

コンピュータプログラミングの体験学習を目的とした 高校数学による音楽生成の実習の試み

The music synthesis using high-school mathematics for the experiential learning of computer programming

金子 格¹
Itaru Kaneko

1. まえがき

楽器音や音楽を高校数学の関数のみで合成する体験学習を試みた。コンピュータ教育において論理的思考力やアルゴリズム的な教育に比べ、信号処理を含む教育は少ない。映像、音響、無線、気象、金融といった、大量の数値データを扱う信号処理の応用が拡大しつつあり、若い学習者に信号処理への関心を持ってもらう意義は大きい。ここで紹介する体験学習は、身近な音楽を題材とすることで信号処理に楽しく親しんでもらい、高校数学の意外な用途も発見してもらうことを目的としている。

2. 従来の研究

プログラミング教育の題材としては、古くはソートなどのデータ処理が多く、近年ではネットワークやロボットなどを題材とすることも多い¹⁾。しかし音や音楽の生成をプログラミングの題材とする例はあまりないようだ。

音や音楽を扱うプログラミング言語としては、CLM, C-Soundなどが有名である。CLMはCommon Lisp上で音声信号を扱うライブラリであり様々な音や音楽の合成が可能である。C-SoundはC言語に近い文法で音色、音楽の製作、演奏が可能である。しかしいずれもかなり高度な知識が必要であり、短時間の体験学習に利用するのは難しい。

3. 提案システム

提案する体験学習では、基礎的なプログラミングの知識と、高校3年程度の数学の知識のみを前提とするようにした。またこれらを完全に習得していなくても、数時間で簡単に習得でき、自分の音楽作品を完成できる体験学習を目指した。

処理系にはscilabを使用した。他にも数多くの優れた候補があるが、体験学習の後に学生が自分で無料ダウンロードして利用できることと、比較的通常の数式に近い方法で記述できることからscilabを選んだ。

ライブラリとサンプルプログラムを構築し、それらを使用して音楽の合成方法を提示し、受講者はサンプルプログラムを学びながら簡単に自分の作品を創作できる

音色や音楽の合成方法の説明に1~2時間程度行い、その後2時間程度で受講者自身の作品を制作してもらい、その後発表会を行った。制作された作品はweb上で公開した。

4. プログラム例

次に体験学習で使用したプログラムのいくつかを示す。

```
//
// プログラム 1: 周波数を 2^(n/12)ずつ変化させて
// 半音階を作る
stacksize(1e7);
fs=44100;
l=20;
w=0:8191;
t=w/fs;
// 12 音音階の生成
x=zeros(1:20*fs);
for i=0:14
s=i*fs/4;
n=i;
f=440*2^(n/12);
x(w+s+1)=x(w+s+1)+sin(2*pi*f*t)*.1;
end
sound(x,fs);
//
// プログラム 2: 切り捨て処理で、ハ長調の音階を作る
//
x=zeros(1:20*fs);
for i=0:14
s=i*fs/4;
n=int((i+2)/7*12+.4);
f=440*2^(n/12);
x(w+s+1)=x(w+s+1)+sin(2*pi*f*t)*.1;
end
sound(x,fs);
//
// プログラム 3: ループによって音楽の構造を作る
//
x=zeros(1:20*fs);
for i=0:64
s=int(i*fs/4);
k=i-int(i/4)*4;
m=k*2;
n=int(m/7*12+0.4);
f=440*2^(n/12);
x(w+s+1)=x(w+s+1)+sin(2*pi*f*t)*.2;
end
//メロディ 2
for i=0:64
s=int(i*fs/4);
k=i-int(i/8)*8;
```

¹ 東京工芸大学

```

m=int(k/2)*3+7;
n=int(m/7*12 + 0.4);
f=440*2^(n/12);
x(w+s+1)=x(w+s+1)+sin(2*pi*f*t)*.2;
end
sound(x,fs);
//
// プログラム 4:音色が異なるメロディを重ねる
//
x=zeros(1:20*fs);
// メロディ 1
i=0:96;
c=int(i/16)-int(i/32)*2;
d=c/7*12-int(c/7*12);
w=0:fs;
t=w/fs;
a=(1-exp(-t/0.01));
r=exp(-t/0.3);
for i=1:96
s=int(i*fs/6);
k=i-int(i/4)*4;
m=k;
n=int(m/7*12 + 0.4 + d(i+1));
f=440*2^(n/12);
m=sin(2*pi*f*t).*sin(20*t)*2;
x(w+s+1)=x(w+s+1)+a.*r.* sin(2*pi*f*t+m)*.2;
end
//メロディ 2
a=(1-exp(-t/0.001));
r=exp(-t/0.05);
for i=1:96
s=int(i*fs/6);
k=i-int(i/8)*8;
m=int(k/2)*3-2;
n=int(m/7*12 + 0.4 + d(i+1));
f=440*2^(n/12);
x(w+s+1)=x(w+s+1)+a.*r.* sin(2*pi*f*t);
end
//リズム 1
for i=0:96
k=int(i/8)*8;
j=i-k;
s=int((k+9-12/(9-j))*fs/6);
w=0:fs;
t=w/fs;
a=(1-exp(-t/0.001));
r=exp(-t/0.03);
n=rand(t)-0.5;
x(w+s+1)=x(w+s+1)+a.*r.*n;
end
//再生
sound(x,fs);

```

5. 受講者の感想

体験学習の内容の中で講演とデモンストレーション部分のみを大学1年生向けの特別講義として行った。その講義のレポートから感想をピックアップし、表1に示す。

表1 大学1年次生の感想

芸術学部の学生	工学部の学生
<p>やってみたくと思った。数学が初めて役に立つと思った。聞いているだけで楽しい授業だった。サンプルの音楽がよかった。数式だけでメロディが出るのは不思議だった。サイラボやってみたく。あんなに苦手な数学が音と関係するとは驚き。</p>	<p>数学の式で音を出すのに驚いた。まさか数式で音楽が演奏できるとは思わなかった。音色が楽器のようにきれいだ。数学の授業をまじめに受けようと思った。ここまで数式と音に関連があるとは思わなかった。</p>

6. 議論

よく聞かれる質問と筆者の考えを示す。

Q1. もっと高度なアルゴリズム(進化アルゴリズム等、たとえば⁴⁾)を使ったらどうか。

A1. 個人的には関心がある。体験学習用としては短時間で理解できるようにプログラムの単純さを保つ必要がある。高校生が理解できる範囲で手法を選んでいる。

Q2. 音楽を題材とすることの反応は?

A2. 概ね好評だ。「こんなことができるとは想像していなかった」という感想が多い。「数学の面白さ」を教える方法の一つになるとおもう。

Q3. 自由制作はうまくいくか?

A3. 始めて30分くらいは、とまどっている。「とにかく数値やプログラムを自由に修正してごらん」と促すと、ようやく試行錯誤を始め、だんだんとコツを飲み込んで、2時間程度で個性的な作品を完成する。音楽が得意な人の方が、かえってメロディをそのまま入力するという考えから抜けられず苦勞する。

7. まとめ

コンピュータ・プログラミングの学習を目的として、音楽合成を題材とした体験学習を行った。反応は良好で、プログラミングと数学に対する関心が高まった。作品作りも楽しんでもらった。

提案した体験学習は、高校で習った範囲の数学知識を利用し、音楽合成数学やプログラミングを行うことが実感でき、数学やプログラミングに対する関心を高めるのに効果的だと考えられる。

[参考文献]

- 1) 西ヶ谷浩史, 「変わりつつある情報教育: 3. 中学校における教育実践事例」, 情報処理, 48(11), 1191-1195 (2007-11-15)
- 2) CLM ホームページ, <https://ccrma.stanford.edu/software/clm/>
- 3) C sound ホームページ, <http://www.csounds.com/>
- 4) Eduardo Reck Miranda, "At the Crossroads of Evolutionary Computation and Music: Self-Programming Synthesizers, Swarm Orchestras and the Origins of Melody", ACM "Evolutionary Computation", Volume 12 Issue 2 June 2004 "