

離散フーリエ変換

数学というのは、連続な関数を扱うほうが簡単な場合が多いのですが、コンピュータで扱える計算は離散的なものだけです。そこで、サンプリング（標本化）によって離散化された関数のフーリエ変換を、離散的なまま計算する方法について考えてみましょう。

サンプリングされた関数のフーリエ変換

もとの連続な関数 $f(t)$ を間隔 T でサンプリングした関数 $f_T(t)$ は、前回説明した、デルタ関数が等間隔に並んだくし形関数 $\text{comb}_T(x)$ を使って

$$f_T(x) = f(x)\text{comb}_T(x) \quad (1)$$

と表されます。

サンプリングされた関数 $f_T(x)$ のフーリエ変換は、 FT でフーリエ変換を表すとして

$$\begin{aligned} FT[f_T(x)](\nu) &= FT[f(x)\text{comb}_T(x)](\nu) \\ &= \int_{-\infty}^{\infty} f(x)\text{comb}_T(x) \exp(-i2\pi\nu x) dx \end{aligned} \quad (2)$$

となります。このフーリエ変換は、前回説明したように、 $FT[f(x)](\nu) * \text{comb}_{1/T}(\nu)$ という、たたみ込み積分（コンヴォリューション）で表されます。周波数空間においては、もとの関数 $f(x)$ のフーリエ変換が $1/T$ 間隔で繰り返し現れます。

周波数もサンプリング→実空間で周期関数に

この状態では、実空間の関数 $g(t)$ はサンプリングによって離散化しましたが、周波数空間では離散的にはなっておらず、まだ「コンピュータで計算するために離散化する」という目的は達せられていません。

そこで、周波数空間のほうもサンプリングすることにします。周波数空間での1周期 $1/T$ の間に N 回のサンプリングをすることになると、サンプリング間隔は $1/NT$ になります。

このような周波数空間でのサンプリングは、実空間では何をしていることになるのでしょうか？ 周波数空間での間隔 $1/NT$ のサンプリングは、周波数空間で間隔 $1/NT$ のくし形関数 $\text{comb}_{1/(NT)}$ をかけ算することに相当します。フーリエ変換するとそのように「とびとび」になるのは、実空間では周期 NT の周期関数のはずです。

実空間の関数 $f_T(x)$ は間隔 T でサンプリングされていました。ということは、周波数空間でサンプリングされた関数は、「実空間の $f_T(x)$ のうち N 個の一続きのサンプルだけを取り出し、この N 個のサンプルを無限に繰り返してコピーして、周期 NT の周期関数を作ったもの」をフーリエ変換したものになっているわけです（詳しくは付録1をみてください）。

連続な関数はもう忘れよう

ここまでの説明では、 $f(\nu)$ が先にあって、 $f_T(x)$ は $f(x)$ をサンプリングしたものと考えてきました。ここで、発想を転換して、 $f(x)$ がどんなものかは不明で、 $f_T(x)$ だけが与えられていると考えます。そ

うすると、 $f_T(x)$ は単なる数値の列となりますから、これを $u(n)$ と表すことにします。 $u(n)$ は、 $f(x)$ やそれをサンプリングした $f_T(x)$ のように、連続的に変化する x にしたがって「測る」ものではなく、 $n = 1, 2, \dots$ と「数える」ものとなります。このような表現は、「 $u(n)$ で n が 1 だけちがうことは、 $f_T(x)$ のほうでは長さ（あるいは時間） T だけ離れている」ということに対応していますから、サンプリング間隔 T を 1 ([単位長さ（あるいは単位時間）]) とする新しい単位で時間を表現していると考えられることもできます。このようにして、

$$u(n) = f_T(nT) \quad (3)$$

と表します。

さて、ここまでの説明で述べたように、 $f(x)$ は、間隔 T でサンプリングされ、周期 NT の周期関数になっているとみなしていますから、(2) 式のフーリエ変換の実数全体での積分のかわりに、 $u(n)$ の 1 周期分 (N 項) だけを計算することにします。また、(2) 式は、デルタ関数の並びとある関数との積を実数全体で積分する形になっています。一方、デルタ関数は「インパルスのある場所以外では 0、積分すると 1」ですから、ある関数とデルタ関数との積を実数全体で積分すると、その関数からインパルスのある位置での値だけを切り出したものになります。

したがって、離散的に計算する場合は、積分の代わりに、インパルスの位置でのその関数の値を合計するという、単純な計算になります。したがって、(2) 式のフーリエ変換のかわりに

$$U(k) = \sum_{n=0}^{N-1} u(n) \exp\left(-i2\pi \frac{k}{N} n\right) \quad (k = 0, 1, \dots, N-1) \quad (4)$$

を計算します。

離散フーリエ変換

(4) 式を**離散フーリエ変換** (discrete Fourier transformation, DFT) といいます。計算機で行うフーリエ変換は、すべてこの離散フーリエ変換であり、本来のフーリエ変換とは異なることに注意してください。

同様に、周波数空間でも、周期関数がサンプリングされた形になっており、1 周期分が N 項からなりますから、

$$u(j) = \frac{1}{N} \sum_{k=0}^{N-1} U(k) \exp\left(i2\pi \frac{j}{N} k\right) \quad (j = 0, 1, \dots, N-1) \quad (5)$$

を計算し、これを**離散逆フーリエ変換**といいます。(5) 式の右辺に (4) 式を代入すると、

$$\begin{aligned} & \frac{1}{N} \sum_{k=0}^{N-1} \left(\sum_{n=0}^{N-1} u(n) \exp\left(-i2\pi \frac{k}{N} n\right) \right) \exp\left(i2\pi \frac{j}{N} k\right) \\ &= \frac{1}{N} \sum_{n=0}^{N-1} u(n) \sum_{k=0}^{N-1} \exp\left(i2\pi \frac{j-n}{N} k\right) \end{aligned} \quad (6)$$

となります。後ろ側の総和は、 $n \neq j$ のとき、公比 $\exp\left(i2\pi \frac{j-n}{N}\right)$ 、項数 N の等比数列の和なので、

$$\frac{1 - \{\exp(i2\pi)\}^{(j-n)k}}{1 - \exp\left(i2\pi \frac{j-n}{N}\right)} = \frac{1 - 1^{(j-n)k}}{1 - \exp\left(i2\pi \frac{j-n}{N}\right)} = 0 \quad (7)$$

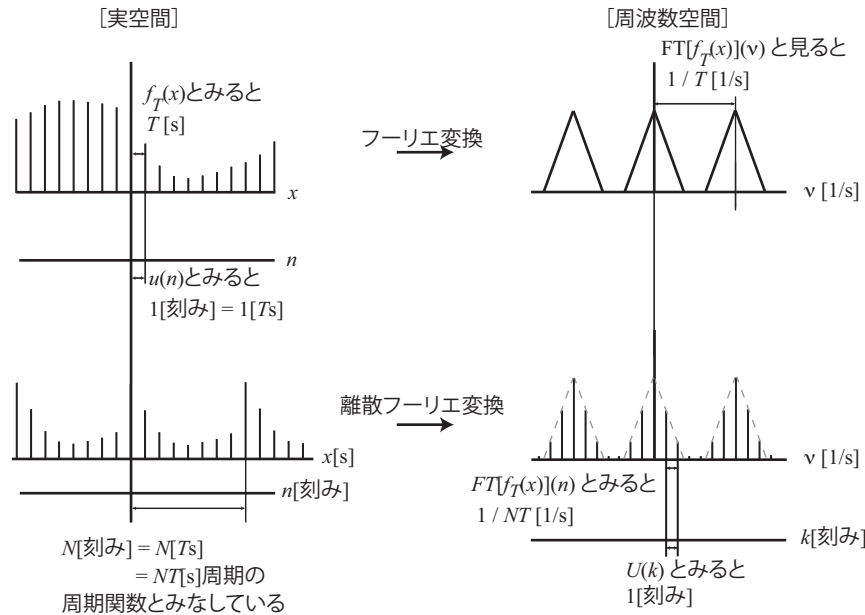


図 1: 離散フーリエ変換.

となり, $j = n$ のときは $\sum_{k=0}^{N-1} 1 = N$ となります。よって, (6) 式は $\frac{1}{N} N \cdot u(j) = u(j)$ となり, (4) 式の $u(n)$ と (4) 式の $U(k)$ が, 確かにフーリエ変換対になっていることがわかります¹。

例として, サンプルング間隔 $T = 1$ ([秒]) でサンプルングされ, $N = 256$ 点からなる信号 $u(n)$ がある とします。この信号を離散フーリエ変換して $U(k)$ を得たとき, k の 1 刻みは, もとの信号における周波数空間では何 ([1/秒]) の周波数に相当するかを考えてみましょう。ここまでの説明の通り, 周波数空間においては, サンプルングの間隔は $1/NT$ ([1/秒]) です。この例では $T = 1$ [秒], $N = 256$ (刻み) なので, 1 刻みは $1/256$ ([1/秒]) に相当します。

付録 1. 周波数空間でのサンプルングと, 実空間での周期関数の関係

実空間でサンプルングされた関数 $f_T(x)$ から, 幅 NT に入る N 個のサンプルを切り出したものは,

$$f_T(x) \times \text{rect}\left(\frac{x}{NT}\right) \quad (\text{A1})$$

と表されます。ここで, $\text{rect}(x)$ は**矩形関数**とよばれるもので,

$$\text{rect}(x) = \begin{cases} 0 & (|x| > \frac{1}{2}) \\ 1 & (|x| < \frac{1}{2}) \end{cases} \quad (\text{A2})$$

と定義されます。

(A1) 式の関数を, さらに周期 NT で繰り返したものは, くし形関数を使うと

$$f_T(x) \times \text{rect}\left(\frac{x}{NT}\right) * \text{comb}_{NT}(x) \quad (\text{A3})$$

¹ 離散逆フーリエ変換の係数 $1/N$ は, この関係がなりたつようにつけられています。第 2 部で, 画像圧縮に関して再びこのことに触れます。

となります。(A3) 式の関数のフーリエ変換は、前回の付録 1 で説明したかけ算とコンヴォリューションの関係により、

$$\begin{aligned} FT[f(x)\text{comb}_T(x) \times \text{rect}(\frac{x}{NT}) * \text{comb}_{NT}(x)] \\ = FT[f(x)\text{comb}_T(x)] * FT[\text{rect}(\frac{x}{NT})] \times FT[\text{comb}_{NT}(x)] \end{aligned} \quad (\text{A4})$$

となります。ここで、矩形関数のフーリエ変換は

$$\begin{aligned} FT[\text{rect}(\frac{x}{a})] &= \int_{-\infty}^{\infty} \text{rect}(\frac{x}{a}) \exp(-i2\pi\nu x) dx \\ &= \int_{-\frac{a}{2}}^{\frac{a}{2}} \exp(-i2\pi\nu x) dx \\ &= \frac{1}{-i2\pi\nu} [\exp(-i2\pi\nu x)]_{-\frac{a}{2}}^{\frac{a}{2}} \\ &= \frac{1}{i2\pi\nu} (\exp(i\pi a\nu) - \exp(-i\pi a\nu)) \\ &= \frac{\sin(a\pi\nu)}{\pi\nu} \end{aligned} \quad (\text{A5})$$

となります。この関数を **sinc 関数** とよび、 $\text{sinc}(a\nu)$ で表します。この結果、(A3) 式の関数のフーリエ変換は、

$$FT[f_T(x)] \times \text{comb}_{\frac{1}{NT}}(x\nu) * \text{sinc}(\frac{\nu}{1/(NT)}) \quad (\text{A6})$$

と表すことができます。

この式の前半のかけ算は、実空間でサンプリングされた $f_T(x)$ のフーリエ変換を、さらに間隔 NT でサンプリングすることを示しています。サンプリングされた結果は、デルタ関数の並びになっています。前回説明したように、ある関数とデルタ関数とのたたみ込みは、その関数自身になります。したがって、(A6) 式の後半のたたみ込みにより、sinc 関数が間隔 $1/(NT)$ で並ぶことになります。しかし、関数 $\text{sinc}(\frac{\nu}{1/(NT)})$ は、間隔 $1/(2NT)$ ごとにゼロになりますので、あるデルタ関数のところに位置する sinc 関数の影響は、他のデルタ関数の位置には及びません。したがって、(A3) 式の周期関数のフーリエ変換が、確かに $f_T(\nu)$ をフーリエ変換し、周波数空間でさらにサンプリングしたものになっています。

付録 2. 離散フーリエ変換するとデータが増えているのか？

離散フーリエ変換では、 N 個の実数値を N 個の複素数値に変換しています。ということは、データの量が 2 倍に増えているのでしょうか？ そうではありません。

N 個の実数値からなる数列の離散フーリエ変換では、 $U(k)$ の複素共役を $U^*(k)$ で表すとき

$$U^*(N-k) = U(k) \quad (\text{A7})$$

となります。なぜならば、 $u(n)$ が実数列ならば、

$$\begin{aligned} U^*(N-k) &= \sum_{n=0}^{N-1} u^*(n) \exp(i2\pi \frac{N-k}{N} n) \\ &= \sum_{n=0}^{N-1} u(n) \exp(i2\pi n) \exp(i2\pi \frac{-k}{N} n) \end{aligned} \quad (\text{A8})$$

となり、 n が整数のとき $\exp(i2\pi n) = 1$ ですから

$$U^*(N - k) = \sum_{n=0}^{N-1} u(n) \exp(i2\pi \frac{-k}{N} n) = U(k) \quad (\text{A9})$$

となるからです。このことは、 N 点の離散フーリエ変換は、周波数空間で最大 $N/2$ の周波数までしか表現していないことを意味しています。すなわち、周波数空間で本当に意味があるのは「 $U(0)$ から $U(N/2)$ の $N/2 + 1$ 個の数だけ」であるということになります。