

平成 24 年度 卒業論文

ギター演奏者のためのリアルタイム ベースライン生成システム

指導教員 北原鉄朗専任講師

日本大学文理学部情報システム解析学科

大塚匡紀

2013 年 2 月 提出

概 要

伴奏を行うと同時に演奏者が自らメロディーを行う演奏形態を弾き語りという。合奏相手がいなくても成立し、特に、ギターにおいて一本で行える手軽さから様々な場面で行われている。しかし、手軽さの反面、ベースやドラムなどと合奏するバンド形式に比べると演奏できる楽曲の特徴が限定されてしまう。演奏データを前もって用意することによって計算機を用いてバンド形式での演奏も可能になるが、演奏データを準備できない状態では対応できない。そこで、本研究では、演奏データの準備なしでバンド形式でのセッションシステムを目指す。その一環としてリアルタイムでベースラインを生成するシステムを提案する。しかし、前持ったコード進行の情報なしに次々と変化する曲の展開に適切なベースラインを付与するのは難しい。そこで、旋律やコード進行の繰り返しである「リフ」を利用してベースライン生成を行う。ポピュラー音楽ではギターによるリフは数多くの楽曲で演奏されている。さらに、リフが楽曲の大半で繰り返し演奏され楽曲の顔となっている楽曲が多く見られる。本研究では、リフに着目したリアルタイムでのベースライン生成を目指すシステムを開発し、実行結果からベースラインの音楽的妥当性を確認した。

目 次

目 次	iii
図 目 次	v
第 1 章 序 論	1
1.1 本研究の背景	1
1.2 本研究の目的	2
1.3 本論文の構成	3
第 2 章 関連研究	5
2.1 バンドライクなインタラクティブ作曲支援システム [1]	5
2.2 バンドライクな音楽アシスタントシステムについて [2]	6
2.3 Guitarist Simulator : 演奏者の振舞いを統計的に学習するジャムセッ ションシステム [3]	6
2.4 Interactive Music System : Cypher[4]	7
2.5 Nueuro-Musician[5]	8
2.6 仮想ジャズセッションシステム : Virja Session[6]	9
2.7 BayesianBand : 旋律の予測に基づいた自動伴奏システム [7]	9
2.8 まとめ	10
第 3 章 本研究における方針	11
3.1 ベースライン生成の課題	11

3.2	想定する楽曲の構成	12
3.3	システム構成	14
第 4 章	システム詳細	19
4.1	ギターによるリフ演奏の取得	19
4.2	旋律かコードかの判定	19
4.3	旋律リフの場合のベースライン生成	20
4.4	コードリフの場合のベースライン生成	21
4.4.1	コード認識	21
4.4.2	リズム認識	21
4.4.3	ベースライン生成	22
4.5	MIDI ペダルキーボードによるコード変更	22
第 5 章	実行結果	25
第 6 章	おわりに	31
	参考文献	33

目 次

3.1	リフの例	12
3.2	想定する楽曲構成例	12
3.3	The Beatles : Day Tripper のリフ	13
3.4	システム構成	14
3.5	システムの主な流れ	16
3.6	MIDI ギター	17
3.7	MIDI ペダルキーボード	18
4.1	クオンタイズ	19
4.2	旋律リフの例	20
4.3	コードリフの例	20
4.4	旋律におけるベースラインの生成例	20
4.5	旋律におけるベースラインの生成例	22
4.6	コード変更後のベースラインの例	23
5.1	試用に用いる楽曲のギター演奏	27
5.2	図 5.1 の楽譜の続き	28
5.3	ベースライン生成結果	29

第1章 序 論

1.1 本研究の背景

ポピュラー音楽においてギター人気は高く、様々な楽曲において、ギターを使った演奏を聞くことができる。人気の理由として、以下のようなギターの特徴が考えられる。1つ目に、肩にストラップをかけることにより立ちながら演奏することが可能であり、コンパクトで持ち運びに困らないことからどこでも演奏できる利便性がある。ストリートミュージシャンにギター演奏者が多いことからそのようなことが伺える。2つ目に、コードカッティングなどのギター独自の演奏方法や音色の豊かさがある。最後に、ポピュラー音楽においての使用頻度が高いなどが考えられる。これらの特徴を生かした演奏方法の一つに「弾き語り」がある。弾き語りとは歌唱が伴う楽曲の演奏形態であり、演奏者が演奏をしながら自らメロディを歌うもので、合奏相手がいなくても成立するため、様々な場面で行われている。ギター以外の楽器でも演奏可能であるが、ギターを使っの弾き語りの多さは顕著である。しかし、ベースやドラムスなどを伴ってバンド形式で演奏するのに比べ、ギターでは、演奏できる楽曲の特徴が限定されるのは否めない。その理由として音域の狭さが挙げられる。実際の例として、ピアノの音域がA0～C8なのに比べ、ギターの音域はE2～C6なことから、ギターのみでは低音域がカバーできない。このような問題が発生するため通常のバンド編成では、ベースパートが欠かせない。ベースパートが加わることによってギターには出せない低音域を補うことが可能になり、楽曲の厚みが増すことが予想される。弾き語りをする場合、楽曲があらかじめ決まっていれば、演奏データをあらかじめ打ち込むことで、

計算機と合奏することもできるが、演奏データの打ち込みには労力を必要とする。さらに、即興演奏などの演奏データが未入力 of 楽曲には対応することができない。

1.2 本研究の目的

本研究では、あらかじめ演奏データを入力していなくても、ユーザの弾き語りに対してリアルタイムに伴奏を付与するセッションシステムの実現を目指す。この一環として、本稿では、ギターの弾き語りに対して、その演奏に合うベースラインをリアルタイムに生成するシステムを提案する。これまでセッションシステムに関しては様々な研究が行われてきた。高柳らは、バンド形式に近い作曲支援システムを提案した [1]。しかし、コード進行は予め決まっており、リアルタイム性も検討されていなかった。青野らは、バンド形式に近いセッションシステムを提案した [2]。ユーザの演奏からコード進行を随時予測して伴奏を付与するシステムであるが、ベースラインの生成に関しては未検討である。セッションシステムの多くに共通して言えることだが、演奏入力を「ピアノ」と想定している研究が多く、「ギター」を想定しているものが少ない。浜中らが、ギターを対象としたセッションシステムを提案した [3] が、ギター演奏者のモデル化が目的であり、ギター演奏からのベースラインの生成は検討していない。

本研究では、コード進行などの情報が一切事前に与えられていないことを前提とするため、任意の楽曲に対して適切にベースラインを付与するのは極めて難しい。しかし、ポピュラー音楽には、「リフ」と呼ばれる数小節の旋律やコードの演奏が、楽曲の最初から最後まで繰り返され、それが楽曲の特徴を形作っているものも数多く存在する。そこで、このような楽曲を対象を絞り、まず最初の 4 小節でユーザにリフを演奏してもらい、それ以降は、それが繰り返されるという前提でベースラインを生成する。

1.3 本論文の構成

本論文は以下の構成とする。2章では、関連研究を述べ、3章では、本研究が想定する楽曲の構成を述べ、そこからベースライン生成を実現するためのシステムにおける必要な機能を記す。4章では、システムの詳細を述べ、具体的なベースラインの生成方法を記す。次に5章では、実際のギター演奏結果からのベースライン生成の結果を載せ、考察を述べる。最後に6章において、本研究の結論、今後の課題を述べる。

第2章 関連研究

ここでは、セッションシステムの関連研究について述べ、そのシステムの課題を述べる。

2.1 バンドライクなインタラクティブ作曲支援システム [1]

初心者でも手軽に作曲・編曲を楽しめる環境を提供することを目的に作られたシステムである。音楽経験がない初心者を対象に、感性レベルで作曲作業のできる環境を目指している。ドラムスパート、バックিংキーボードパート、ヴォーカルパートに相当するメロディパートの3つがあり、ユーザが入力したメロディパートのシーケンスに伴奏のアレンジを付与するものである。伴奏のアレンジにおける、コード付けとリズムのアクセントの「定形」、「非定形」を定義する。ユーザが曲全体の定形度を指定でき、特定のジャンルにとらわれない、幅の広いアレンジを可能にしている。しかし、ベースパートの生成はなく、リアルタイム性は検討されていない。

2.2 バンドライクな音楽アシスタントシステムについて

[2]

ユーザが、コンピュータを他のメンバーとして構成されるバンドの一員となりセッションを行う。バンドらしさを実現するために「アクセントパターン抽出機能」、「コード認識機能」、「フレーズ抽出機能」、「4小節単位のシーケンス抽出機能、及び保存機能」、「シーケンスの変化を予測する機能」、「任意にシーケンスを組み立てることによる簡易作曲機能」の6つの機能があり、バンド感覚でセッションを行うことを可能にしている。ユーザとコンピュータが双方でインタラクティブ性を感じながらセッションを行えるシステムである。

2.3 Guitarist Simulator : 演奏者の振舞いを統計的に学習するジャムセッションシステム [3]

ある人間の演奏者を模倣した仮想演奏者と、人間の演奏者とがインタラクティブに即興演奏できるジャムセッションである。人間の演奏者は個性を持っており、相手の演奏に対してどのような演奏で反応するかは異なっている。この研究では、それを「演奏者の振る舞い」と呼ぶ。従来のシステムでは、仮想演奏者がどのような振る舞いをするかは、ルール群などによって決定されており、実在する人間の演奏者の振る舞いをさせることは困難であった。この研究では、演奏者の入出力関係を統計的に学習することにより演奏者の振る舞いモデルの獲得を手法を提案している。システムの概要は、ギタートリオで12小節のブルース進行上でセッションを行い演奏者の振る舞いをモデル化する。演奏楽器にキーボードを想定した研究が多い中、ギターを想定した研究である。しかし、ギタリストの振る舞いのモデル化に主観が置かれているため、ギター演奏からのベースラインの生成は未検討である。

2.4 Interactive Music System : Cypher[4]

コンピュータを1人の作曲家兼演奏家として考え、人とコンピュータとがインタラクション(相互作用)を行いながら、人とコンピュータとの共作による音楽を創作することを目的とした研究。この研究では、音楽聴取と解析部である「リスナ」と演奏生成部である「プレイヤ」が独立しており、リスナ、プレイヤ共に2段階の階層的な構造をもっている。リスナの低次の解析部分では、音の特徴が調べられる。その際に、以下の7つの特徴量を解析する。

1. density : 音の密度、単音もしくは和音。
2. attack speed : 速度、1つ前のイベントと現在のイベントの発音時刻の間隔。
3. dynamic : 音のダイナミクス、音量。
4. resister : イベントの属する音域。
5. duration : 音の長さ。
6. harmony : 和声。
7. beat tracking : 拍位置検出、拍位置の間隔からテンポを取得する。

リスナの高次の解析部分では、低次の解析部によって得られた音の特徴量を用いて2つの処理を行う。1つ目に、フレーズの抽出を行う。2つめに、抽出されたフレーズから、規則性また不規則的に変化する特徴量を発見する処理で行う。高次のリスナの2つの処理で、音の規則性、方向性を知ることによってその後の音楽の流れを予想することを目的としている。

次に、リスナ部分について述べる。この研究では、作曲方法として、入力された音楽を様々な方法で変形させて出力する方法をとっている。まず、低次の変形処理ではアクセントを与える、コードを分散して演奏、装飾音を付加、クレシェンド、デクレッシェンドなどの変形を行う。高次の変形処理では、フレーズを超え

た長時間の処理を行い、ベース演奏を開始、一定の頻度でランダムな音程変化などの変形を行う。

cypher によって生成される演奏は、ジャムセッションのような演奏とは種類の異なるものであるが、音楽的特徴量の抽出の処理は非常に参考になる。

2.5 Nueuro-Musician[5]

このシステムは、ニューラルネットワークをベースとしたセッションシステムで、8小節や16小節の長さの即興ソロを、複数人のミュージシャンの間で交互に行う「掛け合い」を人間とコンピュータの間で行うシステムである。相手が提示したソロ演奏をモチーフとして、旋律的・リズム的特徴を持ちつつ、変形を加えていくことが掛け合いの醍醐味である。

Neuro-Musician では、入力層・隠れ層・出力層の3層からなるニューラルネットワークを用いている。約30の入力演奏と出力演奏例の対が与えられて、バックプロパゲーション法でそれぞれの層を形成している。

このシステムでは、掛け合いによるソロ演奏においては、以下の4つの情報が重要であるとしている。

1. 旋律の概形
2. 音程の変化とリズム
3. 発音タイミング
4. コード進行とそのアベイラブルノートスケール

4つの情報を基にしてニューラルネットワークを用いて演奏を生成する。このシステムでは、ユーザが演奏を行う際に実装されたベーシストとドラマーがサポートを行う。演奏情報は、コード進行とともにそれに基づいたMIDIデータの演奏情報を入力しておく必要がある。

2.6 仮想ジャズセッションシステム：Virja Session[6]

人間とコンピュータとが影響を与え合いながら演奏する新しい統合演奏環境を実現することが目的の研究。ジャズでは演奏者間のインタラクションが重要であり、コンピュータと人間とがお互いの演奏を聞き合ってインタラクションするためにコンピュータグラフィックスを用いてコンピュータの演奏者の姿を視覚化し、ジェスチャーをカメラ等を通じて確認できる。こうして、すべてのプレイヤー間で、聴覚と視覚の両方を通じたマルチモーダルインタラクションを実現する。

このシステムでは、ジャズのピアノトリオの演奏を対象とし、人間がピアニスト、コンピュータがベーシストとドラマーを担当し、テンポが一定で演奏される4ビートのジャズスタンダード曲を扱う。あらかじめ、システムにコード進行やテーマ旋律のデータを入力し、テーマやソロの順番を規定するためのデータも入力する。計算負荷の分散のために、ネットワークに接続された複数のコンピュータ上に実装しており、ベーシストとドラマーを独立したプロセスとして実現し、さらにこれらの演奏理解・生成部とCG表示部も別々のプロセスとしている。

このシステムの最大の特徴は、演奏者間の主従関係が対等な関係なセッションが行えるである。

2.7 BayesianBand：旋律の予測に基づいた自動伴奏システム [7]

ユーザの演奏をリアルタイムに予測して伴奏を付与する自動伴奏システムである。演奏家が、初めて聞く楽曲でも、自身の経験に基づいて次の展開を予想し、伴奏をすることができる。このシステムは、こういった音楽家の能力である「相手の行動を予測する」といったものを計算機上で実現することである。

このシステムでは、コード進行も旋律の情報も事前に一切与えない状態で、ユー

ザが演奏する未知の旋律に対して、その次の展開を予測して、リアルタイムでコードを推定し、伴奏を行う。推定の際に、ベイジアン・ネットワークを用いてユーザの旋律を予測し、音楽的に自然なコード進行を出力する。ユーザはシステムが決定するコード進行を予測しながら即興演奏をすることで、相互予測が調和したときには「親近性」による快感を覚える。また、調和しなかったときでも、音楽的にはそれほど不自然ではないとき、「新奇性」による快感を覚える。上記の2つの快感を感じながらインタラクションを取り入れたジャムセッションが可能である。

2.8 まとめ

本稿で紹介した関連研究において、リアルタイムでのベースライン生成を扱ったものは、「バンドライクな音楽アシスタントシステム」[2]、「Cypher」[4]、Nueuro-Musician[5]、「Virja Session」[6]である。

本研究では、コード進行の情報を事前に一切与えない状態でリアルタイムでのベースライン生成を行う。ベースライン生成を行うシステムの中で「バンドライクなインタラクティブ作曲支援システム」[2]は本研究に近く、コード進行が未知の状態でベースラインを生成する。

しかし、ベースラインを演奏から抽出という流れ（ピアノの演奏ならば左手の最低音をベースラインとして抽出）のため、ギター演奏を基に1からベースラインを生成する本研究とは方針が異なる。

第3章 本研究における方針

本章では、初めに本研究でのベースライン生成の際の方針を述べ、次に、ベースライン生成の際の方針から想定される楽曲の構成を述べ、最後に、対象楽曲に対してリアルタイムにベースラインを生成するためのシステムを構成を述べる。

3.1 ベースライン生成の課題

コード進行を事前に与えられていない状態でリアルタイムでベースライン生成を行うのは難しい。何故なら、次々と変化する曲の展開に適切なベースラインを付与するには、曲の展開を予測する必要がある。さらに、リアルタイムでベースを演奏する際に生成に遅れが生じるのは致命的である。よって、展開を予測し、なおかつベースライン生成の遅れを解消する必要があると考えられる。

そこで、ポピュラー音楽における「リフ」(図:3.1)に着目した。リフとは、数小節の旋律やコードの演奏であり、歌の伴奏として成り立つような、コード感を持ちつつ、メロディも感じられるフレーズになることが多い。リフを繰り返し演奏することにより、楽曲の印象を強く根付かせ、リフを聞けば、その楽曲が思い浮かぶと言ったことがある。その結果、曲を覚えやすくすると言った効果が考えられる。

以上のような理由から、ポピュラー音楽ではひとつのリフがバンド全体で楽曲の大部分で繰り返され、楽曲の「顔」となり「骨」となっているものが多く見受けられる。

本研究では、上記のようなリフの性質を利用して構成される楽曲に対象を絞り、

リアルタイムでのベースライン生成を目指す。



図 3.1: リフの例

3.2 想定する楽曲の構成

本研究では、例として、図 3.2 に示す、リフを中心とした楽曲を対象とする。

	イントロ	イントロ2	Aメロ	Bメロ	アウトロ
ボーカル	なし	なし	あり	あり	なし
ギター	メイン リフ	メイン リフ	メインリフ をアレンジ	Am-F-G-C などの コード進行	メイン リフ
小節数	4小節	4小節	16小節	16小節	8小節
ベース	なし	メイン リフ	メイン リフ	コード進行に 伴った ベースライン	メイン リフ
ドラム	バスドラ のみ	8ビート ドラム パターン	8ビート ドラム パターン	8ビート ドラム パターン	8ビート ドラム パターン

図 3.2: 想定する楽曲構成例

1. ギターによるリフ演奏のみからスタート（イントロ）。
2. そのリフを全員（ギター、ベースなど）でユニゾン演奏する（イントロ2）。
3. Aメロが始まると、ベースはそのままリフ演奏を続け、それが基本のベースラインとなる（Aメロ）。
4. Aメロでは、ギターはコードのカッティングなどリフ以外の演奏を行う（Aメロ）。

5. サビでは、コード進行が変化する (B メロ)。
6. サビの後は、リフのユニゾン演奏に戻る (アウトロ)。
7. その後は、2 番の A メロに入ったり、ギターソロがあったりして、エンディングを迎える。

実在するリフを中心とした楽曲の具体例として、The Beatles の Day Tripper がある。イントロにおいて、演奏されたリフ (図 3.3) が楽曲の大部分で繰り返し演奏される。



図 3.3: The Beatles : Day Tripper のリフ

図 3.2 で示した楽曲構成に近く、以下に示す構成になる

1. ギターによるリフ演奏のみからスタート (イントロ)。
2. そのリフをギターとベースでユニゾン演奏する (イントロ 2)。
3. ギターとベースでの演奏ドラムが加わり、ユニゾン演奏をする (イントロ 3)。
4. A メロが始まると、ギターとベースはそのままリフ演奏を続け、それが基本のベースラインとなる (A メロ)。
5. A メロの途中の 2 小節のみリフが変化する (A メロ 2)。
6. サビでは、コード進行が変化する (B メロ)。
7. A メロに戻り、B メロへと続く。
8. ギターソロの演奏がある (C メロ)。
9. A メロに戻り、B メロへと続く。
10. リフのユニゾン演奏に戻る (アウトロ)。

3.3 システム構成

本研究で想定する楽曲に対してリアルタイムにベースラインを生成するため、図3.4に示す構成とする。

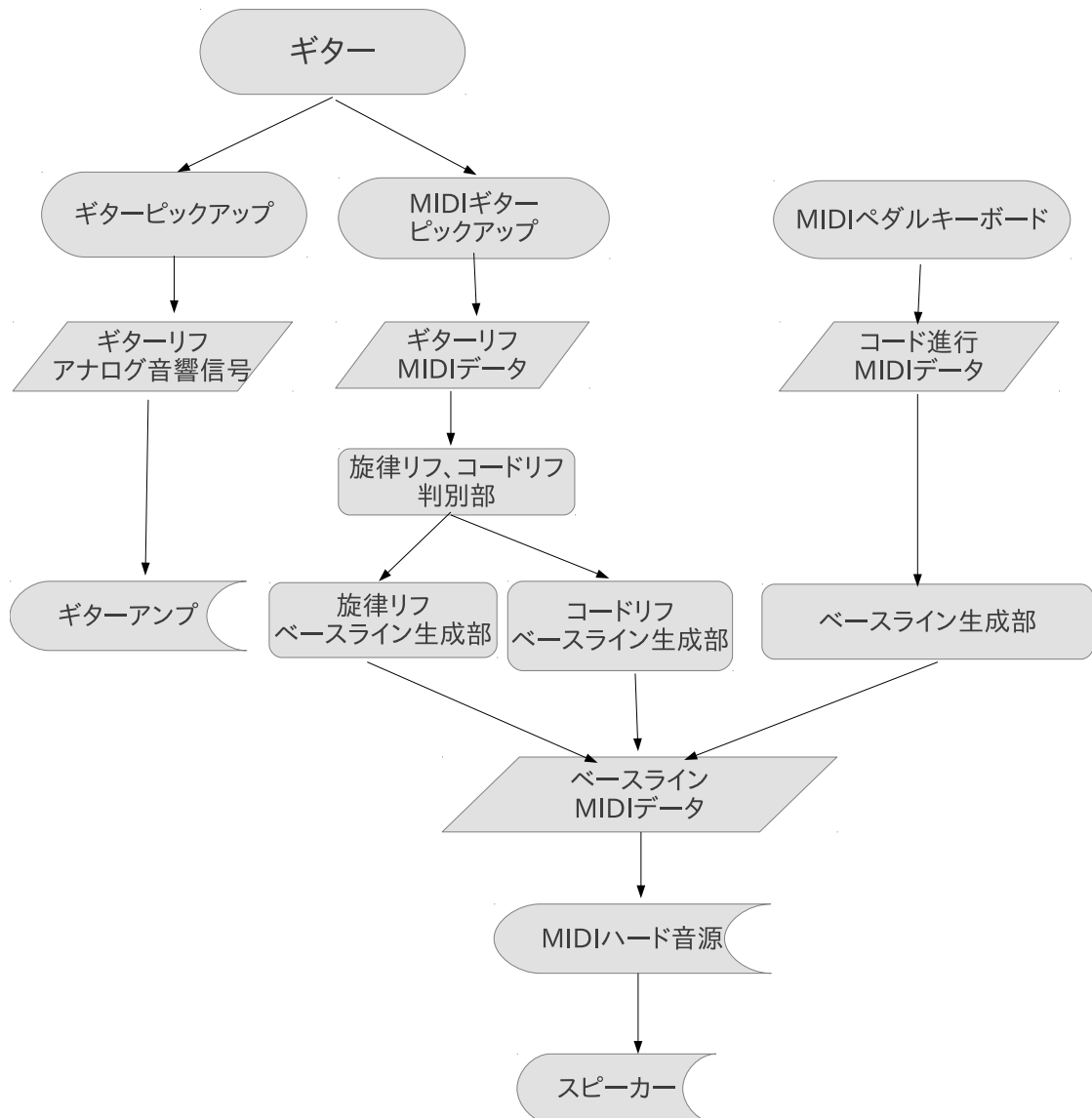


図 3.4: システム構成

- ギターの演奏には MIDI ギター (図 3.6)[8] を用いるものとする。ただし、ギターの MIDI ピックアップはギターの演奏内容をシステムが取得するためだけに用い、ギター演奏者には、音色にこだわりを持っている者が多く見られることからギターアンプにはギター本体の通常のピックアップの出力を接続し、音色の設定は自由に行う。
- 最初の 4 小節は、ギターが単独でリフを演奏するものとし、システムはギターの演奏内容を取得する。その間、ベースは演奏されない。
- 5 小節め以降は、ギターの 1~4 小節めの演奏に基づいてベースが演奏を始める。1~4 小節めのギターの演奏が単音をベースとした旋律的な演奏であれば、この旋律を、音域を下げた上でそのまま模倣する。コード演奏の場合は、コード名を認識してそのルート音を演奏する。
- 途中でコード進行が変わる場合は、MIDI ペダルキーボード (図 3.7)[9] を用いてコードが変わる直前 (1~2 拍前) にコードのルート音を指示する。ただし、現状では 4 分の 4 拍子を前提とし、各小節の 1 拍めと 3 拍めでのみコードを変えられるものとする。
- 楽曲全体を通して、ドラムパートはシステムが既定のパターンを演奏するものとし、ギターはそのテンポに合わせて演奏するものとする。
- 入力される演奏のキーは A_{m} を前提とする。

よって、システムの主な流れは (図 3.5) になる

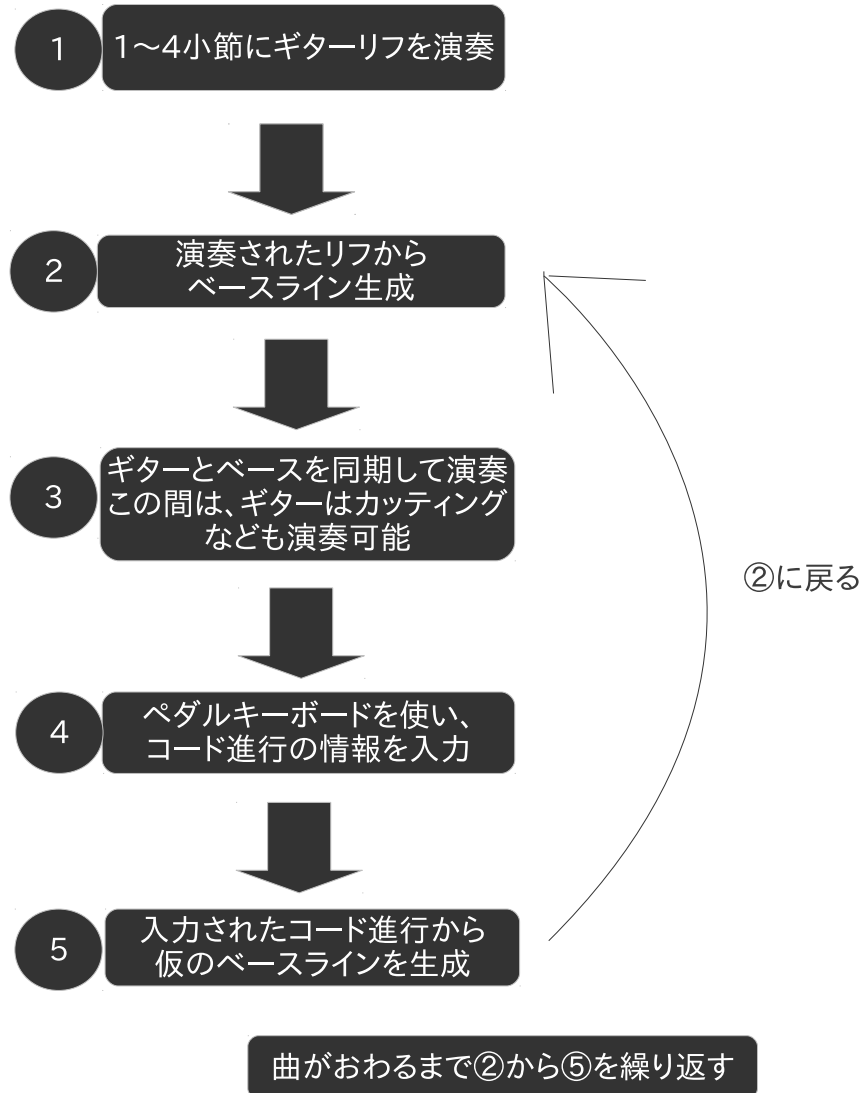


図 3.5: システムの主な流れ



図 3.6: MIDI ギター



図 3.7: MIDI ペダルキーボード

第4章 システム詳細

本章では、ベースライン生成の具体的な方法を詳細に述べる。

4.1 ギターによるリフ演奏の取得

最初の4小節は、ベースラインを生成せずに MIDI ギターから得られた演奏情報を取得する。その際、演奏情報は発音のタイミングやズレを修正するクオンタイズが行われる。16 分音符にクオンタイズを行い、16 分音符より細かいリズムは強制的に 16 分音符に修正される。

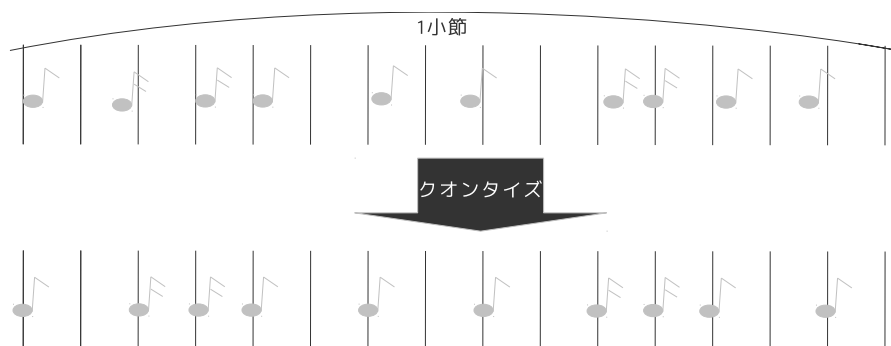


図 4.1: クオンタイズ

4.2 旋律かコードかの判定

取得した演奏情報が、単音が主で音の動きが多くメロディアスなリフ (図 4.2) (「旋律リフ」と呼ぶ) かコード演奏を主に使ったリフ (図 4.3) (「コードリフ」と呼ぶ) かを判定する。

判定方法は、16分音符ごとに同時発音数を計算し、各小節で同時発音数が2以上の時刻が50%以上あるとき、その小節をコードリフと判定し、そうでないものを旋律リフと判定する。



図 4.2: 旋律リフの例



図 4.3: コードリフの例

4.3 旋律リフの場合のベースライン生成

旋律リフの場合は、その演奏内容をそのまま音域だけを下げたものをベースラインとする (図 4.5)。

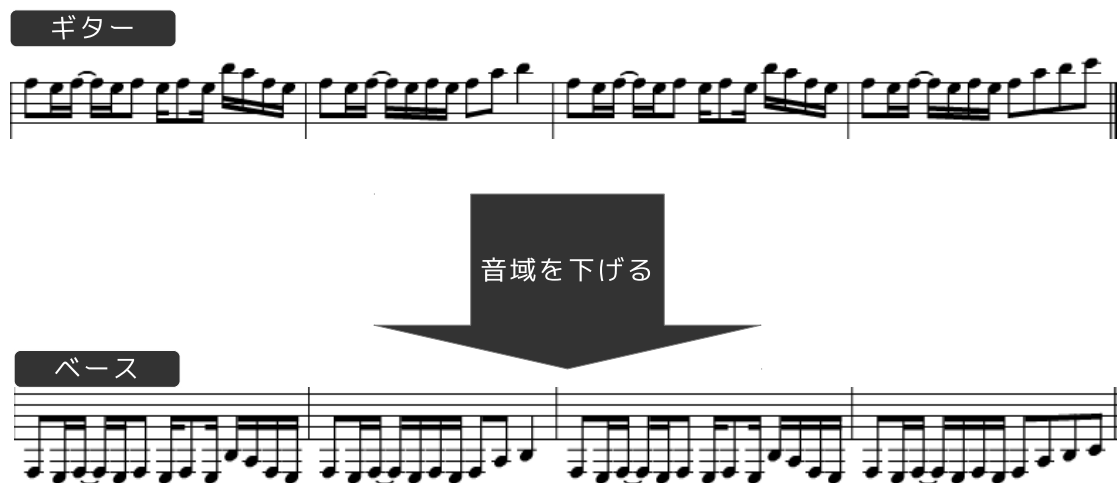


図 4.4: 旋律におけるベースラインの生成例

4.4 コードリフの場合のベースライン生成

コードリフの場合は、次の手法でコードとリズムを認識し、その結果を元にベースラインを生成する。

4.4.1 コード認識

認識対象を A_m のキーのダイアトニックコードとパワーコードの 14 種類に限定し、各コード区間 (2 拍ごと) に対して、次の手順によりコードを認識をする。

1. 認識対象のコードごとに、そのコードらしさを表す変数 $x_1, \dots, x_{14}$ を用意し、0 で初期化する。
2. 発音中の音の音長を音名ごとに累積する。
3. 音長累積値が高い順に音名を 4 つ取得する。
4. 取得した 4 つの音名の各々に対して、その音を構成音として含むコードに対応する x_i に 1 を足す。
5. x_i が最大のコードに対して、
 - $x_i \geq 2$ なら x_i を認識結果とする。ただし、 $x_i = 2$ で候補が複数ある場合はパワーコードを優先する。
 - $x_i = 1$ の場合は、コードは不明とする。

4.4.2 リズム認識

コード演奏の発音時刻を取得する。発音時刻は、16 分音符ごとにクオンタイズされた演奏データに対して同時発音数を求め、これがしきい値以上の場合に、その時刻を発音時刻とする。ただし、カッティング時のゴーストノートなどを検出しないように、音長に比例した重みをかけた上で、しきい値処理を行う。

4.4.3 ベースライン生成

??で検出された各発音時刻において、??節で認識されたコードのルート音を演奏する。

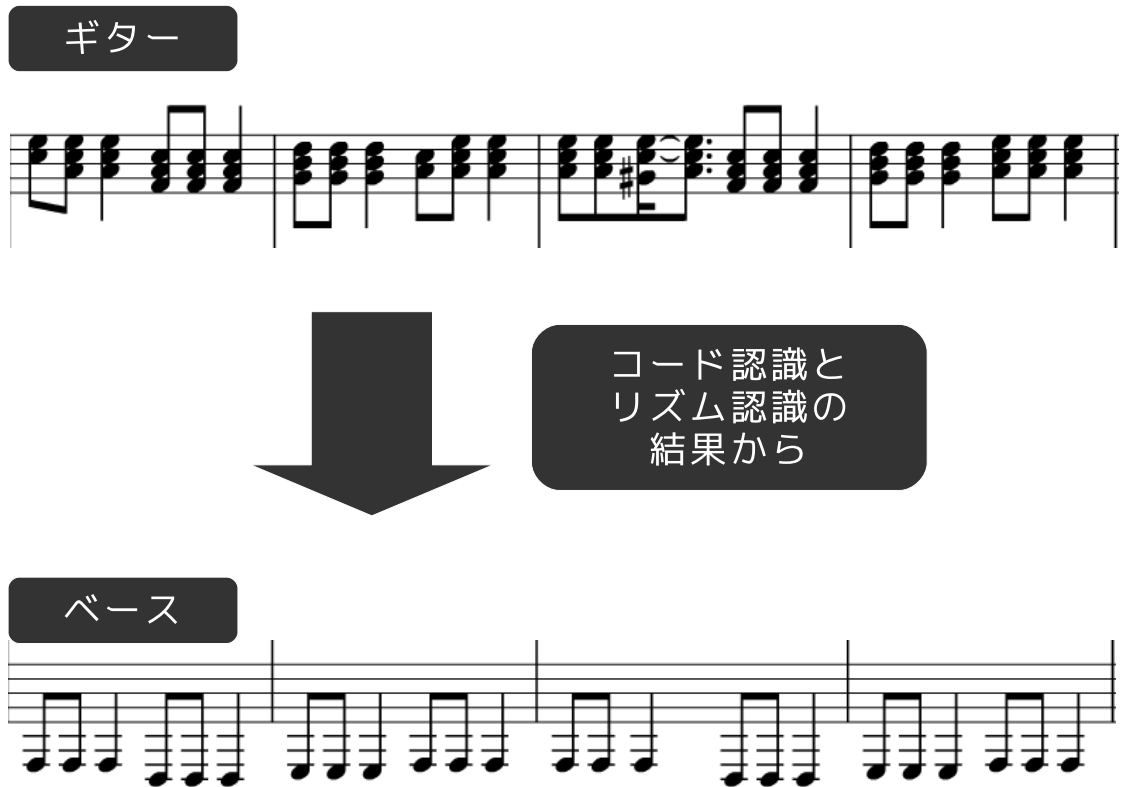


図 4.5: 旋律におけるベースラインの生成例

4.5 MIDI ペダルキーボードによるコード変更

コード進行を変更したい場合は、変更したいタイミングの直前2拍以内に MIDI ペダルキーボードを用いて変更後のコードのルート音を入力する。現状では、コードチェンジは各小節の1拍めと3拍めに限定している。MIDI ペダルキーボードによる入力があると、その次にコードチェンジできるタイミングで、コードが変化

する。このようにしてコードが変えられた場合、コードのルート音が4分音符で演奏される。



図 4.6: コード変更後のベースラインの例

最初のリフに戻る場合は、MIDI ペダルキーボードのコントロールボタンを押すことで、最初のリフに戻ることができる。

第5章 実行結果

本章では、試用楽曲を用いてリアルタイムでのベースライン生成を検証する。

図 5.1, 図 5.2 が試用楽曲のギター演奏の楽譜である。

試用楽曲は、イントロのリフが楽曲の大部分で演奏される楽曲構成になっている。リアルタイムでのベースライン生成を検証するために、MIDI ペダルキーボードを使ったコード進行の変更があり、楽曲の途中でコード進行が変化する楽曲構成になっている。

この楽曲を実際に演奏して得られたベースラインを図 5.3 に示す。これより、次のことが分かる。

- 最初の 4 小節はベースパートは演奏されず、5 小節目以降は図 5.1 に示したギターパートの冒頭 4 小節のリフがオクターブが下げられて、演奏された。
- コード進行が変更される直前の 2 拍前に MIDI ペダルキーボードを使いコード進行の情報を入力し、“B” の冒頭の 4 小節はコード進行のルート音を 4 分音符のリズムで鳴らした仮のベースラインを生成している。
- “B” の 5 小節目以降は、図 5.1 の “B” の冒頭 4 小節のギター演奏を基にコード進行とリズムを正確に認識し、コード進行のルート音をギターリフと同じリズムでベースラインを生成している。
- “Bridge” に移行する前に、MIDI ペダルキーボードのコントロール・チェンジボタンを押したことにより、リフが呼び出された。

- “B” に移行する前に、MIDI ペダルキーボードで入力されたコード進行が “B” と同じだったので、記録しておいた “B” でのベースラインが演奏された。
- “Outro” に移行する前に、MIDI ペダルキーボードのコントロール・チェンジボタンを押したことにより、リフが再び呼び出された。

Intro



Intro2

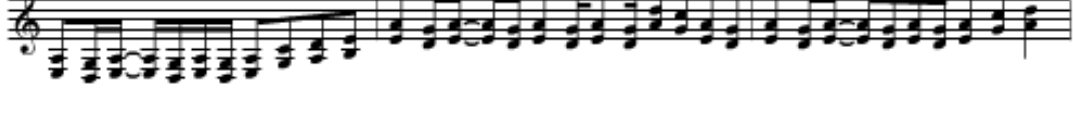


A

Am Am Am Am Am



Am



B

Am F G Am Am F G Am Am F G Am Am F G Am



図 5.1: 試用に用いる楽曲のギター演奏

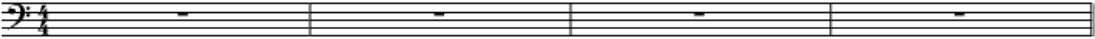
Bridge

B'
Am F G Am Am F G Am Am F G Am Am F G Am

Outro

図 5.2: 図 5.1 の楽譜の続き

Intro



Intro2~A



$\text{B}^{\flat}$ Am F G Am Am F G Am



Am F G Am Am F G Am



Bridge



$\text{B}^{\flat}$ Am F G Am Am F G Am



Outro



図 5.3: ベースライン生成結果

第6章 おわりに

本稿では、リフが中心となっている楽曲を対象とし、ギター演奏に自動的にベースラインを付与するシステムを提案した。本システムでは、最初の4小節のギターリフが楽曲の大半でループされることを前提にした楽曲を対象とし、演奏されたリフが旋律リフかコードリフかを判定し、旋律リフなら音域を下げたベースライン、コードリフならコード認識とリズム認識を行い、その結果からコードのルート音をギターリフのリズム通りにベースラインを生成を行った。一時的にコード進行を変更する際にも、MIDI ペダルキーボードを用いることでベースパートを途切れることなく演奏を行い、結果として、ギター演奏者を対象とした、リアルタイムでのベースラインの生成を可能とした。

しかし、現状の課題として、4小節単位のリフのしか対応できておらず、それ以外のものには対応できない。MIDI ギターの演奏入力に対する、ゴーストノートの処理の向上、演奏入力の補正ができていない。簡単なフレーズに対するベースライン生成はうまくいったが、リズムの難しい演奏などにはベースライン生成がうまくいかなかった。コードリフとペダルキーボードによってコード進行が入力された際にルート音のみが演奏される、旋律リフには音域を下げたベースラインのみなどが考えられる。

本研究の今後の方向性として、一つにベースラインの改善があげられる。先程、述べた通り現在のベースライン生成は単純なアルゴリズムで行なっている。今後は、ルールや確率モデルなどを導入して適切な経過音が挿入された高度なベースライン生成を目指したい。また、ドラムパートは現在規定のパターン繰り返すの

みであり、リフのリズムを考慮したドラムパターンの生成、曲の流れに合わせてドラムパターンの変更などが考えられる。最後に、もっと幅広い楽曲構成への対応を行いたい。

これらを通じて、ギター演奏者と計算機の間でのバンド演奏が、人間同士の本物のバンド演奏により近づけるようなセッションシステムの開発を目指したい。

参考文献

- [1] 高柳 剛, 片寄 晴弘, 井口 征士, “バンドライクなインタラクティブ作曲支援システム”, 情報処理学会研究報告, MUS-21-9, 1997.
- [2] 青野 裕司, 片寄 晴弘, 井口 征士, “バンドライクな音楽アシスタントシステムについて”, 情報処理学会研究報告, MUS-8-9, 1994 .
- [3] 浜中 雅俊, 後藤 真孝, 麻生 英樹, 大津 展之, “Guitarist Simulator : 演奏者の振舞いを統計的に学習するジャムセッションシステム”, 情報処理学会誌, vol.45 No.3, 2004 .
- [4] R.Rowe, “Interactive Music Systems ”, The MIT Press, 1993.
- [5] 渡辺和之, 西嶋正子, 柿本正憲, 村上公一, “ニューラルネットワークを用いたジャズセッションシステム-ニューロミュージシャン-”, 情報処学会研究報告, 2-199, 1997.
- [6] 後藤 真孝, 日高 伊佐夫, 松本 英明, 黒田 洋介, 村岡 洋一, “仮想ジャズセッションシステム : VirJa Session” , 情報処理学会誌, vol.40 No.4, 1999 .
- [7] 北原 鉄朗, 戸谷 直之, 徳網亮輔, 片寄 晴弘, “バンドライクなインタラクティブ作曲支援システム”, 情報処理学会論文誌, vol.50 No.12, 2009.
- [8] Roland:GK-3, “<http://www.roland.co.jp/>”
- [9] Hammond:XPK-100, “<http://www.hammondorganco.com/>”

謝 辞

本論文制作にあたり、昼夜問わず、お忙しい中、北原鉄朗専任講師から熱心な指導をいただきました。鈴木峻平氏、土屋裕一氏から貴重なアドバイスをいただきました。研究室の同期、後輩の方々にも数多くの支援をいただきました。自分を支えてくれたすべての方々に感謝します。