

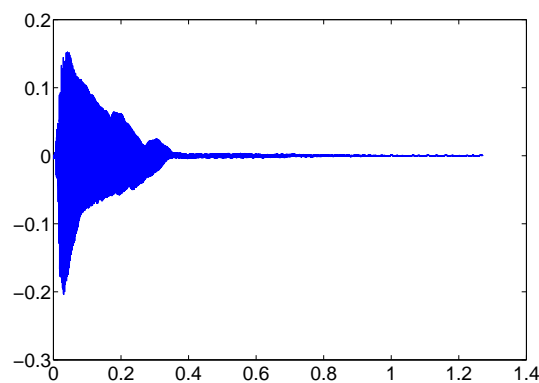
1 はじめに

簡単なデジタルシンセサイザーの合成方法に Wavetable 合成方式 (以下、Wavetable 法とよぶ) がある。任意の 1 波長の音のデータをテーブルに格納し、そのデータを組み合わせることで、音色の時間変化などを表現する方法である。

本資料では、その最も単純なプログラム例として、ADSR エンベロープの各区間の音色をテーブルに格納しておいて音を合成する方法を説明する。

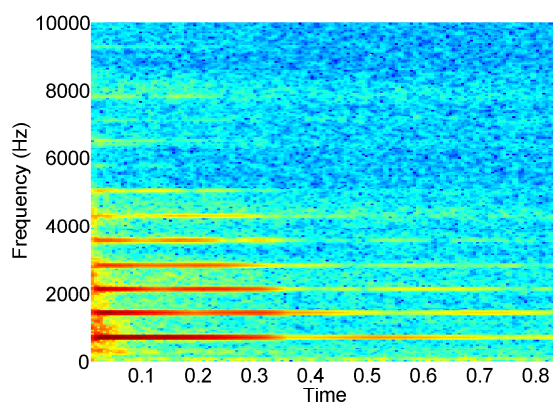
2 ADSR エンベロープ

この図は、MIDI 音源のピアノの音を弾いた時間波形である。



この図でも観察できるが、楽器の音は、最初に急激に無音から大きな音量に変化し (Attack)、大きな音になったら減衰し (Decay)、その後、キーを押す間や息が続く間、持続する場合があります (Sustain)、最後に余韻 (Release) がある、という構造として大雑把に近似できる。

この音のスペクトログラムは次の通りである。



このように、時間によって倍音構造が変化していることがわかる。

この振幅の変化を 4 つの直線で近似するのが ADSR エンベロープである。

上記の音色の ADSR エンベロープを考えるためには、振幅の変化を見る必要がある。振幅の変化を見るためには、変位の絶対値を観察するのがよい。

練習 1 MATLAB で波形の変位の絶対値をプロットしてみよ。ただし、波形の平均値が 0 でない場合は、まず、波形データの各点からその平均値を引いておいた方がよい (このような処理を直流成分の除去という場合がある)。

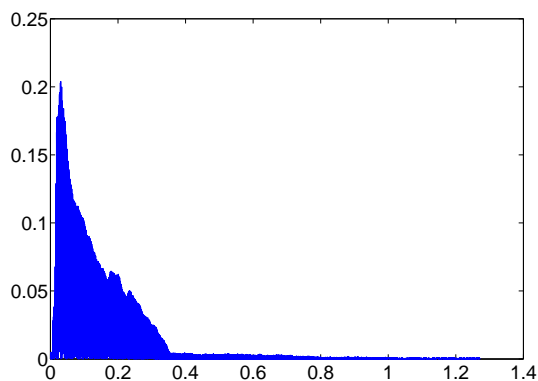
次のリストは系列からその平均値を引く例である。

```
>> a = [1 2 3 4];
>> a = a - -----(a);
>> a

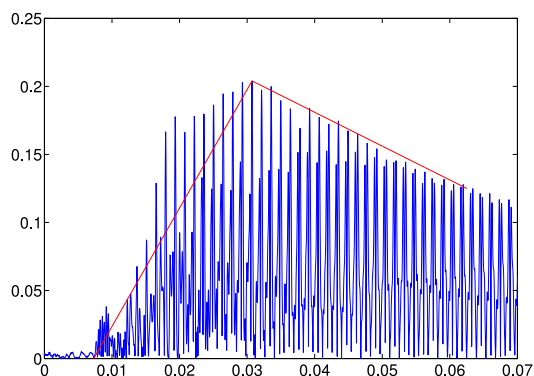
a =
-1.5000 -0.5000 0.5000 1.5000
```

下線部には平均値を求める関数名が入る。

上記のピアノの波形の変位の絶対値は次のようになる。



立ち上がりの部分を拡大してみる。

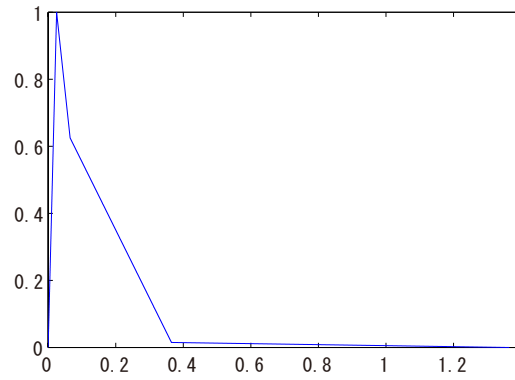


ADSR と 4 つの直線で近似するのであれば、例えば、A の区間は、0.01 秒より少し前に開始し、0.03 秒の少し後まで、D の区間は、A の終了点から、0.06 秒の少し後まで継続する、というふうに見なせる。

このようにして、ADSR を近似すると、例えば、以下のようなになるだろう。(A の開始時刻を 0 とする)

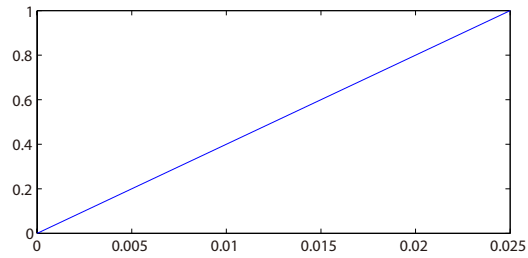
区間名	開始時刻	終了時刻	終了時刻の変位の大きさ
A	0	0.025	0.2
D	0.025	0.065	0.125
S	0.065	0.365	0.003
R	0.365	1.365	0

この ADSR の形をプロットすると、次のようになる。(ただし、最大値が 1.0 になるように正規化している。)

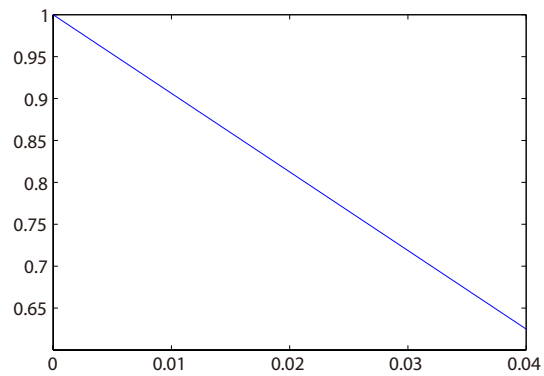


練習 2 上記の ADSR の概形を MATLAB でプロットしてみよ。例えば、最初の A の部分は次のようにプロットできる。

```
>> t1 = 0:1/fs:-----;
>> y1 = 1/-----*t1;
>> plot(t1,-----)
```

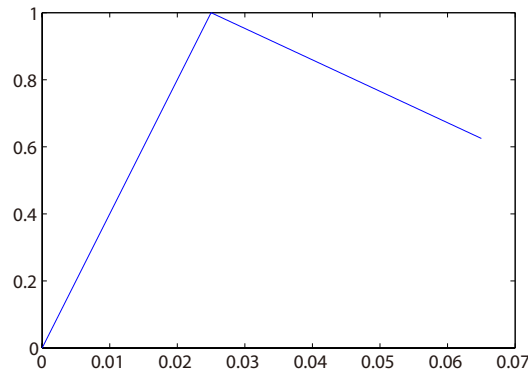


また、0.025 秒からの D の部分は、次のようにプロットできる。(とりあえず、開始時刻を 0 とする)



この D の直線のデータが y2 という変数に格納されているとすると、次のようにすると、AD の部分をプロットできる。

```
>> plot([t1 -----+t2],[y1 y2])
```



3 楽器音からの波形データの抽出

とりあえず、楽器音の適当な部分の 1 周期分を抽出してみる。

抽出するためには、デジタルデータの 1 周期分の長さを推定しなければならない。この際、デジタルデータは、データはサンプリング周期ごとのとびとびのデータ (離散データ) であることに注意しなければならない。また、周期の単位はいろいろ考えられるが、プログラミングのためには、データの点数 (データが何個あるか) がわからないといけない。

録音した楽器音の 1 周期の長さを推定する方法はいくつかある。

1. 楽器音の音階がわかっている場合は、その音階に対応する周波数の値を用いて推定する。

練習 3 サンプリング周波数が 48KHz のとき、440Hz のラの音の周期は何点になるか?

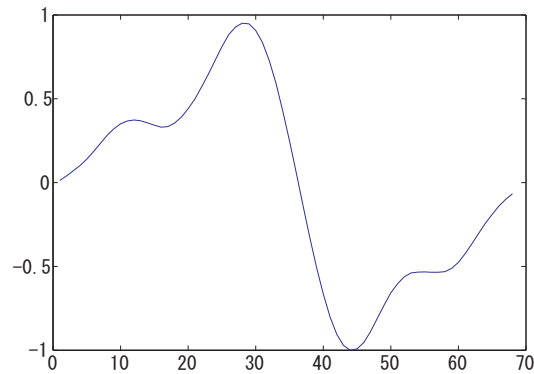
2. 波形を拡大して目視で観察して求める。

練習 4 自分が録音した音の時間波形のプロットを拡大して 1 周期が何点かを調べてみよ。

3. 自己相関などの方法を用いて推定する。

上記のピアノ音源 (サンプリング周波数 48KHz) の 1 周期は 68 だった。S の部分から適当に 1 周期分抜き出して、その区間での最も大きな変位の部分の絶対値が 1 になるように正規化してプロットしたのが次の図である。(楽器音データは data という変数に列ベクトルとして格納されているとする。)

```
>> r=data(82001:-----);  
>> r=r/----(-----(r));  
>> plot(r)
```



この波形を ADSR の長さ分だけ繰り返し、そのデータを ADSR エンベロープで変形させることで、元の楽器音をシミュレートしてみる。(adsr というデータに ADSR エンベロープが格納されているとする)

```
>> tau=68; % 1周期の長さ  
>> len=length(adsr); % adsr エンベロープの長さ  
>> iter=-----; % adsr エンベロープの長さが何周期分かを計算(関数を用いた式が入る)  
>> wave= repmat(r,iter,1); % 列ベクトルを繰り返して波形データを生成  
>> wave = wave(1:len).*adsr'; % 波形データを ADSR エンベロープで変形  
>> plot(wave);  
>> soundsc(wave,fs)
```

元データと同じ音階の音が確認できるはずである。