

## 10 FIR フィルタ

### 10.1 線型システム

デジタル信号処理の分野では、何らかの入力信号を出力信号に変換するものを「システム」と呼ぶ。

これまでに見てきた処理の中でも、例えば、 $x_1$  という波と  $x_2$  という波を足し合わせて  $y_1$  という波を作成する、という場合は、システムは「2つの入力を加算する」という変換を行う。これは式では

$$y_1 = x_1 + x_2 \quad (4)$$

となる。この入力信号、出力信号は、実際にはデジタルデータ (離散データ) である。そのことを反映して、次のように書くのが普通である。

$$y_1[n] = x_1[n] + x_2[n] \quad (5)$$

$n$  はデータの添字をあらわす。音声データの場合は、時刻と同じようなものだが、 $n$  と書くような場合は整数であることも意味している。

別の例として、入力信号  $x_3$  を  $a$  倍して  $y_2$  を作成するという場合は、

$$y_2[n] = ax_3[n] \quad (6)$$

と書ける。

あるシステムによって  $x_1$  が  $y_1$  に変換され、 $x_2$  が  $y_2$  に変換されるとする。ある条件を満たすシステムでは、 $x = a_1x_1 + a_2x_2$  が  $y = a_1y_1 + a_2y_2$  に変換される。このようなシステムは線型 (線形) システムとよばれる。

上記のシステムでは、例えば、 $u_1, u_2$  を加算する場合は、 $u_1 + u_2$  となる。 $a_1u_1, a_2u_2$  を加算すると  $a_1u_1 + a_2u_2$  となる。したがって、2つの信号を加算するシステムは線型システムである。同様に式 (6) で表わされるシステムも線型システムであることが簡単に示せる。

練習 28 線型システムでないシステムの例を考え、式を示せ。そのシステムが線型システムでないことを (式を用いて) 示せ。

### 10.2 遅延演算

デジタル信号処理の基本的な演算として、信号を定数倍する演算や、加算する演算を紹介してきた。もう一つの基本的な演算として、遅延がある。遅延は、入力を単位時間の定数倍だけ遅らせる演算である。(単位時間とは、例えばサンプリング周波数が 8kHz だとすると 1/8000 秒のことである。離散データにおいては、添字を 1 変化させることに対応する。)

遅延を式でどのように表わすかを考えてみる。1 単位時間遅らせて出力信号とする場合は、時刻  $n$  の入力 が 時刻  $n+1$  の出力となるということである。入力信号を  $x$ 、出力信号を  $y$  であらわすと、

$$y[n+1] = x[n] \quad (7)$$

となる。

練習 29 式 (7) を  $y[n] =$  の形で書き換えよ。また、5 単位時間遅らせる場合の式を  $y[n] =$  の形で書け。

遅らせたただけだと余り意味はないが、遅らせた信号を元の信号に足し合わせると信号を変化させられる。

### ソースコード 9: 遅延演算を用いたフィルタリング

```
1 >> Fs=8000;
2 >> t=0:1/Fs:1-1/Fs;
3 >> s=sin(2*pi*800*t)+sin(2*pi*500*t); % 2つの成分からなる信号を生成
4 >> soundsc(s,Fs)
5 >> subplot(3,1,1); plot(s(1:100));
6 >> sd=wshift(1,s,-5); % s を 5 点分遅らせる
7 >> subplot(3,1,2); plot(sd(1:100));
8 >> soundsc(sd,Fs)
9 >> ss=s+sd;
10 >> subplot(3,1,3); plot(ss(1:100));
11 >> soundsc(ss,Fs)
```

subplot は複数のプロットを並べる関数である (詳しい使い方は help subplot で調べよ)。また、wshift は信号を遅らせるのに使っている。厳密にいうと、ただ遅らせているわけではないが、s の長さを変化させないために wshift を利用している。

練習 30 5点遅らせて足し合わせることで、どちらかの成分が除去された。音を聞いたり、プロットを確認することで、どの成分が除去されたかを確認せよ。

練習 31 ソースコード 9 とは別の成分を除去するには、何点遅らせればよいか?

このように遅延演算と加算、定数倍演算を上手く組み合わせると、ある成分を除去したり強調したりすることができる。一般にはある成分を除去することが多いため、このような処理をデジタルフィルタと呼ぶ。

### 10.3 移動平均フィルタ

簡単だけれども音声処理だけでなく、画像処理や株価の計算など非常に様々な分野で利用されるフィルタに移動平均フィルタとよばれるものがある。

入力の連続する 3 点の平均を出力とするフィルタを考えてみる。式であらわすと次のようになる。

$$y[n] = \frac{x[n] + x[n-1] + x[n-2]}{3} \quad (8)$$

このフィルタの効果を音を対象に調べてみる。

まず、雑音を作る。

### ソースコード 10: 白色雑音の生成

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

ここでは、白色雑音 (ホワイトノイズ) とよばれる雑音を 1 秒間生成している。白色雑音とは一様分布の乱数列である。MATLAB では、音データ  $y$  は、 $-1 \leq y \leq 1$  を満たすようになっている。3 行目では、 $(0,1)$  区間の一様乱数を生成する rand 関数を利用して、 $(-1,1)$  区間の一様乱数を 1 秒間分生成している (どういうベクトルになっているかは、rand 関数と size 関数を help コマンドで調べて理解すること)。

練習 32 ソースコード 10 の r はどのような周波数成分を持つか、スペクトルもしくはスペクトログラムをプロットして考察せよ。

この雑音 r を少しだけ正弦波に加えると雑音混じりの音が生成できる。

### ソースコード 11: 雑音混じりの正弦波の生成

```
1 >> s=sin(2*pi*440*t);
2 >> sn=0.8*s+0.1*r;
3 >> n=1:100;
4 >> plot(t(n),sn(n),t(n),0.8*s(n))
5 >> sound(sn,fs)
```

4行目の plot 関数では、sn と s を重ねてプロットしている。

練習 33 ソースコード 11 の sn はどのような周波数成分を持つか、スペクトルもしくはスペクトログラムをプロットして確認せよ。

#### ソースコード 12: for ループを用いた 3 点移動平均フィルタ

```
1 >> for i=3:length(t),
2   y(i)=1/3*sum(sn(i+[-2 -1 0]));
3 end
4 >> plot(t(n),sn(n),t(n),y(n),t(n),0.8*s(n))
5 >> sound(y,fs)
6 >> sound(s,fs)
7 >> sound(sn,fs)
```

練習 34 ソースコード 12 を実行し、どのような効果があるかを確認せよ。また、s と sn と y を比較し、どのような変化が生じているかを述べよ。

式 (8) を書き直すと次のようになる。

$$y[n] = \frac{1}{3}x[n] + \frac{1}{3}x[n-1] + \frac{1}{3}x[n-2] \quad (9)$$

この式は、遅延演算と定数倍と加算だけで構成されていることがわかる。遅延演算は入力を遅らせるだけなので、いくつ遅らせるかがわかれば十分である。そこで、上記の演算を  $[1/3 \ 1/3 \ 1/3]$  と遅延の位置に定数倍の係数を与えて作成できるベクトルで表わすことがある。この係数ベクトルを利用して簡単にフィルタをかける演算が用意されている。その演算が畳み込み (コンボリューション) である。MATLAB では conv 関数で計算できる。

```
>> h=ones(1,3)/3;
>> y2=conv(h, sn);
>> plot(t(n),y2(n),t(n),y(n))
```

練習 35 help コマンドを利用したり、conv([1 2 3 4],[5 6 7 8]) などの計算を試すことで、conv とはどのような計算かを確認せよ。

練習 36 conv 関数を利用し、様々な長さの移動平均フィルタを作成せよ。それらの移動平均フィルタを用いてソースコード 11 で作成した sn に適用せよ。適用した結果を sound 関数で聴取したり、スペクトルやスペクトログラムで観察し、移動平均フィルタの長さを長くすることでどのような効果が得られるかを考察せよ。

練習 37 ソースコード 9 のフィルタの係数をあらわすベクトルを作成し、そのベクトルを conv で畳み込むことでソースコード 9 と同等の効果を作り出せ。

## 11 インパルス応答

システムの効果を示すために用いられる方法にインパルス応答がある。応答とは、入力に対して得られる出力のことを示す語である。インパルス応答とは、入力をインパルスとしたときの出力のことをいう。

次の式は単位インパルスを表す。

$$x[n] = \begin{cases} 1 & (n = 0) \\ 0 & (n > 0) \end{cases} \quad (10)$$

このように 1 点だけ値がある信号をインパルスという。

1000 点のインパルスは MATLAB では次のように作ることができる。

```
>> i1000(1)=1;
>> i1000(1000)=0;
```

練習 38 インパルスのスペクトルを求め、どのような特徴があるか述べよ。

3 点の移動平均フィルタのインパルス応答は次のように求められる。

```
>> h=ones(1,3)/3;
>> h_i=conv(h,i1000);
>> stem(h_i)
```

練習 39 適当なフィルタのインパルス応答のスペクトルを観察せよ。また、そのフィルタをいくつかの信号に適用し、その結果をインパルス応答のスペクトルを用いて考察せよ。

インパルス応答のスペクトルのことをフィルタの周波数応答とよぶ(周波数特性ともいう)。

## 12 IIR フィルタ

信号処理では、出力をもう一度入力として利用することで効率よい処理を行えることがある。

$$y[n] = -0.5y[n-5] + x[n] \quad (11)$$

という式を考える。この式は

$$y[n] = -0.5x[n-5] + x[n] \quad (12)$$

と形は似ているが、得られる出力は全然違う。

練習 40 10 点の単位インパルス

$$x[n] = \begin{cases} 1 & (n = 0) \\ 0 & (1 \leq n \leq 10) \end{cases} \quad (13)$$

に対し、式 (11) と式 (12) のインパルス応答を計算してみる。

式 (13) を式 (11) にあてはめると、

$$y[0] = -0.5y[-5] + x[0] = 1$$

$$y[1] = -0.5y[-4] + x[1] = 0$$

...

となる。

この後の値を計算し、インパルス応答を完成させよ。また式 (12) についても同様にインパルス応答を計算せよ。

このように出力が入力に用いられることをフィードバックと呼ぶ。フィードバックがあるとインパルス応答は無限に続くためこのようなタイプのフィルタは IIR (infinite impulse response) フィルタと呼ばれる。一方で、フィードバックがないフィルタのインパルス応答は有限なので、FIR (finite impulse response) フィルタと呼ばれる。

IIR フィルタの係数はフィードバックを考慮しないといけないため、FIR フィルタの係数とは同様には表現できない。そこで、入力に関する項は右辺、出力に関する項は左辺とまとめる。例えば、式 (11) は次のように変形する。

$$y[n] + 0.5y[n-5] = x[n] \quad (14)$$

これを出力の係数、入力の係数、別々のベクトル  $\mathbf{b}$  と  $\mathbf{a}$  で表すとする。この 2 つのベクトルを用いると、`filter` 関数でフィルタを入力に適用できる。

```
>> ir_iir = filter(b,a,i1000);
```

練習 41 a と b を適切に定めて式 (11) の周波数応答を求めよ。

## 13 フィルタ設計のツール

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。

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 のフィルタ係数は次のように計算できる。

### ソースコード 13: fir1 を用いた LPF の設計

```
>> h_lp2k=fir1(40,0.5);  
>> stem(h_lp2k)
```

かなり複雑な係数であることがわかる。

練習 42 ソースコード 13 のフィルタの周波数応答を求めよ。

練習 43 ソースコード 13 のフィルタをソースコード 11 の sn に適用しどのような音になるか調べよ。

練習 44 fir1 を用いて、550Hz 以下の低域だけを通過させる LPF を設計せよ。その LPF をソースコード 11 の sn に適用し、効果を確認せよ。

LPF は IIR フィルタでも設計できる。

### ソースコード 14: butter を用いた LPF の設計

```
>> [b,a]=butter(10,0.5);
```

練習 45 ソースコード 14 のフィルタの周波数応答を求めよ。

練習 46 IIR フィルタでは、FIR フィルタでは実現が難しいノッチフィルタ (notch filter) の設計も可能である。ノッチフィルタとはどのようなフィルタかを調べ、iirnotch を用いて適当なノッチフィルタを設計せよ。また、実際の音に適用し、その効果を観察せよ。

## 課題

以下の課題をパワーポイントファイルに作成し、授業支援システムのみ Navi の第 4 回目の「準備課題」のところにアップロードすること。パワーポイントファイルは、18 ポイント以上の大きさの文字を使って作成すること。締切は 10/10(日) 午後 8 時とします。自動で締め切ってしまいますので、十分に注意して下さい。音を生成・加工する課題については、音データもパワーポイントのスライドに貼り付けるようにすること。

1. 練習 28 を解答せよ。
2. 練習 29 を解答せよ。
3. 練習 32 を解答せよ。
4. 練習 36 を解答せよ。
5. 練習 40 を解答せよ。
6. この資料を読んで、わからなかった点があれば、わからなかった点が解消されるような回答が期待できるような具体的な質問を考えて書くこと。(これに関しては、スライドではなく、別のファイルに作成してアップロードせよ)