2007 年 8 月 7 日

グリッサンド音の作成について

新潟工科大学 情報電子工学科 竹野茂治

1 はじめに

音は、コンピュータ上ではデジタルデータとして保存されていて、そのルールはかなりシンプルなものなので、単純な音のデータなら容易に作成することができる。

ここでは、その音のデータ形式の概念を簡単に述べ、その上で「グリッサンド音」を作成するにはどうしたらいいかを、数学的に考察してみることにする。

2 音のデジタルデータ

音には、音階、音量、音色などの色々な要素がある。また、音は複数の音が重なっている場合でも、我々はそれを複数の音の重なりであると認識できる。

同じく波動現象である光にも、色の種類、色の明るさ、色の鮮やかさなどの要素があるが、色は複数の色が重なると、我々はそれを複数の色とは認識せずに、それを合成した一つの色としか認識できない。これは光と音の認識の違いの特徴的なところで、音の方はそれにより和音や合奏、そして複数の楽器編成による音楽というものが成立することになっている。

一方、音楽を記録する場合、大編成のオーケストラのスコア (総譜) のようにこの楽器はこの音、この楽器はこの音、のように記録する方式[1]では、楽器編成によりデータ量が変わり、また、同じ曲でも演奏者毎の微妙な演奏の違いを再現することは難しい。

よって、レコードや CD も含め通常の記録方式は、実際の音の波動の物理的な表現である空気中の波の振幅を時系列毎に保存する、という方法を取っている。これは、スコアを記録する方式に比べてデータ量ははるかに大きくなる。

コンピュータで使用されるデジタル音声データ (WAV, AU, AIFF などの形式) の場合もこの後者の方式で、物理的な振幅を電気信号の振幅に置き換え、それを時間と振幅

[1] MIDI はこのような記録方式を取っている。

に関してデジタル化 (標本化と量子化) をしたものであり、数千、あるいは数万分の 1 秒毎に、その音の波動の振幅を整数値 (8bit あるいは 16bit の整数) として順に保存している。

逆に言えば、$\sin x$ などの単純な振動関数を用いてこのような振動のデータを作り出せば、比較的容易にデジタル音声データを自作することができる。

例えば、

$$y = f_1(t) = A \sin 2\pi ht \quad (0 \leq t \leq T) \tag{1}$$

という関数 (A, h は正の定数) は、周期が $1/h$、すなわち周波数 (= 1 秒間の振動数) が h であるような周期関数なので、これは周波数が h Hz である音波振動に対応する。

これを、1/44100 秒毎に、-32767~32767 の範囲の整数値として出力すれば、いわゆる linear PCM データとなる。そのようなものを生成する C 言語で書けば、例えば以下のようになる。

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main(void)
{
    double T=3.0;    /* 3 秒のデータ */
    double h=440.0; /* 440Hz の音 */
    double vol=0.7; /* 70% のボリューム */
    double sample=44100.0; /* 標本化周波数 44100Hz */
    double t,y;
    short int z;     /* -32767~32767 の整数 */

    for(t=0.0; t<=T; t+=1.0/sample){
        y=vol*sin(2.0*M_PI*h*t); /* M_PI=円周率 (in math.h) */
        z=32767*y;
        putchar(z&0x00ff);       /* 下位 8 bit を出力 */
        putchar((z>>8)&0x00ff); /* 上位 8 bit を出力 */
    }
    return 0;
}
```

3 音階と周波数の関係

弦楽器を考えればわかると思うが、短かく、強く張った弦ほど高い音が出る。これは、そのような弦の方が速く振動し、その振動が空気を伝わるからである。つまり、高い音程は速い振動に、低い音程は遅い振動に対応する。

それに対し、音量は、2 節のプログラムにあるように振幅の大きさ、すなわち式 (1) の A に対応し、また音色は波の形に対応する。(1) のような三角関数による音は単純な、純粋な音で、実際の楽器の音はより複雑な振動波形になっている。

ギターなどの弦楽器を引く人はわかるだろうが、倍の周波数 (半分の弦の長さ) にすると丁度 1 オクターブ上の音程の音になる。よって、4 倍の周波数は 2 オクターブ上の音になる。1 オクターブの中には、半音間隔で、

ド、ド ♯ (= レ ♭)、レ、レ ♯ (= ミ ♭)、ミ、ファ、ファ♯ (= ソ ♭)、
ソ、ソ ♯ (= ラ ♭)、ラ、ラ ♯ (= シ ♭)、シ、ド

と、12 個の間隔がある (ミとファ、シとドは半音しか離れていない)。各音程の周波数は、1 オクターブ (12 半音上) 上がる毎に周波数が倍になるように、音階に対して等比数列的に増えていく[2]。例えば、ド、ド ♯、レ、. . . の音の周波数を $a_0, a_1, a_2, \ldots$ とし、この等比数列の公比を r とすると、

$$a_{12} = a_0 r^{12} = 2a_0$$

であるから、$r^{12} = 2$ より $r = \sqrt[12]{2}$, よって、

$$a_n = a_0 r^n = a_0 2^{n/12} \tag{2}$$

となる。

通常音階の基準音として使用されるラの音 (時報の最初の 3 つの音の音程) は、440Hz 付近の音が使われているようである。よって、

$$a_9 = 440 = a_0 2^{9/12} = a_0 2^{3/4}$$

[2] これは厳密には平均律音階の場合。純正律音階やピタゴラス音階では、等比数列とは微妙に異なる。

より、

$$a_0 = 440 \cdot 2^{-3/4} = 261.63\mathrm{Hz}$$

となる。これに $2^{1/12} = 1.05946\cdots$ を順にかけることで以下が得られる。

$$\begin{array}{lll}
a_0 = 261.63\mathrm{Hz}\ (\text{ド}), & a_1 = 277.18\mathrm{Hz}\ (\text{ド}\sharp), & a_2 = 293.66\mathrm{Hz}\ (\text{レ}), \\
a_3 = 311.13\mathrm{Hz}\ (\text{レ}\sharp), & a_4 = 329.63\mathrm{Hz}\ (\text{ミ}), & a_5 = 349.23\mathrm{Hz}\ (\text{ファ}\sharp), \\
a_6 = 369.99\mathrm{Hz}\ (\text{ファ}), & a_7 = 392.00\mathrm{Hz}\ (\text{ソ}), & a_8 = 415.30\mathrm{Hz}\ (\text{ソ}\sharp), \\
a_9 = 440.00\mathrm{Hz}\ (\text{ラ}), & a_{10} = 466.16\mathrm{Hz}\ (\text{ラ}\sharp), & a_{11} = 493.88\mathrm{Hz}\ (\text{シ}), \\
a_{12} = 523.25\mathrm{Hz}\ (\text{ド}) & &
\end{array}$$

このように、音階を一定に上げる (下げる) には、周波数を等比数列的に上げる (下げる) 必要があることに注意する。

4 グリッサンド音

グリッサンド (ポルタメントとも言う) とは、音程を連続的に変化させる奏法のことを言う。これは、ピアノ、クラリネット、トランペットのように中途半端な音を出せない (出しにくい) 楽器では難しいが[3]、トロンボーンやバイオリンなどでは可能な、特別な奏法である。

例えば、グリッサンドでなく、半音ずつ音階の音を t_0 秒毎に一定に上げて行くには、

$$\begin{aligned}
y &= f_2(t) \\
&= \begin{cases}
A\sin 2\pi a_0 t & (0 \le t < t_0), \\
A\sin 2\pi\{a_1(t-t_0) + t_0 a_0\} & (t_0 \le t < 2t_0) \\
A\sin 2\pi\{a_2(t-2t_0) + t_0 a_0 + t_0 a_1\} & (2t_0 \le t < 3t_0) \\
\dots & \\
A\sin 2\pi\{a_N(t-Nt_0) + t_0 a_0 + \cdots + t_0 a_{N-1}\} & (Nt_0 \le t < (N+1)t_0)
\end{cases}
\end{aligned} \tag{3}$$

のようにすればよい。なお、sin の中に t_0a_0, $t_0a_0 + t_0a_1, \ldots$ のように残してあるのは、t に関して連続関数にするためのものである。

この (3) の sin の中の 2π 以外の部分を $f_3(t)$ (すなわち $f_2(t) = A\sin 2\pi f_3(t)$) とすると、この $f_3(t)$ は、折れ線のグラフになる。

[3]ガーシュインの「ラプソディー・イン・ブルー」には、冒頭にクラリネットのポルタメントから始まる有名なフレーズがあるから、少なくともクラリネットでは不可能なわけではないらしい。

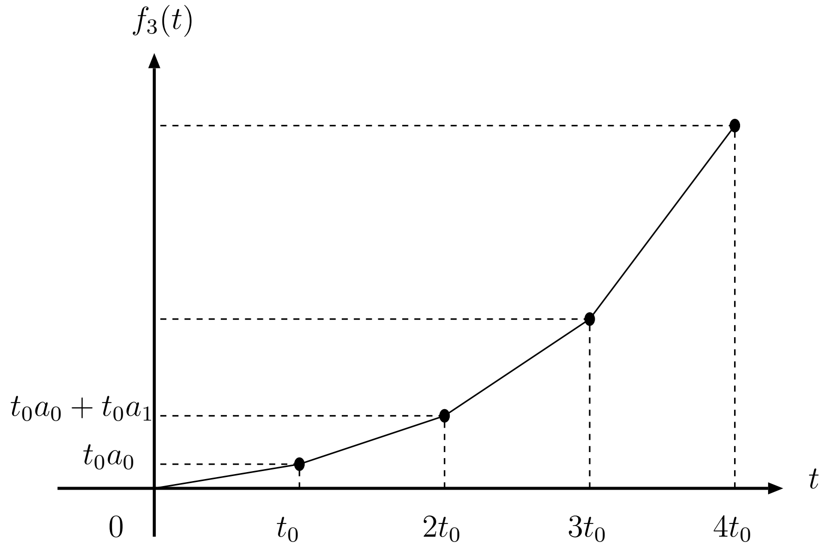

図 1: $f_3(t)$ のグラフ

しかし、この場合は t_0 秒毎に音程が一つずつ上がっていくだけで、連続的に音程が変化するグリッサンドにはなっていない。連続に音程を変化させるには、この折れ線を滑らかなグラフにすればよい。

この関数 $f_3(t)$ を t で微分すると、微分は関数の傾きを意味するから、そのグラフは図 2 のように階段関数となる。このグラフを見るとわかるが、この導関数 $f_3'(t)$ は $f_2(t)$ の音の周波数を表しているようである。まずこの事実について説明しておく。

今、増加関数 $g_1(t)$ $(g_1'(t) > 0)$ に対して $g_2(t)$ を

$$g_2(t) = A\sin 2\pi g_1(t) \tag{4}$$

とするとき、この $g_2(t)$ の $t = t_1$ での周波数を考えてみる。周波数とは 1 秒あたりの波の数であり、$t = t_1$ から $t = t_1 + \Delta t$ の間での波数は、

$$\frac{2\pi g_1(t_1 + \Delta t) - 2\pi g_1(t_1)}{2\pi} = g_1(t_1 + \Delta t) - g_1(t_1)$$

であり、よってこれを Δt で割った

$$\frac{g_1(t_1 + \Delta t) - g_1(t_1)}{\Delta t} \tag{5}$$

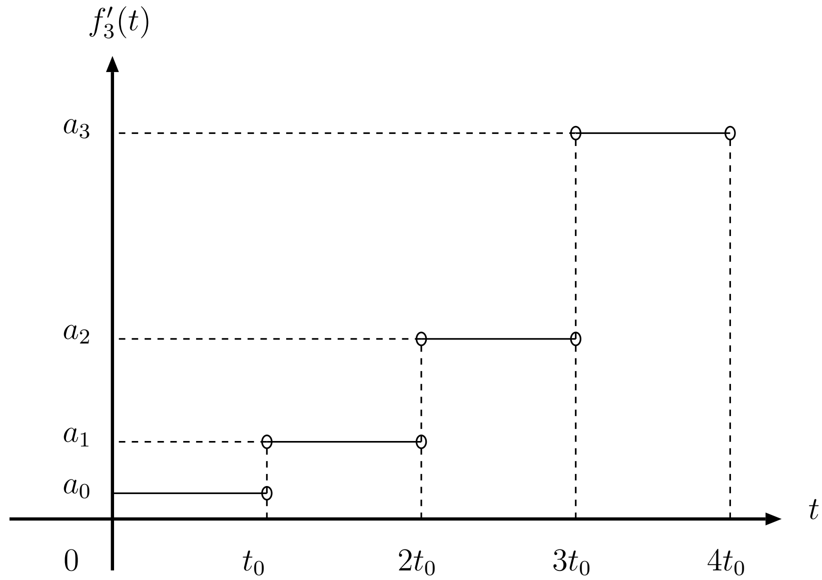

図 2: 導関数 $f_3'(t)$ のグラフ

が 1 秒あたりの波数、すなわち $g_2(t)$ のこの Δt 秒間の平均的な周波数となる。この (5) の式を $\Delta t \to 0$ としたものが丁度 $t = t_1$ での周波数となるが、微分の定義より、その極限は

$$\lim_{\Delta \to 0} \frac{g_1(t_1 + \Delta t) - g_1(t_1)}{\Delta t} = g_1'(t_1)$$

となる。つまり、$g_1(t)$ の微分係数 $g_1'(t)$ は確かに $g_2(t)$ の周波数を表すことがわかる。

今、音程の周波数 a_n に対して、音程と同様に一定に増加する添え字 n を音程を数値化したもの、すなわち下のドを 0、ド♯を 1 のように見ることにすれば、

$$\log a_n = \log a_0 r^n = \log a_0 + n \log r$$

なので、

$$n = \frac{\log a_n - \log a_0}{\log r}$$

となり、この数値化された音程 x と周波数 a の間の関係は、

$$x = s(a) = \frac{\log a - \log a_0}{\log r}, \quad a = h(x) = a_0 r^x \tag{6}$$

と書けることになる。s は周波数を音程に変える関数、h は逆に音程を周波数に変える関数となる。

最初 $(t=0)$ の周波数が a で, $t=T$ のときの周波数が b であり、その間一定に音程が変化するような (4) の形の関数を考えてみることにする。

上の考察により $g_2(t)$ の周波数は $g_1'(t)$ であるので、g_1 が満たすべき条件は、

$$g_1'(0)=a, \quad g_1'(T)=b \tag{7}$$

および音程 $s(g_1'(t))$ が一定に変化、すなわち直線的に変化することである。

$s(a)=p$, $s(b)=q$ とすると、$s(g_1'(t))$ は $t=0$ から $t=T$ にかけて p から q に直線的に変化することになるので、

$$s(g_1'(t))=p+\frac{t}{T}(q-p)$$

と書けることになる。よって、

$$\begin{aligned}\frac{\log g_1'(t)-\log a_0}{\log r}&=p+\frac{t}{T}(q-p),\\ \log g_1'(t)=\log a_0+&\left\{p+\frac{t}{T}(q-p)\right\}\log r\end{aligned}$$

より、$g_1'(t)$ は、

$$g_1'(t)=a_0r^{p+(q-p)t/T} \tag{8}$$

となる。ここで、

$$a=h(p)=a_0r^p, \quad b=h(q)=a_0r^q$$

より、

$$\begin{aligned}g_1'(t) &= a_0r^{(1-t/T)p}r^{tq/T}=a_0(r^p)^{1-t/T}(r^q)^{t/T}=a_0\left(\frac{a}{a_0}\right)^{1-t/T}\left(\frac{b}{a_0}\right)^{t/T}\\ &= a^{1-t/T}b^{t/T}\end{aligned}$$

となるので、結局

$$g_1'(t) = a\left(\frac{b}{a}\right)^{t/T} \tag{9}$$

と書けることになる。これを積分すれば、

$$g_1(t) = \int a\left(\frac{b}{a}\right)^{t/T} dt = aT\frac{1}{\log\frac{b}{a}}\left(\frac{b}{a}\right)^{t/T} + C$$

となり、$g_1(t)$ では定数差は特に意味を持たないので、$C = 0$ とし、また b/a を R と書くことにすれば、

$$g_1(t) = \frac{aT}{\log R}R^{t/T} \tag{10}$$

と書けることになる。よって、

$$g_2(t) = A\sin 2\pi\frac{aT}{\log R}R^{t/T} \tag{11}$$

が、グリッサンドを生成する関数だということになるが、もちろん、sin の代わりに周期が 1 の別の関数 $\alpha(x)$ を使って、

$$g_2(t) = \alpha\left(\frac{aT}{\log R}R^{t/T}\right) \tag{12}$$

としてもよい。これは、(11) とは別の音色のグリッサンド関数となる。

5 最後に

このグリッサンド生成関数 (12) は、実際にこれを実装したプログラム sndglis を、yomi という名前のフリーソフトに付属して公開している ([1])。

また、この考え方を発展させると、音を連続的に上下に変動させることも可能で、それにより、例えば任意の数学的なグラフの見た目を音程を使って表現することもできる。

例えば、$y = f(x)$ $(x_1 \le x \le x_2,\ y_1 \le y \le y_2)$ のグラフを、y_1 が音程 p、y_2 が音程 q に対応するように、T 秒間の音で表すとすると、適当に平行移動や縮尺を行って、

$$s(g_1'(t)) = \left\{ f\left((x_2 - x_1)\frac{t}{T} + x_1 \right) - y_1 \right\} \frac{q-p}{y_2 - y_1} + p$$

とすればよいから、この右辺を $g_3(t)$ と書くことにすれば、

$$g_1'(t) = a_0 r^{g_3(t)}$$

となる。しかし、一般の $f(x)$ に対しては、この右辺は簡単には積分できないので、一般の $f(x)$ を厳密に音程で表現することは難しい。

ただ、コンピュータで関数のグラフを書くときは、通常は曲線として書くのではなく、数百分の 1 位に分割して、目ではわからないくらいの細かい折れ線で近似して書いているので、そのような分点のデータを使えば、個々の折れ線を (12) によるグリッサンドで表すことで、近似的な表現は可能である。

実際にそのようなことを行うプログラムも、yomi の中に tbl2snd という名前で含まれている。これは、gnuplot という有名なグラフプロットソフトの作成する table データという折れ線の分点データを利用し、そこから上の方法でグリッサンド音データを作成するプログラムである。サンプルは、[3] にあるので、詳しくはそちらを参照されたい。

グリッサンドは、音程を単純に一定に上げていくだけであるが、そのような波の生成は簡単なようで、実は微分や積分が関係する、意外に難しい話であることがわかったであろうか。逆に、こういったところにも微積分の考え方が必要になるということを知ってもらえれば、本稿はそれなりに意味があるのではないかと思う。

参考文献

[1] 日本語読み上げソフト yomi の WWW ページ
`http://takeno.iee.niit.ac.jp/~shige/unix/yomi/yomi.html`

[2] グリッサンド音生成ツール sndglis のサンプル
`http://takeno.iee.niit.ac.jp/~shige/unix/yomi/yomi.html`
`#sndglis-sample`

[3] gnuplot グラフの音声化ツール tbl2snd のサンプル
`http://takeno.iee.niit.ac.jp/~shige/unix/yomi/yomi.html#tbl2snd-sample`