自動作曲システムにおけるリズムの生成法

6U−1

前岩　哲哉，西岡　大祐，小谷　善行，西村　恕彦
(東京農工大学　工学部　数理情報工学科)

1．はじめに

言語Ｐｒｏｌｏｇを用いてコンピュータの自動作曲システムにおけるリズムの生成法を考える。歌は詞と曲で構成される。できあがった詞から曲を作成する場合には、曲の導入、印象的なメロディーを繰り返す部分など、その曲の構成を考えて音符の種類を当てはめる作業、すなわちリズム生成が必要である。本研究は、作曲におけるリズム生成の部分を、既存の曲の調査をもとに開発することを目的とする。

2．用語の定義（文献[1]より引用）

一連のメロディーの繰り返しの中で、独立性を保持し得る最小の単位はモチーフと呼ばれ、共通のモチーフを持ち、そのモチーフが再現されるまでの間をブロックと呼ぶ。

同じモチーフを持ったブロックの集まりをセクションと呼び、セクションは類似の構成を繰り返すリフレインと、２つのリフレインをつなぐ他とは違ったモチーフを持つブリッジに分けられる。

楽曲中で一番最後にあり、その楽曲の主題となるリフレインを、サビと呼ぶ。

このようなセクションの繰り返し構造をその楽曲の構成と呼ぶ。

3．曲全体のリズム生成の手順

リズム生成を以下の手順で実現した。

ユーザーが曲の構成と文節分かち書きで区切った詞の文字数を入力する。
↓
構成ごとにリズムを生成する。
↓
類似の構成が存在する場合には、リズムを文字数に合わせて変形する。
↓
生成したリズムをＣｏｄａ[2]形式に変換する。

入力は、あらかじめ次のような形のＰｒｏｌｏｇの事実節として宣言する。

ｋａｓｈｉ（構成，［文節ごとの文字数］）．

既存の曲を例にとって説明する。

例）ブロック：a

リズム

♩♩♩　♪♪♪♪　♪♪♩♪♩♩−♪

詞：むねが　こんなに　ときめいてる

kashi(a,[3,4,6]).

4．構成ごとのリズムの生成

構成ごとのリズムの生成は、以下の手順によって行う。

文字数から小節数を決める。
↓
休符を挿入する。
↓
文字数に対応した音符を決める。

4．1 小節数の決定

小節数は、構成中の詞の文字数の合計によって決定する。

4．2 休符の挿入

休符の挿入は、既存の曲の調査から得られた結果をもとに、入力された詞の文節と文字数の関係によって行う。

具体的には、詞の一文中で、休符の存在する文節番号と文字数をあらかじめ次のような形のＰｒｏｌｏｇの事実節として宣言する。

ｈ＿ｂｕｎ（文節番号，価値）．

ｈ＿ｍｏｊｉ（文字数，価値）．

ここでの文節番号は、文中で何番目の文節かを示すものである。

この事実節から得られた文節と文字数ごとの価値を掛けた値を次の形で新しく事実節として宣言する。

ｈｙｏｕｋａｃｈｉ（価値）．

この作業を入力された、詞すべてに行い、最大の価値を持つ位置に休符を挿入する。

例）１文節３文字目

```
h_bun(1,138).
h_moji(3,69).
hyoukachi(9522).
```

4．3 音符の決定

音符の決定は、既存の曲の調査から得られた結果から確率的に選び出す。

具体的には、既存の曲の調査より、文節ごとの文字数によって分けられた音符の並びをあらかじめ次のような形のＰｒｏｌｏｇの事実節として宣言する。

ｄａｔａ（文字数，数，［音符の並び］）．

ここでの数は、調査結果中に存在する音符の並びの数である。

音符は、内部表現で♪（１６分音符）を３とし、以下音符の長さにしたがって内部表現の値を増やす。また、休符は音符の内部表現を負にしたものである。

この事実における数と、文字数別の音符の並びの総数から、乱数を用いて確率的に選び出す。

Making Rhythm for Automatic Composition System.
Tetsuya MAEIWA, Daisuke NISHIOKA,
Yoshiyuki KOTANI, Hirohiko NISIMURA
Tokyo University of Agriculture and Technology

例）３文字の音符の並びの総数３２までの乱数を発生させる。
乱数：１４
data(3,8,[12,6,18]).
data(3,17,[6,6,6]).
[6,6,6]の音符の並びを選択

5．類似の構成におけるリズムの変形

リフレインにおいて、共通するモチーフを持つべきブロックの詞の文字数が違う場合には、モチーフを変形しなければならない。その場合には、先に生成されたブロックのリズムを変形することで、類似の構成であることを示す共通のモチーフをなるべく損なわないようにする。変形は休符から休符までを一つの単位とし、その単位内の文字数を、先にリズムが生成されたブロックの文字数と比較した結果により変形を行う。

手順は以下の通りである。

先に生成されたリズムに合わせて休符を挿入する。
↓
休符までの文字数を比較する。
↓
文字数の違いによって変形を行う

5．1 変形における休符の挿入

類似の構成におけるリズムの変形は、休符から次にでてくる休符までとした。

そこで変形するブロックにも、同数の休符をなるべく近い位置に挿入しなければならない。そこで、先に生成されたリズムに挿入された位置の文字数になるべく近い文節の終わりに休符を挿入することにした。

具体的には、先に生成したリズムに挿入した文節の位置を次のような形でＰｒｏｌｏｇの事実として残しておく。

ｗｏｒｋ＿ｋｙｕ（文節位置）.

ここで文節位置は、リスト形式で入力された詞の何番目の文節かを示すものである。

上記の文節位置までの文字数の合計を出し、類似のブロックの中にその合計に一番近い文字数の文節の終わりに休符を挿入する。

例）kashi(aa,[4,4,6]).
work_kyu(2).
kashi(aa,[4,4,0,6]).（0は休符を表す。）

5．2 休符までの文字数の比較

5．1 で挿入した休符の間ごとに文字数を比較する。

5．3 文字数の違いによる変形

文字数の違いによる変形は次の三つに分けられる。

1) 文字数が先に生成されたものより多い場合。
2) 文字数が先に生成されたものより少ない場合。
3) 上記の1)、2)に関係なくまったく新しいリズムを生成する場合。

5．3．1 文字数が多い場合の変形

文字数が多い場合には、音符を挿入する場合の他にある音符を分割する場合もある。

挿入する場合には、調査結果によってその位置を決定する。

音符を分割する場合は、調査結果により１拍の音符、または最後の音符が分割されることが分かった。しかしその位置は、文字数、文節数との間に関連がみられないため、乱数で決定する。

例）ブロックａの文字数：３文字
ブロックａａの文字数：４文字
ブロックａのリズム
[12,12,12]
ブロックａａのリズム
[12,6,6,12]

5．3．2 文字数が少ない場合の変形

文字数が多い場合には、音符を削除する場合の他に、２つの音符を合わせる場合もある。

削除する場合には、調査の結果によってその位置を決定する。

音符を合わせる場合は、調査結果より、0.5拍＋0.5拍の並びのある位置、または最後の２つの音符であることが分かった。しかしその位置は、文字数、文節との間に関連がみられなかったので、乱数で決定する。

例）ブロックａの文字数：４文字
ブロックａａの文字数：３文字
ブロックａのリズム
[12,6,6,12]
ブロックａａのリズム
[12,12,12]

6．共通内部形式への変換

本研究室では、共通楽譜データ形式が設計されており、他の自動編曲、自動演奏システムなどの各システム間の接続はこのデータの形式が使用されている。本システム生成されたリズムもこの形式に変換する。

7．まとめ

本システムは、詞の文字数と文節から音符の並びの選択、休符の挿入、類似の構成の変形を行ったので人間の歌うリズムが生成できた。

参考文献

[1]池内 友次郎 他，新音楽辞典，音楽の友社，（１９８４）．
[2]山崎 直子 他，共通楽譜形式の設計，第２９回プログラムシンポジウム，（１９８８）．
[3]プリンセス・プリンセス 監修，プリンセス・プリンセスベスト，シンコーミュージック，（１９８８）．
[4]西岡 大祐 他，曲構造中のモチーフ変形パターンに注目した自動作曲，（１９９０）．
[5]新井 肇 他，曲構造とメロディーのリズム解析による自動編曲，（１９９２）．
[6]野池 賢二 他，曲の構造情報から表情付けを行う自動演奏（１９９２）．