2022 年度秋学期　画像情報処理　第５回

第１部・画像のサンプリングと周波数／ 離散フーリエ変換

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 $\mathrm{comb}_T(x)$ を使って

$$f_T(x) = f(x)\mathrm{comb}_T(x) \tag{1}$$

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

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

となります。このフーリエ変換は，前回説明したように，$FT[f(x)](\nu) * \mathrm{comb}_{1/T}(\nu)$ 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すると，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 $\mathrm{comb}_{1/(NT)}$ 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，$n = 1, 2, \ldots$ 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ることもできます。このようにして，

$$u(n) = f_T(nT) \tag{3}$$

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。

離散フーリエ変換

そこで，(2) 式のフーリエ変換のかわりに

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

を計算します。

(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，

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

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，

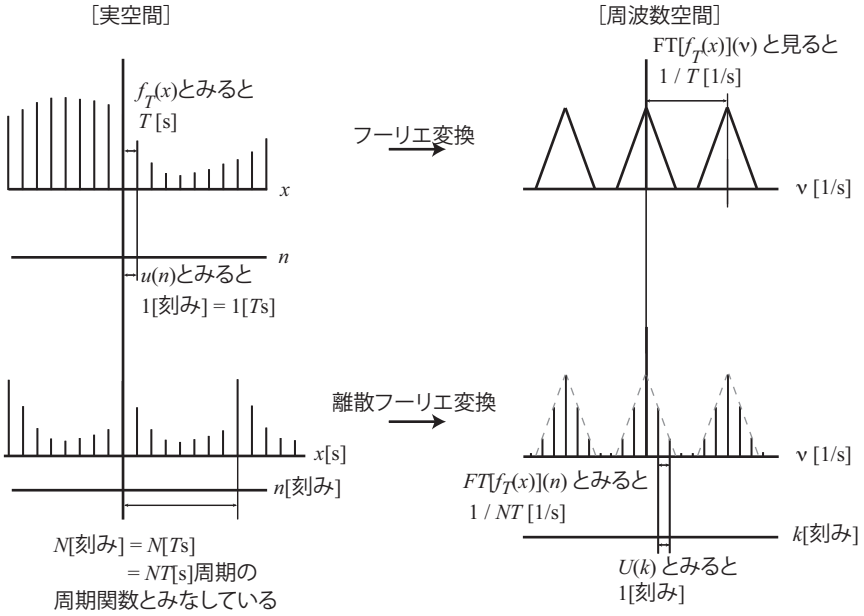

図 1: 離散フーリエ変換.

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

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,

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

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，サンプリング間隔 $T=1$([秒]) でサンプリングされ，$N=256$ 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，サンプリングの間隔は $1/NT$([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。

---

[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。

高速フーリエ変換について

高速フーリエ変換（Fast Fourier Transformation, FFT）とは，指数関数の性質を利用して，計算をうまくまとめることで，離散フーリエ変換の計算に含まれる掛け算の回数を減らす工夫です。コンピュータによる計算では，掛け算は足し算に比べて時間がかかるので，掛け算を減らすと全体の計算にかかる時間を短くすることができます。ここでは，もっとも広く使われている Cooley and Tukey の高速フーリエ変換を，簡単な例を使って説明します。

「計算をうまくまとめて掛け算を減らす」というのは，簡単にいうと，「$5\times 4+3\times 5$」をそのまま計算せずに，「$5\times(4+3)$」にまとめて，掛け算を 2 回から 1 回に減らすことです。これを離散フーリエ変換でどうやるかを，$N=4$，つまり 4 点だけの信号を例にして説明します。

(4) 式の離散フーリエ変換は，$N=4$ とすると

$$U(k)=\sum_{n=0}^{3}u(n)\exp\left(-i2\pi\frac{k}{4}n\right)\quad(k=0,1,\ldots,3) \tag{8}$$

となります。この計算では，$U(0),U(1),U(2),U(3)$ の 4 つの $U(\ )$ を計算しますが，ひとつの $U()$ を計算するために，$u(n)\exp\left(-i2\pi\frac{k}{4}n\right)$ という掛け算を，$n=0,1,2,3$ の 4 回行います。したがって，$U(0),U(1),U(2),U(3)$ をすべて計算するためには，$4\times 4=16$ 回の掛け算を行う必要があります。この掛け算を，計算の工夫によって減らすことを考えます。

ここで，

$$W\equiv\exp\left(-i\frac{2\pi}{4}\right) \tag{9}$$

とおくと，(8) 式の指数関数は $W^{k\times n}$ と表すことができます。このとき，指数関数と三角関数の関係を用いると，

$$\begin{aligned}
W^0&=\exp\left(-i2\pi\frac{0}{4}\right)=\cos 0+i\sin 0=1+0i=1\\
W^1&=\exp\left(-i2\pi\frac{1}{4}\right)=\cos(-\frac{\pi}{2})+i\sin(-\frac{\pi}{2})=0-i=-i\\
W^2&=\exp\left(-i2\pi\frac{2}{4}\right)=\cos(-\pi)+i\sin(-\pi)=-1+0i=-1\\
W^3&=\exp\left(-i2\pi\frac{3}{4}\right)=\cos(-\frac{3\pi}{2})+i\sin(-\frac{3\pi}{2})=0+i=i\\
W^4&=\exp\left(-i2\pi\frac{4}{4}\right)=\cos(-2\pi)+i\sin(-i2\pi)=1+0i=1
\end{aligned} \tag{10}$$

となり，

$$
\begin{aligned}
W^0 &= 1 \\
W^1 &= -i \\
W^2 &= -1 = -W^0 \\
W^3 &= i = -W^1 \\
W^4 &= 1 = W^0
\end{aligned}
\tag{11}
$$

となります。ですから，W^5 以降も，

$$
\begin{aligned}
W^5 &= W^4 \times W^1 = 1 \times W^1 = W^1 \\
W^6 &= W^4 \times W^2 = 1 \times (-W^0) = -W^0 \\
W^7 &= W^4 \times W^3 = 1 \times (-W^1) = -W^1 \\
W^8 &= W^4 \times W^4 = 1 \times W^0 = W^0 \\
W^9 &= W^8 \times W^1 = W^0 \times W^1 = 1 \times W^1 = W^1 \\
&\cdots
\end{aligned}
\tag{12}
$$

とくりかえすことになって，すべての $W^{k\times n}$ を W^0 と W^1 だけで表すことができることがわかります。

さて，(4) 式の離散フーリエ変換で $N = 4$ の場合を，$U(0), U(1), U(2), U(3)$ について W を使って書き下すと，

$$
\begin{aligned}
U(0) &= u(0)W^{0\times 0} + u(1)W^{0\times 1} + u(2)W^{0\times 2} + u(3)W^{0\times 3} \\
U(1) &= u(0)W^{1\times 0} + u(1)W^{1\times 1} + u(2)W^{1\times 2} + u(3)W^{1\times 3} \\
U(2) &= u(0)W^{2\times 0} + u(1)W^{2\times 1} + u(2)W^{2\times 2} + u(3)W^{2\times 3} \\
U(3) &= u(0)W^{3\times 0} + u(1)W^{3\times 1} + u(2)W^{3\times 2} + u(3)W^{3\times 3}
\end{aligned}
\tag{13}
$$

となり，これは

$$
\begin{aligned}
U(0) &= u(0)W^0 + u(1)W^0 + u(2)W^0 + u(3)W^0 \\
U(1) &= u(0)W^0 + u(1)W^1 + u(2)W^2 + u(3)W^3 \\
U(2) &= u(0)W^0 + u(1)W^2 + u(2)W^4 + u(3)W^6 \\
U(3) &= u(0)W^0 + u(1)W^3 + u(2)W^6 + u(3)W^9
\end{aligned}
\tag{14}
$$

と表されます。さらに，さきほど述べた W についてのくりかえしを使うと，

$$
\begin{aligned}
U(0) &= u(0)W^0 + u(1)W^0 + u(2)W^0 + u(3)W^0 \\
U(1) &= u(0)W^0 + u(1)W^1 - u(2)W^0 - u(3)W^1 \\
U(2) &= u(0)W^0 - u(1)W^0 + u(2)W^0 - u(3)W^0 \\
U(3) &= u(0)W^0 - u(1)W^1 - u(2)W^0 + u(3)W^1
\end{aligned}
\tag{15}
$$

となります。

この式の，行の順番を，次のように入れ替えてみましょう。

$$
\begin{aligned}
U(0) &= u(0)W^0 + u(1)W^0 + u(2)W^0 + u(3)W^0 \\
U(2) &= u(0)W^0 - u(1)W^0 + u(2)W^0 - u(3)W^0 \\
U(1) &= u(0)W^0 + u(1)W^1 - u(2)W^0 - u(3)W^1 \\
U(3) &= u(0)W^0 - u(1)W^1 - u(2)W^0 + u(3)W^1
\end{aligned}
\tag{16}
$$

こうしてみると，$U(0)$ と $U(2)$，$U(1)$ と $U(3)$ で，それぞれ W^0 と W^1 の現れる順番が同じで，符号だけが違っています。したがって，それぞれの組で同じように

$$
\begin{aligned}
U(0) &= (u(0) + u(2))W^0 + (u(1) + u(3))W^0 \\
U(2) &= (u(0) + u(2))W^0 - (u(1) + u(3))W^0 \\
U(1) &= (u(0) - u(2))W^0 + (u(1) - u(3))W^1 \\
U(3) &= (u(0) - u(2))W^0 - (u(1) - u(3))W^1
\end{aligned}
\tag{17}
$$

と，まとめることができます。この段階で，全体の掛け算の回数は，16 回から 8 回に減っています。

N が $8, 16, 32, \ldots$ と，2 の累乗になっている場合は，同じように W のくりかえしの性質を使い，上と同じように行を入れ替えるのをくりかえすことで，最終的に上のように 2 行ずつの組に直すことができ，掛け算の回数を減らすことができます。ただ，$N = 8$ の場合でも，ここまでのような式の書き方では，煩雑になってうまく扱えません。この問題は，フーリエ変換を「行列」を使って書くと，かなりすっきりします。次回の講義で行列について説明したあと，$N = 8$ の場合について行列を使ってこの計算を表現して説明し，「どのくらい掛け算が減るのか」を説明します。

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

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

$$
f_T(x) \times \mathrm{rect}(\frac{x}{NT}) \tag{A1}
$$

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

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

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

$$f_T(x) \times \mathrm{rect}(\frac{x}{NT}) * \mathrm{comb}_{NT}(x) \tag{A3}$$

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，

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

となります。ここで，矩形関数のフーリエ変換は

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

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

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

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 $\mathrm{sinc}(\frac{\nu}{1/(NT)})$ 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，データの大きさが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

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

となります。なぜならば，$u(n)$ が実数列ならば，

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

となり，n が整数のとき $\exp(i2\pi n) = \cos(2\pi n) + i\sin(2\pi n) = 1 + 0 = 1$ ですから

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

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。