

## 逐次的リズム音程生成モデルに基づく自動作曲システム

川村 修<sup>†</sup> 大園 忠親<sup>†</sup>  
伊藤 孝行<sup>†</sup> 新谷 虎松<sup>†</sup>

本研究の目的は、過去の演奏の単純なつなぎ合わせではない、コードに調和した新しい旋律を自動作曲することである。本研究は、旋律の作曲をリズム生成の過程と音程付加の過程に分けてモデル化する。本研究では、この2つのモデルを連携させたモデルを逐次的リズム音程生成モデルと呼ぶ。旋律の生成では、まず、リズム生成を行い、次に生成されたリズムに音程を付加することで旋律の生成とする。従来の自動作曲では、過去の演奏をつなぎ合わせたものを出力するか、様々なルールを記述することにより旋律を生成していた。本手法により、従来より少ないルールの記述で過去の演奏のつなぎ合わせでない旋律を生成することができる。

### Algorithmic Composition System based on Sequential Rhythm-Interval Generation Model

OSAMU KAWAMURA,<sup>†</sup> TADACHIKA OZONO,<sup>†</sup> TAKAYUKI ITO<sup>†</sup>  
and TORAMATSU SHINTANI<sup>†</sup>

Our goal is to generate new musical phrases harmonized with chords. In our study, we model a phrase-creation as a process of combination of two processes, rhythm-generation and interval-addition. We call our proposed model *Sequential Rhythm-Interval Generation Model*. First, in generating a phrase, our system generates rhythms. Then, the system adds intervals to the rhythms. Finally, the rhythms with intervals are output and provided. In existing studies, phrases are generated based on existing phrases linked with some of them or based on rules. Our method generates brand-new phrases by less rules than existing studies.

#### 1. はじめに

近年のデジタル楽器や計算機上の音楽制作環境の発展と低価格化に伴い、個人的に音楽を作成する機会が増えてきた。個人のホームページに音楽をつけて演出したり、写真のスライドショーや動画にBGMをつけるといったことが個人のレベルで簡単に行えるようになってきた。作品に用いる音楽には、既存の曲を引用する場合と自分で作曲する場合がある。既存の曲を引用する場合、著作権に気を使わなければならないが、作曲する場合はその必要がない。作曲には音楽に関する専門的な知識が必要であり、音楽制作の環境が普及していたとしても、初心者にとっては敷居の高いものである。したがって、作曲をする専門的な技術が無い人にとっては、作曲という行為をある程度自動的に行うことが作曲を支援する方法の一つとして考えられる。

一方、計算機による自動作曲を実現することは、認知科学の分野からも興味深いことである。作曲は新しいものを作り出すと言う点で個人に大きく依存した活動であると言える。計算機による作曲を実現するには、演奏に関する知識など様々な要素を計算機に実装しなければならない。

本研究では、作曲の中でも旋律の創作、特にジャズにおける旋律の創作に焦点を当て、コード進行に調和する単音旋律を標準MIDIファイルとして出力する自動作曲システムを構築する。ジャズでは、あらかじめ決められたコードに対して、即興で演奏を行う。したがって、旋律とコードの関係が密接でありながら、様々なパターンの旋律が演奏されるので、本研究のテーマにふさわしいと考える。本システムは、標準MIDIファイル形式の演奏記録をデータベースとして用いる。本システムにより生成される旋律は、単にデータベースにある旋律をつなぎ合わせたものではない、新しい旋律である。また、本手法は、既存の手法に比べ少ない知識(ルール)で旋律を生成できるため、複雑さと曖昧さを軽減できる点で有用である。

<sup>†</sup> 名古屋工業大学大学院工学研究科情報工学専攻  
Graduate School of Engineering, Nagoya Institute of Technology  
〒466-8555 愛知県名古屋市昭和区御器所町  
TEL: (052)733-6550 FAX: (052)735-5584  
email: osamu@ics.nitech.ac.jp

以下、本稿では、第2章で、関連研究について述べ、本研究の位置づけを明確にする。第3章では、旋律の生成手法である逐次的リズム音程生成モデルについて記述し、第4章では、提案手法に基づく自動作曲システムの実装について述べる。そして、第5章では、旋律の生成実験を行う。次に、第6章で本研究について考察を行い、最後に第7章で本稿をまとめる。

## 2. 関連研究

遺伝的アルゴリズムを用いて自動作曲を試みる研究では、音楽要素を記号列で表現し、その記号列を遺伝子の列として、遺伝子の交叉、適応度評価、評価の制限、突然変異などといった処理により進化させている [1], [2]。また、初期遺伝子集団をマルコフ過程に基づき決定している。そうして得られた音楽は人間が聞いても不自然でないという点で一定の成果を得ている。しかし、遺伝子の評価と評価の制限において、自然な音楽を目指すあまり、必要以上に厳しい評価基準を与えているので、音楽として面白みに欠けるものとなっている。また、音楽についての経験と知識を評価基準として記述するには、複雑さと曖昧さの点で問題がある。これらの研究では、リズムに関してのマルコフ性について述べられていない。

我々は、過去の研究において、演奏記録を4分音符分の演奏に切り分けて用いることにより、新たな旋律の生成に応用できる可能性を確認している [3]。本研究では、より細かい音のつながりに着目し、旋律を構成する各々の音の発音時刻をマルコフ決定過程の状態と見なし、マルコフモデルを構築する。マルコフ決定過程とはマルコフ性を満たす環境において、状態が確率的に遷移する過程を指す。また環境がマルコフ性を持つとは、過去の環境の履歴すべてが、現在の環境情報に集約されていることを指す。つまり、時刻  $t$  での状態を  $s_t$ 、そこで選択された行動を  $a_t$  とすると、次の時刻  $st + 1$  は、 $st$  以前の状態  $s_0, \dots, s_{t-1}$  に影響を受けず、 $s_t$  と  $a_t$  によりのみ影響を受けて決定されることを意味する。一般に、環境はマルコフ性をみたさな場合があるが、それでもマルコフ性を満たす環境であると近似して考えることは有用な場合が多い [4]。

本研究では、以上のマルコフモデルに基づき、既存の旋律のつなぎ合わせや修正といった旋律の生成方法ではなく、学習データとなる演奏記録に従い、一つ一つの音を生成することで新たな旋律を生成する。したがって、学習データに無いような演奏も生成することができるので、バリエーションに富んだ演奏を生成することができる。また、本研究では、演奏記録を学習

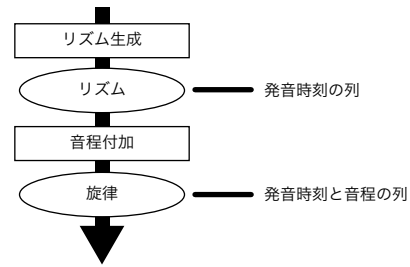


図1 逐次的リズム音程生成モデルによる旋律生成の流れを示す

することにより、演奏記録に暗に含まれる音楽知識を学習することができるので、音楽知識を明示することの複雑さと曖昧さを回避できる。

## 3. 逐次的リズム音程生成モデル

自動作曲を行うために作曲の過程をモデル化する場合、様々なモデル化の方法が考えられる。また、音楽を構成する要素(音楽要素)には、音程、音量、音色、リズム、および、コードなどが考えられる。1つの旋律には、必ず最適ナリズムが存在することから、リズムおよび音程は、一方が決まると他方に制約を与えると考えられる [5]。さらに、いくつかのジャンルは典型的なリズムを持っているように、音楽においてリズムは、特に重要な要素と考えられる [5]。したがって、本研究では、旋律を生成する過程を、リズムを生成する過程、および、音程を生成する過程の二つに分けて構築する。そして、リズムを生成した後、生成されたリズムに音程を付加することで旋律を生成する。このモデルを本研究では、逐次的リズム音程生成モデルと呼ぶ。図1に逐次的リズム音程生成モデルによる旋律生成の流れを示す。

### 3.1 リズム生成モデル

リズムは拍子と密接な関係があり、拍子に対してどのようなタイミングでリズムが刻まれたかが重要となる。例えば、5連符を演奏している途中で16分音符を演奏することはまれである。従って、本研究では、リズム生成モデルとして、音の発音時刻に関する確率的なマルコフモデルを構築する。本研究では、演奏記録から音の発音時刻の分布と状態遷移確率を求め、マルコフモデルを構築する。

#### 3.1.1 リズム生成モデルの定式化

マルコフモデルによるリズム生成モデル  $M_{rhythm} = (Q, A)$  を以下に定義する。

- (1)  $Q = \{s_0, s_1, s_2, q_1, \dots, q_N, r_1, \dots, r_N, e\}$ : 状態の有限集合。

旋律の開始および終端を示す特別な状態として  $s_0$  および  $e$  を用いる。  $s_1$  および  $s_2$  はともに小節の

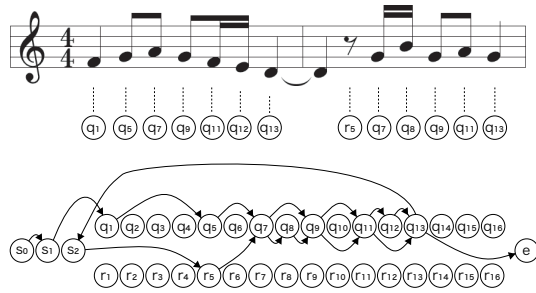


図2 リズムと状態の対応

始めを示す状態である。\$s\_1\$ は音が発音されていないことを示し、\$s\_2\$ は前の状態の発音状態が保持されていることを示す。また、\$q\_i\$ および \$r\_i\$ は小節内の位置を示す。\$q\_i\$ は発音する状態、\$r\_i\$ は発音しない状態（つまり休符）を示す。\$N\$ は発音時刻の量子化の単位と拍子の積を示す整数値である。

- (2)  $A = \{a_{ij}, b_{ij}, c_{ij}, d_{ij}\}$ : 状態遷移確率分布。  
 $a_{ij}$ ,  $b_{ij}$ ,  $c_{ij}$ , および  $d_{ij}$  はそれぞれ、状態  $q_i$  から状態  $q_j$ , 状態  $r_i$  から状態  $r_j$ , 状態  $q_i$  から状態  $r_j$ , および、状態  $r_i$  から状態  $q_j$  への遷移確率である。
- (3)  $\pi = \{\xi_0\}$ : 初期状態確率分布。  
 $\xi_0$  は  $s_0$  が初期状態である確率である。すべての旋律は旋律の開始を示す特別な状態  $s_0$  から始まる。したがって、 $\xi_0 = 1$  となる。

### 3.1.2 リズム生成モデルの例

図2では、4分の4拍子において、発音時刻の量子化の単位を4、(つまり  $N = 16$ )とした場合の、旋律のリズムと各音の状態の対応が示されている。楽譜で示されている演奏は  $s_0, s_1, q_1, q_5, q_7, q_9, q_{11}, q_{12}, q_{13}, s_2, r_5, q_7, q_8, q_9, q_{11}, q_{13}, e$  の順に状態遷移する。まず、リズムの状態遷移は旋律の開始を示す特別な状態  $s_0$  から始まる。次に、小節の始めを示す状態へ遷移するが、まだ音は発音されていないので状態  $s_1$  へと遷移する。その後楽譜上の音に対応した状態を遷移する。1小節目の最後の音と2小節目の最初の音はスラーにより結ばれているので、1小節目最後の音を示す状態  $q_{13}$  は、前の状態の音が発音されたままであることを示す状態  $s_2$  を経由することにより次の状態  $r_5$  へと遷移する。再び楽譜上の音に対応した状態を遷移し、2小節目最後の音を示す状態  $q_{13}$  では、旋律の終端を示す状態  $e$  へと遷移する。

### 3.1.3 リズム生成のアルゴリズム

リズム生成モデルは確率的な状態遷移を持つマルコフモデルにより構築している。本研究では、新しい旋律の生成を目的としているので、状態遷移確率を最大にするようなリズムを得るのではなく、状態遷移確率

に従ったランダムウォークによりリズムを生成するアプローチをとる。これにより、過去の演奏とは違う演奏を出力することができる。

## 3.2 音程付加モデル

先に述べたようにリズムと音程の間には密接な関係がある。音程を決定するには、発音のタイミングだけでなく、音の長さを考慮する必要がある。長い音が用いられる場合は、ハーモニーから制約を受けることとなる[5]。また、音程の変化をコード、もしくは、スケール上の移動ととらえた場合、アルペジオや半音進行などのようにコード、もしくは、スケール上を連続して上昇、および、下降することがある。

本研究では、これらリズムと音程の変化を音程付加モデルとして、確率モデルにより構築する。本手法では、リズム生成を先に行うことにより、音程を決定する前に発音のタイミングと音の長さを知ることができるので、それらを音程の決定に用いることができる。

### 3.2.1 音程変化

本研究では音程変化を次のように定義する。

- (1) STAY: 前の音と現在の音が同じ高さ。  
 (2) CHORD: STAY でなく、かつ、前の音が現在のコードの音に解決。  
 (3) SCALE: CHORD でなく、かつ、前の音が現在のスケールの音に解決。  
 (4) HALF: CHORD でなく、かつ、SCALE でない。  
 また、各音程変化の量を次のように定義する。

- $degree(STAY) = 0$
- $degree(CHORD)$  および  $degree(SCALE)$  は、前の音と現在の音の間にあるコードおよびスケールを構成する音の数に1を加えた値をとる。ただし、前の音より現在の音が低い時はこの値に  $-1$  をかけた値をとる。
- $degree(SCALE)$  は、前の音と現在の音の間にある音の数に1を加えた値をとる。ただし、前の音より現在の音が低い時はこの値に  $-1$  をかけた値をとる。

### 3.2.2 音程付加モデルの定式化

確率モデルによる音程付加モデル  $M_{interval} = (R, B)$  を以下のように定義する。

- (1)  $R = \{u(x, i, l, g)\}$ : 状態の有限集合。  
 ここで、 $g$  は現在の音が拍子の表である場合は真、そうでない場合は偽となる真偽値である。 $u(x, i, l, g)$  は、前の音の音程変化が  $x$ , かつ、 $degree(x) = i$  であり、現在の音の長さが  $l$ , かつ拍子の表かどうか  $g$  である状態を示す。ただし、本研究では、標準 MIDI ファイルを用いるの

で  $-127 \leq i \leq 127$  である.

- (2)  $B = \{o(y, j|x, i, l, g)\}$ : 記号出力確率分布.  
 $o(y, j|x, i, l, g)$  は, 状態  $u(x, i, l, g)$  において  $degree(y) = j$  となる  $y$  を出力する確率を示す.
- (3)  $K = \{\rho_0, \dots, \rho_{127}\}$ : 初期音高確率分布.  
 $\rho_i$  は旋律が MIDI ノートナンバー  $i$  で始まる確率である.

### 3.2.3 音程付加のアルゴリズム

リズム生成のアルゴリズムと同様に, 本研究では新しい旋律を生成するために, 旋律の始めの音高, および, 付加する音程は, 初期音高確率分布, および, 記号出力確率分布に従うランダムウォークにより決定する.

### 3.3 リズム生成モデルと音程付加モデルの連携

逐次的リズム音程生成モデルでは, 前述したように, リズム生成モデルが生成したリズムに対して, 音程付加モデルが音程を付加して旋律の生成とする. 図3にリズムに音程を付加する例を示す. 図で示された例は, 発音時刻の量子化の単位が480, かつ, 拍子が4分の4拍子における例である. 図中の上部にはリズム生成モデルにより生成されたリズムが示されている. そして, 図中の下部には, 音程付加モデルにより音程が付加されたリズム (つまり旋律) が, スケールがCメジャースケール, かつ, コードがFの場合で示されている. まず, 音程付加の準備として, 生成されたリズムの最初の音は,  $x = STAY, degree(x) = 0$  とする. 次に2番目の音に音程を付加する際には, 前の音 (つまり最初の音) の音程を示す  $STAY$ , および,  $degree(STAY) = 0$  と, 現在の音 (つまり2番目の音) のリズムを示す  $FALSE$  (発音タイミングが拍子の表でない), および, 360 (音の長さ) を現在の状態 (つまり  $o(STAY, 0, 360, FALSE)$ ) として現在の音の音程を出力確率分布をもとに出力する. 図では  $x = SCALE, degree(x) = 1$  が出力されている. 以下, 同様に全ての音に対しての音程を出力することでリズムに音程を付加していく.

### 3.4 二つのルールの導入

明らかに不自然な旋律の生成を防ぐため, 2つのルールを設定する.

- (1) リズムの長さについて: リズム生成において生成されるリズムの長さを  $L$ , 学習データとして用いる演奏事例にある旋律の最低の長さを  $min(D)$ , 最大の長さを  $MAX(D)$  とする. 生成されるリズムの長さ  $L$  が  $min(D) \leq L \leq MAX(D)$  を満たさない場合は, 再度, リズム生成をやり直す.
- (2) 音程について: 音程付加によりリズムに付加された音程を MIDI ノートナンバーで表した値の最大値

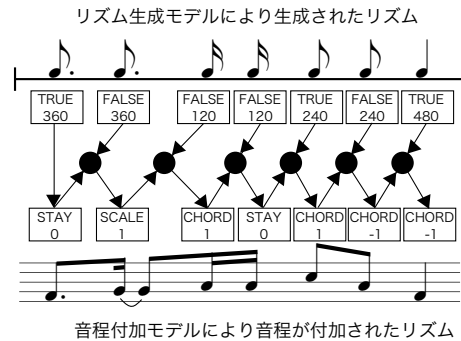


図3 リズムに音程を付加する例

を  $MAX(I)$ , 最小値を  $min(I)$  とする. また, 学習データとして用いる演奏事例を  $J$  とする.  $J$  にある旋律の音を MIDI ノートナンバーで表した値の最大値を  $MAX(J)$ , 最小値を  $min(J)$  とする. リズムに付加された音程が  $MAX(I) \leq MAX(J)$  かつ  $min(J) \leq min(I)$  を満たさない場合は, 再度, 音程付加をやり直す.

## 4. 自動作曲システムの実装

自動作曲システムとして, 逐次的リズム音程生成モデルを計算機上に実装する. 本研究では, コード進行に調和する単音旋律を出力する自動作曲システムを構築する. 本システムでは, コード進行とスケールを入力として用いる. また, 逐次的リズム音程生成モデルの確率分布を学習させるためのデータとして, 単音旋律, スケール, コード進行, 旋律の開始を示すデータ, および, 旋律の終端を示すデータを含む演奏記録を用いる. 本システムでは, 所々のデータの形式として, (1) 入出力ファイル形式: 標準 MIDI ファイルのフォーマット 0, (2) コード, スケール, 旋律の開始, 旋律の終端の記述形式: メタイベントのタイプ 5, の形式をとる. MIDI に関しては MIDI 1.0 規格に従う [6]. 本システムを OS が Mac OS X バージョン 10.3.9, プロセッサが 1.33 GHz PowerPC G4, メモリが 768MB DDR SDRAM の計算機上で Java 1.3.1 により実装した.

## 5. 実 験

計算機上に実装された逐次的リズム音程生成モデルに基づく自動作曲システムによる旋律の自動生成を行った. まず, 逐次的リズム音程生成モデルの確率分布を学習させるため, 学習データとして, ジャズピアノの教本を執筆している藤井氏の著書より, “Just Friends”, “Ladybird”, “On A Slow Boat To China”, “The Lady Is Tramp”, および, “Yesterdays” の計 5 曲を用いた [7]. 各曲に対して, 前処理として次の処理を人手で行った.



図4 実験で生成された旋律

- (1) 単音旋律として、大譜表の高音部譜表にある最高音を抽出する。
- (2) コード、および、スケールを記述する。
- (3) 旋律を適当な長さに区切る。ただし、旋律を構成する音が3つ以上、かつ、旋律の長さが8小節以下となるようにする。
- (4) 区切られた旋律の各々に対して、最初の音の位置に旋律の開始、そして、最後の音の位置に旋律の終端を記述する。

前処理を行ったデータを学習データとして確率分布を学習させた。そして、システムの入力にはコード進行として“Autumn Leaves”のコード進行を冒頭部分の8小節、および、スケールとしてB<sup>b</sup>メジャースケールを入力した[8]。本システムでは、生成される旋律の長さが一定でない。したがって、実験では入力されたコード進行に対して、十分な長さの旋律が生成できるまで、旋律の生成を繰り返し行い、生成された各々の旋律をつなげたものを得ることとした。図4に実験により生成された旋律を示す。図では、各小節の左側に円で囲まれた小節番号を記してある。また、逐次的リズム音程生成モデルにより生成された旋律の最初の音が“START”で、そして、最後の音が“END”で示されている。

## 6. 考 察

今回の実験により生成された旋律の数は、8小節の

入力に対して2つであった。以下、小節番号*i*の小節を小節*i*と呼ぶ。一つ目の旋律は小節1のSTARTから小節5のENDまでである。二つ目の旋律は小節5のSTARTから小節8のENDまでである。

自動作曲システムの評価は、本来ならば生成された旋律について感性評価を行うべきである。しかし、妥当な感性評価を行うためには、評価者の人数は数十程度では不十分であると考えられる。また、評価者は、評価の目的にあった音楽知識と経験を持ち、かつ、目的にあった地域、または、所属から選ばれる必要がある。本稿では、感性評価は今後の課題とし、感性評価にかわり考察を行うことにする。

### 6.1 一つ目の演奏について

小節1におけるコードは、C<sub>m</sub>7である。小節1内の演奏は、コード音を多用し、コード音をつなぐ音としてスケールの音を用いており、各音同士がよく調和している。また、小節1の4拍目から小節1の最後にかけての演奏は、一つおきに特定のコード音を演奏するといった、一般によく見られる演奏である。

小節2におけるコードは、F7である。ここでの演奏は全体的に緊張感がある演奏となっている。その理由はスケール外の音が3つ現れているからである。しかし、これらのスケール外の音は、1拍目の6連符(2組ある3連符の前の方)の2つめの音のようにコード音を導く、および、4拍目の直前から始まる4つの音(小節2の最後の3音と小節3の最初の音)のような半音進行を行う、といった演奏によく用いられる。したがって、緊張感は持ちつつも全体として外れすぎた演奏にはなっていない。

小節3におけるコードは、B<sup>b</sup>M7である。ここでの演奏で注意すべき点は、アポイドノート(あまり用いない方がよいとされる音)が用いられている点である。小節3におけるアポイドノートはE<sup>b</sup>であり、3拍目と4拍目で用いられている。しかし、その用いられ方は、前後のコード音を結ぶための経過音的な用いられ方であり、音も8分音符とあまり長くない音であることから、演奏の調和を崩さないものとなっている。

また、小節4から小節5にかけての演奏は、それ以前とは違ったリズム感がでており、音使いには跳躍を多用しているので、緊張感が出ている。

以上のように、一つ目の旋律は全体が調和した演奏となっており、ほどよい緊張感も含まれている。

### 6.2 二つ目の旋律について

二つ目の旋律は、コードやスケールと調和しない演奏を含んだ旋律となった。小節5、および、小節6のコードはそれぞれA<sub>m</sub>7b5、および、D7である。ここ

では半音進行が多用されており、コードを無視したような演奏が行われ、非常に緊張感が強くなっている。このような出力が得られた理由は、ここで用いられているコードがB<sup>b</sup>メジャースケールのダイアトニックコードではない(ノンダイアトニックコード)ことだと考える。つまり、コードを構成する音がスケールの音以外の音を多く含んでいる。人間はこのことを考慮し、一時的に違うスケールを用いたりするが、本システムでは、スケールを一つしか指定していないため、特殊な状態遷移が続き、半音進行という特殊な演奏を多用したと思われる。

小節7と小節8のコードは共にG<sub>m</sub>7である。ここでは、コードとスケールに調和しない演奏が行われている。これは前の2小節における特殊な状態の影響を受けて、特殊な状態遷移が続いていると考えられる。

以上のことから、二つ目の旋律からは本手法の改善すべき点が伺えた。特殊な状態の出現により、それ以後の状態遷移に特殊なものが続いてしまい、生成される旋律が不安定になる点である。これの改善策として、システムに対してコード進行だけでなく、スケールについても詳細に入力することが考えられる。しかし、実際にはノンダイアトニックコードにおけるスケールの選択は演奏者により異なるので、演奏事例を構築する際には、どのスケールを意識して演奏したかを演奏者に確認できる必要がある。もう一つの改善策は、演奏事例の拡充が挙げられる。スケールを適切に入力していない演奏事例では、音程変化がHALFであると判定されることが多くなる。このことが半音進行の増大につながると考えられるが、演奏事例を多くすることにより、特殊な事例も多く集まるので、特殊な状態に適切に対応できる可能性がある。

### 6.3 手法について

実験結果で出力された旋律に用いられていた半音進行、アポイドノート、および、一般的に用いられるような形の演奏といったものは、音楽知識として明確に記述するのは困難である。また、これらを音楽知識として記述し、ルールや制約として用いる場合も、その適用方法によっては、演奏の特徴を失わせてしまうため、扱いが難しい。我々が提案する逐次的リズム音程生成手法は、このような音楽知識を必要としない。音楽知識の記述無しで、以上のような旋律が生成できたのは、演奏記録に基づいて、リズムを生成した後に音程を付加することによって、拍の表や裏、音の長さといった、演奏の印象に大きく影響を与える要素を適切に処理できたためだと考える。

一方、旋律を切り出す部分(実験における前処理(3)

および(4))には曖昧さが残っている。旋律を切り出すには、どこからどこまでが旋律であるかを判断する必要があるが、特に明確な判断基準がある訳でもない。旋律の切り出しの自動化については、楽曲の構造解析に関する研究の応用が考えられる[9]。

また、音楽におけるマルコフ性について、連続する3つ程度の音が楽曲の特徴を最もよく表すという実験結果もある[10]。これらの研究を本手法に応用することにより、さらに自然な旋律の生成ができる可能性も考えられる。

## 7. まとめ

本稿では、旋律を創造する過程をリズム生成の過程と音程付加の過程に分けてモデル化した、逐次的リズム音程生成モデルの提案を行い、自動作曲システムの構築を行った。そして、旋律の自動生成実験を行いその有効性を確認した。本研究は旋律の生成をリズム生成の過程と音程付加の過程に分けてモデル化した点で新しい。また、曖昧かつ複雑な音楽理論を必要としない点で有用である。システムの完成評価、演奏事例からの旋律の切り出しの自動化、および、より自然な旋律の生成は今後の課題とする。

## 参考文献

- [1] 山田拓志, 椎塚久雄, “遺伝的アルゴリズムを用いた自動作曲について,” 情報処理学会 音楽情報科学研究会 研究報告 1998-MUS-027, Vol.1998, No.96, pp.7-14(1998).
- [2] 三重野芳典, 椎塚久雄, “自動作曲システム,” 情報処理学会 音楽情報科学研究会 研究報告 2000-MUS-035, Vol.2000, No.49, pp.25-30(2000).
- [3] 川村修, 伊藤孝行, 新谷虎松, “演奏事例ベースに基づくアドリブ演奏模倣システムの実装,” 第66回情報処理学会全国大会論文集, pp.439-440(2004).
- [4] 大内東, 山本雅人, 川村秀憲, “マルチエージェントシステムの基礎と応用,” コロナ社, 2002.
- [5] 川上源一, “実用ポピュラー音楽編曲法上巻,” 財団法人ヤマハ音楽振興会, 1988.
- [6] 中島安貴彦, “MIDI バイブル II MIDI 1.0 規格実用編,” リットーミュージック, 1998.
- [7] 藤井英一, “実例ジャズピアノアドリブ集2,” リズムエコーズ, 1999.
- [8] 高島慶司, “スタンダード・ジャズのすべて(1),” 全音楽譜出版社.
- [9] 平田圭二, 青柳龍也, “音楽理論 GTTM に基づく多声音楽の表現手法と基本演算,” 情報処理学会論文誌, Vol.43, No.2, pp.277-286(2002).
- [10] 熊本忠彦, 大田公子, “印象に基づく楽曲検索: 楽曲印象値の自動付与,” 情報処理学会 データベースシステム研究会 研究報告, 2002-DBS-127, No.41, pp.89-96(2002).