データサイエンス特論
Data Science

スペクトル分析

1. 周波数領域で信号を見る
2. フーリエ変換
3. 現実の問題
4. DFT
5. 窓関数
6. ゼロ埋め込み
7. 不規則信号のスペクトル分析
8. PSDの計算手順

(C) 創造技術コース　橋本洋志／大久保友幸
{hashimoto, ohkubo-tomoyuki}@aiit.ac.jp
http://hhlab.org/

不規則性の無い信号（決定的信号とも言う）

不規則（random）信号とは、将来の値を決定できない信号をいう。逆に決定的信号（deterministic signal）とは何か？

例えば、tを時間として、y = at+b, y = sin(t)など関数で表わされる信号は、$-\infty < t < \infty$で、tを与えればyは一意に決まる。このように決定できる信号を言う。

一方、tを与えてもyが一意に決まらない信号を不規則信号という。これには、システム内部が変動して、システム出力そのものが不規則の場合と、システム出力は決定的であるが、ノイズが重畳して、その観測値が不規則の場合がある。前者は外乱、後者は観測雑音というように言い分ける場合がある。

いずれにしても、過去、現在、および将来の信号（観測値と同じ意味でここでは用いている）が「真値でない（決定できない）」ので、何らかの方法で推定または予測を行うということを本章では考える。

周波数領域で信号を見る

周波数とは

- 1秒あたりの波の数, f [Hz]
- 例:高音は周波数が高い, 低音は周波数が低い

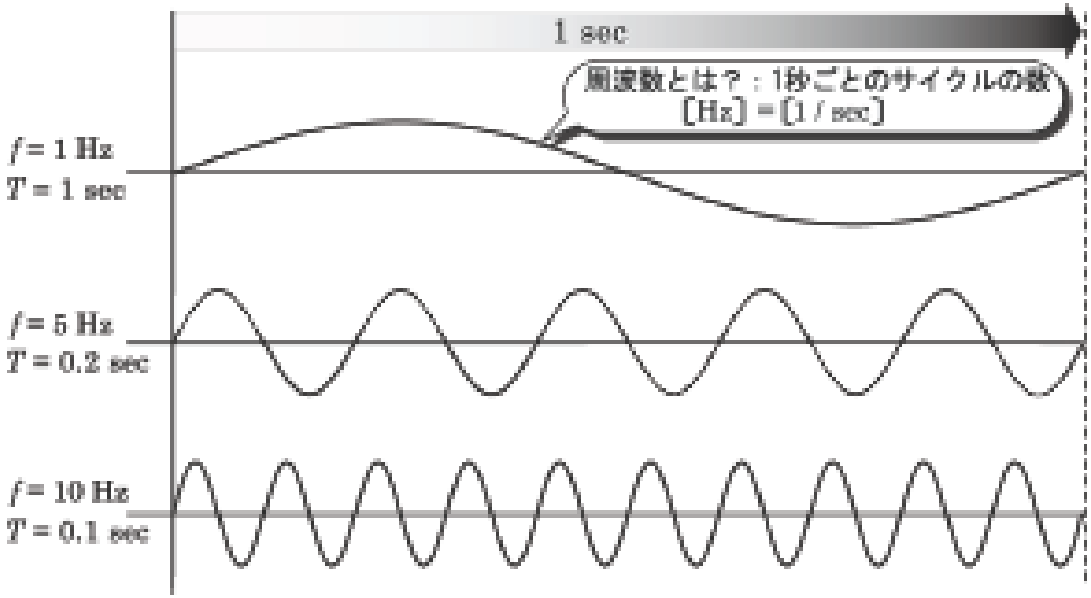


図 9.1: 周波数と周期

表 9.1: 音名と対応する周波数，ラ音は国際規格より 440 Hz，その他の音の周波数は場合により異なる

音階	ド	レ	ミ	ファ	ソ	ラ	シ	ド
周波数〔Hz〕	262	294	330	349	392	440	494	523

PyAudio_DoReMi

周波数領域で信号を見る

- これまでの信号の見方は，横軸が時間で時間変動した。これを時間領域で信号を扱うと言われる。
- これを横軸が周波数（または角周波数）で見ることを周波数領域で信号を扱うと言う。
- 周波数とは
  - 1秒あたりの波の数，f [Hz]
  - 例：高音は周波数が高い，低音は周波数が低い
  - 角周波数　$\omega = 2\pi f$ [rad/s]，　半径1の円の円周は2π，周波数fのとき，1秒間にω回だけ円を回る
  - ωを用いるのは三角関数（sinなど）の引数がradianのため。数学では[rad/s]，現場は[Hz]
- サイン関数（sine function）で信号x(t)を表す

$$x(t) = A \sin(\omega t + \phi)$$

この特徴量は，振幅と角周波数（周波数）。これを用いれば，次の信号は

$$x(t) = A_1 \sin(\omega_1 t + \phi_1) + A_2 \sin(\omega_2 t + \phi_2)$$

角周波数成分がω_1とω_2の信号である，という言い方ができる。

スペクトル分析

概要

- 太陽の光はプリズムを通して，幾つもの色の光に分散（分解）され，これにより元の太陽光の性質を知ることができる。Isaac Newton（数学・物理・天文学者，英国）が1666年にプリズムを用いて太陽光を分解したときに，用語spectrumを導入した。
- 光の色は周波数で定まる。
- このように，ある信号（太陽光）に対して，スペクトル分析を行うことで（プリズムを通すようなもの），信号にどのような周波数成分が含まれているかを分析する。

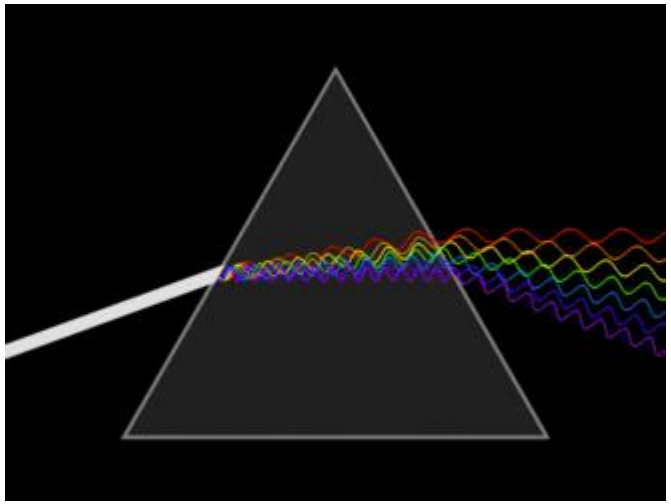
用途

- 振動分析（解析）
  - 乗り心地，建物・構造物強度分析，車や機体の構造に関わる振動解析
- 故障診断
  - 一定回転で回るモータやエンジンの異常振動の検出
- 制御における観測雑音，外乱抑制
  - 制御におけるセンサ出力の観測雑音や外乱の抑制を行う
- 定性分析
  - 元素，物質，構造の特定

英語の使い方:
spectrum （**可算**）**名詞**，**単数形** （ー＞**複数形** spectra）
spectrum analysis
power spectrum estimation
power spectrum
spectrum analysis

英語の使い方:
spectral **形容詞**
spectral estimation
FFT spectral estimation
ESD: energy spectral density
PSD: power spectral density

Prism



https://en.wikipedia.org/wiki/Prism

虹の色は何色？

フーリエ変換

Joseph Fourier（1768-1830, 仏，数学・物理学者）を由来とする。

著書「熱の解析的理論」において，「任意の関数は，三角関数の級数で表すことができる」と主張した。当時，この証明は不十分であったが，後に多くの研究者により厳密化が図られた。

フーリエ変換とフーリエ逆変換

【フーリエ変換とフーリエ逆変換】（角周波数 ω 表現） $x(t)$ の可積分の条件

$$\int_{-\infty}^{\infty} |x(t)|\, dt < \infty \tag{6.13}$$

を満たせば，任意の ω に対して

$$X(\omega) = \int_{-\infty}^{\infty} x(t) \exp(-j\omega t)\, dt \tag{6.14}$$

が存在し，この $X(\omega)$ を $x(t)$ の**フーリエ変換**（Fourier transform）という。このとき，$\exp(-j\omega t)$ をフーリエ変換の**核**（kernel）という。さらに，次をフーリエ逆変換（inverse Fourier transform）という。

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) \exp(j\omega t)\, d\omega \tag{6.15}$$

可積分条件は (6.14) 式の積分が行えることを保証するもので，$\lim_{t \to \pm\infty} x(t) = 0$

j は虚数単位

スペクトル(spectrum)

スペクトルとは

- 信号を周波数ごとの成分に分解し，周波数に対するその分布をいう
- 一般に，横軸をω（またはf），縦軸をX(ω)の何らかの強さ，としたもの

$$X(f) = \mathrm{Re}\,(X(f)) + j\mathrm{Im}\,(X(f)) = |X(f)|\angle X(f) \tag{6.20}$$

しばらくの間，ωとf は同じものと見てください。

ここに，$|X(f)| = \sqrt{\mathrm{Re}(X(f))^2 + \mathrm{Im}\,(X(f))^2}$，$\angle$ は偏角（argument）を表し，次式で定義される。

$$\angle X(f) = \arctan \frac{\mathrm{Im}X(f)}{\mathrm{Re}X(f)} \tag{6.21}$$

(6.20) 式において，$|X(f)|$ は振幅を表すことから，$|X(f)|$ の分布を**振幅スペクトル**（amplitude spectrum），$\angle X(f)$ を**位相スペクトル**（phase spectrum）という。

被積分関数である $|X(f)|^2$ は単位周波数あたりのエネルギーを表す

これより，$|X(f)|^2$ の f に対する分布を**エネルギースペクトル密度**（ESD; energy spectrum density），

スペクトル(spectrum)

ところで，実際の信号（持続する信号など，ただし，周期信号はデルタ関数を導入してフーリエ変換できるので除く）は，可積分条件を満足しないことが多い。このような信号の全エネルギーは無限大であるから，上記のエネルギースペクトル密度は定義できない。そこで，エネルギースペクトル密度の単位時間時間あたりの平均値を考える。すなわち，

$$P(f) = \lim_{T\to\infty}\left[\frac{1}{T}|X(f)|^2\right] \tag{6.23}$$

ならば定義できる。ここで，物理の世界では単位時間当たりのエネルギーをパワーと定義していることから，$P(f)$ を**パワースペクトル密度**（PSD; power spectrum density）という。なお，$x(t)$ が不規則信号の場合には，さらに，期待値操作が導入された PSD が定義され，これについては別の章で説明する。

例：三角関数のスペクトル

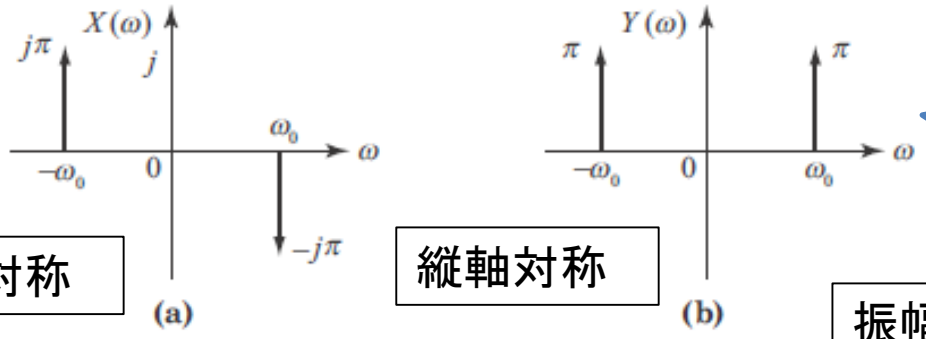
【例題 4】（周期信号のフーリエ変換）　$\sin\omega_0$ と $\cos\omega_0$ のフーリエ変換を求めよう。

アドバンス

$$x(t) = \sin\omega_0 t \Rightarrow X(\omega) = F[x(t)] = \frac{2\pi}{2j}\{\delta(\omega-\omega_0)-\delta(\omega+\omega_0)\} = -j\pi\{\delta(\omega-\omega_0)-\delta(\omega+\omega_0)\}$$

$$y(t) = \cos\omega_0 t \Rightarrow Y(\omega) = F[y(t)] = \pi\{\delta(\omega-\omega_0)+\delta(\omega+\omega_0)\}$$

全員　理解してほしい



原点対称

縦軸対称

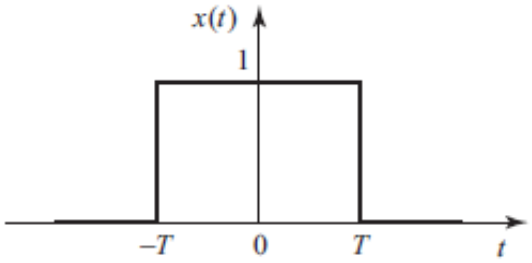
線で表されたスペクトルより，**線スペクトル**(line spectrum)という。
物理の世界では，輝線スペクトルということもある。

図 6.8　$\sin\omega_0 t$, $\cos\omega_0 t$ のフーリエ変換

振幅スペクトル
エネルギースペクトル密度
で考えると？

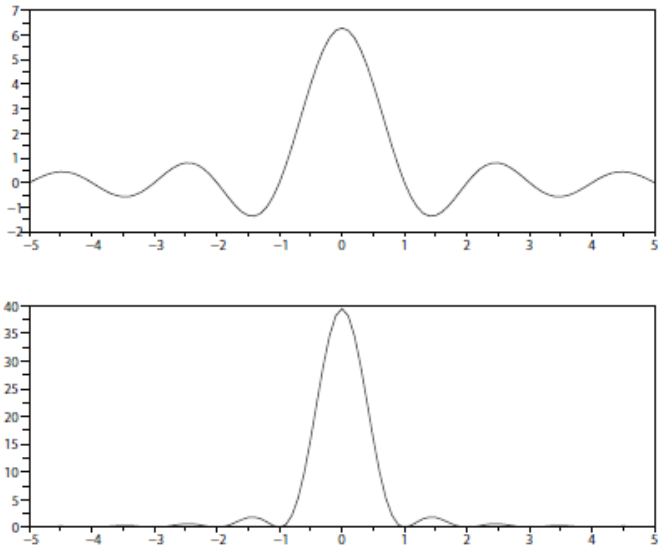
例：矩形波のスペクトル密度

【例題 3】（非周期信号のフーリエ変換）　図 6.6 に示す孤立した矩形波のフーリエ変換 $X(\omega)$ とエネルギースペクトルを求めよう。



$$|X(\omega)|^2 = \left| \frac{2\sin \omega T}{\omega} \right|^2$$

図 6.6　孤立した矩形波



連続スペクトルという

図 6.7　矩形波の $X(\omega)$ と $|X(\omega)|^2$ $(T = \pi)$

サンプリングと有限長の問題

1. **サンプリング周波数とエイリアシング**
   - サンプリング定理
2. **有限長波形のための打ち切りと漏れ**

サンプリング定理　（重要！）

【サンプリング定理】（sampling theorem，**標本化定理**ともいう）　実関数 $x(t)$ のフーリエ変換 $X(\omega)$ が存在し，$|f| \leq f_c$ 以外の周波数成分を含まないとき，$x(t)$ は $t = k/2f_c$（k は整数）の離散的な時点における信号の標本値から再現できる。$1/2f_c$ を**ナイキスト間隔**（Nyquist interval），$2f_c$ を**ナイキスト周波数**（Nyquist frequency）という。

エイリアシング（aliasing [*6]，**折り返し雑音**（folding noise）ともいう）とは，連続信号がサンプリングされて復元されたとき，歪みが生じて別の波形のように見えることである。例えば，**図 6.10** のように周波数 $f = 4$ Hz （周期 0.25 Hz）の正弦波を $\Delta T = 0.2$ sec でサンプリングすると，周期 1 sec の正弦波に見える。

動的システムのサンプリングとは異なる

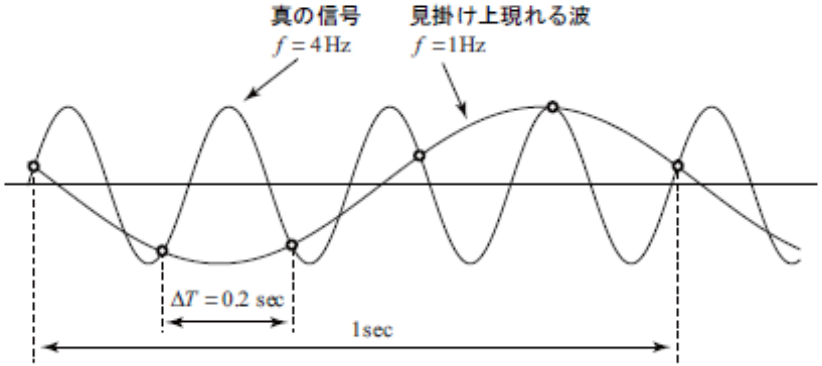


図 6.10　正弦波のサンプリングによるエイリアシング

真の信号を復元しようとするならば，サンプリング定理より，2 × 4 = 8 Hz 以上の周波数でサンプリングを行わなければならない。サンプリング周波数がこれより遅いと，偽の信号が見えることになる。

エイリアシング

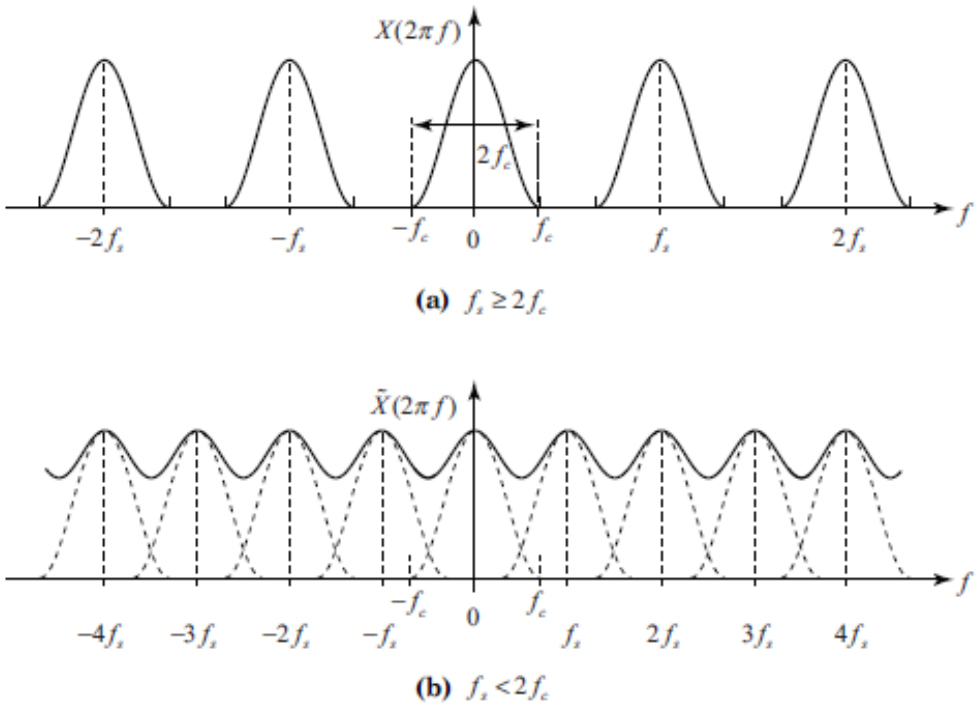


図 6.11　サンプリング周波数 f_s とエイリアシングの関係

いま，真の $X(2\pi f)$ が存在する周波数帯域を $[-f_c, f_c]$ とする。このとき，サンプリング周波数 $f_s = 1/\Delta T$ が $f_s \geq 2f_c$ のとき，(6.52) 式の右辺は図 6.11(a) のように，(6.52) 式の右辺第 1 項と第 2，3 項が互いに重ならず，もとの $X(2\pi f)$ を区別できることを示している。一方，$f_s < 2f_c$ のとき，図 6.11(b) のようになり，(6.52) 式の右辺第 1 項と第 2，3 項が重なりあったもの（図の太線）であるという偽の波形を見ることとなり，真の $X(2\pi f)$ を見出すことはもはや不可能である。

エイリアシングを起こさない

1. もとの波形が有する最高周波数fc の2 倍以上の速さのサンプリング周波数fs を適用する
2. このfs を達成できないならば，fs/2 を遮断域とするローパスフィルタを用いて，もとの波形をフィルタリングする
3. サンプリングしたx_n に，適当な窓関数を乗じる

窓関数は後述

有限長波形のための打ち切りと漏れ

矩形窓（rectangular window）$w(t)$ が $x(t)$ に作用

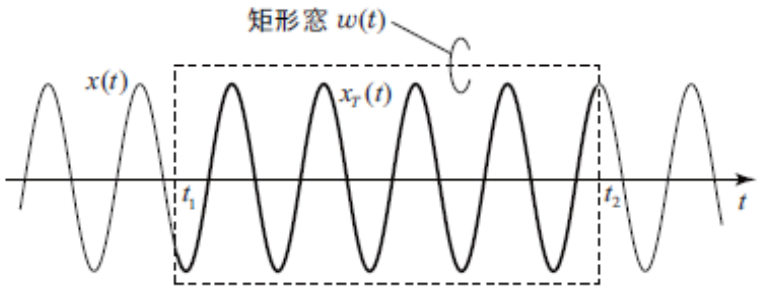


図 6.12　有限長波形

矩形波のフーリエ変換

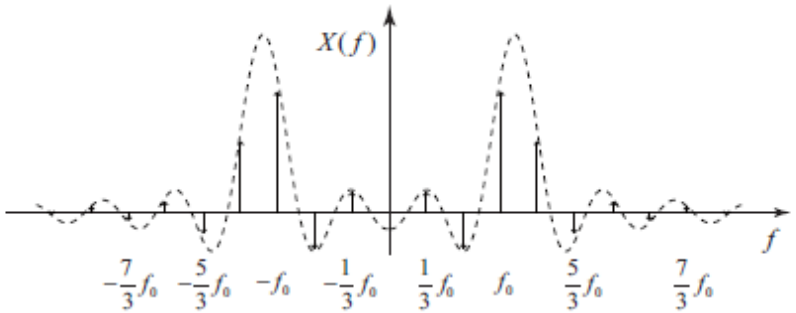


図 6.13　漏れのイメージ図（メインローブとサイドローブ）

離散フーリエ変換

【離散フーリエ変換（DFT）と逆離散フーリエ変換（IDFT）】

サンプリング時間を Δt とし，観測するデータ数を N とする。このとき，記録長 T と，周波数間隔（周波数分解能ともいう）Δf は次となる。

$$T = N\Delta t, \quad \Delta f = \frac{1}{T} = \frac{1}{N\Delta t} \tag{4.25}$$

この条件の下，DFT は次式で定義される。

$$X_k = \Delta t \sum_{n=0}^{N-1} x_n \exp\left(-j2\pi kn/N\right) \tag{4.26}$$

もしも，初めから離散データ系列を与えられる場合には，$\Delta t = 1$ とおいて計算すればよい。また，IDFT は次式で定義される。

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \exp\left(j2\pi nk/N\right) \tag{4.27}$$

フーリエ変換の場合と同様に，$|X_k|$ を振幅スペクトル，$|X_k|^2$ をエネルギースペクトル密度とし，エネルギースペクトル密度を

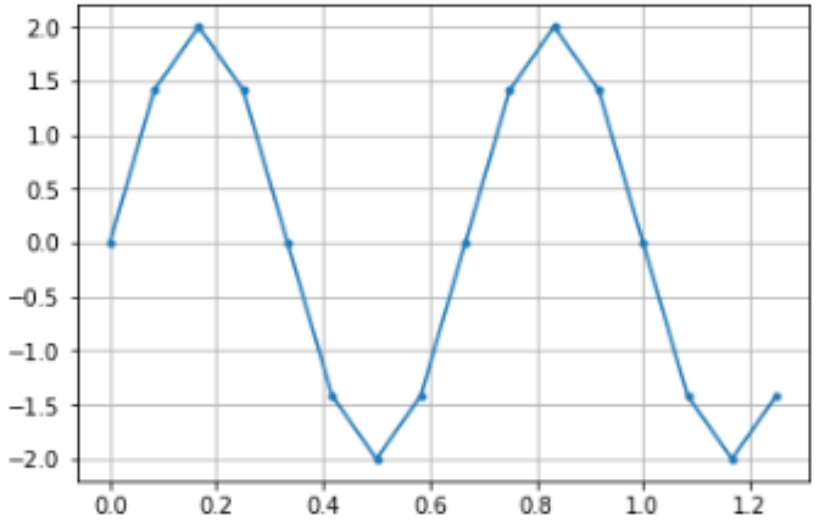
$$\mathrm{PSD}_x(k) = \frac{|X_k|^2}{T} \tag{4.28}$$

A sin(ωt)のDFT

- 周期の整数倍を観測 （打ち切り影響無し）

DFT_Sine

```
f0 = 1.5  # 基本周波数 [Hz]
T = 2/f0  # 観測時間[s]．　分子が観測する周期の数を表す
N = 16    # サンプル数
dt = T/N  # サンプリング時間
df = 1/T  # 周波数分解能
A = 2.0
t = np.linspace(0, N-1, N)*dt  # 時間軸
x = A*np.sin(2*np.pi*f0*t)       # 観測信号

plt.plot(t, x, marker='.')
plt.grid()
```



A sin(ωt)のDFT

scipy.fftpack.fft https://docs.scipy.org/doc/scipy/reference/generated/scipy.fftpack.fft.html
実部：縦軸対称，　虚部：原点対称
サイン波の離散フーリエ変換：虚部のみに値が生じ，原点（中心）対称となる。

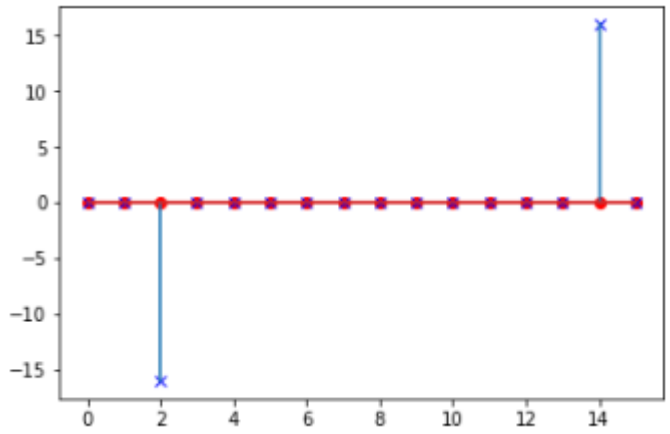
周波数分解能 $\Delta f = 1/T = 1/(N\Delta t)$

$$X(m\Delta f) = \mathrm{DFT}(A\ \sin(2\pi f_0\ \Delta t\ n)) = \frac{NA}{2}(-j\delta(m\Delta f - f_0) + j\delta((m-N)\Delta f + f_0))$$
$$n, m = 1, \cdots, N-1$$

これより，DFTの振幅は (NA)/2 = (162)/2=16, m=2, N-2 で生じる

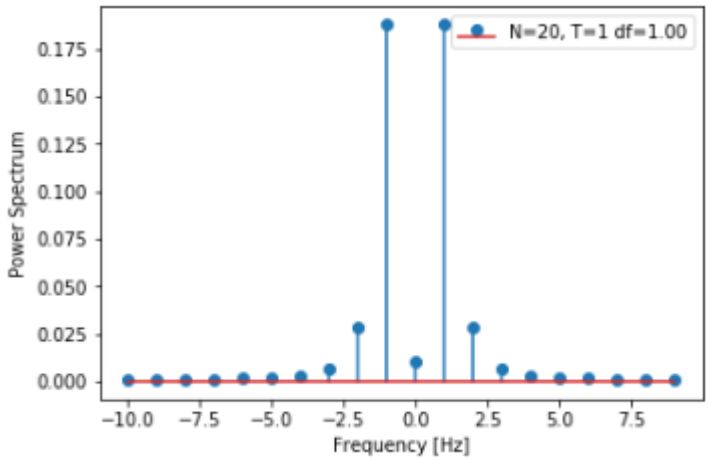
```
1  FTx = scipy.fftpack.fft(x)
2  print('FT = ¥n',FTx)
3  plt.stem(np.real(FTx), markerfmt='ro')
4  plt.stem(np.imag(FTx), markerfmt='bx')
```

```
FT =
 [  1.37803714e-15 +0.00000000e+00j   5.25055602e-16 +2.69664993e-15j
  -7.12190901e-15 -1.60000000e+01j   9.85116443e-16 -2.12495962e-15j
   4.89858720e-16 -4.44089210e-15j  -5.39900348e-18 -1.14524218e-15j
   4.18275669e-15 -1.77635684e-15j   4.54661837e-16 +1.71693250e-15j
  -3.98319700e-16 +0.00000000e+00j   4.54661837e-16 -1.71693250e-15j
   4.18275669e-15 +1.77635684e-15j  -5.39900348e-18 +1.14524218e-15j
   4.89858720e-16 +4.44089210e-15j   9.85116443e-16 +2.12495962e-15j
  -7.12190901e-15 +1.60000000e+01j   5.25055602e-16 -2.69664993e-15j]
```

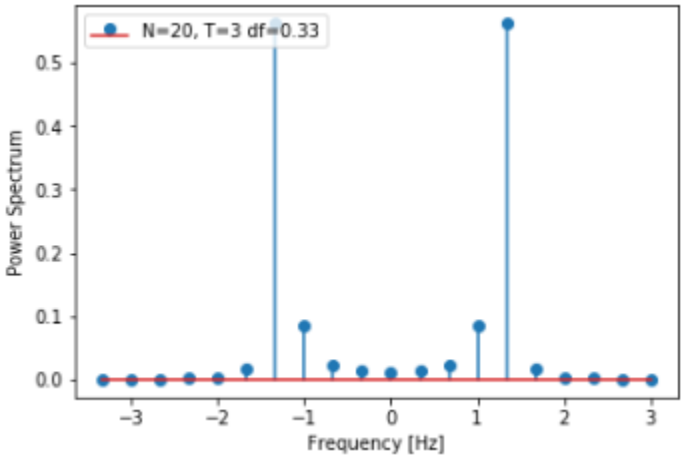


周期の非整数倍を観測

- **基本周波数 f0 = 1.25 [Hz]のサイン波とする。**
  - N:サンプル数, T:観測時間　，　これらを変える
  - サンプリング時間　　dt = T/N
  - 周波数分解能　　df = 1/T

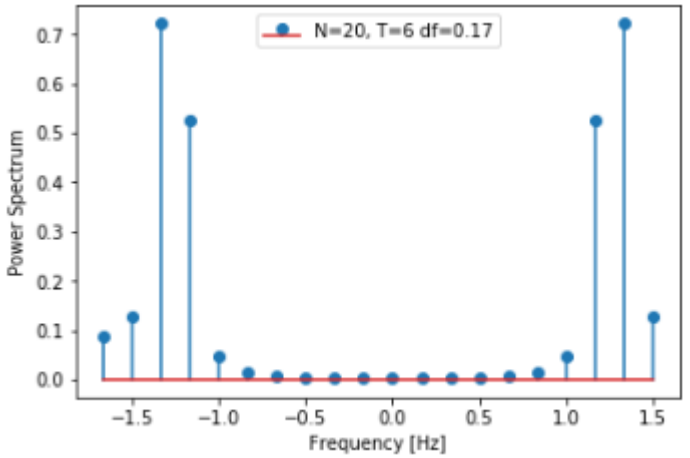



```
f0 = 1.25 # 基本周波数 [Hz]
T = 1     # 観測時間[s]
N = 20    # サンプル数
dt = T/N  # サンプリング時間
df = 1/T  # 周波数分解能
t = np.linspace(0, N-1, N)*dt # 時間軸
x = np.sin(2*np.pi*f0*t)      # 観測信号

FTx = dt*scipy.fftpack.fft(x)
esd = (np.abs(FTx)**2)
psd = esd/T
freq = np.fft.fftfreq(x.size, d=dt) # 周波数軸
```



周波数分解能 df を細かくしたい

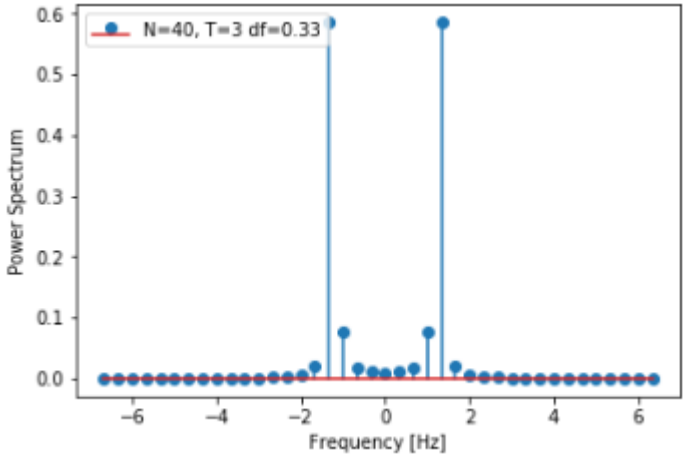
T=3 -> T=6

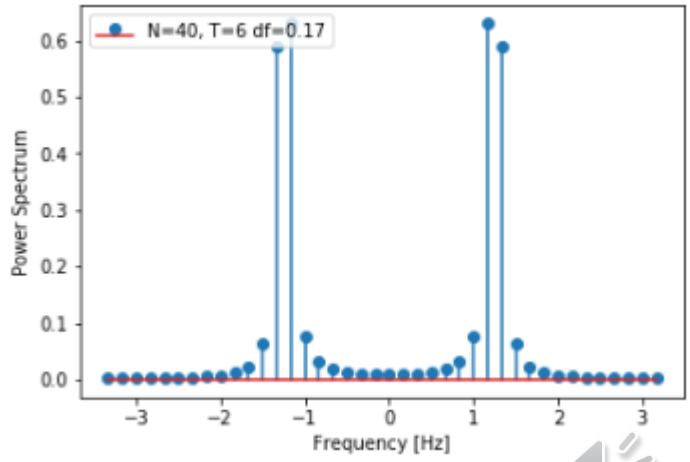


N=20
↓
N=40

N=20
↓
N=40

周波数区間を拡げたい





周波数分解能 Δfを細かくしたい －＞観測時間Tを長くする
周波数区間を広げたい －＞観測データ数Nを多くする

ゼロ埋込み

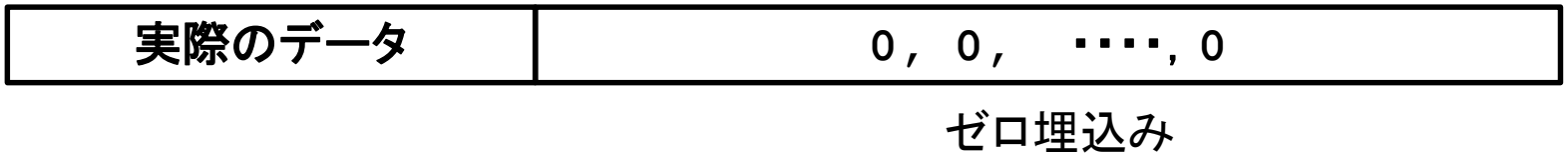
実際の観測データの後にゼロを埋め込む（zero padding）

効用

- もとの周波数成分に影響を与えない。
- 見掛け上，スペクトル分布が滑らかになる。
- 分布が滑らかになると，周波数成分の在りかがわかりやすくなる。
- データ数を2のべき乗にできるので，FFTを適用した高速演算を行える。

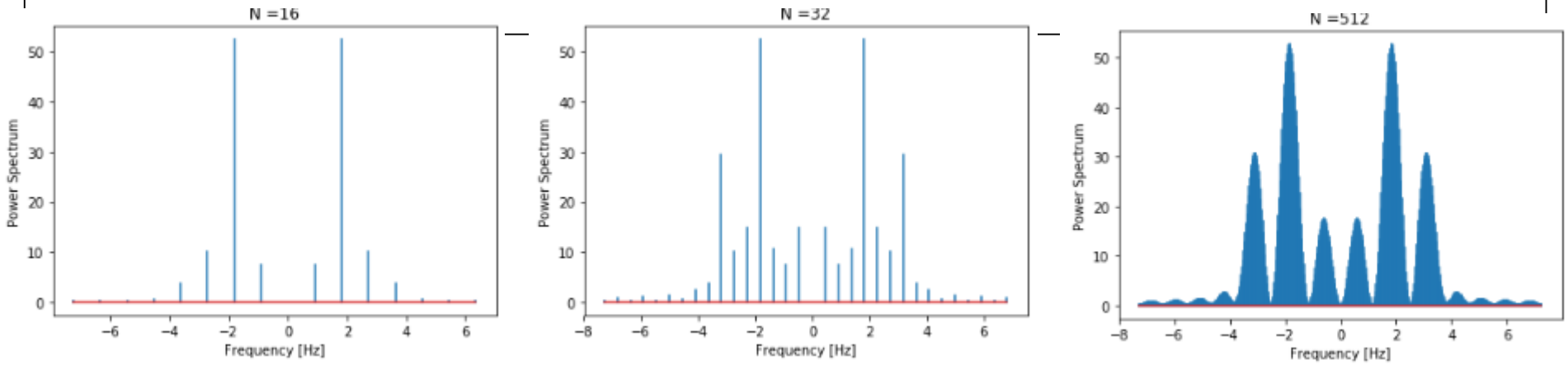
注意

- もともとの周波数分解能が向上したのではなく，周波数点間を内挿（interpolation）しているだけである。



ゼロ埋込み(zero padding)

下記のように，fftにxのデータ数（16点）よりも大きな数をnに与えると，xの最後尾N番目から(Num − 1)番目まで0が埋め込まれ，これに対するDFT計算が行われる



左より右を見て，周波数成分が3つあることがわかる。
真の周波数は 1.1, 1.7, 3.1，これと推定値がずれているのは，サンプル数が少ないためである。
実際の周波数分解能が上がるわけではない。
また，矩形窓の影響（後述）が考えられる。

```
N = 16
T = 1.1

dt = T/N  # サンプリング時間

t = np.linspace(0, N-1, N)*dt # 時間軸
x = 0.5*np.sin(2*np.pi*1.1*t)  + 1.0*np.sin(2*np.pi*1.7*t + np.pi/2) ¥
  + 0.5*np.sin(2*np.pi*3.1*t)

Num = 2**6
dft = scipy.fftpack.fft(x, n=Num
```

DFT_ZeroPadding

窓関数

有限長観測の問題点

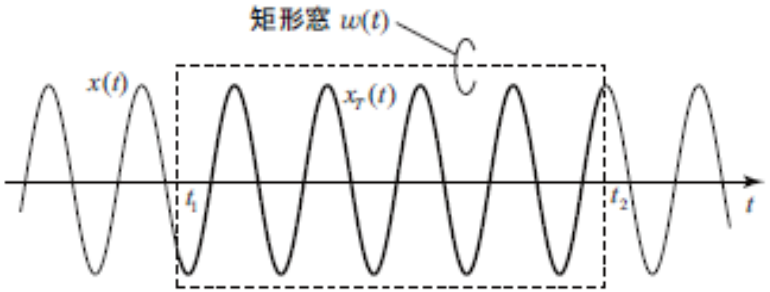


図 6.12　有限長波形

矩形窓のパワースペクトル
スペクトルの漏れ（leakage）

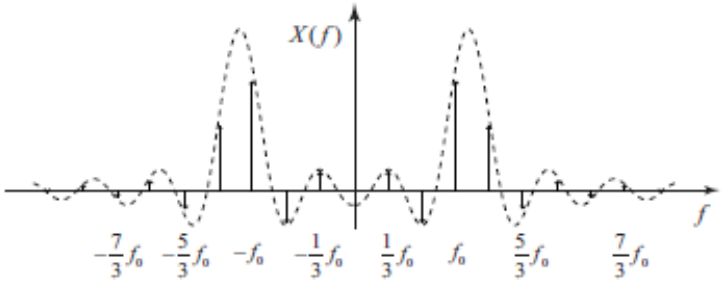


図 6.13　漏れのイメージ図（メインローブとサイドローブ）

窓関数（window function）

- 漏れを抑圧する
- scipy が提供する窓関数　https://docs.scipy.org/doc/scipy/reference/signal.html

窓関数

よく用いられるものを一部紹介

Hamming window

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right)$$

Hann window

$$w(n) = 0.5 - 0.5\cos\left(\frac{2\pi n}{N-1}\right)$$

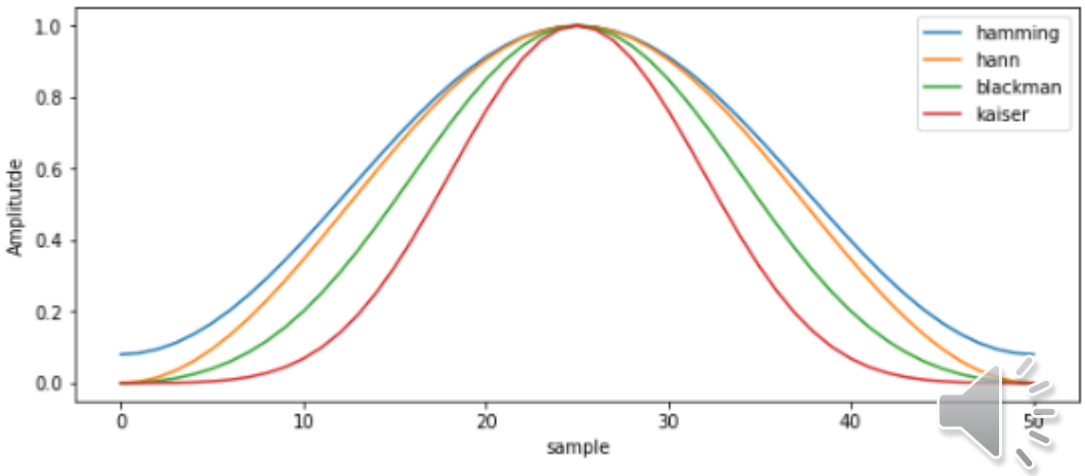
Blackman window

$$w(n) = 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right)$$

DFT_WindowFunction

Kaiser window

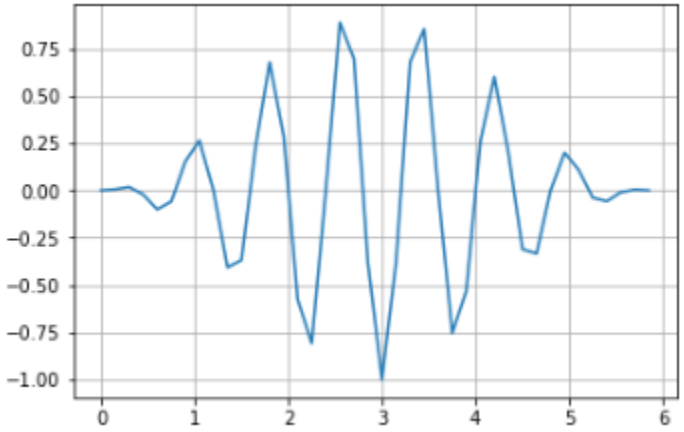
$$w(n) = \frac{I_0\left(\pi\alpha\sqrt{1-\left(\frac{2n}{N-1}-1\right)^2}\right)}{I_0(\pi\alpha)}$$



I_0は第1種の0次の変形ベッセル関数。

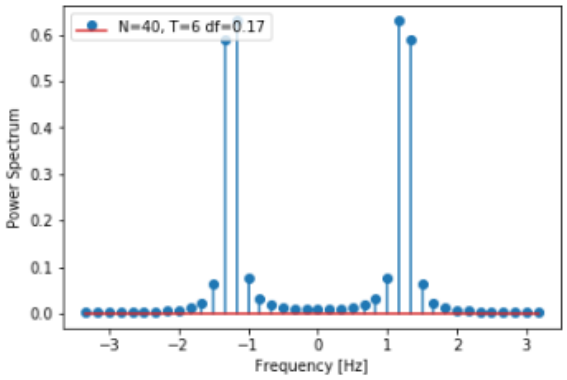
窓関数の使用例

Hann窓の使用例

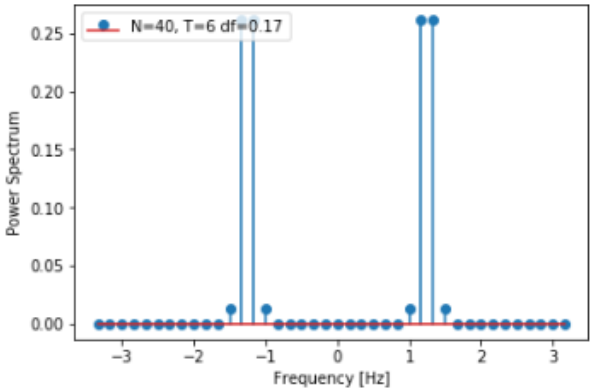



```
f0 = 1.25 # 基本周波数 [Hz]
N = 40    # サンプル数
T = 6     # 観測時間[s]
dt = T/N  # サンプリング時間
df = 1/T  # 周波数分解能
x = np.sin(2*np.pi*f0*t)
```

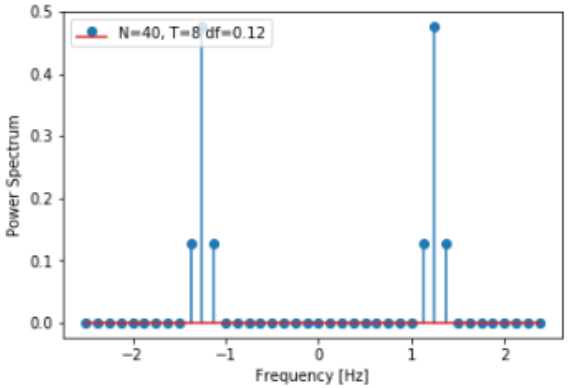
見たい周波数の1/10の周波数分解能が必要な例
⇒観測時間Tの選定



窓　無し（矩形窓）



Hann窓　有
周波数分解能が低いので
基本周波数が2本ある？



Hann窓　有，左よりdfを
細かくする-> df = 0.125
-> T=1/df = 8　とした

不規則信号のスペクトル分析

ここより，信号は確率過程（stochastic process）となる。これをランダム信号，不規則信号ともいう。

確率過程の最大の注意点は期待値操作が必要なことにある。

期待値操作：集合平均が基本となるが、時系列データの場合、時間平均を考えることも多々ある。

不規則信号のスペクトル分析

不規則信号の問題点

連続信号 $x(t)$ が有限のパワーをもつことを式で表現すると次となる。

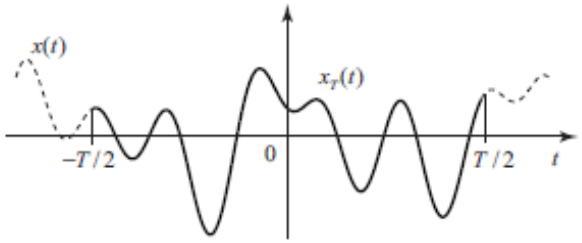
$$\int_{-\infty}^{\infty} x^2(t)dt < \infty \tag{7.1}$$

これは，6.1 節でも述べたように，無限区間を対象としたフーリエ変換の積分を可能とする可積分条件であるが，不規則信号では一般にこの条件が成立しない。なぜならば，可積分条件は

$$\lim_{t \to \pm\infty} x(t) = 0$$

を要求するが，不規則信号は一般にこの条件を満足しない。したがって，そのフーリエ変換は存在しないことになり，6.1 節で説明したエネルギースペクトル密度は定義できない

そこで！



$$x_T(t) = \begin{cases} x(t) & , |t| \le T/2 \\ 0 & , |t| > T/2 \end{cases}$$

不規則信号のスペクトル分析

フーリエ変換

$$X_T(f) = \int_{-\infty}^{\infty} x_T(t) \exp(-j2\pi ft)\, dt$$

$$S'_P(f) = \lim_{T\to\infty} \frac{|X_T(f)|^2}{T}$$

左式は，不規則信号の見本過程に対して導かれたものであり確率変数である。すなわち，見本過程毎に値が異なる。
そのため，期待値操作を導入する。

【不規則信号の PSD】　不規則信号 $x(t)$ の**パワースペクトル密度**（PSD; power spectrum density）を次で表わす。

$$S_P(f) = \lim_{T\to\infty} E\left\{\frac{|X_T(f)|^2}{T}\right\} \qquad (7.9)$$

不規則信号のスペクトル分析

幾つかの分析方法

- Scipyが提供;ペリオドグラム法, Wlech法, Lomb-Scargle法
- 他に:BT法(Blackman-Tukey,自己相関関数を用いる), MEM (Maximum Entropy Method, ARモデルを用いたBurgのアルゴリズムに基づく)
- 古典的ではあるが, 理論展開もシンプルで, 実用的にも良く用いられている**ペリオドグラム法**について説明する

2つの周波数が隣接, 信号に雑音が重畳

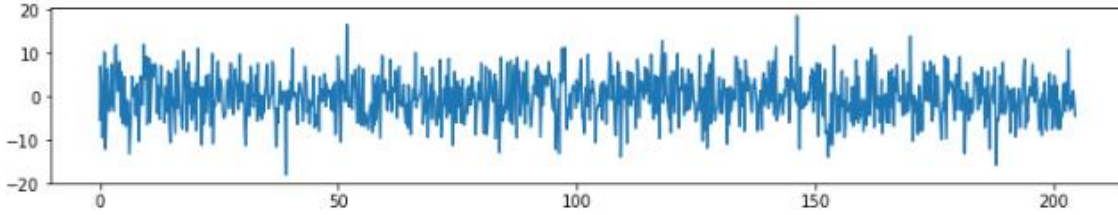
PSD_Periodogram

```
f1, a1  = 1.2, 1.0 # 周波数 [Hz], 振幅
f2, a2  = 1.3, 1.0 # 周波数 [Hz], 振幅
sd = 5.0  # 観測雑音の標準偏差

dt  = 0.2  # サンプリング時間 [s]
T   = 100  # 観測時間 [s]
Num = 1024 # ゼロ埋込みのための, 見掛け上のサンプル数
df = 1/(dt*Num)  # 見掛け上の周波数分解能
t = np.linspace(0, Num-1, Num)*dt # 時間軸
freq = np.fft.fftfreq(Num, d=dt) # 周波数軸

w_hamming = signal.hamming(Num) # ハミング窓

x = a1*np.sin(2*np.pi*f1*t) + a2*np.sin(2*np.pi*f2*t) + ¥
        np.random.normal(loc=0.0, scale=sd, size=Num) # 信号
fig = plt.subplots(figsize=(12,2))
plt.plot(t,x)
plt.show()
```

不規則信号のスペクトル分析

PSD_Periodogram

ある観測波形

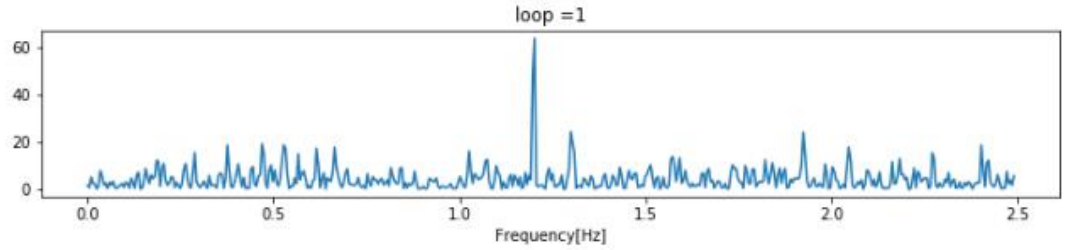


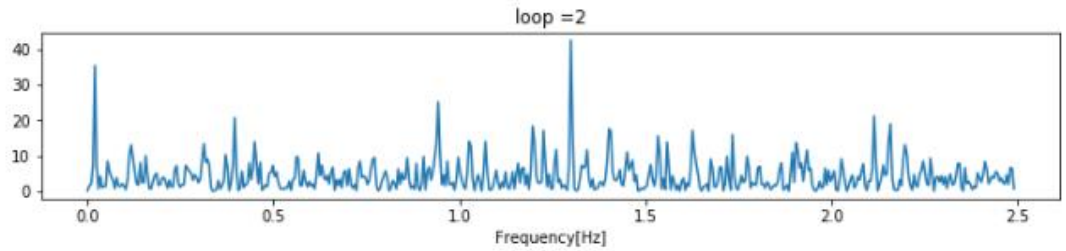
これをイメージ図にすると？

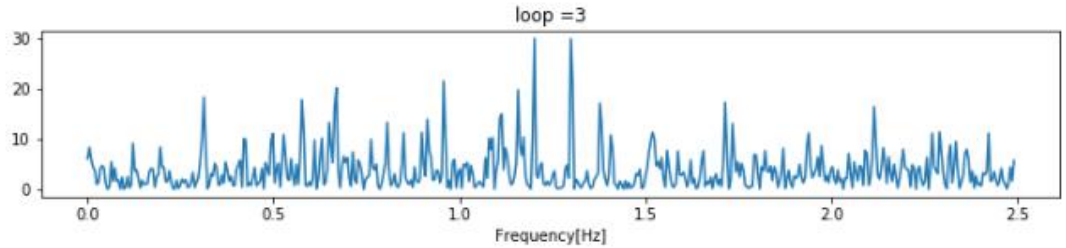
2つの周波数が隣接，信号に雑音が重畳

PSDの平均操作

```
nend=np.int(Num/2)-1
psd_sum = np.zeros(Num)
loop_num = 10
for loop in range(1,loop_num+1):
    x = a1*np.sin(2*np.pi*f1*t) + a2*np.sin(2*np.pi*f2*t) + ¥
        np.random.normal(loc=0.0, scale=sd, size=Num) # 信号
    xw = w_hamming*x
    dft = dt*scipy.fftpack.fft(xw)
    psd = (np.abs(dft)**2)/T
    psd_sum += psd
    fig = plt.subplots(figsize=(4,2))
    plt.plot(freq[0:nend], psd[0:nend])
    plt.xlabel('Frequency[Hz]')
    plt.title('loop ='+str(loop))
# 保存する図番号を文字に変換するためのスクリプト
    if FLAG_fig:
        fname = 'fig_PSDrand'+str(loop)+'.png'
        plt.tight_layout() # xlabelの欠落を防ぐ
        plt.savefig(fname)
    plt.show()
psd_sum /= loop_num
```

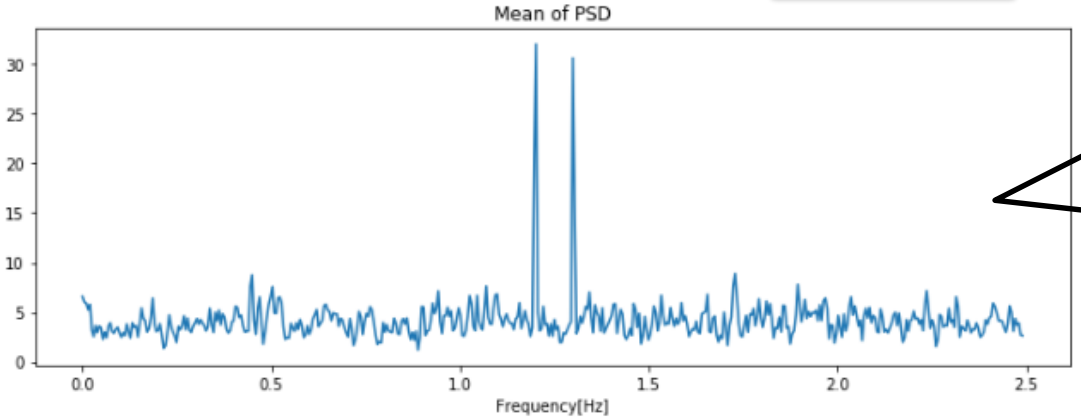






各PSDを見て，どこに信号の周波数があるか，わからない。
各値の分散値（ばらつき）は大きいことが知られている（ランダムデータの統計的処理）

平均操作



PSDの10回平均の結果

信号の周波数が2本見える

演習：

- f1, f2を変えて，試してみる
  - a1, a2はそのまま(変えるパラメータが多すぎると，大変！)
  - f1, f2を変えたときのサンプリング周波数が適正か？　これをどう検証する？

scipy.signal.periodogram

- https://scipy.github.io/devdocs/generated/scipy.signal.periodogram.html
- 先のscipy.fftpack.fft　を用いたパワースペクトル計算と比較して，PSDの値が2倍となる。
- これは，両側フーリエ変換に基づいているためである。
- 一般に，窓関数によって，PSDのピーク値は変化する。
- このため，PSDの見方は，PSDの絶対値を注視するのではなく，PSDの相対値に注目する。
- よって，PSDの最大値を1とした相対値を見ることが多いので，この見方ならば，どちらの方法で求めても良いことになる。

演習：周波数で見なければならない場面

- **以下の分野で探して、**
  - 何（物理量：速度、振動など）の周波数を見ているのか？
  - 周波数成分を見ることて、どのような判断・診断を行おうとしているのか？
- **分野**
  - 土木、建築
  - 医療
  - 音楽
  - 電機機器・機械
  - 自然現象（天候、宇宙、河川、森林）
  - 他

参考文献 References

1. J.S.ベンダット，A.G.ピアソル著，得丸英勝，他，訳：ランダムデータの統計的処理，培風館，1976
2. 日野幹雄：スペクトル解析，朝倉書店，2010
3. 越川常治：信号解析入門，近代科学社，1992
4. 赤池，中川：ダイナミックシステムの統計的解析と制御，サイエンス社（1972）

END