

ポータブルオーディオプレイヤーのイコライザーや、ミキサーのイコライザー、アンプなどのトーンコントロールなどを実現する技術をデジタル信号処理では、フィルタという。MATLAB の関数を利用すると、音声データに簡単にフィルタを適用することができる。代表的なフィルタをかける方法 (フィルタリング) を紹介する。

なお、本稿では、フィルタをかける対象とするのは、白色雑音とする。白色雑音は次のように作成する。

ソースコード 1: 白色雑音の生成

```
1 >> fs=8000;
2 >> t=0:1/fs:1;
3 >> r=0.8*(-1+2*rand(size(t)));
4 >> n=1:100;
5 >> plot(t(n),r(n))
6 >> sound(r,fs)
```

このスペクトルは次のようにして見るができる。

```
>> fftsize=512;
>> S=fft(r(1:fftsize).*hann(fftsize)');
>> LS=log(abs(S));
>> f=1:fftsize/2;
>> plot(f,LS(f))
```

1 MATLAB によるフィルタリングの基本操作

デジタル信号処理のフィルタは、フィルタ係数とよばれる 2 つの数列で定義される。この系列は、MATLAB では、a、b で表わされることが多い。

最も簡単なフィルタの 1 つに移動平均フィルタがある。3 点移動平均フィルタは $a = [1]$ と $b = [1/3 \ 1/3 \ 1/3]$ で定義される。このフィルタを上記の白色雑音にかけるには次のようにすればよい。

ソースコード 2: filter を用いたフィルタリング

```
1 >> b=[1/3 1/3 1/3];
2 >> a=[1];
3 >> soundsc(filter(b,a,r),fs)
4 >> soundsc(r,fs)
```

練習 1.1 移動平均フィルタをかけるとどのように音が変化するか、聞き比べてみよ。

練習 1.2 移動平均フィルタをかけた音のスペクトルを表示して、どのように変化したか観察せよ。

MATLAB では、フィルタの特性を簡単に調べるために、fvtool というツールが用意されている。

ソースコード 3: fvtool の使用例

```
1 >> fvtool(b,a)
```

fvtool は、上記のように利用すると、振幅応答とよばれるフィルタの特性が表示される。この図を見ると、フィルタがどの周波数成分を強めたり弱めたりするかがわかる。

練習 1.3 n 点移動平均フィルタは、奇数の n に対して n 個の $1/n$ という値の数列で定義される。5 点、9 点、11 点の移動平均フィルタを作成し、点数が変わるとどのように特性が変化するかを考察せよ。

2 LPF, HPF, BPF

フィルタ係数をうまく設定することで様々な効果を得ることができる。フィルタは、その効果により分類されている。最もよく用いられるフィルタとして、ある周波数の範囲(帯域とよぶ)を通さない効果を持つものがある。MATLAB では、典型的なフィルタについては、求める効果から係数列を計算する関数が用意されている。

例えば、ある周波数より高い周波数を通さないフィルタはLPF(ローパスフィルタ)と呼ばれる。サンプリング周波数が8kHzの信号に対し、2kHz以上の成分を通さないLPFのフィルタ係数はfir1という関数を用いて次のように計算できる。このときの2kHzのことをカットオフ周波数と呼ぶ。

ソースコード 4: fir1 を用いた LPF の設計

```
>> b_lp2k=fir1(40,0.5)
b_lp2k =
  Columns 1 through 7
  -0.0000  -0.0014   0.0000   0.0024  -0.0000  -0.0046   0.0000
  Columns 8 through 14
  (略)
  -0.0046  -0.0000   0.0024   0.0000  -0.0014  -0.0000

>> stem(b_lp2k)
>> fvtool(b_lp2k,[1])
```

fir1 では、フィルタ係数の b の部分だけを生成する。この例では、40 点のフィルタ係数を計算している。stem でその係数をプロットしているが、かなり複雑な値であることがわかる。fvtool のプロットを見ればわかるように、fir1 では、カットオフ周波数で-6dB になるように減衰するように設計される。

この LPF をかけるには次のようにすればよい。

```
>> soundsc(filter(b_lp2k,[1],r),fs)
```

練習 2.1 ある周波数より低い周波数を通さないフィルタはHPF(ハイパスフィルタ)と呼ばれる。fir1 を用いて、適当な HPF を設計し、その特性を観察せよ。また適当な音にかけて、その効果を聞いて確認せよ。(ヒント: help fir1 とすると、fir1 の利用方法が調べられる。)

練習 2.2 ある周波数からある周波数までの間の成分だけを通すフィルタはBPF(バンドパスフィルタ)と呼ばれる。fir1 を用いて、適当な BPF を設計し、その特性を観察せよ。また適当な音にかけて、その効果を聞いて確認せよ。

3 シェルビングフィルタ

LPF などはよく使われるが、アンプなどのトーンコントロールやポータブル音楽プレイヤーのイコライザーに使われているフィルタは、ある帯域を完全に通さない、というような処理はしていない。

それらに使われるフィルタは、シェルビングフィルタとよばれるフィルタである。シェルビングフィルタを設計するツールは MATLAB では用意されていない。しかし、公開されているツールで設計できる。例えば、MATLAB Central という MATLAB の情報交換サイトからダウンロードできる関数 M-ファイル shelving.m がある¹ダウンロードして zip ファイルを解凍すると shelving.m というファイルが出てくるので、そのファイルを、自分の PC の [ドキュメント] フォルダの中の [MATLAB] フォルダにコピーすると使えるようになる。

正しくコピーできれば、次のようにして低域を弱めるシェルビングフィルタを設計できる。

```
>> [b,a]=shelving(-20,400,fs,0.8,'Base_Shelf');
>> fvtool(b,a)
```

¹<http://www.mathworks.com/matlabcentral/fileexchange/16568> の Download All のボタンを押すとダウンロードできる。

shelving の第 1 引数 (ここでは -20) はゲインといって、どの程度強調するか、もしくは減衰させるかを指定できる。値が負の場合は減衰され、正の場合は強調される。単位は dB である。第 2 引数は、中心周波数であり、減衰 (強調) のカーブの中心の周波数を指定する。単位は Hz である。第 3 引数はサンプリング周波数である。第 4 引数は減衰 (強調) のカーブの傾きを指定するパラメータである。0.7 から 0.8 あたりの値が適当である。第 5 引数は、低域を操作するか、高域を操作するかを指定する。上記の例では低域を操作している。高域を操作したい場合は、'Treble_Shelf' と指定する。

練習 3.1 shelving を使って、様々なシェルビングフィルタを設計し、特性を観察せよ。また、適当な音にかけて、その効果を確認せよ。