

曲構造中のモチーフ変形パターンに注目した 自動作曲

6R-4

西岡大祐、高田正之、小谷善行
(東京農工大学 工学部 電子情報工学科)

1. はじめに

計算機による自動作曲は興味深いテーマであり、曲の構造に注目したもの、確率的推移によるものなど、さまざまな方法による試みがなされている。本研究では、与えられたモチーフを変形することで作曲する方法をとった。

2. 方針

楽曲は、モチーフを繰り返したり、変奏したりして成り立っている。いわゆるニューミュージックに代表されるポップスでは、ある印象的なメロディが繰り返し現れる。これは、自由なモチーフに規則的な変形を施して得られる。そこに注目して、実際の楽曲中の段落のそれぞれでモチーフがどのように繰り返され、変奏されているかを調べたのをもとに、それをシミュレーションすることで曲を作り出す。

処理の流れは次のようになる。まずユーザが生成したい曲のデータ(曲のキー、モチーフとそのコード、拍子)を入力する。システムは、それと内部の知識をもとに、段落構成とそこでのコード進行を生成して1曲分の繰り返し構造をつくり、段落ごとにモチーフを変形していきながらメロディを生成し出力する。出力には、共通楽譜データ形式(Coda)^[5]を用いる。

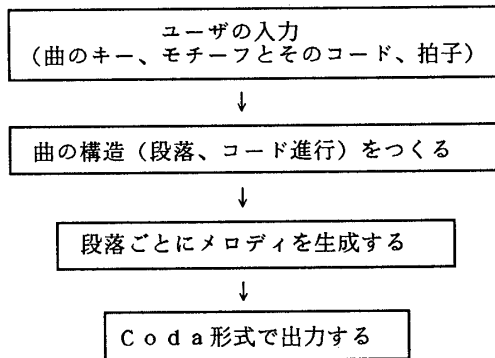


図1 処理の流れ

3. システム

以上の自動作曲システムを、ソニー NWS-830 上の K-Prolog を用いて実現した。ユーザからのひとつ以上のモチーフの入力を用いて作曲する。

3.1 モチーフ

このシステムでは、モチーフを2小節くらいの長さを

もつメロディとする。モチーフは曲の条件とともにユーザが入力する。入力インタフェースは、特に工夫せず音名を使ってProlog形式で直接入力するようにした。次の例は、2小節、キーがCメジャー、4拍子、各小節にひとつのコードのものである。

[('C', 4), [('C', maj), (3, c, 4), (3, d, 4), (3, e, 4), (3, f, 4), ('/', ('A', m), (3, e, 4), (3, d, 4), (3, c, 2), '/']]



3.2 モチーフの変形

モチーフは、変形したものを曲中に配置する。モチーフの変形には、大きく分けて反復、変奏という操作を用意した。それらに、さらに細かいパターンを用意した。

反復: モチーフをそのまま、もしくはコードに応じてシフトして使う。

変奏: リズムの変形が主な操作である。コードに応じたシフトとともに使われる。

平行反復: モチーフの出だしの音がコードトーンにあるときはそのまま反復して、ないときはコードに応じてシフトする。

隣接音反復: 出だしの音が、コードトーンにないときは、出だしの音に近いコードトーンから始まるようにシフトする。

平行反復つなぎ: 最後のコードまで平行反復して、最後のコードの部分の部分を伸ばして終止感をもたせる。

モチーフ変形: モチーフをコードトーンに応じてシフトさせ、後半部分をコードトーンに修正する。

頭拍リズム変奏: モチーフの初めの2拍のリズムを変更する。

システムは以上のものを組み合わせたものを変形パターンの知識として持っていて、それをモチーフに適用させることで作曲する。

3.3 段落構造

入力されたデータ、プログラム中の曲構造に関するデータからメロディの決まっていな構造だけのリストを生成する。これを段落構造と呼んでいる。段落構造は、段落の数だけ生成される。段落構造は段落名、変形パターン、コードの3つからなる。

3.3.1 段落構成

曲の流れとなる段落構成を決める。段落の生成は、実際に調べた曲^[4] 67曲の構成から出現頻度に応じて確率的に選んでいる。頻度が1の例を除くと段落構成とその頻度は次のようになった。同名の段落と「'」のついてる段落は、同じ長さをもつ。

[a, a, b]	15	[a, a', b, c]	4
[a, a', b, a']	6	[a, b, c, c']	3
[a, a, b, c]	6	[a, b, a']	2
[a, a', b]	5	[a, a, b, a]	2
[a, a, b, b]	4	[a, b, c]	2

3.3.2 変形パターン

段落構成が決められたあと、段落ごとの変形パターンを生成する。変形パターンは、モチーフの変形を並べたリストである。ひとつの段落にひとつの変形パターンが対応する。変形パターンは、段落ごとに知識として持っているものから選ぶ。たとえば段落aについての知識の中に次のようなものがある。

変形パターン(a, ['モチーフ', 平行反復,
隣接音反復, 平行反復つなぎ], ('V', 7)).

これは、aという段落は、モチーフ、平行反復、隣接音反復、平行反復つなぎという変形がなされ、V₇のコード(相対的)で終わることを意味している。変形パターンとコード進行を組み合わせてひとつの段落の段落構造を生成する。この例では、一つの段落が8小節になる。

3.3.3 コード進行

コード進行の生成は、モチーフとともに入力されたコードの続きから生成される。モチーフのコードからコード進行を発展させるためにケーデンスを用いずに、直前のコードからの遷移確率で選んでいる(段落構成に段落の最後のコードを書くことでケーデンスのようなものができる)。確率は、長調の楽曲数十曲^[4]をもとに[6]の方法で算出した。絶対的なコードを、一度キーに独立な相対的コードに変換してコード進行を決めたあと、再びキーが決まった形に変換する^[7]。

モチーフの入力のときに任意の位置にコードを置くことができるが、そうすることでモチーフ中でのコード進行のリズムができる。ひとつの段の中でのコード進行は、すべてこのリズムにそって生成される。

3.3.2の構成から段落aについて段落構造をつかった例を示す。

段落構造(a, [[モチーフ, [('C', maj), 1, ('A', m), 1]],
[平行反復, [('F', maj), 1, ('G', 7), 1]]],

[隣接音反復, [('C', maj), 1, ('A', m), 1]],
[平行反復つなぎ, [('F', maj), 1, ('G', 7), 1]]]).

これは、段落名、モチーフの長さ単位毎に施される変形パターン、コード進行というリストである。

3.4 メロディの生成

入力されたモチーフと、段落構造からモチーフを変形してメロディを決定する。生成は段落単位で行われ、変形はモチーフの長さ単位で行われる。

3.5 出力

得られた曲は、内部データを変換して共通楽譜データ形式Codaで出力される。段落aについて生成された結果を譜面にしたもの示す。

Musical score for paragraph a. The score is written on a single staff in treble clef with a key signature of one flat (F major/D minor). The melody consists of quarter notes: C4, D4, E4, F4, G4, A4, B4, C5. The chord progression is Cmaj, Am, Fmaj, G7. The time signature is 4/4.

Musical score for paragraph a, identical to the one above. It shows the same melody and chord progression: Cmaj, Am, Fmaj, G7.

4. まとめ

一つ以上のモチーフの入力から、モチーフの変形によってつくられる楽曲の可能性を示した。結果としてできた楽曲は、パターンが決まりきった印象を受けるが段落構造をより細かく設定し、メロディ生成の段階で自由に変形できればもう少しいろいろな曲が得られると思う。

また、このシステムでは、モチーフ変形パターンと、コード進行を別に扱ったが、実際はそれらが密接に関係しているものが多い。その関係の実現は、今後の課題である。

5. 参考文献

- [1] 田中康裕, 渡邊哲史, 高田正之, 小谷善行: 音程, リズムの遷移確率と主題, 対位の構造に基づく作曲, 情報処理学会第38回全国大会, 4w-4, 1989
- [2] 木ノ下修, 高田正之, 小谷善行: 曲の構造に注目した自動作曲, 情報処理学会第36回全国大会, 3k-11, 1988
- [3] 五味邦彦, 田子正明, 佐藤忍, 野瀬隆, 高田正之, 小谷善行, 西村恕彦: 音の親和性を用いた自動作曲, 情報処理学会第32回全国大会, 6w-2, 1984
- [4] 中島みゆきギター・ソング・ブック, ドレミ楽譜出版社
- [5] 山崎直子, 佐野靖子, 渡邊哲史, 高田正之, 小谷善行: 図形情報を基本とする共通楽譜データ形式, 情報処理学会第36回全国大会, 3k-8, 1988
- [6] 河西保郎: やさしい作曲のしかた, ケイ・エム・ピー, 1987
- [7] 青柳龍也: Jazz in Prolog, bit別冊コンピュータと音楽, 共立出版, 1987