1.6 音の生成

最も基本的な音である正弦波 (サイン波) を作成する。440Hz の正弦波は、時間を t とすると $\sin(2\pi ft)$ (ただし、$f = 440$) とあらわせる。MATLAB では、多くの関数が、行列を引数として、単一の数字と同じ感覚で計算できる。その機能を用いると、周波数が 440Hz の正弦波をサンプリングレートが 16000 で 1 秒間生成するスクリプトは次のように書ける。(`pi` は 初期値として π が与えられた変数である。)

```
>> Fs=16000;
>> t=0:1/Fs:1;
>> y=sin(2*pi*440*t);
>> sound(y, Fs);
>> plot(t,y)
```

このように、数式とほとんど同じ感覚でプログラミングできる。

演習 1-16 様々な周波数の正弦波を生成して再生してみよ。

この波形の一部だけを表示するためには、表示したい範囲のインデックスを指定すればよい。例えば、1001 点目から 1000 点分表示するには、次のようにすればよい。

```
>> t1=1001;
>> t2=t1+1000;
>> plot(y(t1:t2))
```

演習 1-17 上の例では、x 軸の単位が点数となってしまう。x 軸の単位が秒となるようにプロットせよ。

演習 1-18 適当な波形を 4 周期分プロットせよ。

1.7 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。次の例は、440Hz の正弦波 `y440` と 660 Hz の正弦波 `y660` を足し合わせて `y` を作る。

```
>> Fs=16000;
>> t=0:1/Fs:1;
>> y440=sin(2*pi*440*t);
>> y660=sin(2*pi*660*t);
>> y=y440+y660;
>> plot(y(1:200));
```

プロットを見ればわかるように、`y` は最も変位の大きいところでは、2 に近い値を取る。**1.5** に記述したように、`sound` 関数は 1 以上の値を再生できない。したがって、`soundsc` 関数を使うか、最大の変位を 1 以下になるようにしなければならないことに留意すること。

MATLAB では、正弦波以外にも、波を生成する関数が用意されている。

演習 1-19 MATLAB の関数を用いていろいろな波を発生し、それらをプロットして、どのような特徴があるか調べよ。また、それらを再生して、どのような音なのかを確認せよ。(ヒント: 「`lookfor 発生`」 と入力すればいろいろな関数が見つかる。)

生成した信号をはじめとする MATLAB 内のデータはファイルとして書き出すことができる。

演習 1-20 `wavwrite` 関数の使い方を調べて自分で作成、加工したデータをファイルとして保存せよ。

2 音の成分の分析

2.1 フーリエ変換

正弦波を重ね合わせるといろいろな音色の音を作ることができた。逆に、分解することもできる。それを可能にする技術にフーリエ変換がある。MATLAB で扱うディジタル信号に対しては、離散フーリエ変換という技術がある。`fft` という関数で計算することができる。さっそく、`fft` を試してみる。(% はコメントを表わすので、それ以降は入力する必要はない)

```
>> t=0:1/100:6-1/100;                   % 0 から 1/100 ずつ 6 の
                                        % 1/100 手前までベクトルを作る
>> y=sin(2*pi*15*t)+sin(2*pi*40*t);     % 15Hz と 40Hz の成分を持つ
>> f=fft(y);                            % y の長さ点で(600) fft を y にかける
>> f(10)                                % f(10) の値を見る
ans =
   7.7991e-013 +1.5264e-013i            % 複素数になっている
>> s=abs(f);                            % 「大きさ」をとる
>> plot(s)
```

実数の信号をフーリエ変換すると複素数の結果が得られる。その複素数の絶対値をとったものを振幅スペクトルという。この場合、プロットすると、真ん中の 300 点のところを中心に線対称の関係で、2 本ずつの線が観測される。この線の位置が信号の成分をあらわす。

ところで、MATLAB では、複素数は次のように入力でき、表示される。

```
>> z=5+3i

z =
    5.0000 + 3.0000i
```

演習 2-1 複素数の絶対値はどのように計算するか? (忘れた場合は、微積分の教科書などを見返して思い出すこと)

振幅スペクトルのどの位置に線があるかを調べるには次のようにするのが便利である。プロットした結果を見ると、線の場所以外はほとんど 0 であることがわかる。`for` ループなどで `s` の要素を一つ一つ検査することもできるが、MATLAB では次のように一度に求めることができる。

```
>> find(s>0.01)
ans =
     91    241    361    511
```

MATLAB では、行列に対して条件式を書くと条件に対する論理値を返却した行列が得られる。

```
>> a=[1 2 3 4 5];
>> a==1

ans =
      1      0      0      0      0

>> a>3

ans =
      0      0      0      1      1
```

`a==1` のときには、その条件を満たす最初の要素 `1` のところだけが `1` となる行列が得られているし、`a>3` のときには、最後の二つの要素 `4, 5` のところが `1` となる行列が得られている。

find という関数は、help によると、「I = FIND(X) は、配列 X の非零の要素に対応する線形のインデックスを返します。」と書かれている。つまり、次のようになる。

```
>> find(a==1)

ans =
     1

>> find(a>3)

ans =
     4     5
```

したがって、find(s>0.01) の結果に戻ると、91，241，361，511 のところにピークがあることがわかる。

振幅スペクトルの x 軸は周波数に対応している。この例では、サンプリング周波数 100 Hz に対し、600 点のフーリエ変換をおこなっている。600 点で 100Hz をカバーするので、1 点では、$100/600 = 1/6$ Hz をあらわす。MATLAB のインデックスは 1 から開始されるので先程のピークは、$(91-1)/6 = 15$ Hz と、$(241-1)/6 = 40$Hz の位置にあることがわかる。

このように、fft をおこなって得られるスペクトルから、ある信号が、どのような成分から構成されているかがわかるのである。

演習 2-2 上記の s の s(1) から s(301) までをプロットせよ。ただし、x 軸に対応する系列 x を工夫して x 軸の単位が Hz になるようにせよ。

fft はいろいろな長さでかけられる。上記の例に対しより短い範囲でかけてみる。

```
>> plot(abs(fft(y,599)))
>> plot(abs(fft(y,598)))
```

15Hz、40Hz に対応した高い線のふもとの部分が太くなることがわかる。

離散フーリエ変換では、範囲を決めて変換している。実は、その範囲の外側は、範囲の中と同じ変動が繰り返されていると仮定している。つまり、fft(y) (fft(y,600)) のときは、y(600) の次の y(601) の値は y(1) と同じになると仮定している。fft(y,599) の場合は、本来は y(599) の後には y(600) となるはずなのに、y(1) と仮定している。つまり、この部分で実際の周期的変動から外れた変化が起きてしまう。

信号 (に含まれる成分) の周期がわかっていれば、その周期にあわせた幅で fft をかければよい。しかし、どのような成分を含むかを調べるために fft を用いるときには、対象の信号にはどのような成分が含まれるかわからないわけなので、ちょうどよい幅で fft をかけることは不可能である。

このような問題を解決するために、一般には窓関数を利用する。窓関数の例としてハン窓 (ハニング窓) を見てみる。

```
>> w=hann(600);
>> plot(w)
```

このように中央部分は 1 で裾に向かってゆるやかに 0 になるように変化する。

y に上記の窓をかけるとは次のような演算を行うことである。y(1) に w(1) をかける、y(2) に w(2) をかける、と順々に計算し、y(600) と w(600) まで計算する。

MATLAB では、このような行列の要素ごとの掛け算は .* という演算子で計算できる。

```
>> c=[1 5 4];
>> d=[2 3 1];
>> c.*d

ans =
      2     15      4
```

(注: 演算子 * は行列の掛け算である)

演習 2-3 上記の y に対して同じ長さのハン窓をかけてその結果をプロットして観察せよ。ただし、上記の y は行ベクトルであるのに対し、hann 関数が返す窓は列ベクトルであることに留意せよ。

このような窓関数を利用すると、fft の長さ (fft 長) が周期とずれていても、その影響が小さくなることが確認できる。

演習 2-4 上記の y に対して、例えば、y(1:599) など周期のあわないベクトルを作り、それに対して同じ長さのハン窓をかけて fft をかけてみよ。ハン窓をかけたときとかけないときを比較して、どのような変化が生じるか確認せよ。

演習 2-5 音声 2 種類と、音声でない音 2 種類を録音して、様々な条件で fft を使って分析せよ。その結果を用いて、音声と音声でない音の違いを考察せよ。