

信号処理とフーリエ変換 第13回

～デジタル・フィルター (1)～

かつらだ まさし
桂田 祐史

<http://nalab.mind.meiji.ac.jp/~mk/fourier2022/>

2023年1月11日

目次

- 1 本日の内容・連絡事項
- 2 デジタル・フィルタ
 - 離散信号
 - 畳み込みと単位インパルス
 - 線形定常フィルタ
 - FIR フィルタ
 - デジタル・フィルタを作る
 - はじめに
 - 用語の確認 サンプルング、サンプルング周期、サンプルング (角) 周波数
 - 正弦波をサンプルングすると等比数列
 - 正規化 (角) 周波数
 - 元の連続信号の周波数と離散化信号の周波数の関係
 - 離散化した正弦波をフィルタに入力すると — フィルタの周波数特性
- 3 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
- 4 参考文献

本日の内容・連絡事項

- 課題2について質問があるならば、今日の授業終了後あるいはメールで質問して下さい。
- 今回と次回でデジタル・フィルタについて解説する(講義ノート[1]の§8の内容)。線形定常フィルタが単位インパルス応答との畳み込みで表現できることを示し、線形定常フィルタの周波数特性について説明し、例として、ローパス・フィルタを取り上げる。
- 授業の進行はほぼ例年通りなので、期末試験範囲も例年とほぼ同じとなる。内容については過去問が参考になるほか、レポート課題1,2の内容も復習しておくとうい。
- レポート課題3を(次回授業で)出します(デジタル・フィルタについての内容)。〆切りは2023年1月30日(月曜)です。

8 デジタル・フィルター

8.1 離散信号

デジタル・フィルターとは、離散信号を入力して、離散信号を出力するもの (数学的には写像) である。(フィルターという言葉は知っているのかな? ¹⁾)

ここで**離散信号**とは複素数列 (ただし負の番号も使う) のことである。すなわち、複素数列 $x = \{x_n\}_{n \in \mathbb{Z}}$ を離散信号と呼ぶ。離散信号の全体を S で表す。

離散信号は \mathbb{Z} から \mathbb{C} への写像とみなせる。ゆえに $S = \mathbb{C}^{\mathbb{Z}}$ 。

S では、自然に和 $x + y$, スカラー倍 cx が定義される (ただし $c \in \mathbb{C}$)。

$$(x + y)(n) = x(n) + y(n), \quad (cx)(n) = cx(n) \quad (n \in \mathbb{Z}).$$

S は \mathbb{C} 上のベクトル空間となる。

¹filter について、辞書を引くことを勧めます。

8.2 畳み込みと単位インパルス

$x, y \in \mathcal{S}$ に対して、**畳み込み** $x * y \in \mathcal{S}$ を

$$x * y(n) := \sum_{k=-\infty}^{\infty} x(n-k)y(k) \quad (n \in \mathbb{Z})$$

で定める (本当は収束のための仮定が必要であるが省略する)。

交換法則、結合法則、分配法則などが成り立つ — 以上は復習である。

$\delta \in \mathcal{S}$ を

$$\delta(n) = \delta_{n0} = \begin{cases} 1 & (n = 0) \\ 0 & (n \neq 0) \end{cases}$$

で定める (δ_{nm} は Kronecker のデルタ)。 δ を**単位インパルス** (the unit impulse) と呼ぶ。

δ は畳み込みに関する**単位元**である。すなわち、任意の $x \in \mathcal{S}$ に対して

$$(1) \quad x * \delta = \delta * x = x$$

が成り立つ。実際、任意の $x \in \mathcal{S}$, $n \in \mathbb{Z}$ に対して

$$x * \delta(n) = \sum_{k=-\infty}^{\infty} x(n-k)\delta(k) = x(n-0) \cdot \delta(0) = x(n) \cdot 1 = x(n).$$

ゆえに $x * \delta = x$.

□

8.3 線形定常フィルター

S から S への写像を**デジタル・フィルター**と呼ぶ。

デジタル・フィルター $F: S \rightarrow S$ が**線形** (linear) であるとは

$$(\forall x, y \in S) \quad F[x + y] = F[x] + F[y],$$

$$(\forall x \in S)(\forall c \in \mathbb{C}) \quad F[cx] = cF[x]$$

を満たすことをいう。

$x \in S, k \in \mathbb{Z}$ に対して

$$y(n) := x(n - k) \quad (n \in \mathbb{Z})$$

で定まる $y \in S$ のことを $x(\cdot - k)$ と表す。 y は “ x の時間を k だけずらしたものである”。 $k > 0$ のときは遅らせたもの。 $k < 0$ のときは早めたもの。

\cdot は変数をここに代入するという意味である。つまり $x(\cdot - k)$ は、 $n \mapsto x(n - k)$ という写像を意味する記号である。

線形デジタルフィルター F が**定常** (^{じふへん}**時不変**, time invariant) であるとは

$$(\heartsuit) \quad (\forall x \in S)(\forall k \in \mathbb{Z}) \quad F[x(\cdot - k)] = F[x](\cdot - k)$$

を満たすことをいう。

(つづく)

8.3 線形定常フィルター (続き)

条件 (♡) は分かりにくいと思われる。別の書き方をしてみる (2回説明すると、どちらかで分かるかも、という期待)。

これは、時間 k だけのシフト $S_k: S \rightarrow S$ を

$$S_k[x] = x(\cdot - k) \quad (x \in S)$$

で定めたとき ($S_k[x]$ は、信号 x を k だけ遅延させた信号)、

$$(\forall k \in \mathbb{Z})(\forall x \in S) \quad F[S_k[x]] = S_k[F[x]]$$

が成り立つこと、すなわち

$$(\forall k \in \mathbb{Z}) \quad F \circ S_k = S_k \circ F$$

が成り立つこと、と言い換えられる。 F が時間シフト S_k と交換可能ということ。…かえって分かりにくい？

たとえ話: 拡声器があり、 x を入力音声、 $F[x]$ をスピーカーから出力される音声とする。いつでも同じように拡声する (夜中だから音を小さくするとかしない)。それが定常ということである。

8.3 線形定常フィルター 単位インパルス応答

次の定理がきわめつけに重要である。

定理 13.1 (線形定常フィルターは、単位インパルス応答との畳み込みで表せる)

線形定常デジタルフィルター $F: \mathcal{S} \rightarrow \mathcal{S}$ に対して

$$h := F[\delta]$$

とおくと

$$(\forall x \in \mathcal{S}) \quad F[x] = x * h = h * x.$$

$h = F[\delta]$ を F の **単位インパルス応答** (the unit impulse response) と呼ぶ。

LTI フィルター F に対して、単位インパルス δ を入力したときの出力 h が分かれば、任意の入力 x に対する出力は、 h との畳み込みを計算すれば得られる。

お話: 離散信号以外でも、 δ が定義されて、フィルターの h に相当するものがある。微分方程式の場合は、**基本解** がそれに相当する。「畳み込みの例 電荷の作る静電場の電位」や「熱方程式の基本解」はそういう話である。

8.3 線形定常フィルター 定理 13.1 の証明

$x \in \mathcal{S}$ とする。上で見たように $x = \delta * x$. すなわち

$$x = \sum_{k=-\infty}^{\infty} \delta(\cdot - k)x(k).$$

(第 k 項は、信号 x の時刻 k での値だけ取り出した信号である。) ゆえに

$$F[x] = F \left[\sum_{k=-\infty}^{\infty} \delta(\cdot - k)x(k) \right] \quad (\text{代入した})$$

$$= \sum_{k=-\infty}^{\infty} x(k)F[\delta(\cdot - k)] \quad (\text{線形性})$$

$$= \sum_{k=-\infty}^{\infty} x(k)F[\delta](\cdot - k) \quad (\text{定常性})$$

$$= \sum_{k=-\infty}^{\infty} x(k)h(\cdot - k) \quad (F[\delta] = h)$$

$$= h * x. \quad (\text{畳み込みの定義}) \quad \square$$

8.4 FIR フィルター

LTI フィルター F が **FIR (有限インパルス応答, finite impulse response)** とは、 F の単位インパルス応答 h が次の条件を満たすことをいう。

$$(\exists J \in \mathbb{N})(\forall k \in \mathbb{Z} : k < 0 \vee k > J) \quad h(k) = 0$$

言い換えると、 $h(k) \neq 0$ となる k は、 $0 \leq k \leq J$ を満たす。

このとき $h(0), h(1), \dots, h(J)$ を F の**フィルター係数**と呼ぶ。

F が FIR フィルターならば、任意の $x \in S$ に対して

$$F[x](n) = \sum_{k=0}^J x(n-k)h(k) \quad (n \in \mathbb{Z}).$$

(未来の情報を使わない、計算を有限和にしたい、ということ。)

これからデジタル・フィルタを構成する話をするが、どういうことをしたいのかイメージを持ってもらうために、piano-cutoff.nb というサンプル・プログラムを用意してある。

```
curl -O https://m-katsurada.sakura.ne.jp/fourier/piano-cutoff.nb  
open piano-cutoff.nb
```

前々回 (第 10 回) の授業で試してみた。そのときのことを覚えていると仮定して、以下の説明を行う。(まだ視聴していない人は視聴して下さい。)

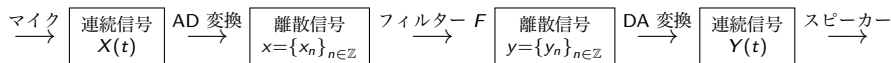
このプログラムは、ある周波数よりも高い周波数の信号成分をカットする、という処理をしている。ここでは信号全体を離散 Fourier 変換してから処理しているが (高い周波数に対応する離散 Fourier 係数を 0 にする)、FIR フィルタを作れば、それをしなくても出来る (ほぼリアルタイムで — 正確に言うとなぜかな時間遅れで — 処理できる)。

8.5 デジタル・フィルタを作る

8.5.1 はじめに

LTI フィルタの、任意入力に対する出力が、単位インパルス応答との畳み込みで表される、という定理 (定理 13.1) を紹介した。ここでは、ローパス・フィルタを例にあげて、より具体的に説明する。

例えば音の場合、「話して」出た声をマイクで拾って、フィルタで処理した後にスピーカーで流したものを「聴いて」効果を確認される。



全体の処理の流れ: 1次元の連続信号 (アナログ信号) をサンプリングして離散信号 (デジタル信号) を求め (ある種の AD 変換をしたことになる)、線形定常なデジタル・フィルタ (LTI フィルタ) F に入力して出力を得て、さらに DA 変換して連続信号を出力する。

8.5.2 用語の確認 サンプルング、サンプルング周期, サンプルング(角)周波数

連続信号 $X = X(t)$ を、**サンプルング周期** T_s で**サンプルングする**とは

$$(2) \quad x_n = X(nT_s) \quad (n \in \mathbb{Z})$$

で $x = \{x_n\}_{n \in \mathbb{Z}}$ を定めることをいう。

サンプルング周波数 F_s , **サンプルング角周波数** Ω_s は

$$(3) \quad F_s := \frac{1}{T_s}, \quad \Omega_s := 2\pi F_s$$

で定義される。

一般に、**周波数とは周期の逆数**である。

周波数というものはあちこちで出て来るが、一般に、周波数に 2π をかけたものを**角周波数**と呼ぶ。

例えば $\sin 2\pi ft$ は周波数 f の正弦波であるが、角周波数 $\omega := 2\pi f$ を使うと、 $\sin \omega t$ という簡潔な式で表せる。

(注: 私の資料は、サンプルング周波数を f_s と小文字の f で書いたりしています。不統一ですが、添字に s をつけるのは他に T_s だけなので、混同して間違えることはないと思います。)

8.5.3 正弦波 $e^{i\Omega t}$ をサンプリングすると等比数列

正弦波 $e^{i\Omega t}$ をサンプリングすると、等比数列になることを説明しよう。

$X(t) = e^{i\Omega t}$ ($\Omega \in \mathbb{R}$) とする。サンプリングすると

$$(4) \quad x_n = X(nT_s) = e^{i\Omega nT_s} = e^{in\omega} \quad (n \in \mathbb{Z}),$$

ただし

$$(5) \quad \omega := \Omega T_s.$$

$x_n = (e^{i\omega})^n$ であるから、 $x := \{x_n\}_{n \in \mathbb{Z}}$ は公比 $e^{i\omega}$ の等比数列である。

複素指数関数はサンプリングすると、等比数列になる。

(等比数列は離散版指数関数みたいなもの、まあ自然)

注意 ここでは $X(t) = e^{i\Omega t}$ のことを“正弦波”と呼んでいる。正弦波とは、本来は $X(t) = C_1 \sin \Omega t + C_2 \cos \Omega t$ (あるいは $X(t) = A \sin(\Omega t + \Phi)$) の形の信号のことを指すが、

$$C_1 \cos \Omega t + C_2 \sin \Omega t = \frac{C_1 - iC_2}{2} e^{i\Omega t} + \frac{C_1 + iC_2}{2} e^{-i\Omega t}$$

であるから、 $e^{i\Omega t}$ について調べれば十分である。

8.5.3 正弦波 $e^{i\Omega t}$ をサンプリングすると等比数列

連続信号として、なぜ特に正弦波 $e^{i\Omega t}$ を考えるのか (そのココロは)? — 任意の信号は $e^{i\Omega t}$ の重ね合わせで表せるから。実際、任意の信号 $X(t)$ は

$$X(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{X}(\Omega) e^{i\Omega t} d\Omega, \quad \hat{X}(\Omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} X(t) e^{-i\Omega t} dt$$

と表せる (Fourier 反転公式) ので、 $X(t)$ は $e^{i\Omega t}$ ($\Omega \in \mathbb{R}$) の線形結合と言える。

さらに、フィルター F が線形の場合は、離散化した正弦波 $x = \{e^{in\omega}\}$ の出力 $F[e^{in\omega}]$ を重ね合わせれば一般の入力に対する出力が得られることにも注意しよう。

線形フィルターの場合は、分解して考えられる (後から総和をとれば良い)

8.5.4 正規化(角)周波数

(正弦波 $X(t) = e^{i\Omega t}$ をサンプリング周期 T_s でサンプリングして、離散信号 $x_n = e^{i\omega t}$, $\omega = \Omega T_s$ を得ている。サンプリング周波数 F_s を用いると、 $\omega = \Omega/F_s$.)

サンプリング定理によると、サンプリング角周波数 $\Omega_s > 0$ でサンプリングして、きちんと復元できるためには、

$$(6) \quad |\Omega| < \frac{\Omega_s}{2}$$

が成り立っていれば良い。このとき次式が成立する。

$$(7) \quad |\omega| < \pi.$$

一般の Ω に対しては、 $\omega := \Omega T_s \in (-\pi, \pi)$ とは限らない。

$$(8) \quad \omega' \equiv \omega \pmod{2\pi}, \quad \omega' \in (-\pi, \pi]$$

となる ω' を取ることが出来る (ω' は ω を 2π で割った余りである。範囲が $(-\pi, \pi]$ であることに注意が必要だが。)

この ω' を**正規化角周波数**と呼ぶ。また次式で定まる f を**正規化周波数**と呼ぶ。

$$(9) \quad f := \frac{\omega'}{2\pi}.$$

正規化角周波数 ω' に対しても、次式が成り立つ (Cf. $x_n = e^{in\omega}$).

$$(10) \quad x_n = e^{in\omega'} \quad (n \in \mathbb{Z}).$$

8.5.5 連続信号の周波数と離散化信号の周波数の関係

連続信号 $e^{i\Omega t} (= e^{2\pi i F t})$ をサンプリングして離散信号 $e^{i\omega n} (= e^{2\pi i f n})$ を求めた。

元の連続信号の角周波数 Ω と離散信号の角周波数 ω の関係は？ ($\omega \stackrel{\text{def.}}{=} \Omega T_s$)
(あるいは元の連続信号の周波数 F と離散信号の周波数 f の関係は？)

$$(11) \quad \Omega = F_s \omega, \quad F = F_s f \quad (\text{どちらも } F_s = \frac{1}{T_s} \text{ をかければ良い}).$$

実際 $\omega = \Omega T_s$ であるから

$$\Omega = \frac{\omega}{T_s} = F_s \omega, \quad F = \frac{\Omega}{2\pi} = F_s \frac{\omega}{2\pi} = F_s f.$$

念のため記号の意味のおさらい

- Ω, F はそれぞれ元の連続信号の角周波数, 周波数 ($X(t) = e^{i\Omega t}, \Omega = 2\pi F$)
- ω, f はそれぞれサンプリングで得た離散信号の角周波数, 周波数
($\omega = \Omega T_s, \omega = 2\pi f$)
- F_s, T_s はそれぞれサンプリング周波数, サンプルング周期 ($F_s = \frac{1}{T_s}$)

問 ある正弦波をサンプリング周波数 $F_s = 44100\text{Hz}$ でサンプリングしたら、得られた離散信号の正規化角周波数 $\omega = \pi/10$ であった。もとの正弦波の周波数 F を求めよ。

解 $\Omega = F_s \omega, \Omega = 2\pi F$ であるから、 $F = \frac{\Omega}{2\pi} = \frac{F_s \omega}{2\pi} = \frac{44100\text{Hz}}{2\pi} \times \frac{\pi}{10} \doteq 2205 \text{ Hz}$. □

8.5.6 フィルターの周波数特性

LTI フィルター F に、離散正弦波 $\{e^{in\omega}\}_{n \in \mathbb{Z}}$ を入力したときの出力を調べよう。

線型定常であるから $h := F[\delta]$ とおくと、 $F[x] = x * h$.

$x_n = e^{in\omega}$ ($n \in \mathbb{Z}$), $y := F[x]$ とおくと

$$y_n = \sum_{k=-\infty}^{\infty} x_{n-k} h_k = \sum_{k=-\infty}^{\infty} e^{i(n-k)\omega} h_k = e^{in\omega} \sum_{k=-\infty}^{\infty} e^{-ik\omega} h_k$$

ゆえに、 h の離散時間フーリエ変換

$$(12) \quad \hat{h}(\omega) := \sum_{k=-\infty}^{\infty} e^{-ik\omega} h_k \quad (\omega \in [-\pi, \pi])$$

を用いると

$$(13) \quad y_n = e^{in\omega} \hat{h}(\omega) \quad (n \in \mathbb{Z}).$$

出力信号 $y = \{y_n\}$ は、入力信号 x と同様な離散正弦波であり、その角周波数は入力信号のそれと同じ ω である。つまり、次のことが分かった。

正弦波を線形定常フィルターに入力すると、同じ周波数の正弦波が出力される。

$\hat{h}(\omega)$ は、フィルター F の**周波数応答** (frequency response), **周波数特性** (frequency characteristic) と呼ばれる。角周波数 ω の信号の“増幅率”とも呼ぶべきものである。

おまけ: z 変換、伝達関数

信号処理のテキストでは、この周波数応答を、 h の z 変換を用いて表現してあるものが多い。一応紹介しておく。 $h = \{h_n\}$ の **z 変換**とは、

$$(14) \quad H(z) := \sum_{k=-\infty}^{\infty} \frac{h_k}{z^k} \quad (\text{Laurent 級数ですね})$$

で定義される複素関数 $H(z)$ で、これを用いると

$$(15) \quad \hat{h}(\omega) = H(e^{i\omega}).$$

$H(z)$ をフィルター F の**伝達関数** (transfer function) と呼ぶ。

周波数応答の絶対値と偏角に名前がついている。

- $G(\omega) := |\hat{h}(\omega)| = |H(e^{i\omega})|$ を**利得** (gain) と呼ぶ。
- $\theta(\omega) := \arg \hat{h}(\omega) = \arg H(e^{i\omega})$ を**位相シフト** (phase shift) と呼ぶ。
(信号処理の本では、 \arg を \angle と書くことがある。)

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

$$(16) \quad \hat{h}(\omega) = \begin{cases} 1 & (|\omega| \leq \omega_e) \\ 0 & (|\omega| > \omega_e) \end{cases} \quad (\text{理想的な周波数特性})$$

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

$$(17) \quad h_n = \frac{1}{2\pi} \int_{-\omega_e}^{\omega_e} e^{in\omega} d\omega = \frac{\omega_e}{\pi} \text{sinc}(n\omega_e).$$

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

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・フィルター 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}]
```

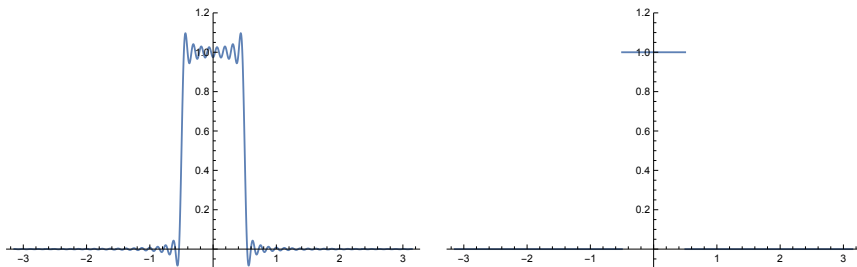


図 1: $\hat{h}_J(\omega)$ ($J = 100$) と目標 $\hat{h}(\omega)$. h_J の周波数特性 $\hat{h}_J(\omega)$ は “波打っている”

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

$$(再掲 16) \quad \hat{h}(\omega) = \begin{cases} 1 & (|\omega| \leq \omega_e) \\ 0 & (|\omega| > \omega_e) \end{cases} \quad (\text{理想的な周波数特性})$$

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。

$$(18) \quad w(x) := \frac{1 - \cos 2\pi x}{2} \quad (0 \leq x \leq 1).$$

hann 窓ってどんな関数？

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

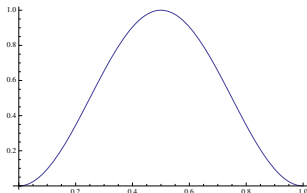


図 2: ハン窓 w のグラフ

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]
```

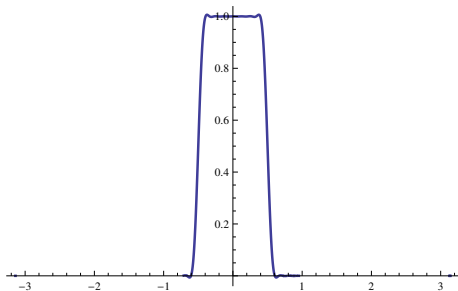


図 3: 窓関数あり $\hat{h}^{J,w}(\omega)$

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。

- [1] 桂田祐史：「信号処理とフーリエ変換」講義ノート，
<https://m-katsurada.sakura.ne.jp/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変更した。(2014～).