果が強調される。そのため、楽曲データのテンポ設定 (BPM) 自体を変更せずに、体感上ではテンポが揺らいでいるかのような表現を実現している。

#### 4.4 UI と操作手順

ユーザが直感的に試行錯誤できるように、画面構成は左側の「操作パネル」と右側の「楽譜表示エリア」に分割した（図 7）。

本システムの操作手順を以下に示す。

##### (1) ファイルの読み込みとパート選択

操作パネルから MusicXML と MIDI ファイルをアップロードし、編集したいパート（例：Trumpet）を選択すると、右側に五線譜が描画される。この際、原曲の再生も可能である。

##### (2) フレーズと頂点の指定

楽譜表示エリアにて、ユーザはフレーズの「開始音」と「終了音」を順にクリックする。すると、4.3.2 項で述べた仕組みにより、システムが推定した頂点候補が黄色でハイライトされる（図 8）。ユーザは候補の中から自身の解釈に合う音符（あるいは任意の音符）をク



図 8: フレーズ範囲と頂点候補の表示

図 9: パラメータ調整パネル (左部分)

図 10: プリセットを楽譜に反映

リックし、頂点として確定させる。

### (3) 発想標語の適用と調整

操作パネルのプルダウンメニューから「Cantabile」などの発想標語を選択する。「楽譜に反映」をクリックすると、楽譜上の該当箇所に発想標語のテキストが追記され、楽譜上の発想標語のテキストをクリックするとパラメータの設定値を確認することができる（図 10）。また、パラメータ調整パネル（図 9）を展開することで、プリセット値（Base, Peak, Onset）をスライダーで数値入力で調整することも可能である。

### (4) 音源生成と履歴管理

「音源を生成」をクリックすると、4.3.4 項の処理を経て WAV ファイルが生成される。完了後、再生プレーヤーが表示され、加工前の演奏と加工後の演奏を聞き比べることができる。なお、納得がいかない場合は「一つ前に戻る（Undo）」ボタンで直前の状態に復帰できるほか、加工結果を MIDI ファイルや PDF（楽譜）として保存することも可能である。

## 5. 評価

### 5.1 適用事例

本節では、実装したシステムを用いて実際の楽曲に演奏表現を付与した事例を示し、生成されたパラメータの変化とシステムの有効性について考察する。

#### 5.1.1 同一フレーズに対するパラメータ変化の検証

まず、同一のフレーズに対して異なる発想標語を適用し、生成されるパラメータ（CC2 および Onset）の違いを定量



図 11: 対象フレーズ (15 小節～18 小節)



図 12: 対象フレーズの範囲指定と頂点設定

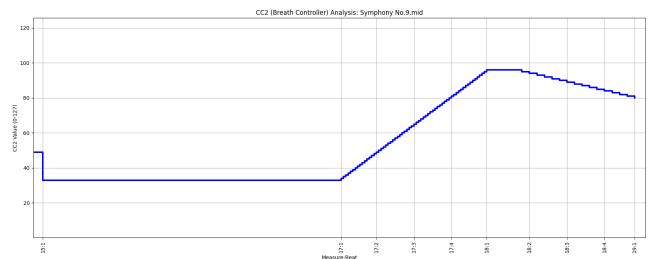


図 13: 適用前の CC2 値の推移

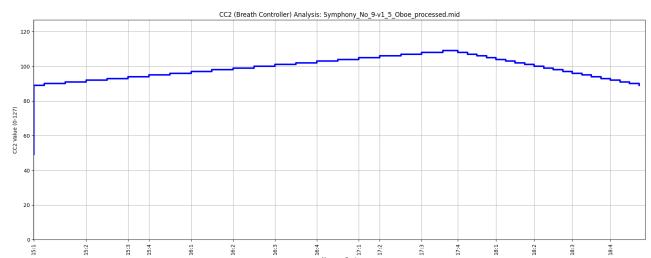


図 14: Cantabile 適用後の CC2 値の推移

的に検証する。

#### (1) 検証の概要

検証には、以下の楽曲を用いた。なお、本検証では Mutopia Project で公開されている楽譜データ [21] を元に、メロディーパート（冒頭～18 小節）を抽出し、Oboe パートとして編集した MusicXML と MIDI データを使用した。

- **楽曲**: 交響曲第 9 番「新世界より」第 2 楽章 (A. Dvorak 作曲)
- **パート**: Oboe (原曲の English Horn を読み替え)
- **対象フレーズ**: 抜粋データの最後 4 小節 (15 小節～18 小節)

対象フレーズの楽譜を（図 11）に示す。このフレーズに対し、性格の異なる 3 つの発想標語「Cantabile（歌うように）」「Maestoso（堂々と）」「Con brio（生き生きと）」をそれぞれ適用した。なお、フレーズの頂点は推定結果により、17 小節 4 拍目と 18 小節目 2 拍目の音符が候補として表示されたが、前者の音符を選択した（図 12）。

#### (2) CC2 値（強弱）の変化

元の音源の CC2 値を図 13 に、各発想標語を適用した際の CC2 値の変化を図 14、図 15、図 16 に示す。

グラフより、cantabile では変化幅が+35 と中程度であ

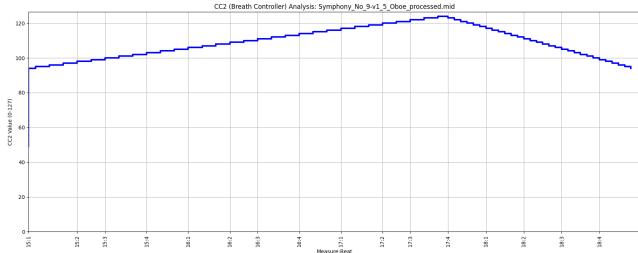


図 15: Maestoso 適用後の CC2 値の推移

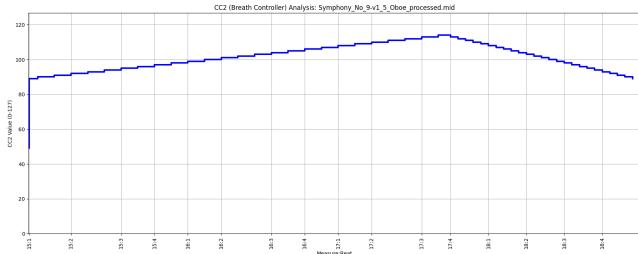


図 16: Con brio 適用後の CC2 値の推移

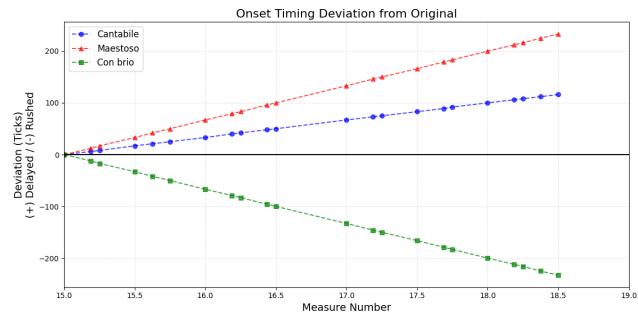


図 17: 発想標語による Onset (発音時刻) の比較

り、旋律を自然に歌うような滑らかな変化となっている(図 14)。一方、Maestoso では、Peak CC2 が+50 と高く設定されているため、頂点に向かって急激な音量の盛り上がりが確認できる(図 15)。Con brio も変化幅は大きいが、後述する Onset (発音時刻の制御) の違いにより、Maestoso とは全く異なる印象を与える(図 16)。

### (3) Onset (発音時刻) の比較

各発想標語を適用した際の Onset のずれを可視化したグラフを図 17 に示す。横軸は時間、縦軸は元の演奏データ(黒線)からの Onset のずれを Tick 単位で示している。グラフが正の領域にある場合は演奏が遅延しており、負の領域にある場合は前倒しになっていることを意味する。

このグラフが直線的な変化を示しているのは、本システムの Onset 制御の設計思想を反映したものである。本システムでは、発想標語が持つ時間的な揺らぎをフレーズが進むにつれて効果が蓄積していくプロセスとしてモデル化した。例えば、「Maestoso (堂々と)」の持つ「溜め」や重厚感は、フレーズの頭から終わりに向かって徐々にテンポが重くなっていくことで表現される。逆に「Con brio (生き生きと)」の持つ疾走感は、フレーズ内で徐々に前進するエ

ネルギーとして表現される。

この累積的な効果を実現するため、4.3.4 で述べたように、発音時刻の補正量をフレーズ開始点からの経過拍数に比例させる線形モデルを採用した。これにより、プリセット値( $P_{onset}$ )が正の場合はフレーズが進むほど遅延量が線形に増加し、負の場合は前倒し量が線形に増加する。

図 17 は、この設計が意図通りに機能していることを明確に示している。Maestoso(赤線)では、プリセット値+40ms という大きな正の値が適用されているため、グラフは急な右上がりの直線となり、フレーズの終わりにかけて大幅に遅延していく重厚な表現が生成されている。Cantabile(青線)では、+20ms が適用され、Maestoso よりも緩やかな傾きの直線となり、自然に歌うようなゆったりとした息遣いが表現されている。一方、Con brio(緑線)では、-40ms の設定により、グラフは右下がりの直線となり、フレーズの進行とともに継続的に前倒しになることで、疾走感や切迫感のあるパラメータが output されていることが確認できた。

以上の結果から、本システムは Expression (CC2) による強弱制御だけでなく、Onset による時間制御においても、発想標語の持つニュアンスをパラメータとして適切に反映できているといえる。

### 5.1.2 楽曲全体を通じた演奏デザインの実践

5.1.1 項では単一フレーズにおけるパラメータ変化を検証したが、実際の楽曲演奏においては、曲の展開や繰り返しに応じて表情を変化させることが求められる。そこで、本節では「G 線上のアリア」[22] の楽曲全体に対し、場面ごとに異なる発想標語を適用することで、物語性のある演奏デザインを試みた。

#### 構成の意図と適用パラメータ

本楽曲(全 36 小節)を音楽的な流れに基づいて 4 つのセクションに分け、主旋律である Flute パートに以下のようにプリセットを適用した。なお、本検証に用いたデータは、Mutopia プロジェクトで公開されているものをもとに、筆者が本検証用に編集したものである。

#### (1) 前半 (1~12 小節) :

- 1~6 小節 : **Cantabile** (歌うように)
  - 意図: 冒頭の主題を滑らかに歌わせる。
  - 頂点: 3 小節 1 拍目
- 7~12 小節 : **Dolce** (甘く)
  - 意図: 反復部分でより繊細な表現とする。
  - 頂点: 9 小節 1 拍目

#### (2) 中間 (13~24 小節) :

- 13~18 小節 : **Appassionato** (情熱的に)
  - 意図: 音域が上昇し感情が高まる部分とする。
  - 頂点: 17 小節 3 拍目
- 19~24 小節 : **Sostenuto** (音を十分に保って)
  - 意図: クライマックスに向け、音を十分に保ちながらエネルギーを溜める。



図 18: 「G 線上のアリア」に対する演奏表現の適用

- 頂点：21 小節 2.75 拍目

### (3) クライマックス (25~30 小節) :

- 25~30 小節 : Maestoso (堂々と)

- 意図：楽曲の頂点として重厚感を表現する。  
- 頂点：26 小節 1.5 拍目

### (4) 終結 (31~36 小節)

- 31~36 小節 : Tranquillo (静かに)

- 意図：曲の終わりに向かって静けさを表現する。  
- 頂点：34 小節 2.75 拍目

本システムで上記の表情指示を適用した結果、生成された楽譜を図 18 に示す。図中の発想標語およびフレーズ範囲のハイライト、頂点を示す緑色のマーカーは、全て本システムの機能によって自動的に描画されたものである。

## 5.2 頂点推定（和声情報の有無による影響）

## 5.3 演奏音源に対する表情付けの定性的な再構築

## 6. 評価

本章では、本システムの有効性と、実装した頂点推定機能の評価について考察を行う。

### 6.1 発想標語と演奏パラメータの対応付け

第 5.1.1 章の適用事例において、同一フレーズに対し、適用する発想標語を切り替えることで、生成される演奏表現が大きく変化することが示された。これは、本システムが定義したプリセットが、各発想標語の持つ音楽的ニュアンスを CC2 (強弱) と Onset (時間) という物理パラメータとして適切にモデル化できていることを示している。

特に、CC2 による強弱表現だけでなく、時間軸上の微細な制御 (Onset) を組み合わせた点が、豊かな演奏表現を

実現する上で効果的であった。例えば、「Maestoso」では Expression (CC2) の大きな山に加え、Onset の遅延が重厚感を生み出し、「Con brio」では Onset の前倒しが疾走感を生み出すなど、発想標語の持つ多面的な性格を再現することができた。これは、Velocity 制御中心のシステムでは表現が困難であった、管楽器などのロングトーンの表現の幅を広げる上で有効である。

また、本システムではフレーズの頂点をユーザが指定する設計とした。これにより、同じ発想標語を適用する場合でも、頂点をフレーズの前半に置くか後半に置くかで強弱の山の形が変化し、演奏のニュアンスをさらに細かく制御することができた。これは、音楽表現が「グループごとに一つの頂点を持ち、その頂点に向かって強弱が増減する」という、保科理論の基本原則を実践的に検証するものであったといえる。

### 6.2 頂点推定機能の評価と課題

楽曲を表情豊かに演奏するためには、フレーズの構造、特にその頂点をどこに置くかを理解することが重要である。しかし、楽器の演奏経験が浅い学習者にとって、楽譜から自力で頂点を読み解くことは必ずしも容易ではない。本システムの頂点推定機能は、このような楽曲解釈を行際の指針を提供することを目的として実装した。

本機能の有効性を検証するため、複数の楽曲のフレーズを対象に、システムが提示する頂点候補と、音楽理論に基づき人間が決定した頂点との比較を行った。その結果、多くの場合において、システムは妥当な頂点候補を提示することが確認された。特に、フレーズ内で最も長い音や、音高が最も高い音を頂点として正しく抽出する傾向が見られ、3.4 項で定義したルールが一定の有効性を持つことが示された。

しかし、一部のフレーズにおいては、システムが提示する候補が、人間による音楽的解釈と一致しないケースも確認された。これは、本アルゴリズムが和声的な文脈を考慮していないことが原因だと考えられる。例えば、音高や音価では目立たない音であっても、強拍に置かれた非和声音 (倚音など) が和声的な緊張感から頂点として判断されることがある。現在のシステムは単旋律の情報のみを扱っているため、このような和声進行に伴う音楽的な重みを判断することが困難である。

本システムの頂点推定は、完全な正解を提示するものではなく、あくまでユーザの解釈を支援するための機能である。そのため、システムが提示した頂点候補を参考にしつつも、最終的な決定権はユーザに委ね、ワンクリックでの選択や候補以外の音符の指定を可能にした。このような UI 設計は、演奏者による主体的な解釈の構築を支援するという本研究の目的に合致したアプローチであるといえる。

### 6.3 持続音楽器を考慮した階層的フレーズ表現

## 7. おわりに

本研究では、五線譜を見ながら直感的に演奏表現をデザインできる環境の構築を目的とし、Web ブラウザ上で動作する支援システムを開発した。管楽器などの音が持続する楽器の特性を考え、CC2 による滑らかな強弱制御と、Onset による時間制御を組み合わせた。また、代表的な発想標語をプリセットとして用意し、ユーザが抽象的な言葉と具体的な音の変化の関係を探求できる環境を実現した。適用事例の検証を通じて、本システムが発想標語の持つイメージをパラメータとしてうまく反映し、豊かな演奏表現を生み出せることを確認した。

本システムの実用性をさらに高め、より高度な演奏デザインを可能にするためには、機能面および応用面においていくつかの課題が挙げられる。

まず、頂点推定機能の高度化である。6.2 項の考察で述べたように、現在の頂点推定は単旋律の音価や音高のみに依存しており、和声的な文脈を捉えきれていない。実際には、フレーズの頂点はメロディだけでなく、伴奏パートとの響きの関係性（協和・不協和）からもフレーズの頂点を判断される。そのため、この和声情報を無視している現状の仕様が、システムの解釈のズレを生む一因となっていた。今後は、簡易的な和声分析機能をバックエンドに導入し、和声情報を含めたスコア計算を行うことで、より音楽の文脈に合った頂点候補を提示できるようアルゴリズムを改良する必要がある。

次に、より複雑な記譜情報への対応である。現在のシステムでは、リピート記号などの楽譜の繰り返し記号を解釈できず、1 回目の演奏にしか表現が適用されないという制限がある。しかし、実際の演奏では 1 回目より 2 回目の方を盛り上げたいなど、繰り返しの中で表現を変化させることが行われる。今後は、これらの繰り返し記号をシステムが認識し、ユーザが 1 回目と 2 回目で異なる発想標語を適用するといった、より実践的な演奏デザインを可能にするための機能拡張が必要である。

さらなる発展の可能性として、アンサンブル演奏への応用が挙げられる。本システムは現在、1 人のパート（単旋律）を編集することに特化しているが、吹奏楽やオーケストラといった合奏での音楽の表現は、単一パートだけでは完結しない。将来的には、複数のパート（スコア全体）を同時に表示・編集し、例えば、「主旋律のパートの頂点に合わせて、伴奏のパートの音量を少し下げる」といったパート間の相互作用を考慮した表現付けを支援する機能へと拡張することが期待される。これにより、個々の奏者の視点だけでなく、合奏全体を俯瞰する指揮者のような視点での演奏デザインを探求できるようになるだろう。

本研究で開発したシステムが、多くの演奏者にとって、

自身の音楽表現の可能性を探るための有効なツールとなることを期待する。

**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。研究テーマに悩んでいた時期からいつも親身に相談に乗っていただき、本研究の完成まで温かくご指導くださいましたことに、心より感謝申し上げます。

また、研究室の 4 年生の皆さんには、日々の生活の中で数えきれないほど助けていただきました。研究で行き詰った時に話を聞いてくれたことや、何気ない雑談の時間が大きな励みになりました。皆さんと過ごした時間がなければ、この研究を乗り越えることはできなかつたと思います。

また、本システムの評価にご協力いただいた 1 年生や 2 年生の皆様にも、心より感謝申し上げます。本当にありがとうございました。

## 参考文献

- [1] 保科 洋: 生きた音楽表現へのアプローチ: エネルギー思考に基づく演奏解釈法, 音楽之友社 (1988).
- [2] 橋田 光代, 田中 駿二, 片寄 晴弘: Mixtract : ユーザの意図に応える演奏表現デザイン支援環境, 研究報告音楽情報科学 (MUS), Vol. 2010-MUS-84, No. 8, pp. 1–6 (2010).
- [3] 島村楽器: 【DTM】DAW ソフトって何？解説と製品一覧ページ！ | 島村楽器 錦糸町パルコ店, <https://x.gd/rnsBz>.
- [4] Dorico SE - 楽譜作成ソフトウェア, <https://www Steinberg.net/ja/dorico/se/>.
- [5] 無料楽譜ソフトウェアダウンロード - Sibelius | まずは, <https://my.avid.com/get/sibelius-first>.
- [6] Cubase | 音楽制作のすべてをカバーする DAW, <https://www Steinberg.net/ja/cubase/>.
- [7] Apple (日本) : Mac のための Logic Pro, <https://www.apple.com/jp/logic-pro/>.
- [8] 無料の作曲・楽譜作成ソフト | MuseScore, <https://musescore.org/ja>.
- [9] NotePerformer 4, <https://www.noteperformer.com/>.
- [10] クリエイター向けオールインワン AI 音楽制作プラットフォーム | ACE Studio, <https://acestudio.ai/jp/>.
- [11] Lerdahl, F. and Jackendoff, R.: A Generative Theory of Tonal Music", MIT Press (1983).
- [12] 橋田 光代, 片寄 晴弘, 野池 賢二, 保科 洋, 河原 英紀: LG-006 音楽聴取に関する一検討：グループと頂点の推定 (G. 音声・音楽), 情報科学技術レターズ, Vol. 3, pp. 145–148 (2004).
- [13] 橋田 光代, 片寄 晴弘, 保科 洋: スラー境界情報に基づくフレーズ頂点の推定, 情報処理学会研究報告ヒューマンコンピュータインタラクション (HCI), Vol. 2003, No. 111(2003-HI-106), pp. 103–109 (2003).
- [14] 浅野 桐子: 持続音系楽器のための演奏表現付け支援システムの開発, 博士論文, 福知山公立大学 (2025).
- [15] Welcome to Flask — Flask Documentation (3.1.x), <https://flask.palletsprojects.com/en/stable/>.
- [16] FluidSynth | Software Synthesizer Based on the SoundFont 2 Specifications, <https://www.fluidsynth.org/>.
- [17] Music21 Documentation — Music21 Documentation, <https://www.music21.org/music21docs/>.
- [18] Wim.Vree.Org/svgParse/Xml2abc.Html, <https://wim.vree.org/svgParse/xml2abc.html>.
- [19] Abcjs: Javascript Music Library, <https://abcjs.net>.

- [20] Mido - MIDI Objects for Python — Mido 1.3.2 Documentation, <https://mido.readthedocs.io/en/stable/>.
- [21] Symphony No. 9 in E Minor, by A. Dvořák (1841–1904), <https://www.mutopiaproject.org/cgibin/piece-info.cgi?id=1793>.
- [22] Music Listing - without Preview Images, <https://www.mutopiaproject.org/cgibin/make-table.cgi?searchingfor=BWV+1068>.