

創作ノート

数式によるモーフィングを取り入れた演奏ソフトウェアの設計 DESIGN OF THE PERFORMANCE SOFTWARE WHICH ADOPTED THE MORPHING BY THE NUMERICAL FORMULA

藤原 尚紀, 中村滋延

Takanori FUJIWARA, Shigenobu NAKAMURA

九州大学大学院芸術工学府

Graduate School of Design, Kyushu University

概要

近年、コンピュータの普及により、個人による音楽創作活動が盛んに行われるようになり、アルゴリズム作曲もそのひとつである。アルゴリズム作曲の中には数学的なアプローチを導入している作品も数多く存在する。またコンピュータ技術の発達によって、コンピュータグラフィックスの手法の一つである「モーフィング(morphing)」が音響面にも応用されている。本発表では、筆者が制作した数式によるモーフィングを取り入れた音楽演奏ソフトウェアについて解説し、音楽におけるモーフィングの可能性の一端を探ることを目的とした。この演奏ソフトウェアでは、数式を直接操作することで音楽を制作し、数式を用いて二つの音楽の間をどのようにモーフィングさせるかを決定させた。結果的に音楽におけるモーフィングは二つの音楽の中間的な音楽を複数演奏することが可能になった。また、音楽制作やモーフィングの設定に数式を用いることで楽譜上では表現できないような音楽を演奏することができた。

Widespread use of computers allows individuals to actively create music, including algorithmic composition. Many musical works take a mathematical approach in algorithmic composition. Morphing is a technique by which computer graphics are applied to sound, made possible by the development of computer technology. In this paper, I comment on music performance software that adopts morphing by means of a mathematical formula, and explore the possibilities of morphing in music. I have developed a system that generates pieces of music through direct manipulation of mathematical formulas. The way of morphing from one piece to another is determined by those formulas. As a result, several different pieces on the continuum of two referential pieces can be generated by way

of the morphing. In addition, the mathematical formulas allow the system to perform some types of music which cannot be notated.

1. はじめに

コンピュータの普及によって、アルゴリズム作曲に取り組む人が増加した [1]。アルゴリズム作曲の中には数学的なアプローチを導入している作品も数多く存在する。作曲家ヤニス・クセナキス (Iannis Xenakis) は、確率などの数学の理論を用いて多くの作品を制作した。また、ペンとタブレットを用いてグラフを描くことで音響を作り出す《UPIC》(1977) は、描かれたグラフに対して回転や反転など数学的アプローチを導入していた [2]。

現在のコンピュータ技術の発達によって、コンピュータグラフィックスの手法の一つである「モーフィング」が音響面にも応用されている。Zynaptiq 社が提供しているプラグインソフト《MORPH 2》(2015)¹ は、物理モデルをリアルタイムに演算し、一方のサウンドをもう一方のサウンドにモーフィングさせることができる (図 1) [3]。また、浜中らによって旋律をモーフィングさせるために「タイムスパン木に基づくメロディモーフィング法」が提案されている [4]。

このように音楽のモーフィングは実際の音楽制作に利用されており、さらに音楽経験のない人のために音楽のモーフィングの研究・制作も行われている [5]。

以上のことから筆者は、音楽制作と音楽のモーフィングに数学的アプローチを用いる考えに至り、数式によるモーフィングを取り入れた音楽演奏ソフトウェア《Transition》、《F(music)》、《MetaMath》を制作した。数学的アプローチとして筆者が導入したのは数式によるグラフである。数式によって描かれるグラフから音楽

¹ <http://www.mi7.co.jp/products/zynaptiq/about.php>



図 1. MORPH 2

を複数制作し、数式によって制作した音楽の間をモーフィングさせる。

本稿では筆者が制作した数式によるモーフィングを取り入れた音楽演奏ソフトウェアとそれまでの作品について解説及び考察を目的とし、最後にそれらの作品の開発によって得られたことから音楽におけるモーフィングの可能性の一端を探ることとする。

2. 制作初期における試み《M.M.M.》

筆者が制作した《M.M.M.》(2011)は、《Transition》(2013)の前身となる作品であり、ループ音楽を演奏しながら音を少しずつずらしていくことで、音楽を少しずつ遷移していく作品である。《M.M.M.》とは、Minimal Music Moveの略である。グラフィックやアニメーション処理には、Processing²を、音響生成にはMax5.0³を使用した。ユーザーが画面上に楽器を表している円形のオブジェクトを置いていき、そこを左から右にインジケータが移動していく画面上部およびGUI⁴部に分かれている。インジケータがオブジェクトと重なったときに、OSC⁵を用いてオブジェクトの縦方向の座標から算出した値をProcessingからMax 5.0に送り、その値に対応した音響を発するようになっている。

音楽を遷移させるには、遷移させたいオブジェクトを遷移させたい場所までドラッグする。そのとき、もとの場所から遷移させたい場所まで白線が出る。そのあとオブジェクトは白線をたどって遷移していく。このときもインジケータは動いているので遷移している途中でも音はなり続ける。音のオブジェクトの配置が連続的に変化することで、音列が連続的に変化することになり、遷移過程の音楽を演奏することができる。

《M.M.M.》は、一つの音楽の中を少しずつ遷移させるものであり、ある状態から別の状態へモーフィングさせているとは言えないものだった。その上、遷移の方法も直線的な動きだけであった。

² 統合開発環境 <https://processing.org/>

³ <http://www.mi7.co.jp/products/cycling74/>

⁴ Graphical User Interface

⁵ OpenSoundControl, 音楽演奏データをネットワーク経由でリアルタイムに共有するための通信プロトコル

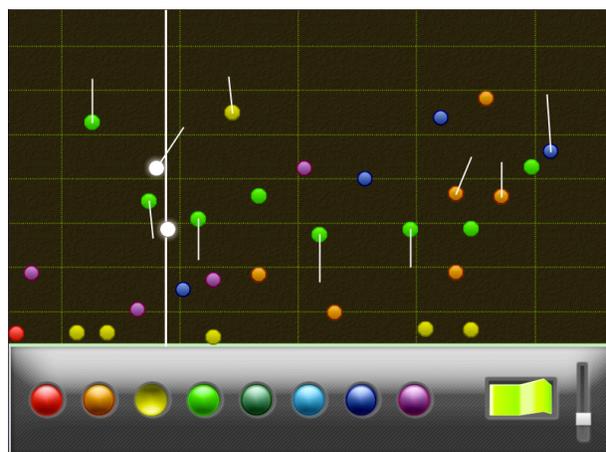


図 2. 《M.M.M.》演奏画面

3. 《TRANSITION》

3.1. 制作指針

《M.M.M.》では、音楽の遷移を取り入れることはできていたが、一つの音楽の間を遷移させていたため、ある状態から別の状態へモーフィングさせているとはいえなかった。また音楽の変化は直線的な変化の一つだけであった。そのため、《Transition》では次の二つを制作指針として挙げた。一つは、二つの音楽を制作し、その二つの音楽の間をモーフィングさせていくこと。次にモーフィングの方法を数式によって決定させることである。

3.2. システム概要

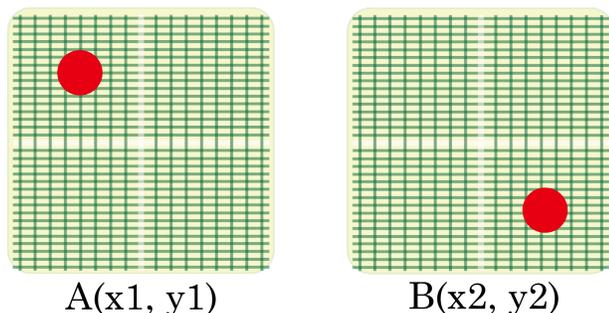
《Transition》は、二つの音楽を制作する composition モードと制作した音楽をモーフィングさせる transition モードに分かれている。

composition モードでは、数学関数の2次元グラフ上に音のオブジェクトを配置することで音楽を制作する。音源は五つある。グラフは二つあり、切り替えながらそれぞれのグラフ上にオブジェクトを配置していく。音楽の制作が完了すると transition モードに移行する。

transition モードは composition モードで制作した二つの音楽同士をモーフィングさせる。二つの音楽をそれぞれ A、B とすると、A の音楽を演奏しながら少しずつ B の音楽に連続的に遷移、つまりモーフィングしていく。モーフィングの方法は詳しくは後述するが、二つのグラフに置かれているオブジェクトの座標から数式を算出し決定される。音楽全体をモーフィングさせることも音源毎にモーフィングさせることも可能になっている。

3.3. 数式の決定方法

《Transition》では、composition モードで二つのグラフ上にみただ領域にオブジェクトを置いていく。そして、transition モードではグラフ上に置かれているそれぞれ対応したオブジェクトの座標から数式を決定していく。対応したオブジェクトの座標とは、二つのグラフ上に置かれている同じ色のオブジェクトの座標である。つまり一つ目の音楽において一番目に置かれた赤の音源と二つ目の音楽において一番目に置かれた赤の音源が対応関係にある。ここでモーフィングの方法が一次関数の場合を例として挙げてみる。図3で示すように二つの音楽 A、B のそれぞれ対応するオブジェクトの座標から一次関数の決定に必要な値を算出する。



$$a = \frac{y1 - y2}{x1 - x2}$$

$$b = \frac{x1 * y2 - x2 * y1}{x1 - x2}$$

$$y = ax + b$$

図3. 数式の決定の様子

3.4. 《Transition》におけるモーフィング

どのようにモーフィングさせるかを決定するために、対応する二つの座標から決定する数式の種類は、一次関数、二次関数の上に凸、下に凸、三次関数の計四種類ある。図4に四種類の数式よりグラフ上をどのようにモーフィングしていくかの例を示す。

3.5. 《Transition》についての考察

《Transition》ではユーザーに二つの音楽を制作してもらい、その二つの音楽の間をモーフィングさせ、二つの音楽の中間にあたる音楽を演奏することができた。《M.M.M.》ではモーフィングの方法は直線的（一次関数）のみであったが、《Transition》では、一次関数、二

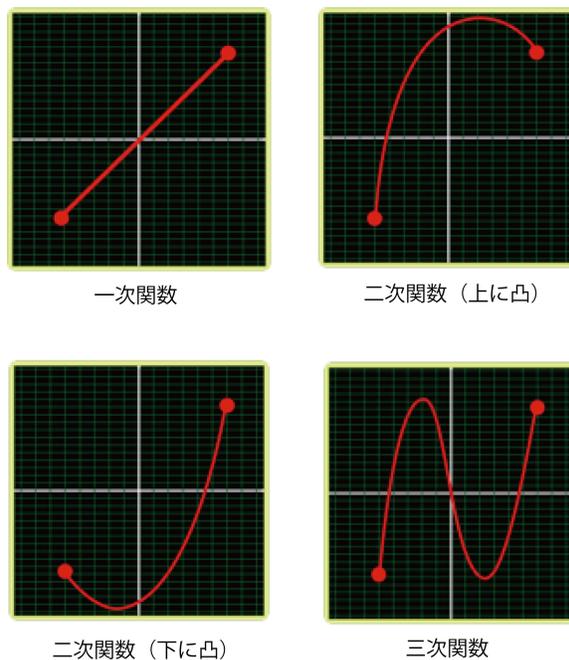


図4. 数式によるモーフィング例

次関数（上に凸、下に凸）、三次関数の四種類実装することができた。モーフィングの種類が増えたことで表現の幅が広がった。また、モーフィングに数式を導入することで楽譜上では表現できない演奏を可能にした。

《Transition》の問題点として挙げられることは三つあると考えられる。まず音楽を制作するときに数式を取り入れていないことである。《Transition》は《M.M.M.》と同様にマウスを用いて音楽を制作している。次に、モーフィングに数式を導入しているが、その数式は表示されないため、ユーザーはどのような数式の上をモーフィングしているかを確認することができないことである。最後にグラフィックのモーフィングを取り入れていないことである。

4. 《F(MUSIC)》

4.1. 制作指針

《Transition》での課題を元に《F(music)》(2014)の制作指針を二つ挙げた。一つは《Transition》では見ることができなかった数式をユーザーに見える形で表示し、その数式から音楽を制作することである。次にグラフィックにモーフィングを取り入れることである。音楽のモーフィングとグラフィックのモーフィングを対応させていくことである。

4.2. システム概要

《F(music)》は、数式によるグラフを楽譜上の音列と捉えることで音楽を演奏するソフトウェアである(図6)。グラフィックに openframeworks[6]⁶ を音響生成には Pure Data[7]⁷ を用いている。また、Pure Data は ofxPd[8]⁸ を導入することにより openframeworks のプログラムの内部に格納している。ユーザーが数式を決定する。それによりグラフが描かれる。グラフが描かれるとその音源の音列が決定される。音源は五つあるので計五つの数式を用いる。

4.3. ユーザーインターフェース

《F(music)》のユーザーインターフェースは演奏画面と数式決定画面の二つがある。

4.3.1. 演奏画面

演奏画面は図5に示しているように主に三つの機能に分かれている。①の部分は数式決定画面への移行ボタンであり、五つのボタンがそれぞれ別々の数式決定画面に移行する。②の部分は音源決定ボタンである。ボタンの色によって音源が決まっており、赤色が sin 波、緑色が三角波、青色がノコギリ波となっている。③の部分は終了ボタンであり、演奏を終了するときを押すところである。

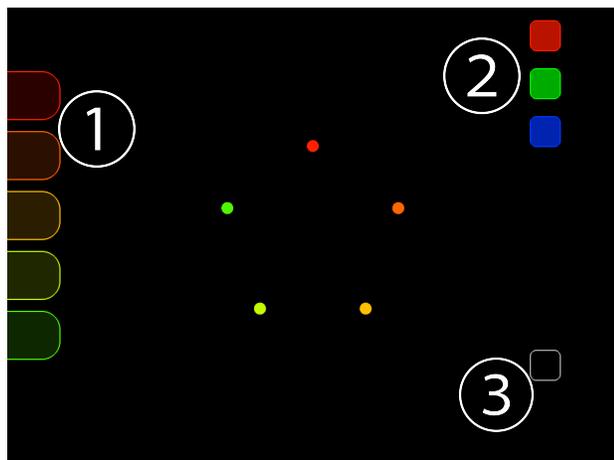


図5. 《F(music)》演奏画面

4.3.2. 数式決定画面

数式決定画面は図6に示しているように主に三つの部分に分かれている。①の部分は数式決定スライダ―

である。スライダーを変化させることで数式が決定する。②の部分は数学的アプローチボタンである。上から Y 軸対称移動、X 軸対称移動、絶対値表示を行うボタンであり、数学的アプローチを可能にしている。③の部分はステップシーケンサー入力部である。16ステップの ON,OFF の切り替えを行う。ステップはそれぞれグラフの X 座標に対応していて、ON になっているステップの X 座標を数学関数に代入し、そこから得られる Y 座標の値が出力される音の高さに相当する。

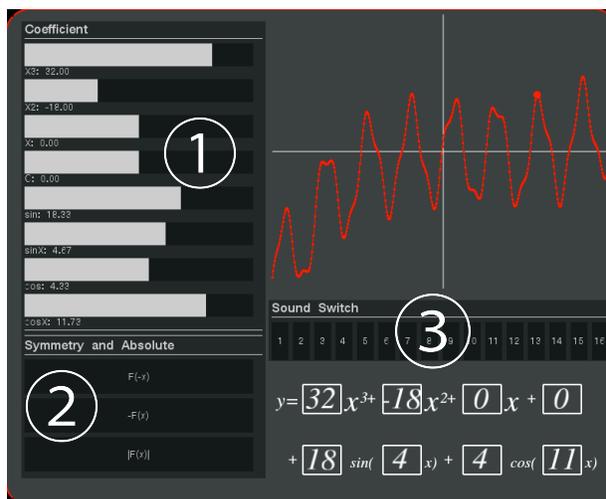


図6. 《F(music)》数式決定画面

4.4. 描画システム

《F(music)》の描画システムは、演奏画面中央にある五つの基準点とその基準点に対応したグラフの Y 座標を利用する。回転させながら各グラフの Y 座標の値が基準点からの距離として描画される。図7に例を示す。赤のグラフが右肩上がりなので(図7右)、回転させながら赤の基準点からの距離が次第に大きくなりながら点が打たれており、そこから線が回転している(図7左)。

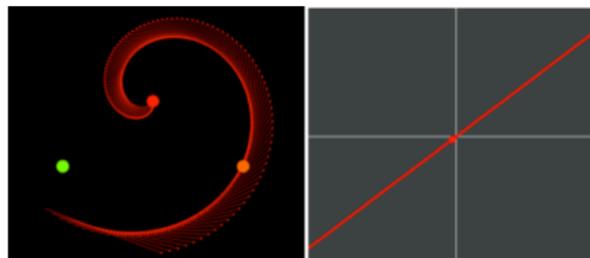


図7. 描画システム

⁶ 統合開発環境 <http://openframeworks.jp/>

⁷ ビジュアルプログラミング言語 <https://puredata.info/>

⁸ <https://github.com/danomatika/ofxPd>

4.5. 音響合成

《F(music)》は数式のグラフを楽譜上の音列と捉えることで音楽を演奏する(図8)。グラフのY軸が音の高さに対応しており、グラフの上部になるほど音が高くなる。グラフのY座標の情報を Pure Data に送り発音する。また、数式の値を用いて FM 変調をすることで音色を変化させることが可能である

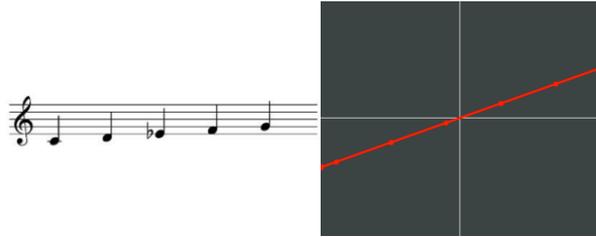


図 8. グラフと音列

4.6. モーフィング

《F(music)》におけるモーフィングは、グラフをさせたときに行う。グラフを変化させる前からグラフを変化させた後の状態へとモーフィングしていく。グラフの形が徐々に変化することになるので音列も徐々に変化することになる。図9にグラフの変化前と変化後、そのあとのモーフィングの様子を記す。

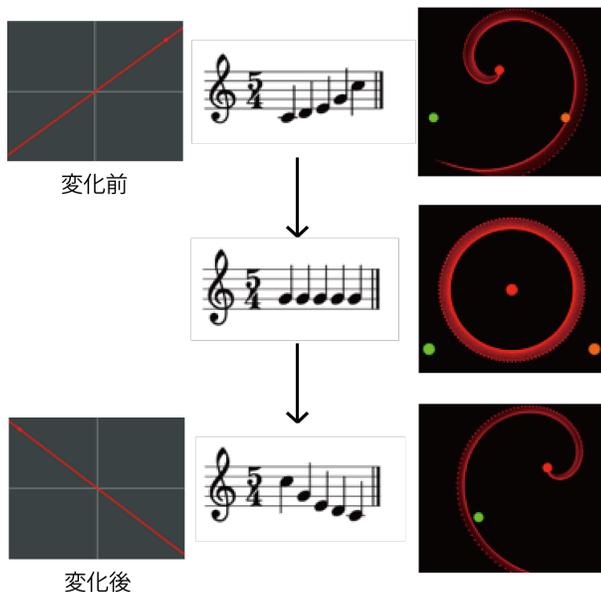


図 9. 《F(music)》のモーフィング

4.7. 《F(music)》についての考察

《F(music)》は、《Transition》ではユーザーが見ることができなかった数式を表示し、その数式から音楽を演奏することができた。また、対称移動や絶対値処理の数学的アプローチを音楽に取り入れることで楽譜上では表現できない演奏を可能にした。グラフィックのモーフィングも取り入れることができ演奏中において視覚的に楽しむことができるようになった。

《F(music)》の課題として挙げられることは三つある。はじめにモーフィングの方法が直線的のみになっていることである。次にモーフィングが途中で停止することがないので中間の音楽を演奏していると感じることが難しいことである。さいごは二つの音楽の間のモーフィングではなく、あくまで一つの音楽の中のモーフィングになっていることである。

5. 《METAMATH》

5.1. 制作指針

《MetaMath》(2015)は、《Transition》と《F(music)》を組み合わせることで制作した作品である。《Transition》と《F(music)》を制作して得た課題を解決するために三つの制作指針を設けた。まず《Transition》のように二つの音楽の間をモーフィングさせるようにすることである。次に音楽制作・モーフィングのどちらにも数式を利用することである。《Transition》ではモーフィング時、《F(music)》では音楽制作時のみ数式を利用していた。そこで《MetaMath》では音楽制作とモーフィングの両方において数式を導入することにした。さいごは、モーフィングの途中で停止させることを可能することである。《Transition》や《F(music)》では、モーフィングの動きを途中で停止させることはできない。そのため、モーフィングによる中間の音楽を演奏することなくモーフィングが終了することもあった。そこでモーフィングの停止・調整を可能にすることで、十分に二つの音楽の中間の音楽を演奏できると考えた。

5.2. システム概要

《MetaMath》は、《F(music)》と同様にグラフィックに openframeworks を音響生成には Pure Data を用いている。Pure Data は openframeworks に内包されている。ユーザーは五つの音源を使用して音楽を制作していく。《F(music)》と同様に数式によるグラフによって音列を決定していく。二つの音楽をそれぞれ A、B とすると A、B とともに五つのグラフがあるので計十つのグラフを操作することができる。モーフィングについての詳細は後述するが、《MetaMath》におけるモーフィングは、対応するグラフ上の点同士を数式つまりモーフィ

ング関数で結んでモーフィングさせていく。また、A、B それぞれの音楽が持っている五つのグラフからグラフィックを生成し、そのグラフィックも音楽と一緒にモーフィングさせていく。

5.3. ユーザーインターフェース

《MetaMath》は《Transition》および《F(music)》のユーザーインターフェースを継承している。ただし、《MetaMath》では、発音部にあたるグラフ上の点にはモーフィングのための数式が与えられる。どの点にどの数式を与えるかは、モーフィング関数決定画面に置いて決定される（図 10）。

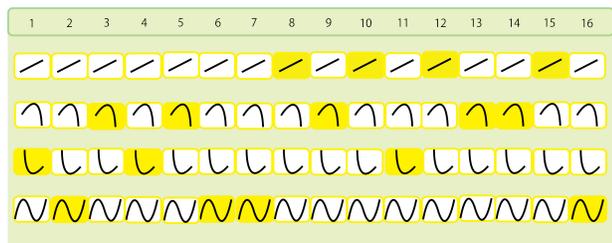


図 10. 《MetaMath》モーフィング決定 UI

5.4. 《MetaMath》におけるモーフィング

《MetaMath》でのモーフィングは、グラフ一つに十六個与えられている発音部の点をモーフィング関数決定画面で決定した数式によってモーフィングさせる。音楽を構成している音それぞれに数式が与えられているため、同じ数式でモーフィングさせることや音ごとに異なる数式でモーフィングさせることが可能となっている。図 11 にモーフィングの様子を示す。

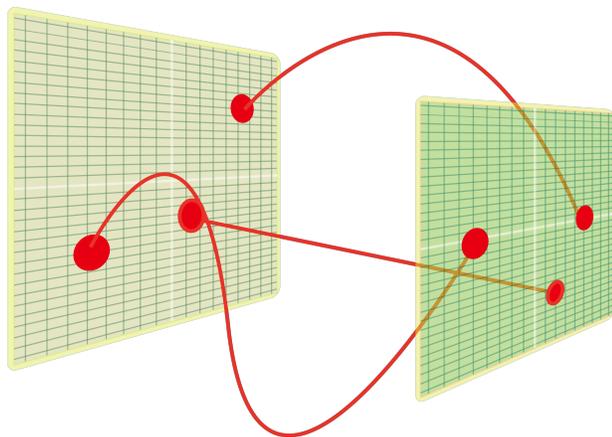


図 11. 《MetaMath》のモーフィング

5.5. 《MetaMath》についての考察

《MetaMath》は《F(music)》の課題であった二つの音楽の間をモーフィングさせることはできた。音楽制作だけでなく、モーフィングにも数式を導入するために、発音部それぞれに数式を与えた。そうすることで、より複雑な数式によるモーフィングを可能とした。そして《MetaMath》におけるモーフィングは途中で停止することを可能にでき、二つの音楽の中間の音楽を演奏しないでモーフィングを終わることはなくなった。その上モーフィングの割合を調整できるようになったことで、二つの音楽の間の音楽の調整が可能となり、モーフィングによる音楽制作を可能とした。

6. おわりに

ここでは、筆者の作品を踏まえた上で、数式による音楽の制作・モーフィングについて以下にまとめた。

《MetaMath》では、数式から得られたグラフを利用して二つの音楽を制作し、その二つの音楽の間を数式によってモーフィングさせることができた。そして、モーフィングを途中で停止・調節することができるようになった。

音楽制作に数式を用いることで音楽経験のない人でも音楽制作が可能になった。そして、絶対値変換による音列の変化など数学的アプローチによって楽譜上では表現することができない音楽を演奏することができた。

数式を用いて音楽をモーフィングさせることで、二つの音楽の中間の音楽を演奏することができた。そして、モーフィングを調整することで、中間の音楽を複数演奏することができた。

以上のことより数式による音楽のモーフィングは楽譜上では表現できない音楽を提供することができ、中間の音楽を複数提案することができるので、編曲への応用を考慮に入れていきたい。

7. 参考文献

- [1] Curtis Roads (青柳龍也 他), コンピュータ音楽 - 歴史・テクノロジー・アート -, 東京電機大学出版局, pp.675-680, 2001
- [2] 同上 pp.266-270
- [3] MORPH 2, <http://www.mi7.co.jp/products/zynaptiq/morph/>, 2015 年 11 月取得
- [4] 浜中雅俊, 平田圭二, 東条敏, タイムスパン木に基づくメロディモーフィング法, 情報処理学会 音楽情報科学研究会 研究報告 2008-MUS-74-19, Vol.2008, No.12, pp.107-112, 2008

- [5] 浜中雅俊, 2つのメロディの内挿を求めるメロディ
モーフィング手法, [http://shingi.jst.go.jp/
abst/p/08/jst/jst806/jst2.pdf](http://shingi.jst.go.jp/abst/p/08/jst/jst806/jst2.pdf), 2015年11月
取得

8. 著者プロフィール

藤原 尚紀 (Takanori FUJIWARA)

1988年山口県生まれ、九州大学芸術工学部音響設計
学科卒業、九州大学大学院芸術工学府の修士課程に在
学中。数式を用いた音楽演奏システムを制作・研究し
ている。

中村 滋延 (Shigenobu NAKAMURA)

1950年生まれ。作曲家、メディアアーティスト、音
楽評論家、映画研究者、交響曲5曲を含む100曲以上
のクラシック系現代音楽を作曲。また「音楽系メディ
アアート」という領域を創成し、視覚要素を含むコン
ピュータ音楽やサウンド重視の映像アートを多数制作。
現在、九州大学大学院教授（芸術工学府コミュニケー
ションデザイン科学コース、芸術工学部音響設計学科）。