

1 音を歪ませる

MATLAB で、ファズやオーバードライブといった、音を歪ませるようなエフェクターと同等の効果を実感するのはそれほど難しいことではない。MATLAB で音の出力に利用する `sound` 関数は、`help sound` とすればわかるが、`sound` 関数は -1 から 1 の範囲の音しか出力できない。

この範囲を越えるとどうなるかを確かめてみる。

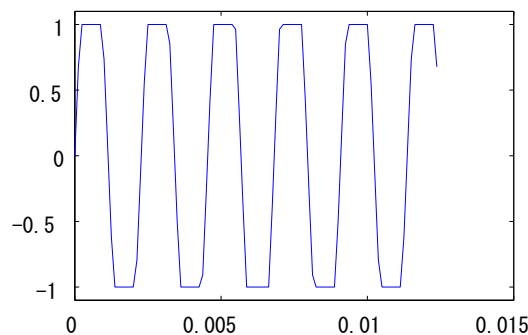
```
>> fs=8000;
>> t=0:1/fs:1;
>> y=sin(2*pi*440*t);
>> sound(y,fs)
>> sound(2*y,fs)
>> sound(4*y,fs)
>> sound(8*y,fs)
```

正弦波なので、歪んだという感じはしないかもしれないが、音色が変化したことはわかるだろう。
 $2*y$ という式は、

```
>> plot(t(1:100),2*y(1:100))
```

の結果からも明かなように、 y の値を 2 倍増幅する。

`sound` 関数では、 -1 以下の値は -1 として、 1 以上の値は、 1 として出力するので、実際には、次のプロットのような波形の音を聞いていることになる。



練習 1 様々な音を様々な倍率で増幅して `sound` 関数でどのように歪むか確認せよ。

上記のプロットのように、ある大きさ以上の変位の部分がつぶれてしまうような現象はクリッピングと呼ばれる。このような処理をプログラミングするには、次のような処理をしなければならない。

$$y(i) = \begin{cases} 1 & y(i) > 1 \\ y(i) & -1 \leq y(i) \leq 1 \\ -1 & y(i) < -1 \end{cases}$$

このような処理は、`for` を使って簡単に書ける。しかし、MATLAB では、このような場合には `for` を使わない。その代わりに、条件を満たす配列 (行列) の要素をまとめて処理する機能がある。

```
>> x=[2 5 3 1 4]
```

```
x =
     2     5     3     1     4
```

```
>> x>1
```

```
ans =  
     1     1     1     0     1
```

```
>> x(x>1)
```

```
ans =  
     2     5     3     4
```

```
>> x(x>1)=2
```

```
x =  
     2     2     2     1     2
```

このように、条件を満たした要素だけを返却値としたり、その要素だけに代入したりすることができる。この機能を利用すると、ファズを簡単に実現できる。

```
>> [y,fs,nbits]=wavread('guitar1.wav');  
>> yd=y*32;  
>> yd(____)=1;  
>> yd(____)=-1;  
>> soundsc(yd,fs)
```

練習 2 上記の下線部を埋めてプログラムを完成させよ。

練習 3 上記の 32 をいろいろな数字に変えて音がどうなるか確かめよ。また、様々な音を加工してみよ。加工した音をプロットして波形も確認せよ。