

信号処理とフーリエ変換 第14回

～デジタル・フィルター (2)～

かつらだ まさし
桂田 祐史

2022年1月12日

- ① 本日の内容・連絡事項
- ② デジタル・フィルタ (続き)
 - デジタル・フィルタを作る (続き)
 - ローパス・フィルタ

いよいよ最終回です。

- デジタル・フィルターの2回目 (講義ノート [1]§8.6の後半)。
今日の授業は短いです。(シラバスに載せていた「CTの数理」の説明はカットします。興味のある人は講義ノート§10を読んで下さい。)
- 期末レポート課題については、第13回授業資料 pp.3-4で紹介した。日程は次のように決定した。
1月21日(金曜) 16:00に課題公開, 1月23日(日曜)23:00に提出。
- それとは別にレポート課題3を出す。〆切は1月30日(日曜)。
- 質問用 Zoom ミーティングは後1回あります (2022年1月18日 12:00-13:00)。

低い音は通すが、高い音はカットする、**ローパス・フィルタ** (low-pass filter) を作る。

最初は素朴に考える。 $F_e > 0$ として、次を目標にする。

F_e 以下の周波数の信号はそのまま通し、 F_e より高い周波数の信号は一切通さない。

その周波数 F_e に対応する正規化角周波数 ω_e は、 $\omega_e = \frac{F_e}{F_s}$ 。そこで、デジタル・フィルタ F で、単位インパルス応答 $h = F[\delta]$ が

$$(1) \quad \hat{h}(\omega) = \begin{cases} 1 & (|\omega| \leq \omega_e) \\ 0 & (|\omega| > \omega_e) \end{cases} \quad (\text{理想的な周波数特性})$$

を満たすものを求める (思い出し: $y_n = \hat{h}(\omega)e^{in\omega}$)。

h_n は? 離散時間 Fourier 変換の反転公式 $h_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{h}(\omega)e^{in\omega} d\omega$ から

$$(2) \quad h_n = \frac{1}{2\pi} \int_{-\omega_e}^{\omega_e} e^{in\omega} d\omega = \frac{\omega_e}{\pi} \text{sinc}(n\omega_e).$$

$$\text{ここで } \text{sinc } x := \begin{cases} \frac{\sin x}{x} & (x \in \mathbb{R} \setminus \{0\}) \\ 1 & (x = 0) \end{cases}.$$

一般に $\int_{-a}^a e^{ibx} dx = 2a \text{sinc}(ab)$ が成り立つので、 $=$ が得られる。

8.5.7 ローパス・フィルター

念のため (前回の記憶がかすかな人のため): つまり、 $\{h_n\}_{n \in \mathbb{Z}}$ を

$$h_n := \frac{1}{2\pi} \int_{-\omega_e}^{\omega_e} e^{in\omega} d\omega = \frac{\omega_e}{\pi} \operatorname{sinc}(n\omega_e)$$

で定めると

$$\sum_{n=-\infty}^{\infty} h_n e^{-in\omega} = \hat{h}(\omega) = \begin{cases} 1 & (|\omega| \leq \omega_e \text{ の場合}) \\ 0 & (|\omega| > \omega_e \text{ の場合}) \end{cases}$$

が成り立ち、

$$y_n = x * h(n) = \sum_{k=-\infty}^{\infty} x_{n-k} h_k \quad (n \in \mathbb{Z})$$

で $\{y_n\}$ を計算すると、離散正弦波 $\{x_n\}_{n \in \mathbb{Z}} = \{e^{in\omega}\}_{n \in \mathbb{Z}}$ に対しては

$$y_n = \hat{h}(\omega)x_n = \begin{cases} x_n & (|\omega| \leq \omega_e \text{ の場合}) \\ 0 & (|\omega| > \omega_e \text{ の場合}). \end{cases}$$

つまり、 F_e より低い周波数の信号はそのまま通し、 F_e より高い周波数の信号は完璧にシャットアウトする。

しかし、フィルターをコンピューター上に実現するとき、**無限級数の計算をすることは現実的ではない**。何らかの形で有限項で打ち切ることになるだろう。

8.5.7 ローパス・フィルタ F^J (単純な打ち切り)

まず素朴に単純な打ち切りを試す。 $J \in \mathbb{N}$ に対して $\sum_{k=-\infty}^{\infty}$ を $\sum_{k=-J/2}^{J/2}$ で置き換える。

これは F の代わりに、次式で定まる F^J を考えていることになる:

$$F^J[x](n) := \sum_{k=-J/2}^{J/2} x_{n-k} h_k \quad (n \in \mathbb{Z}).$$

$\{h_n\}_{n \in \mathbb{Z}}$ の代わりに、次式の $\{h_n^J\}_{n \in \mathbb{Z}}$ を用いる、ということである。

$$h_n^J := \begin{cases} h_n & (|n| \leq J/2) \\ 0 & (|n| > J/2). \end{cases}$$

この F^J の周波数特性は

$$\hat{h}^J(\omega) = \sum_{n=-\infty}^{\infty} h_n^J e^{-in\omega} = \sum_{n=-J/2}^{J/2} h_n e^{-in\omega}.$$

この $\hat{h}^J(\omega)$ のグラフを描いてみよう。

8.5.7 ローパス・フィルター F^J (単純打ち切り) 周波数特性

naivelowpass.nb — $\hat{h}^J(\omega)$ のグラフを描く

```
omega=0.5  
h[n_]:=omega/Pi Sinc[n omega]  
draw[J_]:=Plot[Sum[h[n]Exp[-I n t],{n,-J/2,J/2}],{t,-Pi,Pi}, PlotRange->All]  
draw[100]
```

目標は

```
Plot[If[Abs[x]<omega,1,0],{x,-Pi,Pi}]
```

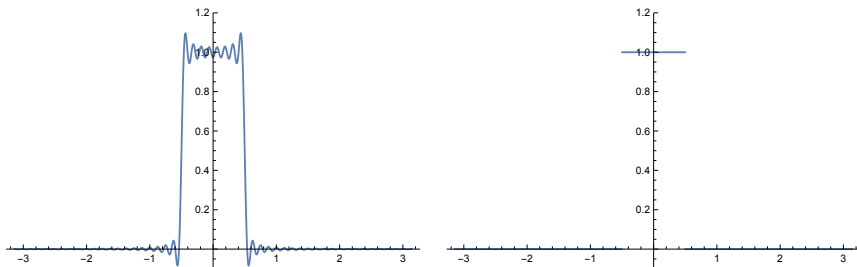


図 1: $\hat{h}_J(\omega)$ ($J = 100$) と目標 $\hat{h}(\omega)$. h_J の周波数特性 $\hat{h}_J(\omega)$ は “波打っている”

8.5.7 ローパス・フィルター F^J (単純打ち切り) 反省

(1枚前のスライド) 何か見覚えがあるような…Gibbsの現象である！

不連続関数の Fourier 級数の部分和は、Fourier 級数の和 (極限) と大きな隔たりがある、ということ。

条件

$$(再掲 1) \quad \hat{h}(\omega) = \begin{cases} 1 & (|\omega| \leq \omega_e) \\ 0 & (|\omega| > \omega_e) \end{cases} \quad (\text{理想的な周波数特性})$$

は1つの理想であるが、不連続関数であるから、無限級数の計算が出来ない限り、かえって良くないことが起こる、ということである。

信号処理では、**windowing** (窓をかける) というテクニックを用いて対処する。

h_n に**窓 (関数)** と呼ばれる (段階的に 0 に近づき、十分遠方では恒等的に 0 に等しい) 関数をかけて、0 でない項を有限個のみにする。窓関数には色々なものがあるが、ここでは次式で定義されるシンプルな **hann 窓** を用いてみる。

$$(3) \quad w(x) := \frac{1 - \cos 2\pi x}{2} \quad (0 \leq x \leq 1).$$

hann 窓ってどんな関数？

```
w[x_] := (1 - Cos[2Pi x])/2;
g = Plot[w[x], {x, 0, 1}]
```

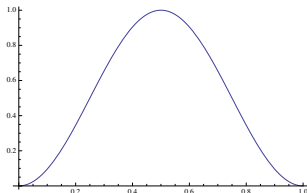


図 2: ハン窓 w のグラフ

この w を用いて、

$$h_n^{J,w} := \begin{cases} w(n/J - 1/2)h_n & (|n| \leq J/2) \\ 0 & (|n| > J/2) \end{cases}$$

で $\{h_n^{J,w}\}_{n \in \mathbb{Z}}$ を定め、 $\{h_n\}_{n \in \mathbb{Z}}$ の代わりに $\{h_n^{J,w}\}_{n \in \mathbb{Z}}$ を用いることにする。

これは F に近いが、 F とは異なるデジタル・フィルター $F^{J,w}$ を用いることになる。その周波数特性は

$$\hat{h}^{J,w}(\omega) = \sum_{n=-\infty}^{\infty} h_n^{J,w} e^{-in\omega} = \sum_{n=-J/2}^{J/2} h_n^{J,w} e^{-in\omega}.$$

この $\omega \mapsto \hat{h}^{J,w}(\omega)$ のグラフを図示してみよう。

```
w[x_] := (1 - Cos[2 Pi x])/2
draw2[J_] := Plot[Sum[w[n/J-1/2]h[n]Exp[-I n t], {n, -J/2, J/2}], {t, -Pi, Pi},
  PlotRange -> All]
```

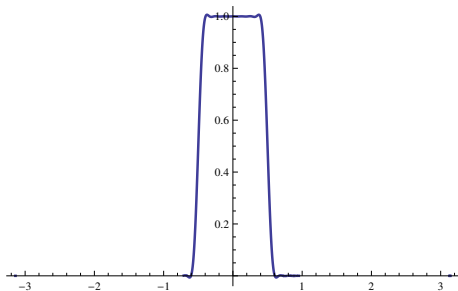


図 3: 窓関数あり $\hat{h}^{J,w}(\omega)$

これは (理想である) $\hat{h}(\omega)$ とは違うけれど、 $\hat{h}_J(\omega)$ よりはずっと良いだろう。

注意

$$y_n = \sum_{k=-\infty}^{\infty} x_{n-k} h_k^{J,w} = \sum_{k=-J/2}^{J/2} x_{n-k} h_k^{J,w}.$$

y_n の計算に $x_{n-J/2}, x_{n-J/2+1}, \dots, x_{n+J/2}$ が必要になることに注意しよう。

フィルター $F^J, F^{J,w}$ は、以前説明した定義に従えば厳密には FIR フィルターではない (未来の情報を使っているから) が、時間遅れを許すことにして

$$\begin{aligned} \tilde{y}_n &:= y_{n-J/2} = \sum_{k=-J/2}^{J/2} x_{(n-J/2)-k} h_k^{J,w} = \sum_{k=-J/2}^{J/2} x_{n-(J/2+k)} h_k^{J,w} \\ &= \sum_{k'=0}^J x_{n-k'} h_{k'-J/2}^{J,w} \end{aligned}$$

とすると、FIR フィルターとなる。

最後まで視聴してくれた人、お疲れさまです。

参考文献

- [1] 桂田祐史：「信号処理とフーリエ変換」講義ノート, <http://nalab.mind.meiji.ac.jp/~mk/fourier/fourier-lecture-notes.pdf>, 以前は「画像処理とフーリエ変換」というタイトルだったのを直した。(2014～).