

## 7 音の振幅の操作

音の大きさは、波の振幅の大きさで決まる。

```
>> t = 0:1/8000:1;
>> xx=sin(2*pi*400*t);
```

次のプログラムは `xx` の全体の振幅を半分にする。

```
>> xx2 = 1/2 * xx;
>> sound(xx2,8000)
```

ボリュームを変化させなければ、音の大きさが `sound(xx,8000)` より小さくなるのがわかる。

## 8 音の重ね合わせ

正弦波を足し合わせることで、様々な音を作ることができる。次のプログラムは、440Hz の正弦波 `x440` と 660Hz の正弦波 `x660` を足しあわせた波 `xx` を作るプログラムである。

```
>> t = 0:1/8000:1;
>> x440=sin(2*pi*440*t);
>> x660=sin(2*pi*660*t);
>> xx = x440 + x660;
>> plot(xx(1:100));
>> sound(xx,8000);
```

プロットを見ればわかるように、最も振幅の大きい部分では、振幅が約 2 となっている。sound 関数は help を見ればわかるように、ベクタの値を  $-1$  以上  $1$  以下と仮定しているため、 $-1$  以下、または  $1$  以上の部分は正しく出力されない。

このため、sound 関数で正しく出力するためには、振幅を  $-1$  以上  $1$  以下になるように正規化する必要がある。matlab でベクタの最大値を求める関数は `max` である。

```
>> max(xx)
```

絶対値を求める関数は `abs` である。

練習 7 以下の [1] から [3] を埋めて `xx` を `sound` 関数で正しく出力するプログラムを完成させよ。

```
>> xmax = [1] ( [2] (xx));
>> xnorm = xx [3] xmax;
>> sound(xnorm, 8000);
```

練習 8 `xnorm` を一部分だけプロットして、振幅の最大値が  $1$  になっていることを確認せよ。

このような正規化を自動的におこなう関数として `soundsc` 関数が用意されている。

```
>> soundsc(xx, 8000);
```

## 9 正弦波による様々な音の生成

周波数の近い音を二つ重ねあわせてみる。

```
>> x430 = sin(2*pi*430*t);
>> x450 = sin(2*pi*450*t);
>> xb = x430 + x450;
>> soundsc(xb,8000);
>> plot(xb(1:500));
```

周波数がある程度近いと、このように音の強弱がある一つの音として聞こえる。この現象をうなりという。

練習 9 うなりの音の強弱はどのような周期で起きるか調べよ。

練習 10 二つの音をいろいろと変化させて、どのような音になるか試してみよ。

正弦波の倍音をいくつか組み合わせると、いろいろな音色を作ることができる。

$$A = \sum_{k=1}^n A_k \sin 2\pi k f \quad (1)$$

ただし、

$$A_k = \begin{cases} \frac{4}{\pi k} & k = 1, 3, 5, \dots \\ 0 & k = 2, 4, 6, \dots \end{cases}$$

次のプログラムはこの波の  $n = 3$  の場合を生成する。

```
>> t = 0:1/8000:1;
>> sq3 = 4/pi * sin(2 * pi * 440 * t) + 4/pi/3 * sin(2 * pi * 440 * 3 * t)
>> sound(sq3, 8000);
```

練習 11 sq3 の適当な部分をプロットして、どのような形の波形か確認せよ。

練習 12  $n = 5, 9$  の波を作成し、波形と音色を確認せよ。

次のような波もある。

$$A = \sum_{k=1}^n A_k \cos 2\pi k f \quad (2)$$

ただし、

$$A_k = \begin{cases} -\frac{8}{\pi^2 k^2} & k = 1, 3, 5, \dots \\ 0 & k = 2, 4, 6, \dots \end{cases}$$

練習 13 この波を  $n = 3, 5, 9$  の場合で生成し、波形と音色を確認せよ。

matlab には、式 (1) であらわされる波を生成する関数 square や式 (2) であらわされる波を生成する関数 sawtooth が用意されている。

```
>> t = 0:1/8000:1;
>> sq = square(2 * pi * 440 * t);
>> plot(sq(1:100));
>> sound(sq, 8000);
>> tr = sawtooth(2 * pi * 440 * t, 0.5);
>> plot(tr(1:100));
>> sound(tr, 8000);
```

square や sawtooth は引数を変化させることで波形を変化させることができる。

練習 14 help で square や sawtooth の使い方を調べて、様々な音を作って音の違いを確かめてみよ。