

# 3. 音声合成

情報学群実験第 3C 3i

門屋陽丈 谷保愛華 内藤熙人 成岡小雪 平林里菜 三上柊 溝口洸熙

Group 10

April 20, 2023

# お品書き

## 1. 初期位相・矩形波

## 2. ノコギリ波

## 3. 課題 1

問題

サンプルコード

結果：ノコギリ波

結果：矩形波

## 4. モノラルとステレオ

## 5. 課題 2

結果

サンプルコード

## 6. 周波数解析

ノコギリ波

振幅スペクトル

fft

fftshift

abs

周波数テーブルの作成

## 7. 課題 3

サンプルコード：解析

サンプルコード：合成

# 1. 初期位相・矩形波

周波数  $f$ , 時刻を  $t$  に設定する.

## 初期位相

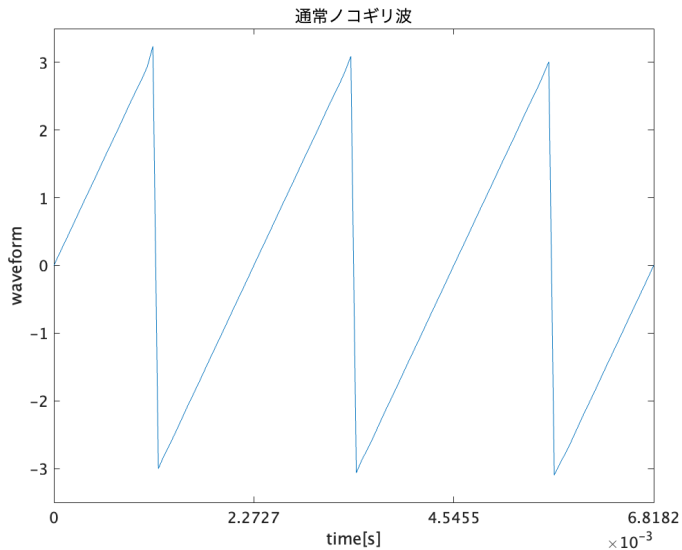
初期位相を  $\phi$  に設定する.

$$y(t) = \sin(2\pi ft + \phi) \quad (1)$$

## 矩形波のフーリエ級数展開

$$y(t) = \sum_{k=1}^{\infty} \frac{1}{2k-1} \sin(2\pi(2k-1)ft) \quad (2)$$

## 2. ノコギリ波



## 2. ノコギリ波

$$y(t) = t \quad (-\pi < t < \pi) \quad (3)$$

を周期  $2\pi$  の関数として周期的に拡張したもの.

$$y(t) = y(t + 2k\pi) \quad (k \in \mathbb{Z}) \quad (4)$$

### ノコギリ波のフーリエ級数展開

$$y(t) = \sum_{k=1}^{\infty} (-1)^{k-1} \frac{2}{k} \sin(kt) \quad (5)$$

$\sum_{k=1}^{\infty}$  の  $\infty$  は今回, 50 としてプログラミングしている.

### 3. 課題 1

矩形波・ノコギリ波を基本周波数を 440Hz 等の可聴域の範囲で作成し、さらに各周波数成分も位相を適当な値に変化させよう。  
(サンプリング周波数  $F_s = 16\text{kHz}$ , 長さ 2s.)

#### ▶ 位相の操作

- ▶ 固定値  $\pi/4$
- ▶ 固定値  $\pi/2$
- ▶ ランダム値\*

---

\*ランダム値は `variable = rand` で格納できる.

#### Tips

```
axis([xmin xmax ymin ymax]);
```

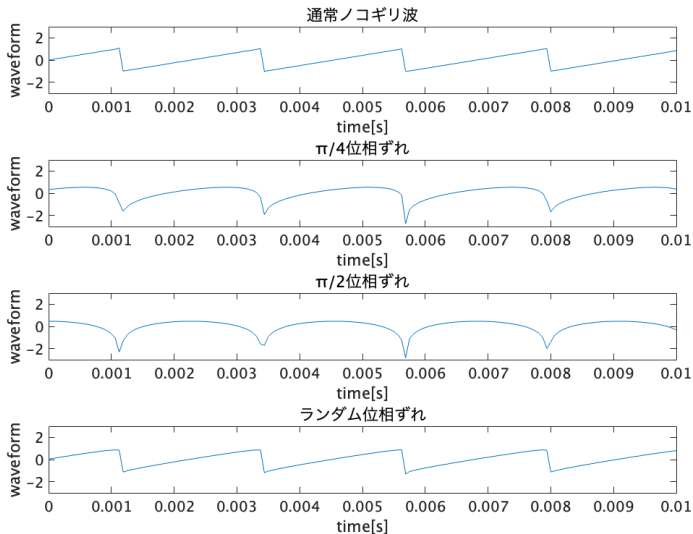
グラフをプロットするときの範囲を指定できる. 周波数  $f$  の周期関数を  $n$  周期分プロットしたい場合は `xmin` を 0, `xmax` を  $n/f$  に設定する.

### 3. 課題1 (サンプルコード)

```
1  clear all;
2  Fs = 16000; % サンプリング周波数
3  f = 400;    % 基本周波数
4  t = [0 : ??] /Fs % 時間軸テーブル
5  phi1 = pi / 4; % 初期位相 pi/2
6  phi2 = pi / 2; % 初期位相 pi/4
7  % - phi3 = rand; % 初期位相 ランダム
8  % --- ノコギリ波生成 ---
9  for k=1:50 % とりあえず50にでも設定しておく
10     y1 = ?? + (-1)^(k-1) * 1/3 * 2/k * sin(?);
11     y2 = ...;
12     y3 = ?? + (-1)^(k-1) * 1/3 * 2/k * sin(? + rand);
13 end
14 ...
```

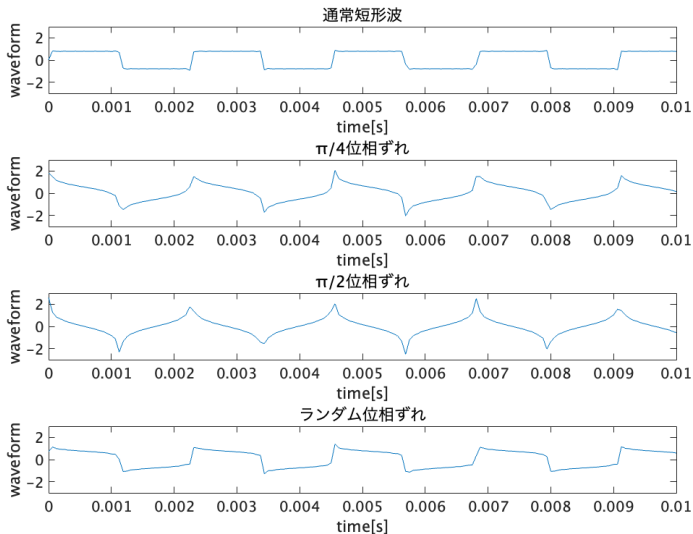
??は、『何かしら入る』という意味なので、いろいろなものが入る。(変数1つとは限らない)

### 3. 課題1 (結果：ノコギリ波)





### 3. 課題1（結果：矩形波）



## 4. モノラルとステレオ (audioread)

`audioread` 関数は, `filename.wav` のサンプルデータ `y` と, サンプリング周波数 `Fs` を保存する関数である.

---

```
[y, Fs] = audioread('filename.wav');
```

---

▶ モノラルオーディオの場合  
`y` は  $N$  行 1 列のデータ列

▶ ステレオオーディオの場合  
`y` は  $N$  行 2 列のデータ列

ステレオからモノラルに変換するには, `y` の 2 列目を削除すれば良い.

## 5. 課題2

自分の母音の音声を 4s 程度ずつ録音し，その音声データの波形の上下を反転さよう．元データと反転後のデータを聴き比べよう．

図 1: それぞれのグラフ

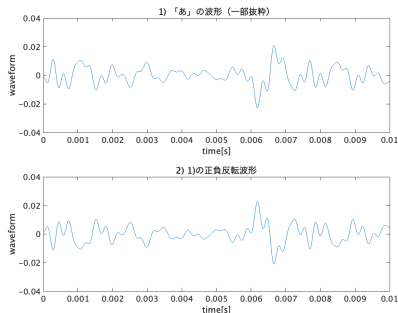
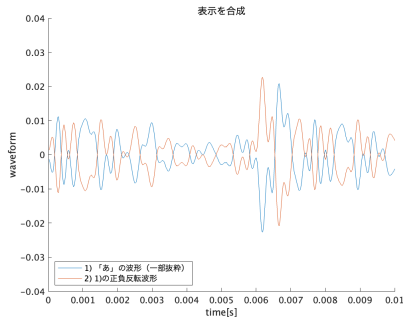


図 2: グラフを重ね合わせてみる



## 5. 課題2 (サンプルコード)

```
1 clear;
2 [y, Fs] = audioread('sound1.wav');
3 y=?? % ステレオからモノラルへの変換
4 N = ??; % yの長さ
5 t = (1:??) /??; % 時間軸テーブル
6 z = ??; % yの各要素に対して正ならば負, 負ならば正にする
7 ...
```

ステレオからモノラルへの変換をすること！

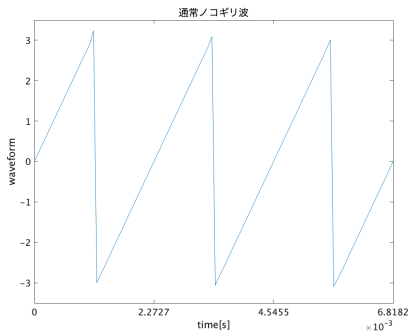
- ▶ 左右それぞれの信号が処理されグラフが見難くなったり
- ▶ そもそもデータ数が想定していたものと異なるのでplotされなかったり

## 6. 周波数解析（ノコギリ波）

ノコギリ波は以下の式で表せる．

$$y(t) = \sum_{k=1}^{\infty} (-1)^{k-1} \frac{2}{k} \sin(kt) \quad (5)$$

$$= 2 \sin(t) - \sin(2t) + \frac{2}{3} \sin(3t) - \frac{1}{2} \sin(4t) + \dots \quad (6)$$



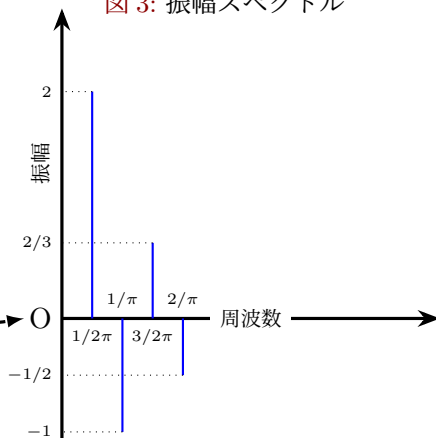
## 6. 周波数解析 (振幅スペクトル)

正弦波	振幅	周波数
$2 \sin(t)$	2	$1/2\pi$
$-\sin(2t)$	-1	$1/\pi$
$\frac{2}{3} \sin(3t)$	$2/3$	$3/2\pi$
$-\frac{1}{2} \sin(4t)$	$-1/2$	$2/\pi$

$\vdots$

$$(-1)^{k-1} \frac{2}{k} \sin(kt) \quad (k \in \mathbb{N})$$

図 3: 振幅スペクトル



## 6. 周波数解析 (fft)

---

```
fft_y = fft(y); % データ列 y に対して fft を行う.
```

---

fft は、データ列  $y$  を高速フーリエ変換する関数であるが、入力データ列  $y$  に対して出力データ列  $\text{fft}(y)$  のデータ長は、入力データ列  $y$  と一致する。

ただし、index とその内容についてはそれぞれ異なる。

- ▶ 入力データ列  $y$  の index は時刻に対応.\*
- ▶ 出力データ列  $\text{fft}(y)$  の index は周波数に対応.†

---

\*†index が時刻や周波数というわけではない。

## 6. 周波数解析 (fftshift)

- ▶ サンプル周波数  $F_s$  の信号  $y$  に対して高速フーリエ変換 `fft` を行うだけでは周波数解析ができない！
- ▶ `fft` では  $-F_s/2 \leq \text{周波数成分} \leq F_s/2$  を得られる.
- ▶ ただし、正のデータ・負のデータが（左右）が入れ替わった状態で出力される.

図 4: `fft` を行った直後 (`fft_y`)

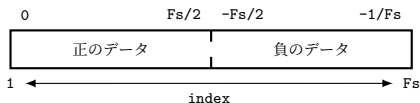


図 5: 理想的な状態





## 6. 周波数解析 (fftshift)

- ▶ 図 4,6 から図 5,7 の状態にするために `fftshift` 関数を用いる.

```
fftshift_y = fftshift(fft_y);
```

図 6: `fft`

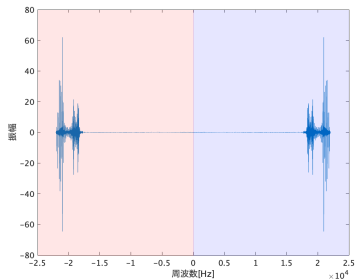
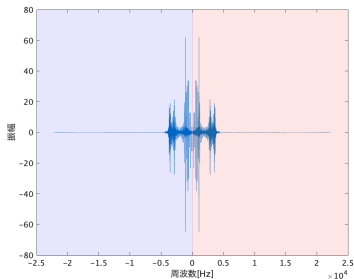


図 7: `fftshift`



## 6. 周波数解析 (abs)

- ▶ 絶対値を取るために abs をとる.

```
fftshift_abs_y = abs(fftshift_y);
```

図 8: fftshift

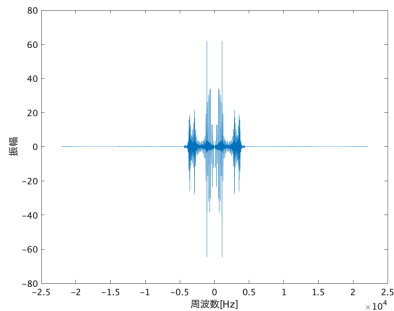
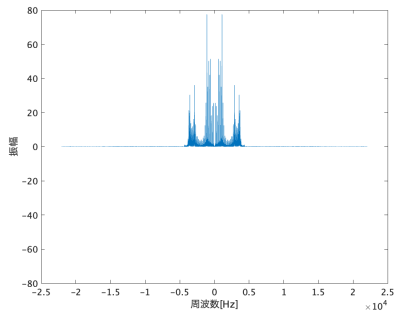


図 9: abs

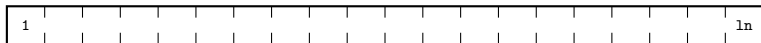


## 6. 周波数解析（周波数テーブルの作成）

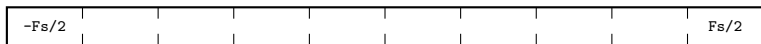
これまでの工程で、0を中心として、正負にナイキスト周波数 ( $F_s/2$ ) 分離れた振幅データを得られる。

1. データの長さ  $ln = \text{length}(\text{abs}(\text{fftshift}(\text{fft}(y))))$  を取得する。  
( $\text{length}(\text{fft}(y)) = \text{length}(y)$ ,  $ln = \text{length}(y)$  となる) → p.15
- ? 10 個のリンゴを 5 つの箱へ均等に入れます。1 箱あたり 10/5 個のリンゴが入っていますね。(間隔 0.5)
2. データ長  $ln$  のデータを, サンプル周波数  $F_s$  長の行列に対応させるためには?

`abs(fftshift(fft(y)))` で得られたデータ



作りたい行列 = freq



← 長さ  $F_s$  →

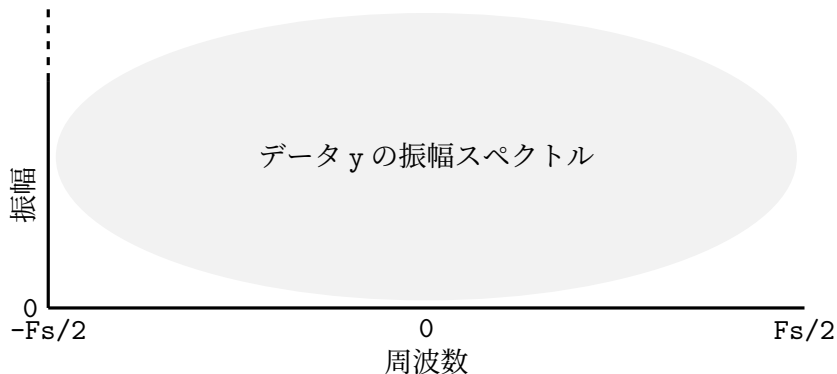
```
freq = [-Fs/2 : Fs/ln : Fs/2 - Fs/ln];
```

## 6. 周波数解析（周波数テーブルの作成）

---

```
plot(周波数テーブル, 振幅データ);
```

---



## 7. 課題3

1. 課題2で録音した各音声を周波数解析し、スペクトル上でピークのある周波数およびその周波数のY軸の値を抽出しよう。
2. 抽出した周波数、Y軸の値の純音を合成し、音の違いを聴き比べる。

- .....
- ▶ 0Hz - 10,000Hz 程度までの範囲で、ピークとなる周波数、及びそのピーク値を高い順から10個程度以上抽出する。
    - ▶ とりあえず、手動で10個抽出する。
  - ▶ 周波数およびピーク値（Y軸の値）は厳密でなくて良い。
  - ▶ a, i の合成音の聞こえ方の違い、及び元の音声との違いについて考察する。

## 7. 課題3 (サンプルコード：解析)

```
1  clear;
2  % --- "a" の音声 ---
3  [y_a, Fs_a] = audioread('sound_a.wav');
4  y_a = y_a(?); % ステレオからモノラルへの変換
5  t_a = [0 : length(?)-?]/ ?; % 時間軸テーブル
6  Y_a = ?; % 高速フーリエ変換
7  YS_a = ?; % fftshift
8  YS_a_abs = ?; % 絶対値を取る
9  YS_a_len = length(YS_a_abs); % YS_a_abs のデータ長を取得
10
11 % 周波数テーブル作成
12 freq_a = [-Fs_a/2 : ? : Fs_a/2 - ?];
13
14 figure;
15 plot(? , ?); % 一度グラフにplotしてみる
16 axis([0 5000 0 80]); % 0Hz - 5000Hz まで表示する
```

## 7. 課題3 (サンプルコード：解析)

図 10: 振幅スペクトル

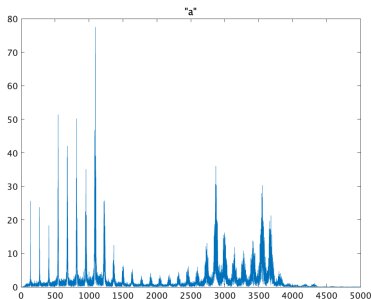
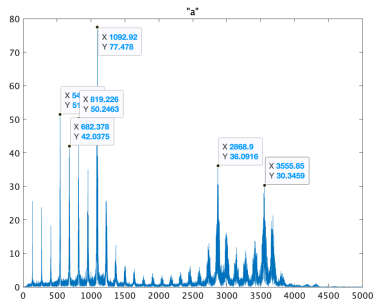


図 11: 振幅が大きいところ 10 点選ぶ



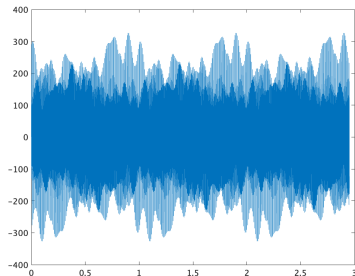
► plot した点を行列に記す.

```
17 a_x = [1093 2867 ... 682]
18 a_y = [77 36 ... ... 42]
```

## 7. 課題3 (サンプルコード：合成)

```
19 ya = 0;  
20 for k=1:10 % t_a は 時間 軸 テーブル  
21     ya = ?? + ?? * sin(2*pi* ?? *t_a);  
22 end
```

図 12:  $y_a$  を  $t_a$  に対してプロットした結果



- ▶  $a_x$  は周波数,  $a_y$  は振幅であることを思い出す.
- ▶ それらの重ね合わせで音が再現されていたことを思い出す.  
(フーリエ級数展開)
- ▶ 周波数  $f$  の正弦波は

$$y(t) = \sin(2\pi ft)$$

であることを思い出す.