

対旋律自動生成アプリケーションの制作

Study on the Production of Automatic Counterpoints Generator

1W130560-8 八木 陽介

指導教員 菅野 由弘 教授

YAGI Yosuke

Prof. KANNO Yoshihiro

概要：本研究は、ユーザーによって入力された音列を定旋律とし、それに対し対位法に基づく対旋律を自動生成するアプリケーションの制作を目的とするものである。対旋律とは、主旋律、すなわちメインとなるメロディに対して、それを効果的に補うような旋律のことを指す。対旋律の自動生成に関する先行研究としては、動的計画法や確率モデルを組み合わせて用いたもの[1][2]が存在するが、ユーザーの存在を想定したアプリケーションの形で対旋律の自動生成を可能としているものは少ない。そこで、本研究ではMIDIデバイスを利用し入力された音列に対して生成された対旋律を、内蔵MIDI音源による発音および楽譜の形式でユーザーが聴覚的、視覚的に把握できるような出力機能を備えたアプリケーションを制作した。加えて、対旋律生成アルゴリズムに採用した手法、並びにその手法を用いた対旋律生成実験の結果に対する評価を踏まえた上でのアルゴリズムの改善に向けた考察を行った。

キーワード：対位法、対旋律、自動生成、アプリケーション

Keywords：counterpoint, counter melody, automatic generation, application

1. はじめに

制作するアプリケーションの内部はユーザーインターフェース部分と対旋律生成アルゴリズム部分に分かれており、本書ではユーザーインターフェース部分の仕様および対旋律生成アルゴリズムに用いた手法と、それを用いて生成された対旋律の評価、並びにその評価に対する考察について述べる。なお、アプリケーションの開発環境として、主にMIDI入力の検出や対旋律生成アルゴリズムにおける演算処理を担う部分をオープンソースの統合開発環境であるProcessingを用いて制作し、内蔵MIDI音源による発音や楽譜表示を担う部分を、Cycling '74社のMax 7を用いて制作した。また、ProcessingおよびMax 7でそれぞれ作成したプログラム間での情報の受け渡しには、CNMATが開発したOSC(Open Sound Control)と呼ばれる通信プロトコルを用いた。

2. ユーザーインターフェースの仕様

制作したアプリケーションはProcessing上で動作する「Counterist_engine.pde」とMax 7上で動作する「Counterist 0.01.maxpat」の2つのプログラムファイルから構成される。ユーザーは2つのプログラムを同時起動させた上で、使用するPCに接続したMIDIデバイスを用いて定旋律の入力を行う。入力完了後、「Counterist_engine.pde」の実行画面を1回クリックすることで対旋律の生成が行われ、その結果がMax側にOSCを用いて送信された後、内蔵MIDI音源による発音および楽譜形式の表示が行われる。

3. 対旋律生成アルゴリズム

対旋律の自動生成については、『二声対位法』[3]を参考に禁則をプログラム上の条件分岐に落とし込み、生成しようとする各音にコストという概念を導入し、コストが最小となる音を辿っていくという手法を採用した。この手法において、対旋律の生成に対し次のようなモデル化を行った。

- ・定旋律中のある1音を入力Aとし、およびその音に対して付与しようとする対旋律中における1音の、一つ前の音を入力Bとする関数Xを考える。この関数Xは、次に付与する対旋律中の1音を、入力Aおよび入力Bを基に決定する。
- ・入力Aおよび入力Bに基づいて関数Xが出力を決定する方法として、コストという概念を導入する。関数Xは次に付与しようとする音を入力Cとし、入力Cが入力AおよびBとの関係においてふさわしくないものであった場合、その入力Cにコストと呼ばれる数値を加算する。関数Xは次に付与しようとする音全てに対して上記の計算を繰り返し、その中でコストが最小であった入力Cを最終的な出力とする。
- ・コストが同一の入力Cが複数存在した場合は、その中からランダムに出力を決定する。
- ・上記の手順を、定旋律が終了するまで繰り返す。

なお、対旋律は1対2とし、定旋律に対して上声に付与することを仕様として定めている。

4. 対旋律の生成例および評価

制作したアプリケーションを用いて実際に対旋律を数種類生成し、それらに対する評価を作曲家、菅野由弘に依頼した。その評価を基にアルゴリズムにおけるコスト設定の見直し、関数が引数としてとる情報および対旋律に利用可能な音域の拡張等の修正をプログラムに加え、再度対旋律の生成および評価を行なったところ、評価に全体的な向上が見られた。しかし、アルゴリズム自体の問題として、常に前の音との関係において次の音を定めるため、弱拍が経過音あるいは刺繍音であるかどうかの判定が行えないことに起因する減点評価が多く発生しており、この問題をいかに解決するかが今後の課題であると言える。

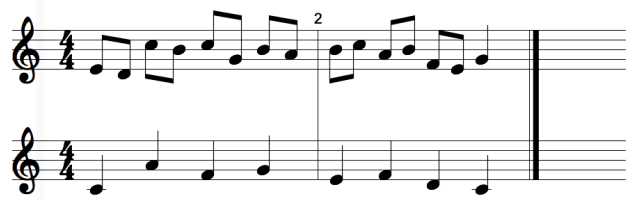


図1 対旋律の生成例（修正前）

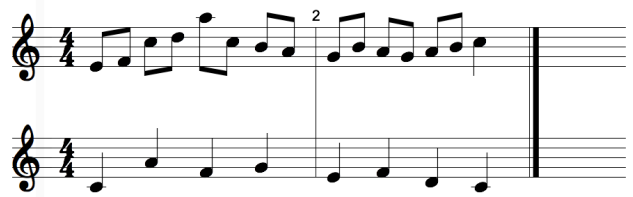


図2 対旋律の生成例（修正後）

5. 結論および今後の展望

対旋律生成アルゴリズムについては、評価に基づく修正を繰り返すことでさらなる評価の向上に繋がると考えられる。しかし、模範的な対旋律の生成例を得るためには、対旋律全体の旋律動向を把握するために、生成後に再度旋律全体を見直すような追加の処理を行う必要があることが分かった。今後はそうしたアルゴリズム部分の精度の向上を目指しつつ、アプリケーションの形であることを活かし、ユーザーが対旋律の旋律動向について、禁則を避けつつもある程度生成結果に介入できるようなインターフェースの実装を行なっていきたい。

参考文献

- [1] 中潟昌平, 西本卓也, 嵯峨山茂樹 (2004) 『動的計画法と音列出現確率を用いた対位法の対旋律の自動生成』
- [2] 田中翼, 西本卓也, 小野順貴, 嵯峨山茂樹 (2010) 『確率モデルを用いた対位法および模倣に基づく自動作曲』
- [3] 池内友次郎 (1965) 『二声対位法』 音楽之友社。