

AD変換および AD変換器のSN比

傘 昊

東京都市大学 知識工学部
情報通信工学科

Email: hsan@tcu.ac.jp

Tel/Fax: 03-5707-2132

目次

AD変換の基礎:

サンプリング, 量子化, コード化

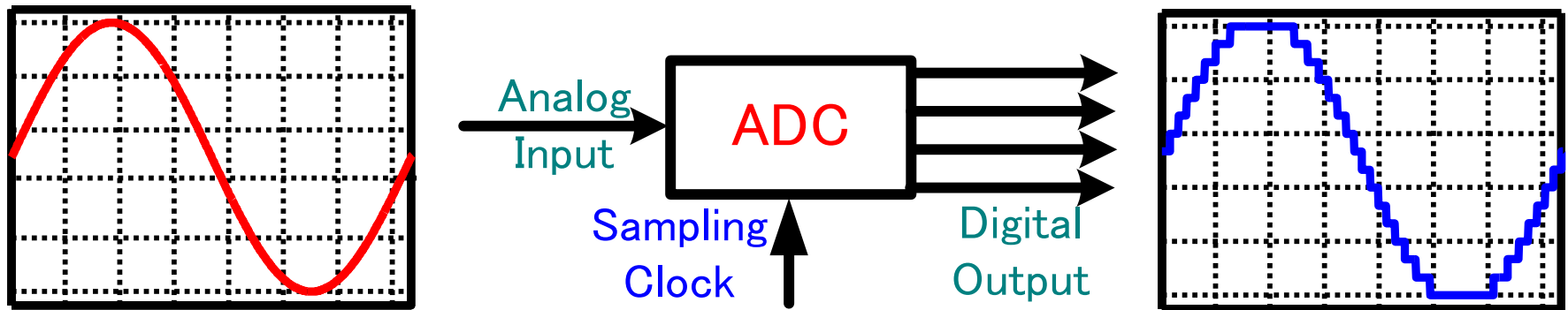
AD変換の分解能とADCのSNR

FFTを用いた信号処理の基礎

AD変換の基礎

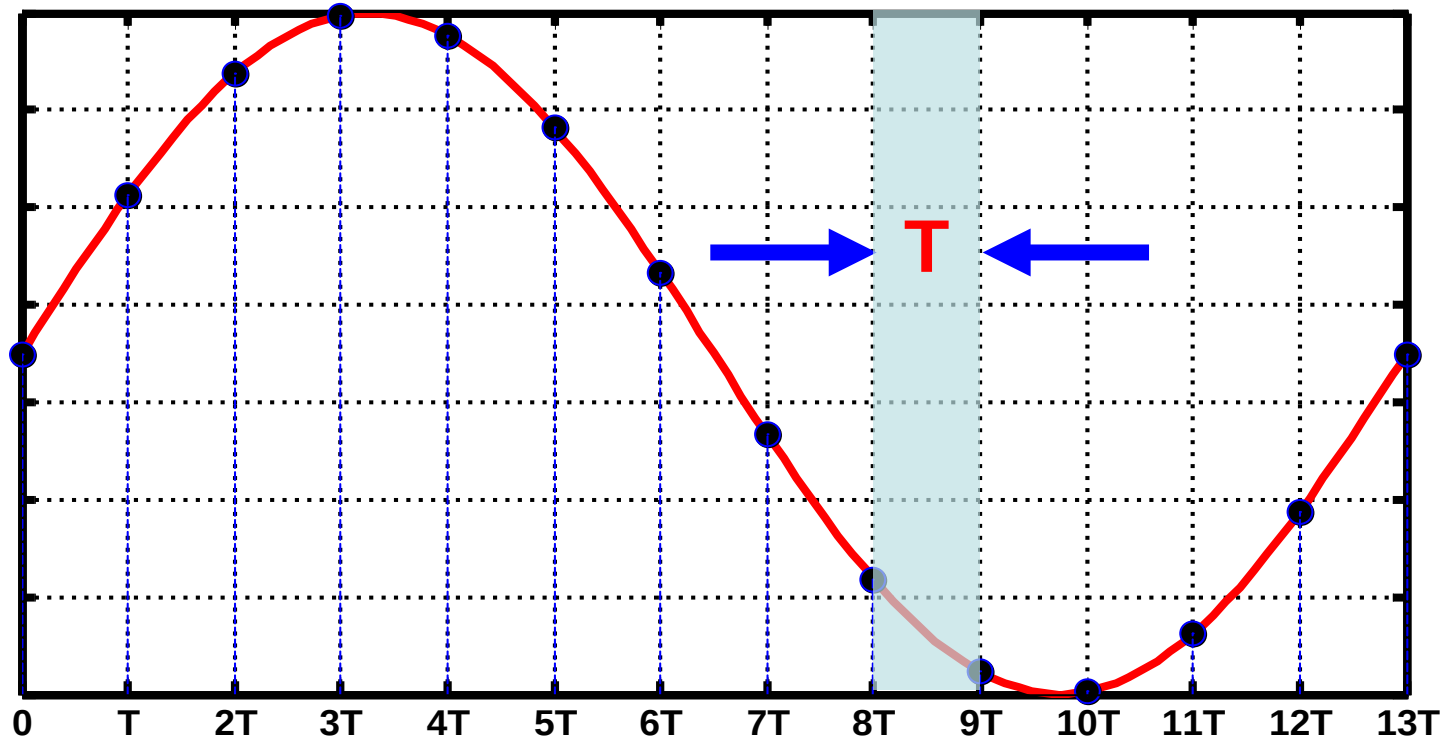
AD変換

- アナログ信号(電波、音声、電圧、電流等)をデジタル信号 (0,1,1,0,...) に変換する



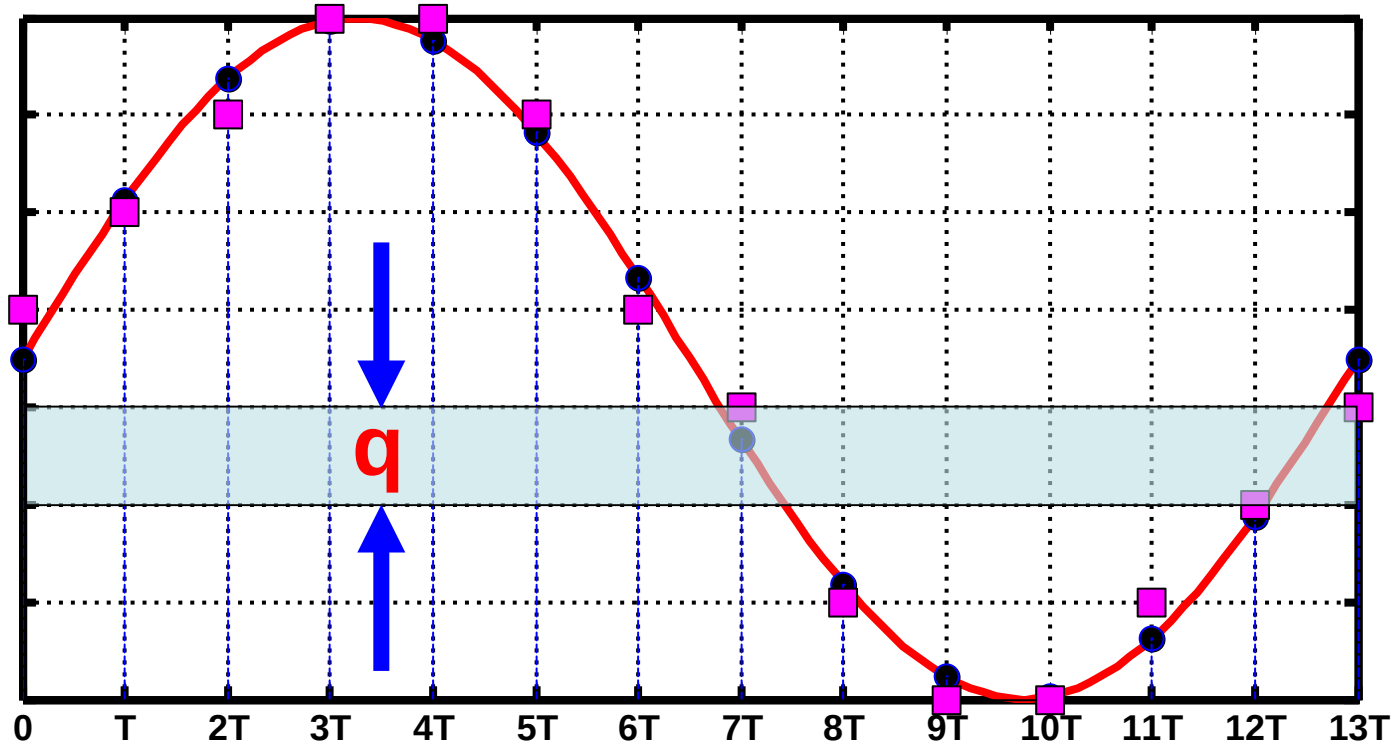
- 連続信号 → 離散信号
⇒ デジタル信号処理が可能

標本化(サンプリング)



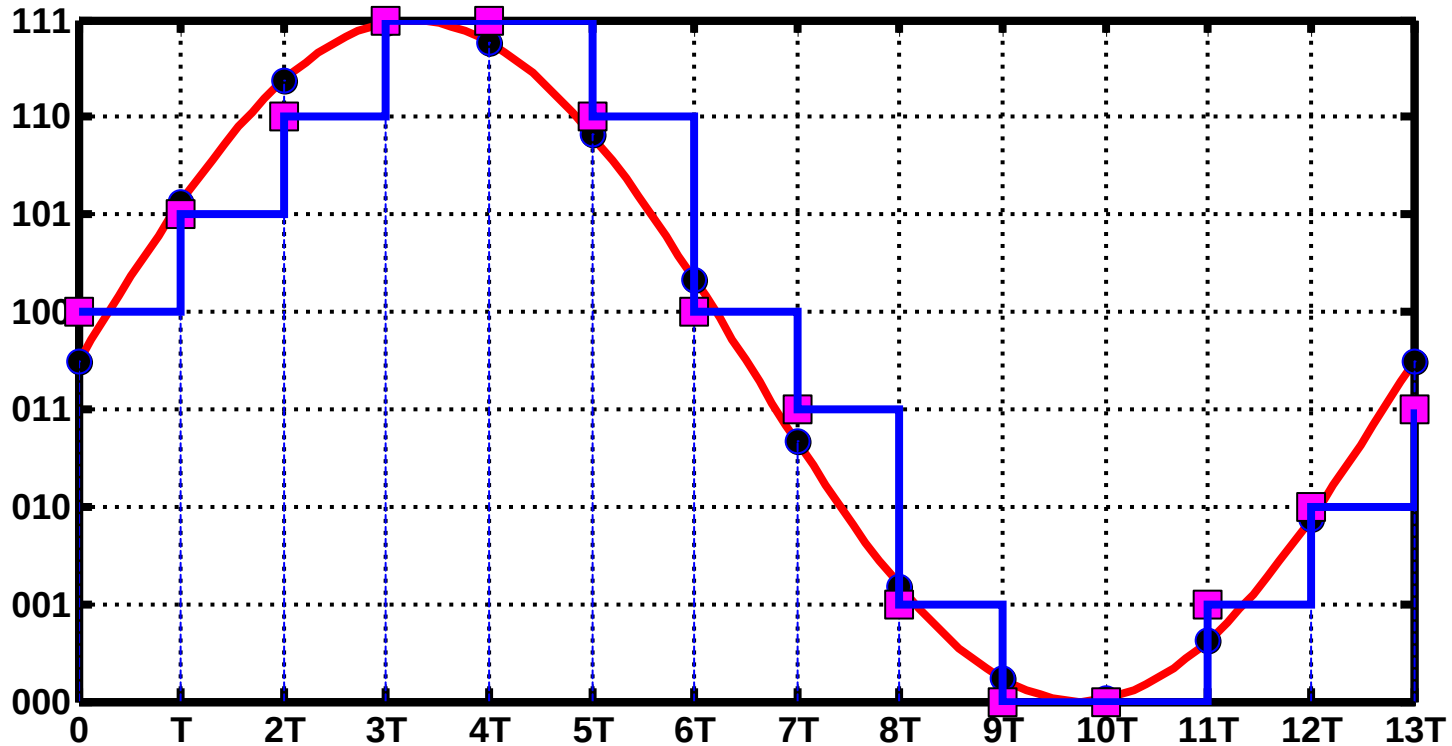
- 連続信号の振幅を一定の時間間隔(周期 T)で切り出す
- 時間離散的な瞬間値(データ)に置き換える
- 時間軸の離散化

量子化



- 標本化された振幅を単位振幅(q)の整数倍にする
- 振幅データを四捨五入で当てはめる
- 振幅軸の離散化

コード化



- 量子化された振幅を「0」,「1」に符号化
- デジタル信号を出力

デジタルによるアナログの表現

3ビットによるアナログ量の表現例

デジタル列 (2^2 2^1 2^0)			アナログ
1	1	1	7
1	1	0	6
1	0	1	5
1	0	0	4
0	1	1	3
0	1	0	2
0	0	1	1
0	0	0	0

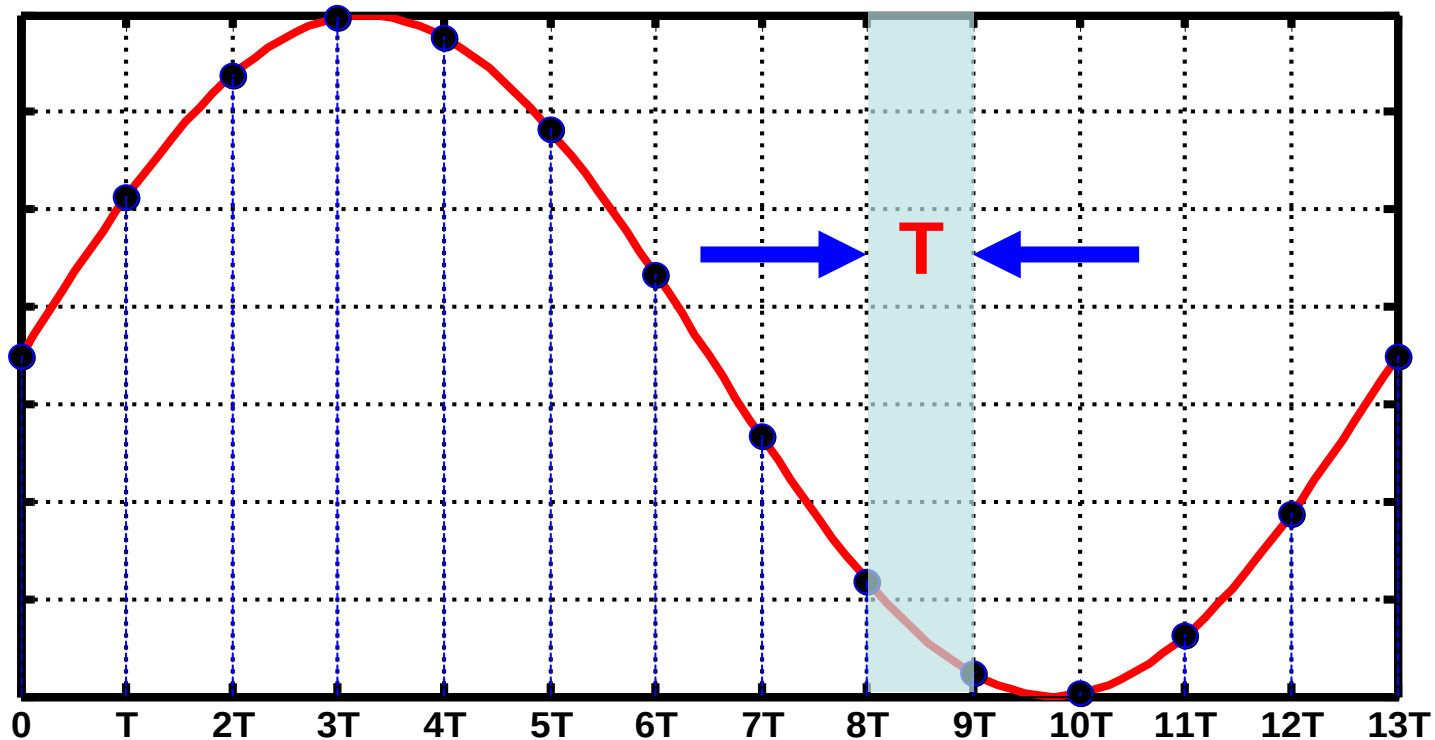
$$1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 = 5$$

多ビットの例

デジタル列	アナログ値
3ビット⇒	8レベル
4ビット⇒	16レベル
8ビット⇒	256レベル
10ビット⇒	1024レベル
16ビット⇒	65536レベル

サンプリングおよびサンプリング定理

サンプリング周波数

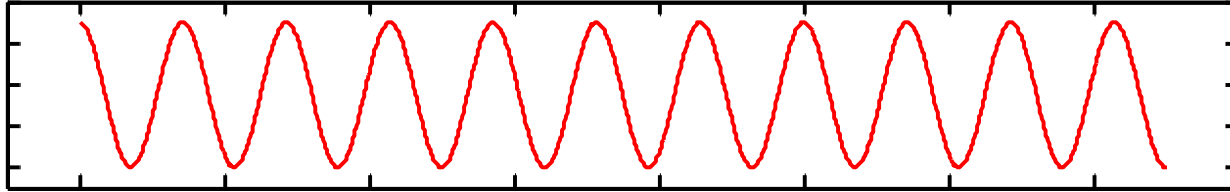


- サンプリング周波数 : $F_s = 1/T$
⇒ AD変換速度 (SPS : Sampling Per Second, Hz)

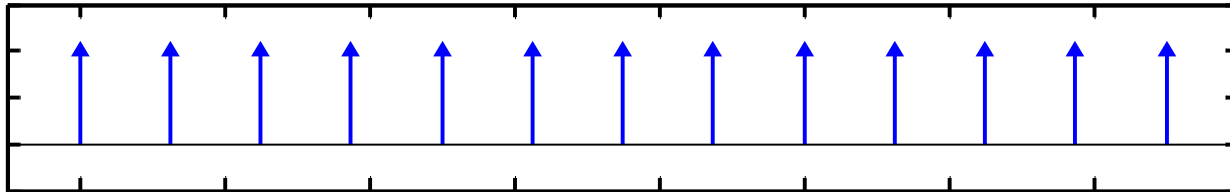
サンプリング周波数とエイリアシング

入力信号

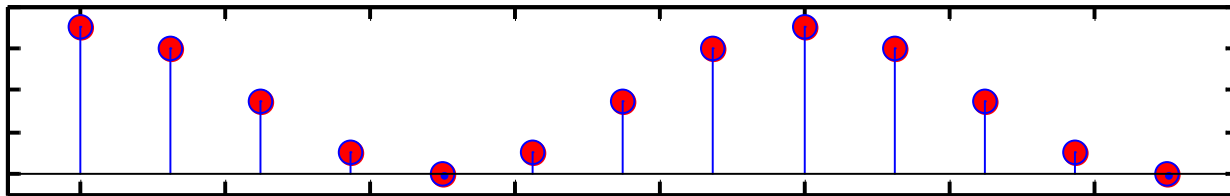
F_{in} と F_s
が近い



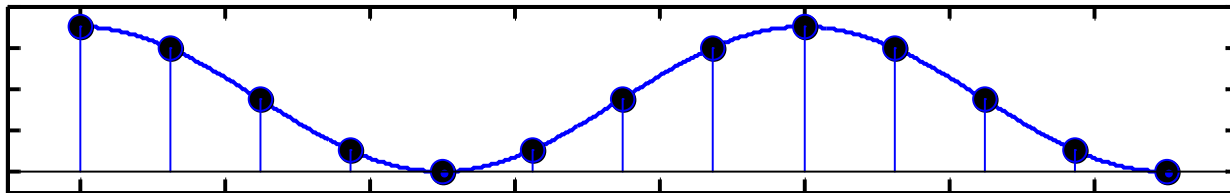
サンプリング
信号



離散信号



再生信号



異なる信号
が再現される

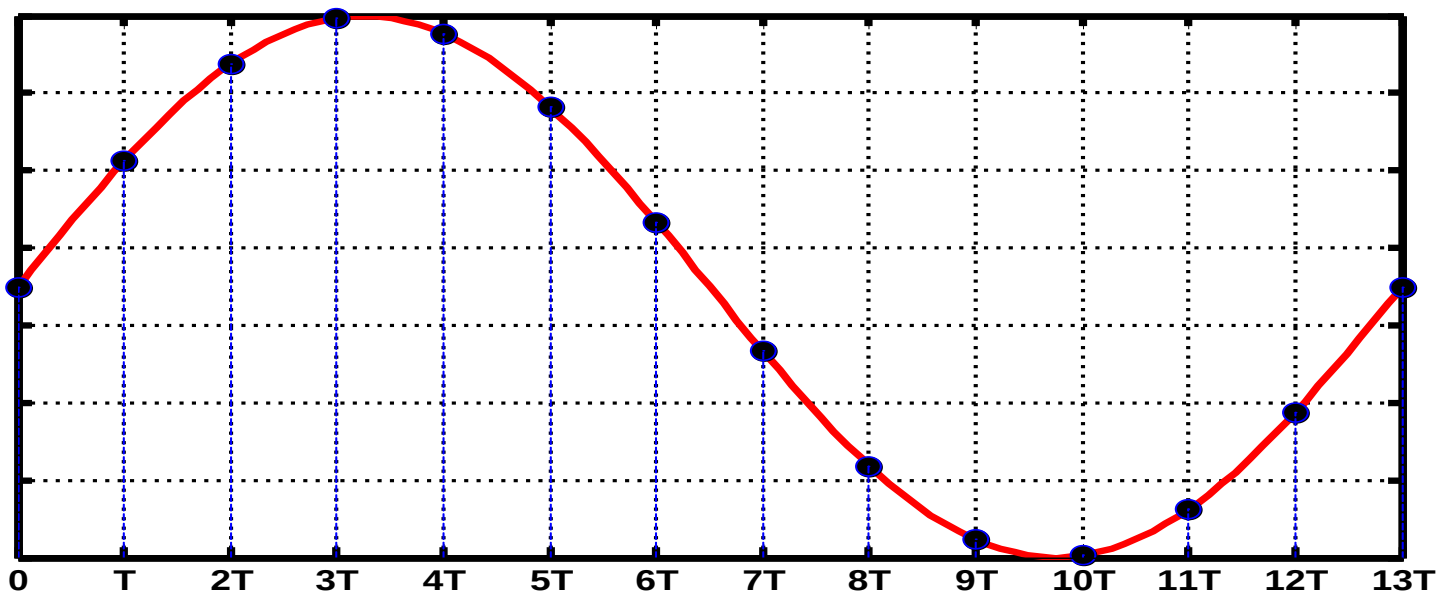
エイリアシング: Aliasing (折り返し)

エリアシング



サンプリング定理

ナイキスト定理、ナイキスト・シャノンの定理

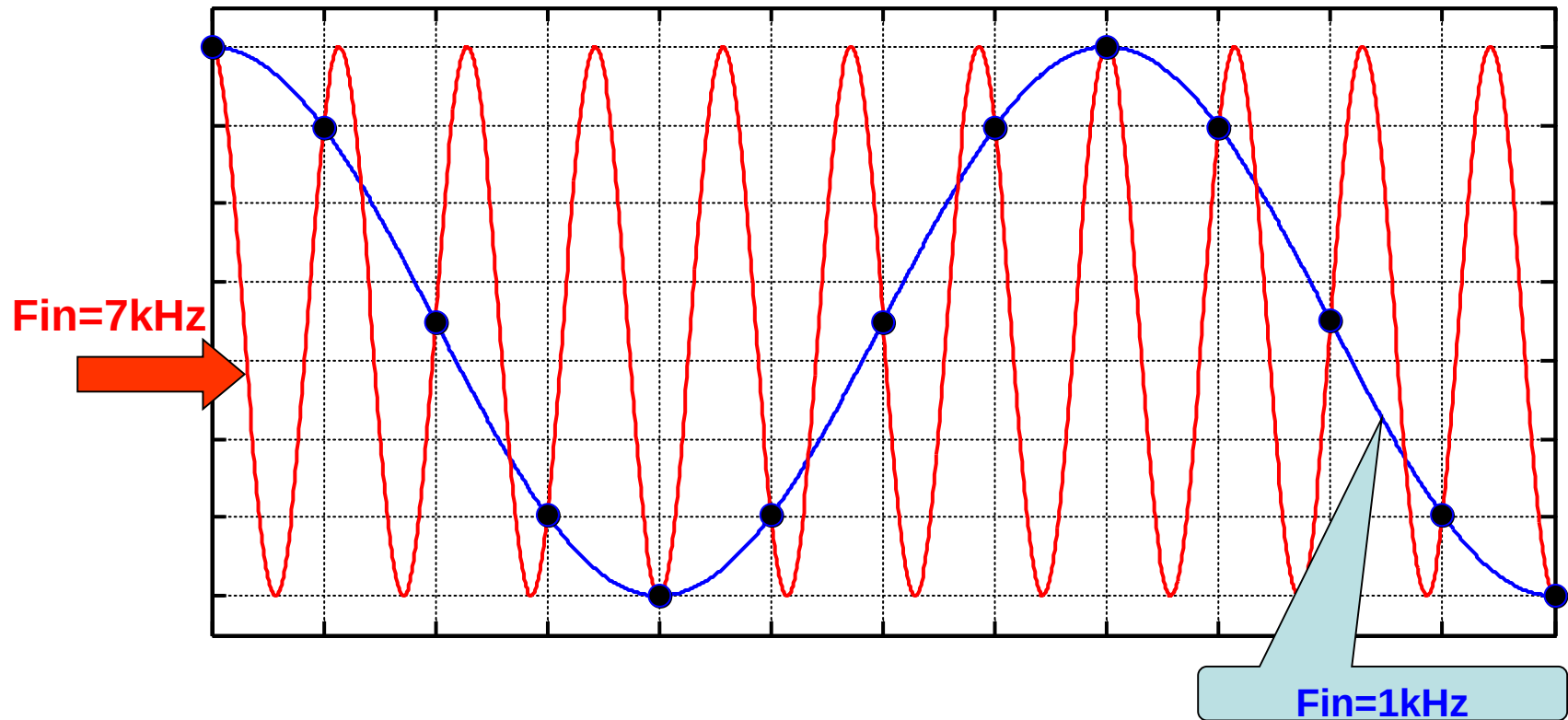


- エイリアシングを防ぐため:

$F_s > 2F_{in}$ が必要

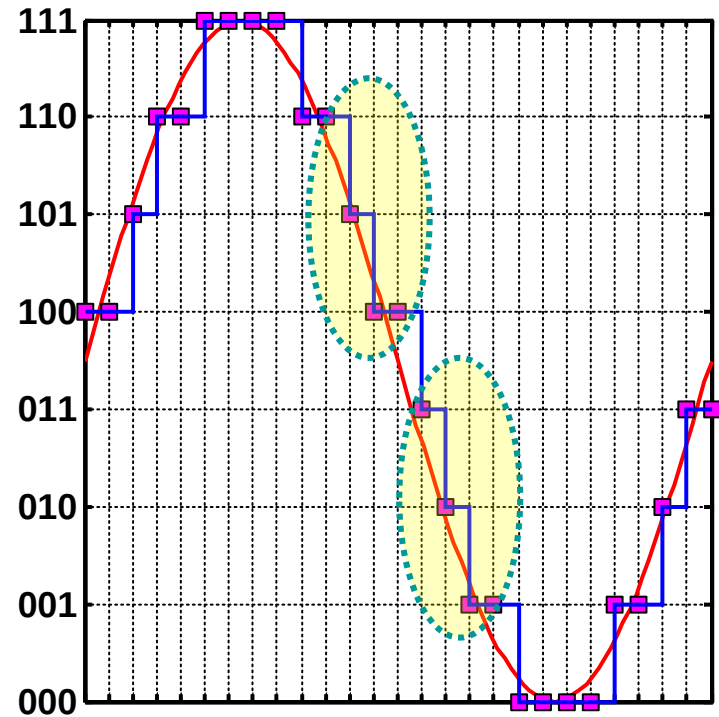
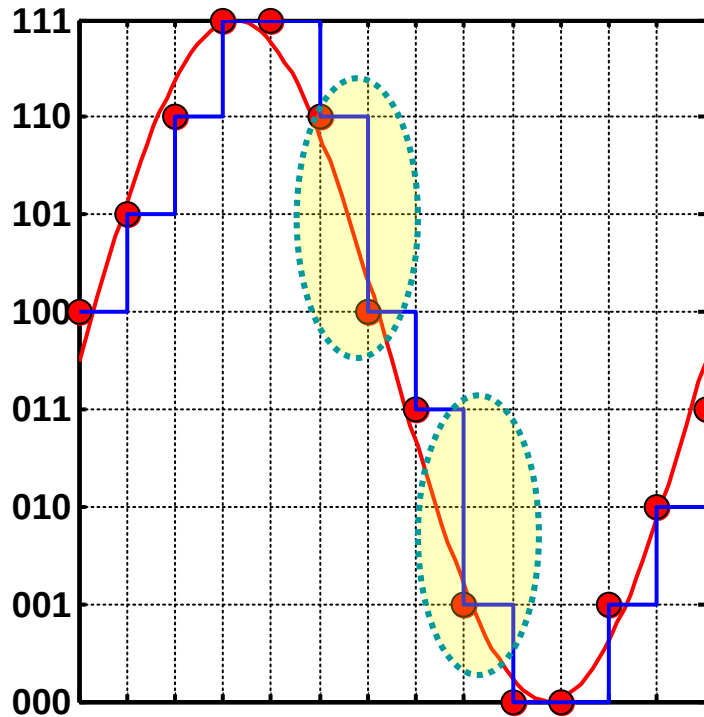
- $F_s > 2F_{in}$** : デジタル信号からアナログ信号が復元できる
- ナイキスト周波数(Nyquist Frequency): $F_n = F_s/2$

エイリアシング(Aliasing)



- $F_s=8\text{KHz}$ の時、 1KHz と 7KHz の信号は区別できない
- ナイquist周波数以上の入力信号を含むと、異なる連続信号がサンプリングでは区別できない

オーバーサンプリング

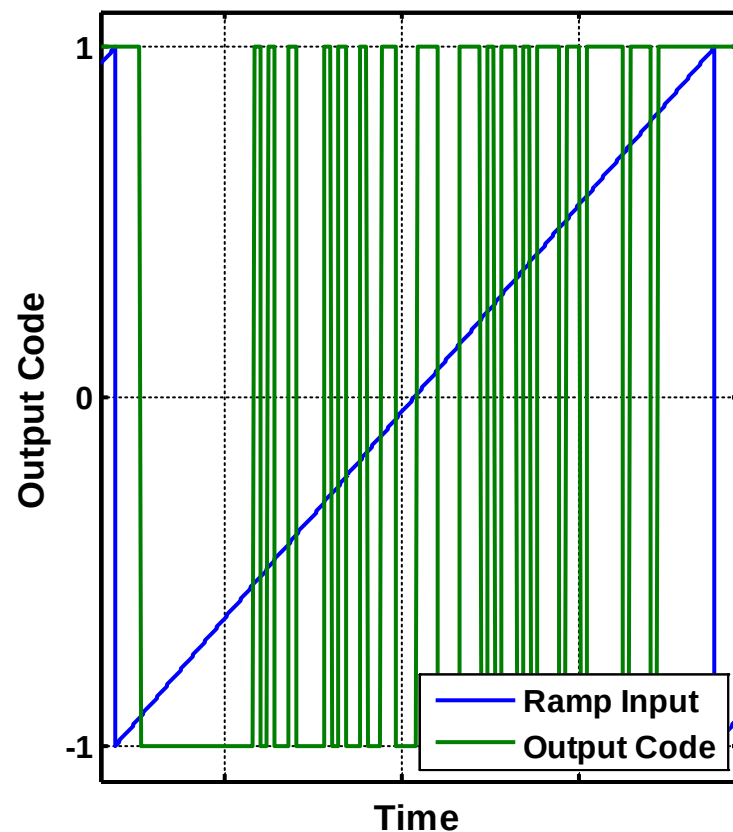
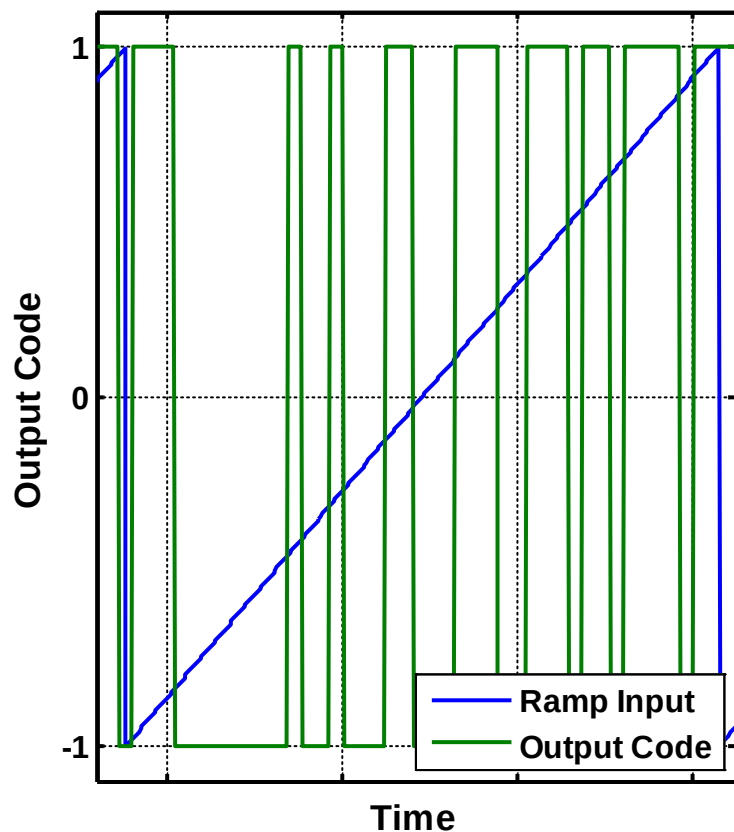


- ナイquist周波数を超える周波数でサンプリングを行う

$$F_s \gg 2F_{in}$$

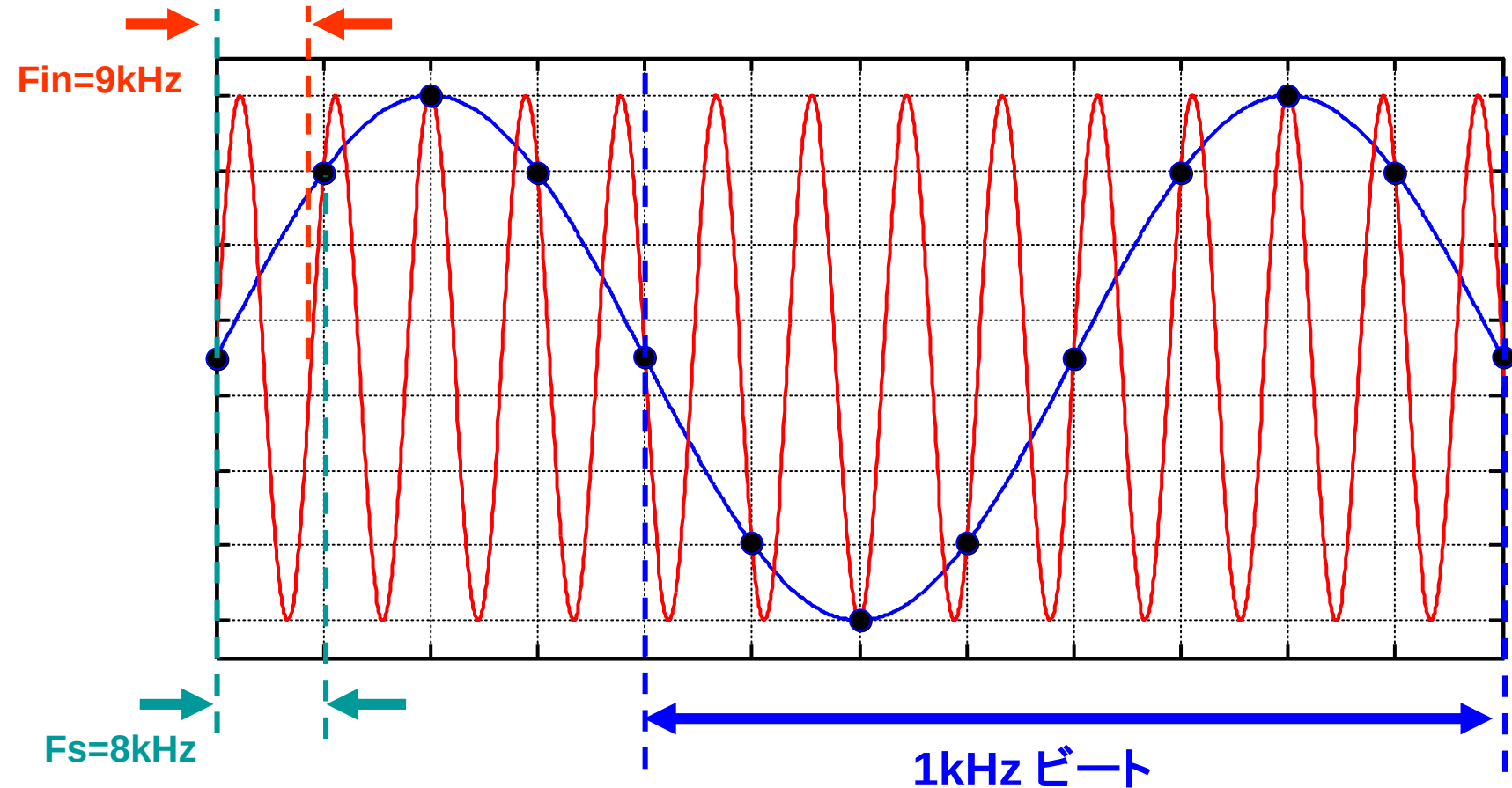
- F_s が大きいほど、AD変換の誤差が小さくなる

オーバーサンプリング



- オーバーサンプリングを行うとAD変換の精度が向上

アンダーサンプリング



- ナイquist周波数以下の周波数でサンプリングを行う: $F_s < F_{in}$
- サンプリング定理から、エイリアシングが起こるが、高周波搬送波の検波、通信用広帯域信号の交換に用いられる

サンプリング方法のまとめ

オーバーサンプリング

$$F_{in} < F_{nyquist}/2 = F_s/4$$

オーバーサンプリングADC
 $\Delta\Sigma$ ADC

音声・通信ベースバンド帯
AD変換

ナイキスト・サンプリング

$$F_{in} < F_{nyquist} = F_s/2$$

ナイキストADC
Flash型, SAR型
Pipeline型

汎用AD変換

アンダー・サンプリング

$$F_{in} > F_s$$

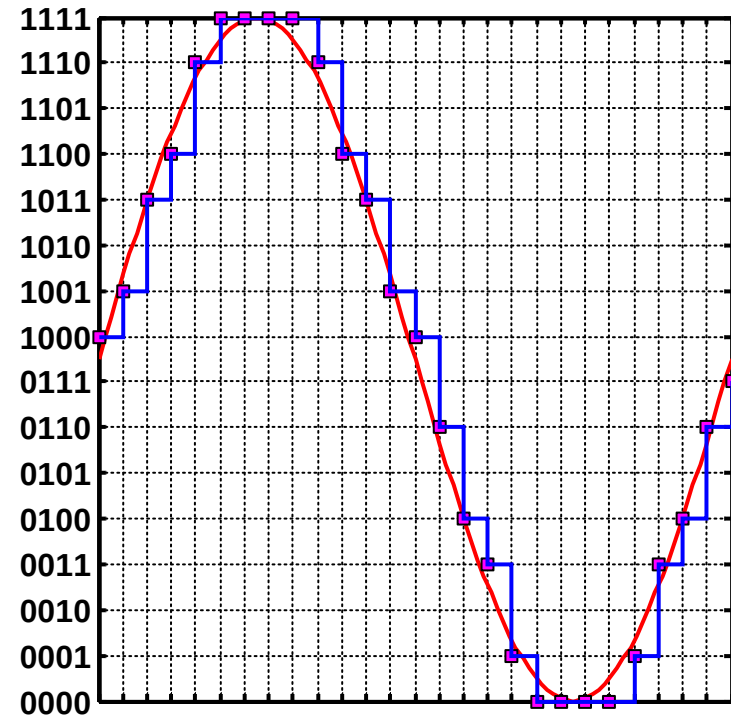
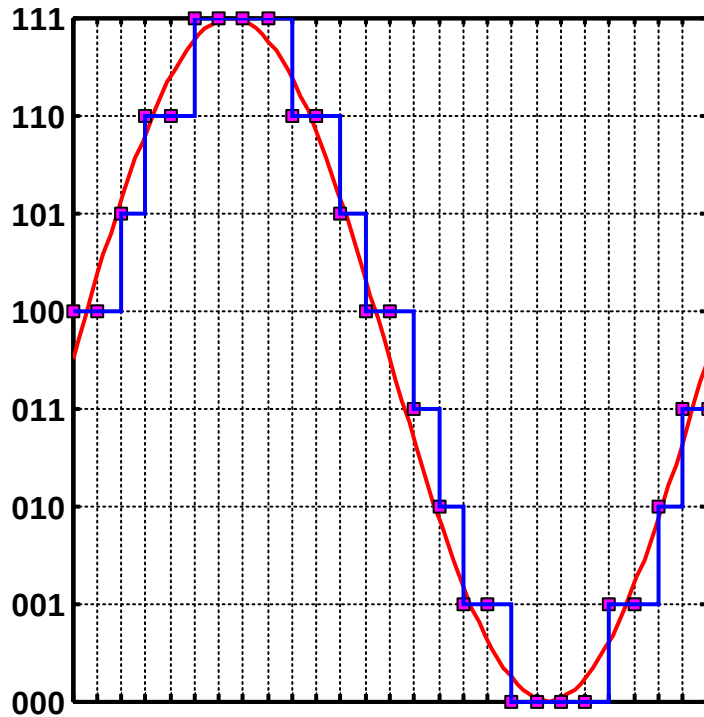
Pipeline型ADC

通信IF/RF帯の広帯域
AD変換

AD変換の分解能

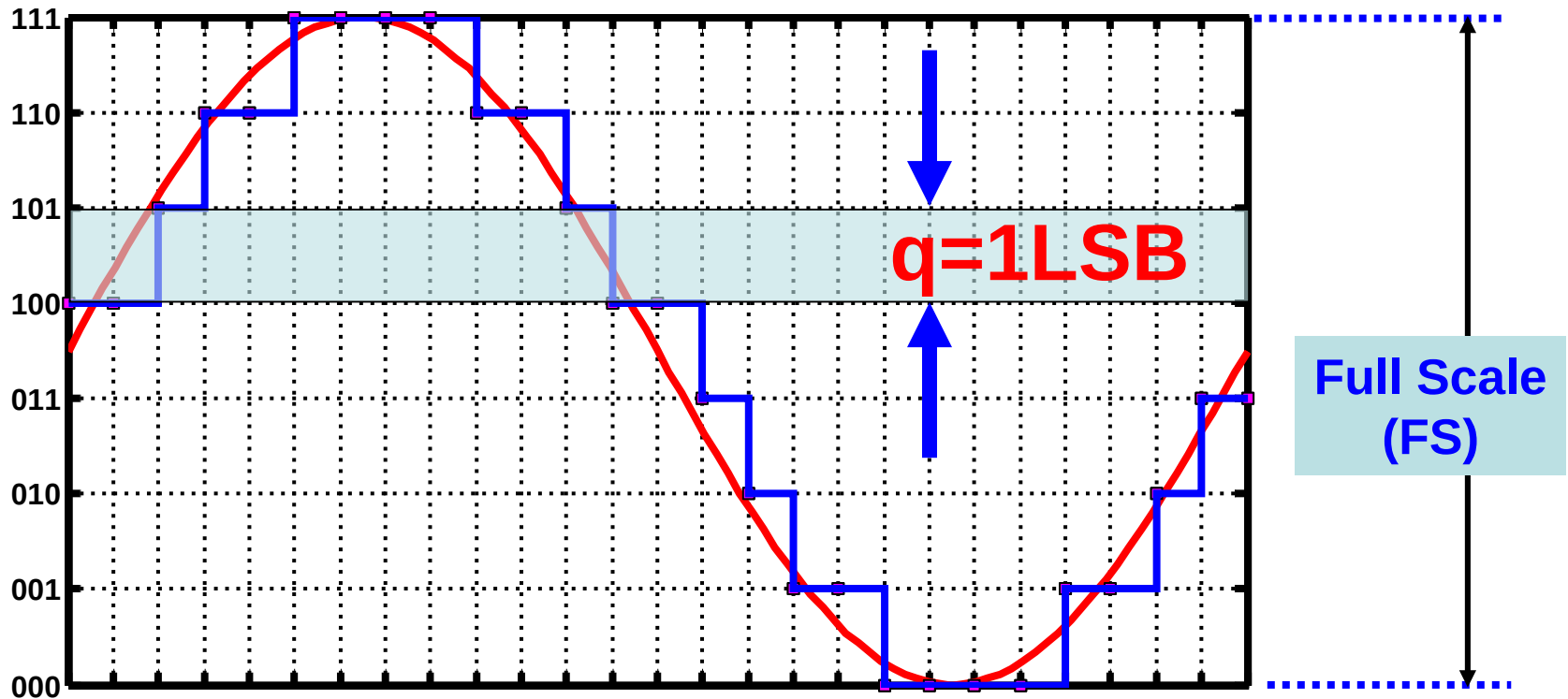
分解能(Resolution)

単位: bit



- 分解能とは、一回AD変換で出力できるbit数(N)
- 量子化のための基準値の数は 2^N
⇒AD変換の精度

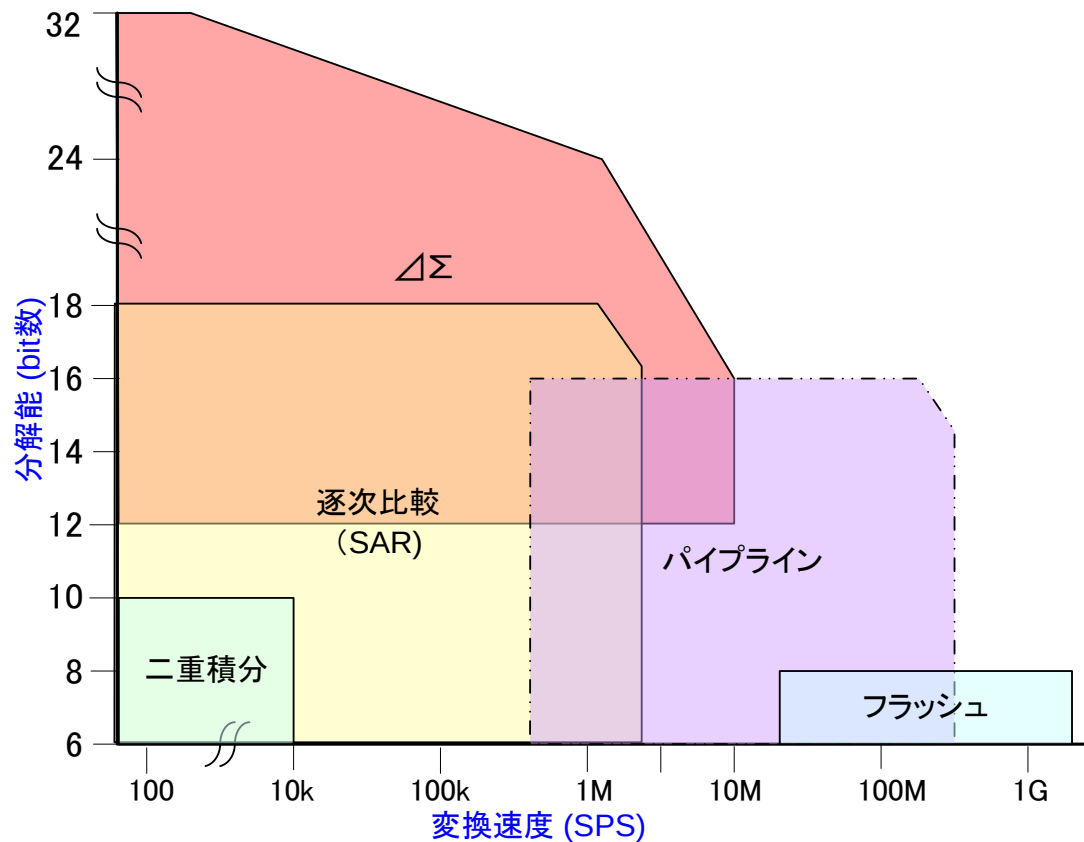
最小分解能



- AD変換で区別できる最小振幅の値(量子化の刻み)
- $1\text{LSB} = \text{FS} / (2^N - 1)$ (N: bit数)

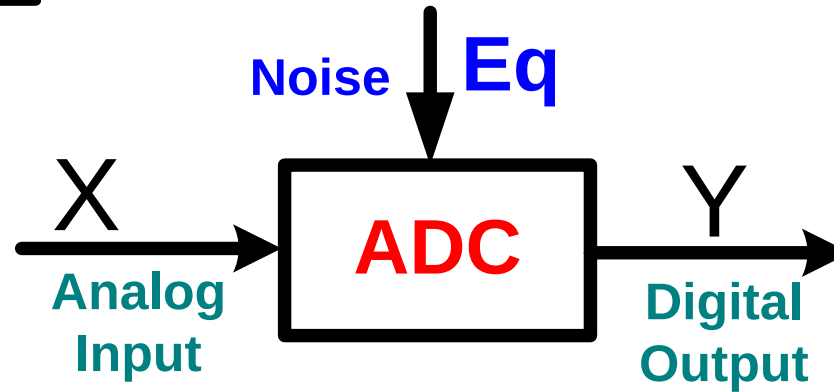
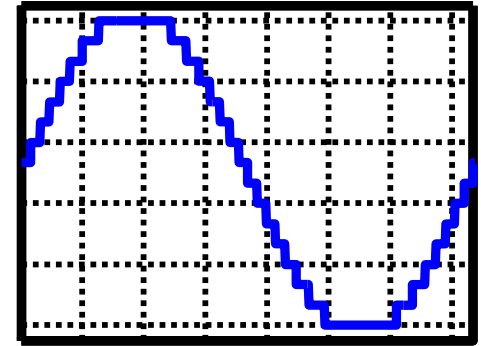
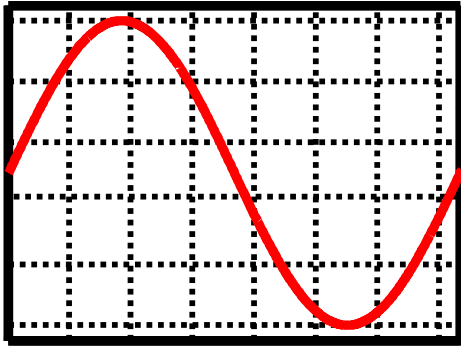
ADCの重要仕様

- 変換速度: サンプルング・レート(SPS)
- 分解能: bit数



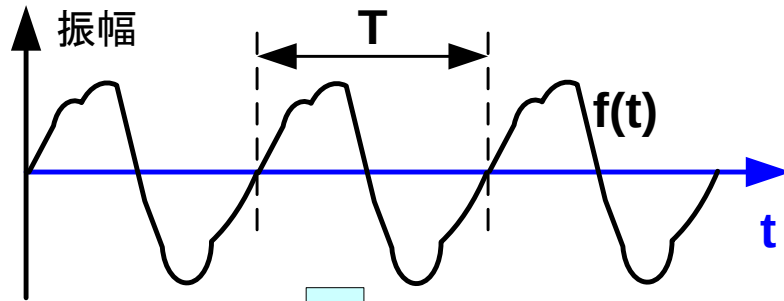
AD変換の量子化誤差とSNR

ADCのモデル

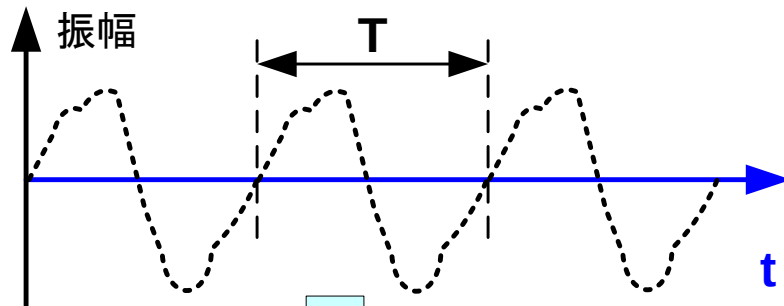


$$Y = X + Eq$$

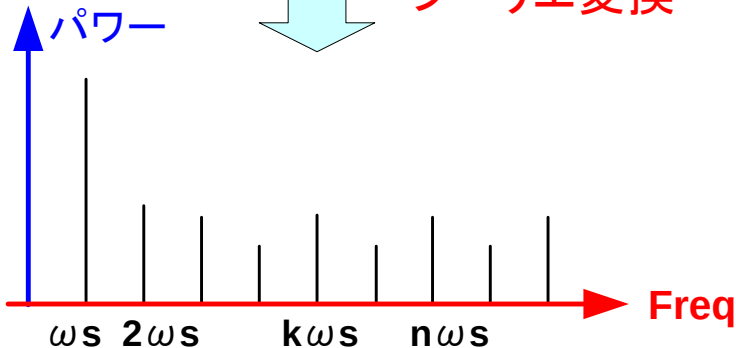
アナログ信号のスペクトル解析



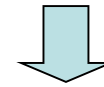
↓ サンプルング



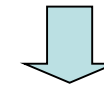
↓ フーリエ変換



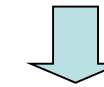
- 繰返し信号は正弦波と余弦波の積和



- 離散時系列データ

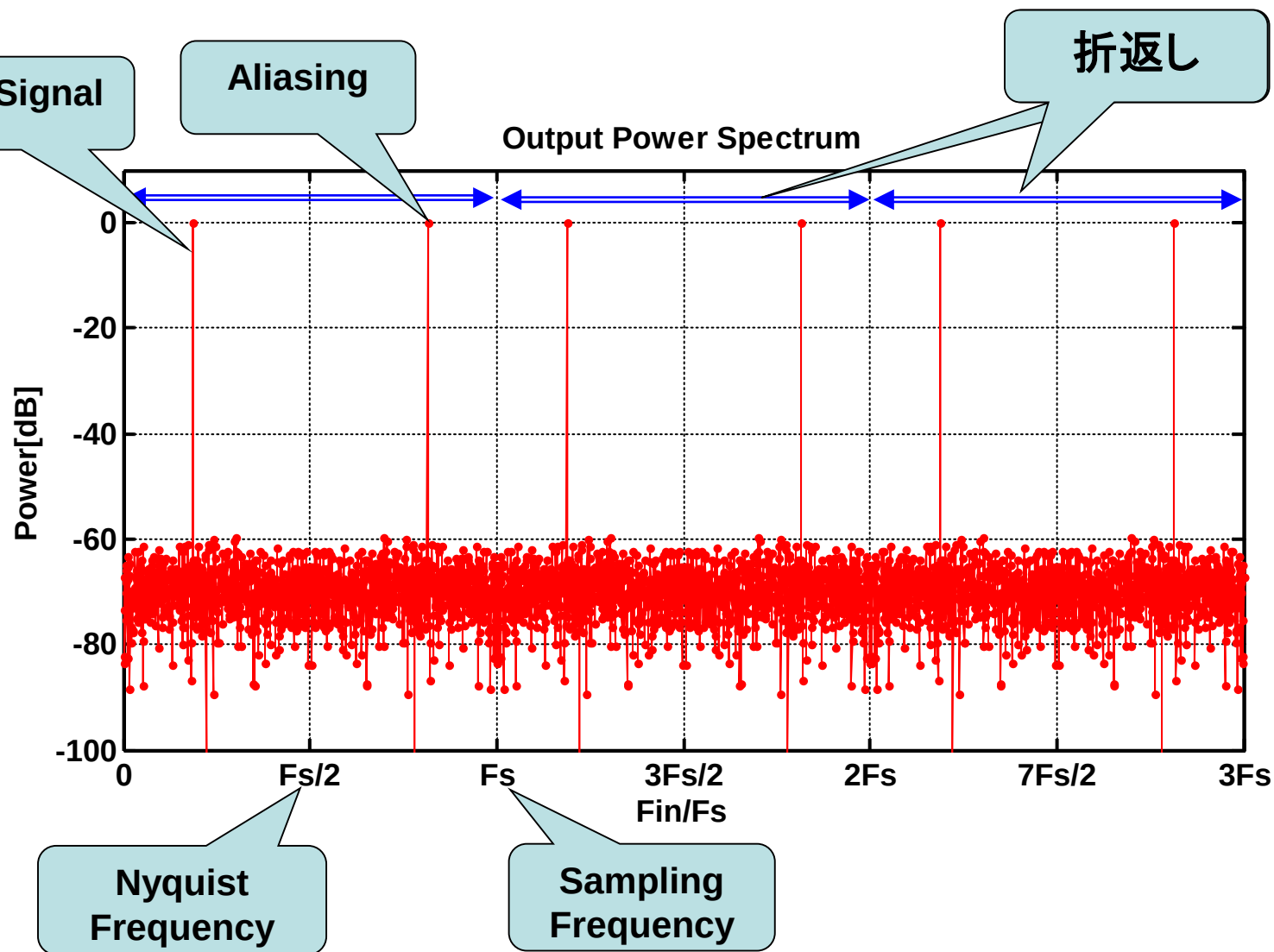


- 連続信号のスペクトル

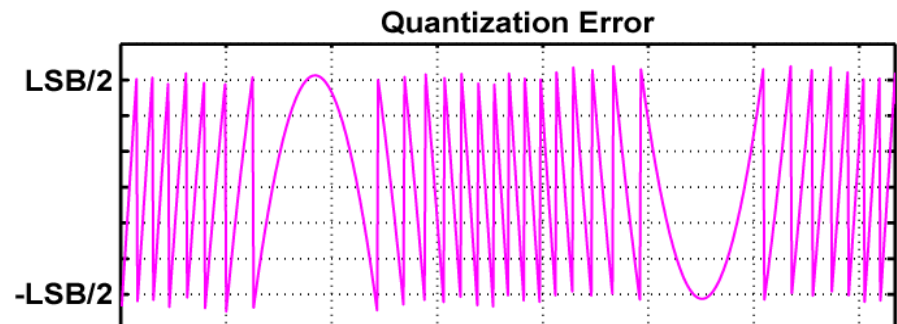
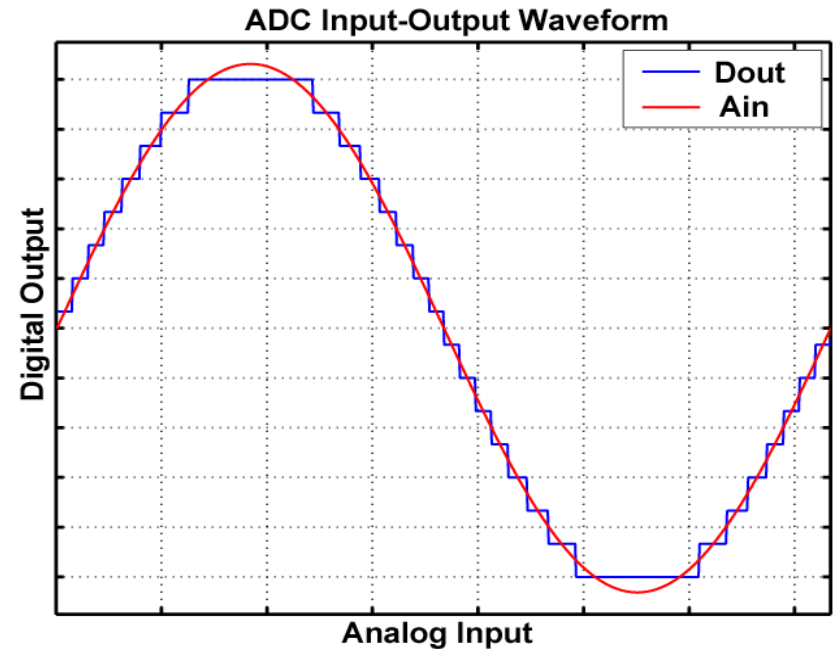
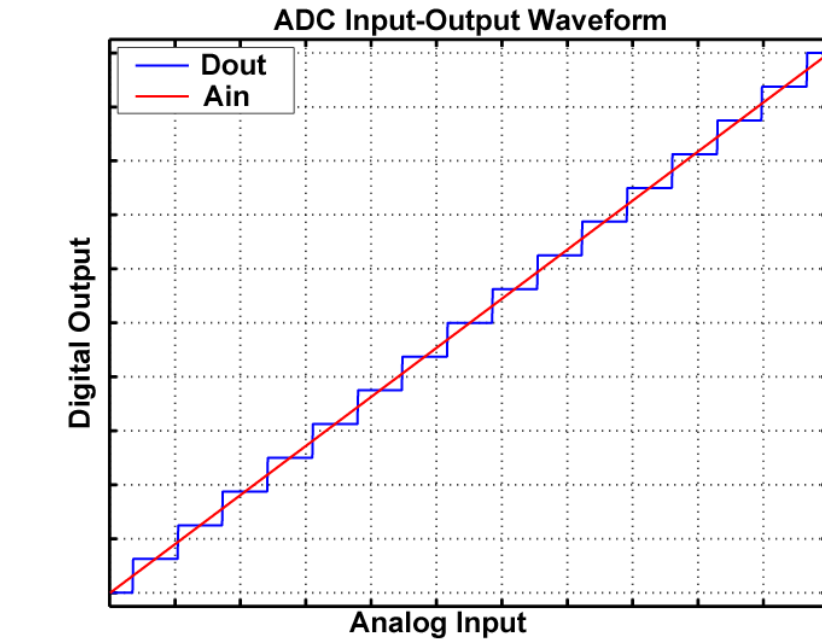


- 周波数成分解析
信号とノイズの分離
フィルタリング etc.

正弦波入力時ADC出力スペクトル



AD変換の量子化誤差

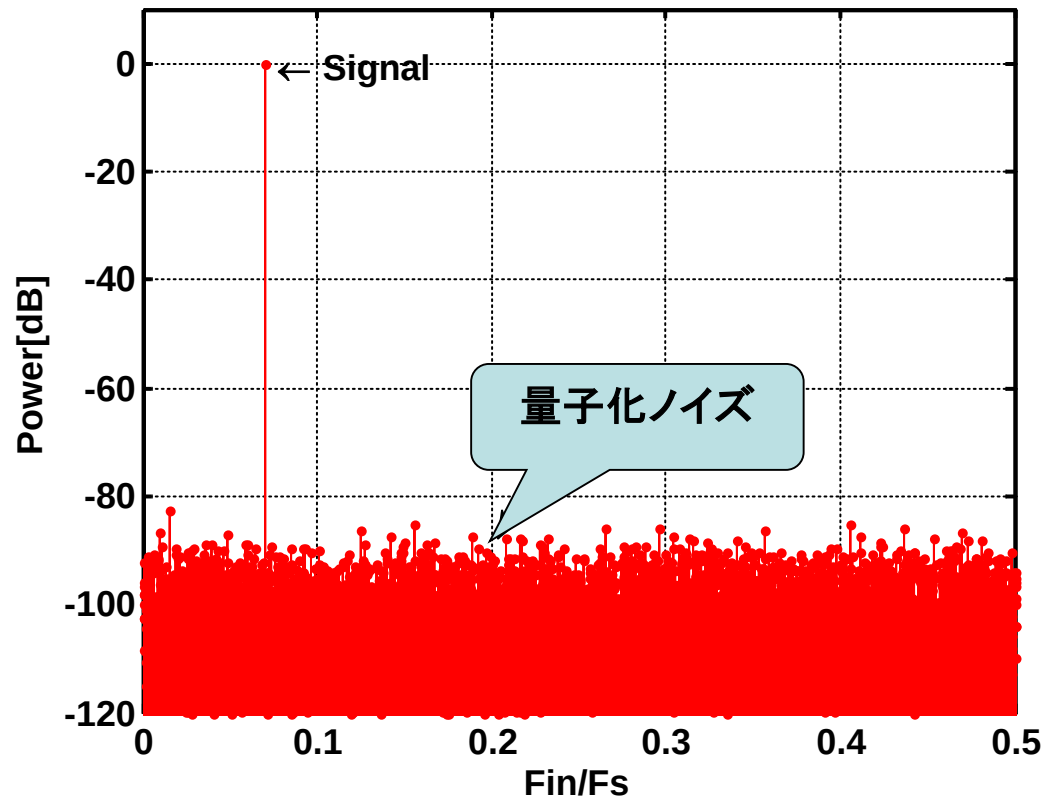


量子化誤差はAD変換における本質の誤差、
小さくするにはBit数(分解能)をあげるしかない

AD変換器のSNR

Signal-to-Noise Ratio

Output Power Spectrum



ADCのパワースペクトラム
(Doutに対し、FFT処理で得られる)

SNR[dB]

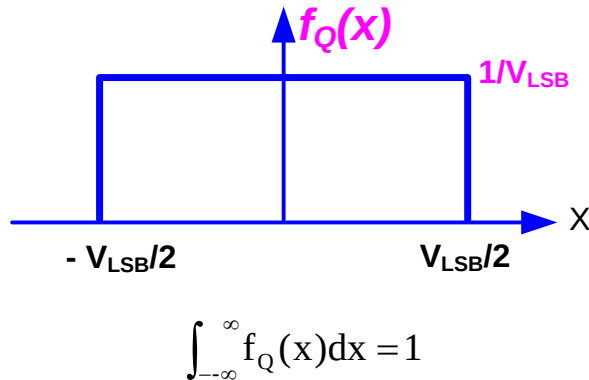
$$= 10 \cdot \log \left[\frac{\text{Signal Power}}{\text{Noise Power}} \right]$$

$$= 10 \cdot \log \frac{V_{\text{Signal(rms)}}^2}{V_{\text{Noise(rms)}}^2}$$

$$= 20 \cdot \log \frac{V_{\text{Signal(rms)}}}{V_{\text{Noise(rms)}}}$$

量子化誤差とSNR

Probability density function
for quantization error



量子化ノイズの電圧

$$\begin{aligned} V_{Q(\text{rms})} &= \left[\int_{-\infty}^{\infty} x^2 f_Q(x) dx \right]^{1/2} \\ &= \left[\frac{1}{V_{\text{LSB}}} \int_{-\frac{V_{\text{LSB}}}{2}}^{\frac{V_{\text{LSB}}}{2}} x^2 dx \right]^{1/2} = \frac{V_{\text{LSB}}}{\sqrt{12}} \end{aligned}$$

入力信号の電圧

$$V_{\text{Sin(rms)}} = \frac{V_{\text{Asin}}}{\sqrt{2}} = \frac{2^N \cdot V_{\text{LSB}} / 2}{\sqrt{2}} = \frac{2^N \cdot V_{\text{LSB}}}{2\sqrt{2}}$$

理論値

$$\text{SNR[dB]} = 20 \cdot \log\left(\sqrt{\frac{3}{2}} \cdot 2^N\right) = 6.02N + 1.76[\text{dB}]$$

N: ADCのbit数

ADCの有効ビット

(ENOB: Effective Number of Bit)

ADCのbit数	SNR[dB]
6	37.9
8	49.9
10	62.0
12	74.0
14	86.0
16	98.1
18	110.1

$$\text{SNR[dB]} = 6.02N + 1.76 \text{ [dB]}$$

$$\text{ENOB} = (\text{SNR} - 1.76) / 6.02$$

SNRの6dB向上



ADCの1bitの精度向上

信号処理の基礎

- データの取得
- 信号の時間応答と周波数応答
- 波形の整形
- 信号の特徴抽出

波形の時間応答(表示)

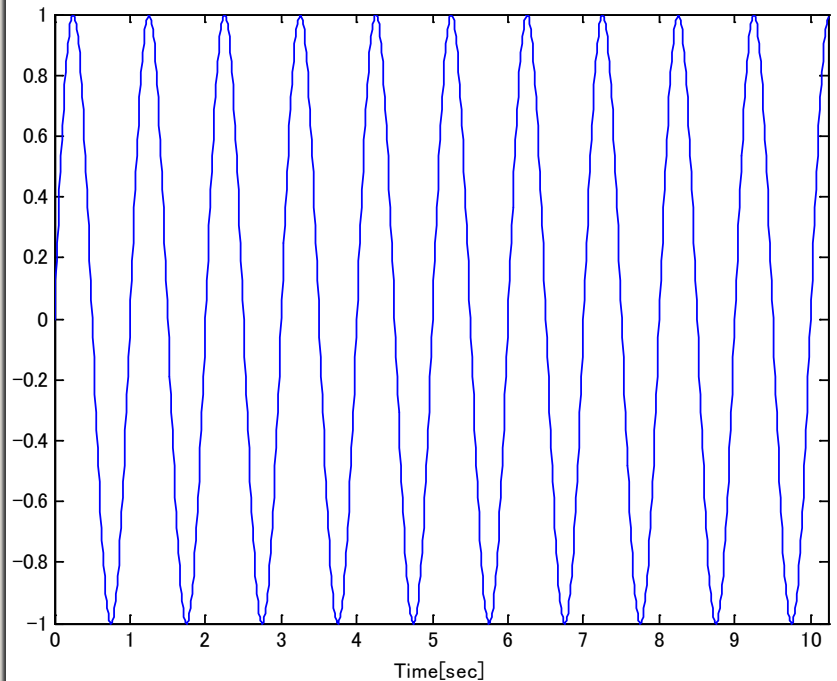
```

1  n=1024;dt=0.01;
2  t=(1:1024)*dt-dt;
3  omega=2*pi*1;% [Hz]
4  y=sin(omega*t);
5  plot(t,y);
6  xlabel('Time[sec]');
7  ylabel('Magnitude');
8  axis([0 t(end) -1 1]);
0

```

arfftcomp.m × p136.m ×

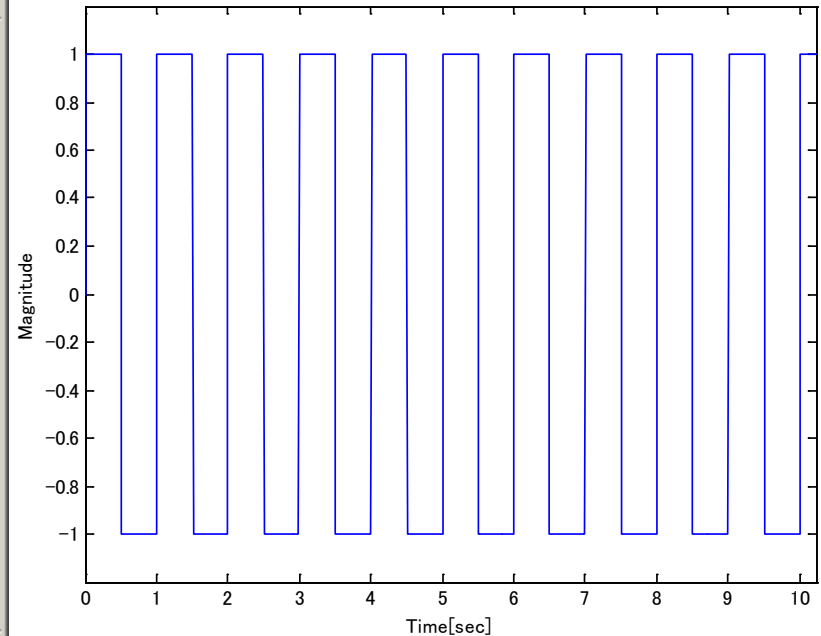
スクリプト 行 4 列 16 上書



Data Index	1	2	...	$n/2 + 1$	$n/2 + 2$	...	n
Time	$(1-1) dt = 0$	dt	...	$(n/2) dt$	$(n/2 + 1) dt$	...	$(n-1)dt$
Data Index	1	2	...	513	514	...	1024
Time	0	dt	...	$512 * dt$	$513 * dt$	...	$1023 * dt$

矩形波の時間応答(表示)

```
エディタ - C:\Users\san\Documents\MATLAB\MATLAB_Handbo...
ファイル(F) 編集(E) テキスト(T) 移動(G) セル(C) ツール(O)
- 1.0 + ÷ 1.1 × % % %
1 n=1024;dt=0.01;
2 t=(1:1024)*dt-dt;
3 omega=2*pi*1;% [Hz]
4 y=sign(sin(omega*t));
5 plot(t,y);
6 xlabel('Time[sec]');
7 ylabel('Magnitude');
8 axis([0 t(end) -1.2 1.2]);
arfftcomp.m x p136.m x p137.m x
スクリプト 行 4 列 22 上書
```



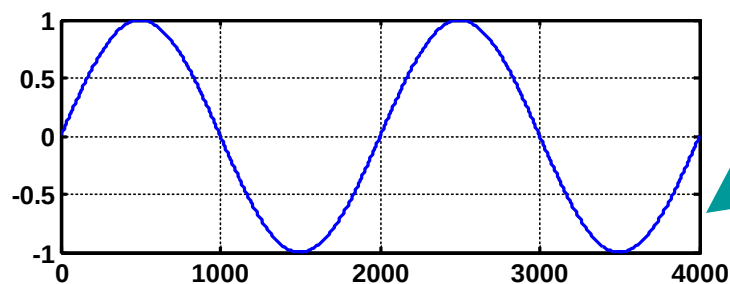
フーリエ変換(Fourier Transform)

- 見た目：
「時間領域」の信号 $\Rightarrow$ 「周波数領域」に表現
- 「位置」情報によって記述された関数に対し、
「周波数」による記述をあたえる関数空間
の間の変換
- フーリエ変換によって、
音や電磁波などの波の波形から
どのような周波数の波が含まれているかを
抜き出すことができる。

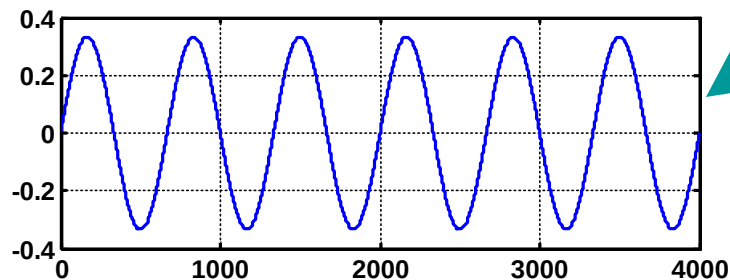
フーリエ級数(Fourier Transform)

- 複雑な周期関数は
正弦関数や余弦関数の和として表される

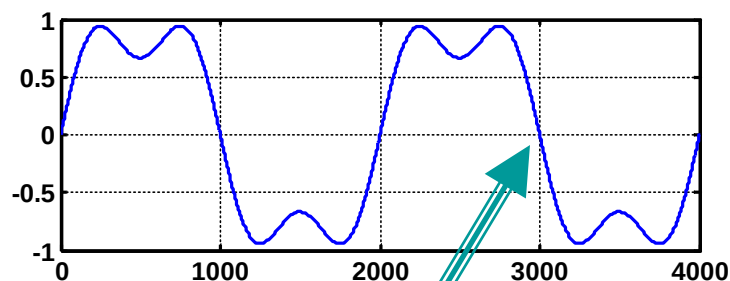
矩形波のフーリエ級数加算



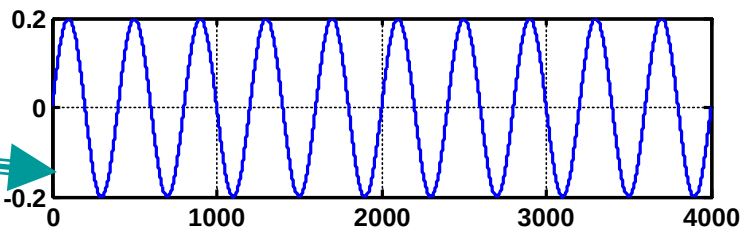
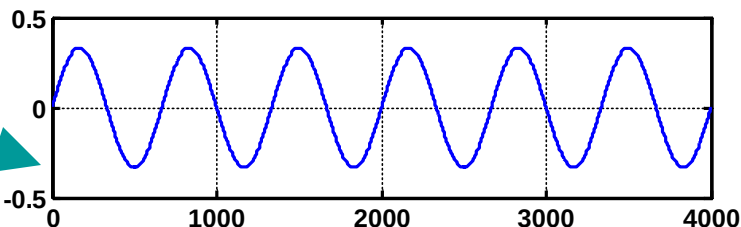
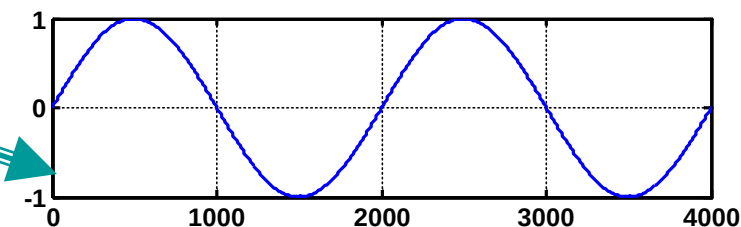
基本波



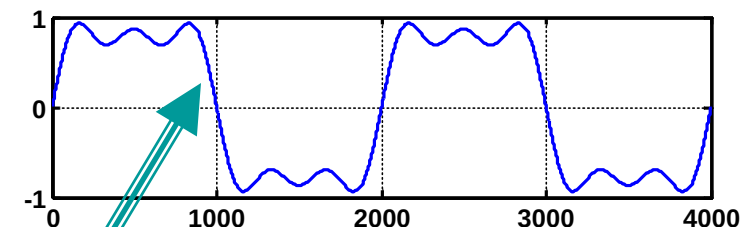
3次高調波



基本波 + 3次高調波

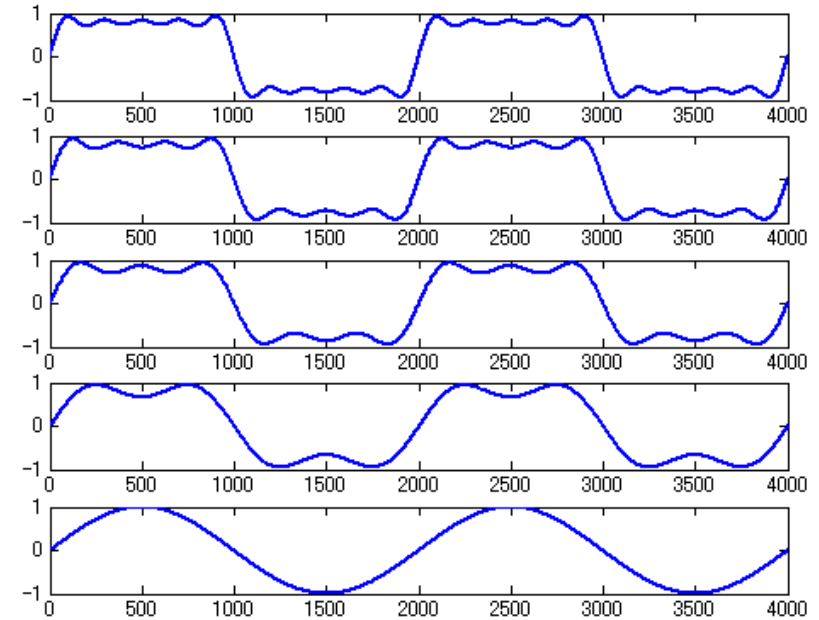
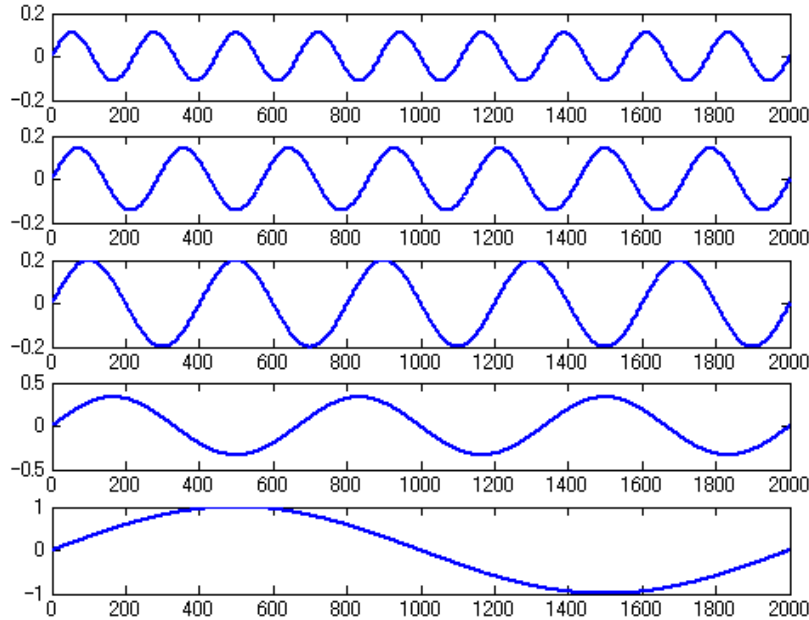


5次高調波



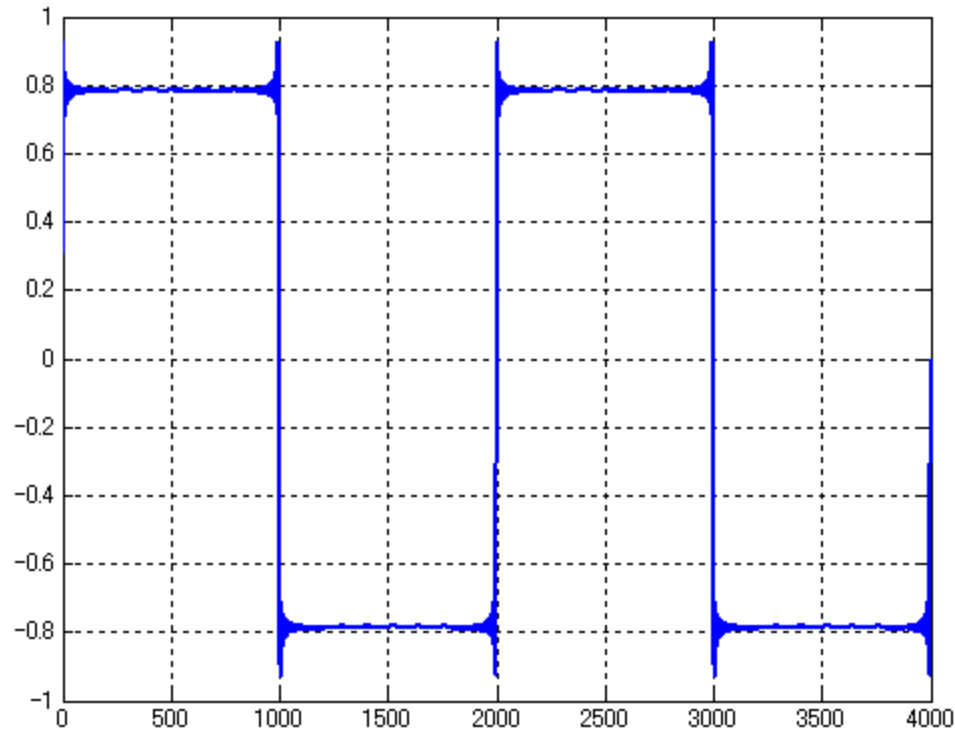
基本波+3次高調波+5次高調波

矩形波のフーリエ級数加算



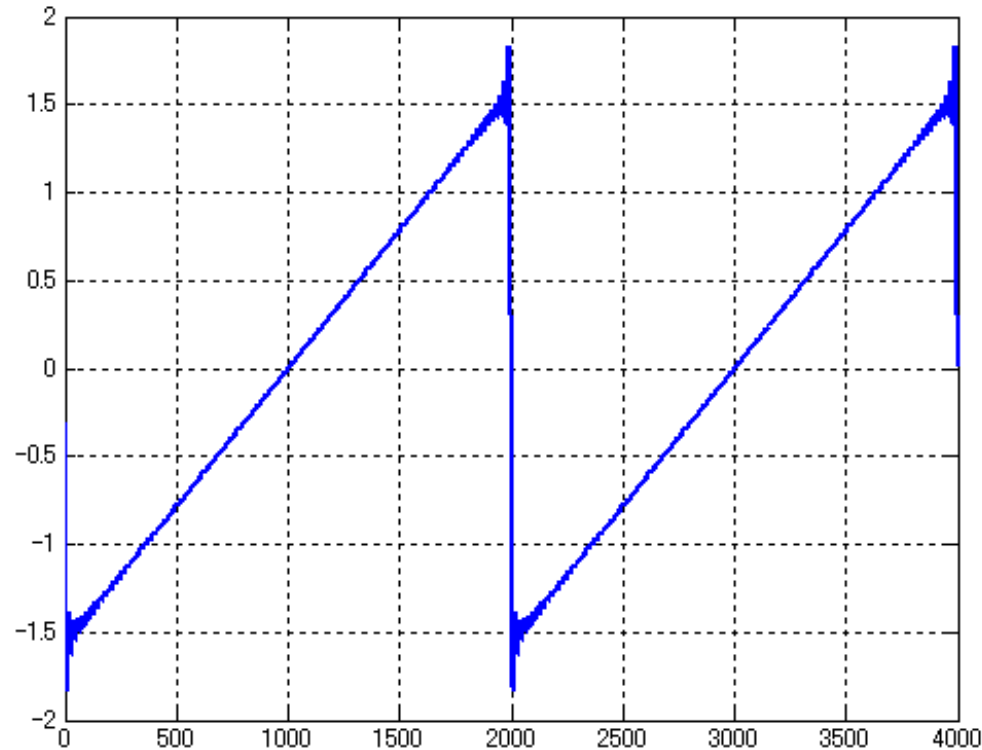
基本波+3次高調波+5次高調波+7次高調波+9次高調波

矩形波のフーリエ級数加算



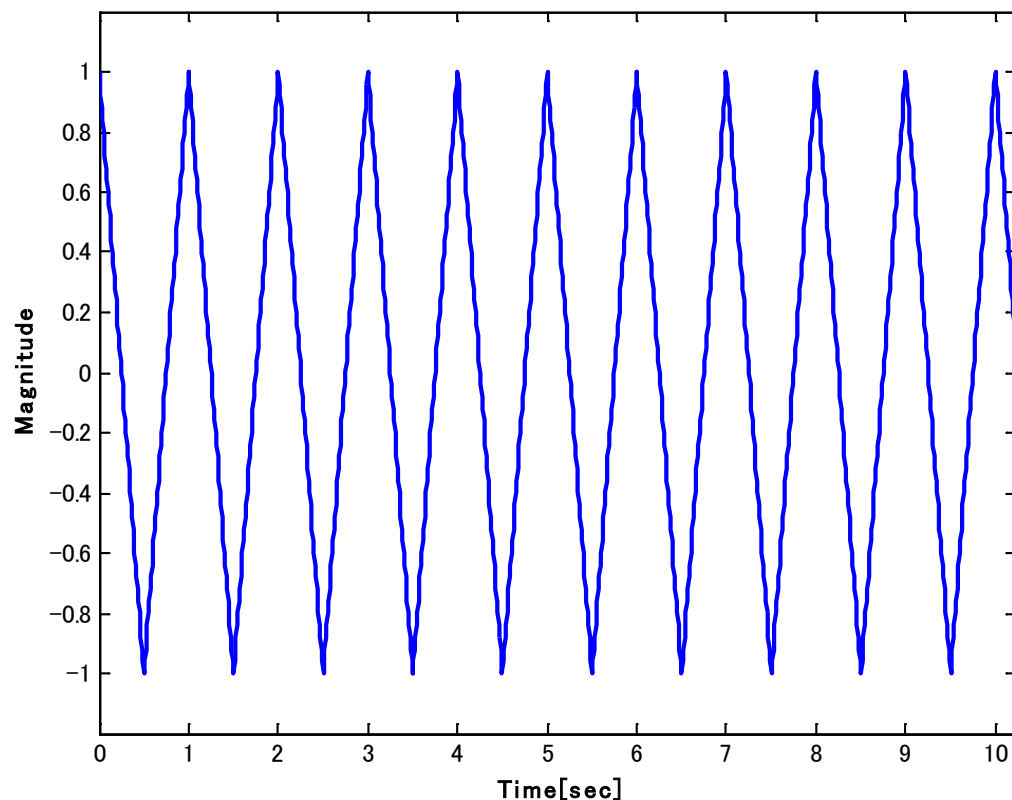
基本波+3次高調波+5次高調波+... +99次高調波
($\sin\omega t + \sin 3\omega t/3 + \sin 5\omega t/5 + \dots \sin 99\omega t/99$)

ランプ波のフーリエ級数加算



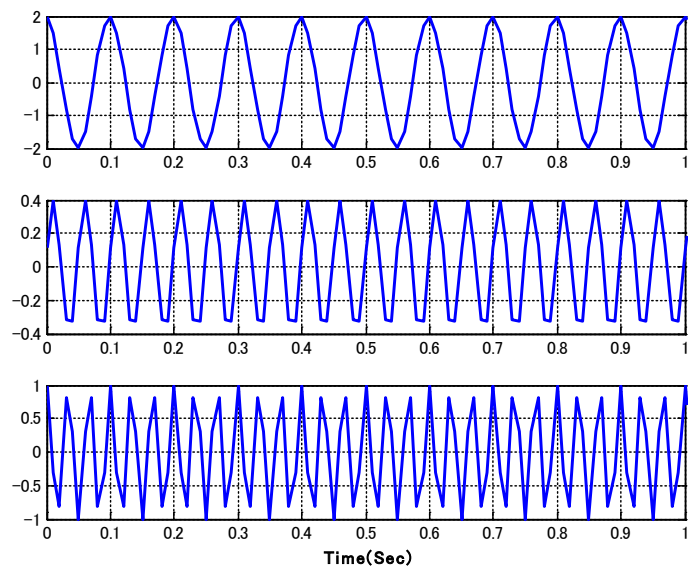
- (基本波+2次高調波+3次高調波+... +100次高調波)
- ($\sin \omega t + \sin 2 \omega t + \sin 3 \omega t + \cdots \sin 100 \omega t$)

三角波のフーリエ級数加算

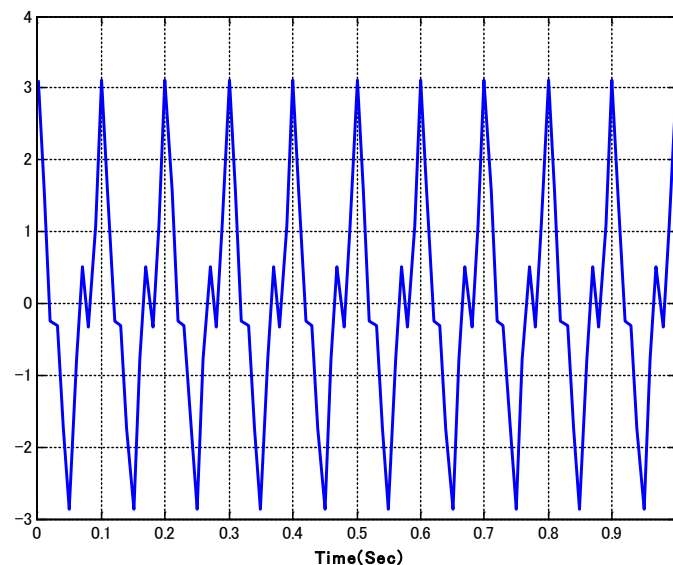


$$\begin{aligned} y(t) &= \frac{8}{\pi^2} \left(\cos \omega t + \frac{\cos 3\omega t}{9} + \frac{\cos 5\omega t}{25} + \dots + \frac{\cos((2n-1)\omega t)}{(2n-1)^2} \right) \\ &= \frac{8}{\pi^2} \sum_{n=1}^{\infty} \frac{\cos((2n-1)\omega t)}{(2n-1)^2} \end{aligned}$$

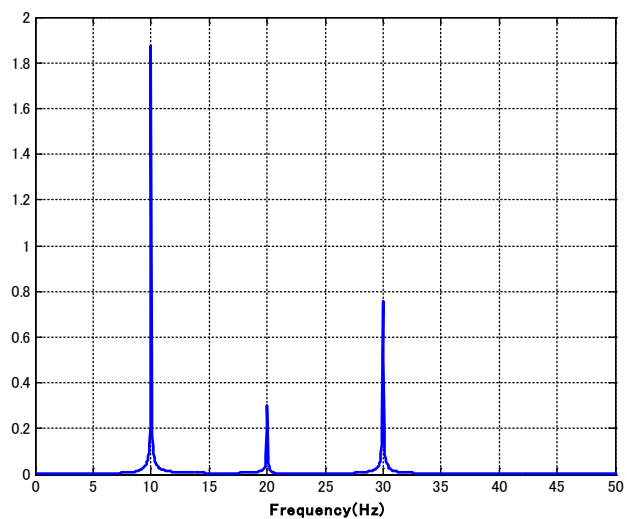
フーリエ変換



Σ
Time Domain

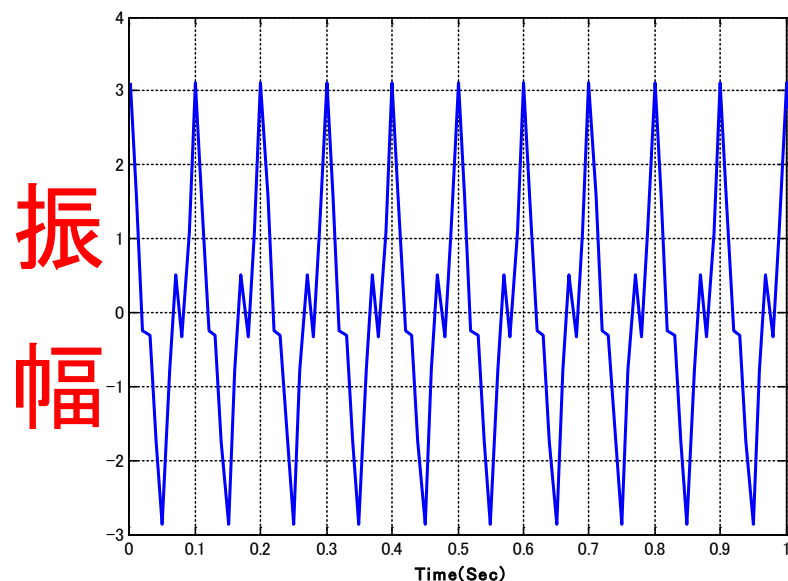


Fourier
Transform



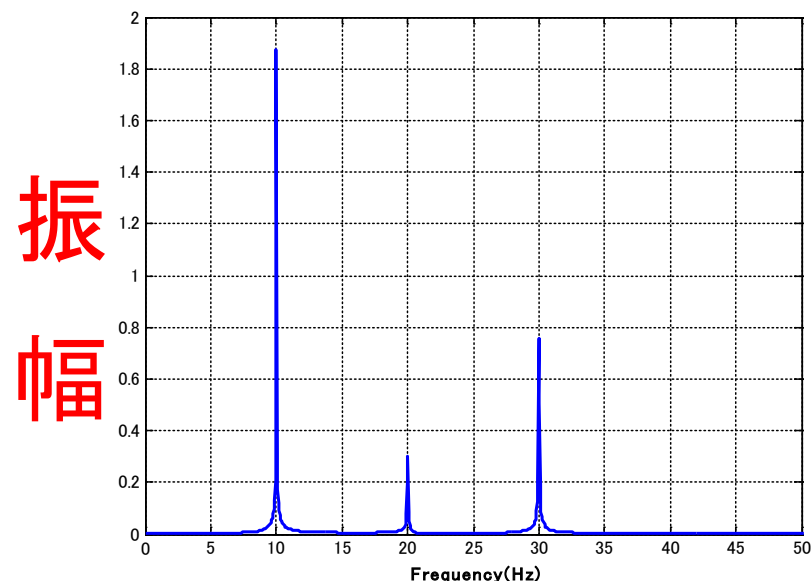
時間領域⇒周波数領域への変換

FFT前



時 間

FFT後



周波数

直交関数の内積(2信号の相関)

- 複雑な信号から、複数の基本波形に分解できるのは、なぜ?
- 理由は直交関数である。
- 直交関数条件 Sin Cosが知られている!
 ➡ 周期信号を積分 $\Rightarrow$ 0 or 定数になる

$$\int_0^T b_i(t)b_j(t)dt \Rightarrow \begin{cases} 0 & \text{while } i \neq j \\ \text{有限値} & \text{while } i = j \end{cases}$$

sin, cos関数の直交性

$$\omega = 2\pi / T$$

$$\int_0^T 1 \cdot \sin(n\omega t) dt = 0$$

$$\int_0^T 1 \cdot \cos(n\omega t) dt = 0$$

$$\int_0^T \cos(n\omega t) \cdot \sin(m\omega t) dt = 0 \quad m \neq n$$

$$\int_0^T \sin(n\omega t) \cdot \cos(m\omega t) dt = 0 \quad m \neq n$$

$$\int_0^T \cos^2(n\omega t) dt = T/2$$

$$\int_0^T 1^2 dt = T$$

- sin, cos 波形

微分⇒

位相ずれるが、

波形は変わらない

数学的に扱いやすい

- ハードウェア実現が簡単

正弦波と余弦波はフーリエ変換とラプラス変換の基本である

FFT(Fast Fourier Transform)

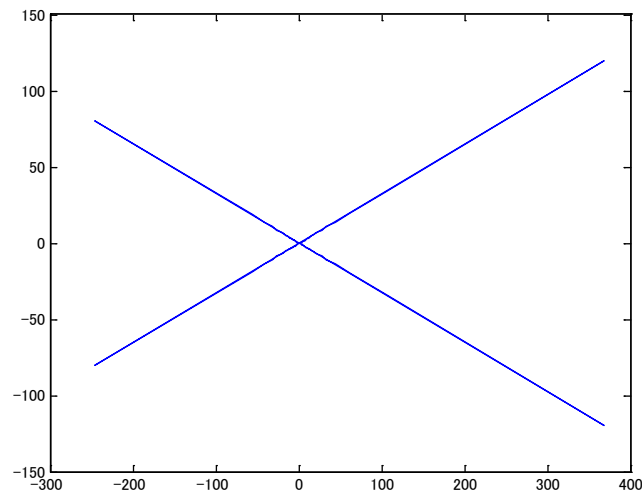
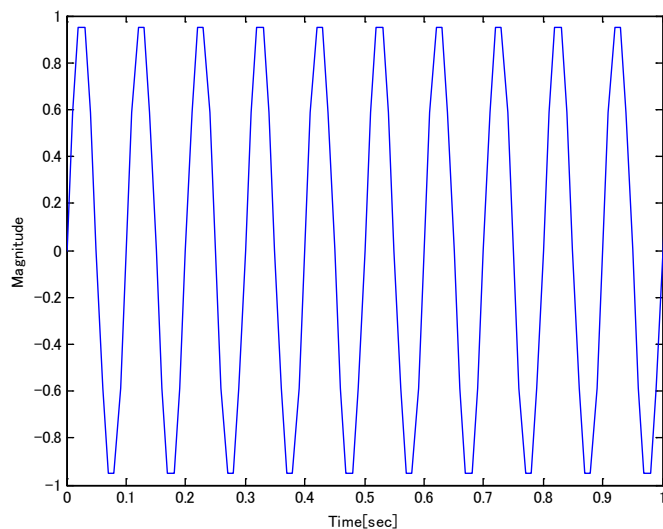
- 複雑な信号に対し、FFTを行うと

→ 信号含む周波数成分
とその大きさが分かる

注意: データの点数は 2^N が望ましい
FFTを行うだけでなく、**abs値**をとる.

fft関数と時系列データの関係

```
1 close all; clear;  
2 n=1024;dt=0.01;t=(1:n)*dt-dt;  
3 Fin=10; %%%% [Hz]  
4 data = sin(2*pi*Fin*t);  
5 figure; plot(t,data);  
6 xlabel('Time[sec]');ylabel('Magnitude');xlim([0 1]);  
7 figure; plot(fft(data));
```



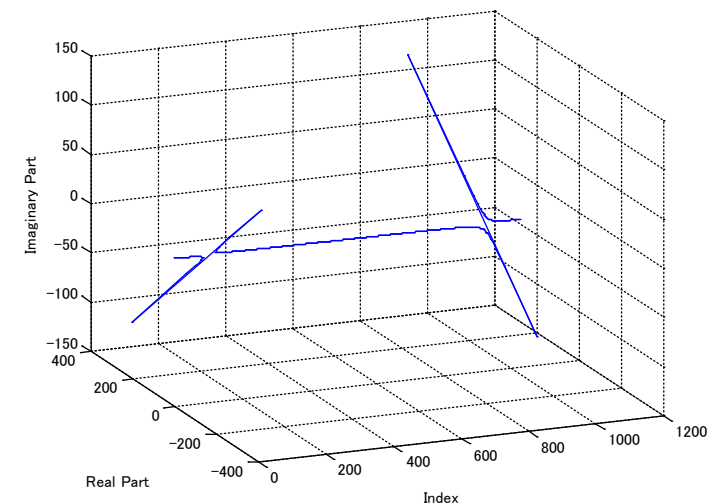
fft関数の実行結果

- `fft(data)`の結果: 複素数
- `plot(fft(data))`: 複素数平面(X:Re, Y: Im)

```

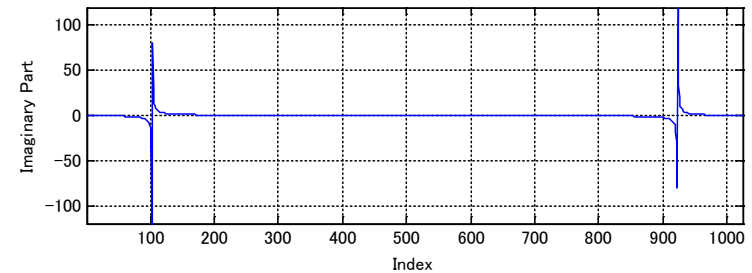
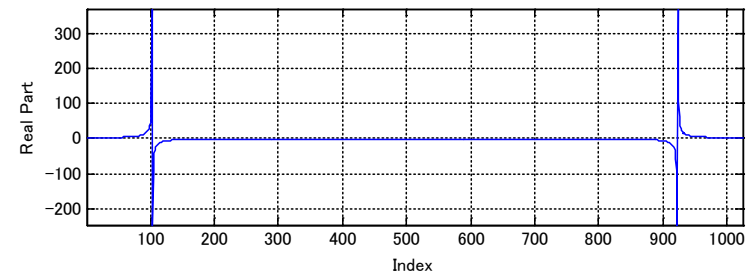
1 close all; clear;
2 n=1024; dt=0.01; t=(1:n)*dt-dt;
3 Fin=10; %%%% [Hz]
4 data = sin(2*pi*Fin*t);
5 figure; plot(t, data);
6 xlabel('Time[sec]'); ylabel('Magnitude'); xlim([0 1]);
7 figure; plot(fft(data));
8 figure; plot3(1:length(data), real(fft(data)), imag(fft(data)));
9 grid on; xlabel('Index');
10 ylabel('Real Part'); zlabel('Imaginary Part');

```



fft実行結果(Real, Imaginary)

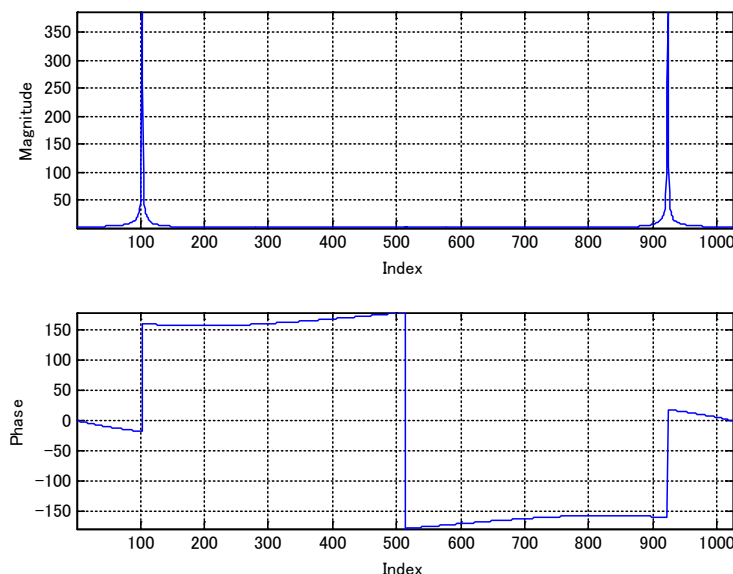
```
13 figure;  
14 subplot(211);  
15 plot(real(fft(data)));  
16 axis tight; grid on;  
17 xlabel('Index'); ylabel('Real Part');  
18 subplot(212);  
19 plot(imag(fft(data)));  
20 axis tight; grid on;  
21 xlabel('Index'); ylabel('Imaginary Part');
```



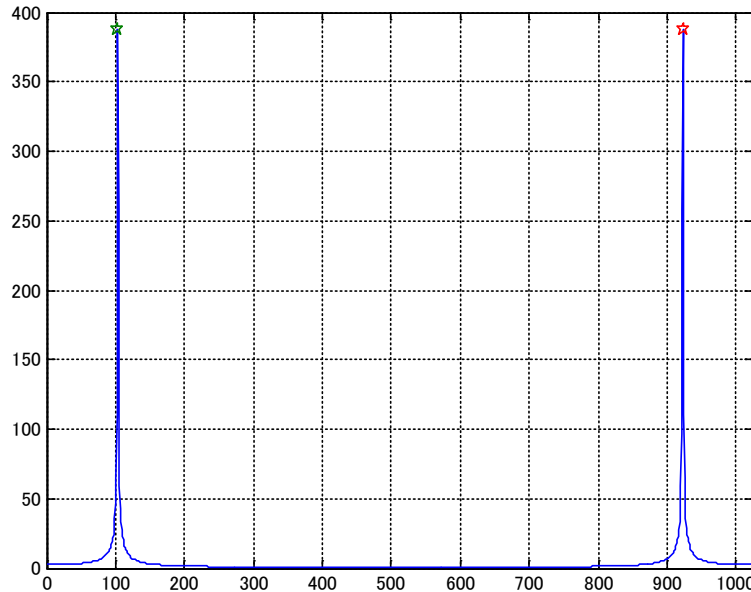
fft実行結果(Magnitude, Phase)

- fftの結果: 複素数 (大きさと位相で表す)

```
エディタ - C:\Users\san\Documents\MATLAB\MATLAB_Handbook\III\ch3\Waveform_plot...
ファイル(F) 編集(E) テキスト(T) 移動(G) セル(C) ツール(O) デバッグ(B) デスクトップ(D)
- 1 + ÷ 1.1 × % % % % %
23 figure;
24 subplot(211);
25 plot(abs(fft(data)));
26 axis tight; grid on;
27 xlabel(' Index '); ylabel(' Magnitude ');
28 subplot(212);
29 plot(angle(fft(data))*180/pi);
30 axis tight; grid on;
31 xlabel(' Index '); ylabel(' Phase ');
```



FFT処理結果の解析(Freq)



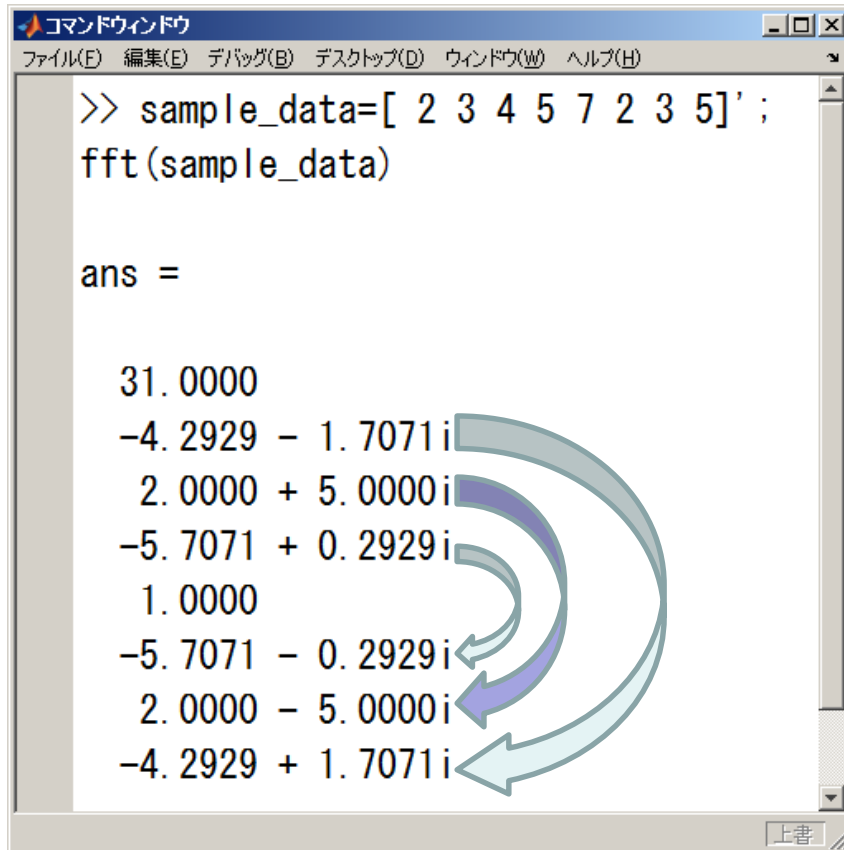
Data: 1024 pts
Peak: 103番 と 923番

$F_{in} = ??$

$$\frac{F_{in}}{F_s} = \frac{103}{1024} \quad \& \quad \frac{F_{in}}{F_s} = \frac{923}{1024}$$

$F_s = ??$

FFT処理結果は対称?



コマンドウィンドウ

```
ファイル(F) 編集(E) デバッグ(B) デスクトップ(D) ウィンドウ(W) ヘルプ(H)
```

```
>> sample_data=[ 2 3 4 5 7 2 3 5]';  
fft(sample_data)
```

ans =

31.0000	
-4.2929 - 1.7071i	↗
2.0000 + 5.0000i	↖
-5.7071 + 0.2929i	↘
1.0000	
-5.7071 - 0.2929i	↙
2.0000 - 5.0000i	↗
-4.2929 + 1.7071i	↖

上書

- 一番目の値はDC成分
 $\text{sum}(\text{sample_data})$
DC成分を無くすためには?
- 8個のデータは
4番目のデータではなく、
5番目のデータを中心に
対称となる.

SNRを求める計算の注意事項

- F_{in} , F_s の関係は要注意.
- FFTの結果に基づき, Powerを計算する.
- SNRの計算結果はdBで表示する

課題

- 8bit & 10bit AD変換のプログラムを作成
- Sin(Ramp)波(振幅-1~1)の波形とその入力信号をAD変換後の波形を示す
- Sin波のAD変換後のデジタルコードに対し、FFT処理を行い、パワースペクトルを示す.
- AD変換のSN比を計算する.

レポートの提出について

- 提出内容: ファイル **g1x81xxx.zip** (下記内容を含む)
 1. AD変換プログラム
 2. FFT及びSNR計算プログラム
プログラム言語は自由, ソースコードを提出してください
 3. PPTファイル(**ファイル名g1x81xxx.pptx**)
AD変換結果とFFT結果のグラフ, SNR値が必須
実行結果のグラフに説明文を加え,
PPTファイルにまとめてください.
PPT表紙に氏名と学生記番号を明記すること.
- 提出期限: **H29/07/20(木) 12:00まで**
- 提出先: hsan@tcu.ac.jp