

情報工学科 講義 デジタル信号処理A (2025/04/10)

第1回 イントロダクション，デジタル信号

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

イントロダクション

この講義 (ディジタル信号処理A) について

サブタイトル アナログからディジタルへ、そして、その応用へ。その根本的な理解をめざします。

担当者名 高道 慎之介

単位 1

年度・学期 2025 春(学期前半)

曜日時限 木3

キャンパス 矢上

授業実施形態 対面授業 (主として対面授業)

登録番号 00733

設置学部・研究科 理工学部

設置学科・専攻 情報工学科

学年 3

科目概要

画像や音声、データなどあらゆる「情報」の多くは「信号」の形として扱われる。本講義ではフーリエ変換や離散フーリエ変換、 z 変換などの信号処理の基礎から、伝達関数や周波数応答などの離散時間システムについての解説を行う。



世の中にあふれる (デジタル) 信号処理



Hearing Aids



教員の紹介



高道 慎之介 (たかみち しんのすけ)
情報工学科 准教授 (2024-)

専門：
音声合成変換・信号処理など

高道研 学部生の研究テーマ：

- 音楽心理分析，ゲーム音声実況，音声による行動変容 (2024)
- 方言音声処理，環境音合成，マンガ情報処理，合唱合成，アカペラ合成，音声による行動変容 (2025)

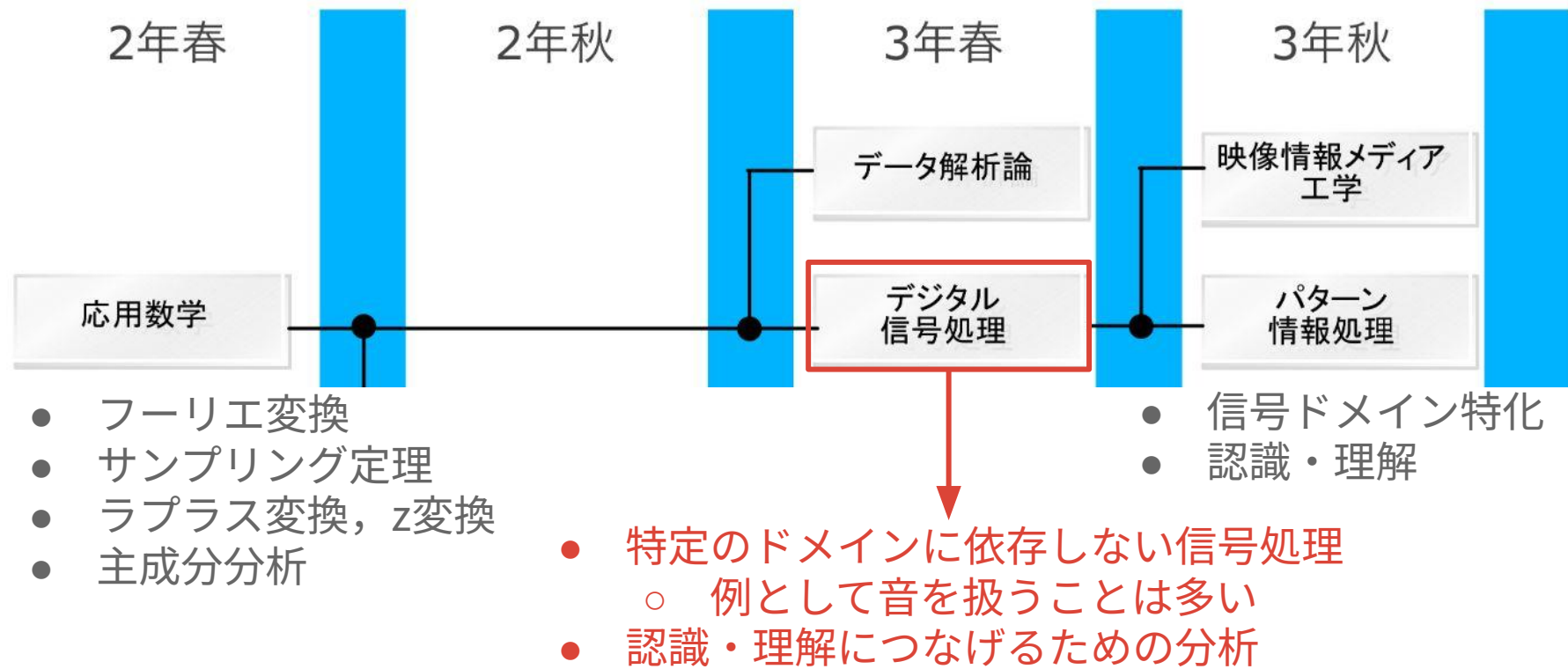
成績の付け方

- 内訳
 - 30 / 100 … 講義での演習
 - 70 / 100 … 期末試験 (A/Bそれぞれなので注意)
- 演習
 - 毎講義の終わりに課題を出します。提出は K-LMS 上。
 - 締切は次の講義の前日 23:59。詳細は各講義の演習内容を参照。
- 期末試験 (筆記試験)
 - 演習の内容を理解していれば半分は取れるよう設計するつもり。
 - 本講義の内容が皆さんの人生に (いつか) 資すると思うなら、上の点数を目指すこと。

基盤モデル (大規模言語モデルなど) の使用

- 基盤モデルとは.
 - ChatGPT, Google notebookLM, etc.
- 活用は妨げない.
 - 演習課題で利用した場合は, 使用した旨を明記し, 生成履歴 (例えばスクショ) を記載せよ. 利用すること自体による加減点は無い.
 - その生成履歴を踏まえて, 自分の言葉で回答せよ.
- **自分 (人間) を成長させるために, 講義も基盤モデルも利用せよ.**
 - スクショを残すことで, 自分の理解の過程を反芻してほしい.
 - 成長を期末試験で問う (期末試験では基盤モデル使用は不可).

講義の関係性





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)	応用数学の復習
第01回	2025/04/11	イントロダクション, ディジタル信号処理	
第02回	2025/04/18	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	2025/04/25	ラプラス変換から z 変換へ	
第04回	2025/05/02	インパルス応答と伝達関数, 安定性	
第05回	2025/05/09	ディジタルフィルタ	昨年のBの途中まで
第06回	2025/05/16	高速フーリエ変換と短時間フーリエ変換	
第07回	2025/05/23	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)	


- シラバスの内容と少し変わっています.
- 信号処理 B: 前半までで昨年の内容を終え, 後半で新規内容の予定.


講義資料と演習課題は高道研 github にある。 (K-LMSにも同じものをアップロード)


 [takamichi-lab](#) / [digitalsignalprocessing-keio](#) Public

 Notifications


[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)

 main


 1 Branch


 0 Tags

[Code](#)

 **vebmaylrie** Update readme.md ✓


fe80ca8 · 10 months ago

 32 Commits

 2024

Add files via upload


10 months ago

 readme.md

Update readme.md

10 months ago

README



デジタル信号処理 B [矢上 14-204]

慶應義塾大学 理工学部 講義

2024年度 (01~06はデジタル信号処理Aの講義を参照)

- 第07回 :
 - 講義資料(FFT)

[Readme](#)

[Activity](#)


[Custom properties](#)

[10 stars](#)

[1 watching](#)

[0 forks](#)

[Report repository](#)



<https://github.com/takamichi-lab/digitalsignalprocessing-keio>

数式の気持ちを理解しよう！

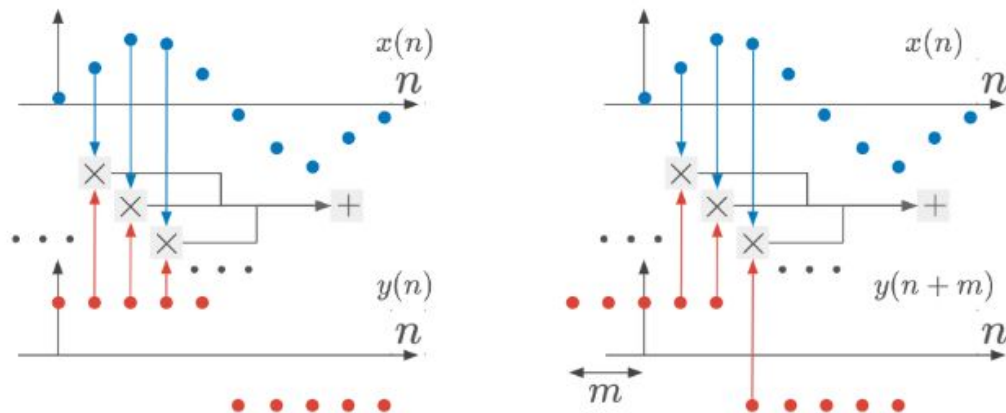
言葉

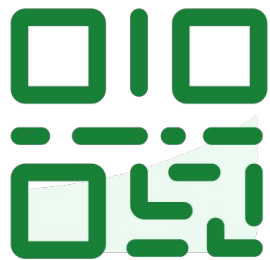
文字面を追うのではなく、
その式は何を意味するのか？
言い換えると？図示すると？
簡単な数字を当てはめると？

(系列) の一致度を測る関数である。この関数は、片方の信号に対してずらした時間を
返す。2つの異なる信号 x, y 間で測る相関関数を**相互相関関数**という。

$$\phi_{xy}(m) = \frac{1}{N} \sum_{n=0}^{N-1} x(n)y(n+m) \quad (3)$$

m はずらした時間値である。この式は、信号 y を時刻 m だけずらした時の、信号同士を掛けて総和したもの (= 式(1))
である。イメージは以下のとおり。





Join at **slido.com**
#3271892





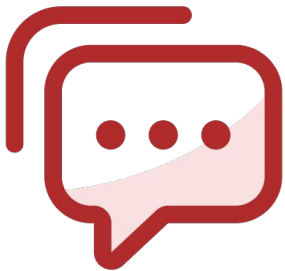
応用数学（齋藤英雄先生）を受講した？





信号処理について理解できていることは？





Audience Q&A



Presenting with animations, GIFs or speaker notes? Enable our [Chrome extension](#)

質問の方法

- 授業中 (授業直後) の方法：
 - slido を使って質問する
 - 口頭で質問する
- 授業外の方法：
 - K-LMS で質問する

本日の内容

デジタル信号とアナログ信号の変換



物理世界にあるのはアナログ信号。それをデジタル化してPCに取り込んでいる。
そこには数学が使われている。それを理解しよう。

本日の内容

- 数学的準備
 - これからの講義で使用する数を定義します.
- デジタル信号とアナログ信号の相互変換
 - まずは概念を理解し，その後に数式を理解しよう.
- (時間があれば) 畳み込みの計算.
 - 畳み込みとは何ぞや？ 計算してみる.

数学的準備：
これからの講義で使用する数を定義します.

変数，定数

表記	数	表記ルール
a, A	スカラ	イタリック
\mathbf{a}	ベクトル	イタリック，ボールド，原則小文字
\mathbf{A}	行列	イタリック，ボールド，大文字

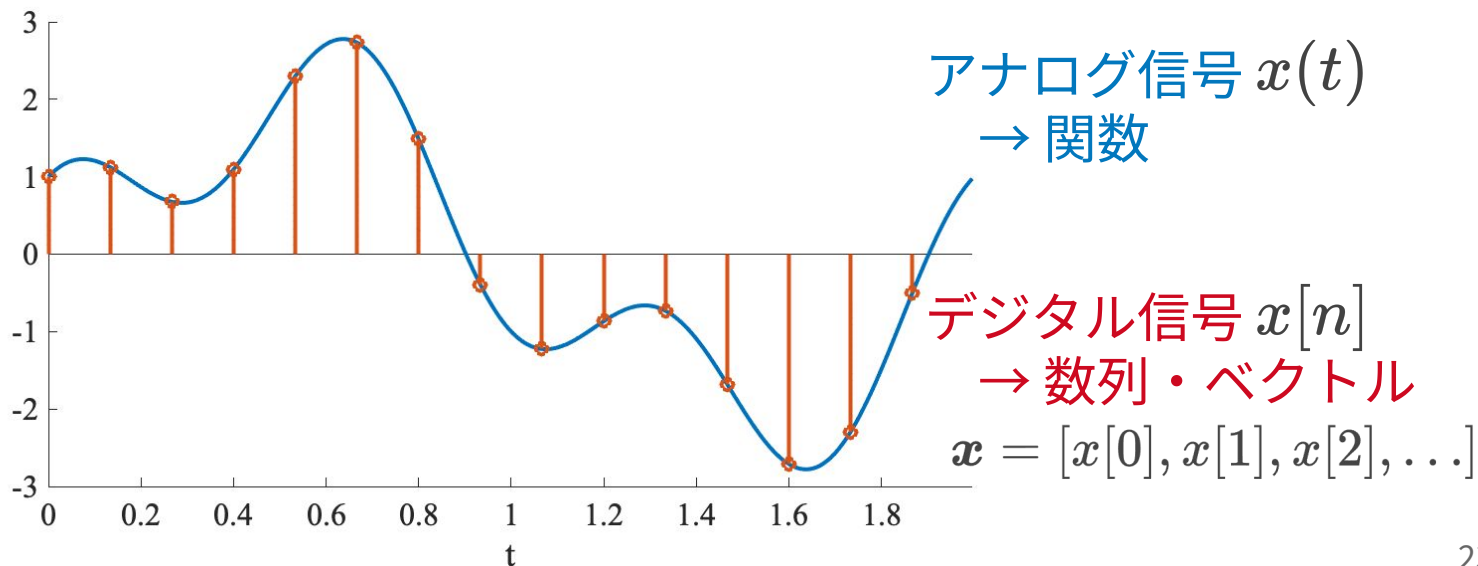
表記	意味
e	自然対数の底
π	円周率
j	虚数単位 (信号処理では慣習的に i を用いない)
\mathbf{I}_M	$M \times M$ 単位行列
$\mathbf{0}_{M \times N}$	$M \times N$ 零行列

複素数 $a + jb$ or $r \exp(j\theta)$

表記	意味	
$\Re(z)$	複素数の実部	$\Re(a + jb) = a$
$\Im(z)$	複素数の虚部	$\Im(a + jb) = b$
$ z $	複素数の絶対値	$ r \exp(j\theta) = r$
$\arg(z)$	複素数の偏角	$\arg(r \exp(j\theta)) = \theta$

信号 (詳細は後述)

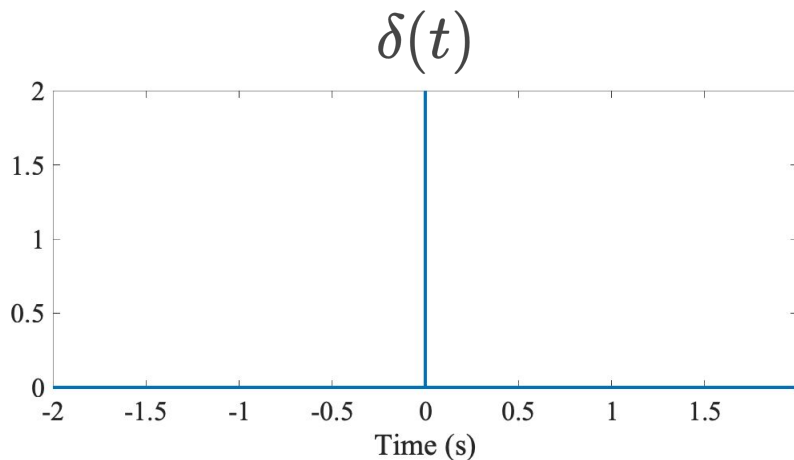
表記	意味
$x(t)$	連続時間信号 (アナログ信号). t は実数.
$x[n]$	離散時間信号 (デジタル信号). n は整数.



デルタ関数 $\delta(t)$ or $\delta[n]$

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

※厳密には正しい記述ではない．詳細は「超関数」

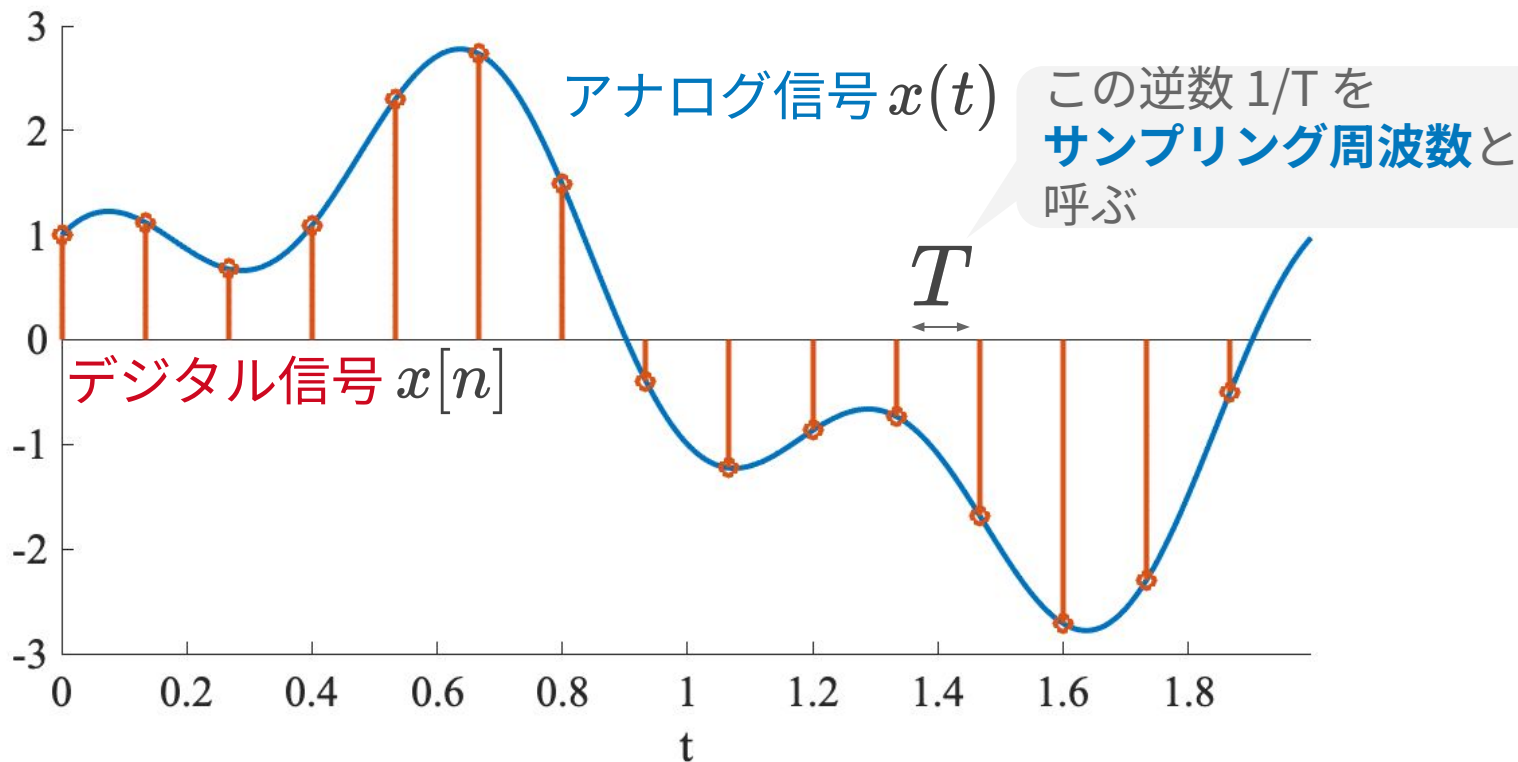


$$\int_{-\infty}^{\infty} \varphi(t) \delta(t) dt = \varphi(0)$$

ある関数 (波形) の瞬時的な値を切り出す役割を持つ．上記の例では $t=0$ の値．

デジタル信号からアナログ信号へ

ディジタル信号からアナログ信号へ

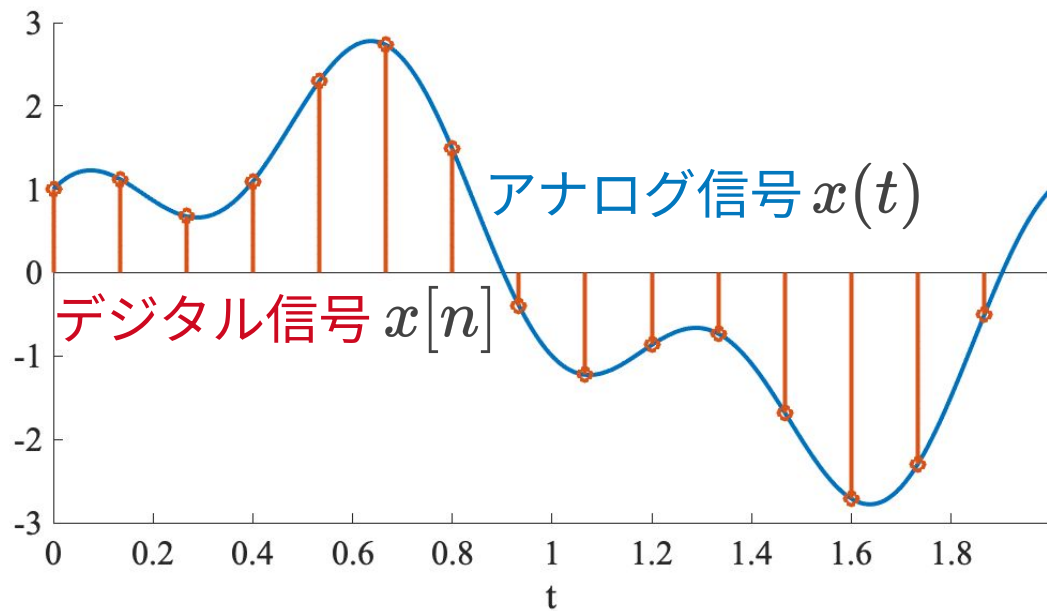


アナログ信号を周期 T でサンプリングしたものをディジタル信号として考える。
この周期を**サンプリング周期 (sampling period)** という。

数式として書くと

$$x[n] = x(nT) \longrightarrow x[n] = x(0)\delta[n] + x(T)\delta[n-1] + x(2T)\delta[n-2] + \dots$$

離散時間信号のデルタ関数を使って書くと





サンプリング周波数が 4 kHz
のとき、サンプリング周期はい
くらになるか？



サンプリング周波数が 1 kHz
で、1000サンプルをもつ離散
時間信号の時間長はどれか？



ある信号を2秒間サンプリングし、サンプリング周波数が500 Hzであった。この信号のサンプル数はいくつか？



さらなるクイズ：あるアナログ信号 $x(t)$ について, $x[0]$, $x[1]$, $x[2]$... を求めてみよう.

$$x[n] = \underbrace{x(0)}_{x[0]} \delta[n] + \underbrace{x(T)}_{x[1]} \delta[n-1] + \underbrace{x(2T)}_{x[2]} \delta[n-2] + \dots$$

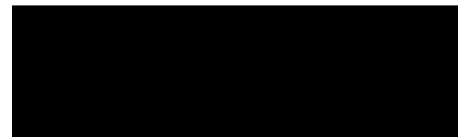
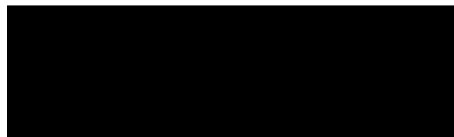
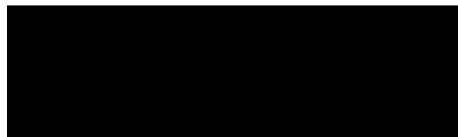
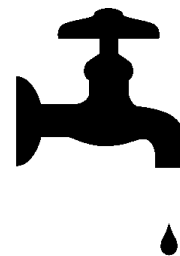
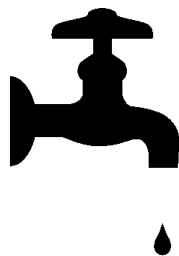
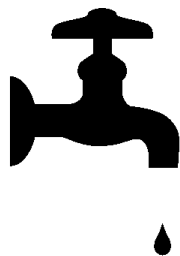
次のページで具体的なアナログ信号を提示します.



サンプリング周波数を 1000 Hz とする. 2つのアナログ信号 $x_1(t) = \sin(2\pi * 100t)$, $x_2(t) = \sin(2\pi * 1100t)$ のそれぞれについて, $x[0]$, $x[1]$, $x[2]$ を求めよ.

サンプリング周期の違う 3 つのデジタル信号を考える

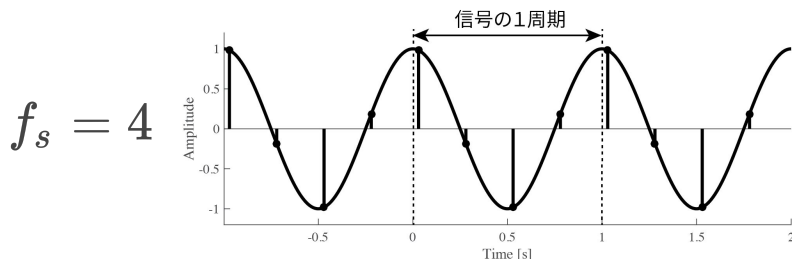
ストロボの周期：0.5秒 ストロボの周期：1.0秒 ストロボの周期：1.5秒



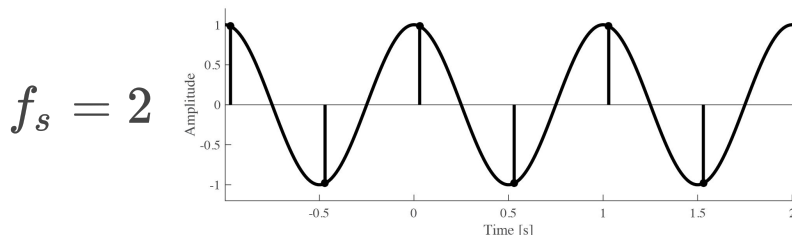
すべて水滴のたれるアナログ信号のはずだが、右だけ水滴が上がるように見える

信号の周波数とサンプリング周期の関係を 具体的な数値を入れて定性的に確認してみよう。

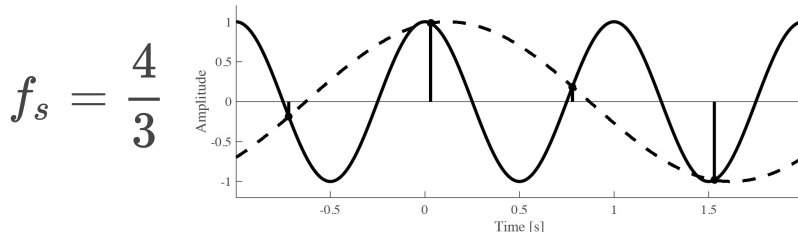
- 信号の周期を1[sec], サンプリング周波数を f_s [Hz]とする
 - サンプリング周波数は、ストロボの周期の逆数に対応



良さそう (サンプリング点が連続
時間信号をよく表している)



良さそう (サンプリング点が連続
時間信号をよく表している)

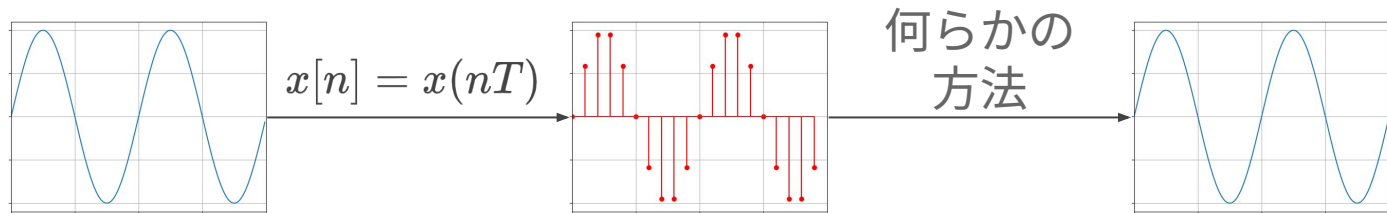


実線の信号と破線の信号で同じサ
ンプリング点になっている
(区別できない)

サンプリング定理

サンプリング定理

連続時間信号 $x(t)$ の最大周波数の上限が f_0 のとき、サンプリング周波数が $2f_0$ 以上でサンプリングした離散時間信号 $x[n]$ から $x(t)$ を完全に復元できる。



最大周波数 f_0 サンプリング周波数が $2f_0$ 以上 元の信号を復元可能

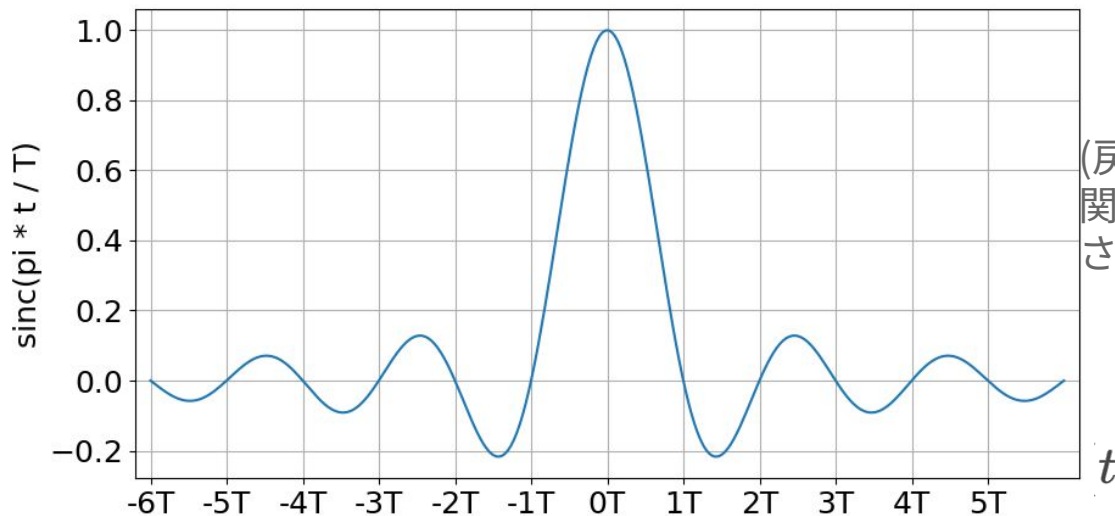
- Q. $2f_0$ の 2 はどこから来た？
 - A. フーリエ変換を理解できれば分かる．次回以降に説明．
- Q. サンプリング定理を満たしたいけど、最大周波数が分からない…
 - A. 最大周波数がサンプリング定理を満たすように、アンチエイリアジングフィルタというローパスフィルタを連続時間信号にかける．次回以降．

ディジタル信号からアナログ信号にどう戻す？

$$x(t) = \sum_{n=-\infty}^{\infty} x[n] \text{sinc} \left[\pi \left(\frac{t}{T} - n \right) \right]$$

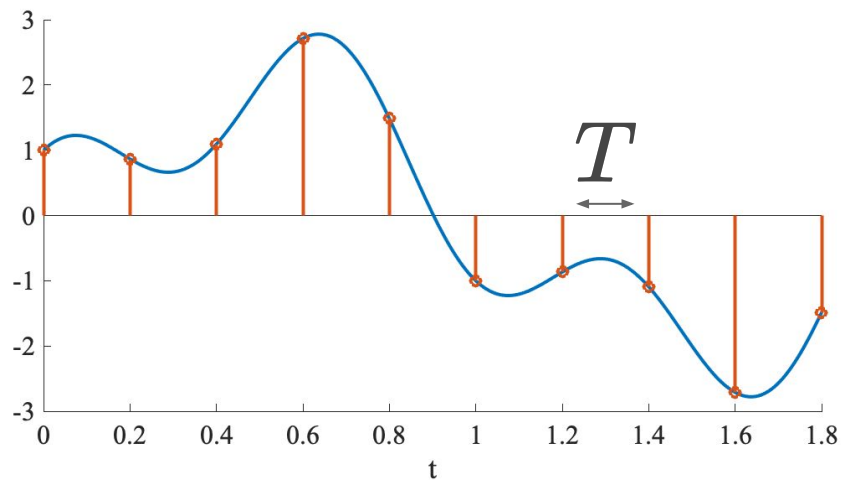
$$\text{sinc}(x) = \frac{\sin x}{x}$$

sinc 関数の係数が デジタル信号の時刻 n に応じて
デジタル信号の値 シフトされる sinc 関数



(戻すときの帯域制限の関数からsinc関数が導出される．どこかでやるかも)

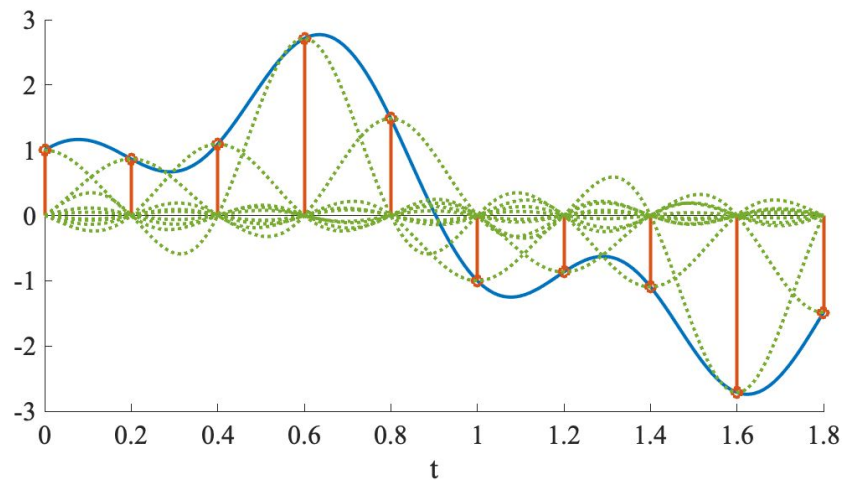
デジタル信号からアナログ信号へ



$$x[n] = x(nT)$$

アナログ信号 → デジタル信号

サンプリング周期 T で取り出す



$$x(t) = \sum_{n=-\infty}^{\infty} x[n] \text{sinc} \left[\pi \left(\frac{t}{T} - n \right) \right]$$

デジタル信号 → アナログ信号

sinc 関数で畳み込み

畳み込み

ギターのリバーブ (響き効果) を考える.



信号をある(線形時不変)システムに入れたときの出力を求める→ **畳み込み演算**

畳み込み：2つの{関数, 数列}から別の{関数, 数列}を求める操作

- 連続系 (アナログ信号)

$$\begin{aligned} y(t) &= h(t) * x(t) = \int_{-\infty}^{\infty} h(\tau)x(t - \tau)d\tau \\ &= \int_{-\infty}^{\infty} h(t - \tau)x(\tau)d\tau \end{aligned}$$

- 離散系 (デジタル信号)

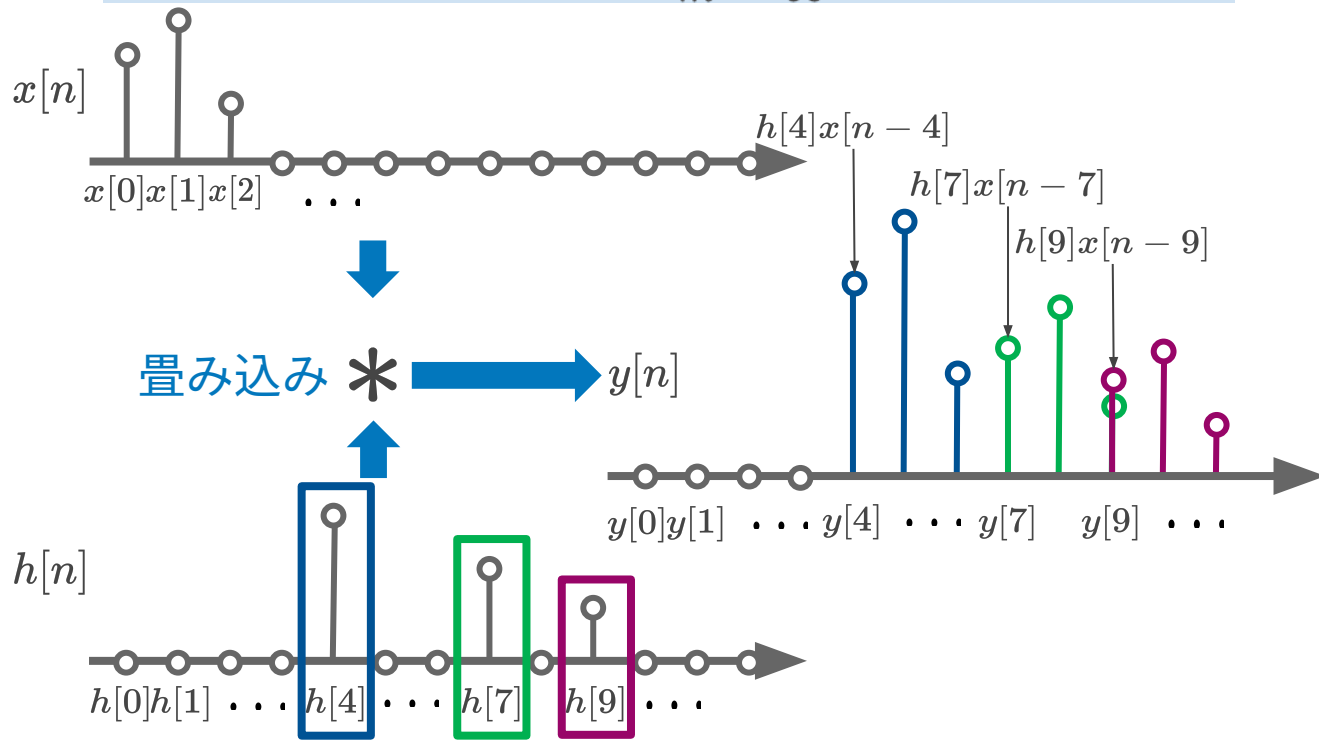
畳み込み演算子

$$\begin{aligned} \underline{y[n]} &= \underline{h[n]} * \underline{x[n]} = \sum_{m=-\infty}^{\infty} h[n - m]x[m] \\ \text{出力} & \quad \text{システム} \quad \text{入力} \\ & \quad \text{応答} \quad \text{信号} \\ &= \sum_{m=-\infty}^{\infty} h[m]x[n - m] \end{aligned}$$

本講義では
こっちだけ扱う

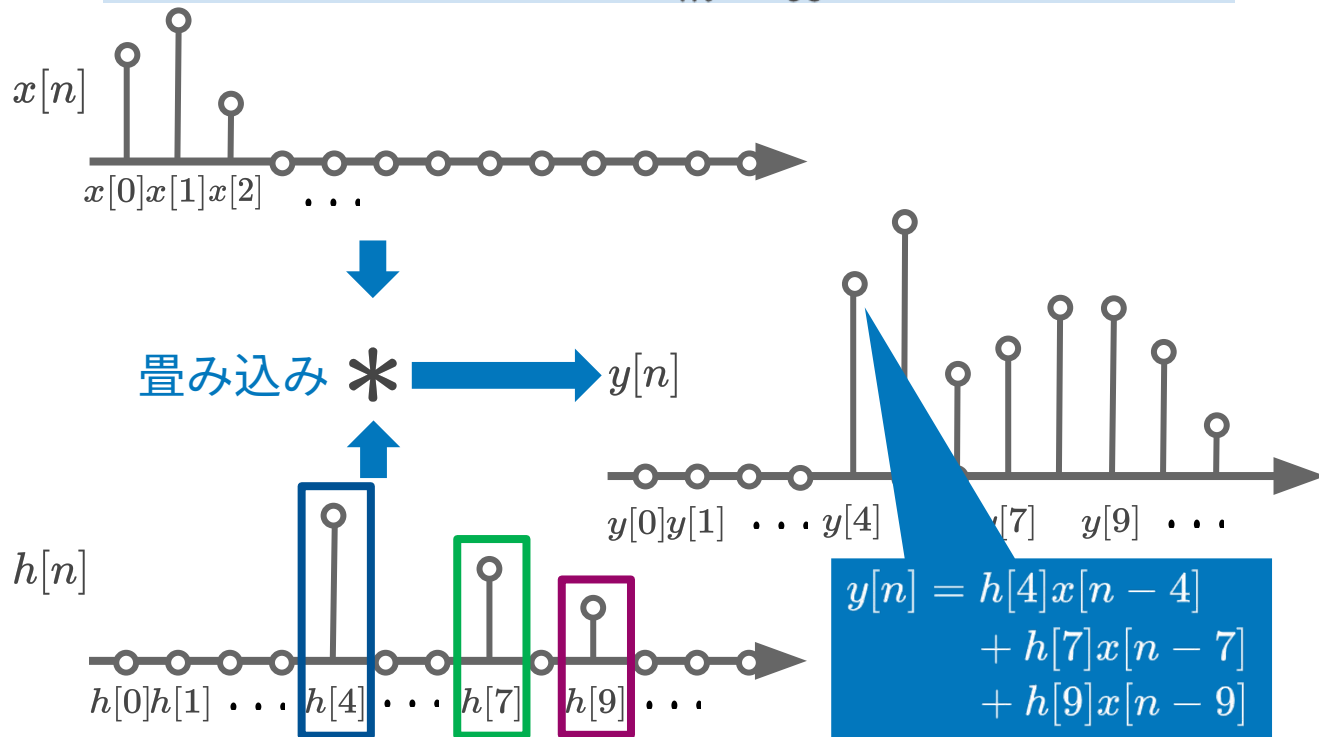
畳み込みのイメージを捉えよう

$$y[n] = h[n] * x[n] = \sum_{m=-\infty}^{\infty} h[m]x[n-m]$$



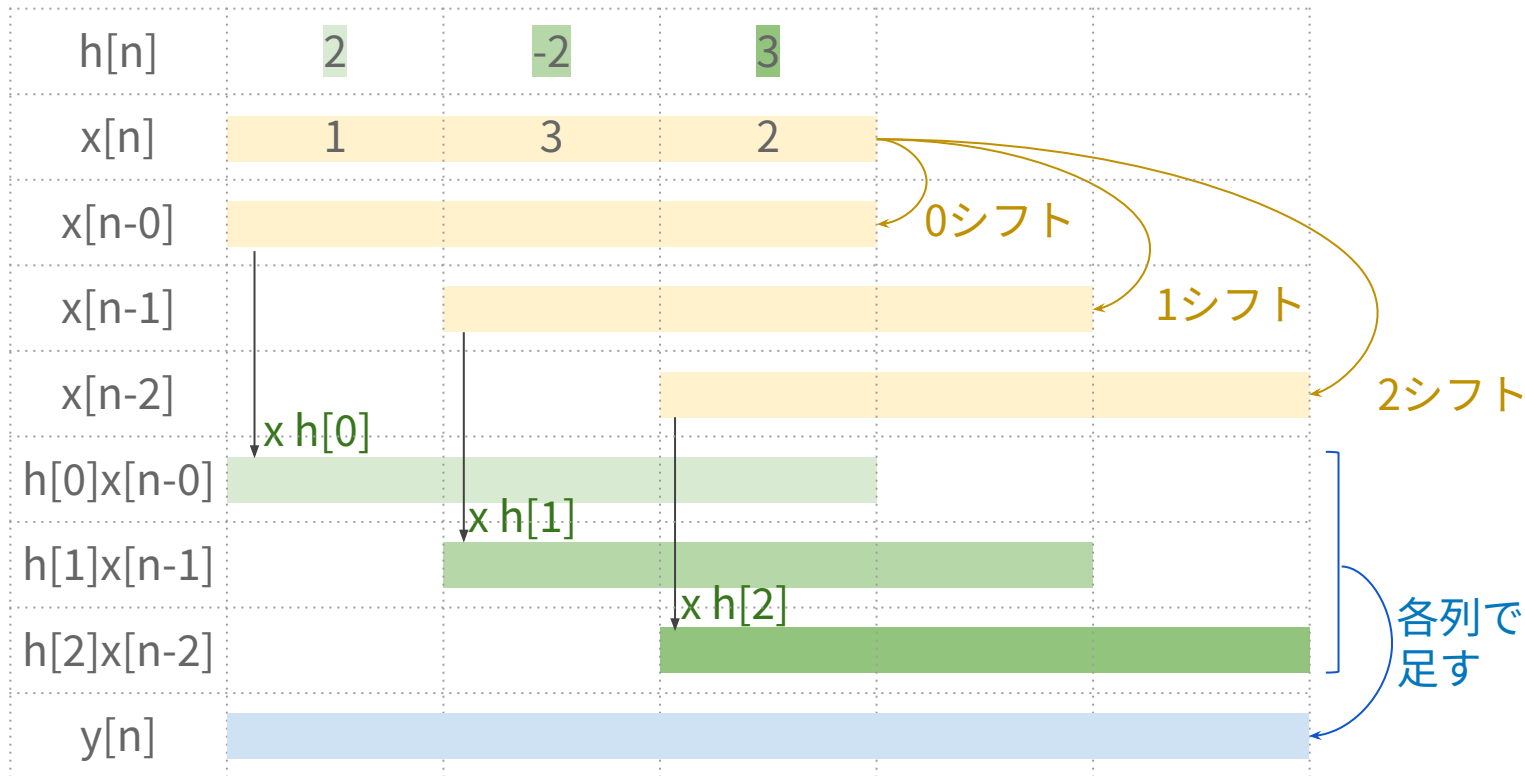
畳み込みのイメージを捉えよう

$$y[n] = h[n] * x[n] = \sum_{m=-\infty}^{\infty} h[m]x[n-m]$$



[演習] 手計算してみよう

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





畳み込み演算の演習

課題

以下の課題についてレポートを作成し、 LMS 上で提出せよ。

- 以下の問題について、その過程も含めて回答せよ。回答は PDF or DOCX 形式とする。手書き or PC利用は問わない。
1. サンプリング周波数を 250 Hz とする． 750 サンプル (点) のデジタル信号の時間長は何秒か．
 2. デジタル信号からアナログ信号を復元するプログラムがある．
この抜けている箇所を穴埋めし，プログラムと，復元できる旨を示す図を示せ．
 3. $x = [3, 0, 1, 0, 2]$, $h = [4, 2, 1]$ とする．これらの畳み込み結果を示せ．

課題 2 のプログラム

```
import math
import matplotlib.pyplot as plt

T = 0.1 # sampling period [sec]
n_points = 20 # number of points of the digital signal

# analog signal: sin(2 * pi * t)
x_t = lambda t: math.sin(2 * math.pi * t)

# digital signal: x[n] = x(n * T), n = 0, 1, ..., n_points-1
x_n = [x_t(n * T) for n in range(n_points)]

def reconst_x(t: float) -> float:
    def _sinc(z: float) -> float: # sinc function
        return 1 if abs(z) < 1e-10 else math.sin(z) / z

    x = 0
    """
    code here and return reconstructed x(t) using x_n (2行くらいでかけるはず)
    """

    return x

# plot the signals
t_plot = [n * T / 10 for n in range(n_points * 10)]
plt.plot(t_plot, [x_t(t) for t in t_plot], '-', label='Original signal')
plt.stem([n * T for n in range(n_points)], x_n, "r", label='Digital signal')
plt.plot(t_plot, [reconst_x(t) for t in t_plot], '--', color="k", label='Reconstructed signal')
plt.xlim(0, n_points * T)
plt.ylim(-1.1, 1.1)
plt.xlabel("t [sec]")
plt.grid()
plt.legend()
#plt.savefig("reconstruction.png") # uncomment to save the figure
```