音の工学的特徴とそれに対する聴覚の特性についての考察

溝口 洸熙*

May 7th, 2023

*高知工科大学 情報学群 学籍番号:1250373 Group10

概要

音声技術は日常生活で様々な場面で利用されている．例えば駅の放送案内や AI アシスタントの合成音声，イヤフォンのノイズキャンセリング機能，音声をテキストに変換するソフトウェアなど挙げればきりがない．これらの技術はどのようにして実現できているのだろうか．そのために音の工学的特徴を捉える．

また，音は様々な周期や振幅を持つ正弦波の重ね合わせで再現されていることを踏まえて，周波数成分を取り出して解析する周波数解析を行い，周波数解析の結果を用いて周波数を編集し波形を再構成する．

最後に人間の聴覚特性を再現する．人間の聴覚特性である音脈分凝，連続聴効果またミッシングファンダメンタル波刺激を起こす音波を計算機上で作成する．

目次

図目次

表目次

ソースコード

第 1 章

音の工学的特徴

1.1　周波数の異なる純音の生成

1.1.1　実験の目的

我々は音の「高い」「低い」をどのようにして認識しているのだろうか．音が高いまたは低いと感じるためには何かと比較するはずだがその比較の指標は何だろうか．この実験ではまず，正弦波の生成をプログラミングを用いて作成する．そして周波数の変化に対して正弦波グラフおよび音の違いを実験を通して確認し，考察する．

1.1.2　実験の方法と考え方

■実験に用いる装置　このレポート内全ての実験に用いる言語は MathWorks®社の MATLAB®を用い，表 1-1 の環境を用いて実験する..

表 1-1　実験環境

実験機	Mac Studio 2022（Apple 社）
プロセッサ	Apple Silicon M1 Max
メモリ	32GB
MATLAB®	R2023a Update1 9.14.02239454 64-bit (maci64) March 30, 2023

また，このレポートない全ての実験では MATLAB®でプロットしたグラフを出力するために，関数 src.1-1 を用いている．

src. 1-1　グラフ出力

```
exportgraphics(figurename,'path/figure_name.pdf','ContentType','vecto')
```

■正弦波について　時刻 t に対して周波数 f の正弦波は，式 (1.1) で得られる．

$$y = \sin(2\pi f t) \tag{1.1}$$

時間軸データ t を 1 行 N 列のベクトルに代入する．時間軸データの作成について，サンプリング周波数 `Fs` に対して m 秒間の正弦波を生成するためには，src.1-2 のように `0` から `Fs` まのベクトルに対して，各要素をサンプリング周波数で割ると時間軸テーブルを作成することができる．

t の各要素 t_n に対して三角関数 $\sin(2\pi f t_n)$ を演算し，ベクトル y の要素 y_n に代入する．従って y も 1 行 N 列のベクトルになる．生成した正弦波を `plot` 関数を用いて y を t の関数として描画する．このように，ただ一つの正弦波からなるような音を**純音**という．[1, p.1] また，サンプリング周波数を `Fs` とし，データ列 `y` を再生するためには `sound` 関数を用いる．

src. 1-2　時間軸作成と正弦波の作成

```
t = (0 : m*(Fs-1)) /Fs;
y = sin(2*pi*f*t);
```

■実験内容　この実験では，周波数を $f_1 = 440\text{Hz}$，$f_2 = 660\text{Hz}$ の 2 種類を用いてそれぞれ正弦波 y_1，y_2 を生成する．生成した $y_n (n = \{1, 2\})$ に対して，t を横軸に取りグラフを作成し，サンプリング周波数を `Fs` $= 16000\text{Hz}$ として再生する．1.1 節ソースコードは A-3 ⇒p.12．

1.1.3　実験の結果

各正弦波のグラフを図 1-1 に示す．音を聴き比べた結果，f_2 の周波数を用いた正弦波は f_1 を用いた正弦波に比べて音が高かった．具体的には f_1 が A の音*であるのに対して，f_2 の音は完全 5 度大きい E の音†であった．さらに，聴音確認・目視確認では音の大きさ，音色，振幅や波形の変化は確認でなかった．

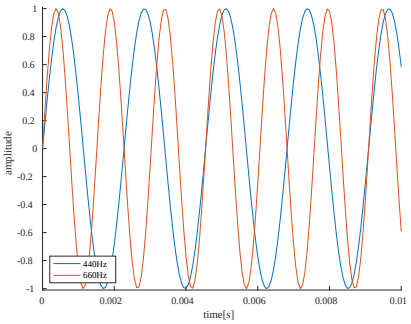

図 1-1　周波数が異なる正弦波のグラフ

1.1.4　考察

周波数 (f) とは 1 秒間の振動回数であり，これが音の高さを決める．実験結果より周波数が大きい，つまり 1 秒間により多く振動すれば音が高くなることが分かった．また，1 回振動あたりの時間を周期 (T) と言うが，周期と周波数は反比例の関係であり，式 (1.2) が成り立つ．

$$f = \frac{1}{T} \qquad \left(\text{無論}\quad T \neq 0,\quad f \neq 0\right) \tag{1.2}$$

図 1-1 より，周波数が大きい正弦波は周期が短く，逆もまた確認できる．周波数のみを変更したので，振幅や正弦波の波形変化はない．

また，周波数が f_1 正弦波の音と f_2 正弦波の音はそれぞれ純音だがこれらの間にも関係がある．数学的には $f_1 : f_2 = 2 : 3$ の簡単な整数比になっている．

> ヒトの耳は 2 つの音を同時に聞いた場合，その 2 つの音の周波数が簡単になっているほど協和して聴こえる．（略）
> このように周波数比が整数比になっている音程を純音程とよぶ．　[1, p.46-p.47]

つまり，周波数 f_1 正弦波と周波数 f_2 正弦波の加算合成波はは協和してきこえる．

1.2　振幅・位相の確認

1.2.1　実験の目的

音の大小は何によって決まるのだろうか．音の大小に関わる波の振幅や，波を構成する上で重要な初期位相を変化させ，変化の前後での音の違いを聞き取り，人間の耳に初期位相の変化や振幅の変化がどのように感じるか実験を通して考察する．

1.2.2　実験の方法と考え方

時刻 t に対して周波数 f の正弦波は式 (1.1) で得られるが，その初期位相（*initial phase*）を ϕ とすると，その正弦波は式 (1.3) となる．

$$y = \sin(2\pi f t + \phi) \tag{1.3}$$

また，同様な正弦波：式 (1.1) の振幅を A 倍して得られる正弦波は式 (1.4) となる．

$$y = A\sin(2\pi f t) \tag{1.4}$$

*イタリア語音階で「ラ」
†イタリア語音階で「ミ」

■実験内容　この実験では，初期位相の変化と振幅の変化，それぞれ実験し変化前と変化後の音や波形の違いを発見する．周波数は $f = 440$Hz とし，サンプリング周波数を Fs $= 16000$Hz とする．左から純音，右から波長や初期位相を変化させた音を再生するようにして (src.1-3)，音の変化を確認する．1.2 節ソースコードは A-4 ⇒p.12．

src. 1-3　左右から別の音を出力

```
Fs = 16000; % サンプリング周波数
y1 = 音声データ1; % N行1列（左）
y2 = 音声データ2; % N行1列（右）
y = [y1 y2]; % N行2列　列結合を行う
sound(y, Fs);
```

表 1-2　振幅・位相の確認 実験内容

実験対象	振幅（基準倍）	初期位相	生成される正弦波
純音	基準	基準	$y_0 = \sin(2\pi ft)$
振幅	0.5	0	$y_1 = 0.5 \times \sin(2\pi ft)$
	0.25	0	$y_2 = 0.25 \times \sin(2\pi ft)$
初期位相	1	$+\frac{\pi}{2}$	$y_3 = \sin(2\pi ft + \frac{\pi}{2})$
	1	$+\pi$	$y_4 = \sin(2\pi ft + \pi)$

1.2.3　実験の結果

聴音確認の結果を表 1-3，図 1-2 及び図 1-3 に示す．振幅の変化について，振幅の変化に比例した音量変化を確認できたが，初期位相の変化による音の変化は確認できなかった．

表 1-3　振幅・位相の確認 実験結果

実験対象	振幅（基準倍）	初期位相	生成される正弦波	純音との比較
純音	基準	基準	$y_0 = \sin(2\pi ft)$	—
振幅	0.5	0	$y_1 = 0.5 \times \sin(2\pi ft)$	音量がおおよそ 1/2 に聞こえた．
	0.25	0	$y_2 = 0.25 \times \sin(2\pi ft)$	音量がおおよそ 1/4 に聞こえた．
初期位相	1	$+\frac{\pi}{2}$	$y_3 = \sin(2\pi ft + \frac{\pi}{2})$	違いは分からなかった．
	1	$+\pi$	$y_4 = \sin(2\pi ft + \pi)$	違いは分からなかった．

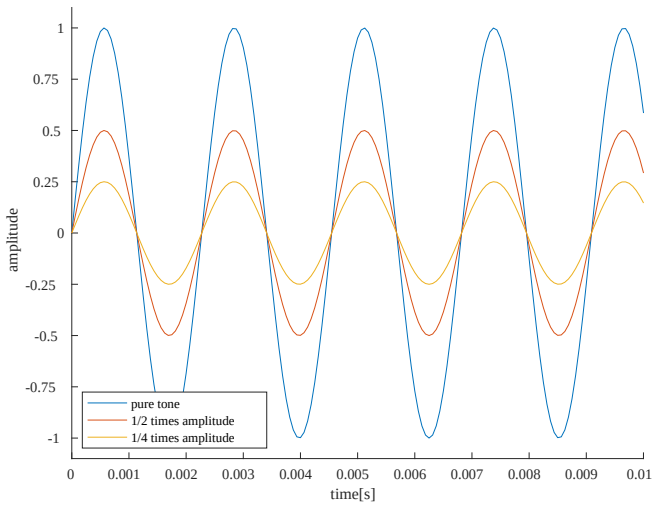

図 1-2　振幅・位相の確認 実験結果（振幅の変化）

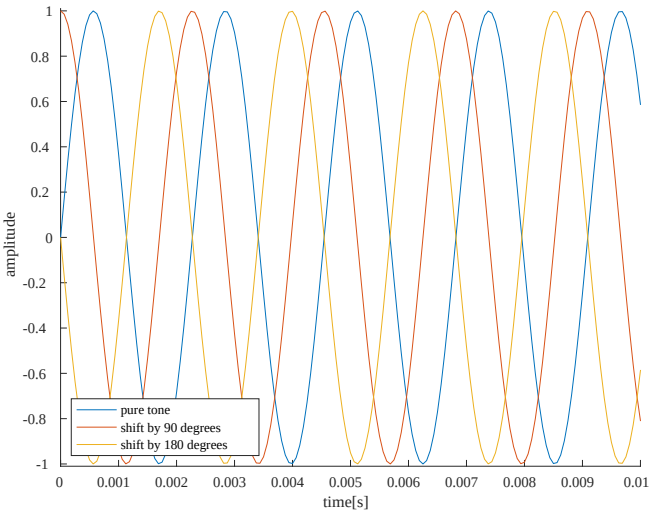

図 1-3　振幅・位相の確認 実験結果（初期位相の変化）

1.2.4　考察

■振幅の変化について　振幅の変化は，音量と比例することを確認できた．図 1-2 より，各正弦波において周期や初期位相が等しいので，縦軸が 0 になる時刻や最大値を迎えるは等しいく，振幅のみが異なる．

■初期位相の変化について　位相の変化による音の変化は確認できなかった．図 1-3 より，振幅や周期は等しいが，初期位相が異なるため最小値や最大値を迎える時刻が初期位相分異なる．

1.3　うなり

1.3.1　実験の目的

ティンパニやギターのチューニングを行うとき，音叉やチューナーから音を出して，異なる互いに周波数であれば気づきチューニングする．それぞれ異なる周波数が異なる周波数が周波数の違いによるうなりの発生やその原因を数学的観点から考察する．

1.3.2　実験の方法と考え方

うなりとは，異なる周波数が干渉するとき起こる現象である．異なる周波数同士の正弦波が干渉することで，強め合いの場所と弱め合いの場所が周期的に現れることで発生する．例えば周波数 f_1 の正弦波に対して周波数 $f_2(\neq f_1)$ の正弦波を干渉させると，式 (1.5) より 1 秒間に N 回のうなりを聞くことができる．

$$N = \left|f_1 - f_2\right| \tag{1.5}$$

f_1 と f_2 の差が大きければ，当然うなり（N）の回数は多くなり，f_1 と f_2 の差が小さければうなりの回数は少なくなる．

本実験では，サンプリング周波数を `Fs` $= 16000$Hz，提示時間 4 秒，周波数が $f_1 = 440$Hz の純音 y_1，$f_2 = 441$Hz の純音 y_2 を加算合成 $\left(y_1 + y_2\right)$ した波形を生成し再生する．1.3 節ソースコードは A-5 ．⇒p.14

1.3.3　実験の結果

うなりは 4 回聞こえた．また，時間軸に対して加算合成したデータを描画すると図 1-4 となった．

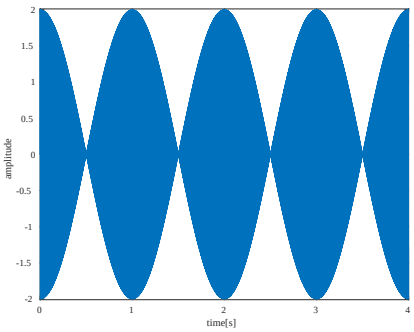

図 1-4　うなり 実験結果

1.3.4　考察

式 (1.5) に従って，$\left|f_1 - f_2\right| = \left|440 - 441\right| = 1$ より 1 秒あたり 1 回のうなりが聞こえるはずなので，この実験結果は論理的に正しい．加算合成波図 1-4 の振幅に着目しても，周波数 f_1，f_2 それぞれの正弦波の振幅が 1 であることを考えると，その加算合成波の振幅が 2 であることも理解できる．さらに，この加算合成波は $t = 0$ のとき最大値を迎えており，これは初期位相が同一の正弦波の加算合成であることを表している．

1.4　フーリエ級数展開

1.4.1　実験の目的

矩形波は周期関数だが，見た目は正弦波ではない．これをプログラム上でどのように再現するのか．フーリエ級数展開を用いた矩形波の描画やフーリエ級数展開のプログラム上で再現し，理想的な矩形波との比較を行う．

1.4.2　実験の方法と考え方

■フーリエ級数展開　周期 T の任意の周期信号 $f(t)$ に対して，それはより短い周期 $T/n(n = 1, 2, \dots)$ を持つ正弦波の重ね合わせで表すことができる．具体的には式 (1.6) の $N = \infty$ で表すことができ，これをフーリエ級数展開という．[2, p.18-p.19] 特に式 (1.6) の $N = 1$ の時を基本周波数と呼ぶ．また，$k > 1$ のときの周波数を持つ正弦波を高調波と呼ぶ．

$$f(t) = \frac{1}{2}a_0 + \sum_{k=1}^{N}\left(a_k \cos(k\omega_0 t) + b_k \sin(k\omega_0 t)\right) \qquad \omega_0 = \frac{2\pi}{T} \tag{1.6}$$

■矩形波　矩形波は周期関数である．その周期を T とすると，関数 f は時刻 t に対して，フーリエ級数展開するとになる．式 (1.7)$(N = \infty)$ は周期 $T = 1/2$ で定義した矩形波で，式 (1.7) を $N = 50$ で出力した．

$$f(t) = \sum_{k=1}^{N} \frac{1}{2k-1} \sin\left(2\pi(2k-1)ft\right) \tag{1.7}$$

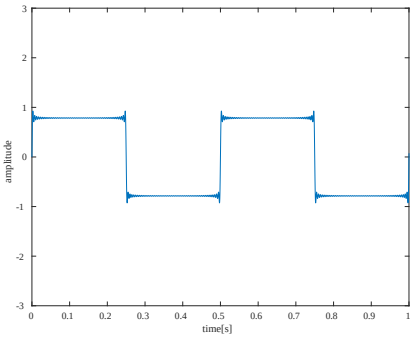

図 1-5　矩形波（$1 \leq k \leq 50$）

■実験内容　周期 $T = 1/2$ として，式 (1.7) の N の値を $N = 1, 5, 25$ と増やした時に次第に波形が矩形波に収束していくことを確認する．計算機で和 $\left(\sum\right)$ の演算方法を src.1-4 に示す．1.4 節ソースコードは A-6 ⇒p.14．

src. 1-4　和の演算

```
y = 0;
for k=1:50
    y = y + 1/(2*k-1) * sin(2*pi*(2*k-1)*f*t); % 矩形波の例
end
```

1.4.3　実験の結果

図 1-6 の通り，N を大きくすると，矩形波に収束することがわかった．また，基本周波数と矩形波の周期も目測では一致している．

1.4.4　考察

計算機を用いて N の値を無限大で出力することは不可能である．図 1-5 のように $N = 50$ として出力すると，理想的な矩形波に近づくが完全ではない．

ここで N の値を大きくすると，$T/2, T$ あたりで尖ったもの（ひげ）が確認される．フーリエ級数では不連続で誤差が大きい．このことをギブス現象という．[2, p.34]

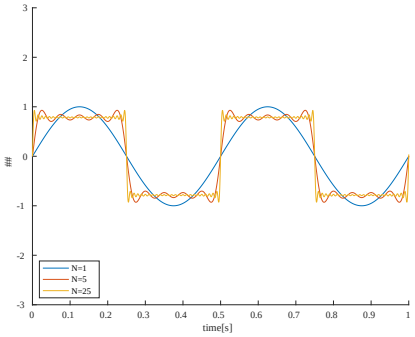

図 1-6　フーリエ級数展開 実験結果

1.5　白色ガウス雑音

1.5.1　実験の目的

白色雑音（*white noise*）は以下のように定義されている．

> すべての周波数成分を均等に含む，パワースペクトルが一定である不規則性の非常に強い波のことである．　[3]

ガウス雑音（*gaussian noise*）は以下のように定義されている．

> Gaussian noise represents statistical noise having the probability density function equal to that of the normal distribution. [4]

つまり，「正規分布の確率関数と等しい確率密度関数を持つ統計的ノイズ」を表す．今回の実験ではガウス雑音を計算機上で構成し，その信号振幅のヒストグラムを作成する．そのヒストグラムがガウス分布*に従っているかを確認する．

1.5.2　実験の方法と考え方

■MATLAB®関数の説明　標準正規分布から乱数の行列を取り出すには，randn 関数を用いる．乱数生成機の初期化を行い，n 行 m 列の乱数ををを取り出すには，src.1-5［1-2 行目］を記述する．また，データ列 X に対して，ヒストグラムを num 段階で分割し表示するためには，src.1-5［3-5 行目］を記述する．

src. 1-5　乱数生成・ヒストグラム

```
1 rng(0, 'twister');
2 X = randn(n,m);
3 num = 100;
4 [h, c] = hist(X, num);
5 plot(c,h);
```

■ガウス分布と標準正規分布　ガウス分布とは，平均値・中央値・最頻値が一致するという特徴を持つ確率分布．その確率変数を x としたときの確率密度関数 f は式 (1.8) のように与えらえる．標準正規分布は，平均 $\mu = 0$，標準偏差 $\sigma^2 = 1$ のガウス分布である．その標準正規分布の確率密度関数 f_N は式 (1.9) となる．

src. 1-6　平均と標準偏差の変更

```
X = randn(n,m); % 乱数生成
X = a.*X + b; % 演算
```

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu^2)}{2\sigma^2}\right) \qquad \left(\mu : \text{平均} \quad \sigma^2 : \text{分散} \quad -\infty < x < \infty\right) \tag{1.8}$$

$$f_N(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \qquad \left(-\infty < x < \infty\right) \tag{1.9}$$

> 確率変数 x に対して，その平均が μ_x，分散が ${\sigma_x}^2$ のとき，$y = ax + b$　(aとbは定数) で定義される確率変数 y の平均は $\mu_y = a\mu_x + b$，分散は ${\sigma_y}^2 = a^2{\sigma_x}^2$ となる． [5]

従って，標準正規分布に従って出力されたデータ列の平均を a，標準偏差を b に変更したい場合には，n 行 m 列データ列 X に対して X の各要素と a の積をとり，データ列 X と b の和をとる．（src.1-6）1.5 節ソースコードは A-7 ⇒p.15.

1.5.3　実験の結果

ガウス雑音の波形を図 1-7，ヒストグラムを図 1-8 に示す．聴音した結果，「ザー」という所謂雑音が聞こえた．

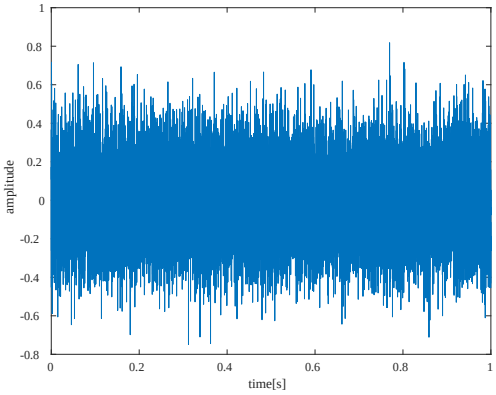

図 1-7　ガウス雑音の波形

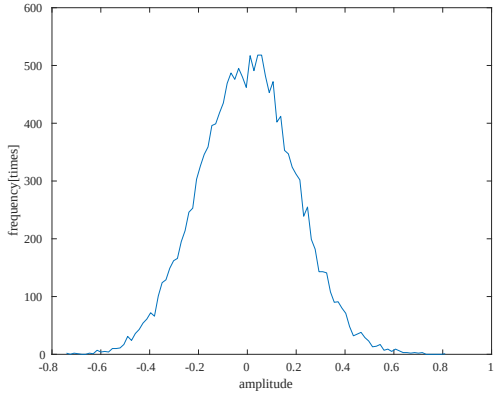

図 1-8　ヒストグラム

1.5.4　考察

聴音確認による，ガウス雑音の特徴は捉えることができなかった．ヒストグラム図 1-8 を確認すると，平均 $\mu = 0$ のガウス分布におおよそ従っていることがわかる．

*以後，課題内容や他の用語との兼ね合いにより「正規分布」を「ガウス分布」と表す．

第 2 章

周波数解析

2.1　フーリエ変換と周波数解析

2.1.1　実験の目的

MATLAB®を用いて周波数解析を行う．我々が普段見る「波形」は時間軸を横，振幅を縦にしたグラフである．このグラフからわかるのは時刻に対しての振幅だ．ただし波形の分析はそれだけでは不十分である．波形というのはフーリエ級数展開の理論により，いかなる波形も式 (1.6) の $N = \infty$ で表せる．波形に対してフーリエ変換を行うと「周波数に対する振幅」を得ることができる．矩形波式 (1.7) の振幅と周波数をグラフに描画したものを振幅スペクトルと呼ぶ．（図 2-1）

今回の実験では，純音と矩形波に対して周波数解析を行う．純音では解析結果より純音の周波数をより多く含んでいるか確認する．また矩形波では高調波の存在を確認し，振幅が式 (1.7) 通りになっているか確かめる．

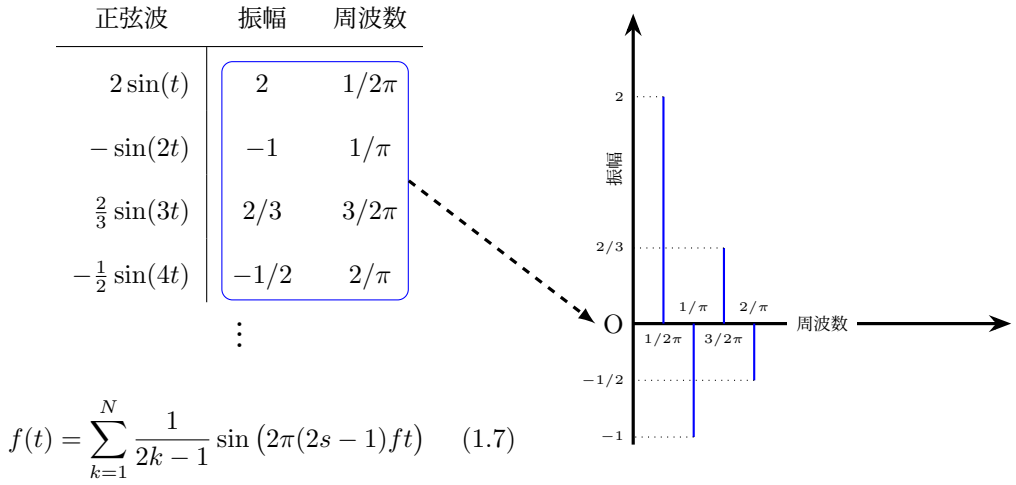

図 2-1　周波数解析

2.1.2　実験の方法と考え方

■高速フーリエ変換の実装　MATLAB®では高速フーリエ変換をする関数（`fft`）がある．ただし，`fft` 関数の仕様上いくつかの注意が必要である．データ列 `y` に対して振幅スペクトルを取得する手順は以下の通りである．（src.2-1）

1. 高速フーリエ変換を行う．
   `fft` 関数は出力として，サンプリング周波数 `Fs` に対して，`[-Fs/2, Fs/2]` 範囲の周波数に対する振幅のデータを得る．
2. 出力データ列 `fft(y)` のデータを整列させる．
   `fft` 関数の出力は，正のデータ・負のデータ（左右）が入れ替わった状態で出力される．（図 2-2，図 2-3）
3. これまでの過程で出力されるデータは複素数データである．絶対値を取るために `abs` 関数を用いる．（図 2-4）
4. グラフとして描画するために，周波数軸を作成する．（src.2-1）

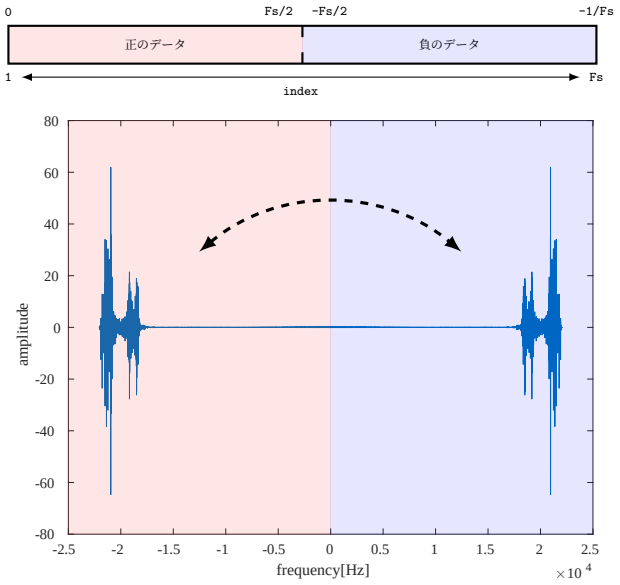

図 2-2　`fft` 直後の出力データ

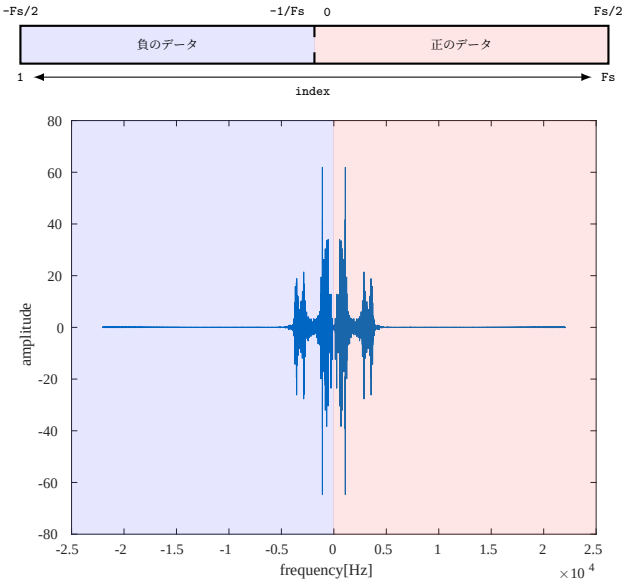

図 2-3　`fftshift` 後の出力データ

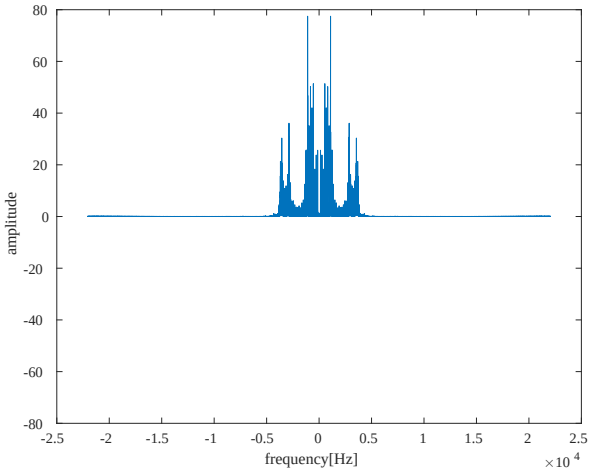

図 2-4　`abs` を適用した後のデータ（グラフ）

src. 2-1　振幅スペクトルの取得

```
1 y; % データ列
2 y_fft = fft(y); % 高速フーリエ変換
3 y_fftshift = fftshift(y_fft); % 左右交換
4 y_abs = fft_y; % 絶対値
5 ln = length(y_abs);
6 % 周波数テーブルの作成
7 freq = [-Fs/2 : Fs/ln : Fs/2 - Fs/ln];
8 plot(freq, y_abs); % グラフ描画
```

周波数テーブルの作成について，得られたデータ数 `ln` に対して，周波数テーブルの長さ `Fs` に収める必要があるので，ステップ幅 `Fs/ln` となる`-Fs/2` から `Fs/2` の配列を作成する．

■実験の内容　今回の実験では，純音と矩形波の周波数解析を行う．サンプリング周波数を `Fs` = 8192Hz と設定する．

純音　440Hz の純音で実験する．

- 純音から 1024 点取り出す．
  データ列の先頭から 1024 個のデータを選択する方法で行う．
- 振幅スペクトルを 0Hz から 1kHz まで描画する．
- 440Hz にピークが来ているか確認する．

矩形波　128 点値 0 が続き，128 点値 1 が続く波形を 4 回繰り返す矩形波．（図 2-5）

- 振幅スペクトルを 0Hz から 200Hz まで描画する．
- 高調波が発生しているか，その振幅が式 (1.7) 通りになっているか確認する．

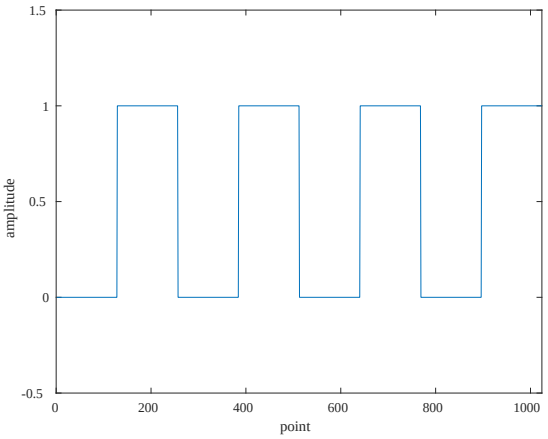

図 2-5　作成した矩形波

2.1 節ソースコードは B-1 ⇒p.16 . src.B-2. ⇒p.16

2.1.3　実験の結果

出力された結果に対して，特出する点の座標を表示させる．図 2-6 について，440Hz の軸に対して振幅が大きくあることがわかった．

図 2-7 については図中にある点の高調波が確認された．

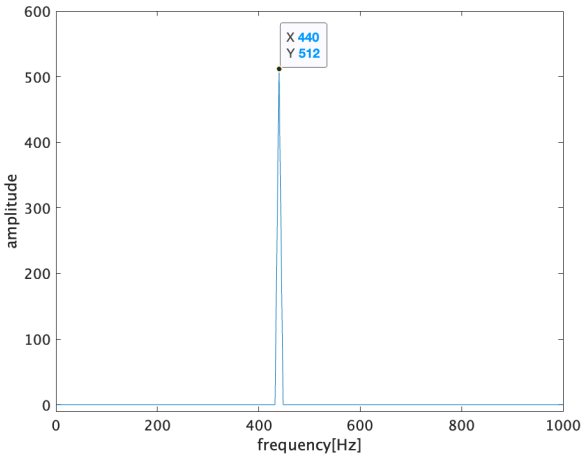

図 2-6　純音の振幅スペクトル

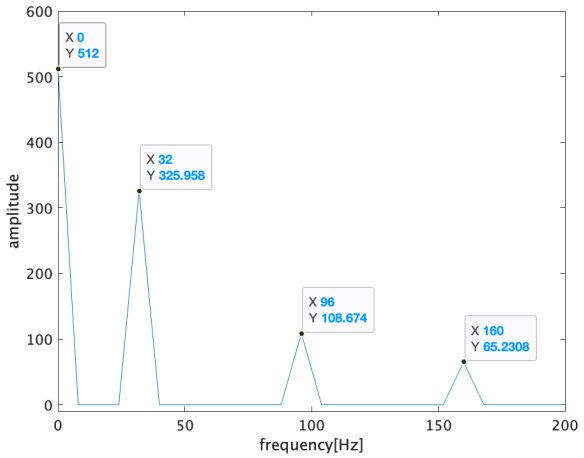

図 2-7　矩形波の振幅スペクトル

2.1.4　考察

■純音　実験結果より，純音の周波数 440Hz が一番多く，他の高調波はほとんど目立たない．

■矩形波　図 2-7 より高調波の存在は確認できる．矩形波のフーリエ級数展開は式 (1.7) である．$k = \{1, 2, 3, 4\}$ のときの正弦波の式を式 (2.1) に示す．

$$\begin{aligned} \sin(2\pi(2-1)ft) &= \sin(2\pi ft) && (k=1) \\ \frac{1}{3}\sin(2\pi(4-1)ft) &= \frac{1}{3}\sin(6\pi ft) && (k=2) \\ \frac{1}{5}\sin(2\pi(6-1)ft) &= \frac{1}{5}\sin(10\pi ft) && (k=3) \\ \frac{1}{7}\sin(2\pi(8-1)ft) &= \frac{1}{7}\sin(14\pi ft) && (k=4) \end{aligned} \tag{2.1}$$

これからわかるように，周波数が大きくなるにつれて振幅は小さくなることがわかる．これは図 2-7 と一致する．ただし，式 (2.1) のように $\frac{1}{3}, \frac{1}{5}, \ldots$ と変化しないのは `abs` 関数の利用の際に振幅が変化するためと考えられるが（図 2-4），詳しくは不明である．また，図 2-7 と式 (2.1) の周波数に関する関係も不明である．

2.2　LPF（ローパスフィルタ）

2.2.1　実験の目的

ローパスフィルタ（以下 LPF）は以下のように説明されている．

> ある周波数（カットオフ周波数）より高い周波数を通さないフィルタは LPF（ローパスフィルタ）と呼ばれる．
>
> [6, p.65]

今回の実験では LPF を作成して，別途作成した音に対してフィルターを適用する．フィルターを適用する前後での振幅スペクトルを比較し，フィルターが機能しているかを確認するとともに，フィルターが適用されているか聴音確認する．

2.2.2　実験の方法と考え方

今回フィルターを適用させる音源は，ドイツ語音階で「C4」から「C5」まで 0.25 秒づつ音を提示する．提示する音の順序を式 (2.2) に，各音階の周波数を表 2-1 に示す．

$$\mathrm{C4} \rightarrow \mathrm{D4} \rightarrow \mathrm{E4} \rightarrow \mathrm{F4} \rightarrow \mathrm{G4} \rightarrow \mathrm{A4} \rightarrow \mathrm{H4} \rightarrow \mathrm{C5} \tag{2.2}$$

音源に対して，G4 より低い音のみを通過させる LPF を作成しフィルターを適用する．カットオフ周波数を G4 の周波数に設定すると，G4 の周波数より小さいわずかな成分が除去できないため，F4 と G4 の中間に位置する周波数（375Hz）に設定する．

表 2-1　音階と周波数

音階	C4	D4	E4	F4	G4	A4	H4	C5
周波数 Hz	261.38	293.67	329.63	349.23	392.00	440.00	493.88	523.23

■LPF の作成と適用　今回はフィルターの行列を作る方法ではなく，周波数テーブルの閉区間 $\left[-375\mathrm{Hz}, 375\mathrm{Hz}\right]$ 以外の振幅を 0 にする方法でフィルターを適用する．（src.2-2）2.2 節ソースコードは B-3 ⇒p.17．

src. 2-2　フィルターを適用する

```
% データ列yをフーリエ変換後，Shiftしてabsをとったものを "fft_y" に格納している．
% 周波数テーブルの変数名は "freq" である．
for k=1:length(fft_y)
    if freq(k) >= 375 || freq(k) <= -375
        fft_y(k) = 0;
    end
end
```

2.2.3　実験の結果

LPF 適用前の音声波形と，LPF 適用後の音声波形をに示す．また，図 2-10 には上にフィルタ適用前の振幅スペクトル，下にフィルタ適用後の振幅スペクトルを示している．聴音確認の結果 G4 以降の音は聞こえなかった．

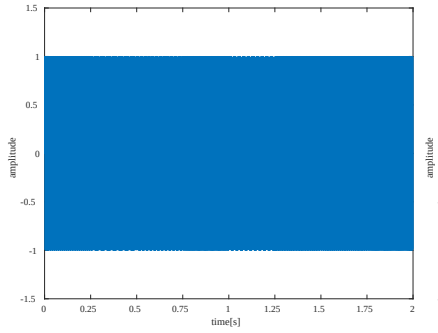

図 2-8　LPF 適用前の波形

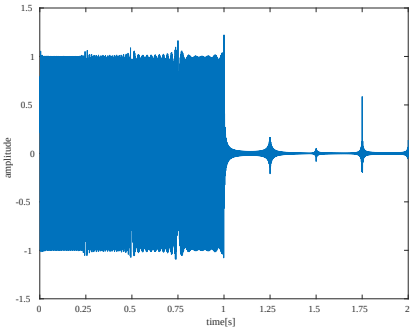

図 2-9　LPF 適用後の波形

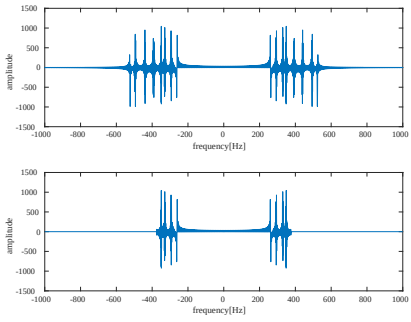

図 2-10　振幅スペクトルの確認

2.2.4　考察

結果より 1 秒後の音以降が聞こえない．1 音階 0.25 秒あるので，4 音階分（C4，D4，E4，F4）までが聞こえることを考えると，実験結果は理論的に正しい．図 2-10 より，特定周波数以降が切り取られていることも確認できる．

ただし，1 秒後以降の波形も完全に 0 ではない理由は不明である．

参考文献

[1] 飯田一博. 音響工学基礎論. コロナ社, 2012.
[2] 和田成夫. よくわかる信号処理 —フーリエ解析からウェーブレット変換まで—. 森北出版, 2009.
[3] 日本機械学会機械工学事典. 白色雑音. https://www.jsme.or.jp/jsme-medwiki/doku.php?id=13:1010098, 2017.
[4] Tudor Barbu. Variational image denoising approach with diffusion porous media flow. In *Abstract and Applied Analysis*, Vol. 2013. Hindawi, 2013.
[5] MathWorks. 指定の平均値と分散を持つ正規分布からの乱数. https://jp.mathworks.com/help/matlab/math/random-numbers-with-specific-mean-and-variance.html, 2023.
[6] 花泉弘. 小泉悠. 伊藤克. MATLAB で学ぶ実践画像・音声処理入門. コロナ社, 2019.

付録

A　課題 1

src. A-3　01-01: 周波数の異なる純音の生成（第 1.1 節 → p.1）

```
clear;

Fs = 16000; %
f1 = 440;   % [Hz]
f2 = 660;   % [Hz]

t1 = (0:(Fs-1)) /Fs;
y1 = sin(2*pi*f1*t1);

t2 = (0:(Fs-1)) /Fs;
y2 = sin(2*pi*f2*t2);

sound(y1,Fs);
pause(3)
sound(y2,Fs);

fig0 = figure;
hold on;
plot(t1,y1);
plot(t2,y2);
hold off;
xlabel('time[s]');
ylabel('amplitude');
legend({'440Hz','660Hz'},'Location','southwest');
axis([0 0.01 -1.01 1.01]);
exportgraphics(fig0,'../Figures/01_01.pdf','ContentType','vector');
```

src. A-4　01-02: 振幅・位相の確認（第 1.2 節 → p.2）

```
clear;

Fs = 16000; %
f1 = 440;   % [Hz]

t1 = (0:(Fs-1)) /Fs;
for k=0:Fs
    y1 = sin(2*pi*f1*t1);
end

p = pi/2;
for k=0:Fs
    y2 = sin(2*pi*f1*t1 + p);
```

```
end

p = pi;
for k=0:Fs
    y3 = sin(2*pi*f1*t1 + p);
end

for k=0:Fs
    y4 = 0.5 * sin(2*pi*f1*t1);
end

for k=0:Fs
    y5 = 0.25 * sin(2*pi*f1*t1);
end

ys2 = [y1;y2]';
ys3 = [y1;y3]';
ys4 = [y1;y4]';
ys5 = [y1;y5]';

sound(ys2,Fs); % pi/2
pause(3)
sound(ys3,Fs); % pi
pause(3)
sound(ys4,Fs); % 0.5倍
pause(3)
sound(ys5,Fs); % 0.25倍

fig0 = figure;
hold on;
plot(t1,y1);
plot(t1,y2);
plot(t1,y3);
xlabel('time[s]');
ylabel('amplitude');
axis([0 0.01 -1.01 1.01])
legend({'pure tone','shift by 90 degrees','shift by 180 degrees'},'Location','southwest');
hold off;


fig1 = figure;
hold on;
plot(t1,y1);
plot(t1,y4);
plot(t1,y5);
yticks(-1:0.25:1);
xlabel('time[s]');
ylabel('amplitude');

axis([0 0.01 -1.1 1.1]);
legend({'pure tone','1/2 times amplitude','1/4 times amplitude'},'Location','southwest');
hold off;

exportgraphics(fig1,'../Figures/01_02_1.pdf','ContentType','vector');
exportgraphics(fig0,'../Figures/01_02_2.pdf','ContentType','vector');
```

src. A-5　01-03: うなり（第 1.3 節 → p.4）

```
clear;

Fs = 16000; %
f1 = 440;   % [Hz]
f2 = 441;   % [Hz]

t = (0:4*(Fs-1)) /Fs;
for k=0:Fs
    y1 = sin(2*pi*f1*t);
end

for k=0:Fs
    y2 = sin(2*pi*f2*t);
end

y = y1 + y2;
% sound(y,Fs);

fig0 = figure;
plot(t,y);
xticks(0:1:4);
xlabel('time[s]');
ylabel('amplitude');
axis([0 4 -2.01 2.01]);
exportgraphics(fig0,'../Figures/01_03.pdf','ContentType','vector');
```

src. A-6　01-04: フーリエ級数展開（第 1.4 節 → p.4）

```
clear;

Fs = 16000;
T = 1/2; % 周期
f = 1/T; % 周波数

t = (0:4*(Fs-1)) /Fs;

y = 0;
for k=1
    y = y + (1/(2*k-1)) * sin(2*pi*f*(2*k-1)*t);
end

fig0 = figure;
hold on;
plot(t,y);
y = 0;
for k=1:5
    y = y + (1/(2*k-1)) * sin(2*pi*f*(2*k-1)*t);
end
plot(t,y);
y = 0;
for k=1:25
    y = y + (1/(2*k-1)) * sin(2*pi*f*(2*k-1)*t);
end
plot(t,y);
axis([0 1 -3 3]);
```

```
xlabel('time[s]');
ylabel('振幅');
hold off;
legend({'N=1','N=5','N=25'},'Location','southwest');

fig1 = figure;
y = 0;
for k=1:50
    y = y + (1/(2*k-1)) * sin(2*pi*f*(2*k-1)*t);
end
plot(t,y);
axis([0 1 -3 3]);
xlabel('time[s]');
ylabel('amplitude');
exportgraphics(fig0,'../Figures/01_04_2.pdf','ContentType','vector');
exportgraphics(fig1,'../Figures/01_04_1.pdf','ContentType','vector');
```

src. A-7　01-05: 白色ガウス雑音（第 1.5 節 → p.5）

```
clear;

Fs = 16000;
t = (0 : (Fs-1)) /Fs;
tl = length(t);

rng(0,'twister'); % 乱数生成器 初期化

a = 0.2; % 標準偏差
b = 0; % 平均

y = a.*randn(tl,1) + b;

fig0 = figure;
plot(t,y);
xlabel('time[s]');
ylabel('amplitude');


num=100; % ヒストグラム分割数

[h, c] = hist(y, num);

fig1 = figure;
plot(c,h);
xlabel('amplitude');
ylabel('frequency[times]');

sound(y,Fs);
exportgraphics(fig0,'../Figures/01_05_0.pdf','ContentType','vector');
exportgraphics(fig1,'../Figures/01_05_1.pdf','ContentType','vector');
```

B　課題 2

src. B-1　02-01-1: フーリエ変換と周波数解析（第 2.1 節 → p.7）（純音）

```
clear;

Fs = 8192;
t = (0:(Fs-1)) / Fs;
f = 440;
fs = (-Fs/2 : Fs/1024 : (Fs/2) - Fs/1024);
y = sin(2*pi*f*t);

fft_y = fft(y,1024);
ln = length(abs(fftshift(fft(y))));
fft_ys = fftshift(fft_y);
fft_ys = abs(fft_ys);
fig0 = figure;
plot(fs,fft_ys);
xlabel('frequency[Hz]');
ylabel('amplitude');
axis([0 1000 -10 600]);
exportgraphics(fig0,'../Figures/02_01.pdf','ContentType','vector');
```

src. B-2　02-01-2: フーリエ変換と周波数解析（第 2.1 節 → p.7）（矩形波）

```
clear;

Fs = 8192;
t = (0 : 4*(Fs-1)) / Fs;
f = 440;
fs = (-Fs/2 : Fs/(128*2*4) : (Fs/2)-Fs/(128*2*4));

y_0 = zeros(1,128);
y_1 = ones(1,128);

y = [y_0 y_1];
y = [y y y y];
fig0 = figure;
plot(y);
xlabel('point');
ylabel('amplitude');
axis([0 1024 -0.5 1.5]);

fft_y = fft(y);
ln = length(abs(fftshift(fft(y))));
fft_ys = fftshift(fft_y);
fft_ys = abs(fft_ys);
fig1 = figure;
plot(fs,fft_ys);
xlabel('frequency[Hz]');
ylabel('amplitude');
axis([0 200 -10 600]);
exportgraphics(fig0,'../Figures/02_021.pdf','ContentType','vector');
exportgraphics(fig1,'../Figures/02_022.pdf','ContentType','vector');
```

src. B-3　02-02: LPF（ローパスフィルタ）（第 2.2 節 → p.9）

```
clear;
clc;

Fs = 8192;
t = (0 : 0.25*(Fs-1)) /Fs;
t8 = (0 : 2*(Fs)-1) /Fs;

f0 = 261.38;
f1 = 293.67;
f2 = 329.63;
f3 = 349.23;
f4 = 392.00;
f5 = 440.00;
f6 = 493.88;
f7 = 523.23;
y_0 = sin(2 * pi * f0 * t);
y_1 = sin(2 * pi * f1 * t);
y_2 = sin(2 * pi * f2 * t);
y_3 = sin(2 * pi * f3 * t);
y_4 = sin(2 * pi * f4 * t);
y_5 = sin(2 * pi * f5 * t);
y_6 = sin(2 * pi * f6 * t);
y_7 = sin(2 * pi * f7 * t);
y = [y_0 y_1 y_2 y_3 y_4 y_5 y_6 y_7];

fig0 = figure;
plot(t8,y);
xticks(0:0.25:2);
xlabel('time[s]');
ylabel('amplitude');
axis([0 2 -1.5 1.5]);

fft_y = fft(y);
fft_y = fftshift(fft_y);
f = (-Fs/2 : Fs/length(fft_y)  : Fs/2 - Fs/length(fft_y));

fig1 = figure;
subplot(2,1,1);
plot(f,fft_y);
xlabel('frequency[Hz]');
ylabel('amplitude');
axis([-1000 1000 -1500 1500]);

for k=1:length(fft_y)
    if f(k) >= 375.00 || f(k) <= -375.00
        fft_y(k) = 0;
    end
end
subplot(2,1,2);
plot(f,fft_y);
axis([-1000 1000 -1500 1500]);
xlabel('frequency[Hz]');
ylabel('amplitude');
ifft_y = ifftshift(fft_y);
```

```
55 ifft_y = ifft(ifft_y);
56 ifft_real = real(ifft_y);
57
58 fig2 = figure;
59 plot(t8,ifft_y);
60 xlabel('time[s]');
61 ylabel('amplitude');
62 xticks(0:0.25:2);
63 axis([0 2 -1.5 1.5]);
64 % sound(ifft_real,Fs)
65 exportgraphics(fig0,'../Figures/02_20.pdf','ContentType','vector');
66 exportgraphics(fig1,'../Figures/02_21.pdf','ContentType','vector');
67 exportgraphics(fig2,'../Figures/02_22.pdf','ContentType','vector');
```