

2026年度秋学期

画像情報処理

第1部・画像のサンプリングと周波数 /

第5回

離散フーリエ変換,
フーリエ変換の実例



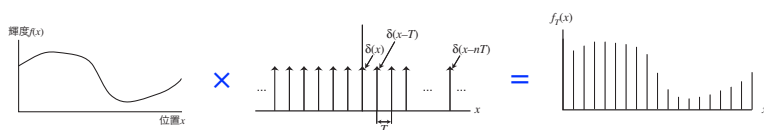
関西大学総合情報学部
浅野 晃

離散フーリエ変換 🤔

2 | 20

サンプリングされた関数のフーリエ変換

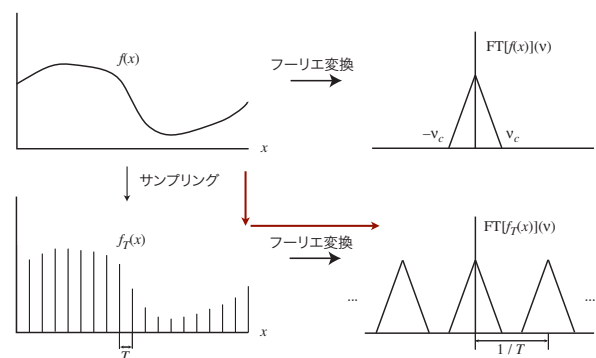
サンプリング $f_T(x) = f(x)\text{comb}_T(x)$



サンプリングされた関数のフーリエ変換

$$\begin{aligned} FT[f_T(x)](\nu) &= FT[f(x)\text{comb}_T(x)](\nu) \\ &= \int_{-\infty}^{\infty} f(x)\text{comb}_T(x) \exp(-i2\pi\nu x) dx \end{aligned}$$

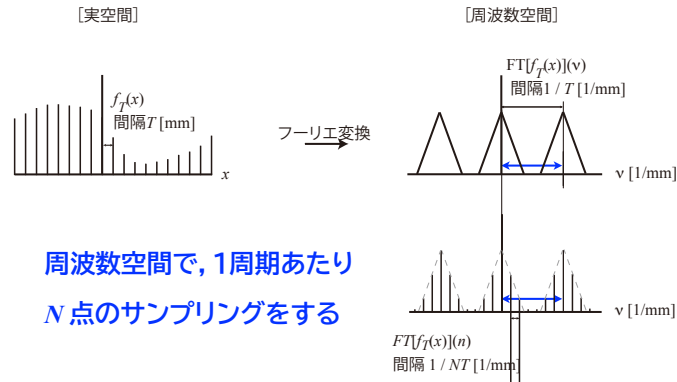
サンプリングされた関数のフーリエ変換



こちらは離散的だが

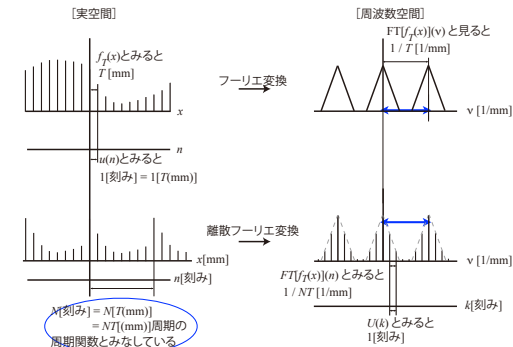
こちらは離散的でない→コンピュータで扱えない

周波数空間でもサンプリング



実空間ではどうなる？

実空間でサンプリング→周波数空間で周期的に現れる
周波数空間でサンプリング→実空間で周期的に現れる



「周波数空間でサンプリング」とは

周波数空間でサンプリング→実空間で周期的に現れる 🤖

周波数空間でサンプリング, つまり「周波数がとびとび」

それはつまり

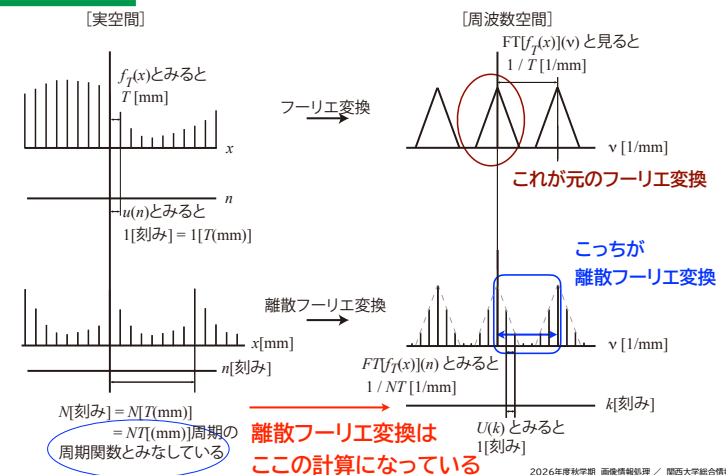
$$f(x) = \text{波長 } L + \text{波長 } L/2 + \text{波長 } L/3 + \dots + \text{波長 } L/n + \dots$$

フーリエ級数

ということは, 周期関数を三角関数の足し合わせで表している

離散フーリエ変換

元のフーリエ変換とはだいぶ違う

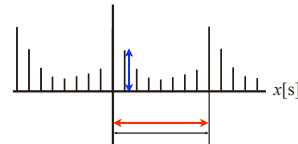


数列の計算にする

元の関数は忘れて、サンプリングされたものを数列とみなす

デルタ関数の並びではなく、単に間隔 T でとびとびに取り出された関数の値を数列 $u(n)$ とする

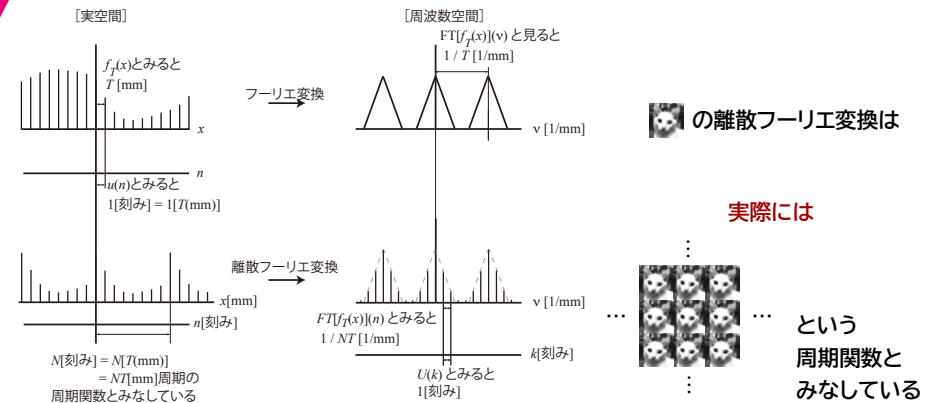
デルタ関数の並びの積分だったのが
→数列の場合は、その値を合計するだけ



離散フーリエ変換(DFT)

$$U(k) = \sum_{n=0}^{N-1} u(n) \exp\left(-i2\pi \frac{k}{N} n\right) \quad (k = 0, 1, \dots, N-1)$$

離散フーリエ変換



フーリエ変換の実例💡
(MATLABを使って示します)

テキスト付録1:
周波数空間でのサンプリングと、実空間で
の周期関数の関係🤔

「周波数空間でサンプリング」とは

周波数空間でサンプリング→実空間で周期的に現れる 🤖

周波数空間でサンプリング, つまり「周波数がとびとび」

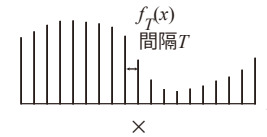
それはつまり

$$f(x) = \text{波長 } L + \text{波長 } L/2 + \text{波長 } L/3 + \dots + \text{波長 } L/n + \dots$$

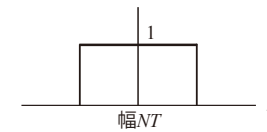
フーリエ級数

ということは, 周期関数を三角関数の足し合わせで表している

「実空間の周期関数」のほうから考えてみる

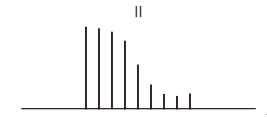


実空間でサンプリングされた関数 $f_T(x)$



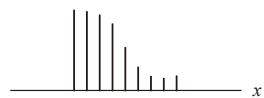
幅 NT だけ切り出す矩形関数 $\text{rect}(\frac{x}{NT})$

$$\text{rect}(x) = \begin{cases} 0 & (|x| > \frac{1}{2}) \\ 1 & (|x| < \frac{1}{2}) \end{cases}$$



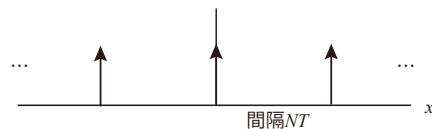
切り出した $f_T(x) \times \text{rect}(\frac{x}{NT})$

サンプリングして, さらに周期関数にする

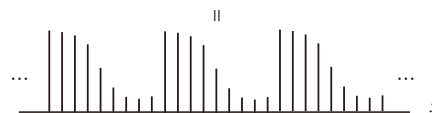


切り出された $f_T(x) \times \text{rect}(\frac{x}{NT})$

* コンヴォリューション



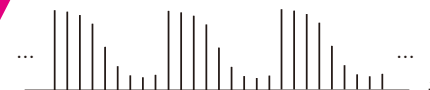
間隔 NT のくし形関数 $\text{comb}_{NT}(x)$



周期 NT の周期関数にした

$$f_T(x) \times \text{rect}(\frac{x}{NT}) * \text{comb}_{NT}(x)$$

そのフーリエ変換は



周期 NT の周期関数になった

$$f_T(x) \times \text{rect}(\frac{x}{NT}) * \text{comb}_{NT}(x)$$

そのフーリエ変換は

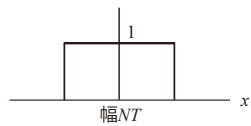
$$\begin{aligned} & FT[f_T(x) \text{comb}_T(x) \times \text{rect}(\frac{x}{NT}) * \text{comb}_{NT}(x)] \\ &= FT[f_T(x) \text{comb}_T(x)] * FT[\text{rect}(\frac{x}{NT})] \times FT[\text{comb}_{NT}(x)] \end{aligned}$$

フーリエ変換すると

$$\begin{aligned} \times &\rightarrow * \\ * &\rightarrow \times \end{aligned}$$

かけ算はコンヴォリューションに
コンヴォリューションはかけ算に

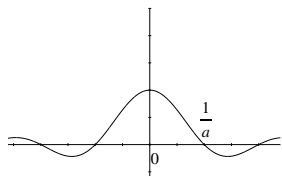
矩形関数のフーリエ変換は



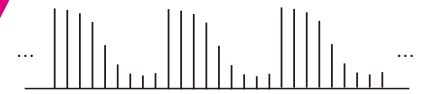
幅 a の矩形関数 $\text{rect}(\frac{x}{a})$ の
フーリエ変換は

$$\begin{aligned} FT[\text{rect}(\frac{x}{a})] &= \int_{-\infty}^{\infty} \text{rect}(\frac{x}{a}) \exp(-i2\pi\nu x) dx \\ &= \int_{-\frac{a}{2}}^{\frac{a}{2}} \exp(-i2\pi\nu x) dx \\ &= \frac{1}{-i2\pi\nu} [\exp(-i2\pi\nu x)]_{-\frac{a}{2}}^{\frac{a}{2}} \\ &= \frac{1}{i2\pi\nu} (\exp(i\pi a\nu) - \exp(-i\pi a\nu)) \\ &= \frac{\sin(a\pi\nu)}{\pi\nu} \end{aligned}$$

sinc関数といい, $\text{sinc}(a\nu)$ で表す



サンプリング／周期化してフーリエ変換すると



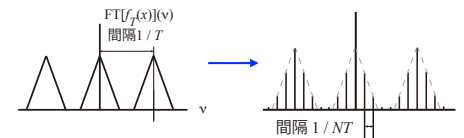
周期 NT の周期関数になった

$$f_T(x) \times \text{rect}(\frac{x}{NT}) * \text{comb}_{NT}(x)$$

このフーリエ変換は $FT[f_T(x)] \times \text{comb}_{\frac{1}{NT}}(x\nu) * \text{sinc}(\frac{\nu}{1/(NT)})$

実空間でサンプリングされた $f_T(x)$ のフーリエ変換を
間隔 NT でサンプリング

こっちは？

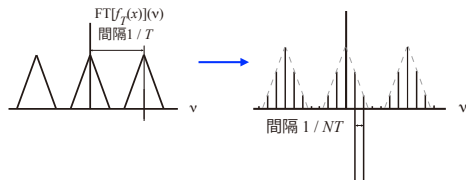


sinc関数はどうなるのか？

$$FT[f_T(x)] \times \text{comb}_{\frac{1}{NT}}(x\nu) * \text{sinc}(\frac{\nu}{1/(NT)})$$

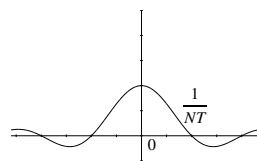
実空間でサンプリングされた $f_T(x)$ のフーリエ変換を
間隔 NT でサンプリング

デルタ関数の間隔 $1/(NT)$ の並びと
sinc関数のコンヴォリューション



sinc関数 $\text{sinc}(\frac{\nu}{1/(NT)})$ が
間隔 $1/(NT)$ で並ぶ

$\text{sinc}(\frac{\nu}{1/(NT)})$ は 間隔 $1/(NT)$ ごとにゼロになるから,
間隔 $1/(NT)$ で並んだsinc関数の影響はない



テキスト付録2:
離散フーリエ変換すると
データサイズが2倍に増えているのか？🤔

離散フーリエ変換の結果は複素数

N個の実数値は, N個の複素数値に変換される

複素数は $a + bi$ の形で, 2つの実数の組になっている

離散フーリエ変換によって,
データの大きさが2倍になっているのか? 🤔

そんなことはないはずです。

離散フーリエ変換と対称性

数列 $u(n)$ を離散フーリエ変換したものが 数列 $U(k)$ であるとき

$$U^*(N - k) = U(k) \quad \text{という対称性がある}$$

*は複素共役

$$U = a + bi \text{ のとき, } U^* = a - bi$$

なぜならば, 🤔

離散フーリエ変換と対称性

数列 $u(n)$ を離散フーリエ変換したものが 数列 $U(k)$ であるとき

$$U^*(N - k) = U(k) \quad \text{なぜならば, 🤔}$$

$u(n)$ が実数ならば, $u^*(n) = u(n)$ なので

$$\begin{aligned} U^*(N - k) &= \sum_{n=0}^{N-1} u^*(n) \exp(i2\pi \frac{N-k}{N} n) \\ &= \sum_{n=0}^{N-1} u(n) \exp(i2\pi n) \exp(i2\pi \frac{-k}{N} n) \end{aligned}$$

離散フーリエ変換と対称性

$$\begin{aligned} U^*(N - k) &= \sum_{n=0}^{N-1} u^*(n) \exp(i2\pi \frac{N-k}{N} n) \\ &= \sum_{n=0}^{N-1} u(n) \exp(i2\pi n) \exp(i2\pi \frac{-k}{N} n) \end{aligned}$$

n が整数のとき 指数関数と三角関数の関係から

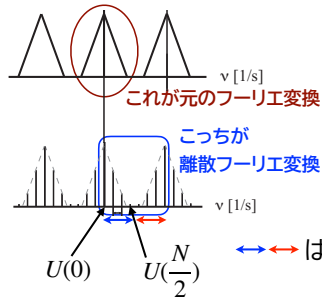
$$\exp(i2\pi n) = \cos(2\pi n) + i \sin(2\pi n) = 1 + 0 = 1, \text{ よって}$$

$$U^*(N - k) = \sum_{n=0}^{N-1} u(n) \exp(i2\pi \frac{-k}{N} n) = U(k)$$

やっぱりデータサイズは2倍にはなってない

n が整数のとき 指数関数と三角関数の関係から

$\exp(i2\pi n) = \cos(2\pi n) + i \sin(2\pi n) = 1 + 0 = 1$, よって



$u(n)$ が実数ならば, $U^*(k) = U(N - k)$ なので
離散フーリエ変換で表されている
最大の周波数は $N/2$

「指数関数2つの組でひとつの波」だから,
当然といえば当然ですね

第2部へ

第2部は画像データ圧縮

画像の細かいところを, 見た目にはわからないようにごまかして, データ量を減らす

「細かいところ」はどのように表現されるか?

→ 周波数で表現される

そういうわけで, もう少しフーリエ変換とおつきあいください。

もっと一般的な原理から説明します。まずは数学の「行列」から。