

## dial : 音・映像構成のためのロータリーシーケンサ

的 場 寛<sup>†1</sup> 中村 滋 延<sup>†2</sup>

《dial》は複数の反復する時間をもとに音・映像を構成するシーケンスソフトである。個々に独立な時間軸をもつ回転運動を 3D 空間に配置し組み合わせることで並列的なリズムや回転する音場の生成を試みる。

### dial : A Rotary Sequencer for Sound and Visual Composition

HIROSHI MATOBA<sup>†1</sup> and SHIGENOBU NAKAMURA<sup>†2</sup>

《dial》 is a sequence software for audiovisual composition using loops. We attempt to make complicated polyrhythm and rotating surround acoustic field by putting and combining circular movements which has independent timeline in 3D graphical score.

#### 1. はじめに

音楽は 3 次元空間に時間を加えた 4 次元での空気振動の出来事と言えるが、人間の分かりやすさのために、または紙というメディアの制限から 2 次元の楽譜に情報を変換し記述される。その際もとの次元が含んでいた情報の多くは捨て去られるか、もしくは値を変換、または固定することで簡略化される。例えば五線譜の場合、時間の情報はそのまま楽譜の横方向の 1 次元に割り当てられるが、音の空間での定位情報は演奏者が動き回ることは滅多にないことから、パラメータは固定されて楽譜には表記されない。そのかわり楽器がふるわせる空気の周波数 (音高) を楽譜の縦軸の次元に割り当てることで、空気の振動現象を簡略

的に表している。

このように楽譜は複雑な空気の現象のひとつの捉え方にすぎないにもかかわらず、一部のグラフィックスコアを除いてこれまでの楽譜ではいつもまったく同じ表記方法 (横軸に時間、縦軸に音高という直線的な表記法) を利用してきた。私たちの話す言語が身の回りの世界の認識に深くかかわっているように、音楽の言語である記譜法は音楽の発想、構成法に強く関わり、音楽の思考に大きな影響を与えている。記譜法を再考することでこれまでにない音楽の発想や構成が生まれ、新しい表現へつなげる可能性がある。

筆者らは五線譜のような直線的な表記にかわる方法として、円形楽譜の記譜法に注目し、時間軸を円として表わすシーケンスソフトウェア Overbug<sup>\*1</sup> を開発してきた。

円形楽譜のような時間を円で表わす記譜法を円形記譜法 (Circular Notation System) と呼ぶことにする。さらに円を使った記譜法の中でもリアルタイムの CG や映像などの動的なメディアを使用した楽譜を動的円形記譜法 (Dynamic Circular Notation system/Rotary Notation System)、紙媒体のような静的なメディアを利用した円形記譜法を静的円形記譜法 (Static Circular Notation System) と呼びそれぞれ区別することにする。

Overbug の開発により、円形記譜法はリズムの直観的把握、ポリリズムな表現、リズムの位相ずれの表現において大きな利点があることがわかっている。現在、Overbug の円形記譜法を発展させたシーケンスソフトウェア《dial》を制作している。本稿では《dial》の制作を通して明らかになった円形記譜法の新たな可能性を論じる。

#### 2. 円形記譜法

Terry Rirey の《Keyboard study No.1》(図 1.a) や George Crumb の《Makrokosmos Vol.1, No.8 The Magic Circle of Infinity [Leo]》(図 1.b)、武満徹の《ピアニストのためのコロナ》(図 1.c)、《弦楽器のためのコロナ II》(図 1.d)、などの円形楽譜の作品がある。これらはすべて紙に印刷された静的な楽譜だが、Overbug の円形楽譜は PC を利用し、グラフィックをリアルタイムにディスプレイに表示する。Overbug ではインジケータの回転をアニメーションとして表示可能であり、動的円形楽譜と言える。また同時に音と映像を構造

<sup>†1</sup> 九州大学大学院芸術工学府  
Graduate School of Design, Kyushu University

<sup>†2</sup> 九州大学大学院芸術工学府  
Faculty of Design, Kyushu University

\*1 Overbug は [www.dominofactory.net](http://www.dominofactory.net) からダウンロードできる。また以下の文献がある。的場寛・中村滋延: ループの構築と崩壊による音楽構成, 情報処理学会研究報告 2008-MUS-077, 情報処理学会, pp. 37-40 (2009). 的場寛: ループインターフェイスを用いた音楽演奏ツールの設計, 九州大学大学院芸術工学府平成 20 年度修士論文 (2009). 的場寛・中村滋延: ループインターフェイスによる音・映像構成, 芸術工学会誌 No. 51, 芸術工学会, pp. 42-43 (2009).

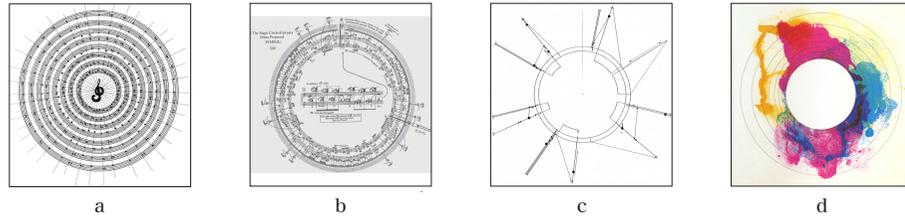


図 1 円形楽譜  
Fig. 1 Circular Score

的に出力することで、音のみの時と比べて観客の鑑賞の理解度を向上させ、なおかつオーディオビジュアル作品として密接にリンクした音と映像によるあらたな価値を生むことを目指している。本稿では Overbug について詳細に述べないが、演奏ソフトウェアアートとして「音楽発生の仕組み」、「音楽的特徴」、「画面構成」に着目した分析<sup>1)</sup>と Overbug の円形記譜法についての分析をそれぞれ表 1 と表 2 に示す。

表 1 Overbug の分析  
Table 1 Analysis of Overbug

音楽発生の仕組み	
入力媒体	マウス入力、補助入力としてキーボード
入力操作	ドラッグおよびクリック、補助入力としてキpの押下
入力対象	図=円
発音のタイミング	虫インジケータの走行後
音楽的特徴	
音素材	リズム、音色選択可
音楽的外見	変奏反復、声部数は無限
音楽形式	反復音楽
音楽を語るキーワード	音の雲、音響テクスチャ
画面構成	
画面の基本構成	任意の場所に、任意の大きさで、任意の数で描かれる円
発音時の可視化	サウンドポイントに輪が発生
音楽発生の枠組みの可視化	リズム中心の反復音楽、反復の際の様々な変奏・変化の加え方、反復単位の無制限な増加、などが可視化
画面構成の入力への関与	音楽発生の仕組みの可視化により、次の入力への刺激

表 2 Overbug の円形記譜法  
Table 2 Circular Notation System of Overbug

タイムラインの数	0 - 無限
タイムラインの同期方法	すべての円で各速度が同じ
シーケンスの周期	すべての円で同じ。 ただし円を自動回転させることで個別に設定することも可能。
2D/3D	2D *1
音源	wav ファイルを再生
音高	円の大きさ
音価	固定
音強	インジケータ (円周上の虫) の大きさ
音色	wav ファイルの音
定位	画面の位置 (左右)

Overbug の分析と十数回ライブパフォーマンスを行った経験から、Overbug の動的円形記譜法の 1 つの大きな問題点がわかってきた。「音高の変化が乏しいため反復をそのままにしておくとなんとなく単調になってしまう<sup>1)</sup>」ことである。音高の表現力に加えて、反復されるフレーズ、リズムの周期がすべての円で同じであることも反復が単調になってしまう原因として考えられる。Overbug のタイムラインの同期方法は「すべての円で各速度が同じ」であるため、シーケンスの周期が「すべての円で同じ」になっている。タイムラインの同期方法を「すべての円で角速度が同じ」ではなく「すべての円で速度が同じ」にすることで各タイムラインのシーケンスの長さが一定でなくなり、すべてのシーケンスの長さの最小公倍数の周期でリズムが変化し続けるシステムを作ることが可能であると考えられる。

また Overbug は 2D の CG だったが、3DCG に発展させることで、楽譜にサラウンドサウンドの情報を記述できるようになり、さらに複数のスクリーンを使用することで VR 的な没入型の映像出力を得られることが予想される。

引き続き動的円形記譜法について深く可能性を探り、検証するために下記 2 つの改良を加えたシステムを持つ作品《dial》を制作した。

- すべての円で速度が同じである同期方法
- 3DCG の利用  
次章より《dial》について説明する。

### 3. ロータリーシーケンサ 《dial》の制作

《dial》は発音システム、入力インターフェイス、API、音源などのシステムから構成される。(図2)

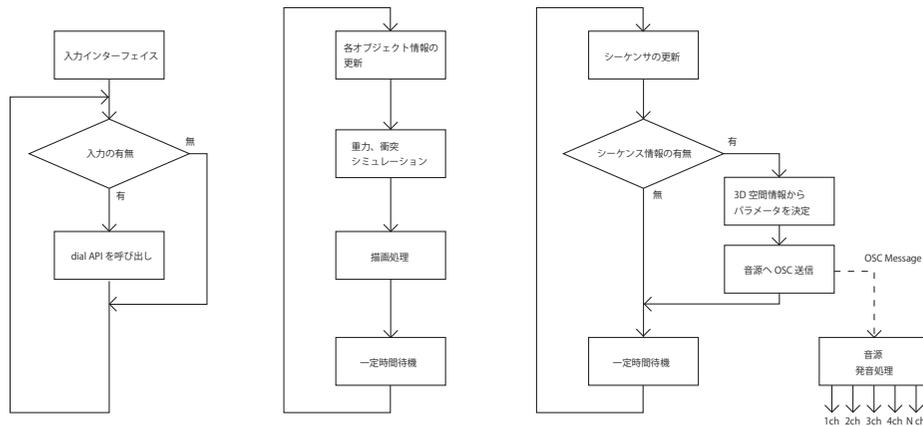


図2 《dial》のシステムフローチャート  
Fig.2 System Flow Chart of 《dial》

それぞれのセクションについて説明する。

#### 3.1 発音システム

##### 3.1.1 《dial》のオブジェクト

《dial》は図3に示すように Fixed Star Object(恒星オブジェクト)、Planet Object(惑星オブジェクト)の2つのオブジェクトで構成される。各オブジェクトについて説明する。

- Fixed Star Object(恒星オブジェクト)  
仮想宇宙上に静止し、その他の惑星オブジェクトの中心に位置するオブジェクト。
- Planet Object(惑星オブジェクト)  
周期的に恒星または他の惑星の周りを公転しているオブジェクト。各惑星により異なる公転周期、公転軸を持つ。音楽的には惑星オブジェクトはシーケンサのインジケータの役割を担う。また惑星オブジェクトは Orbit(軌道)と Trigger(トリガ)の子要素を含む。

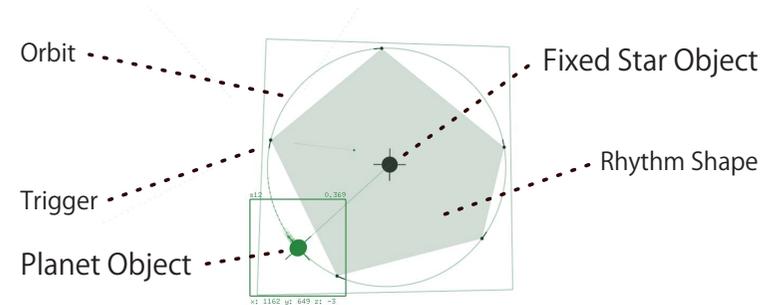


図3 《dial》のオブジェクト  
Fig.3 Objects of 《dial》

Orbit(軌道)は惑星オブジェクトの公転軌道を意味し、軌道の半径により惑星オブジェクトの公転周期が決まる。音楽的には公転周期はシーケンサの長さを意味する。半径が大きくなればなるほど、1周する時間が長くなる。また、新しく追加される Orbit は必ず単位半径の整数倍の半径を持つように設定される。そのため、すべてのシーケンサ周期の関係は必ず整数比となる。

Trigger(トリガ)は軌道オブジェクト上に配置することができるシーケンサポイントであり、音楽的には音符の役割を持つ。Orbit に Trigger を3つ以上追加するとそれぞれの Trigger が線で結ばれ円に内接する多角形、Rhythm Shape(リズムシェイプ)が表示される。Rhythm Shape の面積や形を見ることで Orbit 上の Trigger の密度やバランスなどのシーケンサ情報を素早く捉えることができる。

##### 3.1.2 発音

上述したオブジェクトを配置して発生させる。Planet Object が Trigger の上を通過するときに OSC メッセージを発行する。その様子を図4に示す。

発音される音のパラメータなど、音源については後ほど詳しく述べる。

#### 3.2 《dial》API

外部のソフトウェアやハードウェアなど様々な入力インターフェイスから、《dial》を使用できるように機能を API としてまとめた。表3に代表的な API を挙げる。

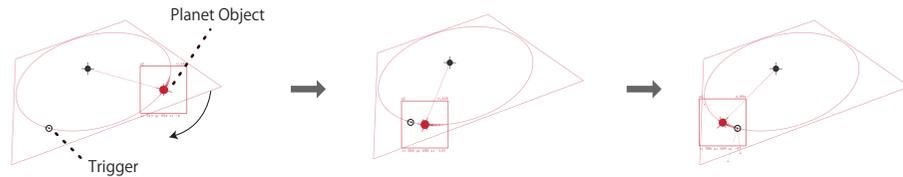


図 4 OSC メッセージの発行  
Fig. 4 Sending OSC Message

表 3 《dial》の API  
Table 3 《dial》API

コマンド (パラメータ)	機能
createFixedStar(int x, int y, int z)	Fixed Star Object(恒星) を生成する。
addPlanet(string Type, int radius)	Fixed Star Object(恒星) に Planet Object(惑星) を追加する。
rotate(int speed, string axis)	Planet Object(惑星) の公転軸を回転させる。
reverse(void)	Planet Object(惑星) の公転方向を逆方向に変更する。
moveTo(int x, int y, int z)	Fixed Star, Planet Object(恒星・惑星) を指定した位置に移動する。
addSound(int noteNumber)	Orbit(軌道) に Trigger(トリガ) を追加する

### 3.3 入力インターフェイス

ダイアルの API を呼び出す入力インターフェイスにはマウスやキーボード、MIDI コントローラ、センサなど様々な方法が考えられるが、今回は図 5 のようにアプリケーション内の CUI(コマンドラインユーザインターフェイス) から API を呼び出す方法を採用した。

### 3.4 音源

《dial》では v002(バージョン 002) からシーケンスのシステムと音源を分離して設計した。分離したことで《dial》と並行して既存の DAW<sup>\*1</sup>ソフトウェアを実行し発音するなど、さまざまな使用法をサポートできる。開発を進めながら次の 3 つの音源を試した。

サウンドファイルの再生 《dial v001》 あらかじめ用意したサウンドファイル (.wav) を 《dial》シーケンスからトリガし、再生した。画面内のサテライト・オブジェクトの位置からユーザとの距離、方向を計算し、ステレオスピーカの音量と定位を制御した。

Cycling'74 Max5 《dial v002》 Cycling'74 社の Max5 を使用し、距離、方向に加え、

\*1 Digital Audio Workstation



図 5 《dial》の CUI  
Fig. 5 《dial》CUI

サテライト・オブジェクトの円軌道上の位置、回転スピードを Max5 へ送信し、それらの情報から音を生成して出力した。プロトコルには OSC(Open Sound Control) を利用した。また画面の背景色の RGB 値により出力の最終段のエフェクトをコントロールできるようにした。またサウンドサウンドに対応した。

Ableton Live! max for live 《dial v003》 Ableton 社が開発しているループベース・シーケンサ Live!を利用した。max for Live プラグインを利用し《dial》から OSC 情報を Live!へ送信し発音する。パラメータは距離、方向、惑星オブジェクトの円軌道上の位置、回転スピードを使用した。Live!でもサウンドサウンドに対応した。

## 4. 考 察

### 4.1 プログラム実装の考察

《dial》に下記 2 つのシステムを実装し、動作を確認した。

- すべての円で速度が同じである同期システム  
すべての Planet Object は同じ速さで Orbit 上を進むよう設計・実装した。Orbit 半径の異なる、複数の Planet Object を配置すると Orbit 半径の最終公倍数の時間 (拍数) でシーケンスの組み合わせが変化してゆき、単調なループに陥る問題を解決できた。しかし、Overbug の場合、複数の円上に適当に音を配置してもそれなりにまとまりのある反復フレーズを構成することができたが、《dial》ではシーケンス同士の周期比を考慮して、的確にシーケンスデータ (Trigger) を構成しないとランダムに音が発生しているように聞こえてしまう、という問題が出てきた。

この問題はタイムラインが単純な反復周期の場合と複雑な反復周期の場合の間にあるトレードオフと言えるだろう。つまり、単純な反復は構成しやすく、複雑な反復は構成しづらい。円形記譜法では角速度同期の場合が単純反復であり、速度同期の場合が複雑な反復となる。このことは「円形記譜法の複数タイムラインの同期方法の違いによる特徴」といえる。

#### ● 3DCG の利用

3DCG を描画し、3D 空間上で音楽が構成できるように設計・実装した。Overbug では決められた平面上にしか円を配置できなかったが、《dial》では 3D 空間のどこにでも円を配置したり、移動させたり、回転させたりできる。3D 空間での Trigger と視点の位置関係（方向と距離）からサラウンドサウンドシステムの各スピーカーの定位を計算することで、映像と音が高度に結びついた出力を得ることができた。また、3D 空間になったことにより空間上の位置  $(x, y, z)$ 、回転軸の向き  $(x, y, z)$  など円形記譜法でハンドルすることができるパラメータが飛躍的に増加した。《dial》v002(バージョン 002)以降からはそのパラメータを音源のソフトウェア側でシンセサイザやエフェクトのパラメータにアサインして発音したところ、出力音の周期的変化と Planet Object の回転運動が同期し、音が物質になったかのような表現が可能となった。

#### 4.2 発表

《dial》はこれまでライブパフォーマンス、展示ともに数回の発表の機会を得た。まだ開発段階であるためその都度システム、音源ともに大きく変更しながら発表を行った。その際観客からは、「途中から今どこで音がなっているのか、わからなくなった」、「グラフィックが洗練されている」、「宇宙の音楽のようだ」、「CUI の使い方がわからない」、「CUI の英語が読めない」、「宇宙のようなグラフィックとピアノの音が意外な組み合わせだが心地よい (v001)」などの感想があった。

感想からもわかるように、3DCG を利用したことの弊害として 2D に比べて OSC がトリガされる瞬間を捉えづらくなったことが言える。CUI については初めから予想したことはあったが、多くのプログラミング経験のない鑑賞者は難しいと感じたようだった。Overbug はマウス操作が基本だったため、小学生でもアプリケーションを操作したり鑑賞したりすることが可能だったが、CUI を使ったことで英語の知識がない人にはそれが大きな障壁になった。

#### 5. おわりに

過去に行った円形記譜法の研究をさらに進めるために、Overbug から明らかになった円形記譜法の利点、問題、また新しいアイデアを、実際に動作するシステムを作って、発表し、検証した。

タイムライン上を進む速度を同じとする複数タイムラインの同期方法の実装、発表からは単調な反復を避けることができる半面、複雑になりすぎるといった新たな問題が生じた。また Overbug の同期方法と比較することで、円形記譜法の同期方法の違いによる影響が明らかになった。

円形記譜法のグラフィックに 3DCG を使った実装、発表からはサラウンドサウンドやマルチプロジェクションなどの表現の幅が広がったが、音がどこから発生しているかわかりづらい、という問題が生じた。

CUI を利用した入力には開発時には便利であるが、GUI に比べて理解されづらい。幸い《dial》API として機能をまとめているため、入力インターフェイスの部分は容易に取りかえができる。本稿では取り上げなかったが、現在ロータリーエンコーダによる入力を発想し、専用コントローラを開発している。記譜法のシステムを研究しつつも、それに最適なインターフェイスについても引き続き開発していく。

音源とシーケンス部を分離したことにより、より早く、イメージに近いサウンドデザインが可能になった。また《dial v002, v003》では同じ研究室であり友人の古田伸彦にサウンドパートを担当してもらい、2 台の PC で発表をおこなった。今後も積極的にコラボレーションの機会をつくりたい。

円形記譜法を用いる理由は既存の記譜法からの逸脱、さらには新しい音楽の創出であるため、今後は《dial》のような複雑なタイムラインをどのようにしてうまく構成するかという問題に焦点をあてソフトウェアや作曲法について研究していく。

#### 参考文献

- 1) 中村滋延, 藤岡定, 古田 伸彦, 的場寛: 演奏ツールとしてのソフトウェアアート-その創造性と可能性-, 九州大学大学院芸術工学研究院紀要, Vol.13, pp.9-30 (2010).