修正内容

スライドの各ページに番号を振っていなかったものを修正

情報工学実験B
（メディア処理）

音声処理実験2020 ミニレポート

学生番号：09430509

氏名：今田将也

提出日：2020年11月19日

締切日：2020年11月19日

概要

じゃんけんの三手を認識できるような，単語音声認識器を構築することを目指す．音声認識器を作るために音声（メディア情報）をディジタルデータとして処理する方法，ディジタルデータから所望の情報を表現しうる特徴量ベクトル（行列）に変換する手法，また，特徴量ベクトル（行列）を用いた識別器を構成するという操作を，それぞれ理解する．そして，簡単な音声の特徴量を用いて，基本的な識別器を構築することで，単語音声認識器の可能性を検討する．

まず，MATLAB　Octaveでの音の録音，再生，保存，読み出しについてデモ音を用いてその音声波形を表示し，音をデータとして扱う方法について学ぶ．

そして，得られた音のデータを離散フーリエ変換により，音声の周波数的な特徴をパワースペクトルというもので目視するための手法を正弦波を用いて出力する．

次に，スペクトログラムというものを用いて，時間経過による周波数の特徴の変化についても目視できるようにする．

概要

さらに，グーチョキパーという自分の音声を録音し，時間波形やパワースペクトルについて観察したあと，参照パターンと入力音声のスペクトル距離の比較に基づく単語音声認識について考察する．その後，自身で自動音声認識器を実装し，声を正しく認識できるか見てみる．

最後に最適経路探索問題を応用して，Dijkstra’s algorithmでスペクトログラムによる音声間の距離計算を実装し音声認識器の精度向上も行ってみる．

問題1

MATLAB/Octave上で音や音声を扱うための基礎知識を，簡潔に説明せよ．

問題1の小問題

1. MATLAB/Octaveにおける音のI/O処理について，以下の4つの観点に整理して簡潔に説明せよ
   1. 音の録音（ToDo: 第4回(4A節)で実施）
   2. 音の再生
   3. 音の音声ファイルへの保存
   4. 音の音声ファイルからの読み出し
2. デモ音（あるいは，自身の録音音声）を利用して，その音声波形を時間波形として表示せよ．なお，以下3つの情報を明記すること．
   1. 音のサンプリング周波数（単位は Hz か kHz）
   2. 音の長さ（単位は s か ms　※秒単位かミリ秒単位のどちらかということ）
   3. 発話内容（例えば，「漢字と読み仮名」あるいは「ひらがな」などのいずれかで書く．）

1-1. 音の録音・再生

音の録音(4Aの内容）

1. Audiorecorder関数で録音用のオブジェクトを作成
2. Recordblocking関数を用いて録音
3. getaudiodata関数を使い，音データを取得

音の再生

Sound関数を用いることで可能.

1. 第1引数：再生するデータの変数
2. 第2引数：再生するときのサンプリングレート

```
sound(y, Fs)
```

1-1.音声ファイルへの保存・読み出し

音声ファイルへの保存

Audiowrite関数を利用すると，音声データを音声ファイルに書き出せる.

- 第1引数：ファイル名
- 第2引数：指定した変数のデータ
- 第3引数：サンプリング周波数をHz単位で指定

音声ファイルの読み出し

```
[y, Fs] = audioread('example_1.wav');
```

MATLAB/Octaveでは，コマンド一つで実行できる.

- audioread第１引数：読み込みたい音声ファイル名
- 1つ目の戻り値yには，第1引数で指定したファイル内の音声データが，ベクトルとして読み込まれる.
- 2つ目の戻り値Fsには，同ファイルが保存しているサンプリング周波数が，スカラーとして読み込まれる.

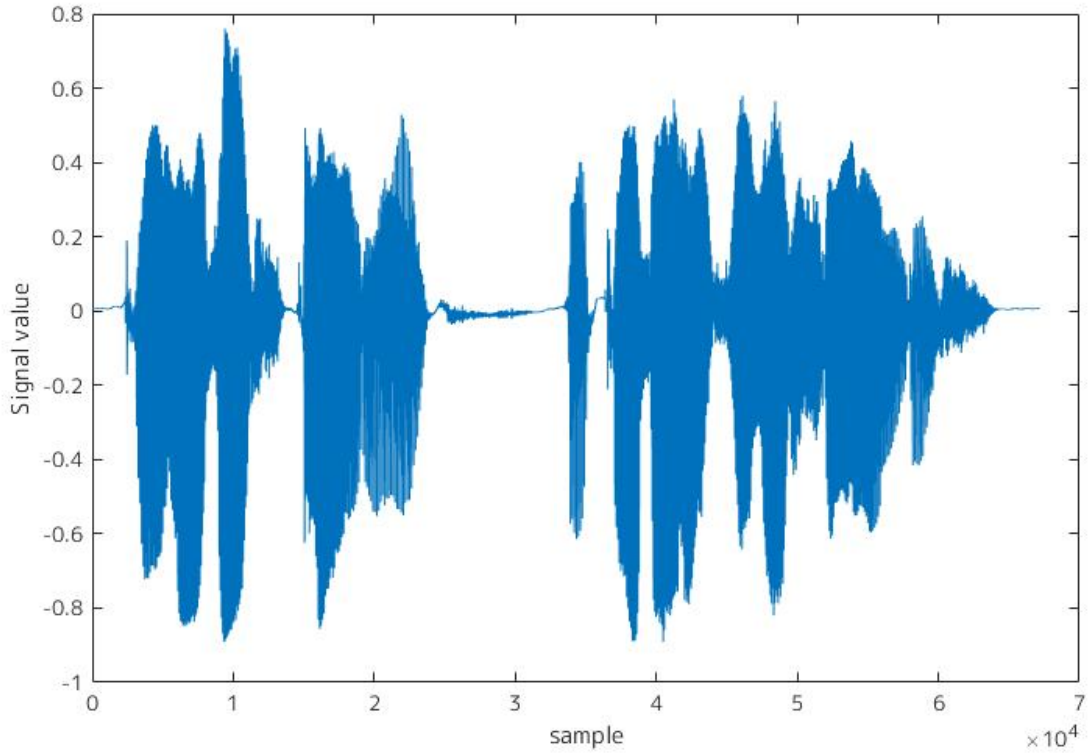
1-2.デモ音の音声波形を時間波形として表示

サンプリング周波数
- 24000Hz

音の長さ
- 2.7994秒

発話内容
- このばすていはおかだいひがしもんです

サンプル数
- 67185



問題1のまとめ

問題

MATLAB/Octave上で音や音声を扱うための基礎知識を，簡潔に説明せよ.

まとめ

音声合成のようなサンプル音声を用いて，音の再生をした

実際に録音を行い，ファイルへの保存，読み出しまでの一連の流れを示した.

問題2

純音（正弦波）のパワースペクトルを観察し，パワースペクトルから得られる情報と聴感的な印象の関係について考察せよ．

問題2の小問題

1. MATLABのソースコードを示しながら，純音の生成から，DFT (FFT)を用いたパワースペクトルの表示までの手順を説明しなさい.
2. 純音の時間波形とパワースペクトルを図示しなさい．作成に関わるパラメータを明記すること.
3. MATLABのソースコードを示しながら，DFT (FFT)を活用して，実世界尺度を考慮したパワースペクトルを表示するための手順を説明しなさい.
4. 波数を変えた純音のパワースペクトルをいくつか並べて図示して，以下の観点から考察しなさい.
    1. 正弦波信号の周波数と図示された正弦波のパワースペクトルの形状にはどのような関係があるか？
    2. 正弦波信号の周波数と聴感的な印象（「高い音」や「低い音」）にはどのような関係があるか？

2-1.純音の生成から，DFT (FFT)を用いたパワースペクトルの表示までの手順(1/2)

正弦波信号の生成

- アナログ信号の信号長[sec]をデジタル信号の点数に換算する．ここでは9600個の点
- サンプリング周波数 16,000 Hz で，0.6秒の音を表現するためには，9,600個の数字列が必要

sin関数を使い，$x(t)=A\sin(2\pi ft)$を満たす正弦波を生成

- 1:9600で1から9600までの数字列（9600個の点）を作る
- 時刻が0から始まることを考慮し0から9599までの数字列とする
- サンプリング周波数 16,000 Hzでは，点間は秒間にして1/16000秒に相当するため，16000で割る
- sin関数の最後のカッコの後に，シングルクオートをつけて転置し，音データと縦方向にベクトルの向きを揃える

```
signal_length_sec = 0.6;
sampling_rate = 16000;

signal_length_pt = signal_length_sec * sampling_rate
t =  ((1:signal_length_pt) - 1) / sampling_rate;

A = 0.2;    % Amplitude
f = 10;     % Frequency [Hz]
t = ((1:9600) - 1) / 16000;
x = A * sin(2 * pi * f * t)';
```

2-1.純音の生成から，DFT (FFT)を用いたパワースペクトルの表示までの手順(2/2)

時間波形として出力

- Plot関数の第1引数にt，第2引数にxを入れる．

```
A = 0.2;    % Amplitude
f = 10;     % Frequency [Hz]
t = ((1:9600) - 1) / 16000;
x = A * sin(2 * pi * f * t)';

plot(t, x)
```

DFTでのパワースペクトル出力

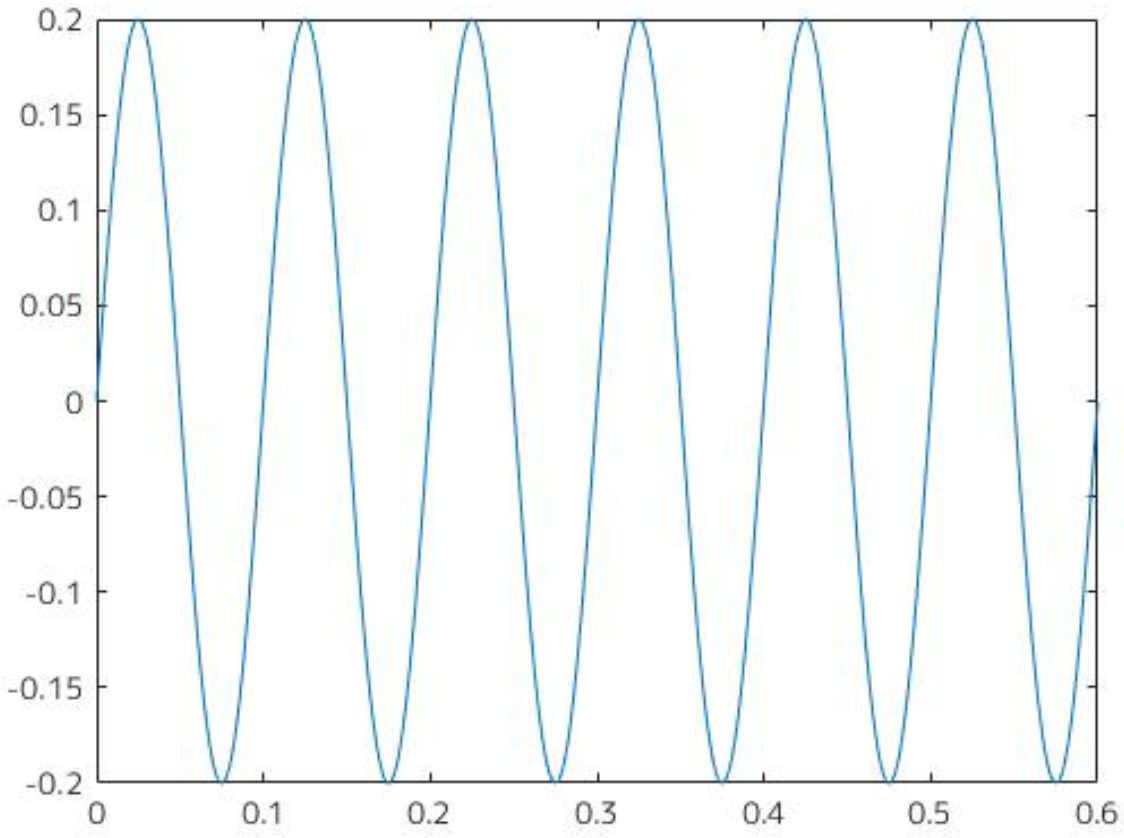
- MATLABに標準搭載されているFFT関数を使うことでDFTを実装可能
- 信号をfft関数に適用し，その配列数(length)で割る
- 結果は複素数のベクトルであるため，絶対値の二乗を行う
- 上記の結果をplot関数で表示する

```
%DFT
X = fft(x) / length(x);
XPow = abs(X) .^ 2;
plot(XPow, 'o-')
xlim([1 10]) %plot array index from 1 to 10
```

2-2.純音の時間波形と設定パラメータ

パラメータ設定

- 秒数　　:0.6sec
- サンプリングレート:16000Hz
- 振幅A　　:2
- 振動数　　:10hz

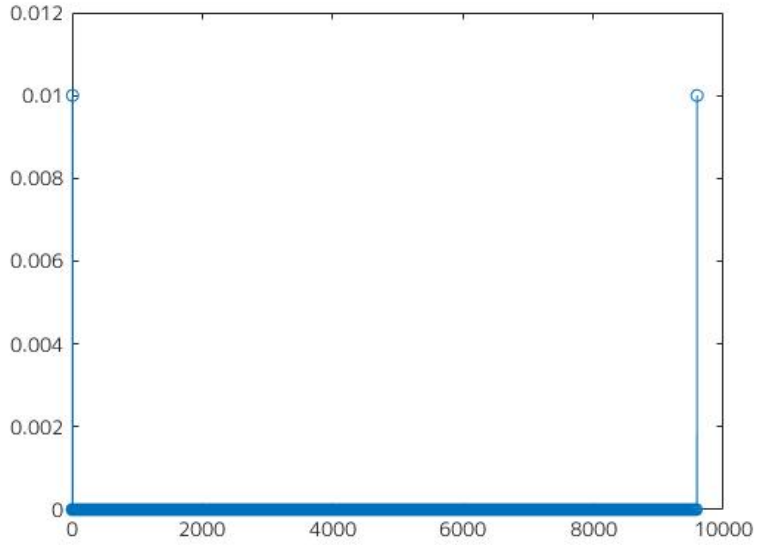


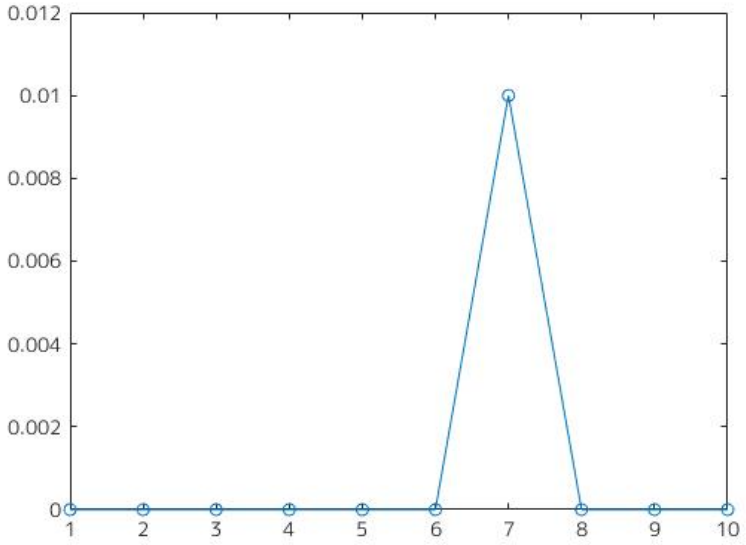
純音の時間波形

2-2.パワースペクトルの図示

下の左側の画像は0．6秒分のサンプル数9600個分すべてのパワースペクトルを図示したものである．始まりと終わりに強い値を持つが，詳細がわからない．

そこで，先頭の10個分を見てみると7点目におおきな値を得られていることがわかった．





2-3.実世界尺度を考慮したパワースペクトルを表示するための手順

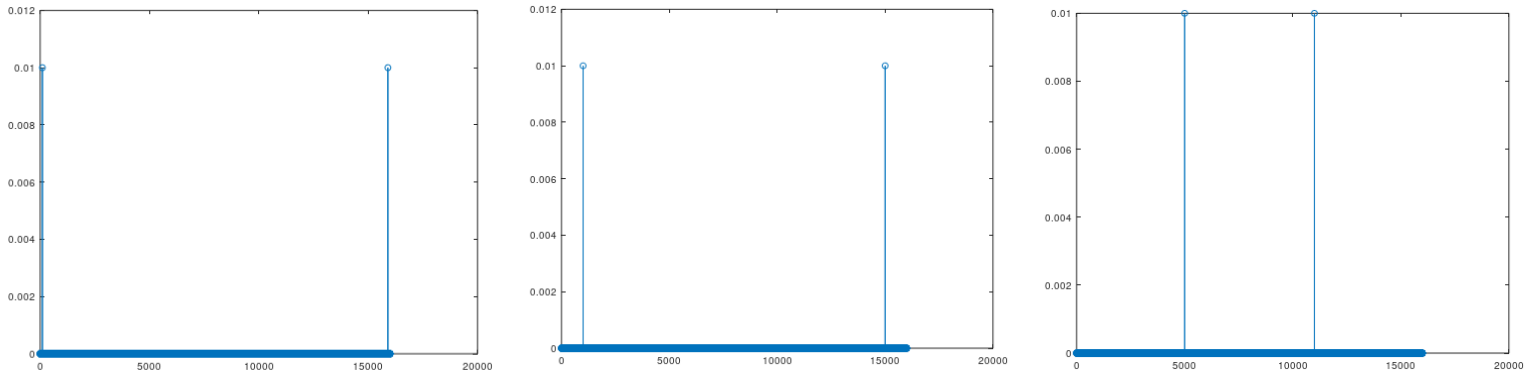
1. 周波数軸（X軸）の換算
   1. 周波数番号kに対応した周波数を持つベクトルf_kを用意する.
   2. MATLABの配列は1から始まるため，k=1+k’として用意する.
2. 片側化
   1. サンプリング定理に従えば，サンプリング周波数A Hzにおいては，2/A Hzより大きい周波数成分は存在しない
3. 対数化
   1. パワースペクトルは縦軸の数値を対数スケールに換算してdB（デシベル）という単位で示す
   2. XPow_dB = 10 * log10(XPow)という変換式を用いる

```
f_k = linspace(0, sampling_rate, signal_length_pt+1);  % 16000を9600+1分割した配列
f_k(end) = [];% 最後消す
XPow((2+signal_length_pt/2):end) = [];% 折り返した後半を消して片側のみ
f_k = linspace(0, sampling_rate/2, 1+signal_length_pt/2);%等分割
plot(f_k, XPow);
xlabel('Frequency [Hz]');
xlim([1 20]) %plot array index from 1 to 10
```

2-4.正弦波信号の周波数と図示された正弦波のパワースペクトルの形状

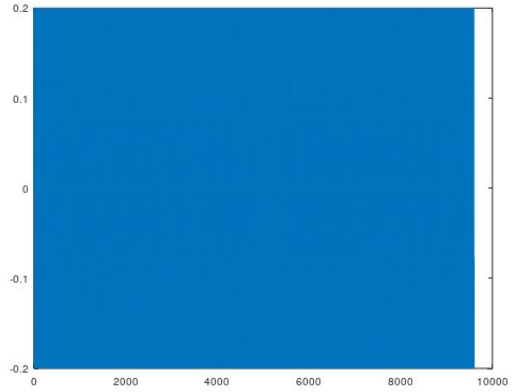
下図に示したのは左から100hz,1000hz,5000hzの正弦波のパワースペクトルの図である.

- サンプリングレートは16000hz
- 周波数が高くなるにつれ，パワースペクトルの特徴点の位置はナイキスト周波数である8000hzに集まっていくように見える



2-4.正弦波信号の周波数と聴感的な印象

1. 実験に用いた16000hzのサンプリングレートのナイキスト周波数である8000hzでは音が聞こえなくなった
2. 周波数が高いと耳を劈くようなきつい音がなる
3. 周波数が低いとボーッというようなこもった感じの音がなる
4. 周波数が低いとはいえ，１秒間に200回もなっていると波形は見ずらい
5. 周波数1000hzとかになると青いただの四角のグラフに見えた



問題2のまとめ

問題

純音（正弦波）のパワースペクトルを観察し，パワースペクトルから得られる情報と聴感的な印象の関係について考察せよ．

まとめ

- ソースコードを示しながら，純音の生成から，DFT (FFT)を用いたパワースペクトルの表示までの手順を説明した．関数一つでパワースペクトルは表示できた
- パワースペクトルの強い点を示しているところを拡大して図示した
- MATLABのソースコードを示しながら，DFT (FFT)を活用して，実世界尺度を考慮したパワースペクトルを表示するための手順を説明できた
- 周波数を変えた純音のパワースペクトルをいくつか並べて図示して，周波数の高い時，低いときについて考察した

問題3

適当な信号のスペクトログラムを表示して，その算出過程や結果について考察せよ.

問題3の小問題

1. 本問題の設定における番号（データ番号および周波数番号）と実尺度（時刻および周波数）の換算式を書きなさい．
    1. 換算表，あるいは，MATLABのコードとして書いても良い．
    2. 時刻を測る実尺度は s 単位（秒単位），または，ms 単位（ミリ秒単位）とすること．
    3. 周波数を測る実尺度は Hz 単位，または，kHz 単位とすること．（※角周波数ではない）
2. 適当な正弦波信号を作成し，その信号のスペクトログラムを表示し，考察しなさい．

3-1.変換式

横軸が時刻tに応じて変わる信号f(t)を，DFTを用いてパワースペクトルにすると，横軸は周波数番号というものに変わる

そのときの関係式を以下に示す

$\Delta n = 1[pt] \leftrightarrow \Delta t = \frac{1}{Fs}[sec]$　１個あたりのデータ番号はサンプリング周波数の逆数の大きさになっている

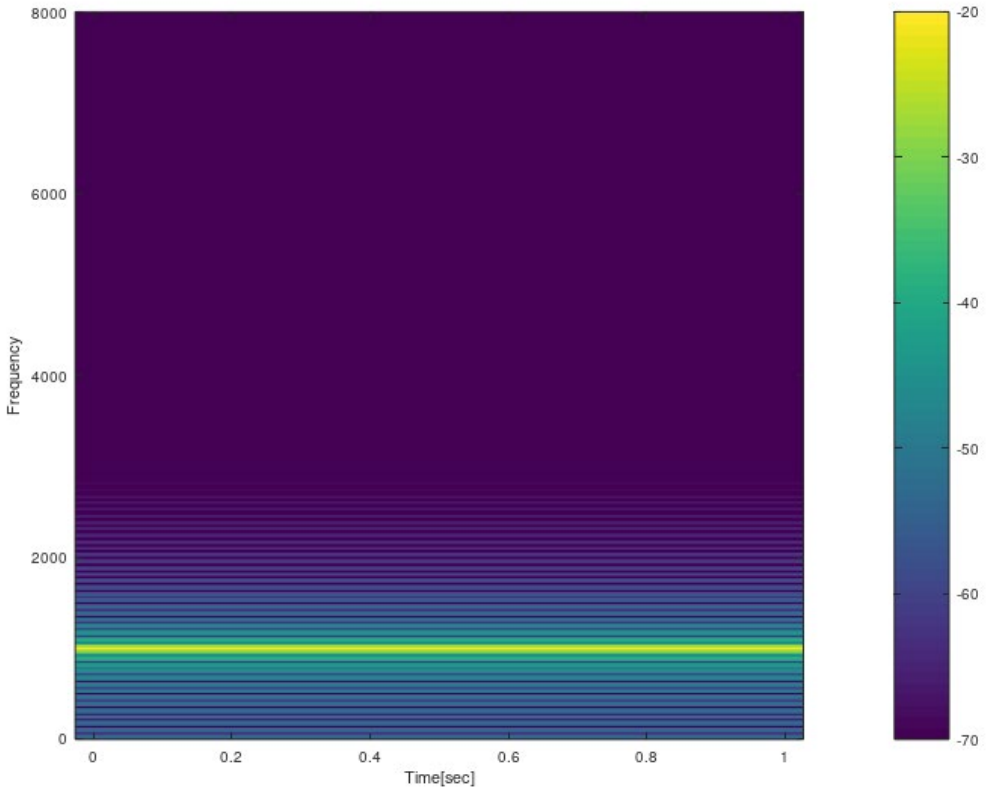
$\Delta k = 1[pt] \leftrightarrow \Delta f = \frac{Fs}{N}[Hz]$　周波数番号の１個あたりはサンプリング周波数をデータの長さの個数で割ったもの

```
signal_length_pt = signal_length_sec * sampling_rate;
t =   ((1:signal_length_pt) - 1) / sampling_rate;
```

```
X = fft(frame_x, fft_len) / fft_len; % 各フレームの信号に対するFFTの結果として，1024x20の複素数の行列X
Pow_X = abs(X) .^ 2; % 20個のFFTの結果を，20個のパワースペクトルに変換
Pow_X((2+fft_len/2):end, :) = [];  % 後ろから(半分-1)個を消す＝前から(半分+1)個を生かす 片側化
Pow_X_dB = 10 * log10(Pow_X);% 対数化
```

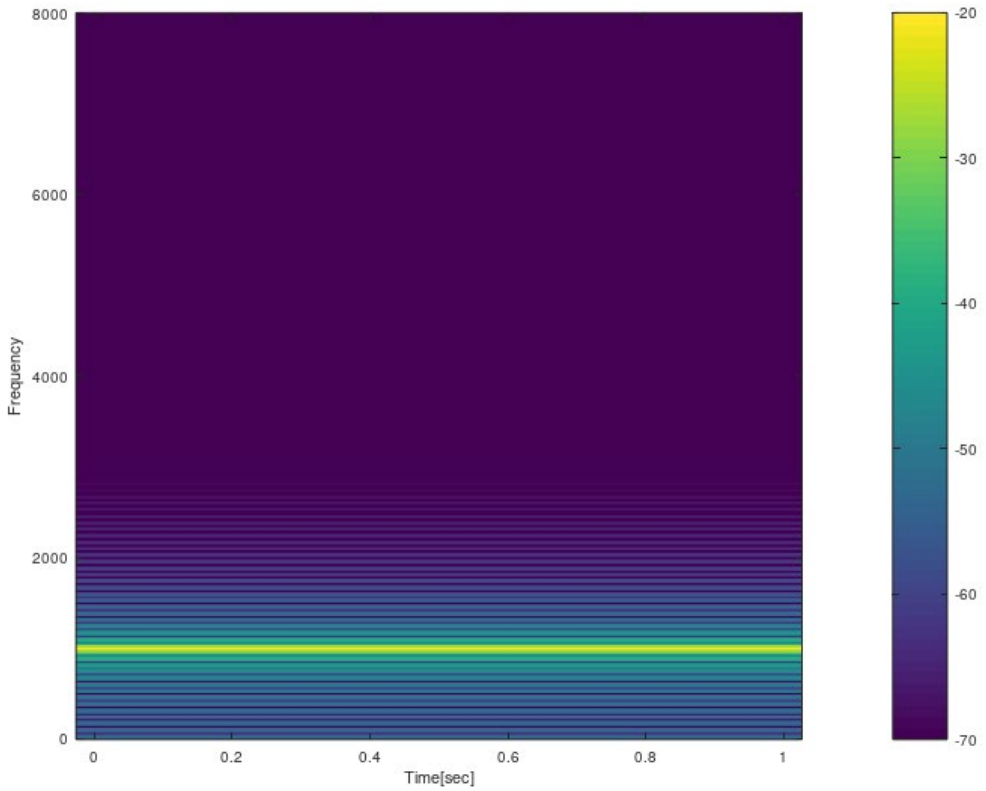
3-2.用いた尺度とスペクトログラム

- ✓サンプリング周波数は 16 kHz (16,000 Hz)
- ✓信号長1秒
- ✓周波数1,000 Hzの純音
- ✓信号長が16000点
- ✓800点ずつに分割
- ✓フレーム数は20個
- ✓秒数は1秒



3-2. スペクトログラムの考察

- ✓1000hzの正弦波を生成し，スペクトログラムを表示した.
- ✓純音であるため，ある一定の値のPowerが強く反応していることが見て取れる.
- ✓おそらく，1000hzの周波数番号が黄緑に近い色の部分に格納されている
- ✓時間経過しても周波数のデータは損なわれない



問題3のまとめ

問題

適当な信号のスペクトログラムを表示して，その算出過程や結果について考察せよ．

まとめ

- 正弦波の実データ（時間，周波数）を講義の設定された番号への変換を行うための式を示し，変換の算出過程のコードも示した
- スペクトログラムを示し，時間と周波数の強さの特徴について考察した

問題4

じゃんけんの3手（「グー」「チョキ」「パー」）の単語発声を音声ファイルとして録音し，今後の音声認識の実験で利用できるよう準備せよ.

- いずれの音声ファイルも，サンプリング周波数 16 kHz，信号長 600 ms とすること.
- 各パターンにつき，少なくとも2つは収録すること.

問題4の小問題

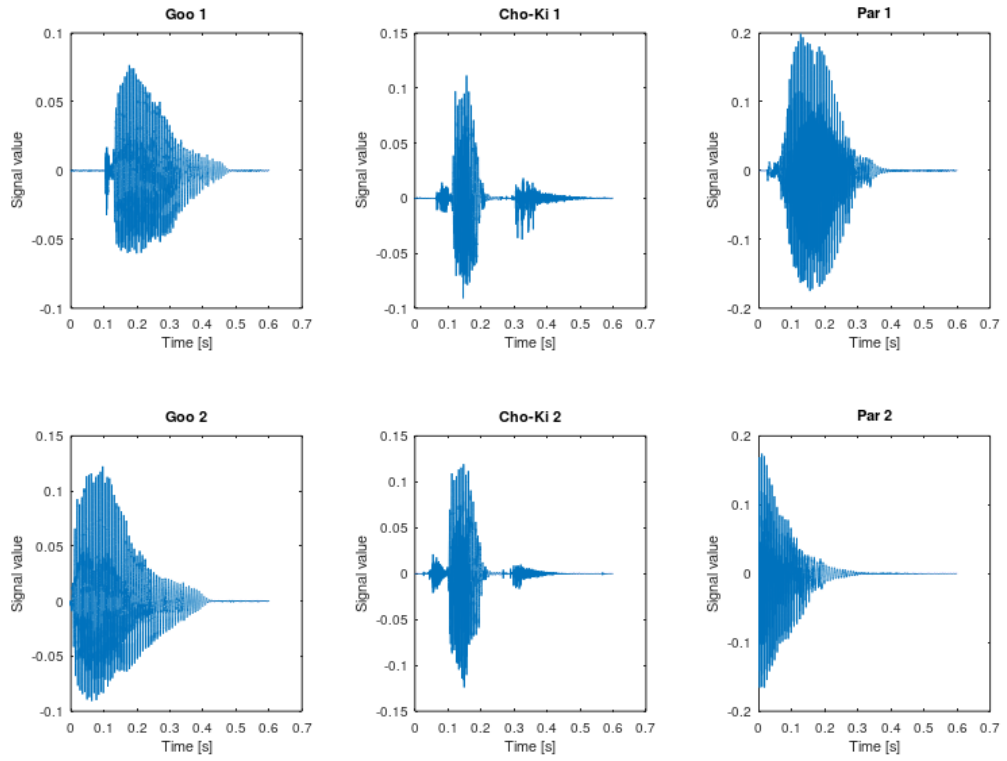
1. 音声収録の際に，注意したことや心がけたことがあれば，説明しなさい.
2. 記録した6つの音声（グー，チョキ，パーそれぞれ2つ）の時間波形を図示しなさい.
3. 記録した6つの音声（グー，チョキ，パーそれぞれ2つ）の対数パワースペクトルを図示しなさい.
4. 時間波形や対数パワースペクトル等の「観察結果」や「考察」を示しなさい.
5. 記録した6つの音声（グー，チョキ，パーそれぞれ2つ）のスペクトログラムを図示しなさい.

4-1.音声収録の際に，注意したことや心がけたことがあれば，説明しなさい．

1. 該当コードに到達するとすぐに録音が開始されるので，はじまるまえにpause(1)を挟んで録音待機をさせた
2. ０．６秒に収まるように発話タイミングを調整した
3. 音の大きさが振りきれて俗に言う音割れを起こさないように録音する音量を調整した

4-2.6つの音声の時間波形

MATLAB のsubplot関数を使い，一度に6つの波形を表示させた



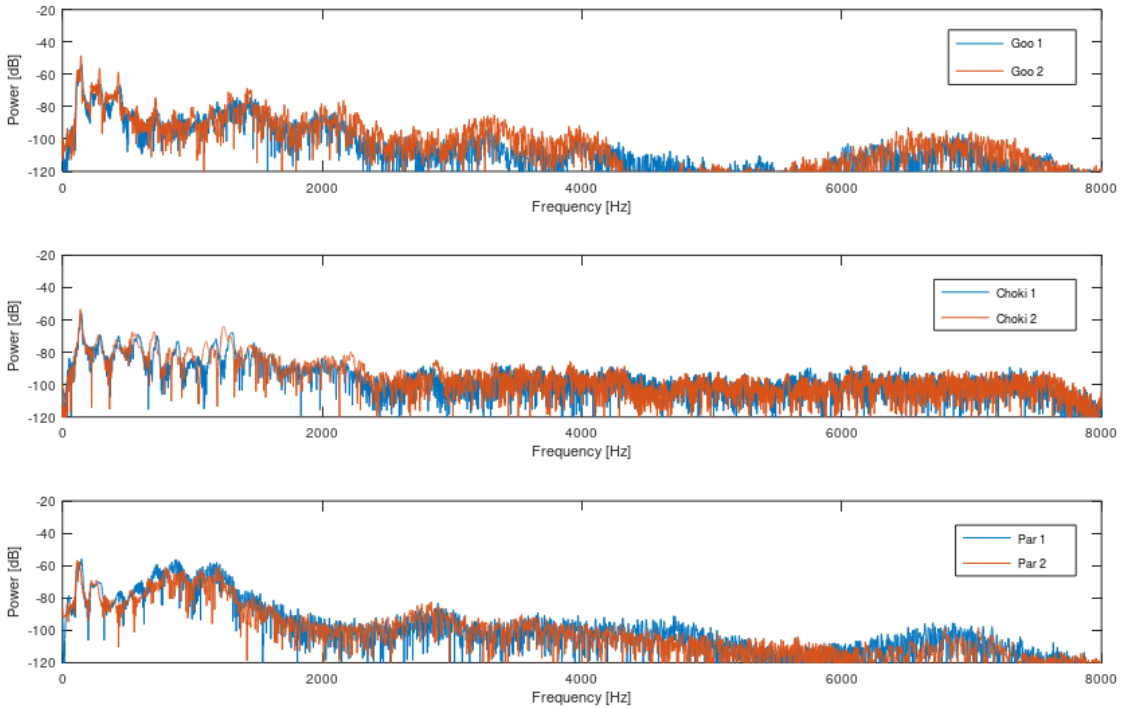
4-3.記録した6つの音声（グー，チョキ，パーそれぞれ2つ）の対数パワースペクトル

右図に示す．

✓一番上がグーという発音2つの音の対数パワースペクトル

✓真ん中がチョキという発音2つの音の対数パワースペクトル

✓一番下がパーという発音2つの音の対数パワースペクトル



4-4. 「観察結果」や「考察」

観察結果

- グーとパーを見てみると，殆どの周波数の部分で2つの音声のパワースペクトルに一致している
- チョキはずっと反応が起きている
- 2000hz以下の部分でのパワーが大きい

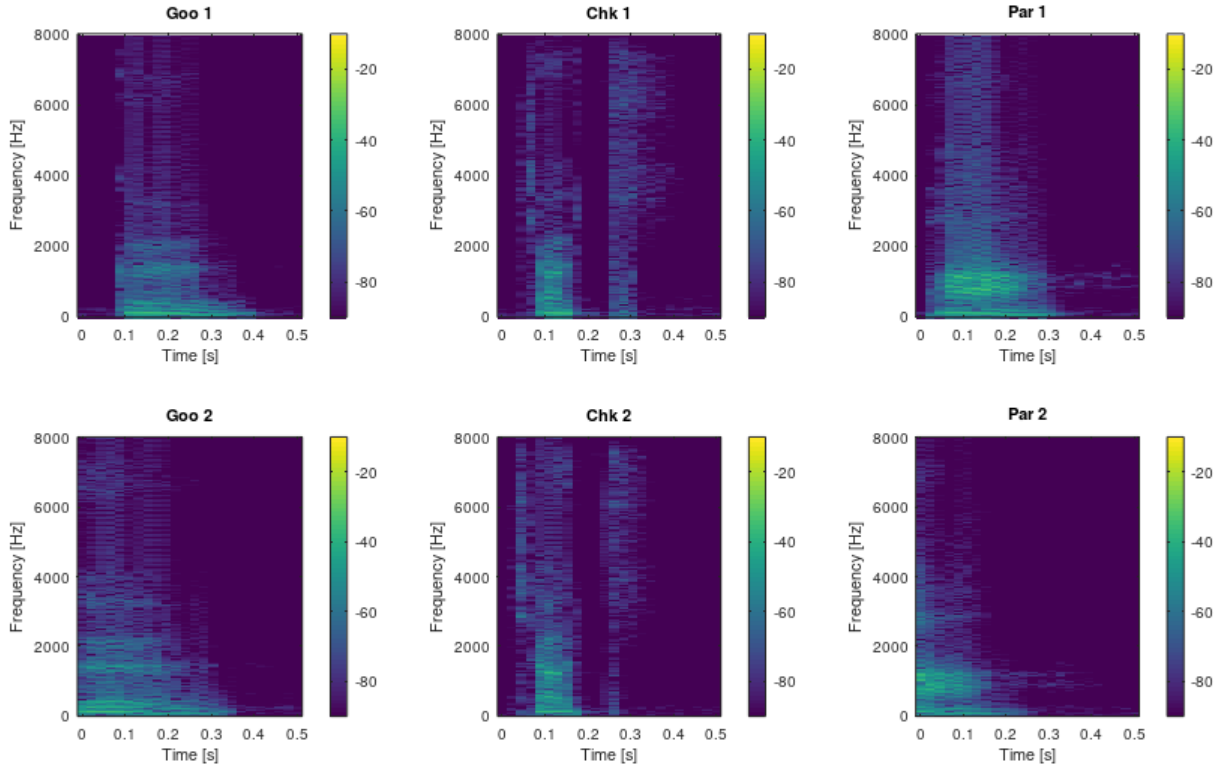
考察

- 発話者の声が低いため，低周波数域のパワーが強くなっていると思う
- チョキがずれている理由は恐らく発音の問題だろう．もしくは同じ周波数域でのノイズが混在している可能性がある

4-5.記録した6つの音声(グー，チョキ，パーそれぞれ2つ)のスペクトログラムを図示

グー，チョキ，パー，それぞれ2個ずつのスペクトログラムを表示した.

それぞれ縦はHzで，横は時間である.



問題4のまとめ

問題

じゃんけんの3手（「グー」「チョキ」「パー」）の単語発声を音声ファイルとして録音し，今後の音声認識の実験で利用できるよう準備せよ.

まとめ

- MATLABを使いじゃんけんの3手それぞれ2つずつのサンプルを録音し，保存した
- それぞれの時間波形を示した
- 記録した6つの音声（グー，チョキ，パーそれぞれ2つ）の対数パワースペクトルをしめし，それぞれの観察結果および考察について述べた
- スペクトログラムを図示，スペクトログラムの表示手順を確認した.

問題 5

参照パターンと入力音声のスペクトル距離の比較に基づく単語音声認識について，考察せよ.

問題5の小問題

1. どのようなアプローチで音声認識を実現するのか，簡単に説明しなさい．
2. 問題4で収録した3+3=6つの音声の（対数）パワースペクトルを用いて，3x3=9種類の組み合わせで距離を計算し，その結果を表としてまとめなさい．

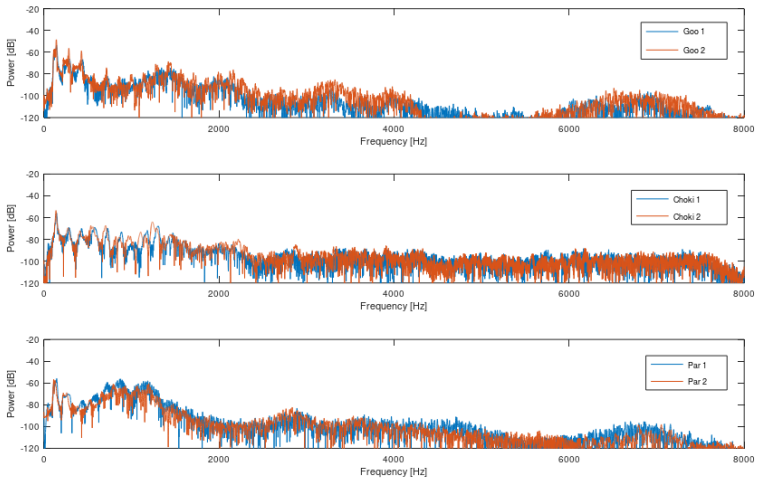
5-1.どのようなアプローチで音声認識を実現するのか，簡単に説明しなさい．

元となる参照データと，識別したい入力データがどれだけ似ているか2音声の距離計算にスペクトル距離を用いる

ベクトル$x = [x_1\ x_2\ \cdots x_N\ ]^T$とベクトル$\mathrm{y} = [y_1\ y_2\ \cdots y_N\ ]^T$の距離$D(x, y)$を以下のように計算する

$$D(x, y) = \sqrt{\sum_{n=1}^{N} (x_n\ - y_n\ )^2}$$

5-2.問題4で収録した6つの音声のパワースペクトルを用いた，9種類の距離の結果表



同音同士の対数パワースペクトル（再掲）

	Goo2	Chk2	Par2
Goo1	888.69	1365.46	1148.65
Chk1	1313.32	737.83	1146.56
Par1	1263.56	1110.19	767.11

5-2.問題4で収録した6つの音声のパワースペクトルを用いた，9種類の距離の結果表

【表からわかること】

グーとグー同士などの同じ音同士の距離は4桁まで大きくならないぐらいの距離になった

異なる音同士では，距離が大きくなっていて，それだけ似ていないということがわかる

同音同士つまり対角要素の距離の大きさが最短であり，距離だけを見た音声比較による認識でも，自分の声なら上手くいきそうである.

問題5のまとめ

問題

じゃんけんの3手（「グー」「チョキ」「パー」）の単語発声を音声ファイルとして録音し，今後の音声認識の実験で利用できるよう準備せよ.

まとめ

どのように音声認識をするのかのアプローチについて式を提示しつつ，距離について見ることについて説明した

9種類の音の組み合わせについて距離の大きさを表にした

問題6

自身で実装した自動音声認識器を用いて，音声インタフェースを作成し，考察せよ．

- 単語認識ができればよい．
- マイクから音声を入力して，結果を画面に出力するまでの一連の流れを，MATLAB/Octaveのスクリプト（あるいは，関数）として実行できればよい．

問題6の小問題

1. MATLAB/Octave等のソースコードを用いて，実装と動作例を示しなさい．
2. 音声認識器の”良さ”を，何らかの客観的な数値を示すことで，評価しなさい．

6-1.実装手順

以下の流れでコマンド群を実行していく

1. 初期化
2. 参照パターン用の音声ファイルを読み込む
3. 読み込んだ音声ファイルのデータを対数パワースペクトルに変換する
4. 入力パターンとして音声を録音する
5. 録音した音声を対数パワースペクトルに変換する
6. 入力パターンと参照パターンそれぞれの距離を計算して，最短の距離を持つ参照パターンを見つける
7. 参照パターンに相当する文字列を表示する

6-1.実装(1/2)

%0 初期化
- すべての変数をクリアし，録音のための変数，結果表示用文字列の配列を用意する

```
%% 0. Initialization
clear;
rec = audiorecorder(16000, 16, 1);  % 16000 Hz, 16 bit, 1 channel
fft_len = 16384;
result_string_table = {'Goo', 'Chk', 'Par'};
```

%1 参照パターン用の音声ファイルを読み込む
- スクリプト化してあるため簡略的に読める

%2 読み込んだ音声ファイルのデータを対数パワースペクトルに変換
- スクリプト化してあるため簡略的に読める

```
% 1. Load waveform from WAV files
JAN_LOAD_WAVEFILES;

%% 2. Convert them to power spectrums
Jan_calc_powerspecs;
```

%3 入力パターンとして音声を録音
- マイクから入力された，0.6秒の音声は，変数xに格納

```
%% 3. Record an input waveform
disp('3'); pause(1); disp('2'); pause(1); disp('1'); pause(1); disp('Go!'); % count down
recordblocking(rec, 0.6);
x = getaudiodata(rec);
```

6-1.実装(2/2)

%4　録音した音声を対数パワースペクトルに変換

- 以前作成したスクリプトを利用

```
%% 4. Convert the input waveform to power spectrum
[PowX_dB, PowX] = calc_powerspec(x, fft_len);
```

%5-1　距離計算

```
%% 5-1. Calculate distance between the input pattern and every reference patterns
Dist(1) = sqrt( sum( (PowX_dB(:,1) - Jan_Goo_PowX_dB(:,1) ) .^2 ) );
Dist(2) = sqrt( sum( (PowX_dB(:,1) - Jan_Chk_PowX_dB(:,1) ) .^2 ) );
Dist(3) = sqrt( sum( (PowX_dB(:,1) - Jan_Par_PowX_dB(:,1) ) .^2 ) );
disp(Dist) % for debug
```

%5-2　最短距離の参照パターンの探索

- 「配列の中で最小値を持つ配列のインデックスを得る」ことで実現

```
%% 5-2. Select the pattern that has a minimum distance
[~, idx] = min(Dist);
```

%6　参照パターンに相当する文字列を表示

```
%% 6. Display the result stiring!
disp(result_string_table{idx});
```

6-2.動作例

- ASR.mというファイルで作成をしたため，ASRというスクリプトで実行する.
- 3 2 1 Go! という表示のあとに録音をする
- 録音されたものが参照パターンとどれだけ似ているか距離を表示する
- そのなかで最も小さいものを認識結果として表示
- 認識率は自分の声なので100%だった

```
>> ASR

3
2
1
Go!
    937.65    1313.69    1235.70
Goo


>> ASR
3
2
1
Go!
   1533.20     829.29    1252.95
Chk
>> ASR
3
2
1
Go!
   1191.90    1116.14     775.40
Par
```

6-3.客観的に良さを示す(1/2)

- ✓友人2人に協力してもらって，グー・チョキ・パーそれぞれの音データをもらい，自分以外のデータでも認識できるのか試す．
- ✓2x6ファイル分のループを処理するコードを書いて行った．
- ✓グー1・2，チョキ1・2，パー1・2の順で読み込んだ

それぞれの声を正しく認識できていればこの識別器は良いと言えるだろう

```
for k = 1:length(pararent_files)
    for i = 1:length(filenames)

      filename = strcat(strcat(pararent_files(k)),filenames(i));
      x = audioread(char(filename));

      %% 4. Convert the input waveform to power spectrum
      [PowX_dB, PowX] = calc_powerspec(x, fft_len);

      %% 5-1. Calculate distance between the input pattern and every reference patterns
      Dist(1) = sqrt( sum( (PowX_dB(:,1) - Jan_Goo_PowX_dB(:,1) ) .^2 ) );
      Dist(2) = sqrt( sum( (PowX_dB(:,1) - Jan_Chk_PowX_dB(:,1) ) .^2 ) );
      Dist(3) = sqrt( sum( (PowX_dB(:,1) - Jan_Par_PowX_dB(:,1) ) .^2 ) );
      disp(Dist); % for debug

      %% 5-2. Select the pattern that has a minimum distance
      [~, idx] = min(Dist);

      %% 6. Display the result stiring!
      disp(result_string_table{idx});
    end
end
```

6-3.客観的に良さを示す(2/2)

結果は右図のようになった

✓一人目の2回目のグーが正しく認識できない

✓二人目の音声についてはグーのみ正しく認識できていない

➢参照パターンが私のみであるから，ここの学習データを複数人の別々のデータを用いれば精度があがるのだろうと推測した

➢時間のずれについて考慮されていないので，参照パターンとのずれがあるため上手く認識できていないのかもしれない

```
>> ASR2

   1111.4   1370.7   1196.1
Goo
   1172.8   1253.6   1154.3
Par
   1911.4   1158.0   1385.6
Chk
   1961.9   1172.1   1479.3
Chk
   1505.0   1442.7   1116.8
Par
   1493.4   1503.7   1115.5
Par
   1990.0   1430.0   1518.6
Chk
   1923.9   1509.8   1450.4
Par
   2872.9   1996.5   2342.0
Chk
   2475.8   1678.5   1927.1
Chk
   1956.5   1606.2   1284.7
Par
   1874.7   1524.5   1216.5
Par
```

問題6のまとめ

問題

自身で実装した自動音声認識器を用いて，音声インタフェースを作成し，考察せよ．

まとめ

ソースコードをしめしながら，実装について説明し，実際に動作させて動作例を示した

友人に協力してもらい，自分以外のデータを識別させて，識別させた数値を示し考察をした

謝辞

データ提供していただいた大嶋陵示さん，小原俊一さんに感謝申し上げます．

問題7

最適経路探索問題を応用して，スペクトログラムによる音声間の距離計算を実装し，考察せよ.

- 少なくとも Dijkstra’s algorithm による実装をおこなうこと.

問題7の小問題

1. 探索結果をグラフとして図示し，考察しなさい．
   1. Dijkstra’s algorithmによる探索で何か問題は発生していないか？
   2. 探索効率は良いと言えるだろうか？
2. 問題4で収録した6つの音声を用いて，各組み合わせ間の距離を計算し，その結果を表としてまとめなさい．
3. 音声認識の実現という観点から，距離計算の表について考察しなさい．
   1. 例えば，想定どおりの傾向を示しているか？

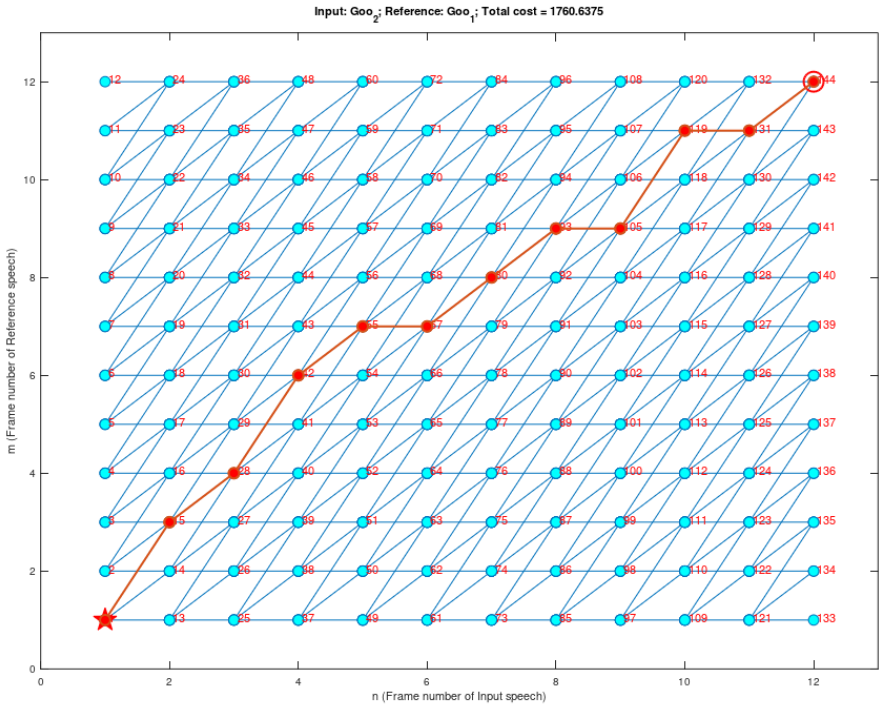
7-1.探索結果をグラフとして図示し，考察

以前録音したGoo1とGoo2で探索をした

【考察】

音声データを12個のフレームに分割した
Dijkstra’s algorithmによる探索でも上手く目標ノードまで探索が出来ているように見える

計算過程を見ると，ノードの繋がっているところ(三叉全て)もみて最短距離を通っていたため，総じて最短の経路を探索している




```
>> mon7

Got a Target node!
route =

     1    15    28    42    55    67    80    93   105   119   131   144

distance =  1760.6
```

7-2. 6つの音声の各組み合わせ間の距離

グーの1つ目と，グーの2つ目・チョキの2つ目・パーの2つ目

チョキの1つ目と，グーの2つ目・チョキの2つ目・パーの2つ目

パーの1つ目と，グーの2つ目・チョキの2つ目・パーの2つ目

をそれぞれ計算し距離の表にしたのが右表である

	Goo2	Chk2	Par2
Goo1	1760.6	2864.8	2202.8
Chk1	2559.4	2259.1	2789.3
Par1	2182.4	2920.9	2272.4

7-3.音声認識の実現という観点から，距離計算の表について考察

【考察】

音声認識の観点から見ると，同じ音同士つまり表の対角要素にある値が小さいはずである．しかし，パーの音だけ，グーと100程度しかコストが違わないため誤認識してしまう可能性がありそうであった．

グーとグー・グーとパーの比較では確実にグーと認識してくれそうだが，パーとグー・パーとパーのコストの差が逆転してしまっていることが不可解．音声波形が途切れていることが原因だと考えられる．

問題4でのスペクトログラムを見ると，割とグーとパーが似ているため，同じ用な音に対して何らかの特徴付けのための要素を考える必要がありそう．または，参照データを多くかき集めた優秀なものを作る必要があると考えた．

問題7のまとめ

問題

最適経路探索問題を応用して，スペクトログラムによる音声間の距離計算を実装し，考察せよ.

- 少なくとも Dijkstra’s algorithm による実装をおこなうこと.

まとめ

- Dijkstra’s algorithmによる探索を実装し，時間的な音声の違いも認識できるようにした
- コストの表を作成し，音声認識の観点から考察を行った

まとめ

じゃんけんの三手を認識できるような，単語音声認識器を構築することを目指し，音声認識器を作るために音声（メディア情報）をディジタルデータとして処理する方法や所望の情報を表現しうる特徴量ベクトル（行列）に変換する手法と，特徴量ベクトル（行列）を用いた識別器を構成するという操作を理解し，簡単な音声の特徴量を用いて，基本的な識別器を構築することで，単語音声認識器の可能性を検討した．

まず，MATLAB　Octaveでの音の録音，再生，保存，読み出しについてデモ音を用いてその音声波形を表示し，音をデータとして扱う方法について，ソースコードを示すことでそれらについて理解をした．

そして，得られた音のデータを離散フーリエ変換により，音声の周波数的な特徴をパワースペクトルにてグラフ出力することで見られたことをまとめた．

また，スペクトログラムというものを用いて，時間経過による周波数の特徴の変化についても目視できるように，実装を行い考察をした．

まとめ

さらに，グーチョキパーという自分の音声を録音し，時間波形やパワースペクトルについて観察したあとに，参照パターンと入力音声のスペクトル距離の比較に基づく単語音声認識について考察した．その後，自身で自動音声認識器を実装し，声を正しく認識できるか見て，友人のデータも借りてその識別器の精度についても検討を行った．

最後に最適経路探索問題を応用して，Dijkstra’s algorithmでスペクトログラムによる音声間の距離計算を実装し音声認識器の精度向上も行った．