

# 基礎デジタル計測

Basics of digital measurements

提出者：08A17153 羽田充宏  
共同実験者：08A17154 畠中俊平

実験実施日：2019年6月14日

実験実施日：2019年6月17日

実験実施日：2019年6月21日

実験実施日：2019年6月24日

実験実施日：2019年6月28日

実験実施日：2019年7月1日

## Introduction

この実験ではコンピュータやその周辺機器を使った実験の制御方法を学ぶ。また実験を通して C 言語を用いてコンピュータや電気回路を制御する方法を学習する。こうしたスキルを使うことで、話し手によりマイクロフォンより発せられる音声のキャプチャーを開発し、制御工学を学ぶ。さらに、キャプチャーした音声を信号処理することで周波数フィルターを作製する。

- パソコンを使つての C 言語学習 (1st day).
- AD 変換, DA 変換や C 言語を用いた AD/DA 変換を学習する (2nd day).
- 音声キャプチャーのための電気回路を製作する (3rd day).
- GNU Octave の学習および理解をした後、キャプチャーした音声の高速フーリエ変換 (FFT) を観察する (4th day).
- 周波数領域における信号処理 (5-6th day).

## 1 Learnig basic C programing Task 1.3 (1<sup>st</sup> day)

### 1.1 Purpose

今回の実験では C 言語プログラミングの基本的な操作を理解する。

### 1.2 Procedure

まず Windows OS のノートパソコンを用意し”Visual C++ 2010 Express”を起動する。その後与えられた 1.1 から 1.3 の課題に取り組む。

#### Task 1.1

整数 1 から 12234 までの総和を表示せよ。

#### Task 1.2

1 から 100 までの数を配列に格納したのち奇数の総和を表示せよ。

#### Task 1.3

1000 個の点をとることにより正弦波 (振幅=1, 位相=0) 5 波長分の値を text ファイルに出力しその出力結果をエクセルを用いて表示せよ。

### 1.3 Result

実験の結果まず, task 1.1 では 74841495 の値が得られた。

また task 1.2 では 25000 の値が得られた。

さらに task 1.3 では図 1 のようなグラフがエクセルにより得られた。

### 1.4 Discussion

今回の課題 task1.1 において 1 から 12234 までの総和を for 文を用いて計算を行なったが総和の公式から以下のようにも書き換えられるこ

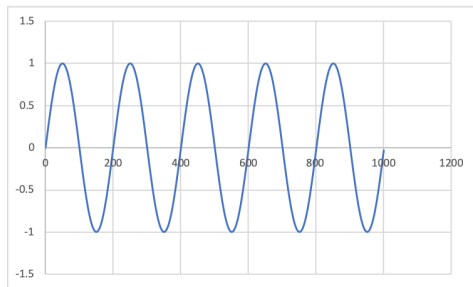


図 1: 振幅=1, 位相=0 の正弦波

とが予想できる.

$$\sum_{k=1}^n k = \frac{n(n+1)}{2}$$

この数式を用いて総和を求める場合の計算時間は  $O(1)$  で計算できる一方で for 文を用いて総和を求める場合は  $O(N)$  の時間がかかることが予想できる. またこの数式に  $n=12234$  を代入すると確かに 74841495 の値を得ることができるので今回の結果は正しいと考えられる.

また, task1.2 においても前問同様に総和の公式を用いると

$$\begin{aligned} \sum_{k=1}^n 2k-1 &= 2 \frac{n(n+1)}{2} - n \\ &= n^2 \end{aligned} \quad (1)$$

この式において  $n=50$  を代入すると 2500 の値が得られるのでプログラムによる計算結果は正しいと考えられる. また, 今回も同様に総和公式を使った場合の最悪時間計算量は  $O(N)$  で抑えられることができる.

## 1.5 Appendix

今回の課題で使ったソースコードを以下に示す.

ソースコード 1: task1.1

```
#include <iostream>
#include <math.h>
int main(){
    int sum = 0;
    for(int i=1; i<=12234;i++){
        sum += i;
    }
    cout << sum << endl;
}
```

ソースコード 2: task1.2

```
#include <iostream>
#include <math.h>
int main(){
    int sum = 0, a[100];
    for(int i = 0; i < 100; i++){
        a[i] = i + 1;
        if(i%2 == 0){
            sum += a[i];
        }
    }
    cout << sum << endl;
}
```

ソースコード 3: task1.3

```
#include <iostream>
#include <stdio.h>
#include <math.h>
#define PI 3.141592653
int main(){
    double a[1000];
    FILE *fp;
    fp = fopen("test.txt","w");
    for(int i = 0; i < 1000; i++){
        a[i] = sin(PI*i/100);
    }
    for(int i = 0; i < 1000; i++){
        fprintf(fp,"%f\n",a[i]);
    }
    fclose(fp);
}
```

## 2 Understanding AD,DA conversion , and C programing for controlling AD/DA conversion (2<sup>nd</sup> day)

### 2.1 Purpose

今回の実験ではアナログからデジタル、デジタルからアナログへの信号変換をコンピューターを用いて行うことが目的である。

### 2.2 Procedure

#### Task 2.1 DC 信号の取り込み

乾電池により作り出した直流電源の DC 信号を AD/DA ボードでパソコンに取り込み (AD 変換), 取り込んだファイルをテキストファイルとして出力する. テキストファイルをエクセルにより描画して電圧の推移を調べる. その変化量より量子化について考える.

#### Task 2.2 時間変化する信号の取り込み (サンプリング定理)

ファンクションジェネレーターによって正弦波を出力し, その電圧を 10Hz, 50Hz, 100Hz のサンプリング周波数で取り込む. また取り込んだ値をエクセルにより描画する. この操作をファンクションジェネレーターで 10Hz, 20Hz, 50Hz の三種類の正弦波について行う. この際波形の確認はオシロスコープを用いて確認を行う.

#### Task 2.3 DA 変換

C プログラムにより”ド”と高い”ド”の音階の周波数それぞれ 262Hz, 523Hz を発生させそれをオシロスコープにより観測した.

### 2.3 Result

#### Task 2.1

実験の結果以下のようなグラフが得られた.

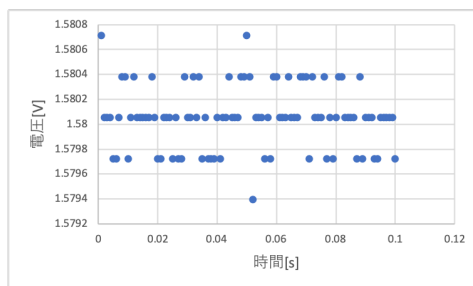


図 2: 乾電池の AD 変換

また最小値を 0 に設定した際は以下のように  
なる。

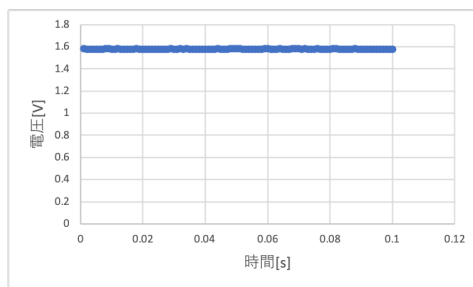


図 3: 乾電池の AD 変換 (2)

この結果より乾電池の電圧は一定値を取ること  
が読み取れる。

## Task 2.2

測定の結果それぞれの正弦波について以下のよ  
うなグラフが得られた。

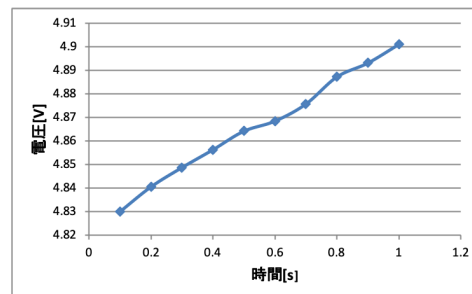


図 4: sin 波 (10Hz), サンプル周波数 (10Hz)

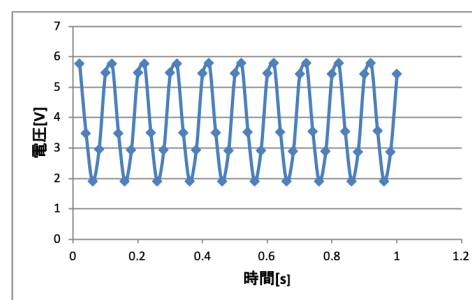


図 5: sin 波 (10Hz), サンプル周波数 (50Hz)

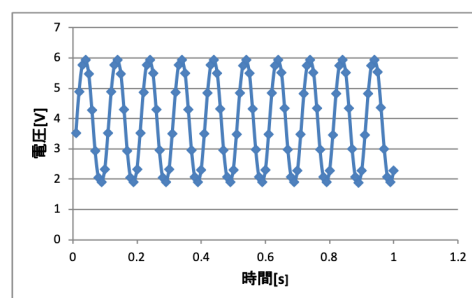


図 6: sin 波 (10Hz), サンプル周波数 (100Hz)

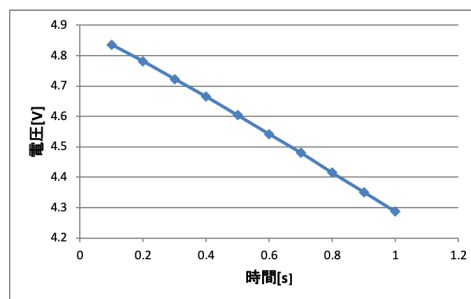
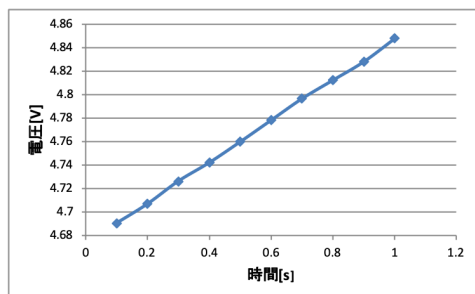


図 7: sin 波 (20Hz), サンプルング周波数 (10Hz)    図 10: sin 波 (50Hz), サンプルング周波数 (10Hz)

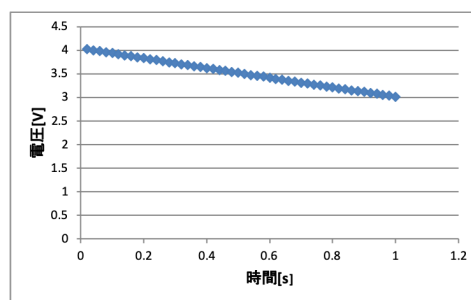
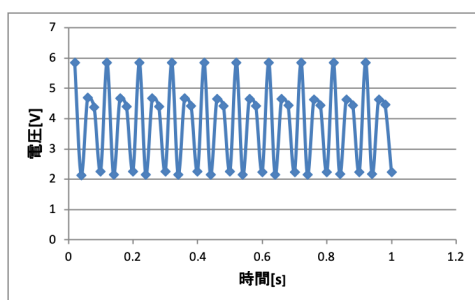


図 8: sin 波 (20Hz), サンプルング周波数 (50Hz)    図 11: sin 波 (50Hz), サンプルング周波数 (50Hz)

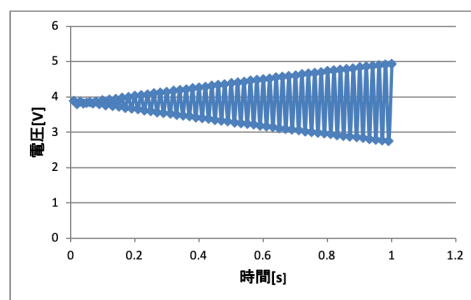
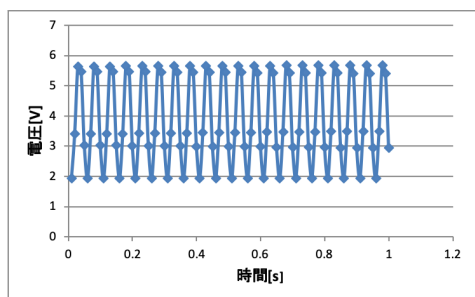


図 9: sin 波 (20Hz), サンプルング周波数 (100Hz)    図 12: sin 波 (50Hz), サンプルング周波数 (100Hz)

**Task 2.3 DA 変換実験よりオシロスコープから得られた波形を以下に示す。**

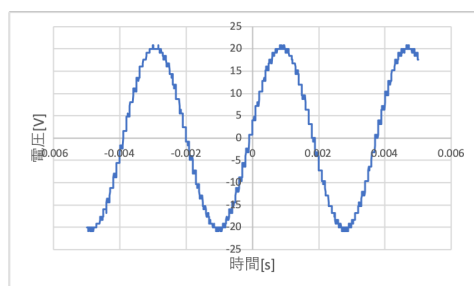


図 13: sin 波 (262Hz), サンプル周波数 (10kHz)

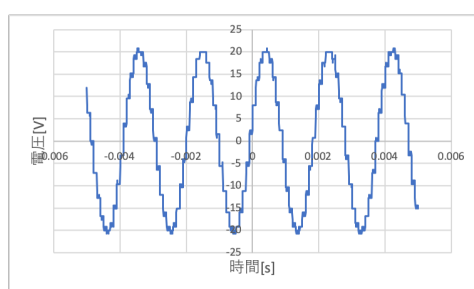


図 14: sin 波 (523Hz), サンプル周波数 (10kHz)

## 2.4 Discussion

### Task 2.2

電池の電圧は 1.5V で一定値を取るため正しく測定できたことが予想できる。またグラフは 0.0003V ごとに離散値を取ることが読み取れる。これは今回使用した National Instruments の計測器が 16bit の制度を持ち今回のプログラムでは測定範囲を 10V にしたため  $20/2^{16} \sim 0.0003$  となることより測定できる最小単位が 0.0003V 刻みであるためだと考えられる。また測定の際に 5 種類ほどの測定値が出てきたのは測定中の

振動等のノイズによりものと予測でき、その値は誤差として考えらえる。

また正弦波の測定の実験において図 4,7,10,11 のように正弦波の周波数がサンプリング周波数の約数となると正弦波の位相の等しい地点を測定していることになるので直線に近い値となることが予想でき実験結果とも概ね一致する。ここで直線に傾きが生じてしまうのはサンプリング周期に若干のずれが生じているからだと考えられる。

さらに図 6,9 のようにサンプリング周波数が正弦波の周波数の周波数に比べ十分に大きい場合はサンプリング測定により正弦波の概形を正確に測定することができることが確認された。

一方で図 5,8 のように正弦波の周波数に対してサンプリング周波数が十分に大きくない場合正弦波の一部が欠損してしまうことが考えられる。また Task2.3 の実験では低い”ド”の音波が高い”ド”の音波に比べて周波数が低いことが読み取れ、それぞれの周期の逆数を取ると 262Hz,523Hz に一致することより正しく正弦波を生成できたことが予想できる。また今回のサンプリング周波数では測定正弦波がギザギザになったがサンプリング周波数を上げていくことで滑らかな正弦波を測定することが可能になると考えらる。

### 3 Making electric circuits for capturing sound waves (microphone) and emitting sound waves (speaker) (3<sup>rd</sup> day)

#### 3.1 Purpose

今回の実験では電気回路を作りその回路を使い、音声を電気信号に変えたり逆に電気信号を音声に変える方法を学ぶことが目的である。

#### 3.2 Procedure

##### Task 3.1

まず音声信号を電気信号に変換するための電気回路を作る。図 15 に示されるように電気回路を製作する。この際プリント基板に半田付けして回路を実装する。

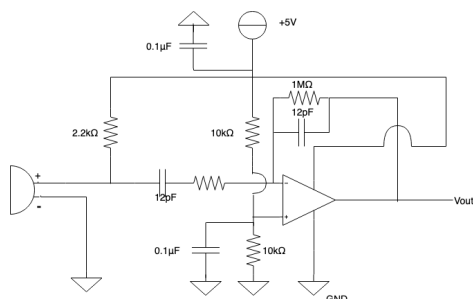


図 15: capture 回路

##### Task 3.2

先ほどの実験で作った capture 回路を AD/DA 変換機に接続し C プログラムにより音叉の音声データを取得する。この際にサンプリング定理に気をつけて音波を測定する。また回路の都合上出力電圧が 2.5V 付近を中心に 0V から 5V の間を振動することに気をつける。取得したデータをテキストファイルとして保存する。また今

回の実験ではデータ数を 1000, サンプリング周波数を 10kHz として測定をおこなった。

##### Task 3.3

次にデジタル信号を音声に変換するためのスピーカーを作る。図 16 に示すようにスピーカー回路を作る。その後 C プログラムにより作った正弦波を出力し音波を発生させる。さらに周波数が近いと同時に鳴らすことによりうなりを観測する。この際入力電圧は 0V を中心として電圧を入力することに注意する。

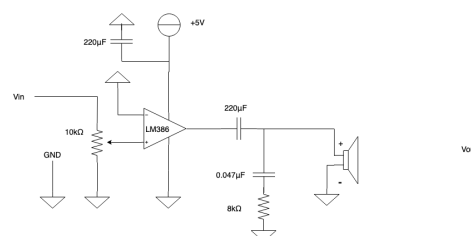


図 16: speaker 回路

##### Task 3.4

マイクを用いて声を録音し、録音時と同じサンプリング周波数、データ数で再生し正しく再生できたかを確認する。次にサンプリング周波数を録音時のサンプリング周波数とくらべ 2 倍, 3 倍にして声がどのように変化したのかを確認する。この際データ数を 20000 点、サンプリング周波数を 20kHz として「こんにちは」という声の測定を行なった。

#### 3.3 Result

##### Task 3.1 電気回路の作成

今回はマイク回路がうまく作ることができなかったため既製品を使用した。スピーカー回路に関しては正弦波を入力とし音を鳴らすことで動作を確認した。

##### Task 3.2

実験より以下の図に示すような波形を得た。



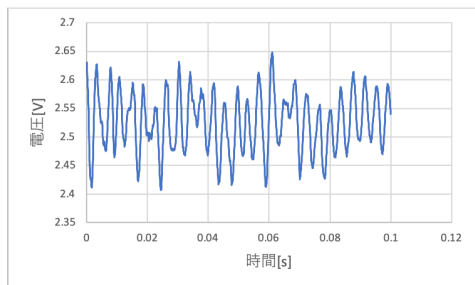


図 17: 262Hz 音叉

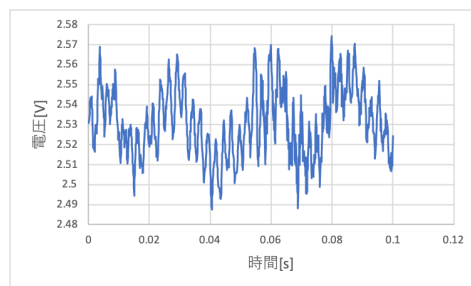


図 20: 392Hz 音叉

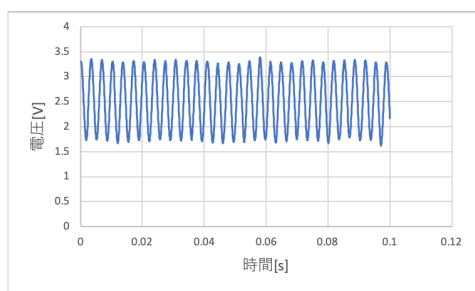


図 18: 294Hz 音叉

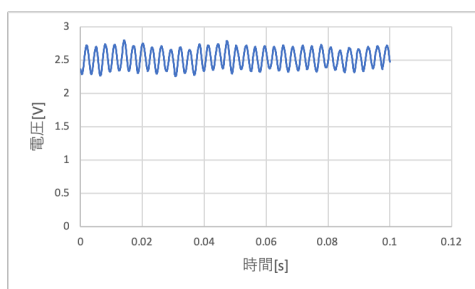


図 19: 330Hz 音叉

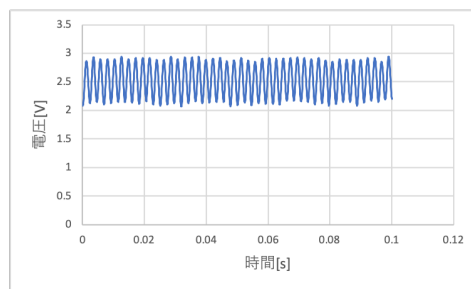


図 21: 440Hz 音叉

### Task 3.3

スピーカーで入力正弦波を 262Hz から 440Hz までの 5 種類を順番に再生したところ、再生された音が徐々に高くなっていくことが確認された。また 263Hz の正弦波をプログラムにより作り出しスピーカーで出力し、同時に 262Hz の音叉を鳴らしたところ約 1 秒ごとにうなりを観測できた。

### Task 3.4

測定した際と同じサンプリング周波数、データ点数で音声をスピーカーで再生したところ元の音源と同じ「こんにちは」という音を観測することができた。また、サンプリング周波数を計測時の 2 倍、3 倍である 40kHz、60kHz にしたところ音声は高い声となって再生された。

## 3.4 Discussion

### Task 3.1

マイク回路で取得したデジタル信号をスピーカー回路接続することで適切な音源を得られたのでこれらの回路は正確に昨日していたと考えられる。

### Task 3.2

図 17, 20 図 17 から周期的な波が得られているので音叉の音波を正確に即てできたと考えられる。一方図 17, 20 においては波が周期的ではなく振幅もバラバラとなってしまった。この原因としては測定するタイミングが音叉の振動している範囲に比べて早すぎた、または遅すぎた、また外部の音がノイズとして入り込んだなどが考えられる。

また今回のサンプリング間隔が適切であったかの考察をする。まずサンプリング定理より [1] サンプリング間隔  $T$ 、最大周波数  $f_M$  が満たすべき条件は以下のようなになる。

$$T < \frac{1}{2f_M} \quad (2)$$

今回の測定で最大周波数は 440Hz であり、サンプリング間隔は  $1.0 \times 10^{-4} [s]$  であるので明らか

に条件を満たしていると考えられる。したがって今回の実験では適切に量子化できたと考えられる。

### Task 3.3

今回の実験で正弦波の周波数を上げてい行った際にスピーカーから聞こえた音も次第に高くなっていったため適切に出力できたと考えられる。また二つの正弦波 (周波数  $f_1, f_2$ ) の合成によるうなり周期  $T$  は以下のように表される。

$$T = \frac{1}{|f_1 - f_2|} \quad (3)$$

したがって、263Hz の波を出力した際に 262Hz の音叉と約 1 秒間隔のうなりが聞こえたことはこの公式より適切であると考えられる。

### Task 3.4

声の測定時はサンプリング周波数は 20kHz、データ数は 20000 個であったので再生する際のサンプリング周波数を測定時と同じ 20kHz にした場合測定時と同じ速度で計測した声が再生されたと考えられ実験結果に一致する。さらに再生時にサンプリング周波数を 40kHz にした場合には 20kHz で再生した場合に比べて倍速で再生されると考えられる。その結果再生される声の周波数は 2 倍となり結果として高い声出力されたと考えられる。同様にサンプリング周波数を 30kHz にした場合は周波数が 3 倍となったと考えられる。逆にサンプリング周波数を 10kHz にした場合は周波数が 1/2 倍されると考えられ低い声が観測されると考えられる。

## 4 Learnig Matlab compati- ble scientific programing language software (Octave) and understanding First Fourier Transform (FFT) of the captured sound waves (4<sup>th</sup> day)

### 4.1 Purpose

今回の実験では Matlab 互換の数値計算ソフトウェアである GNU Octave を用いて取り込んだ音声データをフーリエ変換 (離散型高速フーリエ変換) することで、周波数分解する。これらの実験を通して、信号のフーリエ変換を理解する。

### 4.2 Procedure

#### Task 4.1

任意の周波数、振幅の Sin 波および Cos 波のフーリエ変換を行い、それぞれ実部、虚部、パワースペクトルを表示する。この時横軸を Hz にすることに注意する。またこの得られたグラフが何を意味するのかを考える。今回の実験では振幅 1、周波数 262Hz の Cos 波 Sin 波の FFT を行った。

#### Task 4.2

Task3.2 と同様の方法で音叉 (二種類) から音声信号を読み込みデジタル信号を取得する。この際サンプリング周波数を 1kHz、データ数 1000 として測定を行なった。今回は 330Hz と 440Hz の音叉をそれぞれ鳴らした。その後その時間領域でのデジタル信号を Octave でフーリエ変換し周波数領域の信号を得る。得られた周波数領域の信号において 330Hz, 440Hz の周波数が含まれているのかを調べる。

### 4.3 Result

#### Task 4.1

262Hz の正弦波、余弦波のフーリエ変換の結果を以下に示す。グラフは上から実部、虚部、パワースペクトルの順で表されている。

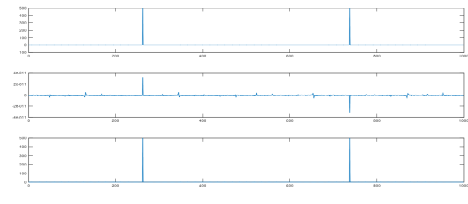


図 22: 262Hz 余弦波 FFT

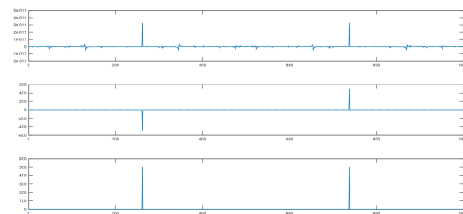


図 23: 262Hz 正弦波 FFT

#### Task 4.2

330Hz と 440Hz の音叉をそれぞれ鳴らした際にマイク回路により得られた音波を 0V 中心に振動するように 2.5V 引いた時の測定値をプロットしたグラフを図 24, 図 25 に示す。さらにその結果をフーリエ変換した際の周波数領域のグラフを図 26, 27 に示す

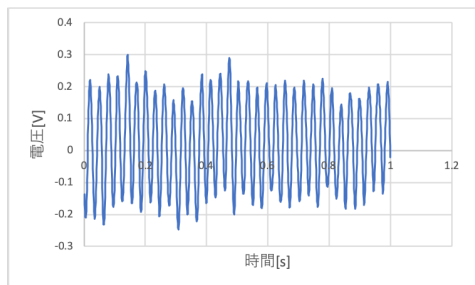


図 24: 330Hz 音叉の測定値

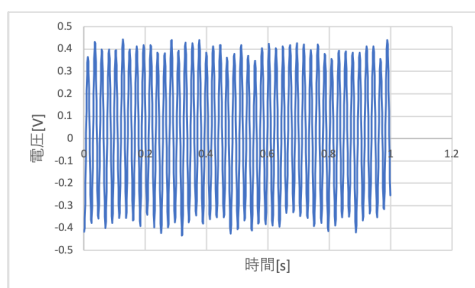


図 25: 440Hz 音叉の測定値

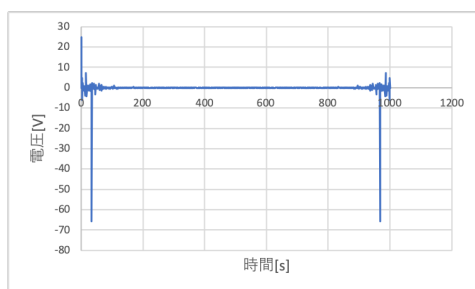


図 26: 330Hz 音叉の FFT

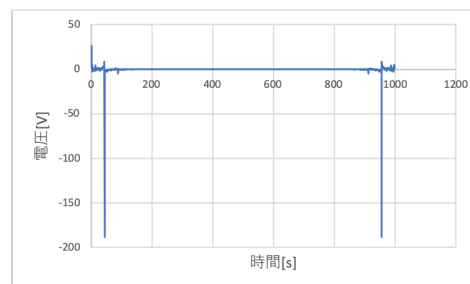


図 27: 440Hz 音叉の FFT

## 4.4 Discussion

### Task 4.1

まず今回の実験におけるのフーリエ変換の式は以下のように与えられる.

$$F(w) = \int_{-\infty}^{\infty} f(t)e^{-2j\pi wt} dt \quad (4)$$

このことから今回の FFT の結果は横軸が角周波数  $w$  になるのではなくサンプリング周波数  $f$  になることが読み取れる. したがって, 図 22 において FFT した結果時間領域 [s] の余弦波が周波数 [Hz] 領域の関数に変換されたと考えられ, 3 番目のパワースペクトルのグラフにおいて 262Hz 付近の値が大きいことは時間領域における信号が 262Hz の周波数を多く含む振動をしていたことを意味するが, 実際に今回は 262Hz の余弦波をフーリエ変換したので正しく結果が表示されていると考えられる.

また真ん中の虚部の表示においては  $10^{-11}$  オーダーでの値を示しているのはほぼゼロとみなせる. 一方で実部の値は他の値に比べて十分に大きな値を取っていることがわかる. この原因を考察する. まず, 今回の実験でフーリエ変換の定義域を  $[0, 1-0.001]$  とする. この定義域においてフーリエ変換で積分を行うと正弦波と余弦波の積の積分値が 0 となるため周波数領域での  $F(w)$  が虚部を含まない関数となったことが考えられる. 同様の理由で図 23 の実部に

において値がほぼ0となっていることにも説明がつく。

さらに今回の時間  $t$  は Octave で  $t = [1:0.001:1]$  と定義されているので、1秒間に1000点のサンプルを取ることと同値であると考えられるため今回のサンプリング周波数は1000[Hz]と予想できる。またFFTの結果が横軸が500のところで折り返しの関係になっている理由としては、まずフーリエ変換においては周波数はサンプリング定理より  $F(w)$  はサンプリング周波数の周期で繰り返されていると予想できる。つまり周波数領域においてサンプリング周波数を  $f$  とすれ  $[-f/2, f/2]$  の繰り返して  $F(w)$  が繰り返されていると解釈できる。今回実験の結果で表示されているのは  $[0, f]$  となっている。したがって結果が  $f/2$  において線対称になっていることに対して説明がつく。

#### Task 4.2

測定において時間領域の信号データは一定の周期で振動していることが確認できるので測定に際して大きなノイズは発生しなかったと考えられる。測定はサンプリング周波数10kHz、サンプリング数1000点で行なったので  $F(w)$  の繰り返しの周期は1000となり、周波数の軸は1/10に収縮していると考えられる。実際FFTしたグラフを観察するとそれぞれ  $f = 34$ ,  $f = 45$  のところにトゲが位置しているので概ね正しく測定できたと考えられる。ところでFFTしたグラフは負の値を持っている。これはOctaveでabsコマンドを打ち損じたことが原因だと考えられる。フーリエ変換は時間領域で与えられた関数がどのような周波数を含んでいるのか調べるのが目的のため結果はパワースペクトルで示すのが正しいと考えられる。また周波数軸は(サンプリング周波数)/(データ数) 伸縮するので時間領域の信号にどのような周波数が含まれているのか厳密に知りたいときはサンプリング周波数に対してデータ数を多取ると周波数領域におい

てグラフが引き伸ばされ測定しやすくなると考えられる。

## 5 Signal processing in frequency space (Highpass and lowpass filter) (5 – 6<sup>th</sup> day)

### 5.1 Purpose

今回の実験では取り込んだ音声信号の周波数空間 (フーリエ空間) での信号処理を学ぶ。特定の周波数のみを通す簡単な周波数フィルタの実戦から、窓関数を用いたローパスフィルタおよびハイパスフィルタを学ぶ。具体的には、異なる共鳴周波数を有する複数の音叉からの音声信号の中から、信号処理によって特定の周波数のみを取り出す。また人が発する高い声と低い声を同時に録音し、信号所によって高い声のみ、または低い声のみを取り出す。

### 5.2 Procedure

#### Task 5.1 特定周波数の取り込み

異なる周波数を有する音叉を二つ選びそれらを同時に鳴らし、音声信号を取得する。その音声信号を一方のみを通す周波数フィルタを Octave で政策しフーリエ変換した周波数領域の信号に掛け算する。次にそれを逆フーリエ変換を行なってスピーカーで鳴らすことで正しく分離できたかを確認するとともにオシロスコープで波の振動数を計測する。今回は 330Hz と 440Hz の音叉を同時に鳴らしてフィルタにより 440Hz の音叉成分のみを取り出すようなフィルタを用意した。また今回のサンプリング周波数は 20kHz、サンプリング数は 20000 点とした。

#### Task 5.2 ハイパスフィルタ、ローパスフィルタ

同時に高い声と低い声で異なる言葉を発生し、その声をコンピュータに取り込む。取り込んだ音声信号をフーリエ変換する。その後 Octave で窓関数を作り  $[0, F]$  までのデータ点数を 0,  $[F+1, 40000]$  までを 1 にするステップ関数をか

けて逆フーリエ変換して低い声のカットされるような整数  $F$  を探した。今回はサンプリング周波数 20kHz、サンプリング数 40000 点として測定した。Task 5.3 ボイスチェンジャー  
声を取り込み、周波数を変化させることで、声の高さを変えてスピーカーで再生する。

### 5.3 Result

#### Task 5.1 特定周波数の取り込み

取り込んだ時間領域の声の信号は図 28 のようになった。また図 29 はそのうち最初の 1000 点のみを表示した。またこの信号のフーリエ変換は図 30 のようになった。さらにこのフィルタ後の関数を逆フーリエ変換したのちスピーカーで再生した。さらにオシロスコープで周波数を確認すると 440Hz の値を得た。

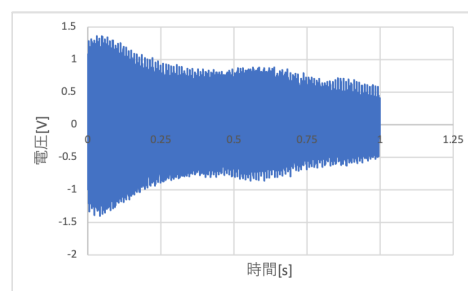


図 28: 330Hz, 440Hz 音叉

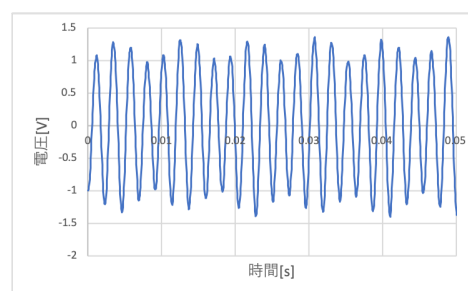


図 29: 330Hz, 440Hz 音叉

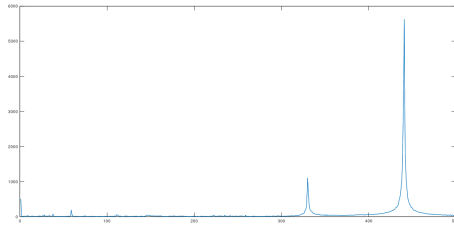


図 30: 信号の FFT

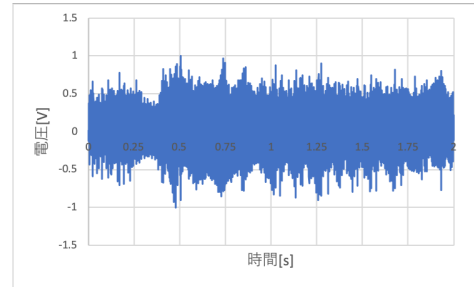


図 32: 取り込んだ声

### Task 5.3 ボイスチェンジャー

前回の実験同様サンプリング周波数をあげて先ほど取り込んだ音声を再生すると高い声に声を変化させることに成功した。

## 5.4 Discussion

### Task 5.1 特定周波数の取り込み

取り込んだ時間周波数領域の信号は周期的に振動していたので適切に測定することができたと予想できる。また FFT したのちのグラフにおいては 330Hz と 440Hz にツノが立っていることよりも正しく取り込めたと予想できる。

### Task 5.2 ハイパスフィルター，ローパスフィルター

サンプリング周波数 20kHz，サンプリング数 40000 としたので 1 データあたり 0.5Hz 刻みになっていると予想できる。したがって窓関数を作った際に  $F = 3000$  を境に低い声が聞こえなくなったということは低い声は 1500Hz 以下の声であったと考えられる。

### Task 5.3 ボイスチェンジャー

前回の実験で考察したので割愛する。

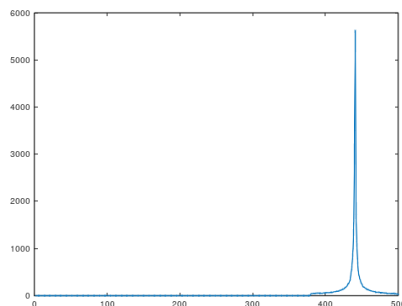


図 31: フィルター後の関数

### 全体のまとめ

今回は窓関数にステップ関数を用いたがこれはフーリエ逆変換する際に微分不可能点を生じさせることになり境界が滑らかにならないという欠点がある．そのため窓関数を正規分布のガウス関数を用いたり  $\cos$  関数を用いるなどしてフィルターをかけた際の周波数領域関数を滑らかにすることでより正確な分離を行うことができると予想できる．

最後にフーリエ変換の応用例について考える．今回はローパスフィルター，ハイパスフィルターを用いて音声をカットすることを考えたがこれは音声ファイルや画像ファイルにおいてデータ量を削減したいときに人間の目や耳では感知できないような高周波，低周波成分をカットすることによりデータ量を削減するなどの技術を考えることができる．

### 参考文献

- [1] なっとくするフーリエ変換 小暮陽三
- [2] デジタル諡号処理 府川和彦



## 学生実験レポートに関するチェックシート

学籍番号

氏名

各自チェックを行い、学生実験レポートの最終ページに添付して提出してください。レポートが受領されるには、全項目に斜線（もしくは塗りつぶし）が記入されている必要があります。

以下の項目を確認し、事実である場合は□に斜線（／）を入れて下さい。

- ☐ 友達または先輩のレポートを写していない。
- ☐ レポートに用いた図や文章は自分で作成したものである。（同じ実験グループの人とデータを共有することは許可するが、図の作成作業は各自行うこと。）
- ☐ Web ページの図や文章を copy-paste をしていない、また参考文献としていない。（参考文献には信頼できる情報（書籍等）を活用すること。web の情報を参考にすることは問題ないが、かならず信頼できる書籍等で内容を確認すること。）
- ☐ 実験テキストの図を scan または撮影してそのままレポートに用いていない。（手書きまたはグラフィックソフトによって新たに書きなおすことは許可する。）
- ☐ 図と表の情報が重複していない。
- ☐ 実験結果のグラフをコンピューターのドロー機能（例 WORD の描画機能）で書いていない。
- ☐ 式には通し番号が付けられており、かつレポート本文でもそれらが引用されている。（式が無い場合は、□を塗りつぶし）
- ☐ 図と表には通し番号、内容説明が付けられており、かつレポート本文でもそれらが引用されている。（図、表が無い場合は、□を塗りつぶし）
- ☐ グラフ（オシロスコープで測定した波形を含む）には、横軸と縦軸の内容が明記されている。（グラフが無い場合は、□を塗りつぶし）
- ☐ レポートを書く際に参照した文献を、参考文献として明記している。（多くの書籍を参照してレポートを作成することを推奨するが、必ず引用元を明記すること。文献を参照していない場合は、□を塗りつぶし）