

第4章 デジタル信号処理

計測したデータを計算機で処理する場合を考える。このときは、信号は有限時間でとられており、また計算機内では、連続信号ではなくデジタル信号としてあつかわれる。ここではデジタル信号処理に特有の問題を考える。

4.1 有限時間データであるための効果

4.1.1 有限時間パワースペクトル

信号 $x_T(t)$ が有限時間に限定されている ($-T/2 \leq t \leq T/2$) 場合は、そのフーリエ変換 X_T が存在し、Parseval の定理により

$$\frac{1}{T} \int_{-T/2}^{T/2} |x_T(t)|^2 dt = \int_{-\infty}^{\infty} \frac{2\pi}{T} |X_T(\omega)|^2 d\omega \quad (4.1)$$

が成り立つ。よって $\frac{2\pi}{T} \langle |X_T(\omega)|^2 \rangle$ によって有限時間パワースペクトル密度を定義できる。

4.1.2 ウィンドウ（窓関数）

データが $[-T/2, T/2]$ の有限区間にあることは、無限に長いデータに図 4.1 のような箱型のウィンドウ（窓関数）

$$w(t) = \begin{cases} 1 & (|t| \leq T/2) \\ 0 & (|t| > T/2) \end{cases} \quad (4.2)$$

をかけたことに相当する。

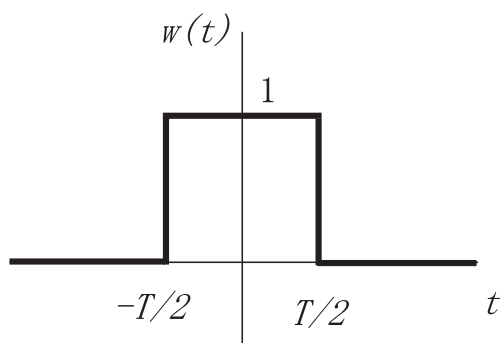


図 4.1: 箱型ウィンドウ

$x(t)$ を無限時間データとして

$$\begin{aligned} x_T(t) &= x(t)w(t) & (4.3) \\ x(t) &\xrightarrow{\text{FT}} X(\omega) \\ w(t) &\xrightarrow{\text{FT}} W(\omega) \end{aligned}$$

とすると、convolution を用いて

$$X_T(\omega) = \int_{-\infty}^{\infty} X(\omega')W(\omega - \omega')d\omega' \quad (4.4)$$

となる。よってデータが有限時間であることは、本来のスペクトル $X(\omega)$ を $W(\omega)$ で歪ませる効果をもつ。理想的なウィンドウは $W(\omega) = \delta(\omega)$ である。今の場合（箱型ウィンドウ）

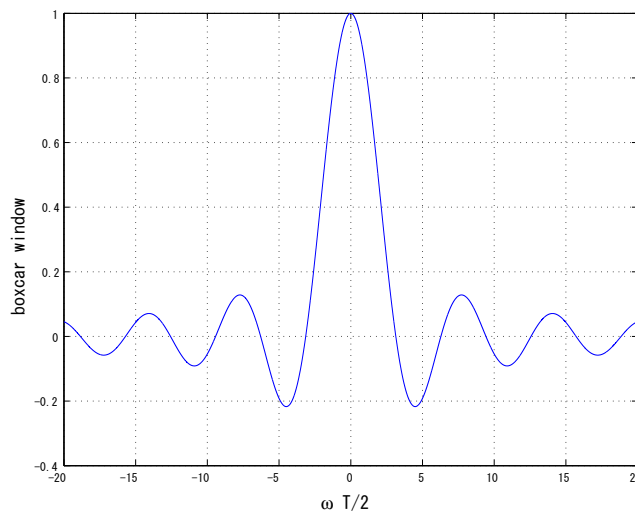
$$W(\omega) = T \frac{\sin \omega T/2}{\omega T/2} \quad (4.5)$$

である。 $W(\omega)$ を図 4.1.3 に示したが（ $W(0) = 1$ になるように規格化してある）、中心のピーク幅が狭いので分解能は高いが、裾（サイドローブ）が大きいため周りからの周波数成分の回りこみが多いことが欠点である。

4.1.3 MATLAB：箱型ウィンドウのフーリエスペクトル

```
>> t=[-20: .003: 20];
>> plot(t, sin(t)./t)

>> grid on;
>> xlabel('\omega T/2', 'fontsize', 14)
>> ylabel('boxcar window', 'fontsize', 14)
```



図：4.1.3

箱型以外、目的に応じていろいろウィンドウが考えられている。良く使われるのは、例えば、次のような hanning ウィンドウである。

$$w(t) = \begin{cases} \cos^2(\pi t/T) & (|t| \leq T/2) \\ 0 & (|t| > T/2) \end{cases} \quad (4.6)$$

4.1.4 MATLAB : hanning ウィンドウの形状

```
>> t=[-0.5: .001: 0.5];  
>> y=cos(pi*t).^2;  
>> plot(t, y)  
  
>> grid on;  
>> xlabel('t/T', 'fontsize', 14)  
>> ylabel('hanning window', 'fontsize', 14)
```

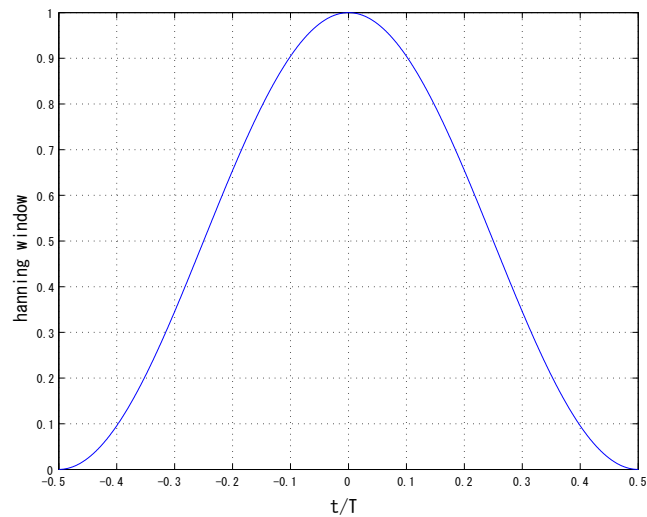


図 : 4.1.4

4.1.5 例題 : hanning ウィンドウのフーリエ変換

hanning ウィンドウのフーリエ変換が次のようになることを示せ。

$$W(\omega) = \frac{T \sin \omega T/2}{4\pi \omega T/2} \frac{1}{1 - (\omega T/2\pi)^2} \quad (4.7)$$

hanning ウィンドウのフーリエスペクトルを図 4.1.6 に示した。比較のために、箱型ウィンドウのスペクトルを破線で書いてある。hanning ウィンドウでは裾が低くなっているため、ここからのもれが少ない。2つのグラフは共に、 $W(0) = 1$ になるように規格化してある。

4.1.6 MATLAB : hanning ウィンドウのフーリエスペクトル

```
>> t=[-20: .003: 20];
>> plot(t, sin(t)./t./(1-(t/pi).^2),'r', t, sin(t)./t, ':')

>> grid on;
>> xlabel('\omega T/2', 'fontsize', 14)
>> ylabel('hanning window', 'fontsize', 14)
```

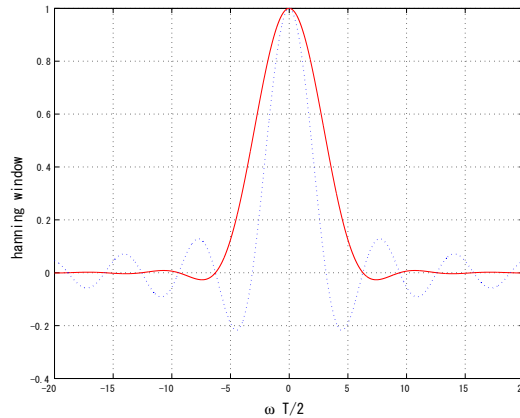


図 : 4.1.6

4.1.7 ウィンドウを用いた場合の補正

有限時間 T をもつウィンドウ $W(\omega)$ をかけたときのパワースペクトルを $S_T(\omega)$ とする。

$$S_T(\omega) = \frac{2\pi}{T} \langle |X_T(\omega)|^2 \rangle$$

$S(\omega)$ を本来の無限時間のパワースペクトルとすると

$$S_T(\omega) = \frac{2\pi}{T} \int_{-\infty}^{\infty} |W(\omega - \omega')|^2 S(\omega') d\omega' \quad (4.8)$$

の関係がある。

$S_T(\omega)$ でスペクトルを推定するときは、式 (4.8) より、ウィンドウに固有の補正が必要であることがわかる。そこで、 $S(\omega) = S_0$ (定数 : white noise) の場合について、この補正を考える。

$$\begin{aligned} S_T(\omega) &= \frac{2\pi}{T} \int_{-\infty}^{\infty} |W(\omega - \omega')|^2 S(\omega') d\omega' \\ &= \frac{S_0}{T} \int_{-T/2}^{T/2} |w(t)|^2 dt \end{aligned} \quad (4.9)$$

$$= S_0 \overline{w^2(t)} \quad (4.10)$$

よって、

$$\frac{1}{\overline{w^2(t)}} = \left[\frac{1}{T} \int_{-T/2}^{T/2} |w(t)|^2 dt \right]^{-1} \quad (4.11)$$

を $S_T(\omega)$ にかけて補正すればよい。

hanning ウィンドウの補正係数は、以下に求めるように $1/0.3746$ である。

4.1.8 MATLAB : hanning ウィンドウの補正係数

```
>> t=[-0.5: .001: 0.5];  
>> y=cos(pi*t).^2;  
>> y*y'/size(y, 2)
```

```
ans =  
    0.3746
```

4.1.9 自由課題 : hamming ウィンドウ

hamming ウィンドウについて調べよ。

4.1.10 MATLAB : hanning ウィンドウの効果

ウィンドウの効果を実際に見るために、MATLAB の periodogram という関数を利用してみる。periodogram は 1 回限りのパワースペクトルを求めるのに使われ、ウィンドウを指定することができる。ここでは正弦波と白色雑音がある場合、ウィンドウを使わないとき (箱型ウィンドウに対応)、hanning ウィンドウを使ったときのスペクトルの違いを調べる。

```
>> fs = 300;  
>> t = [0: 1/fs: 1];  
>> y = randn(1, length(t))+sin(2*pi*t*75);  
  
>> clf;  
>> subplot(211), periodogram(y, [],length(y),fs)  
>> subplot(212), periodogram(y, hanning(length(y)),length(y),fs)
```

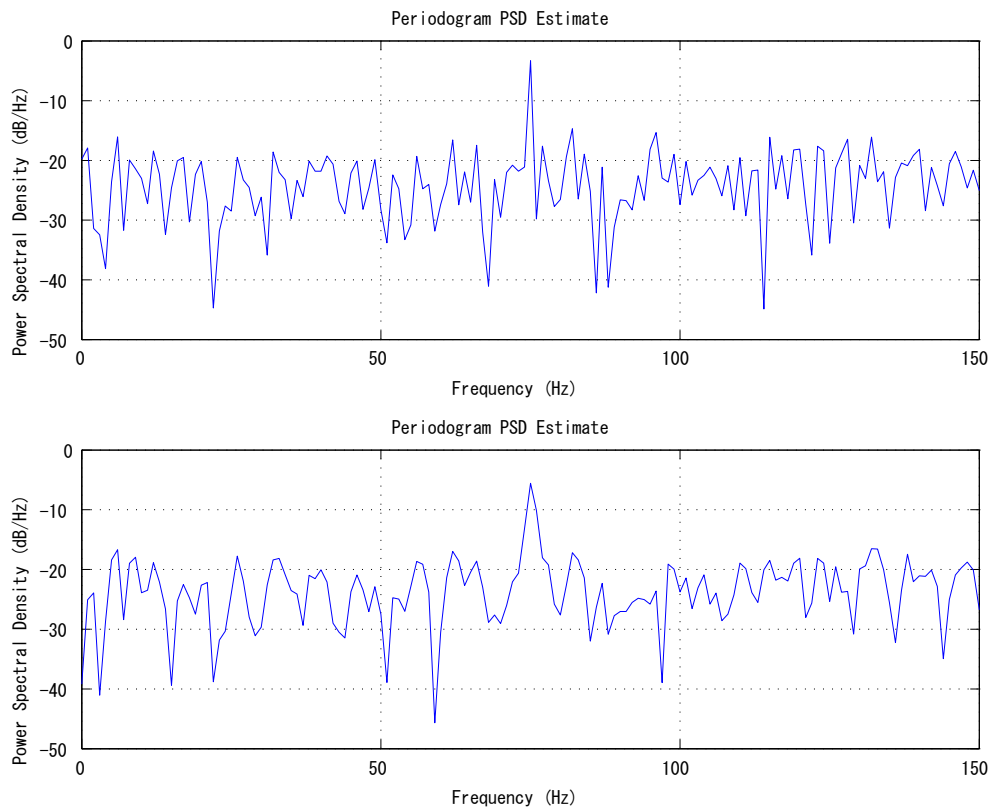


図 : 4.1.10

4.2 離散時間データである効果

4.2.1 サンプリング定理

連続信号 $x(t)$ が ν_c 以上の周波数成分を全く含まないなら、サンプリング周波数: $\nu_s \geq 2\nu_c$ として時間間隔 $\Delta t = \frac{1}{\nu_s}$ ごとにサンプリングすれば、得られたサンプリング信号 $x(n\Delta t)$ ($n = 0, \pm 1, \pm 2, \dots$) から元の信号を完全に再現できる。

以上のサンプリング定理を証明する。
サンプリングパルスを $d(t)$ とする。

$$d(t) = \sum_{n=-\infty}^{\infty} \delta(t - n\Delta t) \quad (4.12)$$

ここでサンプリング周波数 ν_s 、サンプリング角周波数 ω_s とする。

$$\omega_s = 2\pi\nu_s = \frac{2\pi}{\Delta t} \quad (4.13)$$

サンプリングされた信号

$$x_s(t) = x(t)d(t) \quad (4.14)$$

の両辺をフーリエ変換する。

$$X_s(\omega) = \int_{-\infty}^{\infty} X(\omega')D(\omega - \omega')d\omega' \quad (4.15)$$

$$\begin{aligned} D(\omega) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(t - n\Delta t)e^{-i\omega t} dt \\ &= \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} e^{-i\omega n\Delta t} \end{aligned} \quad (4.16)$$

以下の公式を用い、

$$\frac{1}{2a} \sum_{n=-\infty}^{\infty} e^{in\pi x/a} = \sum_{m=-\infty}^{\infty} \delta(x - 2ma) \quad (4.17)$$

$x = \omega, a = \pi/\Delta t$ とおく。

$$\begin{aligned} D(\omega) &= \frac{1}{2\pi} \frac{2\pi}{\Delta t} \sum_{n=-\infty}^{\infty} \delta(\omega - \frac{2\pi}{\Delta t}n) \\ &= \frac{1}{\Delta t} \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_s) \end{aligned} \quad (4.18)$$

よって

$$X_s(\omega) = \frac{1}{\Delta t} \sum_{n=-\infty}^{\infty} X(\omega - n\omega_s) \quad (4.19)$$

つまり、 $X_s(\omega)$ は $X(\omega)$ を ω_s ごとに繰り返す周期関数となる。 $\nu_s \geq 2\nu_c$ の場合、カットオフ ν_c の理想的なローパスフィルターを通せば元の信号が復元できる。もし $\nu_s < 2\nu_c$ の場合は、関数の繰り返しが重なってしまい、 $X_s(\omega)$ は歪んでしまう。これをエリアジング (aliasing) とよぶ。 ν_s のサンプリングに対して、 $\nu_s/2$ をナイキスト周波数という。つまりサンプリングされた信号のスペクトルはナイキスト周波数までしか意味がない。

$\nu_s \geq 2\nu_c$ のときは、デジタル化した値から元の波形が再現される。以下にこの表式を求める。デジタル信号を、カットオフ ν_c の理想的なローパスフィルターに通したとすると式 (4.19) より、

$$x(t) = \Delta t \int_{-\omega_c}^{\omega_c} X_s(\omega)e^{i\omega t}d\omega \quad (4.20)$$

となる。

$$X_s(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} x(t) d(t) e^{-i\omega t} dt \quad (4.21)$$

$$= \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} x(n\Delta t) e^{-i\omega n\Delta t} \quad (4.22)$$

よって

$$x(t) = \frac{\Delta t}{2\pi} \sum_{n=-\infty}^{\infty} x(n\Delta t) \int_{-\omega_c}^{\omega_c} e^{i\omega(t-n\Delta t)} d\omega \quad (4.23)$$

$$= \frac{\Delta t}{\pi} \sum_{n=-\infty}^{\infty} x(n\Delta t) \frac{\sin \omega_c(t-n\Delta t)}{t-n\Delta t} \quad (4.24)$$

を計算することによって元の信号が得られる。

4.3 DFT

以下で離散フーリエ変換 (DFT: Discrete Fourier Transform) を定義する。
 $x[n], n = 1, 2, \dots, N$ を離散時系列データとして

$$\begin{aligned} x[n] &\xrightarrow{\text{DFT}} X_D[k] \\ x[n] &\xleftarrow{\text{IDFT}} X_D[k] \end{aligned} \quad (4.25)$$

とする。

$$X_D[k] = \sum_{n=1}^N x[n] W_N^{(k-1)(n-1)}, \quad k = 1, 2, \dots, N \quad (4.26)$$

$$x[n] = \frac{1}{N} \sum_{k=1}^N X_D[k] W_N^{-(k-1)(n-1)}, \quad n = 1, 2, \dots, N \quad (4.27)$$

$$W_N = e^{-i(2\pi/N)} \quad (4.28)$$

4.4 FFT

FFT(Fast Fourier Transform) は DFT を効率よく計算するアルゴリズムである。例えば、 $N = 1024$ の場合、DFT には $N^2 \sim 10^6$ 回の計算が必要であるが、FFT で行くと $N \log_2 N \sim 10^4$ 回で済む。特にデータ数 $N = 2^m$ のとき高速な計算が可能となる。
 MATLAB を用いて実際に FFT を計算してみる。

4.4.1 MATLAB : FFT の計算

```
>> x = [4, 3, 7, -9, 1, 0, 0, 0]' ;  
>> y = fft(x)
```

```
y =  
6.0000  
11.4853 - 2.7574i  
-2.0000 -12.0000i  
-5.4853 +11.2426i  
18.0000  
-5.4853 -11.2426i  
-2.0000 +12.0000i  
11.4853 + 2.7574i
```

```
>> x = [4, 3, 7, -9, 1, 0, 0 ]' ;  
>> y = fft(x)
```

```
y =  
6.0000  
11.5206 - 4.8312i  
-7.9623 - 7.7059i  
7.4417 +13.9204i  
7.4417 -13.9204i  
-7.9623 + 7.7059i  
11.5206 + 4.8312i
```

これで見ると、FFT は N 個の実数データを与えると N 個の複素数データを式 (4.26) にしたがって計算出力するだけである。気をつけることは、サンプリング定理のところでもやったように、FFT 出力のうち独立なデータは半分しかないことである。また、この数は N が偶数か奇数かによって変わる。

表 4.1: FFT の周波数ポイント数

N	独立な周波数ポイント数	規格化された独立な角周波数 $\hat{\omega}$ の範囲
偶数	$N/2 + 1$	$[0, \pi]$
奇数	$(N + 1)/2$	$[0, \pi)$

4.5 離散時系列とその規格化

データが

$$x[n], \quad n = 1, 2, \dots, N$$

のように等間隔の N 個のデータに離散化されているとする。
次の 3 つの諸量を定義する。

$$\begin{aligned} \text{sampling time} &: \Delta t \\ \text{total time} &: T = (N - 1)\Delta t \\ \text{sampling frequency} &: \nu_s = \frac{1}{\Delta t} \end{aligned}$$

4.5.1 規格化された時間

$$\text{時系列 } t_n = (n - 1)\Delta t, \quad n = 1, 2, \dots, N$$

$$\text{実際の時間} : t_n \iff \text{規格化された時間} : n - 1$$

規格化された時間に Δt をかけると実時間が得られる。

4.5.2 規格化された周波数

$$\omega_k = \frac{2\pi(k-1)}{\Delta t N}, \quad \omega_1 = 0, \quad \omega_2 = \frac{2\pi}{\Delta t N} (= \Delta\omega), \dots, \omega_N = \frac{2\pi(N-1)}{\Delta t N}$$

$$\nu_k = \frac{1}{\Delta t} \frac{(k-1)}{N}, \quad \nu_1 = 0, \quad \nu_2 = \frac{1}{\Delta t} \frac{1}{N} (= \Delta\nu), \dots, \nu_N = \frac{1}{\Delta t} \frac{N-1}{N}$$

$$\text{実際の角周波数} : \omega_k \iff \text{規格化された角周波数} : \hat{\omega}_k = \frac{2\pi(k-1)}{N}$$

$$\text{実際の周波数} : \nu_k \iff \text{規格化された周波数} : \hat{\nu}_k = \frac{k-1}{N}$$

規格化された周波数を Δt で割ると実周波数、逆に実周波数で $\Delta t = 1$ と置くと規格化された表現が得られる。規格化された周波数 $\hat{\nu}_k$ でいうと $[0, 1)$ が全範囲である。 $\Delta\nu$ は周波数分解能を与える。

4.6 デジタル信号のパワースペクトル

$\sum_{n=1}^N |x[n]|^2$ が有限であるから、Parseval の定理が成立。

$$\sum_{n=1}^N |x[n]|^2 = \sum_{k=1}^N \frac{1}{N} |X_D[k]|^2 \quad (4.29)$$

上式を変形して

$$\frac{1}{N} \sum_{n=1}^N |x[n]|^2 = \sum_{k=1}^N \frac{1}{N^2} |X_D[k]|^2 \quad (4.30)$$

有限時間 T のパワースペクトルと比較して $\frac{1}{N^2} |X_D[k]|^2$ をパワースペクトルと考えるべき。

$$\Delta\nu = \frac{1}{\Delta t} \frac{1}{N} \quad (4.31)$$

であるから、式 (4.30) の右辺は以下のように書ける。

$$\sum_{k=1}^N \frac{1}{N^2} |X_D[k]|^2 \frac{1}{\Delta\nu} \Delta\nu = \sum_{k=1}^N \frac{\Delta t}{N} |X_D[k]|^2 \Delta\nu \quad (4.32)$$

よってパワースペクトル密度 (PSD: Power Spectral Density) は

$$\frac{\Delta t}{N} |X_D[k]|^2 \quad [1/\text{Hz}], \quad @v_k = \frac{1}{\Delta t} \frac{k-1}{N} \quad [\text{Hz}] \quad (4.33)$$

規格化された PSD は

$$\frac{1}{N} |X_D[k]|^2, \quad @v_k = \frac{k-1}{N} \quad (4.34)$$

となる。DFT の性質からわかるように、 N 個の周波数ポイントにおける値のうち半分しか独立でないで、片側スペクトル (onesided) だけを示すのが普通である。このときの周波数ポイントの範囲は N によって異なり、表 4.1 に従う。このときの PSD および規格化された PSD は次のようになる。

$$\frac{2\Delta t}{N} |X_D[k]|^2 \quad [1/\text{Hz}], \quad @v_k = \frac{1}{\Delta t} \frac{k-1}{N} \quad [\text{Hz}] \quad (4.35)$$

$$\frac{2}{N} |X_D[k]|^2, \quad @v_k = \frac{k-1}{N} \quad (4.36)$$

次に具体例として白色雑音の PSD を計算し、スペクトルがアンサンブル平均をとることによって一定の値に収束することをみる。また同時に、実際に Parseval の定理が成り立つことを確認する。

4.6.1 MATLAB : 白色雑音の PSD

```
>> N=1000; fn=[0: 1/N: 0.5];
>> psd=zeros(1, N);
>> for k=1:100
x=randn(1, N);
psd=psd+2*abs(fft(x)).^2/N;
if k==1
    subplot(3, 1, 1), plot(fn, 10*log10(psd(1: length(fn))/k));
    ylabel('averaging=1, [dB]');
elseif k==10
    subplot(3, 1, 2), plot(fn, 10*log10(psd(1: length(fn))/k));
    ylabel('averaging=10, [dB]');
elseif k==100
    subplot(3, 1, 3), plot(fn, 10*log10(psd(1: length(fn))/k));
    ylabel('averaging=100, [dB]');
    xlabel('normalized frequency');
    z=psd(1: length(fn))/k;
end
end

>> sum(z)/N
ans =1.0007

>> sum(x.^2)/N
ans =1.0069
```

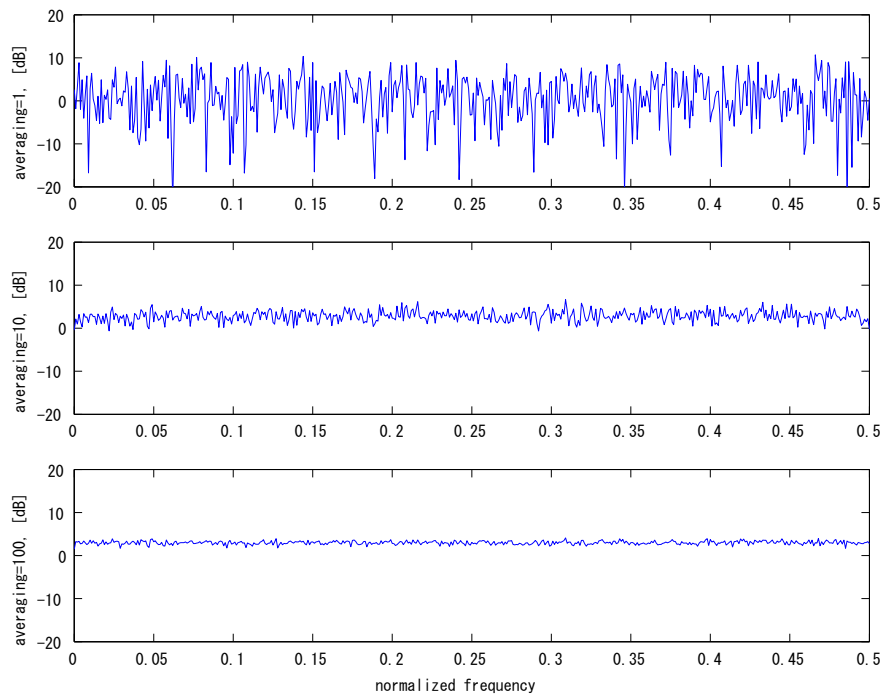


図 : 4.6.1

4.7 線型デジタルシステム

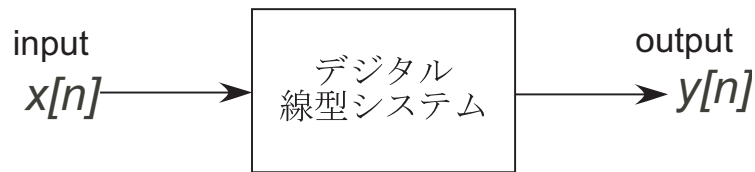


図 4.2: デジタル線型システムの入出力

図 4.2 のようなデジタル線型入出力システムを考える。
最初に、次式は離散的デルタ関数の定義である。

$$\delta[n] = \begin{cases} 1 & (n = 0) \\ 0 & (n \neq 0) \end{cases} \quad (4.37)$$

線型デジタルシステムに関連する諸性質を列挙する。

1. $x[n]$ のデルタ関数 $\delta[n]$ による表現

$$\begin{aligned} x[n] &= \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \\ &= \sum_{k=-\infty}^{\infty} x[n-k]\delta[k] \\ &= x[n] * \delta[n] \end{aligned} \quad (4.38)$$

2. ステップ関数 $u[n]$ とその表現

$$u[n] = \begin{cases} 0 & (n < 0) \\ 1 & (n \geq 0) \end{cases} \quad (4.39)$$

$$u[n] = \sum_{k=0}^{\infty} \delta[n-k] \quad (4.40)$$

3. $\delta[n]$ に対する応答 (インパルス応答) $h[n]$ を用いた出力 $y[n]$ の表現

$$\begin{aligned} y[n] &= \sum_{k=-\infty}^{\infty} x[k]h[n-k] \\ &= \sum_{k=-\infty}^{\infty} x[n-k]h[k] \\ &= x[n] * h[n] \end{aligned} \quad (4.41)$$

4. ステップ関数 $u[n]$ に対する応答 $c[n]$

$$h[n] = c[n] - c[n-1] \quad (4.42)$$

4.7.1 例題：移動平均のインパルス応答

次の差分方程式であらわされるシステムのインパルス応答を求めよ。このようなシステムを移動平均とよぶ。

$$y[n] = \frac{1}{3}[x[n] + x[n-1] + x[n-2]]$$

4.7.2 例題：差分方程式であらわされるデジタルシステムの出力

次の差分方程式であらわされるシステムがある。

$$y[n] = 3x[n] + 2x[n-3] - 3x[n-5]$$

このシステムに

$$x[n] = -3\delta[n-1] + 2\delta[n-3]$$

を入力したときの出力 $y[n]$ を求めよ。

4.8 z 変換

4.8.1 z 変換と逆 z 変換

図 4.3 は、z 変換の概念を示している。

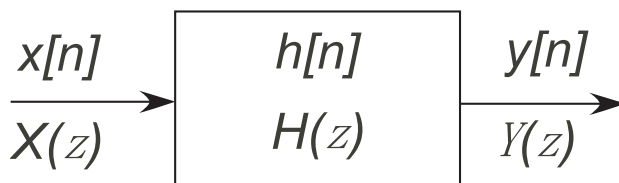


図 4.3: 線型デジタルシステムの表現

以下で z 変換 (ZT: Z Transform)、逆 z 変換 (IZT: Inverse Z Transform) を定義する。 $x[n]$ を離散時系列データとして

$$\begin{aligned} x[n] &\xrightarrow{\text{ZT}} X(z) \\ x[n] &\xleftarrow{\text{IZT}} X(z) \end{aligned} \tag{4.43}$$

とする。

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n} \tag{4.44}$$

$$x[n] = \frac{1}{2\pi i} \oint X(z)z^{n-1}dz \tag{4.45}$$

4.8.2 例題：z変換とその収束領域

次の離散時系列データのz変換を求め、その収束領域を調べよ。

$$x[n] = -a^n u[-n - 1] \quad (4.46)$$

ただし、 $u[n]$ は単位ステップ信号である。

連続時間系の場合、 s 平面の虚軸上 ($s = i\omega$) のラプラス変換がフーリエ変換に対応していた。 $z = e^{i\omega}$ とおけばわかるように、 z 平面の単位円上での z 変換が離散フーリエ変換に対応する。表 4.2 は、いくつかの関数の z 変換とその収束領域をまとめたものである。

表 4.2: z 変換

$x[n]$	$X(z)$	収束領域
$\delta[n]$	1	z 平面全体
$\delta[n - m]$	z^{-m}	z 平面全体
$u[n]$	$\frac{1}{1-z^{-1}}$	$ z > 1$
$a^n u[n]$	$\frac{1}{1-az^{-1}}$	$ z > a $
$\cos(\omega_0 n) u[n]$	$\frac{1 - \cos(\omega_0)z^{-1}}{1 - 2\cos(\omega_0)z^{-1} + z^{-2}}$	$ z > 1$
$\sin(\omega_0 n) u[n]$	$\frac{\sin(\omega_0)z^{-1}}{1 - 2\cos(\omega_0)z^{-1} + z^{-2}}$	$ z > 1$

4.8.3 z 変換の性質

以下は z 変換のいくつかの基本的性質である。

表 4.3: z 変換の基本的性質

$x[n]$	$X(z)$
$x[n - m]$	$z^{-m} X(z)$
$x[-n]$	$X(\frac{1}{z})$
$x_1[n] * x_2[n]$	$X_1(z) X_2(z)$
$nx[n]$	$-z \frac{dX(z)}{dz}$

4.8.4 離散システムの伝達関数

インパルス応答が $h[n]$ であらわされる離散時間システムの入力 $x[n]$ と出力 $y[n]$ の間には、次のような関係がある。

$$y[n] = h[n] * x[n] \quad (4.47)$$

両辺を z 変換することにより、次式が得られる。

$$Y(z) = H(z)X(z) \quad (4.48)$$

ただし、ここで $H(z)$ はシステムの伝達関数であり

$$H(z) = \sum_{k=-\infty}^{\infty} h[k]z^{-k} \quad (4.49)$$

であらわされる。

また、 $z = e^{i\omega}$ とおき $H(e^{i\omega})$ を調べることによりシステムの周波数応答が得られる。

4.8.5 例題：移動平均の伝達関数

例題 4.7.1 のデジタルシステムの伝達関数を求めよ。

4.8.6 例題：伝達関数から差分方程式を求める

次のような伝達関数をもつデジタルシステムの、入力 $x[n]$ と出力 $y[n]$ をむすびつける差分方程式を求めよ。

$$H(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 - 0.8z^{-1} + 0.64z^{-2}}$$

4.8.7 MATLAB：伝達関数の周波数応答

次のような伝達関数の周波数応答を調べる。

$$H(z) = \frac{1}{5} \frac{1 + z^{-1}}{1 - 0.6z^{-1}}$$

```
>> h = tf(0.2*[1, 1], [1, -.6], 0.1)
```

```
Transfer function:
```

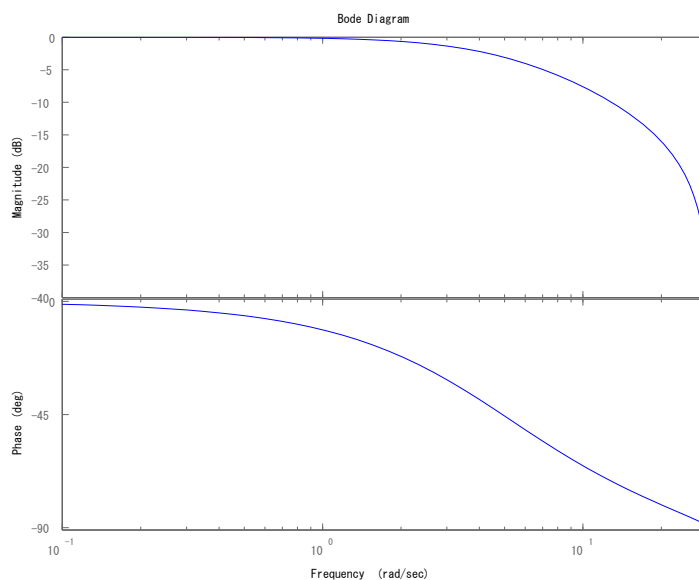
```
0.2 z + 0.2
```

```
-----
```

```
z - 0.6
```

```
サンプリング時間: 0.1
```

```
>> bode(h);
```



図：4.8.7

4.9 デジタルフィルター

デジタルフィルターはアナログフィルターにはない数々の利点をもっている。例えば、パラメータの変更が容易であること、温度変化や経年変化が無いこと、デジタル特有の特性をもつフィルターを作れるなどがある。問題点としては、量子化誤差があること、速度が演算プロセッサのクロックによって制限されることなどである。

4.9.1 FIR フィルターと IIR フィルター

デジタルフィルターは大きく分けて2種類に分類される。例えば、例題 4.7.1 でみたような差分方程式

$$y[n] = \frac{1}{3}[x[n] + x[n-1] + x[n-2]]$$

であらわされるシステムは、出力が入力だけによって表されており非再帰型とよばれる。この場合インパルス応答は有限個の項しか値をもたない。よって、このような型を FIR(Finite impulse response) フィルターとよぶ。伝達関数は次のような形になる。

$$H(z) = b_0 + b_1z^{-1} + \dots + b_nz^{-n} \quad (4.50)$$

また、例えば次のような差分方程式

$$y[n] = y[n-1] + x[n]$$

であらわされるシステムは、出力の一部が入力に戻る形になっているため再帰型とよばれる。この場合インパルス応答は無限に残るので、このような型を IIR(Infinite impulse response) フィルターとよぶ。伝達関数は次のような形になる。

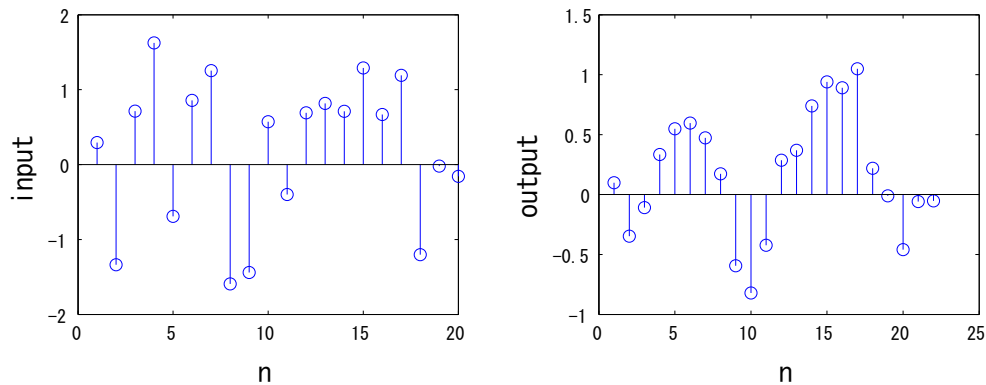
$$H(z) = \frac{b_0 + b_1z^{-1} + \dots + b_nz^{-n}}{1 + a_1z^{-1} + \dots + a_mz^{-m}} \quad (4.51)$$

4.9.2 MATLAB：インパルス応答によるフィルターの表現

$h(0) = h(1) = h(2) = 1/3$ のインパルス応答をもつ移動平均フィルターの出力の計算。

```
>> x=randn(20, 1);
>> h=[1, 1, 1]/3;
>> y=conv(h, x);
>> subplot(2, 2, 1), stem(x)
>> xlabel('n', 'fontsize', 14)
>> ylabel('input', 'fontsize', 14)

>> subplot(2, 2, 2), stem(y)
>> xlabel('n', 'fontsize', 14)
>> ylabel('output', 'fontsize', 14)
```



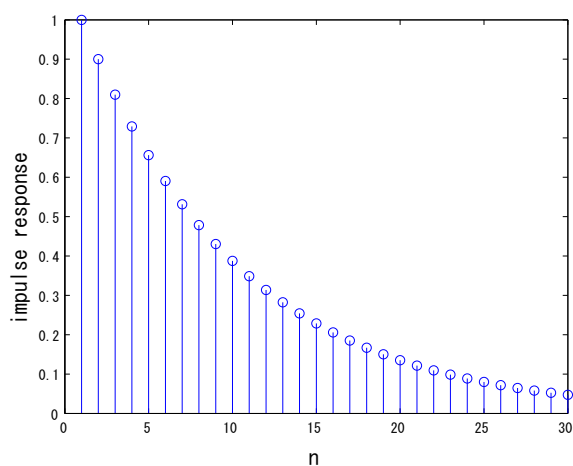
図：4.9.2

4.9.3 MATLAB：伝達関数によるフィルターの表現

次のような伝達関数をもつフィルターのインパルス応答の計算。

$$H(z) = \frac{1}{1 - 0.9z^{-1}}$$

```
>> impulse=[1; zeros(29, 1)];  
>> anu=1; den=[1, -0.9];  
>> y=filter(anu, den, impulse);  
>> stem(y)  
  
>> xlabel('n', 'fontsize', 14)  
>> ylabel('impulse response', 'fontsize', 14)
```



図：4.9.3

4.9.4 基本的デジタルフィルター (1)：積分器フィルター

図 4.4 は、積分器フィルターのブロックダイアグラムを示している。

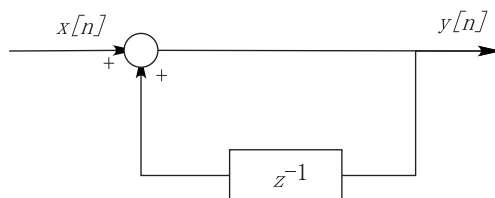


図 4.4: 積分器フィルターのブロックダイアグラム

差分方程式で書くと

$$y[n] = y[n - 1] + x[n]$$

であり、伝達関数は

$$H(z) = \frac{1}{1 - z^{-1}}$$

となるので、IIR フィルターである。その周波数応答は次のようになる。

4.9.5 MATLAB : 積分器の周波数応答

```
>> den=[1, -1]; anu=1;
>> [h1, w]=freqz(anu, den, 512, 2);
>> plot(w, abs(h1))

>> xlabel('normalized frequency', 'fontsize', 14)
>> ylabel('integrator', 'fontsize', 14)
```

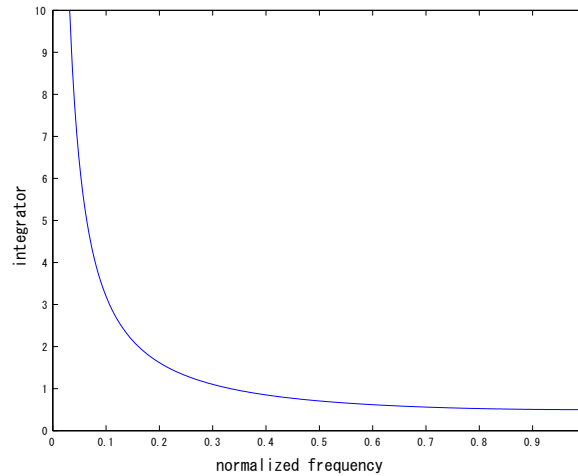


図 : 4.9.5

4.9.6 基本的デジタルフィルター (2) : 差分器フィルター

図 4.5 は、差分器フィルターのブロックダイアグラムを示している。

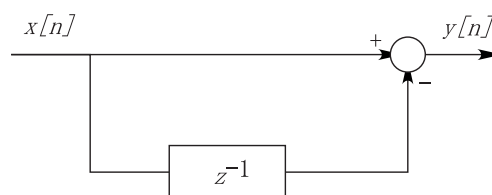


図 4.5: 差分器フィルターのブロックダイアグラム

差分方程式で書くと

$$y[n] = x[n] - x[n - 1]$$

であり、伝達関数は

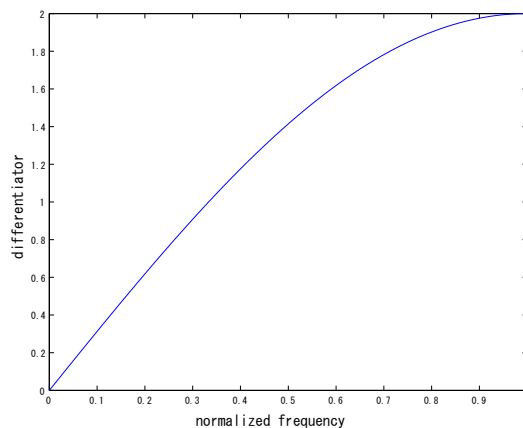
$$H(z) = 1 - z^{-1}$$

となるので、FIR フィルターである。その周波数応答は次のようになる。

4.9.7 MATLAB：差分器の周波数応答

```
>> anu=[1, -1]; den=1;
>> [h2, w]=freqz(anu, den, 512, 2);
>> plot(w, abs(h2))

>> xlabel('normalized frequency', 'fontsize', 14)
>> ylabel('differentiator', 'fontsize', 14)
```



図：4.9.7

4.9.8 基本的デジタルフィルター (3)：m 次 Comb フィルター

図 4.6 は、m 次 Comb フィルターのブロックダイアグラムを示している。

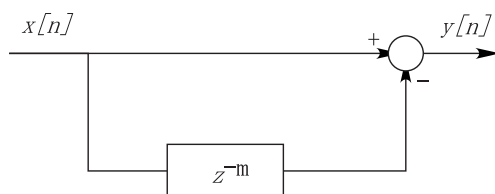


図 4.6: m 次 Comb フィルターのブロックダイアグラム

差分方程式で書くと

$$y[n] = x[n] - x[n - m]$$

であり、伝達関数は

$$H(z) = 1 - z^{-m}$$

となるので、FIR フィルターである。その周波数応答は次のようになり、特定の周波数成分を除去する機能をもつ。このようなフィルターはアナログでは作れない、デジタル特有のものである。

4.9.9 MATLAB : 5次 Comb フィルターの周波数応答

```
>> anu=[1, 0, 0, 0, 0, -1]; den=1;  
>> [h3, w]=freqz(anu, den, 512, 2);  
>> plot(w, abs(h3))  
  
>> xlabel('normalized frequency', 'fontsize', 14)  
>> ylabel('5th comb filter', 'fontsize', 14)
```

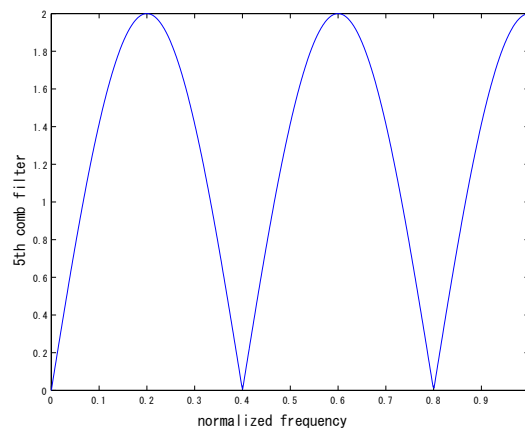


図 : 4.9.9