情報工学科 講義 ディジタル信号処理A (2025/04/24)

第3回
ラプラス変換から z 変換へ

情報工学科 准教授　高道 慎之介

Takamichi Laboratory
慶應義塾大学 高道研究室

ディジタル信号処理Aの授業予定 (仮)

第XX回	日付	内容（順次変わっていくので予想）
第01回	2025/04/10	イントロダクション，ディジタル信号処理
第02回	2025/04/17	フーリエ級数展開・フーリエ変換から離散フーリエ変換へ
第03回	**2025/04/24**	**ラプラス変換から z 変換へ**
第04回	2025/05/01	インパルス応答と伝達関数，安定性
第05回	2025/05/08	ディジタルフィルタ
第06回	2025/05/15	高速フーリエ変換と短時間フーリエ変換
第07回	2025/05/22	総合演習．期末試験の練習としての立ち位置．
期末試験	2025/06/??	(日程は後日アナウンス)

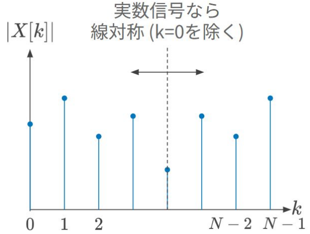
応用数学の復習

昨年のBの途中まで

前回の課題の解答

以下の課題についてレポートを作成し，LMS 上で提出せよ。

1. 周期関数 x(t) のフーリエ級数 c_k と c_{-k}は複素共役であることを示せ



2. フーリエ変換の時間シフトを証明せよ

3. サンプリング周波数を 4000 Hz とする．離散フーリエ変換の周波数解像度 (k と k - 1 の周波数軸上の間隔) を 20 Hz としたい．そのときに，必要な信号時間長 T を答えよ

4. 離散フーリエ変換は行列として表現することができる．この行列を Python で記述せよ．詳細は次のページ

周期関数 x(t) のフーリエ級数 c_k と c_{-k}は複素共役であることを示せ

- c_k の項と c_{-k} の項の和を考える

$$x(t) = \sum_{k=-\infty}^{\infty} c_k \exp\left(j\frac{2\pi kt}{T}\right)$$

- 和は以下の通り。(他の項は直交するので無視してよい)

$$(a_k + jb_k)e^{j\frac{2\pi kt}{T}} + (a_{-k} + jb_{-k})e^{j\frac{-2\pi kt}{T}}$$

- x(t) は実数関数なので、その虚部は 0 になる

$$b_k \cos(\frac{2\pi kt}{T}) + a_k \sin(\frac{2\pi kt}{T}) + b_{-k} \cos(\frac{2\pi kt}{T}) - a_{-k} \sin(\frac{2\pi kt}{T})$$

- これが成立するには、a_k = a_{-k}, b_{k} = - b_{-k} (複素共役)

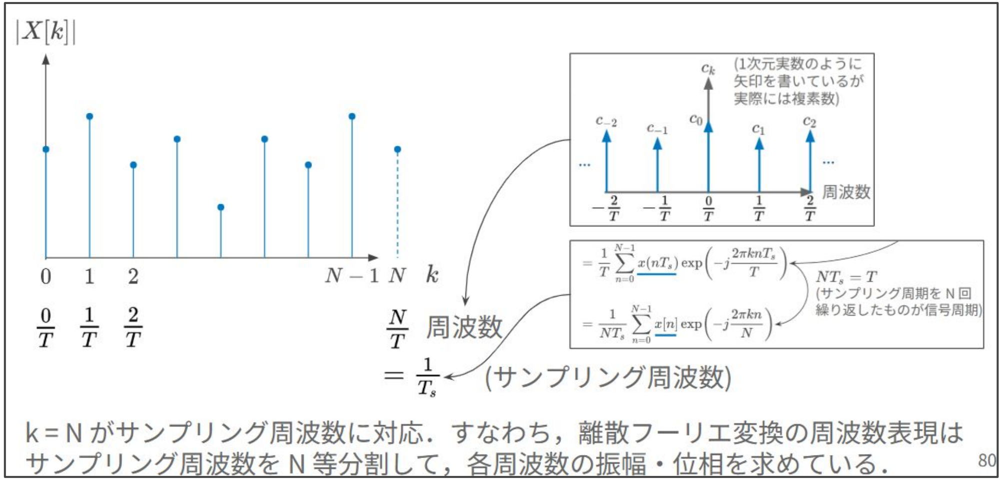
- 別解：c_k の定義から。x(t) が実数関数なので成り立つ

$$c_k = \frac{1}{T}\int_{-T/2}^{T/2} x(t)\exp\left(-j\frac{2\pi kt}{T}\right)dt \qquad c_{-k} = \frac{1}{T}\int_{-T/2}^{T/2} x(t)\exp\left(j\frac{2\pi kt}{T}\right)dt = \left(\frac{1}{T}\int_{-T/2}^{T/2} x(t)\exp\left(j\frac{-2\pi kt}{T}\right)dt\right)^* = c_k^*$$

フーリエ変換の時間シフトを証明せよ

- $\tau = t - a$ として積分の変数変換すれば解ける。

サンプリング周波数を 4000 Hz とする．離散フーリエ変換の周波数解像度 (k と k - 1 の周波数軸上の間隔) を 20 Hz としたい．そのときに，必要な信号時間長 T を答えよ



- サンプリング周波数1/T_{s} = 4000 Hz, 周波数間隔 20Hz
- N = 4000 / 20 = 200, T = 200 / 4000 = 0.05[sec]

- 別解：周波数解像度の逆数が信号時間長 1/20 = 0.05 [sec]

離散フーリエ変換は行列として表現することができる．この行列を Python で記述せよ．

```python
import numpy as np

def DFT_matrix(N):
  W = np.zeros((N,N), dtype=complex) # N-by-N zero matrix
  """
  Generate DFT matrix of size N.
  """
  for n in range(N):
    for k in range(N):
      W[n][k] = np.exp(- 2j * np.pi * n * k / N)
  return W

xn = np.array([1, 2, 3, 4, 5, 6, 7, 8]) # input signal
W = DFT_matrix(len(xn))

# check
print("your answer:\n", W @ xn) # Xk = W xn
print("numpy:\n", np.fft.fft(xn)) # reference. OK if your answer matches to this.
```

(「python で多重ループを書くな」という話はありますが簡単のために)

本日の内容

ハウリング：時間とともに信号が増幅する現象

本日の内容

- ラプラス変換
- z 変換

ラプラス変換，z 変換，フーリエ変換との関係を理解しよう．
これまでにやった ”周波数” を拡張します．

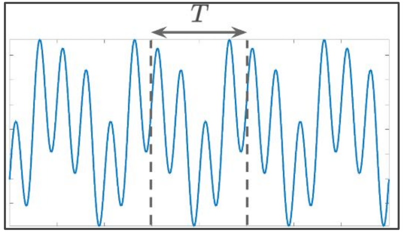
復習：フーリエ変換

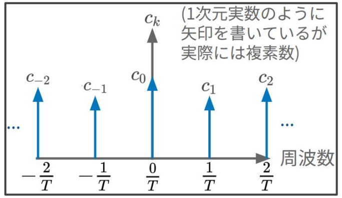
手法の比較

連続時間信号

フーリエ級数展開

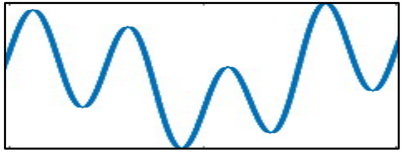
周期**連続**時間信号 → **離散**周波数

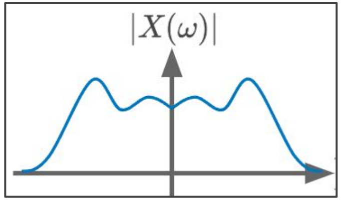




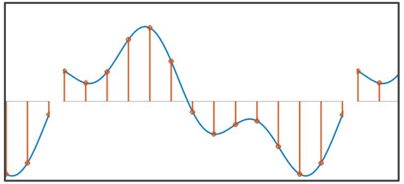
フーリエ変換

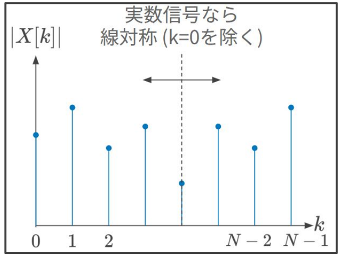
非周期**連続**時間信号 → **連続**周波数





離散時間信号





離散フーリエ変換

周期**離散**時間信号 → 周期**離散**周波数

本講義ではやりません．
興味があれば調べてください．

離散時間フーリエ変換

非周期**離散**信号 → **連続**周波数

離散周波数 (信号の周期性を仮定)

連続周波数 (信号は非周期で良い)

フーリエ変換 とその逆変換 (ωは角周波数，2π×周波数)

- Fourier 変換： 連続時間信号 → 連続周波数

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

$$\mathcal{F}\left[x(t)\right] = X(\omega)$$

- 逆Fourier 変換： 連続周波数 → 連続時間信号

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

$$\mathcal{F}^{-1}\left[X(\omega)\right] = x(t)$$

離散フーリエ変換とその逆変換

- 離散 Fourier 変換： 離散時間信号 → 離散周波数

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp\left(-j\frac{2\pi kn}{N}\right) \qquad \mathrm{DFT}\left[x[n]\right] = X[k]$$

$$(k \in \{0, 1, \ldots, N-1\})$$

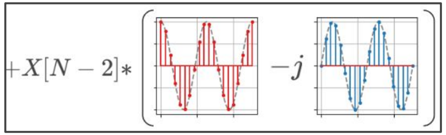
- 逆離散 Fourier 変換： 離散周波数 → 離散時間信号

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \exp\left(j\frac{2\pi kn}{N}\right) \qquad \mathrm{DFT}^{-1}\left[X[k]\right] = x[n]$$

$$(n \in \{0, 1, \ldots, N-1\})$$

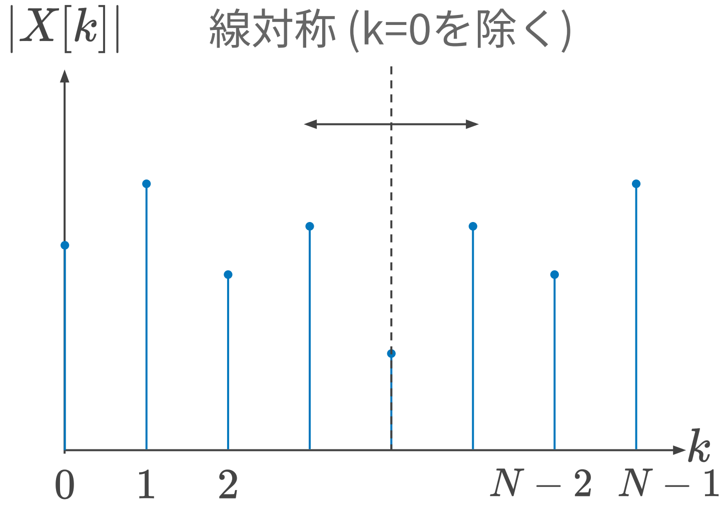
振幅スペクトル $|X[k]|$ と位相スペクトル $\arg(X[k])$

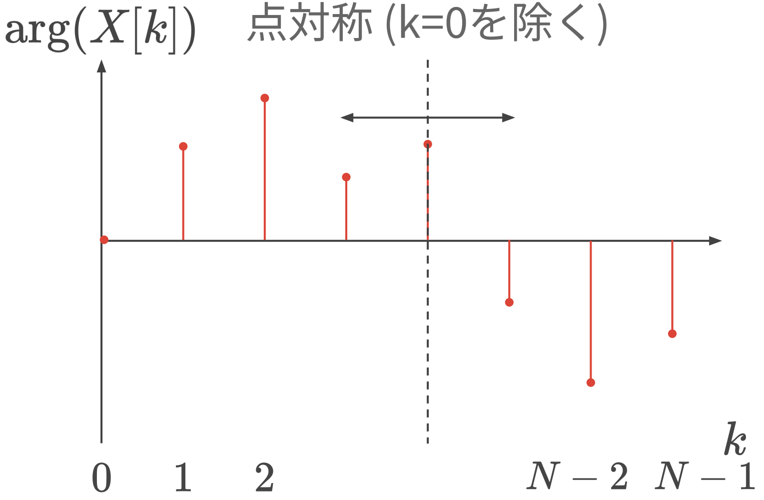
$$X[k] = |X[k]|e^{j\arg(X[k])}$$



$\exp\left(j\frac{2\pi kn}{N}\right)$の大きさを何倍するか

$\exp\left(j\frac{2\pi kn}{N}\right)$の開始時刻をどれくらい動かすか



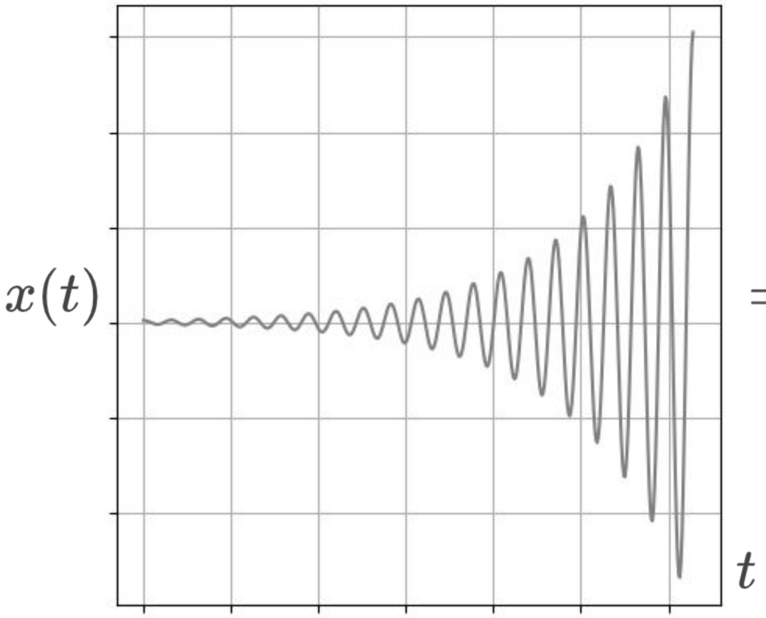


実数信号を離散フーリエ変換する場合には N-1の半分まででよい (図示する場合もそう)

ラプラス変換
(Laplace transformation)

フーリエ変換は全ての信号を変換できるわけではない

ハウリングのように
時間とともに無限大に増大する信号



全ての基底波の足し合わせ

基底波
(cos 波と sin 波)

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

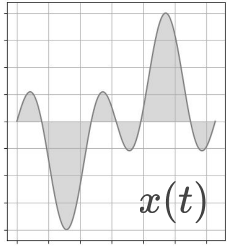
振幅(大きさ)と
位相 (時間遅れ)

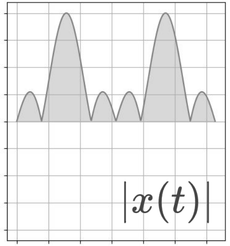
|X(ω)| が有限の値である限り，無限大に増大する信号を，基底波の足し合わせで表すことはできない

フーリエ変換できるのは絶対可積分関数のみ

- 関数 x(t) が絶対可積分であるとは，絶対値の積分値が上に有界

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





- 絶対可積分関数なら，フーリエ変換は上に有界 (有限の値になる)

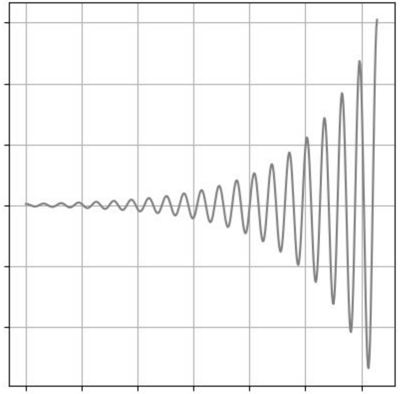
$$\begin{aligned}
|X(\omega)| &= \left| \int_{-\infty}^{\infty} x(t) \exp(-j\omega t) dt \right| \\
&\leq \int_{-\infty}^{\infty} |x(t) \exp(-j\omega t)|\, dt \\
&= \int_{-\infty}^{\infty} |x(t)|\, dt \\
&< \infty
\end{aligned}$$

積分の絶対値は，絶対値の積分以下

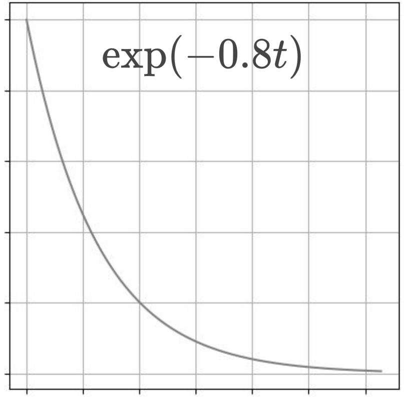
exp項の大きさは 1

絶対可積分の条件

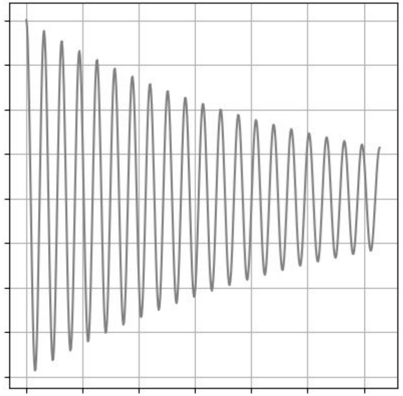
じゃあどうする？：
時間とともに減衰させる関数を乗算すればよい



$\times$



$=$

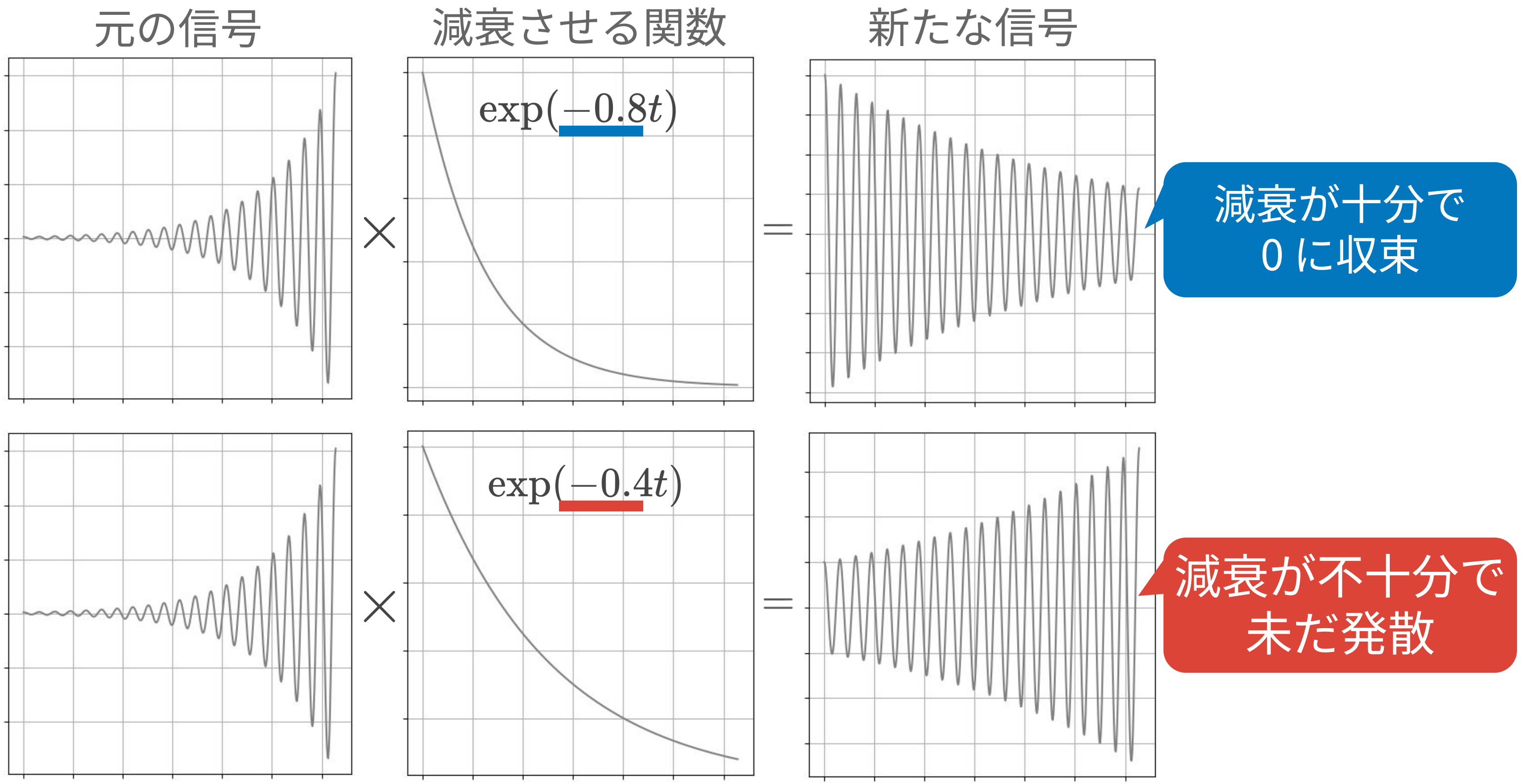


$$= \frac{1}{2\pi}\int_{-\infty}^{\infty} \underline{X(\omega)} \exp(j\omega t)d\omega$$

これだったら
求められそう

時間とともに減衰させることで，変換可能な信号になりそう

さらにイメージしてみよう：
減衰の強さで信号の増減が分かる



関数の減衰パラメータの大小で，元の信号の増減具合を分析できそう

ラプラス変換の定義

- ラプラス変換： 連続時間信号 → **複素**周波数

複素数　　複素数 $s = \sigma + j\omega$

$$X(s) = \int_0^\infty x(t)e^{-st}dt \qquad \mathcal{L}\left[x(t)\right] = X(s)$$

負の時刻で発散しないよう正の時刻のみを扱う

- 逆ラプラス変換： **複素**周波数 → 連続時間信号

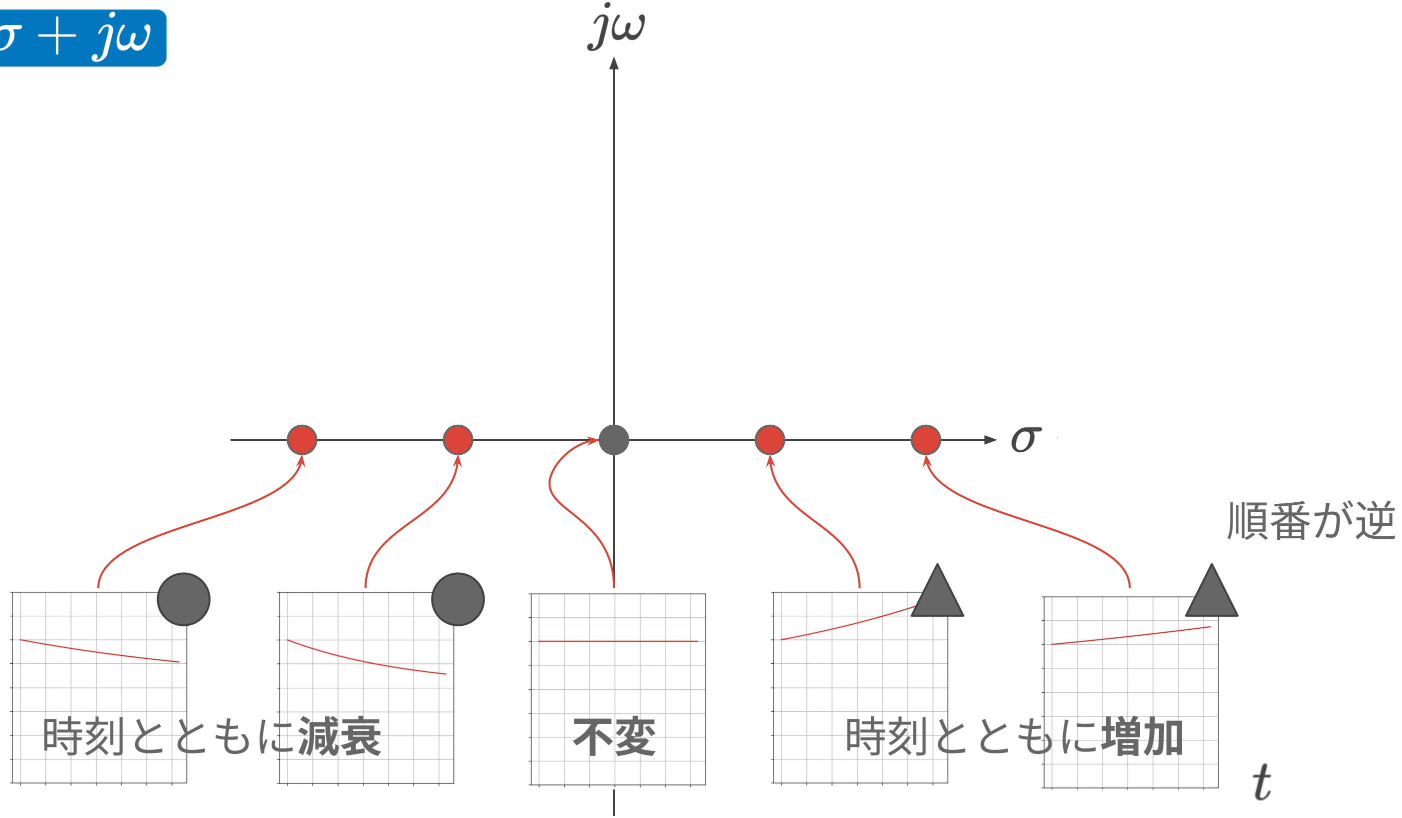
$$x(t) = \frac{1}{2\pi j}\int_{\sigma-j\infty}^{\sigma+j\infty} X(s)\exp(st)ds \qquad \mathcal{L}^{-1}\left[X(s)\right] = x(t)$$

端的に言うなら…

- ラプラス変換は，時間減衰関数を導入することで，フーリエ変換できなかった信号も変換できるようにしたもの
- フーリエ変換で使われる振動だけでなく，信号の増減も表せるもの
- 減衰関数の強さで信号の増減を解析できるもの

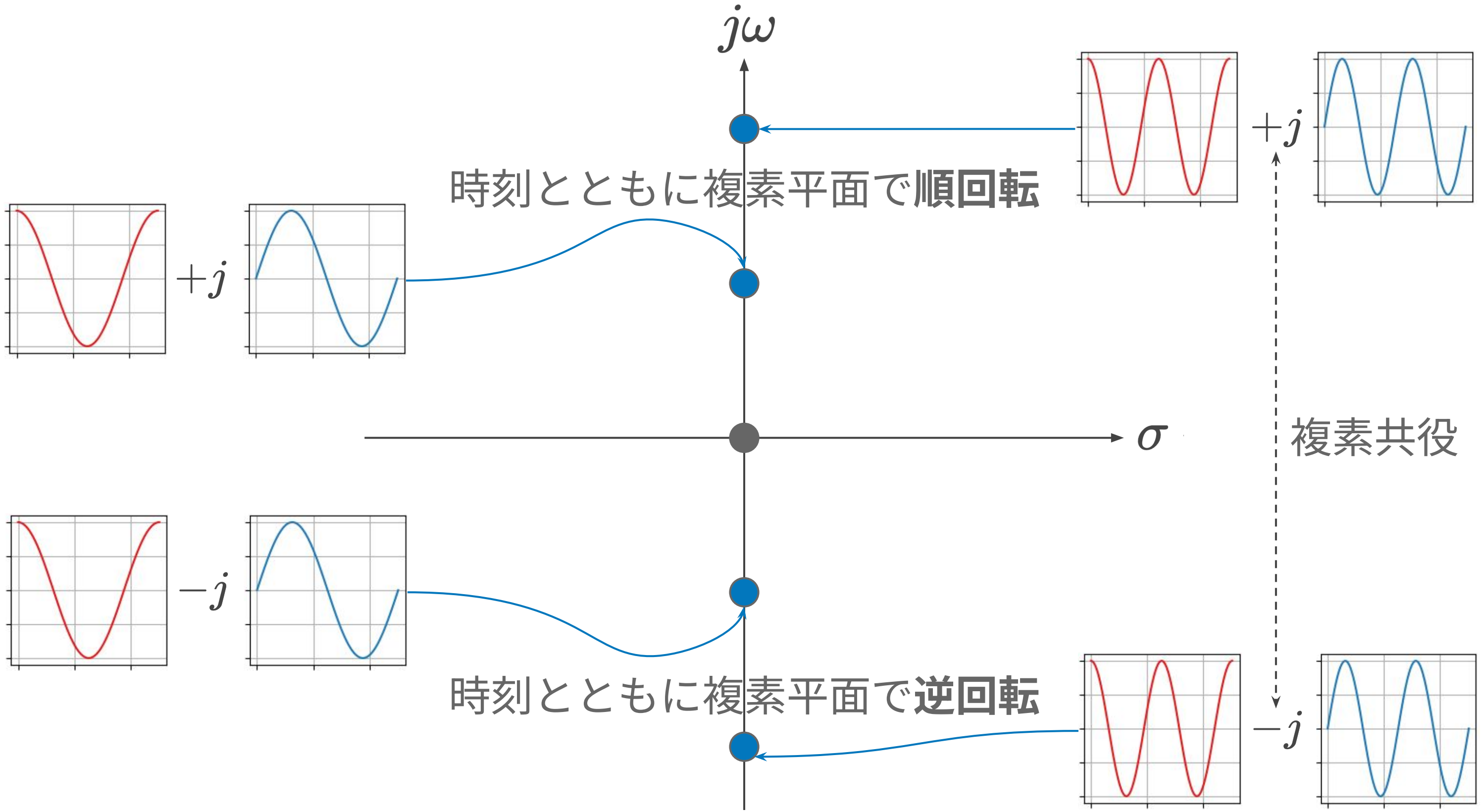
“複素”周波数の意味するところ (s 平面)

複素数 $s = \sigma + j\omega$



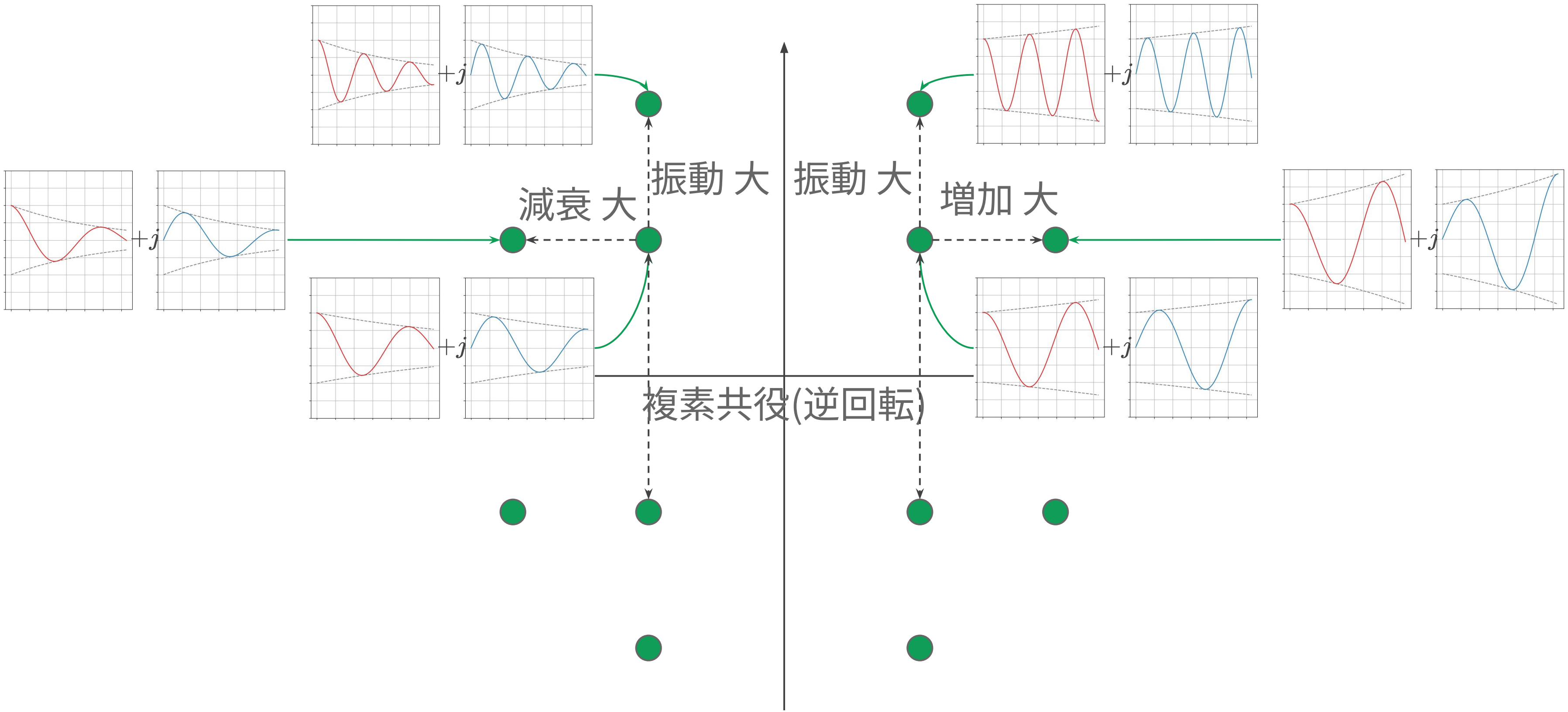
s 平面の**実軸**は時刻とともに {増加，不変，減衰}する**実指数関数**を表す．

“複素”周波数の意味するところ (s 平面)



s 平面の**虚軸**は時刻とともに {順回転，逆回転}する**複素指数関数**を表す．

“複素”周波数の意味するところ (s 平面)



ラプラス変換してみよう

- ステップ関数

$$x(t) = \{1\ (t > 0), 0\ (t \leq 0)\} \xrightarrow{\text{基本形は 1/s}} \mathcal{L}[x(t)] = \int_0^\infty 1 \cdot e^{-st} dt = \left[\frac{e^{-st}}{-s}\right]_0^\infty = \frac{1}{s}$$

- 指数関数 (a > 0)

減衰項で左にシフト
(増加項で右)

$$x(t) = e^{-at} \longrightarrow \mathcal{L}[x(t)] = \int_0^\infty e^{-at} \cdot e^{-st} dt = \left[\frac{e^{-(a+s)t}}{-(a+s)}\right]_0^\infty = \frac{1}{s+a}$$

- 正弦波関数 (ω_0 > 0)

振動項で
上下にシフト

$$x(t) = \sin(\omega_0 t) \longrightarrow \mathcal{L}[x(t)] = \mathcal{L}\left[\frac{1}{2j}\left(e^{j\omega_0 t} - e^{-j\omega_0 t}\right)\right] = \frac{1}{2j}\left(\frac{1}{s - j\omega_0} - \frac{1}{s + j\omega_0}\right) = \frac{\omega_0}{s^2 + \omega_0^2}$$

- 指数関数×正弦波関数

左にシフト＆
上下にシフト

$$x(t) = e^{-at}\sin(\omega_0 t) \longrightarrow \mathcal{L}[x(t)] = \mathcal{L}\left[\frac{e^{-at}}{2j}\left(e^{j\omega_0 t} - e^{-j\omega_0 t}\right)\right] = \frac{\omega_0}{(s+a)^2 + \omega_0^2}$$

X(s) を可視化してみよう (Google colab版)

```
# try on Google Colab or Jupyter Notebook
# from IPython.display import display
import sympy as sp # for symbolic calculation
import numpy as np
import plotly.graph_objects as go # for dynamic drawing
sp.init_printing()

# define symbols, signal, and calculate Laplace transformation
s, t = sp.symbols("s t")
x = sp.exp(-0 * t)
X = sp.laplace_transform(x, t, s, noconds=True)

# print symbols
for message, variable in zip(["x(t)", "L[x(t)]"], [x, X]):
  print(message, "=")
  display(variable)


# convert it to a numerical function for plotting.
X_func = sp.lambdify(s, X, 'numpy')

# create a grid of s values (complex numbers)
real_s, imag_s = np.linspace(-0, 3, 50), np.linspace(-3, 3, 50)
real_s, imag_s = np.meshgrid(real_s, imag_s)
s_vals = real_s + 1j * imag_s

# evaluate the Laplace transform for the complex values of s
X_vals = X_func(s_vals)

# Extract real and imaginary parts
real_X, imag_X = np.real(X_vals), np.imag(X_vals)
```

```
# plot real part
fig = go.Figure(data=[go.Surface(
    z=real_X, x=real_s, y=imag_s, hidesurface=True,
    contours=dict(
        x=dict(show=True, color="red", start=-0, end=3, size=0.25),
        y=dict(show=True, color="red", start=-3, end=3, size=0.25),
    ))])

fig.update_layout(
    scene=dict(xaxis_title='Re(s)', yaxis_title='Im(s)',
zaxis_title='Re[X(s)]'),
    width=None, height=None, autosize=True,
    )
# fig.show()

# plot imag part
# fig = go.Figure(data=[go.Surface(
#     z=imag_X, x=real_s, y=imag_s, hidesurface=True,
#     contours=dict(
#         x=dict(show=True, color="blue", start=-0, end=3, size=0.25),
#         y=dict(show=True, color="blue", start=-3, end=3, size=0.25),
#     ))])

# fig.update_layout(
#     scene=dict(xaxis_title='Re(s)', yaxis_title='Im(s)',
zaxis_title='Im[X(s)]'),
#     width=None, height=None, autosize=True,
#     )
```

X(s) を可視化してみよう

x(t) =

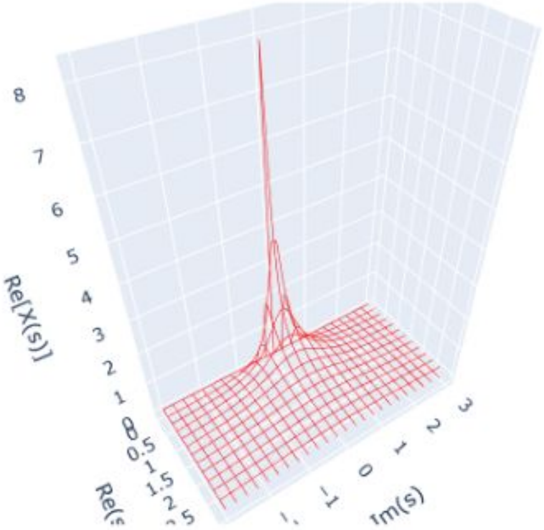
1

L[x(t)] =

$\frac{1}{s}$

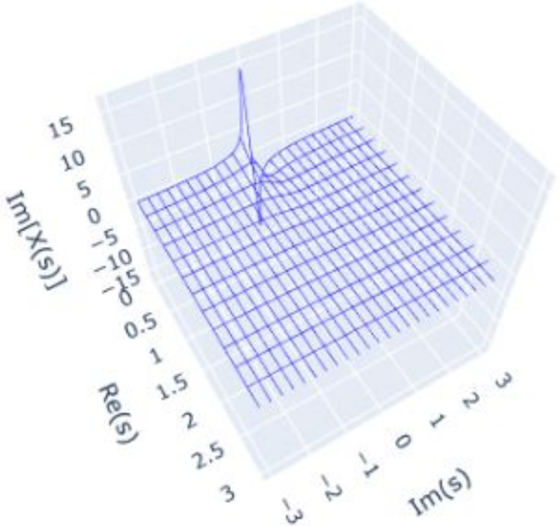
sympy はラプラス変換を
代数的に解いて
くれるので便利
(↑はステップ関数)

X(s) 実部



実軸で線対称

X(s) 虚部



実軸で点対称

ラプラス変換もフーリエ変換と同様の対称性を持つ．(実数関数を変換するとき，虚数部をキャンセルしなければならないため複素共役になる)

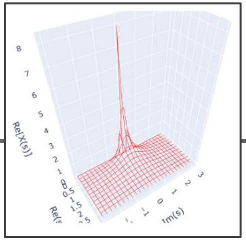
演習 (残りは課題)

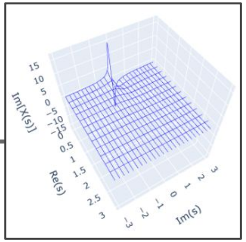
- 以下の関数についてX(s) の実部・虚部をプロットし、その形状を比較・考察せよ。a, ω_0 には適当な値を使用せよ。
  - ステップ関数 ← 前のページのやつ
  - 指数関数 (a > 0)
  - 正弦波関数 (ω_0 > 0)

- sympy の関数
  - numpy と同じように定義されているケースが多い
  - sp.exp(), sp.sin(), sp.pi

- プロットする s の範囲を適宜変更せよ。

極と定義域 (収束域) について

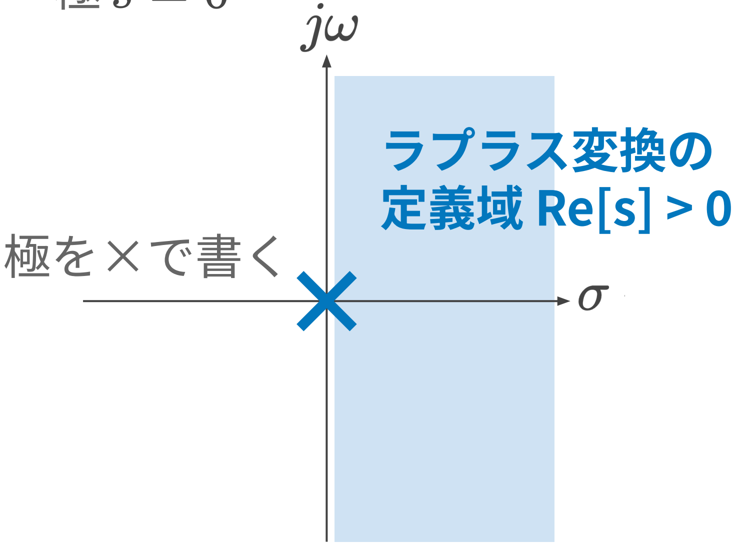




- X(s) が無限となる時の s を**極 (pole)** と呼ぶ

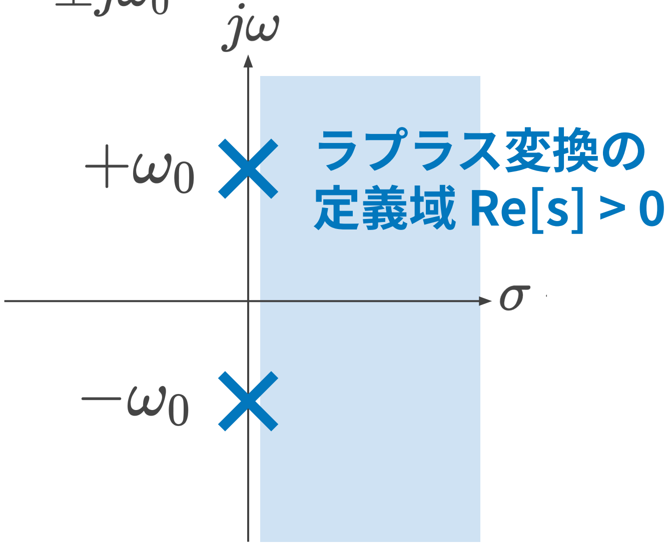
$$x(t) = 1 \longrightarrow \mathcal{L}\left[x(t)\right] = \frac{1}{s}$$

極 $s = 0$



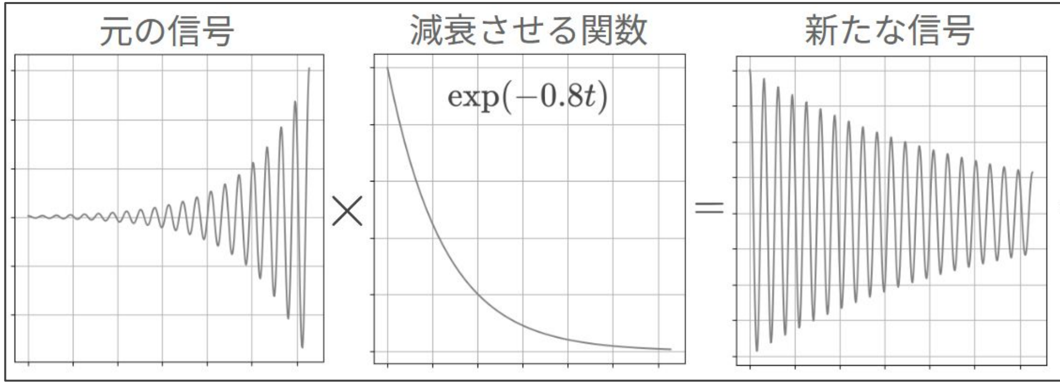
$$x(t) = \sin(\omega_0 t) \longrightarrow \mathcal{L}\left[x(t)\right] = \frac{\omega_0}{s^2 + \omega_0^2}$$

極 $s = \pm j\omega_0$



極の右側がラプラス変換の定義域になる。すなわち、ラプラス変換が有限の値として存在する s の範囲のこと。複数の極があるときは、一番右側の極の右側。

定義域 (収束域) の直感的な説明



$$X(s) = \int_0^\infty x(t) e^{-st} dt$$

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

これだったら
求められそう

◯ $\exp(at)$ $\quad$ $\exp(-\sigma t)$ $\quad$ $\propto \exp((a-\sigma)t)$

新たな信号が絶対可積分になるために：

- a - σ < 0, → a < σ であれば，t の増加とともに 0 に収束する．
- すなわち，極の実部の右側が収束域になる．
  - 元の信号が時間とともに減少 (◯exp(-at)) であれば，
    -a < σ となり非負のσ (すなわち時間増加関数) も収束域に含まれる．

フーリエ変換で成り立つ性質はラプラス変換でも成り立つ

種々の性質 (詳細は応用数学の資料を参考)

- 線形性
  - “N倍はN倍，和は和のまま”

$$\mathcal{F}[ax(t)+by(t)] = aX(\omega)+bY(\omega)$$

- 相似性
  - “時間が a 倍されると周波数は1/a倍”

$$\mathcal{F}[x(at)] = \frac{1}{|a|}X\left(\frac{\omega}{a}\right)$$

- 時間シフト
  - “時間シフトは周波数領域で exp 項”

$$\mathcal{F}[x(t-a)] = X(\omega)e^{-ja\omega}$$

- 周波数シフト
  - “周波数シフトは時間領域で exp 項”

$$\mathcal{F}^{-1}[X(\omega-a)] = x(t)e^{ja\omega}$$

- 変調定理
  - “余弦波の乗算は周波数の変調”

$$\mathcal{F}[x(t)*2\cos(at)] = X(\omega-a)+X(\omega+a)$$

- 微分定理
  - “時間の微分は周波数の乗算”

$$\mathcal{F}\left[\frac{\partial}{\partial t}x(t)\right] = j\omega X(\omega)$$

- パーセバルの定理
  - “フーリエ変換はエネルギーを保存”

$$\int_{-\infty}^{\infty}|x(t)|^2 dt = \frac{1}{2\pi}\int_{-\infty}^{\infty}|X(\omega)|^2 d\omega$$

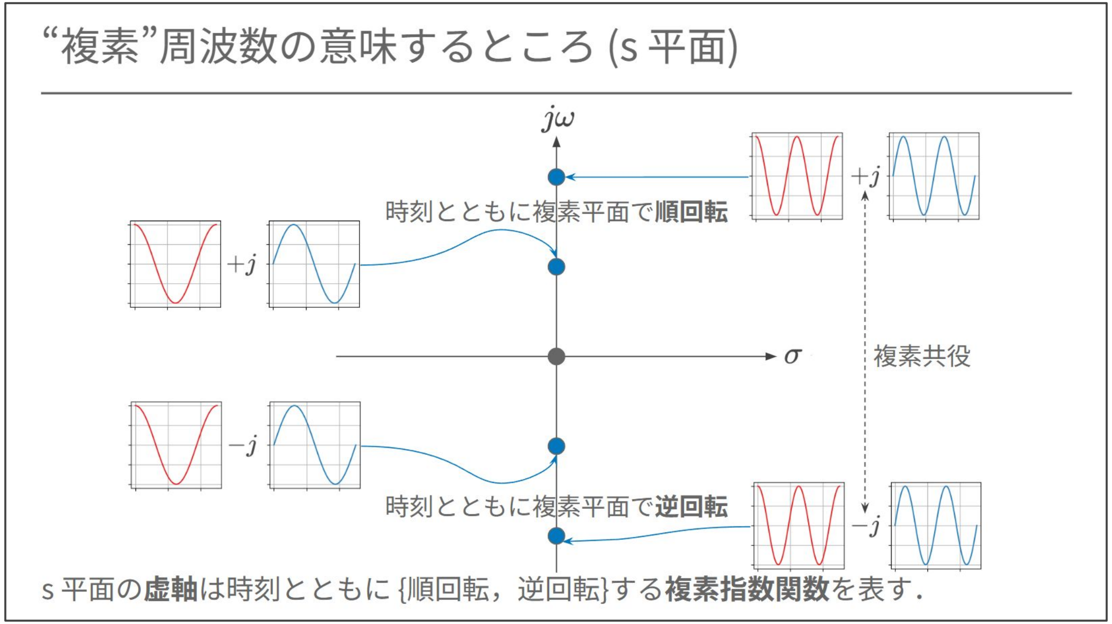
L[x(t-a)] = X(s) e^{-sa}

これらの性質は，後述する離散フーリエ変換でも成立する．

41

ラプラス変換はフーリエ変換に対応する

- ラプラス変換はフーリエ変換の一般化である
  - 具体的には，増加減衰の波を扱うようになった
- 言い換えれば，増加減衰を扱わない範囲はフーリエ変換に一致
  - すなわち，虚軸はフーリエ変換に一致



ラプラス変換はフーリエ級数展開にも対応する

- フーリエ級数展開はラプラス変換の特殊化ではない
  - フーリエ級数展開は周期信号，ラプラス変換は非周期信号
- しかし，周期信号を区間 [0, ∞] でラプラス変換すると対応を見れる．

$$X(s) = \int_0^\infty x(t)e^{-st}dt$$

周期信号 x(t+T) = x(t)

積分範囲を分割

$$= \int_0^T x(t)e^{-st}dt + \int_T^{2T} x(t)e^{-st}dt + \int_{2T}^{3T} x(t)e^{-st}dt + \dots$$

変数変換 t = τ + T．
dt = dτ．積分範囲 [0, T]

$$= \int_0^T x(t)e^{-st}\left(1 + e^{-sT} + e^{-2sT} + \dots\right)dt$$

等比級数の和

$$= \frac{1}{1 - e^{-sT}} \int_0^T x(t)e^{-st}dt$$

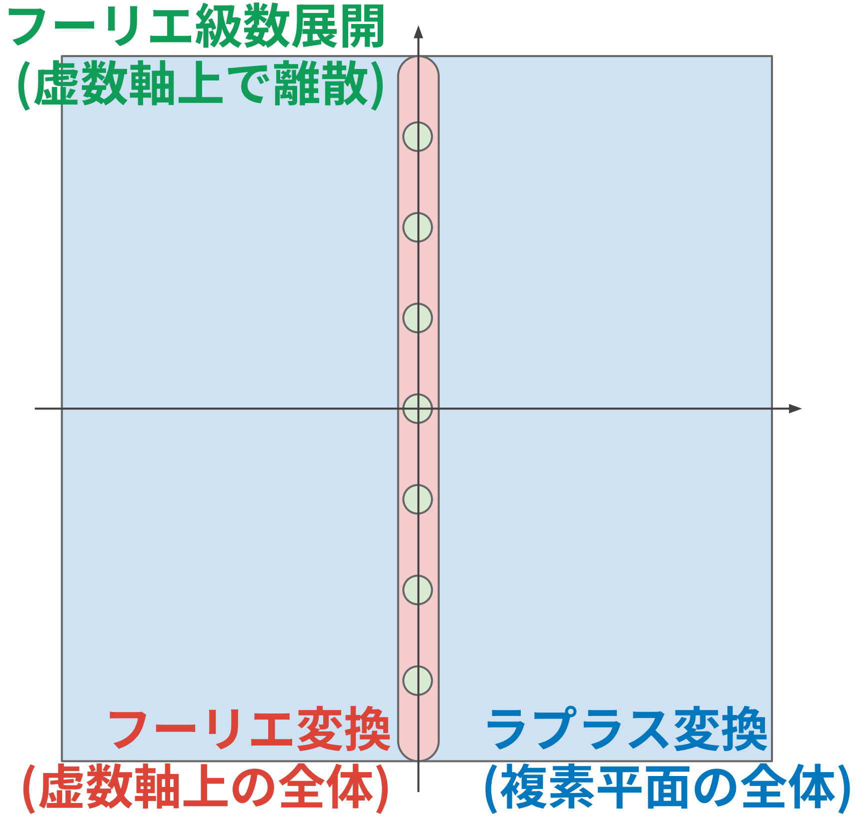
$s = e^{j2\pi k/T}$で一致

フーリエ級数

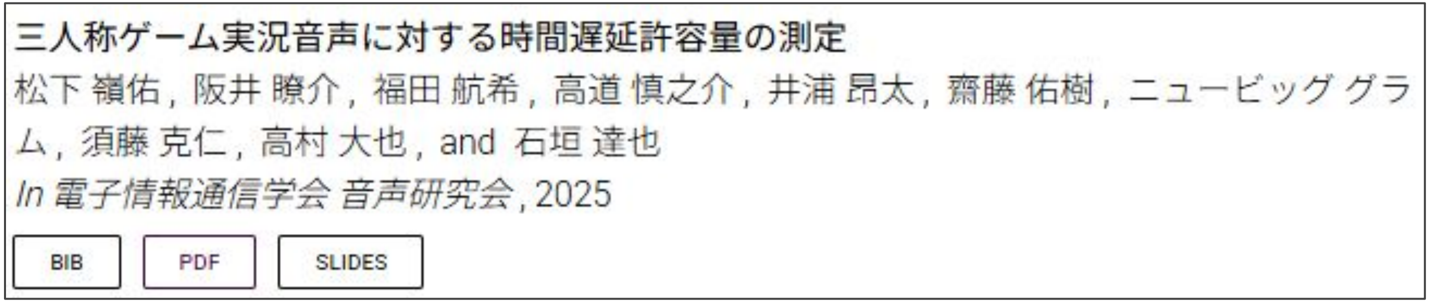
$$c_k = \frac{1}{T}\int_0^T x(t)\exp\left(-j\frac{2\pi kt}{T}\right)dt$$

ラプラス変換を虚数軸上の間隔 1/T の点に限定すると，フーリエ級数展開に対応
(級数展開可能な関数に限る)

ラプラス変換とフーリエ{級数展開, 変換}の周波数対応



閑話休題：高道研における卒業研究の紹介 (テストには出ない)


三人称ゲーム実況音声に対する時間遅延許容量の測定
松下 嶺佑, 阪井 瞭介, 福田 航希, 高道 慎之介, 井浦 昂太, 齋藤 佑樹, ニュービッグ グラム, 須藤 克仁, 高村 大也, and 石垣 達也
In 電子情報通信学会 音声研究会, 2025
BIB PDF SLIDES

https://takamichi-lab.github.io/publications/

研究背景

- スポーツやゲームの試合映像の価値を高めるために，実況を自動生成する技術が開発されている[Ishigaki 21]．

入力：実況をつけたい動画 → 実況自動生成 → 出力：実況のついた動画

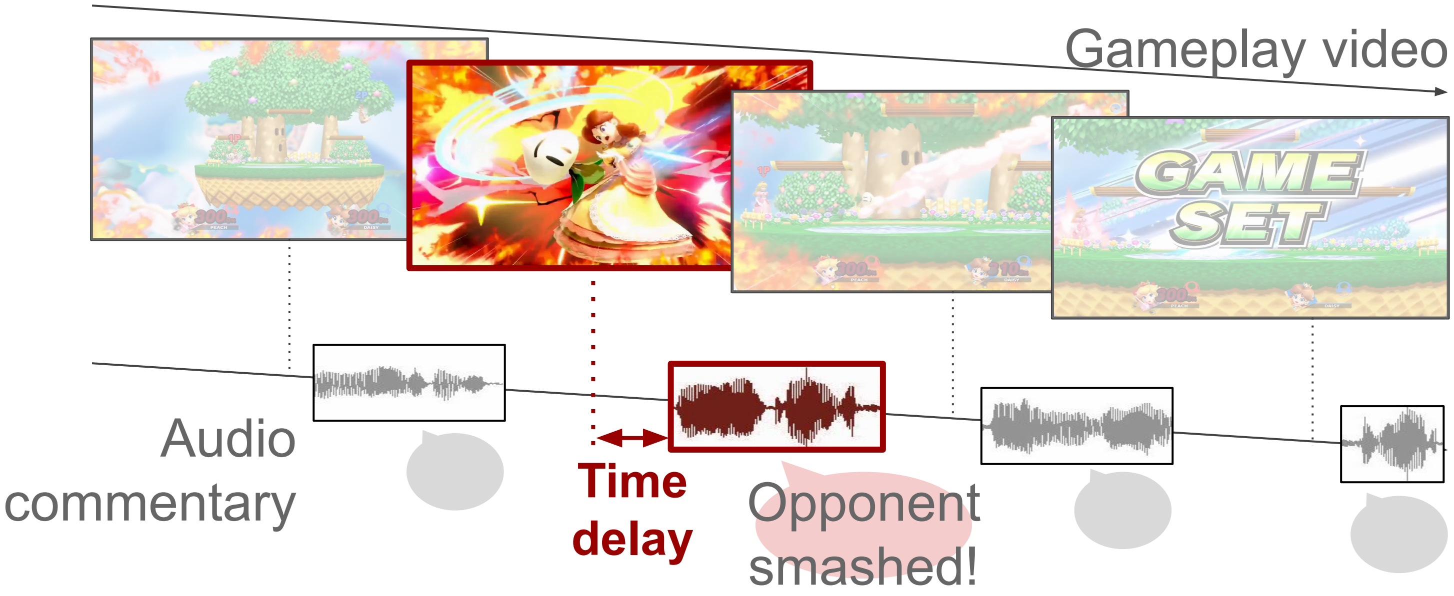
実況音声の遅延

- 一方で，実況を自動生成する場合，動画理解や実況音声の合成に時間を要するため，実況音声に遅延が生じる．
- 自然な実況音声生成のために，**許容される時間**内に遅延を抑えたい．

音声の遅延が大きい例

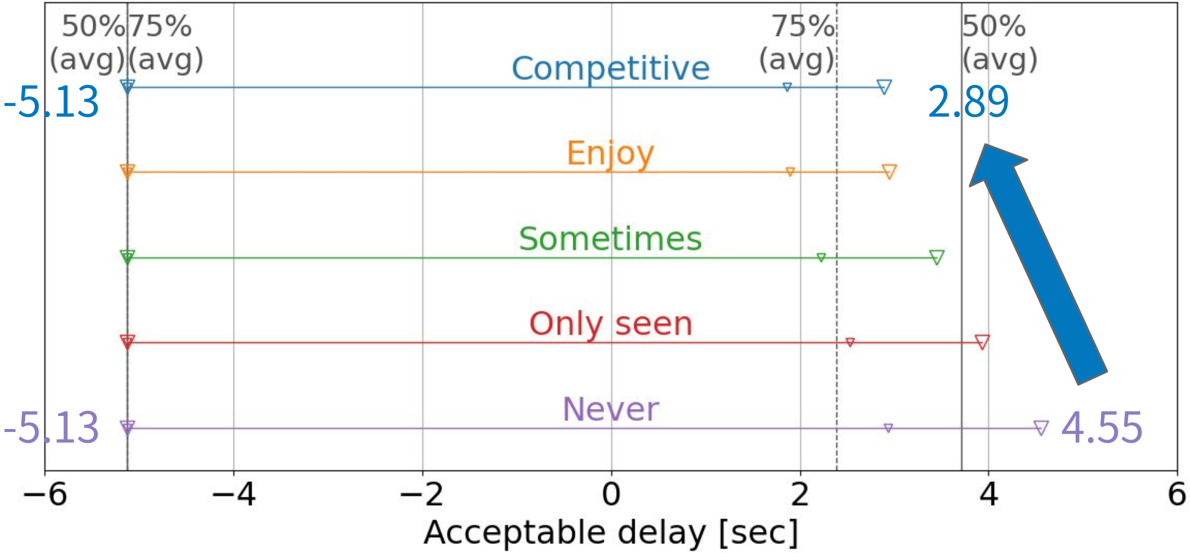
本研究の立ち位置

- 大乱闘スマッシュブラザーズSPECIALのプレイ動画を用い，
  実況の遅延時間について，どれくらい許容できるかを測定する．



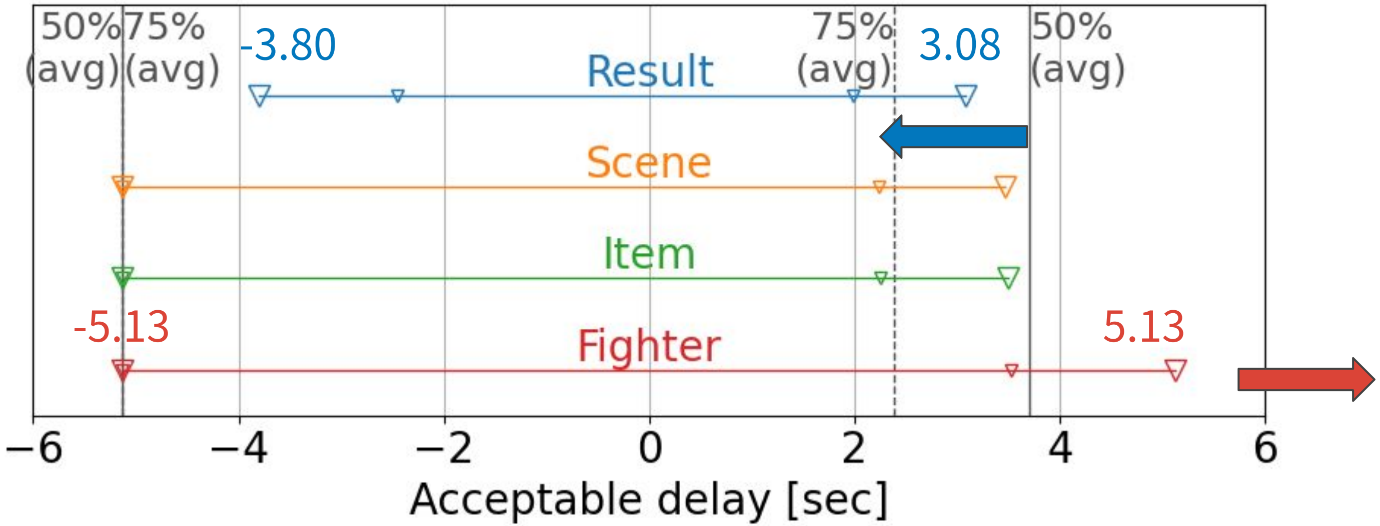
How much delay in commentary is acceptable to viewers?

実験結果：ゲームプレイ経験別



- 許容遅延時間はゲームプレイ経験が深くなるほど短い．
  - 経験のある方が反応時間が早くなるという関連研究 [Repp 06] に一致するため，被験者属性による許容遅延時間の差はあると考えられる．
  - 視聴者層に合わせて，実況自動生成をする必要がある．

実験結果：トピックタグ



- 許容遅延時間は**Result**では極端に短く，**Fighter**では極端に長い．
  - Resultは，視聴者の関心が向くので，許容遅延時間が短い．
  - Fighterは，内容が予測がしやすいので，許容遅延時間が長い．

実際の実況（Result）

遅延0秒

遅延+1.75秒

- Resultは，遅延時間が大きくなると違和感を覚えやすい．

実際の実況（Fighter）

遅延0秒

遅延+1.75秒

- Fighterは，遅延時間が大きくなっても違和感を覚えにくい．

z 変換 (z-transformation)

連続時間信号を振動だけでなく増幅・減衰で扱うのがラプラス変換だった.
じゃあ, それの離散時間信号版はあるの？ ある！

ラプラス変換から z 変換を導出する

$$X(s) = \int_0^\infty \underline{x(t)} \exp(-st)dt$$

$$= \int_0^\infty \underline{\sum_{n=0}^\infty x(nT_s)\delta(t-nT_s)} \exp(-st)dt$$

離散時間信号の連続時間信号的表現
(便宜上等号で結んでいる)

積分と総和の順序入れかえ

$$= \sum_{n=0}^\infty x(nT_s) \underline{\int_0^\infty \delta(t-nT_s)\exp(-st)dt}$$

デルタ関数の定義 $\int_{-\infty}^\infty \varphi(t)\delta(t)dt = \varphi(0)$

$$= \sum_{n=0}^\infty x(nT_s) \underline{\exp(-nsT_s)}$$

$$= \sum_{n=0}^\infty x[n](\underline{\exp(sT_s)})^{-n}$$

= z (複素数)と定義する **→ z 変換**

z変換の定義

- z 変換： 離散時間信号 → 複素周波数

$$X(z) = \sum_{n=0}^{\infty} x[n] z^{-n}$$

$$\mathcal{Z}[x[n]] = X(z)$$

- 逆z 変換： 複素周波数 → 離散時間信号

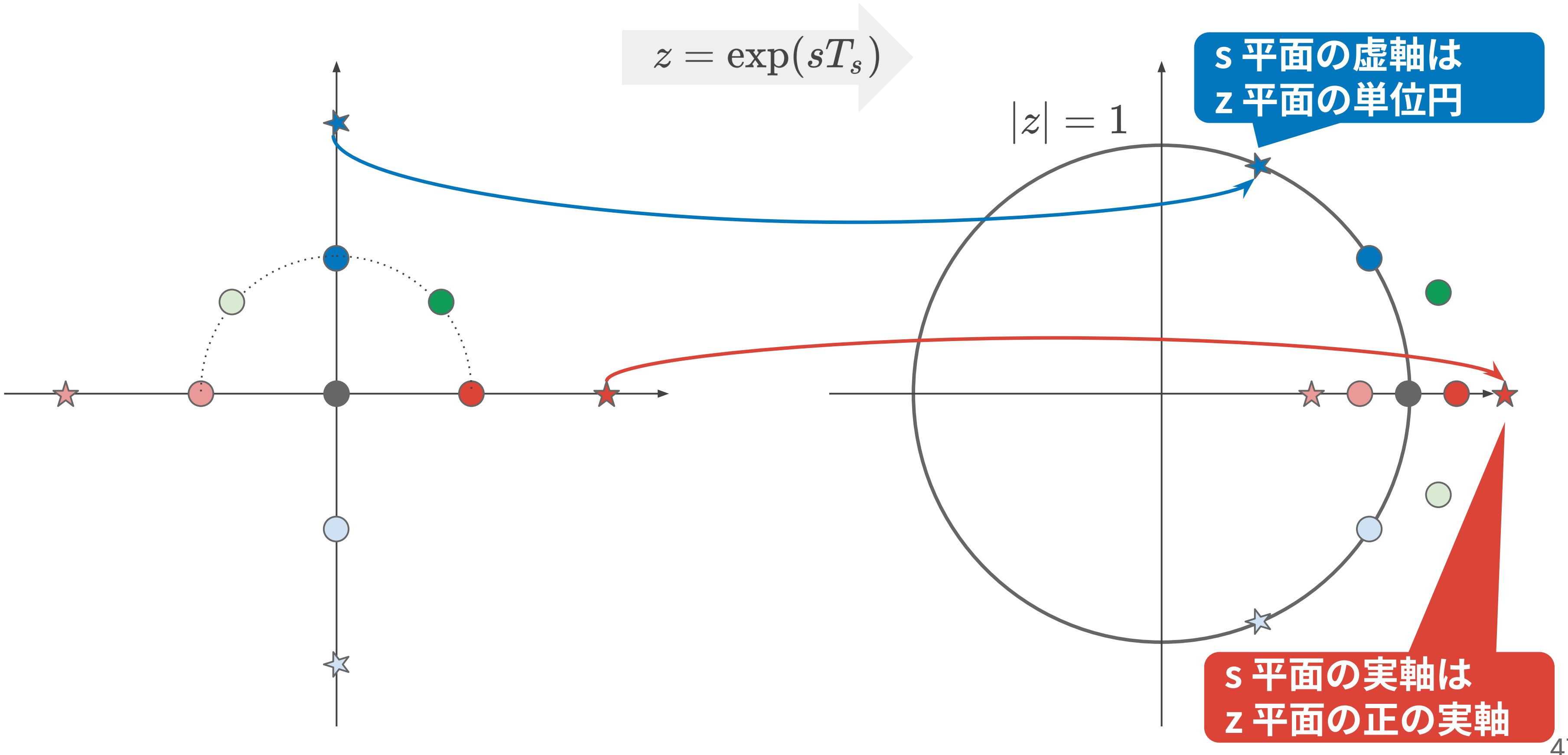
$$x[n] = \frac{1}{2\pi j} \int_{C} X(z) z^{n-1} dz$$

$$\mathcal{Z}^{-1}[X(z)] = x[n]$$

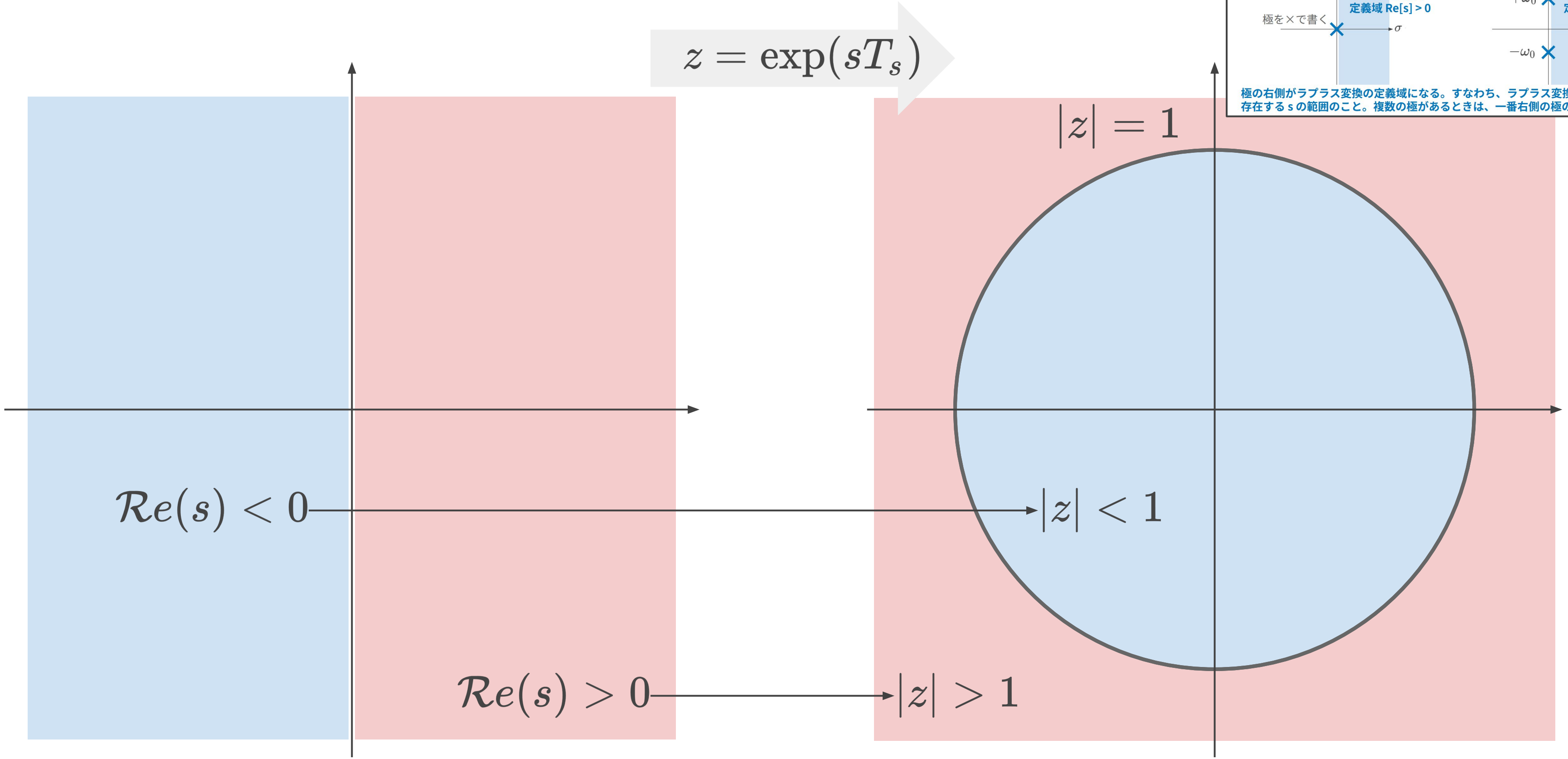
端的に言うなら…

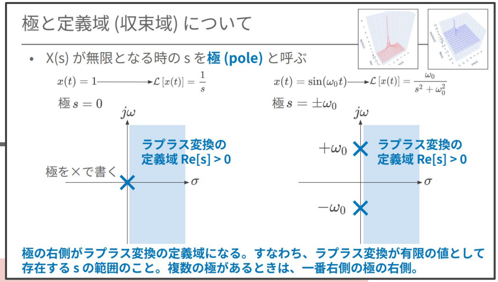
- z 変換は，ラプラス変換の離散時間信号版
- ラプラス変換と同様に，信号の増減を表現できる

s 平面と z 平面の対応



s 平面と z 平面の対応





極と定義域 (収束域) について

- X(s) が無限となる時の s を**極 (pole)** と呼ぶ

$x(t) = 1 \longrightarrow \mathcal{L}[x(t)] = \frac{1}{s}$

極 $s = 0$

$x(t) = \sin(\omega_0 t) \longrightarrow \mathcal{L}[x(t)] = \frac{\omega_0}{s^2 + \omega_0^2}$

極 $s = \pm\omega_0$

極の右側がラプラス変換の定義域になる。すなわち、ラプラス変換が有限の値として存在する s の範囲のこと。複数の極があるときは、一番右側の極の右側。

z 変換してみよう

- 単位パルス信号

$$x[n] = \delta[n] \xrightarrow{\text{基本形は 1}} \mathcal{Z}[x[n]] = 1$$

- 有限長の信号

$$x[n] = 4\delta[n] + 2\delta[n-1] + \delta[n-2] \xrightarrow[\text{z\^{}\{-1\}}]{\text{単位遅れは}} \mathcal{Z}[x[n]] = 4 + 2z^{-1} + z^{-2}$$

- ステップ信号

$$x[n] = u[n] \xrightarrow[\text{1/(1-z\^{}\{-1\})}]{\text{1ずつ乗算されると}} \mathcal{Z}[x[n]] = 1 + z^{-1} + z^{-2} + \ldots = \frac{1}{1 - z^{-1}}$$

- 等比数列信号

$$x[n] = a^n \xrightarrow[\text{1/(1-az\^{}\{-1\})}]{\text{aずつ乗算されると}} \mathcal{Z}[x[n]] = 1 + az^{-1} + a^2 z^{-2} + \ldots = \frac{1}{1 - az^{-1}}$$

- 等差数列信号

$$x[n] = n \longrightarrow \mathcal{Z}[x[n]] = \frac{z^{-1}}{(1 - z^{-1})^2}$$

逆変換も可能

- 例えば

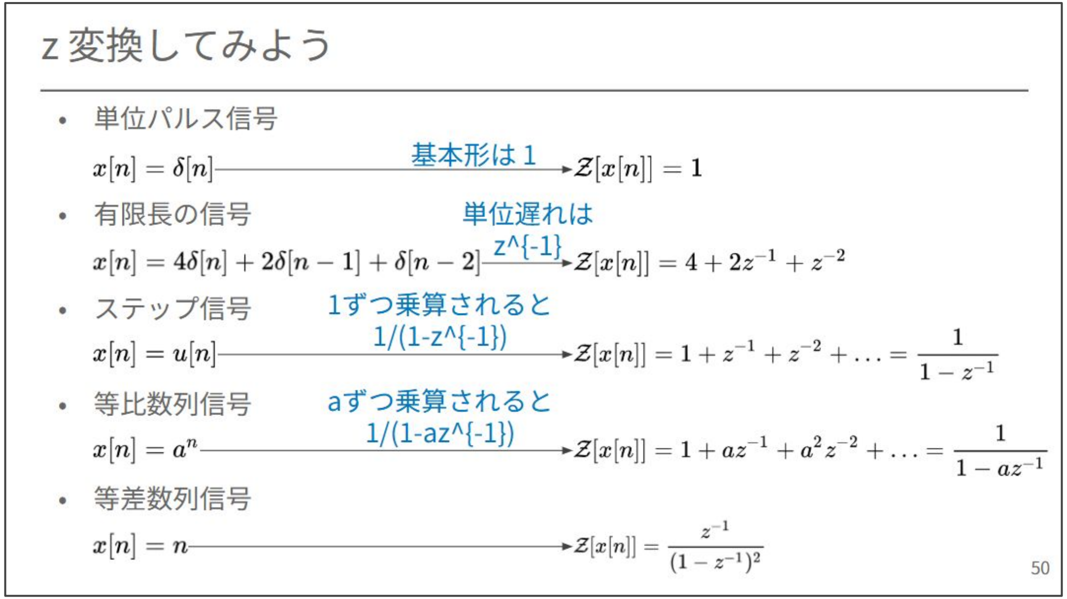
$$X(z) = \frac{7z^2 - 10z}{z^2 - 3z + 2}$$

$$X(z) = \frac{7 - 10z^{-1}}{1 - 3z^{-1} + 2z^{-2}}$$

$$X(z) = \frac{7 - 10z^{-1}}{(1 - 1z^{-1})(1 - 2z^{-1})}$$

$$X(z) = 3\frac{1}{1 - z^{-1}} + 4\frac{1}{1 - 2z^{-1}}$$

$$x[n] = 3u[n] + 4 \cdot 2^n$$



z変換してみよう

- 単位パルス信号

$$x[n] = \delta[n] \xrightarrow{\text{基本形は1}} \mathcal{Z}[x[n]] = 1$$

- 有限長の信号

$$x[n] = 4\delta[n] + 2\delta[n-1] + \delta[n-2] \xrightarrow[\text{z^\{-1\}}]{\text{単位遅れは}} \mathcal{Z}[x[n]] = 4 + 2z^{-1} + z^{-2}$$

- ステップ信号

$$x[n] = u[n] \xrightarrow[\text{1/(1-z^\{-1\})}]{\text{1ずつ乗算されると}} \mathcal{Z}[x[n]] = 1 + z^{-1} + z^{-2} + \ldots = \frac{1}{1 - z^{-1}}$$

- 等比数列信号

$$x[n] = a^n \xrightarrow[\text{1/(1-az^\{-1\})}]{\text{aずつ乗算されると}} \mathcal{Z}[x[n]] = 1 + az^{-1} + a^2z^{-2} + \ldots = \frac{1}{1 - az^{-1}}$$

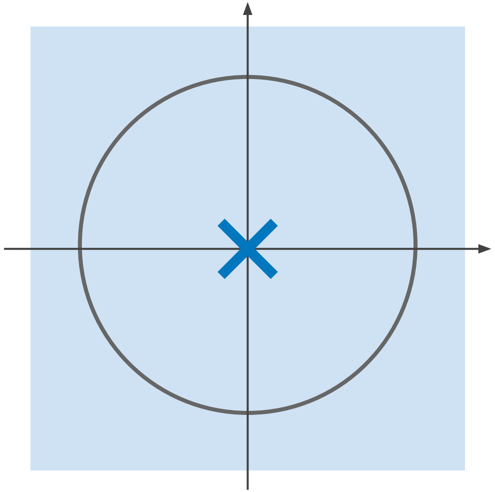
- 等差数列信号

$$x[n] = n \longrightarrow \mathcal{Z}[x[n]] = \frac{z^{-1}}{(1 - z^{-1})^2}$$

50

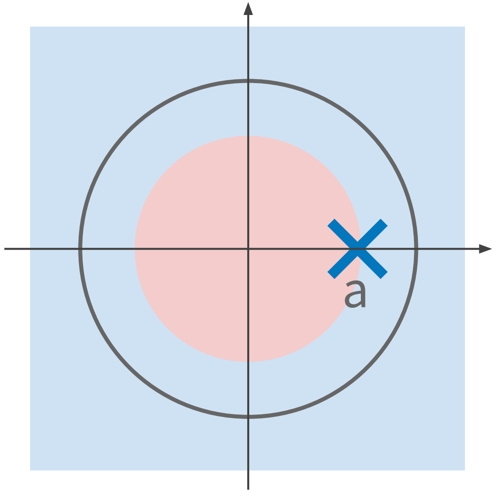
極と定義域 (収束域) について：
X(z) が無限大に発散する z が極になる．

$$\mathcal{Z}[x[n]] = \frac{1}{1 - z^{-1}}$$



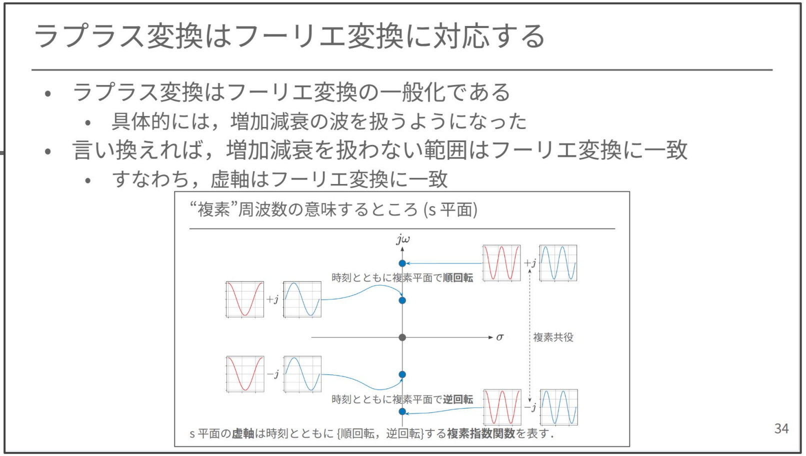
$$\mathcal{Z}[x[n]] = \frac{1}{1 - az^{-1}}$$

(0 < a < 1)



極の絶対値の外側がz変換の定義域になる。すなわち、z変換が有限の値として存在する z の範囲のこと。複数の極があるときは、一番外側の極の外側。

離散フーリエ変換との関係



- z 変換も離散フーリエ変換に対応する

$$X(z) = \sum_{n=0}^{\infty} x[n] z^{-n}$$

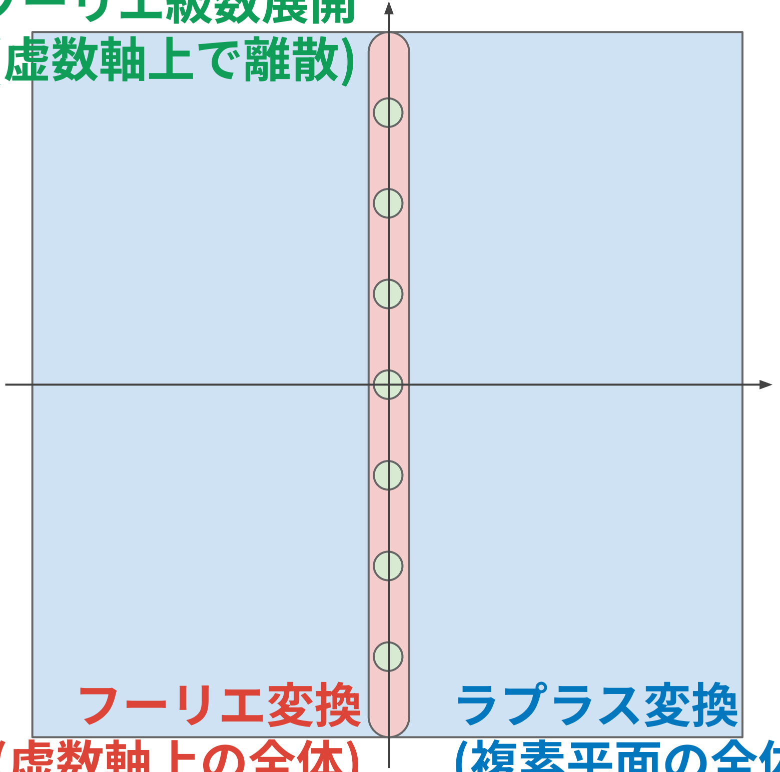
- 上式で z = e^{jω} にする
  - → すなわち単位円上だけを考えると**離散時間フーリエ変換**

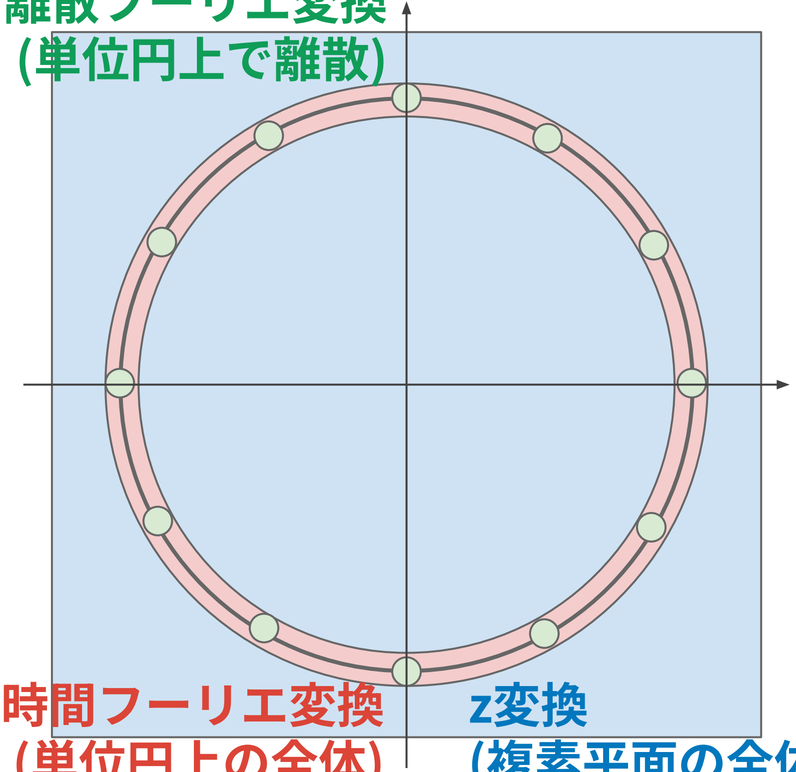
$$X(\omega) = \sum_{n=0}^{\infty} x[n] e^{-j\omega n}$$

- さらに，ω = 2π k / N で離散化して区間を限定する
  - →すなわち，単位円上を N 等分割すると**離散フーリエ変換**

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi kn}{N}}$$

離散フーリエ変換との関係



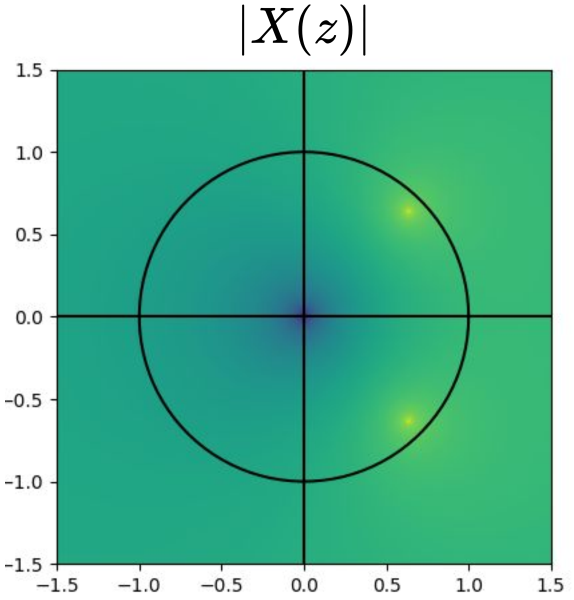


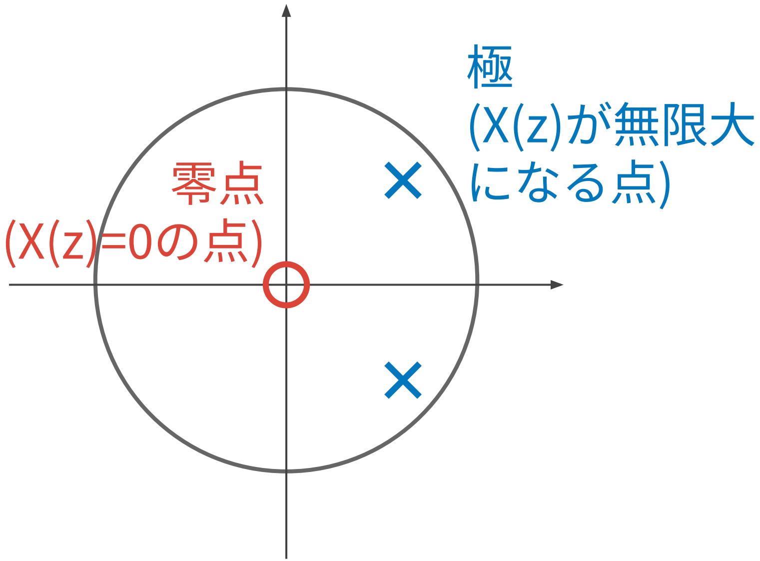
z変換の絶対値 |X(z)|

$$X(z) = \frac{1}{(1-0.9e^{j\pi/4}z^{-1})(1-0.9e^{-j\pi/4}z^{-1})}$$

|X(z)| の解析解の導出手順は概略．

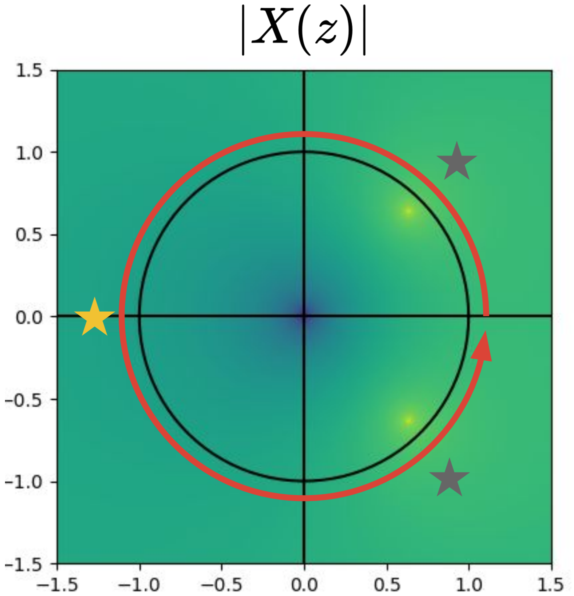
- 分母と分子に z^2 を乗算する．
- z = a + jb と置く．
- 分母と分子に，分母の複素共役を乗算して分母を実数にする

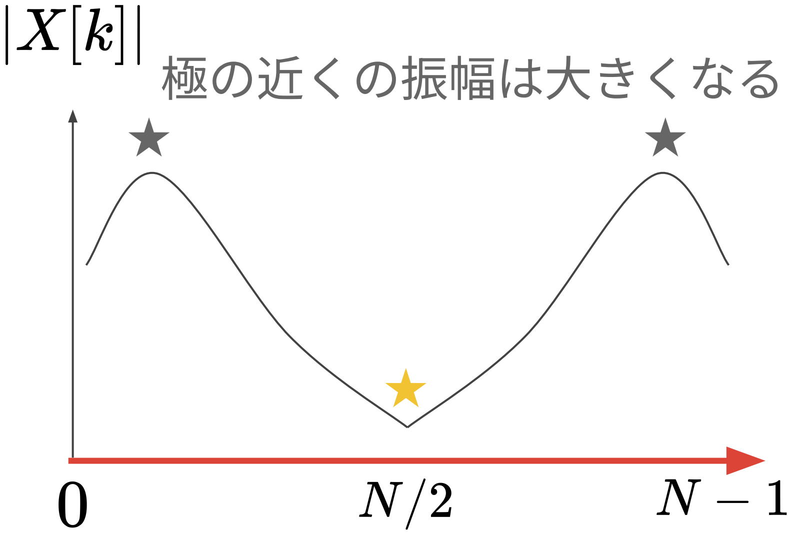




実数信号を扱う場合，極 (と零点) は実軸で線対称になる

z変換の振幅がわかれば，離散フーリエ変換の振幅スペクトルの概形がわかる





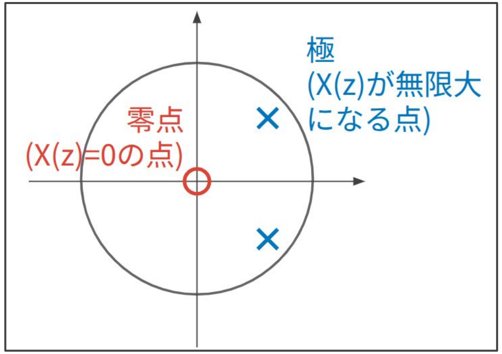
{離散時間, 離散}フーリエ変換の振幅スペクトルは単位円上の絶対値に等しい

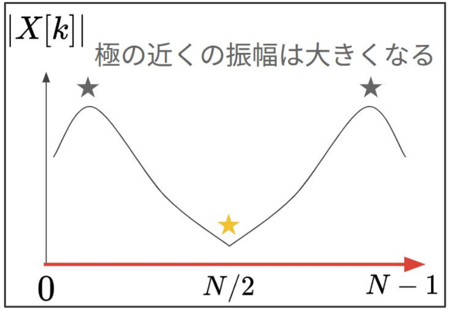
演習 (残りは課題)

- 次の離散時間信号について
  - z 変換を求めよ．
  - 極の位置を描画せよ．
  - 収束域を答えよ．
  - 極の位置から，離散フーリエ変換の振幅スペクトルの概形を描画せよ．

$$x[n] = \sum_{k=0}^{\infty} \left(\frac{1}{2^8}\right)^k \delta[n-8k]$$

(8サンプルごとに非零の値になり値が 1/2^8 ずつ小さくなる等比数列)



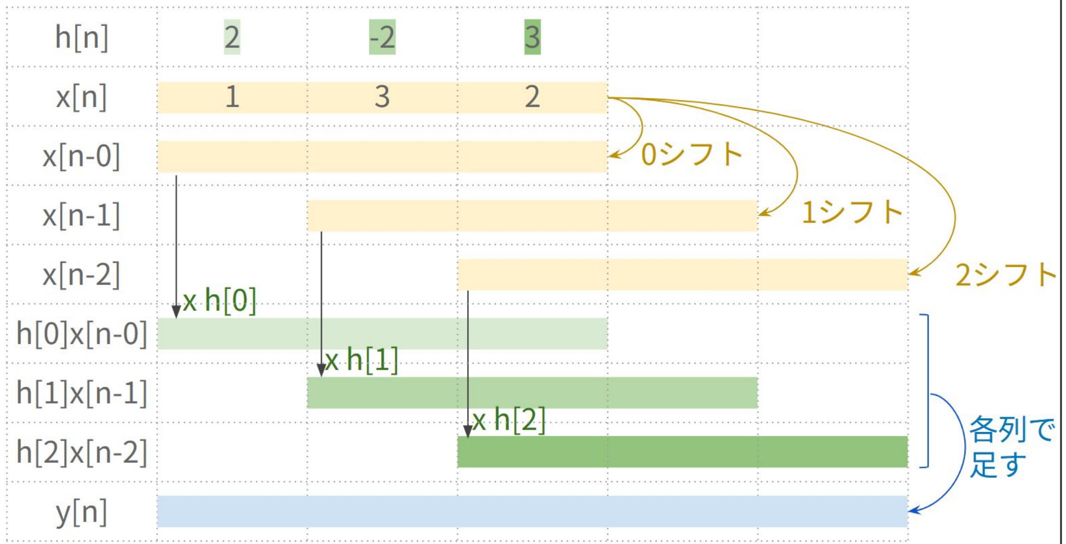


- ヒント
  - 等比数列の形（○^k）の形に変形する．
  - 極は実数解だけでない．複素解析の授業を思い出すこと．
  - 収束域は，最も外側にある極の外側．

時間領域の畳み込み演算は z 領域で乗算になる (これはフーリエ変換，ラプラス変換でも共通する)

$$\mathcal{Z}\left[h[n] * x[n]\right] = H(z)X(z)$$

x = [1, 3, 2], h = [2, 2, 3] とする．畳み込み結果 y [n] を求めよ．



$H(z)$

$X(z)$

$X(z)z^{-0}$

$X(z)z^{-1}$

$X(z)z^{-2}$

$h[0]X(z)z^{-0}$

$h[1]X(z)z^{-1}$

$h[2]X(z)z^{-2}$

$$\underbrace{(h[0]z^{-0} + h[1]z^{-1} + h[2]z^{-2})}_{H(z)}X(z)$$

第1回のスライドから

まとめ
(matome)

まとめ

- ラプラス変換
  - ラプラス変換は，時間減衰関数を導入することで，フーリエ変換できなかった信号も変換できるようにしたもの
  - フーリエ変換で使われる振動だけでなく，信号の増減も表せるもの
  - 減衰関数の強さで信号の増減を解析できるもの

- z変換
  - z 変換は，ラプラス変換の離散時間信号版
  - ラプラス変換と同様に，信号の増減を表現できる

課題
(exercise)

以下の課題についてレポートを作成し，LMS 上で提出せよ。

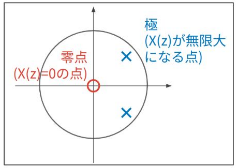
演習 (残りは課題)

- 以下の関数についてX(s) の実部・虚部をプロットし、その形状を比較・考察せよ。a, ω_0 には適当な値を使用せよ。
  - ステップ関数 ← 前のページのやつ
  - 指数関数 (a > 0)
  - 正弦波関数 (ω_0 > 0)
- sympy の関数
  - numpy と同じように定義されているケースが多い
  - sp.exp(), sp.sin(), sp.pi
- プロットする s の範囲を適宜変更せよ。

29

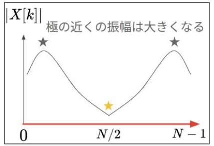
演習 (残りは課題)

- 次の離散時間信号について
  - z 変換を求めよ．
  - 極の位置を描画せよ．
  - 収束域を答えよ．
  - 極の位置から，離散フーリエ変換の振幅スペクトルの概形を描画せよ．



$$x[n] = \sum_{k=0}^{\infty} \left(\frac{1}{2^8}\right)^k \delta[n-8k]$$

(8サンプルごとに非零の値になり
値が 1/2^8 ずつ小さくなる等比数列)



- ヒント
  - 等比数列の形（〇^k）の形に変形する．
  - 極は実数解だけでない．複素解析の授業を思い出すこと．
  - 収束域は，最も外側にある極の外側．

56