

令和5年度 卒業研究

ギターの基礎練習支援システムの開発

Development of a support system for those who practice the basics of guitar

函館工業高等専門学校
生産システム工学科 情報コース
後藤 快地
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 高速フーリエ変換	6
2.2 Numpy	7
2.3 pyaudio	7
2.4 Tkinter	8
2.5 matplotlib.....	8
3. 研究結果	9
3.1 実験内容	9
3.2 解析結果と考察	10
5. 今後の展望	11
謝辞	12
参考文献	12

1.1 概要

1.1.1 和文

ギターはソロでの演奏はもちろん、バンドの演奏には欠かせない楽器であり、人気で身近な楽器である。楽器というものは一人では始めるには非常に難しいモノであり何かからすればよいかわからなく、個人での上達は難しい。

そこで本研究では、音声データの高速フーリエ変換とデータ解析を用いて、初心者に最適な練習方法を提案するシステムの開発を行う。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。

1.1.2 英文

The guitar is a familiar instrument that is popular not only for solo performances but also for band performances. It is very difficult to start playing a musical instrument by yourself, and it is difficult to improve by yourself because you don't know where to start.

Therefore, in this study, we propose the development of a support system that suggests optimal practice methods for beginners using fast Fourier transform of audio data and data analysis.

Through this, we aim to alleviate the frustrating experience of those who want to start playing musical instruments.

1.2 研究背景

ギターはソロでの演奏はもちろん、バンドの演奏には欠かせない楽器であり、人気で身近な楽器である[1]。バンド活動が好きな人ならば一度は目にしたこと、手に触れたことや憧れたことがあると思われる。

しかしギターに限らず楽器というものは一人では始めるには非常に難しいモノであり、何かからすればよいのかわからなく、個人で上達を目指すことは難しい。部活動のようにある程度の技術を有した人たちがいる環境に巡り合うことが出来た人であれば良いが、そうでない人が大半であると思われる。実際にギターを始めたばかりの9割は一年以内に挫折し、楽器を触ることがなくなっているが有名なギターメーカーである fender での研究で明らかになっている。

その要因として挙げられるのが、「上達できる実感がわからない」、「どのような練習をおこなうべきかわからない」等である。また練習用アプリケーションもいくつか配信されているが、それらは広告が多かったり、有料であったりなどの問題がある。

1.3 研究目的

本研究では、ギターの演奏音に対して高速フーリエ変換などの信号処理やデータ解析を行うことで初心者に適した練習方法を提案するシステムを開発する。楽器の練習は個人で行うことは不可能ではないが、サポートがあると上達がより早いので、本研究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_6

OS : Windows10

CPU : Corei5-8250U

実装 RAM : 8GB

使用言語 : Python3. 11. 2

開発環境 : Anaconda, Jupyter Notebook

使用ライブラリ : Numpy, Sounddevice, scipy, tkinter, matplotlib

2. 関連技術

2.1 高速フーリエ変換

本研究では、高速フーリエ変換 (Fast Fourier Transform: FFT) を用いて周波数解析を行った。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]。

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

e は自然対数の底、 j は虚数単位、 π 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 のオーダーで演算を繰り返し行う必要があるため計算時間が長くなる問題がある。

一方 FFT では演算回数 $N * \log_2 N$ のオーダーまで減らすことができる。音声処理ではデータ数が膨大なため FFT が用いられることが多い。FFT の定義は下記の通りである [3]。

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

ここで $W_M = e^{-j\frac{2\pi}{M}}$ は回転子と呼ばれ単位円を M 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。

さらに 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]。

本研究で使用了音声ファイルには1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 pyaudio

pyaudioはリアルタイムで音声の録音や再生等のオーディオ処理を行うためのライブラリである。今回の研究では録音処理のためにpyaudio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。Python に初期から搭載されているライブラリであり、汎用性が高いことから、今回の研究では音の表示（図 1）などの描画処理を行うために Tkinter を使用した。



図 1 音の表示

2.5 matplotlib

Matplotlib は Python におけるグラフ描画のライブラリである。今回の研究では Numpy ライブラリで周波数解析した音声データを可視化するために matplotlib を用いた。

3. 研究結果

3.1 実験内容

始めにギターの一弦を鳴らして周波数表示出来るかどうかの実験を行った。まず図 2 のプログラムを実行し、pyaudio を使用してギター演奏を録音する。その音声データに対して numpy を使って高速フーリエ変換し、matplotlib を使って周波数表示した。

次に録音した音声データの音程を周波数解析によって判別する実験を行った。ギターの正しい音程は一弦から六弦まで順に E、B、G、D、A、E となっているので、これを踏まえて一弦から順に音声データを録音して周波数解析を行った。

```
def Fourier_Transform(record_data):  
    # 音声データを読み込む  
    data = record_data.read(1024)  
    # 読み込んだデータを整数型のnumpy配列に変換  
    audio_data = np.frombuffer(data, dtype='int16')  
    # 取得したデータをフーリエ変換をする(複素配列)  
    F = np.fft.fft(audio_data)  
    # 振幅を正規化  
    F = F / (SAMP_RATE / 2)  
    # ハニング窓関数の適用  
    window = scipy.signal.hann(SAMP_RATE)  
    F = F * (SAMP_RATE / np.sum(window))  
    # 振幅の計算  
    amp = np.abs(F)  
    # 周波数の計算  
    freq = np.fft.fftfreq(1024, d=(1/SAMP_RATE))  
    # パワースペクトルを求める  
    amp = pow(amp, 2)  
    # データを表示  
    Display_Data(freq[:1024//2], amp[:1024//2])
```

図 2

3.2 実験結果と考察

一弦の開放弦の周波数表示結果を図3に示す。ここで一弦の開放弦の周波数は329.62であるため、正しく音程が周波数表示できていることが分かる。

次に録音した音声データに対する音程判別を行った。判別の例を図4に示す。図4は上から順に一弦(E)、二弦(B)、三弦(G)、四弦(A)、五弦(E)、六弦(B)の適合度を表し、この例では三弦(G)として音程が判別されている。一弦、二弦、三弦は正しい音程を判別出来たが、四弦、五弦、六弦は正しい音程とは異なって判別されるという結果となった。

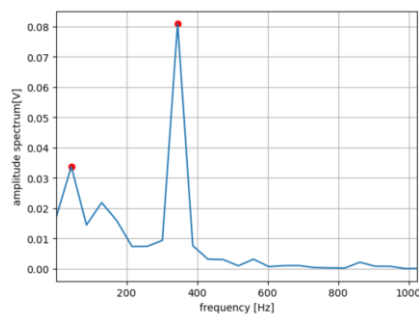


図3 一弦の解析結果

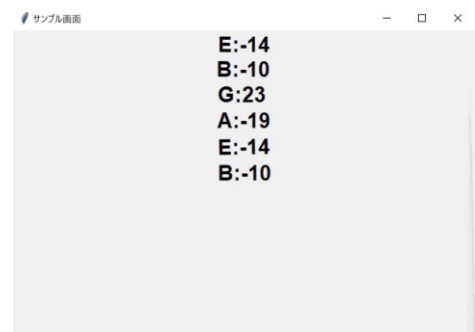


図4 音の表示

4. 今後の展望

本研究では、ギターの演奏音に対して高速フーリエ変換などの信号処理やデータ解析を行うことで初心者に適した練習方法を提案するシステムを開発した。ただし四弦、五弦、六弦の解析が上手くいかなかったことから、プログラムのアルゴリズムの再検討を行う必要がある。また今回はユーザーに対し最適な練習方法を提案するプログラムの実装まで至らなかった。

今後の課題は音声の正確な判定をさせること、そしてユーザーに対し最適な練習方法を提案するプログラムを完成させることである。

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

[4] Numpy

<https://helve-blog.com/posts/python/numpy-fast-fourier-transform/>