

平成 23 年度 卒業論文

音楽理論に基づいた鼻歌作曲支援システム
“ハミコン”

指導教員 北原鉄朗専任講師

日本大学文理学部情報システム解析学科

木村 翔平、鈴木 優、鈴木 智文

2012 年 2 月 提出

概 要

誰もが手軽に作曲する手段の一つとして鼻歌を PC のマイクから入力して MIDI 形式に変換する方法がある。音楽情報処理の分野においても鼻歌を MIDI 形式に変換する研究は数多くされてきたが、音楽経験のない人等が使った際に鼻歌から得られる周波数をそのままノートナンバーに変換してもユーザの意図と異なっていたり音楽的に不自然なメロディになってしまう場合が多い。よって誰にでもできる作曲手段として鼻歌作曲システムを実装する上で重要なことは、元の鼻歌の曲らしさを残しながらも音楽的に自然な形になるように修正を行うことであるが、既存の鼻歌作曲システムの多くはその実現には至っているとは言えない。

そこで本システムでは入力された鼻歌メロディに対して音楽理論に基づいた音高修正とリズム修正をすることで、鼻歌を音楽的に妥当な形で出力する鼻歌作曲システムを試作した。システムを実装し、実行例を検証した結果、本システムの修正ルールによる出力結果が音楽的に妥当な形であるということを確認した。

目 次

目 次	iii
図 目 次	v
第 1 章 序 論	1
1.1 本研究の背景	1
1.2 本研究の目的	2
1.3 本論文の構成	3
第 2 章 作曲支援システムのアプローチ	5
2.1 関連研究	5
2.1.1 鼻歌作曲システム	5
2.1.2 鼻歌以外による自動作曲システム	8
2.1.3 その他の作曲支援システム	9
2.1.4 まとめ	9
2.2 本研究のアプローチ	9
第 3 章 システムの構成	13
3.1 伴奏部	13
3.2 マイク入力部	14
3.3 タップ監視部	15
3.4 リズム修正部	15

3.5	歌唱評価部	16
3.6	音高修正部	17
3.6.1	基本的な考え方	17
3.6.2	ダイアトニックノート化ルール	17
3.6.3	倚音・刺繍音・経過音化ルール	17
3.6.4	和声音化ルール	19
3.6.5	オクターブルール	19
3.7	MIDI 変換部	19
第 4 章	実装と実行例	21
4.1	実行例 1	21
4.2	実行例 2	22
4.3	実行例 3	23
4.4	評価と考察	27
4.4.1	実行例 1 の考察	27
4.4.2	実行例 2 の考察	27
4.4.3	実行例 3 の考察	27
第 5 章	結 論	29
参考文献		31

目 次

2.1	プログラムの一連の流れ	11
3.1	システムの概要	14
3.2	タッピングについて	15
3.3	リズム修正について	16
3.4	ダイアトニックノート化ルール適用前	18
3.5	ダイアトニックノート化ルール適用後	18
3.6	和声音化ルール適用前	19
3.7	和声音化ルール適用後	19
3.8	オクターブルール適用前	19
3.9	オクターブルール適用後	19
4.1	音高修正の一例：(a) 修正前、(b) ダイアトニック化ルール適用後、 (c) 倚音・刺繍音・経過音化ルール適用後、(d) 和声音化ルール適用後	24
4.2	音高修正の一例：(a) 修正前、(b) 倚音・刺繍音・経過音化ルール適 用後、(c) 和声音化ルール適用後、(d) オクターブ修正ルール適用後	25
4.3	音高修正の一例：(a) 修正前、(b) 倚音・刺繍音・経過音化ルール適 用後、(c) 和声音化ルール適用後、(d) オクターブ修正ルール適用後	26
4.4	望ましい音高修正の一例：(a) 修正前、(b) あるべき自然な修正の形	28

第1章 序 論

本章では、研究の背景、目的、従来研究との違いなどを、過去の論文を引用しながら述べる。最後に、本論文の構成を述べる。

1.1 本研究の背景

現在、ニコニコ動画や YouTube などの動画サイトの普及によりオリジナル曲を公開して楽しむ文化が定着してきており、それに伴いオリジナル曲を手軽に作る手段の需要が高まっている。しかし、いざ作曲をするためには楽器も使えないといけない上に、コード進行やスケールなどの音楽知識も必要となる。そんな作曲経験の全くない人でも作曲を楽しめるための手段として自動作曲システムも多く研究が進められてきており、コンピュータを用いて作曲を楽しむ人も増えてきている。自動作曲システムには、ループ素材を使って繋ぎ合わせていく ACID Pro[1]、歌詞を入力すると自動でメロディを出力する ORPHEUS[2] が挙げられるが、このようなシステムは音楽知識を必要としない代わりにユーザの意図を反映できない場合が多く、ほとんどシステム任せの作曲になってしまう場合が多い。また、Band in a Box[3] のように細かい作りができるが和声音の知識なしには簡単に使えないものもある。

楽器やシステムなど数ある作曲方法の一つとして、誰もが経験したことがあり、また、音楽経験や知識がなくてもメロディが奏でられる「鼻歌」という方法がある。音楽情報処理の分野においてこれまでに鼻歌作曲に着目した研究がされてきており、Voice-to-MIDI[4], MAISTA MUSIC[5], ソング頼太 [6] など鼻歌を MIDI 形

式に変換する研究は数多くされてきた。

しかし、鼻歌が誰もができる作曲手法とは言え、音楽経験のない人や音楽知識が浅い人が自分の意図するメロディの音高やリズムを正確に歌うことは容易なことではない。鼻歌から得られた基本周波数を機械的にノートナンバーに変換してもユーザの意図と異なっていたり、音楽的に不自然なメロディになってしまう場合が多い。そのため鼻歌をそのままメロディにすると、音楽的に妥当ではなく作曲システムとしては使いにくい。つまり鼻歌作曲システムによって生成されるメロディには、元の鼻歌メロディの曲らしさを残しながらもある程度音楽的に自然な形になるように修正を行う必要がある。このような鼻歌作曲システムの実装を目的とした研究はなく、作曲支援システムの一つの研究として検討する余地はある。

1.2 本研究の目的

本研究の目的は、入力された鼻歌メロディを音楽的に妥当な形に修正し出力する、作曲支援システムの実装を図ることである。鼻歌作曲支援システムの実装では、ユーザーが歌った鼻歌メロディの音高とリズムを正確に読み取ることが当面の課題となる。そして読み取った音高とリズムを音楽的に妥当な形に修正することが重要な課題として挙げられる。

本稿では、ユーザーが歌ったメロディを音楽的にあるべき妥当な形に修正しつつ、概形を残しユーザーの意図に沿った出力結果になることが必要である。音楽理論に基づいた音高修正ルールの確立を図り、その成果を報告する。また本稿ではメロディの一音一音のリズムを明確にするために、Voice-to-MIDI[4]の提案手法のタッピングを用いることでリズム分析を行い、一音一音の区切りを図る手法を提案する。

1.3 本論文の構成

本論文は次の構成からなる。第2章では、まず従来の作曲支援、鼻歌作曲システムを紹介し、従来システムの課題および本研究のアプローチについて述べる。第3章ではシステムの概要について説明し、伴奏部、マイク入力部、歌唱分析部、リズム修正部、音高修正部のそれぞれの役割および課題について述べる。第4章では、実行例と実際に生成されたメロディに対して考察を行う。第5章では、本論文で述べてきた研究成果についてまとめ、本研究の今後について考察する。

第2章 作曲支援システムのアプローチ

本章ではまず従来の作曲支援システムを紹介し、それらのシステムの課題について述べる。次にそれらを踏まえて本研究のアプローチと特徴を述べる。

2.1 関連研究

ここでは自動作曲システムと鼻歌作曲システムの関連研究を述べ、そのシステムの課題を示す。

2.1.1 鼻歌作曲システム

- ながしミュージシャンシステム：豊次郎 [7]

豊次郎はMIDI入力によるメロディ、あるいはマイクから入力された鼻歌に対して、後続するメロディを予測し、それに対応する伴奏音を出力するリアルタイムシステムである。複数の楽曲のメロディとそれに対応する伴奏音の情報のデータベースを用いてリアルタイムで検索することで後続するメロディを予測する。一般に検索条件が具体的かつ詳細化し過ぎていると、適合する事例が見つからないという事態が起こりうる。ただし適合する事例が存在する場合には、できる限り具体的なものを優先して利用する。このような状況に対処して以下の探索を行う。具体性の高い抽象化構造から抽象度の高い抽

象化構造に向かって探索を行う。より長い（古い）データ列から短い（直近の）データ列に向かって探索を行う。以上の処理の結果、合致する楽曲部分を用い後続する伴奏音を出力する。しかし、検索の合致要素が急激に増えてしまう事態が起こりうる。その場合、その時点で検索条件抽象化を停止し、統計処理により最も尤もらしい合致事例を選び出すようにしている。豊次郎は、入力されるメロディにリアルタイムで和声付けを行うリアルタイム表情付けシステムである。

- 鼻歌入力による実時間作曲支援システムの検討 [8]

山本らは、作曲者のインスピレーションを活かすようにするため、作曲作業をできる限り停止することなく行えるようなシステムを実現するのを目標にした。楽曲を制作する作業は、大きく分けて[入力セクション]と[ミックスセクション]の2つに分けられる。入力セクションでリズムの選択やフレーズ（小節の意）の入力を行い、ミックスやセクションで曲を構成する。入力セクションでは、リズムパターンの選択により鼻歌のためのテンポのガイドラインを行い、フレーズを入力に[音域][音色][クオンタイズ]の各パラメータを設定する。基準となる音域を決めることにより音域を中間値として、数値の増（減）によりオクターブ単位の高揚（低下）を表現する。音楽の専門知識を持たない人にも扱えるように入力を鼻歌によるものとしている。

- 歌詞歌唱による入力可能な Voice-to-MIDI 手法の提案 [4]

コンピュータを用いた音楽制作のための MIDI シーケンスデータ入力手法の鼻歌入力（V-to-M）がある。伊藤らは、歌詞歌唱でも入力精度を大きく向上可能なタッピング併用 V-to-M 手法を提案した。精度が下がる主原因は、歌唱を適切に1音ずつ区切られないことにあり、これが連鎖的に音高や音長の誤認識を引き起こしてのではないかと考えた。そこで提案手法では、歌唱と平行してメロディのリズムに合わせてボタンや鍵盤などをタッピングす

ることによって、区切り情報を入力する。システムに打鍵情報が入力されたらこれをトリガーとして、離鍵まで歌唱の瞬時ピッチ算出処理を繰り返し、瞬時ピッチの時系列を記録する。離鍵後、瞬時ピッチ時系列から半音単位でとったヒストグラムの最頻値を求め、これをこの区間（1音分）の音高として出力する。

- MAISTA MUSIC[5]

MAISTA MUSIC は、メロディをマイクに向かってハミングして作曲を行うシステムである。最初にメロ作成に使う伴奏を Web 上で選択を行う。伴奏は「ジャンル」「テンポ」「曲調」「ステージ」「おすすめ」から絞り込んで選ぶことができる。作曲するパート（Aメロ、Bメロなど）を選択したらマイクをセットして伴奏に合わせてハミングの録音を行う。後はメロディ生成ボタンをクリックするだけでユーザのハミングに合ったメロディが自動的に生成することができる。

- ソング頼太 [6]

鼻歌で MIDI 作曲や発声練習ができ、マイクから鼻歌で音声入力して楽譜を作成し、MIDI ファイルで出力できる。音声入力だけでなく、キーボードやマウスから音符やコードを入力して MIDI 作曲することもできる。また、モジュレーションやピッチベンドなど、特殊な MIDI イベントを細かく設定することもできる。

- ハミングによる単旋律の自動採譜 [9]

清水らは、音高と音価を推定できる方法を提案した。音高についてその絶対値が正しくなくても、相対的に正確に歌えていれば各階名の周波数比は保たれることを利用した。そして、平均律音階の周波数比と等しく並んだくし型のテンプレートをを用いたマッチングにより、利用者にとっての音階位置を認

識し、階名を推定した。音価については、音符の音価が基本的には2のべき乗の体系をしていることを利用し、2のべき乗間隔で並んだくし型のテンプレートを用いた。そして、オンセット間隔の局所的頻度分布とのマッチングをとることにより、テンポが揺らいでも音価を推定できる方法を提案した。

2.1.2 鼻歌以外による自動作曲システム

- ORPHEUS[2]

Orpheus (オルフェウス) は、ユーザが入力した日本語の歌詞に合わせて歌を自動で作曲してくれるインターネット上のシステムである。このシステムは、言葉のイントネーションを利用しており、言葉を読み上げるを読み上げるときの音の上下を決定している。キーボードから入力された歌詞のイントネーションを、標準語のデータベースを用いて解析している。次にそのイントネーションの上下に従ったメロディをデータベースを用いて制作している。また、リズム、コード進行、伴奏(メイン)、伴奏(サブ)、ドラムスを設定することで最もふさわしいメロディを探しだし制作することができる。

- Band in a Box[3]

Band in a Box は、コードの入力、伴奏スタイルの選択から作曲を行うシステムである。伴奏スタイルとしてジャズスイング、ブルース、シャッフル、ミディアムロック、ワルツ、ポップバラードなどが用意されている。目的のジャンルのスタイルを選択するだけで、入力したコード進行通りの演奏を作成することができる。また、コードの知識がなくても、システムが自動でコード進行を作成する。

2.1.3 その他の作曲支援システム

- ACID Pro[1]

ACID Pro は、音楽の専門的知識がなくてもドラッグ&ドロップを中心とした簡単な操作で音楽制作をすることができる。ループシーケンサーというループとは曲の中で繰り返されるミュージックサンプルがあり、ループの種類は、ドラム・ベース・ギター・ピアノ・ストリングス・管楽器系など幅広くある。ループを様々な組み合わせることによってオリジナルの音楽を作成できる。また、ピッチやテンポの自動的マッチ (ピッチやテンポの合わないループ素材でも、自動的にマッチさせる機能) ができる。他に MIDI 打ち込みでの楽曲作成はもちろん、マイクやピアノを繋いで歌や楽器の演奏を読み込むことができる ACID Pro は、ループ素材を使うことで楽器や音楽知識のない初心者でも作曲が可能だと言える。

2.1.4 まとめ

既存の自動作曲システムにユーザの意図するものを伝えるには、調やリズム、和声などの音楽知識を必要とする情報を入力をしなければならない。しかし、音楽知識のないユーザは、あいまいな入力しかできず、意図する作曲が行えるとは言えない。既存の鼻歌作曲システムは、音楽知識のないユーザでも容易に使用できる。ユーザの意図するメロディをそのまま鼻歌で読み込むと多くの場合でユーザの意図とは異なり、音楽知識に不自然なものになってしまう。結果として音楽知識のないユーザでも容易にできる作曲システムとは言い難い。

2.2 本研究のアプローチ

歌唱の素人がメロディを歌った場合、2つの点であいまい性が生じる。

1. 音高が安定しないために、音と音の切れ目が明確でなくなる場合がある。
2. 意図する音高を正しく発声できない場合が多く、音高が安定しなかったり、中途半端な音高（例えば「ド」と「ド#」の中間の音）を発声してしまい、そのまま単純にノートナンバーに変換すると、音楽的に適切でない値になる場合がある。

本研究では、1. に対しては、Voice-to-MIDI[4]と同様に歌唱のみから MIDI への変換を行うのではなく、キーボードのタッピングを併用して MIDI へ変換する。又、これに 8 分音符刻みになるようにリズム修正を加える。つまり、ユーザが各音符の発音時刻に同時にキーをタップすることで発音時刻をあいまい性なく取得する。2. に対しては、一旦ノートナンバーに変換を行ったものを保存しておき、意図する音高を取得するためタップ間のノートナンバーから中央値を取る。その後、音楽理論に基づいて修正を行う。例えば、変換されたノートナンバーが「ソ#」で、そのときのコードが「C」の場合は「ソ」に修正する。プログラムの一連の流れを図 2.1 に示す。

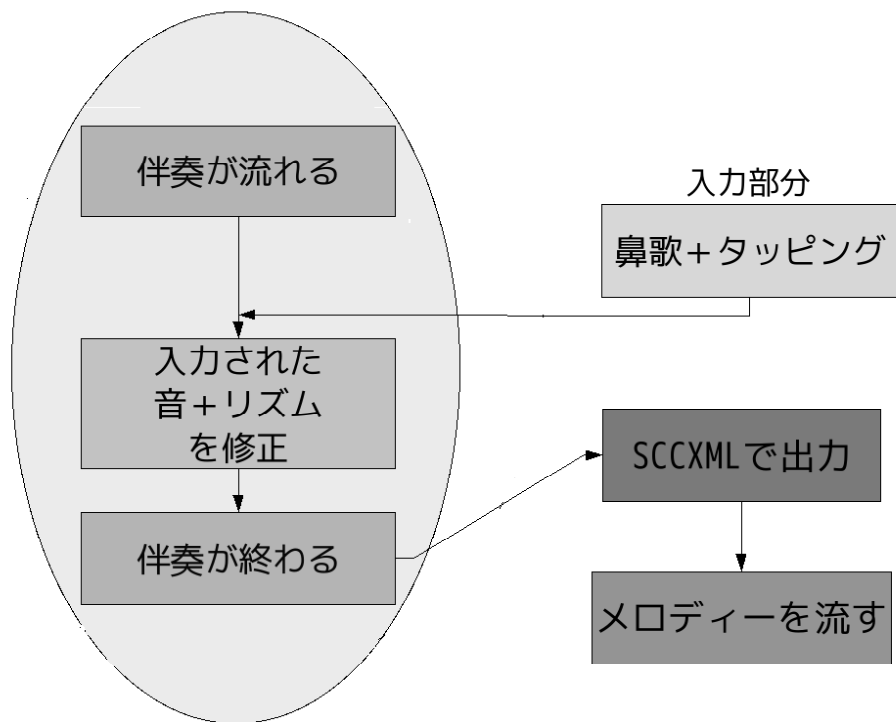


図 2.1: プログラムの一連の流れ

第3章 システムの構成

本システムは、図 3.1 で示されているように伴奏部、マイク入力部、タップ監視部、歌唱分析部、リズム修正部、音高修正部、MIDI 変換部の全 7 つの部で構成されている。

ユーザがシステムを実行すると MIDI デバイスの選択画面、次にオーディオデバイスの選択画面が表示される。それぞれ使用するデバイスを選択すると、正方形のウィンドウが出現し、同時にノイズの測定がはじまる。次に、あらかじめ用意した伴奏用 MIDI データが再生されるので、スペースキーをタッピングしながら鼻歌を歌う。伴奏が終了したら、画面に表示されているウィンドウをクリックするとプログラムが終了する。終了後、鼻歌の音高・リズムが修正されたものが MIDI に変換され、XML ファイルが出力されるので、XML ファイルを SMF に変換し、再生すると伴奏と修正されたメロディを同時に聞くことができる。

鼻歌入力から出力までの間に、音楽理論に基づいたルールを設定しており、最終的に鼻歌が音楽的に妥当なメロディに修正される。作曲に必要な音楽知識は要さない。

本章では各部の機能について述べる。

3.1 伴奏部

伴奏部は、あらかじめ決まっている伴奏用 MIDI ファイルを再生する部分である。また、本システムではあらかじめ決まっている伴奏用 MIDI ファイルのコード進行は既知とする。本研究では簡単のため、小節が変わるタイミングでのみコー

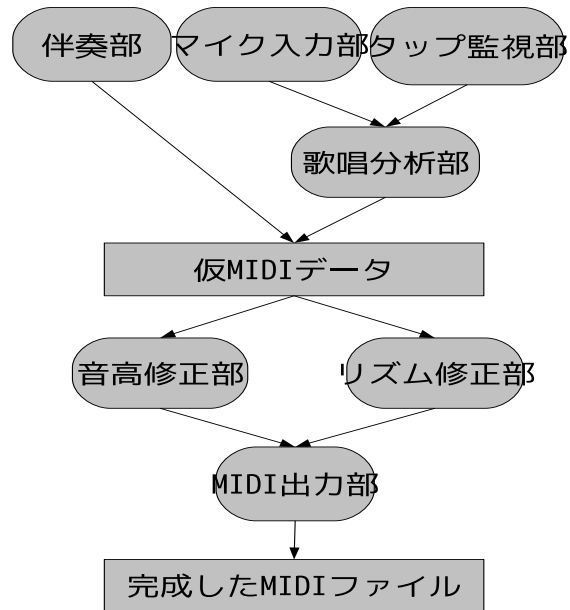


図 3.1: システムの概要

ドが変化するものとする。

3.2 マイク入力部

マイク入力部では、鼻歌をマイクで感知する部分である。マイクで入力する際、マイクのホワイトノイズやパソコンの機械音、周囲の音などのノイズが発生しているとユーザが意図していない音をマイクが感知してしまい、音高を出力する時、ユーザが求めている音高で出力されてしまう。そこで、ノイズ対策として最初の0.1秒をユーザには声を出さないでもらい、その間に周囲のノイズを計測する。計測したノイズの中で一番大きいノイズを基準値とする。その基準値から3倍以上の音でないとシステムが感知しないようになっている。

3.3 タップ監視部

前章の課題で述べたように、鼻歌のみで入力すると、使用するユーザが発声時音高が安定しないために、音と音の切れ目が明確でない場合が多い。

そこでユーザには鼻歌で入力する際、1音1音の区切りを明確にするため、スペースキーをタッピングしながら鼻歌を歌ってもらう。そうすることにより、図3.2のように音と音の切れ目をはっきりし、リズムの正確性があがる。またタッピングをしたティックタイムの情報はタップ監視部で行う。

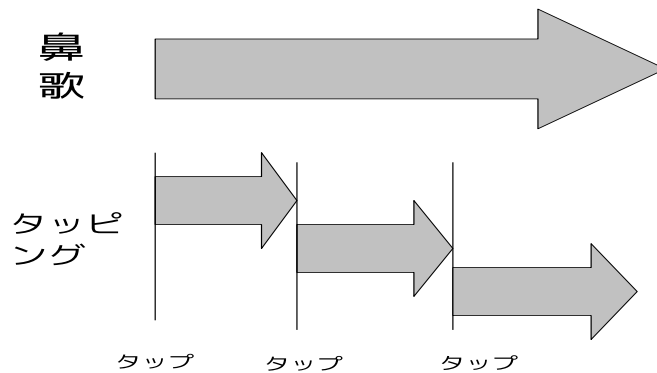


図 3.2: タッピングについて

3.4 リズム修正部

ここでは、音の長さ・音の始まりのタイミングを修正する部分である。音の長さや音の開始のタイミングやズレを修正することをクオンタイズと言う。

本システムでは中途半端なリズムは出力したときに妥当でないリズムになってしまうので、8分音符より細かいリズムは強制的に8分音符の長さに修正する(図3.3)。リズム修正により、出力されたとき妥当なリズムに出力されるようになる。

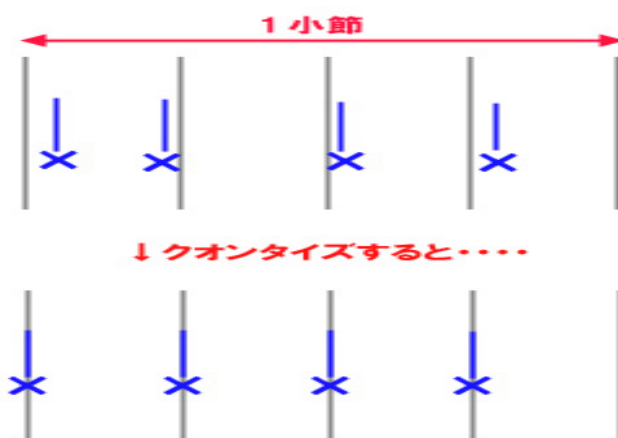


図 3.3: リズム修正について

3.5 歌唱評価部

ここでは、マイク入力部で感知したものと、タップ監視部でタッピングした情報を組み合わせ、音高を求めノートナンバーに変換する。

まず、マイク入力部で感知した周波数を以下の式 3.1 を使いノートナンバーに変換する。

$$f^{-1}(x) = 12 \log_2 \left(\frac{x}{440} \right) + 69 \quad (3.1)$$

次に、タップ監視部で記録したタッピングからタッピングまでのティックタイムの間に得られたノートナンバーを一つの音とする。ここでは1音の値をどう決めるかについて述べる。考えられる値のとり方は2種類ある。一つは平均値、もう一つは今回選んだ中央値である。平均値を使うと、なんらかの理由でユーザの意図しない音が1つでも感知してしまうと出力結果が大幅に変わってしまう。中央値であれば、少し違った反応をしてもユーザが求めたい音が妥当に出力される。よって中央値を使用することにする。

3.6 音高修正部

ここでは、ユーザが歌った鼻歌を音楽的にありえるメロディに音高修正を行う。今回は伴奏があらかじめ設定しており、基本的に黒鍵を使わないコード進行にしている。

3.6.1 基本的な考え方

本研究では、基本的な方針として

- 隣り合う音の高さが離れすぎなくする。
- 伴奏と不協和にならないメロディにする。

の2つを採用する。伴奏と不協和にならないメロディにするもっとも単純な方法は、すべての音を和声音に変換することである。しかし、すべての音を和声音にしてしまうとメロディに使われる音の種類が極端に減り、かえって不自然なメロディになる可能性がある。音楽理論では、メロディにおける非和声音の使われ方としていくつかの分類がなされている。本研究では、非和声音の中で代表的なものとして倚音・刺繍音・経過音(本稿では「許容非和声音」と呼ぶ)を取り上げ、非許容非和声音を許容非和声音に変換するルールを導入する。

3.6.2 ダイアトニックノート化ルール

まず入力された音高がダイアトニックスケール上にない場合、すべて一番近いダイアトニックノートに修正する。

3.6.3 倚音・刺繍音・経過音化ルール

倚音、刺繍音、経過音は、その使われ方に一定の傾向がある。

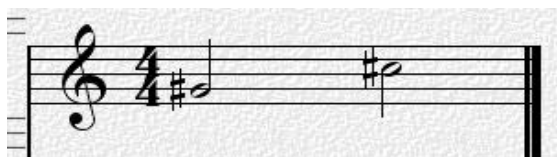


図 3.4: ダイアトニックノート化ルール適用前



図 3.5: ダイアトニックノート化ルール適用後

J-POP の楽曲 69 曲を調べた結果、次の傾向が得られた。

- 倚音 214 箇所のうち、後続音が一番近い和声音になるのが 179 箇所。
- 刺繍音 257 箇所のうち、挟まれた音が挟んだ音と長 2 度の関係になるのが 223 箇所。
- 経過音 191 箇所のうち、挟まれた音とその直前の音と長 2 度の関係になるのが 188 箇所。

以上の傾向に基づき、次のルールを導入する。

1. 小節が和声音にない音から始まる場合、後続音が前の音から高い場合は一番近い上の和声音に修正する。また後続音が前の音から低い場合、もしくは、同じの場合は一番近い下の和声音に修正する。
2. 小節内で 3 つの連続する音があり、同じ和声音で別の音を挟んでいる場合、中央の音が前後の音より高ければ長 2 度上の音に修正する。また中央の音が前後の音よりも低ければ長 2 度下の音に修正する。
3. 小節内で 3 つの連続する音があり、始まりの音と 3 つ目の音が和声音で、音高の流れが上に長 3 度の関係にある場合、中央の音は始まりの音の長 2 度上に修正される。または音高の流れが下に長 3 度の関係にある場合、中央の音は始まりの音の長 2 度下に修正する。

3.6.4 和声音化ルール

次に、前節の3つのルールに該当しなかった非和声音は、一番近い和声音に修正する。伴奏再生部で述べたが、小節が変わるごとにコード進行も変わる。図 3.6 及び図 3.7 に示してあるのは C のコード時の例である。

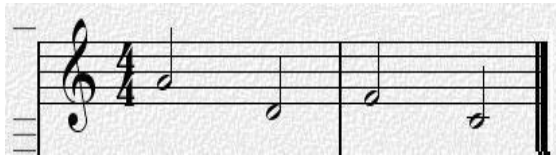


図 3.6: 和声音化ルール適用前



図 3.7: 和声音化ルール適用後

3.6.5 オクターブルール

最後に入力された各音高が直前の音高よりも1オクターブ以上高く離れている時は入力された音を1オクターブ低く修正、また1オクターブ以上低く離れていれば入力された音を1オクターブ高く修正し、音高差を1オクターブ以内にします。

このことにより音の繋がりをよくし、メロディを音楽的に妥当な流れにする。



図 3.8: オクターブルール適用前

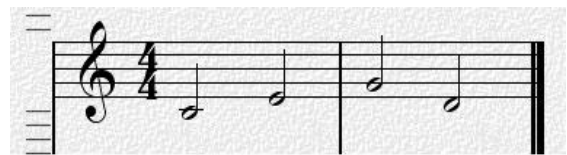


図 3.9: オクターブルール適用後

3.7 MIDI 変換部

上記のリズム修正部・音高修正部にて、リズムと音高を修正し、伴奏用 MIDI が流れ終わったらプログラム終了となり、SCCXML 形式 [10] で出力される。

第4章 実装と実行例

以下の節では本稿で提案するシステムを用いた鼻歌作曲の使用例について述べる。

本稿で提案するシステムは Groovy を用いて実装した。MIDI ファイルの読み込み、再生の生成には CrestMuseXMLToolkit[10] を用いた。

ユーザには実行の際に、既知している伴奏に合わせてタッピングをしながら鼻歌を歌ってもらう。旋律の長さは4小節としコード進行は Am-F-G-C とする。まず実行例1として、ノンダイアトニックノートや倚音・刺繍音・経過音を含むサンプルメロディの修正過程と結果について説明する。

実行例2では、前後の音で1オクターブ以上離れる音を含むサンプルメロディの修正過程と結果について説明する。実行例3では、前後の音で1オクターブ以上離れる音を含むサンプルメロディの修正過程と結果について説明する。

4.1 実行例1

実行例1ではノンダイアトニックノートや倚音・刺繍音・経過音を含むサンプルメロディを入力として用いる。使用例を図4.1に示す。図4.1に、上から、入力したメロディ、ダイアトニック化ルール、倚音・刺繍音・経過音を加味した独自のルール、和声音化ルールを順次適用する過程を表す。実行例1のサンプルメロディには前後の音で1オクターブ以上離れている音はないので、オクターブ修正ルールは適用されない。1小節目の2つの音ド とファ はどちらもダイアトニックスケールにない音であったためダイアトニックノートであるドとソに変更され、その後、和声音化ルールでソをラに修正されている。2小節目はソがコードFの

和声音ではないため、倚音のルールにより後続音のミがファに修正されている。3小節目は刺繍音のルールにより、挟まれた音ファが挟む音レの2度上であるミに修正されている。4小節目はレの音がダイアトニックスケールにない音であったためダイアトニックノートのミに修正され、経過音ルールにより挟まれた音ミが挟んだ音ドとミの2度の音であるレに修正されている。以上の結果、1小節目はドラ、2小節目はソファ、3小節目はレミレ、4小節目はドレミという出力結果が得られる。

4.2 実行例2

実行例1では前後の音で1オクターブ以上離れる音と、倚音を含むサンプルメロディを入力として用いる。使用例を図4.2に示す。図4.2に、上から、入力したメロディ、倚音・刺繍音・経過音を加味した独自のルール、和声音化ルール、オクターブ修正ルールを順次適用する過程を表している。実行例3のサンプルメロディにはノンダイアトニックノートの音はないので、ダイアトニックノート化ルールは適用しない。1小節目は2つ目の音ソが和声音化ルールによりラに修正される。2小節目は2つ目の音である1オクターブ高いソ(以下、ソ[^]と記す)が和声音化ルールによりファ[^]に修正され、その後オクターブ修正ルールによりファに修正される。3小節目の音はいずれのルールにも当てはまらないのでそのまま出力する。4小節目は2つ目の音シが倚音ルールによりドに修正される。すべての修正ルールを実行例3のサンプルメロディで実行すると、1小節目はララ、2小節目はファファ、3小節目はシシレレ、4小節目はシドドという出力結果が得られる。

4.3 実行例 3

実行例 3 では、前後の音で 1 オクターブ以上離れる音を含むサンプルメロディを入力として用いる。使用例を図 4.3 に示す。図 4.3 に、上から、入力したメロディ、倚音・刺繍音・経過音を加味した独自のルール、和声音化ルール、オクターブ修正ルールを順次適用する過程を表している。実行例 3 のサンプルメロディにはノンダイアトニックノートの音はないので、ダイアトニックノート化ルールは適用されない。1 小節目は 2 つ目の音シを和声音化ルールによりドに修正し、その後オクターブ修正ルールによりド[^]、3 つ目の音をラ[^]に修正されている。2 小節目は 2 つ目の音シを和声音化ルールによりドに修正されている。3 小節目は 2 つ目の音ミを和声音化ルールによりレに修正され、3 つ目の音ファを和声音化ルールによりソに修正されている。4 小節目の音はいずれのルールにも当てはまらないのでそのまま出力される。すべての修正ルールを実行例 2 のサンプルメロディにかけると、1 小節目はド[^]ド[^]ラ[^]、2 小節目はドド、3 小節目はレレソソ、4 小節目はドレミという出力結果が得られる。

The figure displays four musical staves, labeled (a) through (d), illustrating the process of pitch correction. Each staff is in 4/4 time and features a treble clef. Above the first staff (a), the chords Am, F, G, and C are indicated. The staves are connected by large downward-pointing arrows, showing a sequential transformation. Staff (a) shows the original melody with some chromaticism. Staff (b) shows the result of applying diatonicization rules, where notes are adjusted to fit the key of C major. Staff (c) shows the result of applying voice/embellishment and passing note rules, resulting in a smoother, more melodic line. Staff (d) shows the final result after applying harmony rules, where the melody is further refined to fit the harmonic context.

図 4.1: 音高修正の一例：(a) 修正前、(b) ダイアトニック化ルール適用後、(c) 倚音・刺繍音・経過音化ルール適用後、(d) 和声音化ルール適用後

(a)

(b)

(c)

(d)

図 4.2: 音高修正の一例：(a) 修正前、(b) 倚音・刺繍音・経過音化ルール適用後、
(c) 和声音化ルール適用後、(d) オクターブ修正ルール適用後

The figure illustrates a four-step process for pitch correction in musical notation. Each step is represented by a staff in 4/4 time with a key signature of one sharp (F#).
(a) The original score. It begins with a treble clef, a 4/4 time signature, and a key signature of one sharp. The melody consists of several eighth and quarter notes.
(b) The score after applying the 倚音・刺繍音・経過音化 (Grace Note, Embellishment, and Passing Note) rules. The notation remains the same as in (a).
(c) The score after applying the 和声音化 (Harmonization) rules. The notation remains the same as in (a).
(d) The score after applying the オクターブ修正 (Octave Correction) rules. The notation remains the same as in (a).
Large grey arrows point downwards from (a) to (b), from (b) to (c), and from (c) to (d), indicating the sequence of rule applications.

図 4.3: 音高修正の一例：(a) 修正前、(b) 倚音・刺繍音・経過音化ルール適用後、
(c) 和声音化ルール適用後、(d) オクターブ修正ルール適用後

4.4 評価と考察

4.4.1 実行例 1 の考察

修正前のメロディはノンダイアトニックノートや非和声音が半分以上もあり、コードと不協和になる音だったためとても聞きにくいメロディであった。修正後はすべての音がダイアトニックスケール上にあり許容非和声音とコードトーンの音であるため、あきらかな不協和が少なくなり、修正前よりも修正後のほうが聞きやすいメロディになったと言える。またすべての音をコードトーンにすると童謡のように単調なメロディになってしまったのに対し、すべての音をコードトーンにする修正よりも許容非和声音を加味したこの修正ルールのほうが自然なメロディになることがわかった。

4.4.2 実行例 2 の考察

修正後のメロディはコードトーンと許容非和声音のみの音になったのであきらかな不協和音はない。2 小節目ではオクターブ修正ルールによりファファ[^]がファファに修正されたため、より音楽的に妥当で自然なメロディになったといえる。

4.4.3 実行例 3 の考察

コードトーンと許容非和声音のみの音になったのであきらかな不協和音は無くなっているが、1 小節目の始めの音に合わせて後続の 2 つの音をオクターブ修正してしまっている。ここでは、始めの音ド[^]をドに修正したほうがユーザの意図からあまり離れずに自然なメロディになるのではないかと考えられる。3 小節目ではレミファソという動きのあるメロディを和声音化ルールによりレレソソというあまり動きのないメロディに修正してしまっている。この場合は修正をかけずにレミファソという音を残した方が自然なメロディの流れなのではないかと考えられ

る。以上より、現在の修正ルールでは前の音に後続音が合わせてオクターブ修正しているが、前の音に合わせるだけではなく曲の流れを優先した修正ルールができれば良い。また現在の修正ルールで経過音は挟まれた音1音分のみを修正対象としているが、挟まれた音が2音以上ある場合も修正対象とする必要がある。望ましいメロディ修正の形を図4.3に示す。

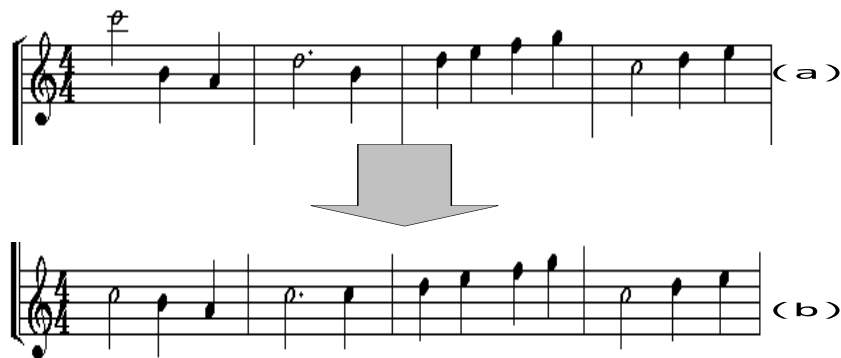


図 4.4: 望ましい音高修正の一例：(a) 修正前、(b) あるべき自然な修正の形

第5章 結 論

本稿では音楽理論に基づいた自動鼻歌作曲支援システム“ハミコン”を提案した。あらかじめ決まっている伴奏用 MIDI ファイルにあわせて鼻歌を歌い、鼻歌のリズムと同時にタッピングする。本システムは、それを音楽的に自然なメロディに修正し出力をする。本システムは、メロディ修正ルールに着目し、音楽的に妥当ではない音を修正することにより、作曲経験のない人、楽器経験、音楽知識がない人でも簡単に使用できることを可能にした。また、タッピングを使用することで1音の区切りをユーザの意図するものにし、さらにクオンタイズに修正することで不自然でないものを実現することを可能にした。評価実験により我々が独自で設定したルールに基づいた場合、修正後のメロディが修正前のメロディと比べて音楽的にもっともらしい形になることがわかった。それにより本稿の目的である、音楽理論に基づいた鼻歌作曲支援システムを実現することができた。

本研究の今後の課題として、第4章で述べたような、楽曲全体の流れを考慮して修正する仕組みを導入することがあげられる。修正ルール以外では、現段階では伴奏データは既知なものであり、あらかじめコード進行・伴奏用 MIDI ファイルを用意し使用しなければならないので、伴奏のコード進行・小節の長さをユーザ自身が自由に調節し決定することがあげられた。また、本システムは入力された鼻歌を修正し MIDI ファイルで出力するのみであるので、そこで今度は出力後シーケンサのように音高・リズムの調整をユーザの自由に編集をできるようにしていきたい。そうすることで、ユーザの意図するものが出力されてしまった場合すぐ手直しが出来、より使用しやすくなると予想される。最後に、音高修正ルールの

参考にしている楽曲数が69曲分しかないので、様々な楽曲データを集め、ルールを改善し、検証していきたい。

参考文献

- [1] SONY : ACID Pro.
- [2] 深山寛, 中妻啓, 米林祐一郎, 酒向慎司, 西本卓也, 小野順貴, 嵯峨山茂樹 :
“Orpheus 歌詞の韻律に基づいた自動作曲システム”, 情報処理学会研究報告, 2008, MUS-76, pp.179–184, 2008.
- [3] 株式会社イーフロンティア : Band in a Box.
- [4] 伊藤直樹, 西本一志 : “歌詞歌唱による入力可能な Voice-to-MIDI 手法の提案”, 情報処理学会シンポジウム論文集, 2007, 4, pp.71–72, 2007.
- [5] 株式会社エキシング : MAISTA MUSIC.
- [6] 有限会社ダイナシステム : ソング頼太.
- [7] 豊田健一, 片寄晴弘 : “ながしミュージシャンシステム豊次郎”, 情報処理学会シンポジウム論文集, 2005, 4, pp.39–40, 2005.
- [8] 山本宏之, 米倉達広 : “鼻歌入力による実時間作曲支援システムの検討”, 1997, Sogo Pt 1, pp.313, 1997.
- [9] 清水純, 丸山剛志, 三浦雅展, 柳田益造 : “ハミングによる単旋律の自動旋律”, 2004, 111(MUS-57), pp.95–100, 2004.
- [10] 北原鉄朗, 橋田光代, 片寄晴弘 : “音楽情報科学研究のための共通データフォーマットの確立を目指して”, 情処研報, 2007, MUS-71, pp.149–154, 2007.

謝 辞

本研究は日本大学文理学部情報システム解析学科北原研究室の卒業研究の研究テーマとして実施されました。本研究を進めるにあたり、ご指導を頂いた卒業論文指導教員の北原鉄朗先生に感謝致します。また、日常の議論を通じて多くの知識や示唆を頂いた北原研究室の同期・後輩の皆様に感謝します。