

令和4年度 卒業研究

# トランペット初心者のための 基礎練習支援システムの開発

Development of a basic practice support system for trumpet beginners

函館工業高等専門学校  
生産システム工学科 情報コース  
稲垣 和音  
指導教員 東海林 智也

# 目次

1. 序論 .....	3
1.1 概要 .....	3
1.1.1 和文.....	3
1.1.2 英文.....	3
1.2 研究背景 .....	4
1.3 研究目的 .....	5
1.4 開発環境 .....	5
2. 関連技術 .....	6
2.1 フーリエ変換 .....	6
2.2 Numpy .....	7
2.3 Sounddevice .....	7
2.4 Tkinter .....	8
3. プログラム概要 .....	9
4. 研究結果 .....	10
4.1 実行結果 .....	10
4.2 解析結果と考察 .....	11
5. 今後の展望 .....	12
参考文献 .....	12
謝辞 .....	12

# 1. 序論

## 1.1 概要

### 1.1.1 和文

本研究の目的は、トランペット初心者を対象とした基礎練習支援システムの開発をすることである。GUI を用いて音の録音を行い、定期的に周波数解析を行うことで音の変化を判断するシステムを開発する。

### 1.1.2 英文

The purpose of this study is to develop a basic practice support system for trumpet beginners using frequency analysis. This study was developed in the Python environment using Anaconda. Fast Fourier Transform was used as the frequency analysis. By running it periodically, we want to determine what sound is being played every 0.25 seconds and display the achievement level. The results showed that the FFT of the input data was successful, but it was not possible to determine what sound was currently being emitted from it.

## 1.2 研究背景

部活動や趣味で楽器を始めたいと考える人は3人に1人以上いるといわれている[1]。その一方で約4割の人が挫折経験をしているという。

楽器を一から始めようすると、楽器教室などに行って指導してもらうか、独学で学んでいくことのどちらかであるため、趣味として始めるのにはハードルが高い。また、周りに楽器経験者で指導してくれる人がいたとしても、日頃からチェックを行い、適切なフィードバックをしてくれるというケースも少ない。よって「何を練習すれば上達し、演奏能力が向上するのかがわからない」という悩みを多くの楽器初心者が持っている。特に、管楽器に関しては何をすれば上手になることができるかなどを理解するのは困難である。

## 1.3 研究目的

本研究では管楽器のひとつであるトランペット初心者を対象に、最も基本的な練習方法である基礎練習を支援するシステムを開発する。

このシステムは、演奏者が吹き込んだ音から現在の完成度を判断し、上手さに応じて更に適切な練習方法を提案する。

これにより楽器を始めてみたいと考えている人の挫折経験を減らすことを目指す。

## 1.4 開発環境

PC : Surface\_pro\_7\_1866

OS : Windows10

CPU : Corei5-1035G4

実装 RAM : 8GB

使用言語 : Python3.9.7

開発環境 : Anaconda, Jupyter Notebook

使用ライブラリ : Numpy, Sounddevice, Tkinter

## 2. 関連技術

### 2.1 フーリエ変換

本研究では、周波数解析を行うためにフーリエ変換を用いた。フーリエ変換の中でも高速フーリエ変換(Fast Fourier Transform: FFT)を用いて解析を行った。この高速フーリエ変換は離散フーリエ変換(Discrete Fourier Transform :DFT)の計算時間を短くするためのアルゴリズムであり、DFT の定義は下記の通りである[2]。

$$f_k = \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi}{N}kn} \quad k = 0, \dots, n-1$$

e は自然対数の底、j は虚数単位、 $\pi$  は円周率である。このように DFT では一つの周波数データに対して  $N^2$  回演算を繰り返し行う必要がある。例えば  $N=512$  の場合、 $512^2 = 262,144$  回の計算が必要となるため、計算時間も長くなる。

一方 FFT ではデータ数  $N$  個の DFT を最終的に 2 点の DFT まで分割することにより、演算回数を  $N^2$  回から  $N * \log_2 N$  回に減らすことができる。 $N=512$  の場合、計算回数は 4,608 回となり約 25,000 回の計算回数が削減されていることがわかる。音声処理ではデータ数が膨大なため FFT が用いられる。FFT の定義は下記の通りである[3]。

$$f_k = \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi}{N}kn}$$

$$f_k = \sum_{n=0}^{M-1} x_{2n} w_M^{nk} + \sum_{n=0}^{M-1} x_{2n-1} w_M^{nk}$$

ここで  $w = e^{-j\frac{2\pi}{N}}$  は回転子と呼ばれ単位円を  $N$  分割した点として定義されている。

ただし FFT はデータ数を 2 の累乗にする必要がある。本研究では、0.25 秒ごとにデータを取り出して FFT を行い、さらに Wave ファイルのサンプリング周波数は 44.1KHz としたので 0.25 秒間に  $44100 \div 4 = 11025$  個のデータがある。そこで一番近い 2 の累乗の値である  $2^{14} = 16384$  個ずつデータを取り出すことにした。

FFT を実行した後でパワースペクトルを求めて周波数解析を行った。P(f) をパワースペクトル、X(f) をフーリエスペクトルとするとその関係は下記の通りとなる。

$$P(f) = |X(f)|^2 = X(f) * X(f)$$

## 2.2 Numpy

Pythonでは通常リストを用いて演算処理を行うため処理に時間がかかる。一方Numpyを使用することで配列を使った計算が行えるため処理速度を向上できる[4]。

本研究でを使用した音声ファイルには一秒間に 44,100 個のデータが含まれるため、周波数解析を行う際に非常に時間がかかる。そこで計算処理速度向上のため今回の研究では Numpy を使用した。

## 2.3 Sounddevice

Sounddevice は音声の録音、再生のために使用したライブラリである[5]。Python3.9でも利用可能で、Numpy 配列を扱うことができるということが出来る点から今回の研究では Sounddevice を使用した。

## 2.4 Tkinter

Tkinter は Python で GUI を使用するためのライブラリである[6]。ボタンの表示 (図 1) や楽譜の表示 (図 2) などの操作を行うために使用した。Python に初期から搭載されているライブラリであり、汎用性が高いということができることから今回の研究では Tkinter を使用した。



図 1 ボタンの表示



図 2 楽譜の表示



### 3. プログラム概要

プログラムを実行すると待機画面が出てくる。そこで処理開始のボタンを押すとカウントダウンが始まり、カウントが0になると楽譜が表示される。それと同時に指定時間以内にマイクに向かって楽器を吹きこむ。マイクに吹き込まれた音声データは Wave ファイルとして保存される。吹きこみが完了すると、Wave ファイルに対して 0.25 秒おきに FFT を実施してパワースペクトルを求め、さらに周波数解析を行って音の正誤判定を行う。それによって吹き込んだ音が最適なかを判定させ達成度を表示する。

## 4. 研究結果

### 4.1 実行結果

プログラムを実行すると図3のような待機画面が表示される。処理開始のボタンを押すと図4のように10秒間のカウントダウンを行い、カウントが0になると楽譜の表示と同時に録音を開始させる。録音終了後に周波数解析を実行し、その結果を表示する。

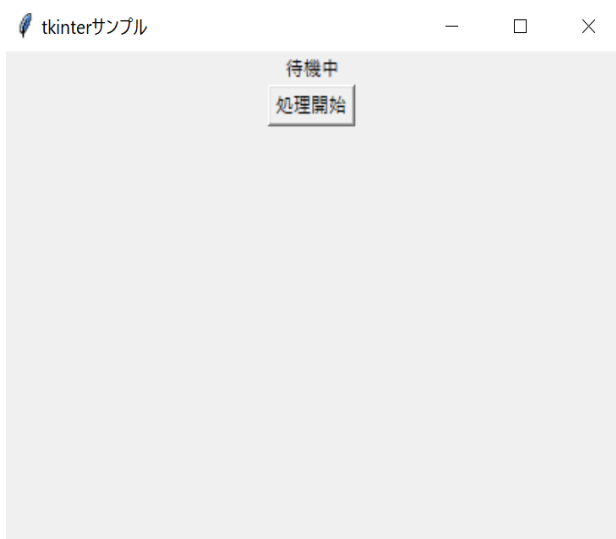


図3 待機画面



図4 カウントダウン

## 4.2 解析結果と考察

読み込んだ Wave ファイルに周波数解析を行った結果が図 5, 6 である。なお 0.25 秒ごとに FFT の実行を行っているため、図 5 と図 6 では FFT の結果に多少のずれが生じている。

今回は周波数解析のサンプルとして周波数が 261.626Hz の C4 の解析を行ったが、261Hz 付近にピークが来るという予想していた結果は得られず、現在出ている音の判断ができなかった。

正しい結果が得られなかった理由は周波数の定数倍である倍音成分を拾ってしまったからだと思われる。また周りの音を雑音として拾ってしまっていることも考えられる。

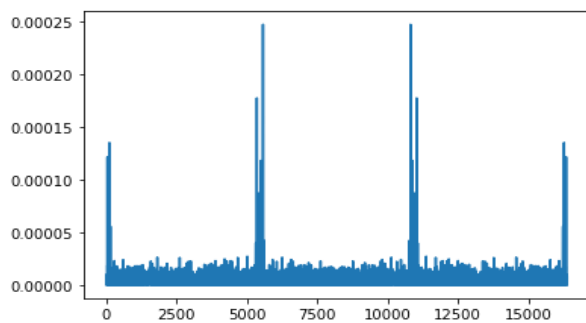


図 5 FFT 実行後 (0~0.25 秒)

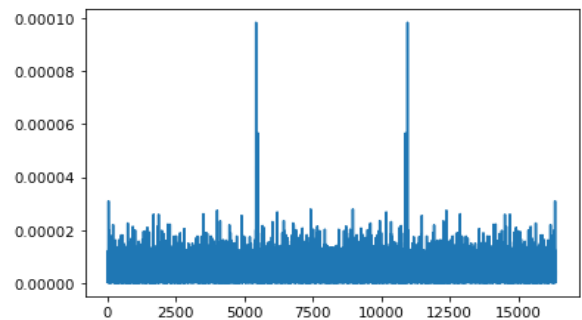


図 6 FFT 実行後 (1~1.25 秒後)

## 5. 今後の展望

今回は周波数解析を上手く行うことが出来なかった。そこで今後はノイズ除去処理を行って雑音とトランペットが出している音を分け、トランペットが出している音だけを周波数解析するようにプログラムを改良する。

また今回は、周波数解析を定期的に行うという処理も出来なかったため、この処理を今後実装する。

将来的には、「何が苦手か」「何が原因で上手くいかないか」を AI に判断させて最適な練習方法の提案を行う機能、基礎練習の種類（ロングトーン・スケールなど）を選択する機能、詳細情報（dur・テンポ・拍数など）を表示する機能等を実装する予定である。

## 謝辞

本研究を進めるにあたり、ご指導くださいました指導教員の東海林智也准教授に感謝いたします。

## 参考文献

[1] 楽器演奏に関する調査。働く男女、3人に1人以上は今後楽器を始めたい！

<https://prtimes.jp/main/html/rd/p/000000004.000024395.html>

[2] 高速フーリエ変換の並列処理に関する研究

<https://core.ac.uk/download/pdf/229757255.pdf>

[3] FFTを使ったバイオリン音のピッチ検出について

[https://www.jstage.jst.go.jp/article/inctkiyoupre/41/0/41\\_KJ00005074107/\\_pdf/char/ja](https://www.jstage.jst.go.jp/article/inctkiyoupre/41/0/41_KJ00005074107/_pdf/char/ja)

[4] Numpy

<https://numpy.org/>

[5] Sounddevice0.4.5,

[https://pypi.org/project/sounddevice/\(2022\)](https://pypi.org/project/sounddevice/(2022))

[6] Tkinter

<https://docs.python.org/ja/3/library/tkinter.html> (2022)