2Q−3

アレンジルールを用いた作曲システムの構成方法について*

道小島　友和　　小林 要†

金沢工業大学§

1 はじめに

近年，任意のメロディと作成したい楽曲のフィーリング情報を選択して楽曲を自動生成するソフトウェア[1]が登場している．あらかじめ用意されたフィーリング情報による作曲方式の場合，作曲の仕方に関する情報の追加・洗練が難しい．

本稿では，音楽アレンジ技術をデータ化したアレンジルールを与えることによって作曲する方法を提案する．アレンジルールは音楽理論に基づき，さまざまな曲に適用できるものが望まれる．このため，本研究では①コードを基にした相対音階，②音の高さのアレンジとリズムのアレンジの分解，③音の高さのデータとリズムのデータの合成方法を提案する．また，コードを基に音を相対的にとらえ，アレンジルールの再利用性を高める方法を導入する．

2 システム構成

図1に，本システムの構成を示す．アレンジルールはデータベースに追加し蓄積する．素材を用意し，アレンジルールを選択し適用することによって作曲する．素材は，「主音列」「基本構成」「コード進行」の3つのモジュールで構成する．3つのモジュールはアレンジルールに従ってアレンジされる．アレンジルールは，

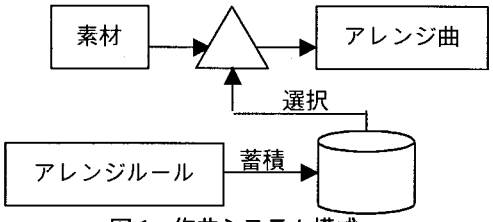

図1：作曲システム構成

アレンジを行う部分を特定し，どのようなアレンジを行うか示すものである．アレンジルールに従ってアレンジされた結果は，さらにMIDIへ変換可能なMML（Music Macro Language）[2]に加工する．

3 相対音階とコード

本手法の特徴の一つは，主音列を絶対音階ではなく，コードを基にした相対音階を用いる点にある．本手法で用いた相対音階は，コードを基に何度の音かを指定することによって音の高さを表現する．したがって，同じ度数であってもコードを変更することにより音の高さを変えることができる．度数が異なっていても，コードが異なると同じ音の場合もある．（表1参照）

表1：相対音階とコードの関係

度	1	3	5	1	3	5
コード	C	C	C	Em	Em	Em
実音	ド	ミ	ソ	ミ	ソ	シ

図2は主音列にアレンジルールを適用させた一例である．アレンジルールを適用し，主音列が変化したことがわかる．アレンジされた音列

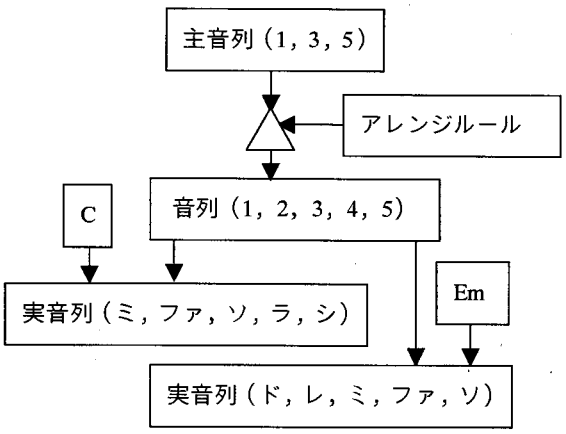

図2：アレンジルール適用例

* A system structure for arrange-rule based music composition
† Tomokazu Doukojima, Kaname Kobayashi
§ Kanazawa Institute of Technology 7-1Ohgigaoka, Nonoichi, Ishikawa921-8501, Japan

にコードを与えると，コードにしたがって，異なる実音列を得る．図2からわかるように，一つの主音列と一つのアレンジルールによって得た音列は，コードにしたがった様々な実音列を得ることができる．

コードを基にした相対音階を用いることによって，アレンジルールは実音にとらわれずに主音列をアレンジできる．つまり，様々な楽曲において適用可能なアレンジルールとなり，アレンジルールの再利用性を高めることができる．

4 データ構造とアレンジ

本システムでは，楽譜を木構造で表し，アレンジルールにしたがって木構造を変換[3]することによりアレンジを実現している．楽譜を「主音列」「構成」「コード」「リズム」のモジュールに分解する．本手法の第二の特徴は，モジュールの各々に対してアレンジルールを作成できる点である．各々のモジュールに独立性を持たせることにより，モジュールを独自にアレンジすることができる．

アレンジルールは「メロディアレンジルール」「伴奏アレンジルール」「演奏アレンジルール」に分類され，順番に適用し作曲する．

5 主音列とリズムの合成

第三の特徴は主音列とリズムの合成ルールを与えられる点である．主音列の要素数 n，リズムの要素数 m を合成するとき，主音列とリズムの対応をとり合成を行う．

・主音列 (1, 2, 3, 4, 5)　　n＝5

・リズム (8, 4, 休 (8), 8, タイ (8, 16), 16, 8)

m＝6

上のようなモジュールの場合，主音列とリズムの対応を

・リズム対応 (0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0)

のようにする．ここで，リズムの要素 i は i 分音符を表す．すなわち，リズム (8) は 8 分音符のことである．また，休符は休符 (i)，タイはタイ (i1, i2, ･･･) と書くことで表現している．リズム対応の要素 0 はリズムの要素に対応し，1 は主音列の区切りを意味する．リズム対応の要素 1 は主音列を区切るため n-1 個必要であり，前後は必ず 0 とし 0 は複数続いても良い．これより，0 は n 個以上必要となる．リズム対応の要素数を N とすると N≧2n-1 の関係が成り立つ．以上より，n≦m のとき，主音列とリズムの合成ルールが適用可能であるとわかる．リズム対応の要素 1 で区切られたまでの 0 の数を主音列とリズムの各要素に対応させることにより，主音列とリズムを対応させ合成することができる．この結果に，例えば C コードを与えることにより，図3の楽譜に合成することができる．

図3：合成後の楽譜

6 おわりに

本システムでは，楽譜を「主音列」「構成」「コード」「リズム」のモジュールに分解した．モジュールに独立性を持たせ，モジュールを独自にアレンジすることが可能となった．リズム対応を導入することにより，主音列とリズムの合成が可能となった．コードを基にした相対音階はアレンジルールの再利用性を高めることができる．

参考文献

[1]YAMAHA：めちゃらく作曲名人 2，http://www.yamaha.co.jp/product/syndtm/p/soft/raku2/index.html (2001 年 7 月 28 日現在)

[2]江頭勝巳：Midi Player & Compiler マニュアル，http://member.nifty.ne.jp/katsumi/ (2001 年 7 月 28 日現在)

[3]小林要:文法や加工手順を入力データとするテキスト加工システム RPS の開発，情報処理学会第 63 回全国大会，講演番号 6Q-02 (2001 年)