第3章　スペクトル分析

音声認識のためのスペクトル分析の目的は，音声波形から，たとえば「あ」と「い」をうまく区別するための特徴を抽出することです．「あ」と「い」を発音するとき，我々は舌の位置や口の構えなどを変化させています．「あ」と「い」の区別は，声の高さを変えても話す人が変わっても変わりません．音声認識に必要なのは声の高さや声の個人性といった特徴ではなく，発声器官，とりわけ**声道（vocal tract）**の形に関する特徴です．

3.1　短時間フレーム分析

スペクトル分析では，音声波形から連続する短い時間区間（数十 ms）を切り出し，切り出された信号が定常信号であると仮定して分析を行います．切り出す単位を**フレーム（frame）**といいます．一定の幅の短時間のフレームを一定の幅で時間軸方向にずらし，フレームの範囲の音声波形データを切り出して分析し，スペクトル特徴量（特徴ベクトル）を求めます（図 3.1）[1]．この処理をフレームが音声波形の終端に達するまで繰り返します．このような分析を**短時間フレーム分析（short time frame analysis）**といいます．

一度に切り出す音声波形の長さを**フレーム幅（frame width）**といいます．フレームのずらし幅を**フレームシフト（frame shift）**といいます．普通，フレームシフトはフレーム幅より小さく設定し，隣り合うフレーム同士が一部重複するようにします．音声認識では，フレーム幅は 20 ms から 40 ms，フレームシフトは 5 ms から 25 ms の範囲のものが使われます．

フレーム分析で得られた一連のスペクトル特徴ベクトルを時間順に並べたものを音声認識の観測系列 $\boldsymbol{O} = o_1 o_2 \cdots o_T$（$T$ はフレーム数）として用います．D を音声波形の継続長，W をフレーム幅，S をフレームシフトとすると，この音声を分析するときのフレーム数は $\lfloor \frac{D-(W-S)}{S} \rfloor$ です[2]．例えば，1 s（=1000 ms）の音声をフレーム幅 32 ms，フレーム間隔 10 ms で分析する場合，フレーム数は $T = \lfloor \frac{1000-(32-10)}{10} \rfloor = \lfloor 97.8 \rfloor = 97$ となります．

3.2　高域強調

人間の音声の周波数成分のパワーは，有声音の場合 −6 db/oct，つまり周波数が 2 倍になるごとに 6 dB 下がる性質を持っています．これは次のような理由によります．音声を生成する過程は，(1) 音源の生成（有声音源・無声音源），(2) 声

[1] 各フレームの分析は，隣接フレームとの関係を考慮せず，個別に行ないます．
[2] $\lfloor\ \rfloor$：床関数．$\lfloor x \rfloor$ は実数 x に対して x 以下の最大の整数．

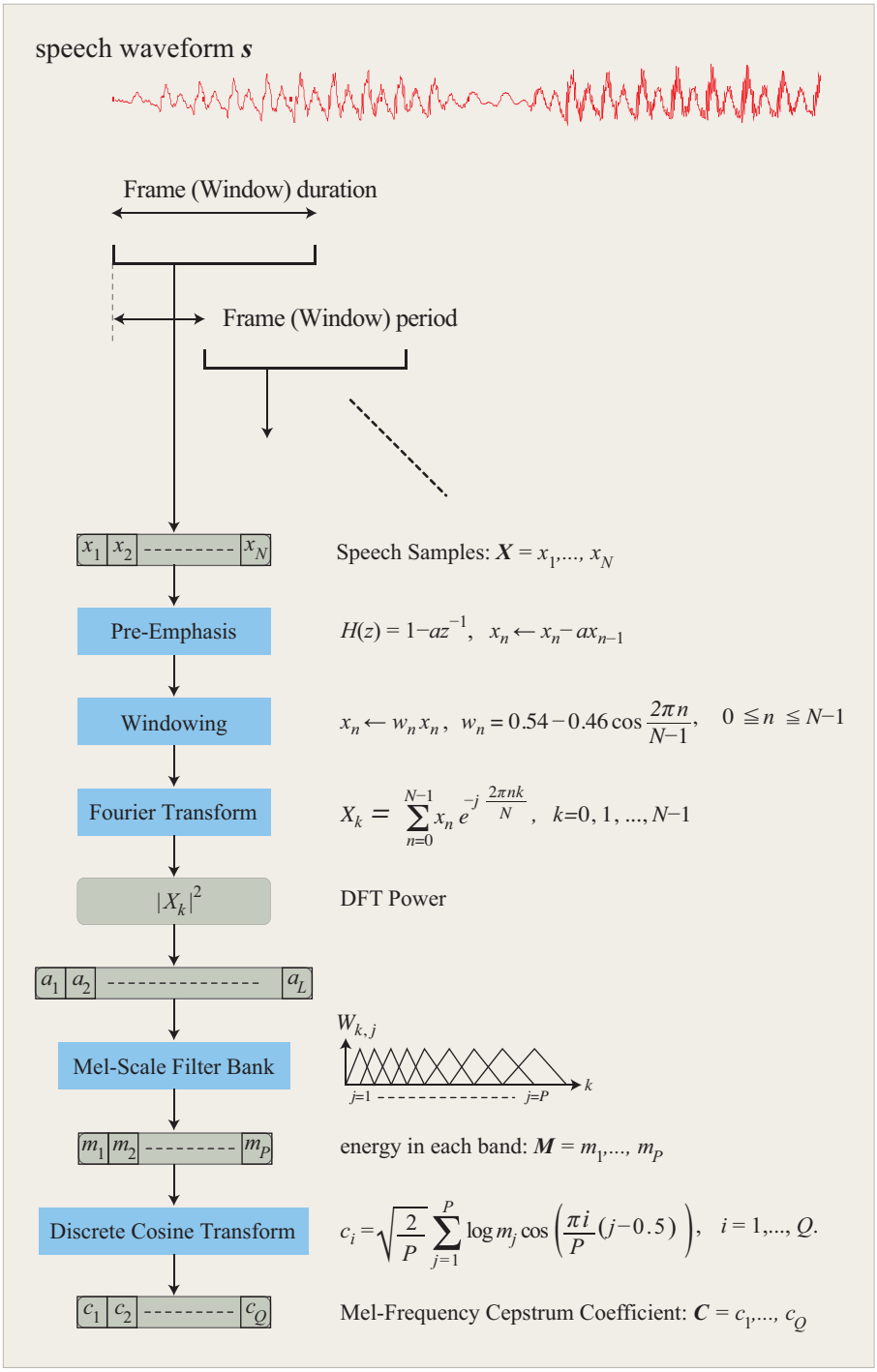


図 3.1: 音声波形からスペクトル特徴量 MFCC を計算する手順

道による調音，(3) 口・鼻からの放射の 3 つの作用の組み合わせとして，モデル化することができます（2.1.1 節）．有声音源はパルス列として近似することができるのですが，これはかなり強い高域減衰特性 −12 db/oct を持ちます．一方，放射特性は，逆に+6 db/oct の特性を持つので，最終的に音声は有声音の場合，−6 db/oct の特性を示すことになるのです（図 3.2）．

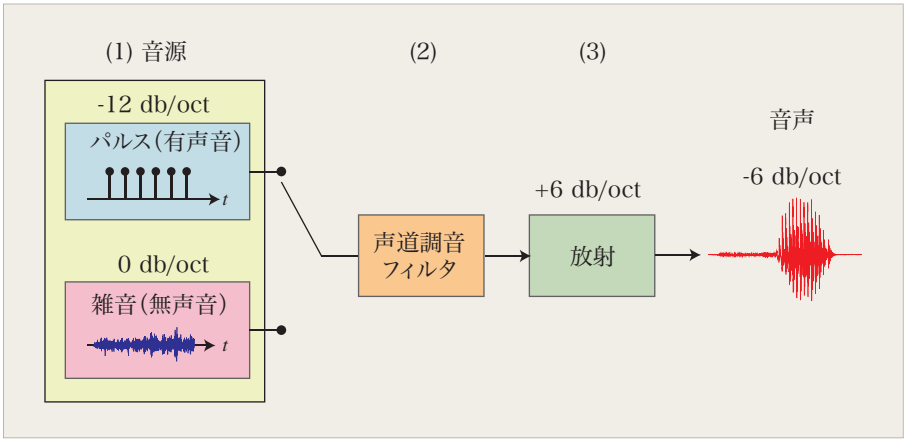


図 3.2: 音声生成の工学的モデル

この実験で我々が音声のスペクトルを計算する目的は，スペクトルに現れる音韻性（「あ」と「い」の性質の違いなど）の抽出です．音韻性は主として舌の形による声道調音フィルタの特性に由来します．したがって，音韻性の情報をより有効に利用するための最初の処理として，音声スペクトルの −6 db/oct の傾斜を元に戻します．すなわち，音声分析の前に +6 db/oct の補正を行います．これを**高域強調（pre-emphasis）**といいます．この処理は，音声振幅波形サンプル x_n に対する差分演算，

$$x_n \leftarrow x_n - \alpha x_{n-1}, \quad n = 1, \ldots, N$$

あるいは 1 次のディジタルフィルタ

$$H(z) \triangleq 1 - \alpha z^{-1}$$

によって行います．ここで，α は 1 に近い値（本実験では 0.97）に設定します．図 3.3 は，母音「あ」について高域強調のスペクトルに対する効果を示したものです．この処理により高周波数成分のパワーが持ち上がっているのがわかります．高域強調によってスペクトルが平坦化されるので，信号のダイナミックレンジを圧縮し，実質的に SNR[3]を向上させる効果もあります．

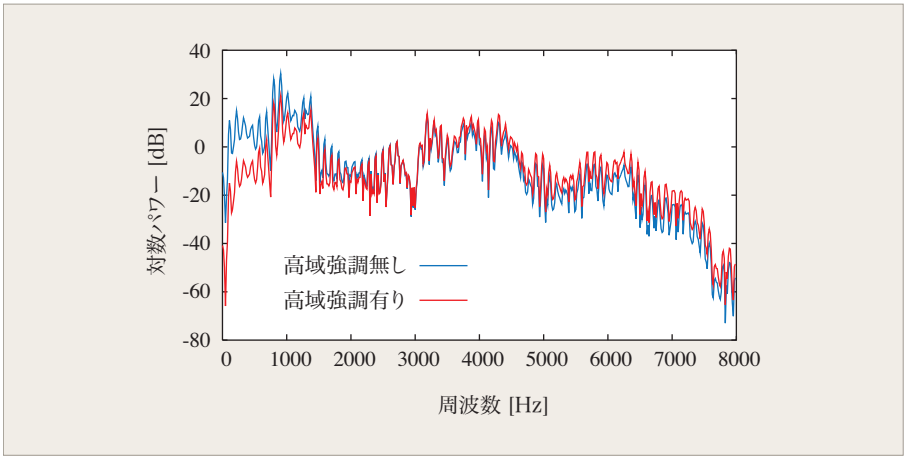


図 3.3: 高域強調の有無による「あ」の対数パワースペクトルの違い．

[3]Singal-to-Noise Ratio： 信号対雑音比．値が大きいほど良い．

3.3 窓関数

フレームで切り出した音声波形の端点は不連続になるので，信号の周期性を前提としたフーリエ変換などのスペクトル分析には不都合です．フレームの切り出しによる不連続の影響を低減するために，切り出した区間の両端の振幅を小さくするように重み関数を掛けることが行われます．このような関数を**窓関数（window function）**といいます．

音声分析で良く用いられる窓関数の 1 つに**ハミング窓（Hamming window）**があります．この窓 w_n は，

$$w_n \triangleq 0.54 - 0.46 \cos \frac{2\pi n}{N-1}, \quad 0 \leq n \leq N-1$$

で定義され，図 3.4（a) のような形をしています．この窓を，音声振幅波形サンプル x_n に掛け，

$$x_n \leftarrow w_n x_n, \quad n = 1, \ldots, N$$

としたものをスペクトル分析の入力として用います．図 3.4（b) は実際の音声波形からフレームで切り取った波形，その波形にハミング窓を掛けた後の波形です．

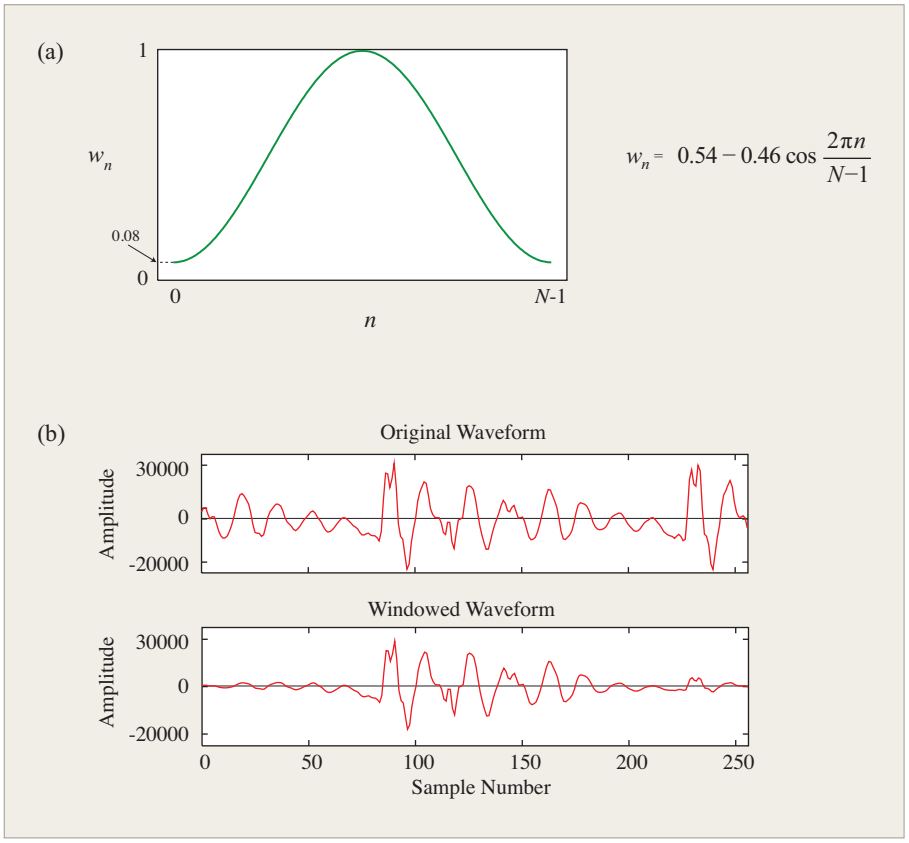


図 3.4: ハミング窓：(a) 窓関数，(b) 音声波形（上段）と窓掛けした音声波形（下段）．

3.4 離散フーリエ変換

音声波形サンプル $x_1, \ldots, x_N$ についての**離散フーリエ変換（Discrete Fourier Transform: DFT）**対は,

$$X_n = \sum_{k=0}^{N-1} x_k e^{-j2\pi nk/N}$$

$$x_k = \frac{1}{N} \sum_{k=0}^{N-1} X_n e^{j2\pi nk/N}$$

と定義されます．DFT を効率的に計算する手法として**高速フーリエ変換（Fast Fourier Transform: FFT）**があります．実習のプログラムでは FFT を用いています．FFT のアルゴリズムについては，信号処理の教科書およびソースコード`~/asr/wrecog/program/ad2fb.c` を参照してください．

3.5 メル周波数

聴覚の周波数分解能は，低い周波数では高く，高い周波数では低いというように，周波数に対して非線形の特性を持っています．スペクトル分析をこの非線型特性に基づいて行うと，音声認識で良い結果が得られることが知られています．聴覚の非線型特性を反映した周波数として良く用いられるものとして**メル周波数（Mel-frequency）**があり，周波数 f[Hz] とは,

$$\mathrm{Mel}(f) \triangleq 2595 \log_{10}(1 + \frac{f}{700}) \text{ [Mel]}$$

という近似関数 (図 3.5) で対応づけられることが知られています．

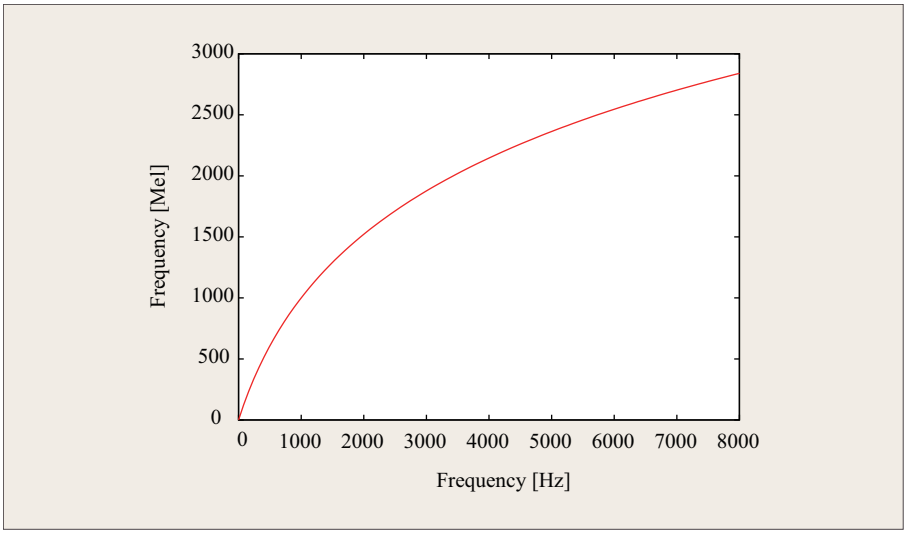


図 3.5: 周波数からメル周波数への変換関数 $\mathrm{Mel}(f)$

3.6 フィルタバンク分析

聴覚の周波数に対する対数的特性を反映した分析方法です．まず，フレームで切り出して，高域強調をし，窓関数を掛けた音声波形サンプルをフーリエ変換

し，パワースペクトル $|X_k|^2$ を求めます．つぎに，パワースペクトルをメル周波数軸上で等間隔，つまり，普通の周波数軸上では周波数が高くなるほど幅が広くなる三角窓が並んだ**フィルタバンク（filterbank）**で処理をします．各三角窓の範囲を**チャネル（channel）**といいます．三角窓毎に窓の範囲にあるパワースペクトル $|X_k|^2$ に対して[4]，窓の重み $W_{k,j}$ を掛けて和をとることにより，そのチャネルの出力

$$m_j \triangleq \frac{1}{A_j} \sum_{k=k_{\rm lo}(j)}^{k_{\rm hi}(j)} W_{k,j}|X_k|^2, \quad j = 1, \ldots, P \tag{3.1}$$

を計算します．P はチャネル数です．フィルタの重み $W_{k,j}$ は，

$$W_{k,j} \triangleq \begin{cases} \dfrac{k - k_{\rm lo}(j)}{k_{\rm c}(j) - k_{\rm lo}(j)}, & k_{\rm lo}(j) \le k < k_{\rm c}(j) \\ \\ \dfrac{k_{\rm hi}(j) - k}{k_{\rm hi}(j) - k_{\rm c}(j)}, & k_{\rm c}(j) \le k \le k_{\rm hi}(j) \end{cases} \tag{3.2}$$

となります．$k_{\rm lo}(j), k_{\rm c}(j), k_{\rm hi}(j)$ はそれぞれ j 番目のフィルタの下限, 中心, 上限のスペクトルの要素番号であり, 隣り合うフィルタ間で $k_{\rm c}(j) = k_{\rm hi}(j-1) = k_{\rm lo}(j+1)$ という関係があります．中心周波数 $k_{\rm c}(j)$ はメル周波数軸上では等間隔に並んでいます（図 3.6）．3.1 式の A_j はフィルタの面積を正規化する係数で，

$$A_j \triangleq \sum_{k=k_{\rm lo}(j)}^{k_{\rm hi}(j)} W_{k,j} \tag{3.3}$$

で定義されます．

フーリエ変換によって得られたスペクトル（図 3.3）には，細かく変化するギザギザの波形と大きく変化する凸凹があります．前者は有声音の音源（図 3.2）に由来する成分で，音源の基本周波数とその高調波成分です．後者は声道調音フィルタの特性を表わす成分です．各チャネルの帯域は基本周波数に比べてずっと広いので，チャネル内での積和計算（＝平均化）により高調波の影響が相殺されます．結果として，フィルタバンク出力には，声道調音フィルタ特性の概形が得られることになります．

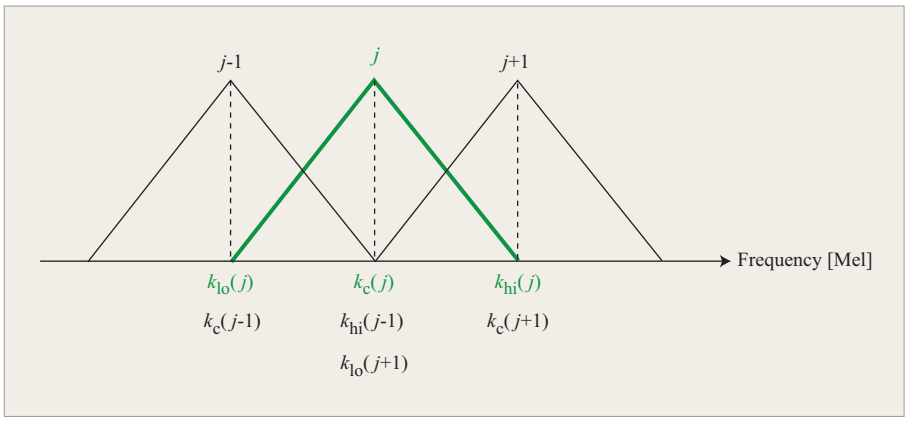


図 3.6: MFCC を計算するためのフィルタバンクの三角窓

[4]振幅スペクトル $|X_k|$ を用いる計算法もあります．

3.7　メル周波数ケプストラム係数（MFCC）

フィルタバンク分析によって得られた各チャネルの出力 $m_j (j = 1, \ldots, P)$ の対数を**離散コサイン変換（Discrete Cosine Transform: DCT）**

$$c_i \triangleq \sqrt{\frac{2}{P}} \sum_{j=1}^{P} \log m_j \cos\left(\frac{\pi i}{P}(j - 0.5)\right), \quad i = 1, \ldots, Q \tag{3.4}$$

することにより，**メル周波数ケプストラム係数（Mel-Frequency Cepstrum Coefficient: MFCC）**が得られます．Q はケプストラムの次元数です．

MFCC の値はマイクロフォンの種類，マイクロフォンと口の位置関係によって変動し，音声認識の結果に影響を与えます．その変動の影響を抑えるのに有効な方法として**ケプストラム平均分散正規化（Cepstral Mean and Variance Normalization, CMVN）**があります．第 t フレームのケプストラム係数ベクトルを $\boldsymbol{c}_t = (c_1^{(t)}, \ldots, c_Q^{(t)})$, ケプストラム係数ベクトルの時系列を $\boldsymbol{C} = \{\boldsymbol{c_1}, \ldots, \boldsymbol{c_T}\}$ とするとき，正規化されたケプストラム $\boldsymbol{c}_{t'}$ は，

$$\begin{aligned} \boldsymbol{\mu} &= \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{c_t} \\ \boldsymbol{\sigma} &= \sqrt{\frac{1}{T}\sum_{t=1}^{T} (\boldsymbol{c_t} - \boldsymbol{\mu})^2} \\ \boldsymbol{c}_{t'} &= \frac{\boldsymbol{c}_t - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \end{aligned}$$

と計算されます．ここで，$\boldsymbol{\mu}$ は平均値ベクトル，$\boldsymbol{\sigma}$ は標準偏差ベクトルです．平均値と標準偏差を計算する時系列 $\boldsymbol{C}$ の範囲は，音声認識処理の応用目的などによって，単語，文，段落，文章などが取られます．本実験では，単語単位の正規化を行ったものを特徴量として用いています．ちなみに，CMVN の処理は `ad2mfcc.c` というソースコードに記述されています．興味があれば確認しておいてください．

3.8　特徴抽出はデータ圧縮

音声波形から MFCC を得るまでの各段階でどのような結果が得られるのか，具体例で見てみましょう（図 3.7）．本実験の分析条件の下で順を追って説明します．サンプリング周波数は 16 kHz，フレーム幅は 32 ms なので，1 フレームの音声サンプル数は $16 \times 32 = 512$ です．フレーム内に/a/の波形が 3 周期入っています．高域強調を行い，窓関数を掛けた音声波形のサンプル数は 512 です．これをフーリエ変換して得た振幅スペクトル，および振幅スペクトルを 2 乗したパワースペクトルの点数は 256 になります．スペクトルで大きな値を示しているのは/a/の**フォルマント（formant）**（2.2.1 節）に対応する周波数成分です．パワースペクトルでは振幅スペクトルに比べて周波数成分の強弱が強調されています．メルフィルタバンク分析のチャネル数は 28 なので出力は 28 次元です．チャネルの幅は低い周波数ほど広いので，スペクトルに現れているフォルマントピークは高周波数側が圧縮された位置で表示されています．フィルタバンク出力の対数を離散コサイン変換して得られる MFCC は 20 次元です．

スペクトル分析の過程において，音声認識に必要な情報を抽出しつつ，パラメータ数は 512 から 20 まで 1/25.6 に減っています．画像など他のメディアのパターン認識においても，このような**特徴抽出（feature extraction）**によって，不要なデータを捨て，必要な情報のみを残して認識に利用しています．つまり，特徴抽出は**データ圧縮（data compression）**とみることができます．

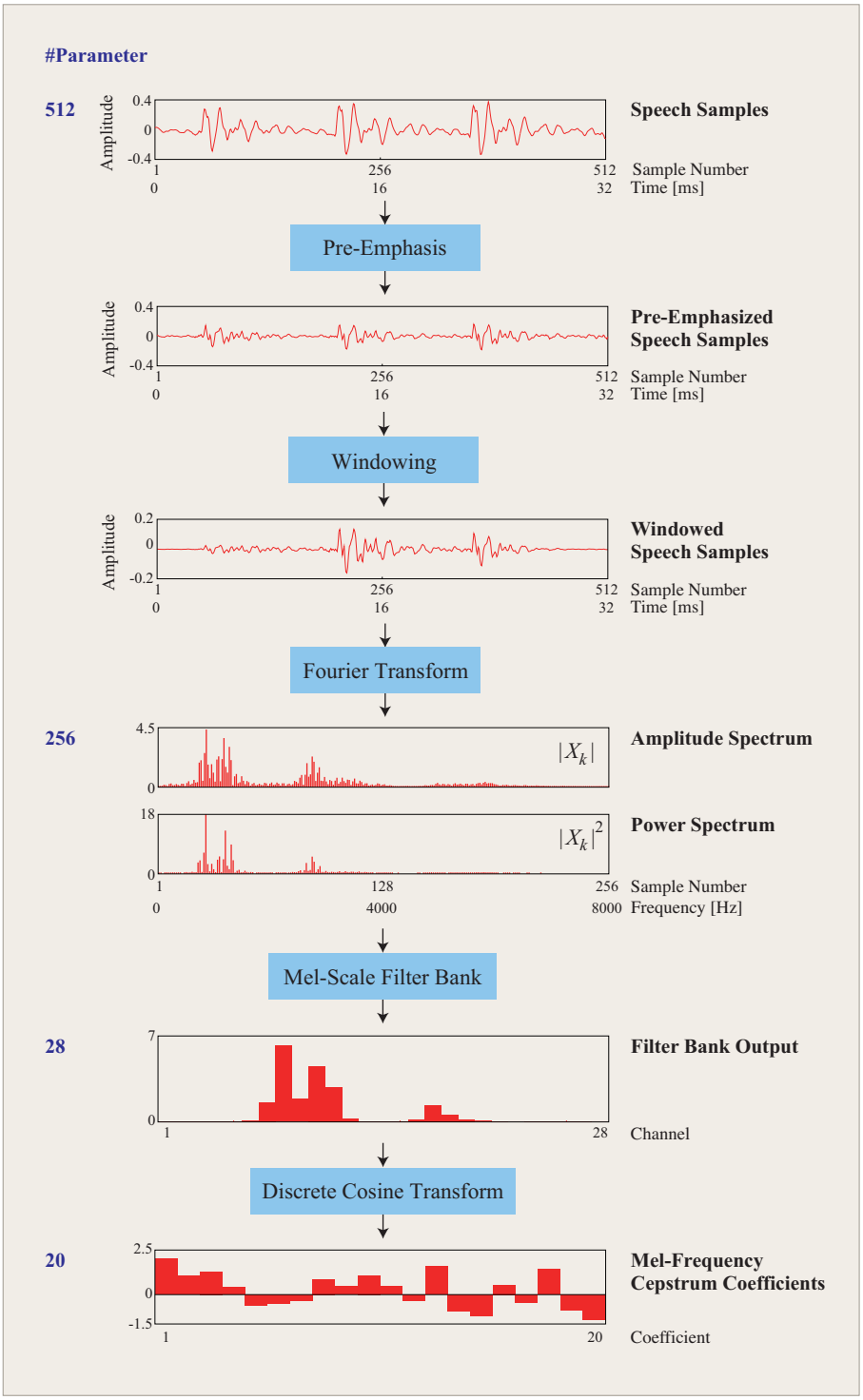


図 3.7: 512 点の/a/の音声波形をスペクトル分析し 20 次元の MFCC を計算．音声認識に必要な情報を抽出しつつ，パラメータ数は 512 から 20 まで 1/25.6 に削減することができました．特徴抽出はデータ圧縮の処理でもあるともいえます．

3.9　スペクトル分析例

実際の音声のスペクトル分析例で，音の違いがどのようにスペクトルのパターンに現われているかみましょう．図 3.8 は/kakuritsu/（「確率」）という単語の音声波形，スペクトログラム，音素ラベル，MFCC の値を表示したものです．

音声波形データの最初と最後には 345 ms の無音区間があります．単語音声は 345 ms から 1000 ms に存在し，その部分では音声波形の振幅が大きくなっています（図 3.8（a））．音声波形に対応して，スペクトログラムの濃淡パターン（図 3.8（b））は時間とともに絶えず変化していますが，局所的なパターンには他の部分とは明らかに異なった特徴があることがわかります．図 3.8（c）には，この波形の音素（第 2.2 節）名および音素の開始時間と終了時間を図示しました．この情報に基づいて音声波形とスペクトログラムに音素境界を引いてあります．そのようにしてみると，音によって波形やスペクトログラムのパターンに特徴があることがわかります．例えば，母音は 4 kHz 以下の低い周波数のエネルギーが強いこと，子音 /ts/ はその逆であること，さらに，同じ音素 /u/ であっても前後の音によってパターンに違いがみられることなどがわかります．MFCC（図 3.8（d））のパターンも，局所的に特徴のあるパターンがみられ，音素ラベルと対応している様子がわかります．

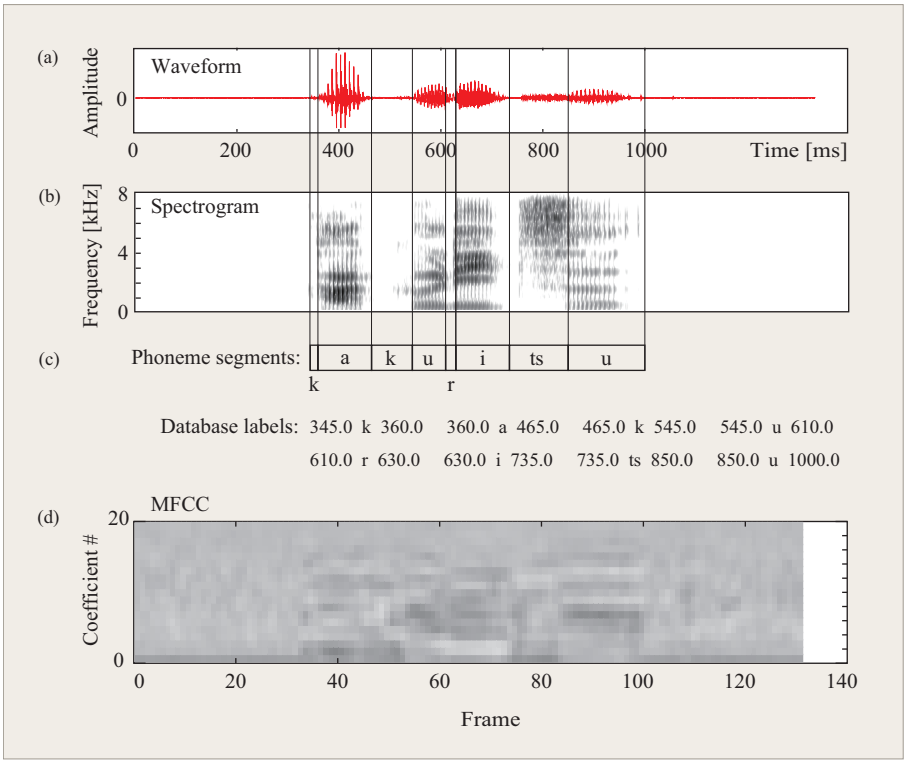


図 3.8: 音声のスペクトル分析例（/kakuritsu/「確率」）．(a) 音声振幅波形, (b) スペクトログラム（周波数成分の強さを濃淡で表わしたもの）, (c) ラベル情報, (d) メル周波数ケプストラム係数（MFCC）．

3.10 実習

音声波形から MFCC を求める過程のうち，フィルタバンク分析と MFCC 分析の計算式を C 言語でプログラムし，サンプル音声を用いて計算結果を確認します．以下の説明は，計算機実験を~/asr/wrecog で行うことを前提としています．端末を開き，事前にこのディレクトリに移動しておいてください．

3.10.1 フィルタバンク分析

program というディレクトリに ad2fb.c という C の言語のソースファイルがあります．これは，音声波形からフィルタバンク出力を計算する関数です．ただし，肝心のコードの部分が空白になっています．穴埋め指示のある部分を埋めてソースコードを完成させてください．完成したら，以下のコマンドを入力してコンパイルを実行します．

```
[~/asr/wrecog]% make -C program fb
```

ソースコードに C 言語の文法エラーがある場合は，その旨メッセージが出力されるので，エラーメッセージが出なくなるまで修正してください[5]．文法エラーがない場合は，fb というコマンドができています．これは，音声波形を分析してフィルタバンクを出力するコマンドです．プログラムが正しく出来ているか確認しましょう．端末で，次のように入力して，計算確認用のサンプル音声~/asr/wrecog/sample/sample.wav を分析し，結果を./mysample.fb に保存します．

```
[~/asr/wrecog]% fb sample/sample.wav ./mysample.fb
```

このファイルと正解ファイルを比較して同じであれば，フィルタバンク分析のプログラムが正しく書けていることになります．比較のために，

```
[~/asr/wrecog]% head -28 sample/sample.fb ./mysample.fb
```

と入力し，正解と自分の分析結果の第 1 フレームのフィルタバンク出力値を表示します．1 列目は時間 [ms]，2 列目はチャネル番号，3 列目はフィルタバンク出力値です．

```
[~/asr/wrecog]% head -28 sample/sample.fb ./mysample.fb
==> sample/sample.fb <==
  10.0   1       6.926785e-04
  10.0   2       5.409857e-01
  10.0   3       4.872707e-01
  （中略）
  10.0   26      4.739638e-02
  10.0   27      2.987086e-02
```

[5] コンパイル時にエラーメッセージが出なくなっても，プログラムの内容に間違いがないとは限りません．

```
  10.0  28     2.556659e-02

==> ./mysample.fb <==
  10.0  1      6.926785e-04
  10.0  2      5.409857e-01
  10.0  3      4.872707e-01
  （中略）
  10.0  26     4.739638e-02
  10.0  27     2.987086e-02
  10.0  28     2.556659e-02
```

この実行例では数値が一致していますが，計算の順序などによっては数値計算の誤差により若干異なる数値になることがあります．その場合でも，少数点以下 5 桁まで一致していれば問題ありません．

3.10.2 MFCC 分析

program というディレクトリに fb2mfcc.c という C の言語のソースファイルがあります．これは，フィルタバンク出力から MFCC を計算する関数です．ただし，肝心のコードの部分が空白になっています．穴埋め指示のある部分を埋めてソースコードを完成させてください．完成したら，以下のコマンドを入力してコンパイルを実行します．

```
[~/asr/wrecog]% make -C program mfccf
```

ソースコードに C 言語の文法エラーがある場合は，その旨メッセージが出力されるので，エラーメッセージが出なくなるまで修正してください．文法エラーがない mfccf というコマンドができています．これは，音声波形を分析して MFCC を出力するコマンドです．プログラムが正しく出来ているか確認しましょう．端末で，次のように入力して，計算確認用のサンプル音声 sample/sample.wav を分析し，結果を./mysample.mfcc に保存します．

```
[~/asr/wrecog]% mfccf sample/sample.wav ./mysample.mfcc
```

このファイルと正解ファイルを比較して同じであれば，MFCC 分析のプログラムが正しく書けていることになります．比較はつぎのように行います．.mfcc ファイルはバイナリ形式なので，まず，prtmfcc コマンドによって，テキスト形式のファイルに変換します．

```
[~/asr/wrecog]% prtmfcc ./mysample.mfcc > ./mysample.mfcc.txt
```

つぎに，

```
[~/asr/wrecog]% head -20 sample/sample.mfcc.txt ./mysample.mfcc.txt
```

と入力し，正解と自分の分析結果の第 1 フレームのフィルタバンク出力値を表示します．1 列目は時間 [ms]，2 列目は次元番号，3 列目は MFCC の値です．

```
[~/asr/wrecog]% head -20 sample/sample.mfcc.txt ./mysample.mfcc.txt
==> sample/sample.mfcc.txt <==
  10.0  1      0.997497
  10.0  2      -2.034434
  10.0  3      1.693631
  （中略）
  10.0  18     2.062192
  10.0  19     -1.878791
  10.0  20     -0.653017

==> ./mysample.mfcc.txt <==
  10.0  1      0.997497
  10.0  2      -2.034434
  10.0  3      1.693631
  （中略）
  10.0  18     2.062192
  10.0  19     -1.878791
  10.0  20     -0.653017
```

この実行例では数値が一致していますが，計算の順序などによっては数値計算の誤差により若干異なる数値になることがあります．その場合でも，少数点以下 5 桁まで一致していれば問題ありません．

3.10.3 音声収録の練習

自分の声を録音してスペクトル分析をしてみましょう．ヘッドセットを用いて音声を録音し，録音した音声を聞いて正しく録音できているか検査します．普通の録音と異なり，実験用データとして音声収録を行うので，その手順には特別な注意が必要です．今回の実習のための手順を説明します．

音声の入出力を伴う実習を行うときは，音声入出力を伴う他のアプリケーション（メディアプレイヤー，YouTube や goo 辞書など）は，終了しておいてください．WaveSurfer など実習に用いる音声入出力アプリケーションの動作に影響を及ぼすことがあります．

1. 録音

   (a) ヘッドセット（ELECOM HS-HP22TBK あるいは USB 接続の HS-HP14SUBK・HS-HP07SUBK）のコードを解き，フレキシブルパイプの根本の固い部分を持って，180 度回転してください．マイクロフォンが左側に来るようにヘッドセットを装着します．マイクロフォンの先端は口のやや下にセットします（図 3.9）．マイクロフォンの角度を変えるときは，本体の取り付け部の固い部分を回してください．マイクアームの柔らかいアームの部分は微調整に用います．マイクアームを急な角度で曲げたり，捩ったりしないように扱ってください．これは，できるだけマイクロフォンと口の位置関係を一定に保つことにより，入力特性を安定させるためです．ヘッドセットを頭に付けずマイクを持って録音しないようにしてください．再生音量を調整するボリュームコントローラーがありますので，確認してください（HS-HP07SUBK はボリュームコントローラーとミュートスイッチがヘッドフォン部にあります．）．

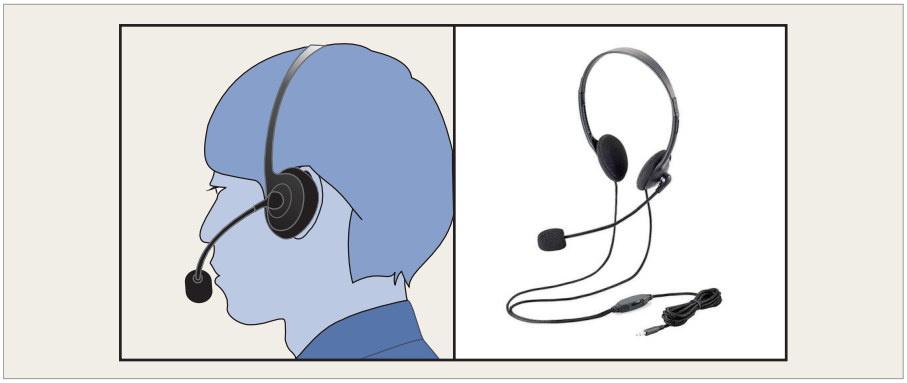


図 3.9: ヘッドセットは頭部にきちんと装着し，マイクは口のやや下にセットしてください．このイラストは HS-HP22TBK です．

   (b) PC の本体前面に音声入出力ジャックがあります．ヘッドセットの 4 極ミニプラグをしっかりと奥まで差し込んでください．プラグが適切に差し込まれると「オーディオデバイスを選択」が表示されるので，「ヘッドセット」を選択してください．（図 3.10）．

図 3.10: ヘッドセットの 4 極ミニプラグが適切に差し込まれると，「オーディオデバイスを選択」が表示される．「ヘッドセット」をクリックする．

(c) 作業用のディレクトリ（`~/asr/wrecog`）に移動します．ディレクトリの移動は端末で `cd` コマンドによって行います．現在のディレクトリを確かめるためには `pwd` コマンドを使います．

(d) 音量調節をします．ランチャー（デスクトップの左下角）から「設定」を起動してください（図 3.11）．

(e) メニューから「サウンド」を選択します（図 3.13）

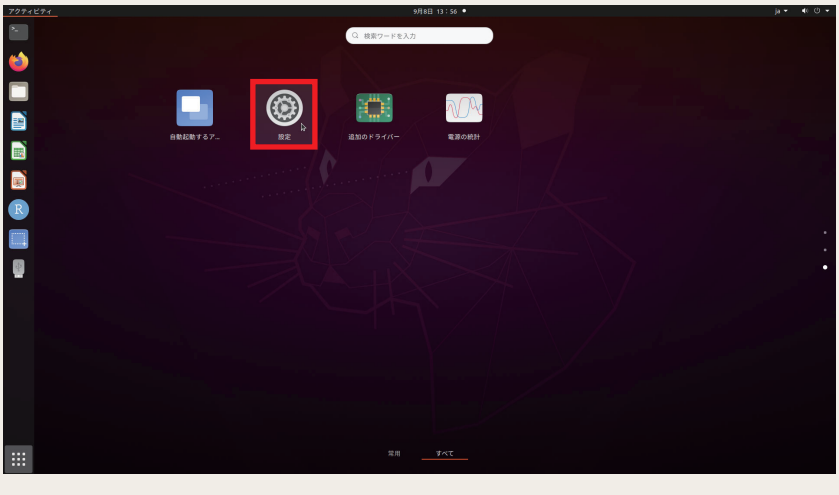


図 3.11: 音声入出力の設定をするために「設定」パネルを起動する．アイコンの配置はユーザによって異なります．

出力 「出力デバイス」メニューで「アナログヘッドフォン-内部オーディオ」を選択します．次に「テスト」クリックして，ヘッドフォンの左右の音の再生を検査します．マイクロフォンが取り付けられている方が左側です．「システム音量」と「音量レベル」を調節して，聴きやすい音量にします（図 3.13）．

入力 「入力ディバイス」メニューで「ヘッドセットマイクロフォン-内部オーディオ」を選択します．次に，音量を調節します．リラックスした楽な発声で適当な文を読み上げ（例えばこの文），音量の表示が 7 割付近で振れる状態になるように，スライドを調節します（図 3.14）．このパネルはデスクトップ画面に出したままにしておいてください．「サウンド」設定パネルが開いだ状態でないと，音声が入力されないことがあります．入出力音量は実験の途中で適宜調節するとよいでしょう．HS-HP07SUBK のマイクは指向性なので，音量が大きく出ない場合はマイクの位置を上下させるなどして試してください．

USB ヘッドセットの場合 PC の本体前面の USB-A のポートに差し込んでください．4 極ミニプラグの場合のように「オーディオデバイスを選択」の表示（図 3.10）はありません．「出力ディバイス」では「デジタル出力（S/PDIF）- USB PnP Audio Device」あるいは「アナログ出力 - USB PnP Audio Device」のいずれかを選択してください（どちらでもよい）(図 3.12)．「入力ディバイス」では「マルチチャンネル入力 - USB PnP Audio Device」あるいは「マイク - USB PnP Audio Device」のいずれかを選択してください（どちらでもよい）(図 3.12)．その後，上記の出力音量および入力音量の調整をします．

図 3.12: USB ヘッドセットの場合

①「サウンド」の「出力ディバイス」を「アナログヘッドフォン - 内部オーディオ」に設定する .

②「テスト」をクリックして，ヘッドフォンの左右の音の再生を検査する．このとき，「システムの音量」と「音量レベル」を調節しておく．

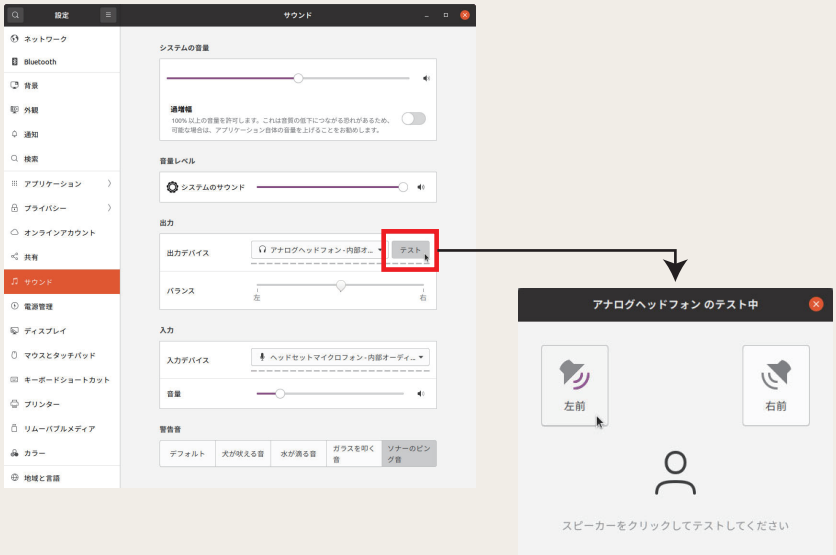


図 3.13: ヘッドフォンの左右確認と音量調節

①「サウンド」の「入力ディバイス」を「ヘッドセットマイクロフォン - 内部オーディオ」に設定する .

② リラックスした楽な発声で適当な文を読み上げ（例えばこの文），
音量の表示が 7 割付近で振れる状態になるように，スライドを調節する．

図 3.14: 入力音量の調節

(f) WaveSurfer [6] を起動します．端末のコマンド入力で，

```
% wavesurfer &
```

と入力してください．

(g) 録音条件を設定します．サンプリング周波数を 16kHz，1 チャンネルに設定してください．まず，「File」メニューの「Preferences...」を選択します（図 3.15）．

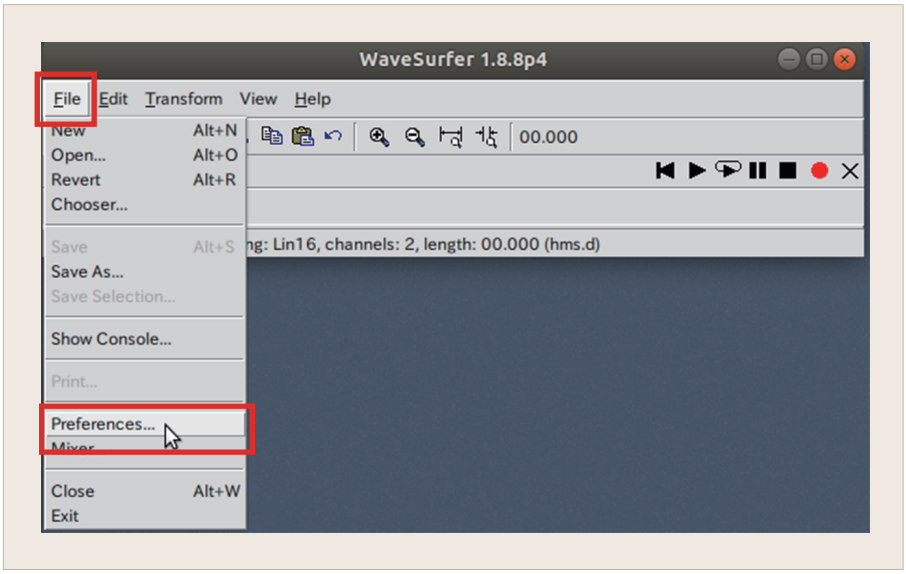


図 3.15: WaveSurfer の録音条件の設定パネルの起動

つぎに，「Sound I/O」パネルの「New sound default rate:」に “16000”，「New sound default channels:」に “1” を設定します．「New sound default encoding:」を “Lin16”，「Record time limit:」を”600” に設定し，パネル下に表示されている「OK」ボタンを押してください（図 3.16）．

[6] スウェーデンの KTH が開発したオープンソースの音声分析ソフトウェア．Linux, Windows, macOS で動作．https://sourceforge.net/projects/wavesurfer/

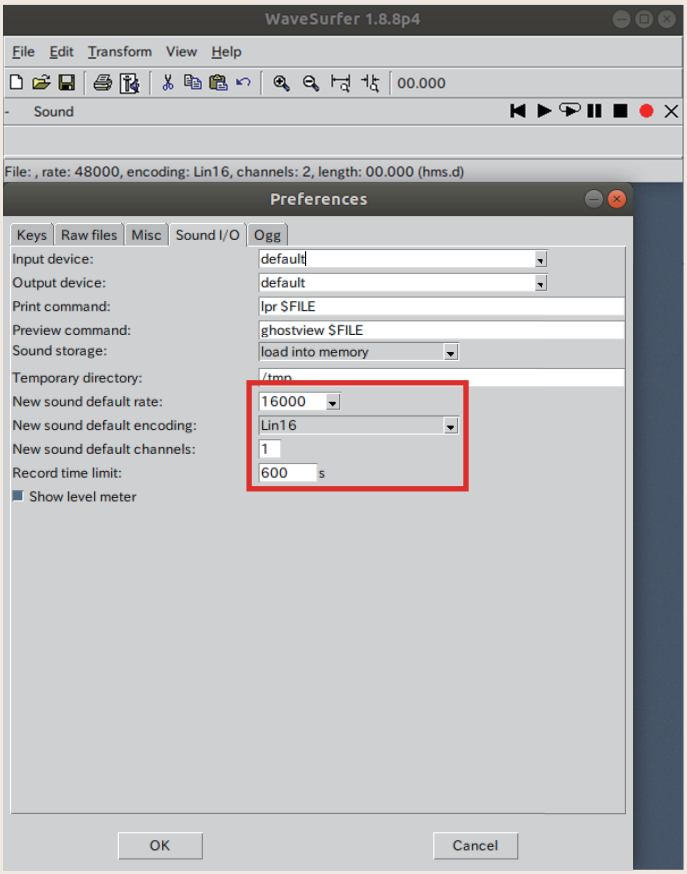


図 3.16: WaveSurfer の録音条件の設定．サンプリング周波数=16000Hz（16kHz），チャネル数=1．録音時間制限は 600 秒．

(h) WaveSurfer で新しい窓を生成します．「File」メニューの「New」を選択すると（図 3.17 (a)），新しい窓の種類を指定するための「Choose Configuration」というダイアログボックスが表示されます．「Waveform」を選択してください（図 3.17 (b)）．

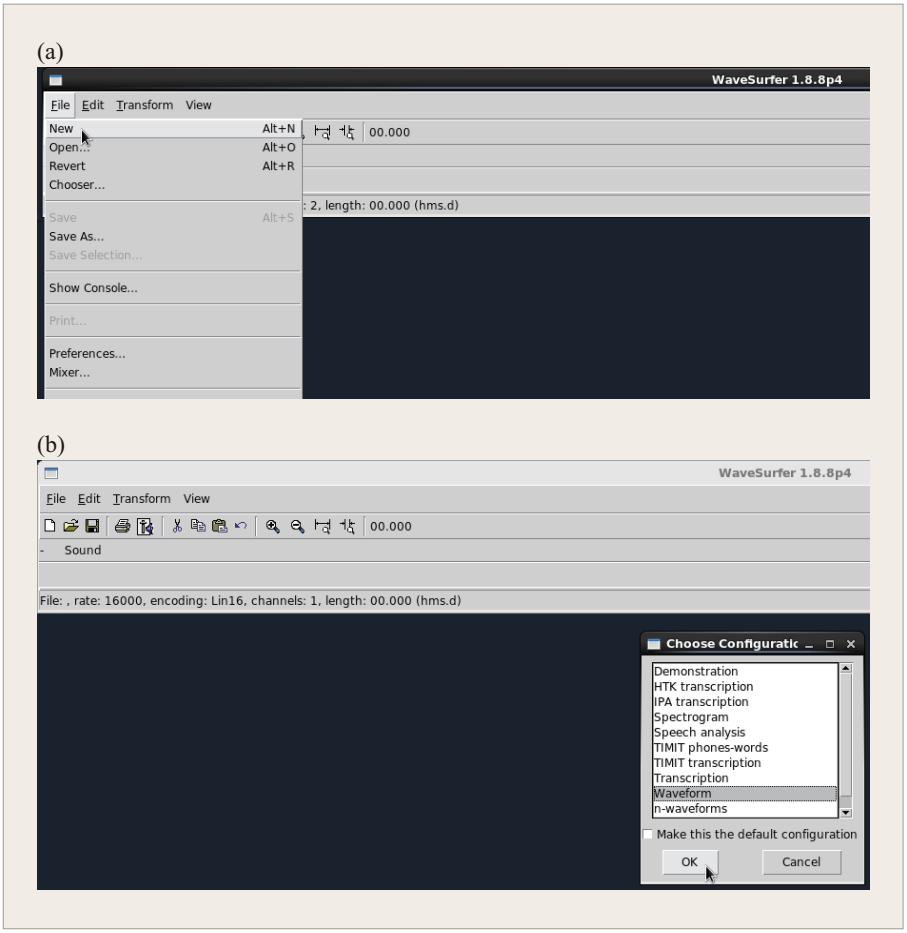


図 3.17: WaveSurfer を起動して新しい窓を作成する．

(i) WaveSurfer の窓の中央でマウスを右クリックして押したままとし，現われたメニューの「Create Pane」を選択して現われたメニューから「Time Axis」を選択して時間目盛を表示します（図 3.18）．時間目盛は左端に「time」と表示されていますが，初期状態では空白です．

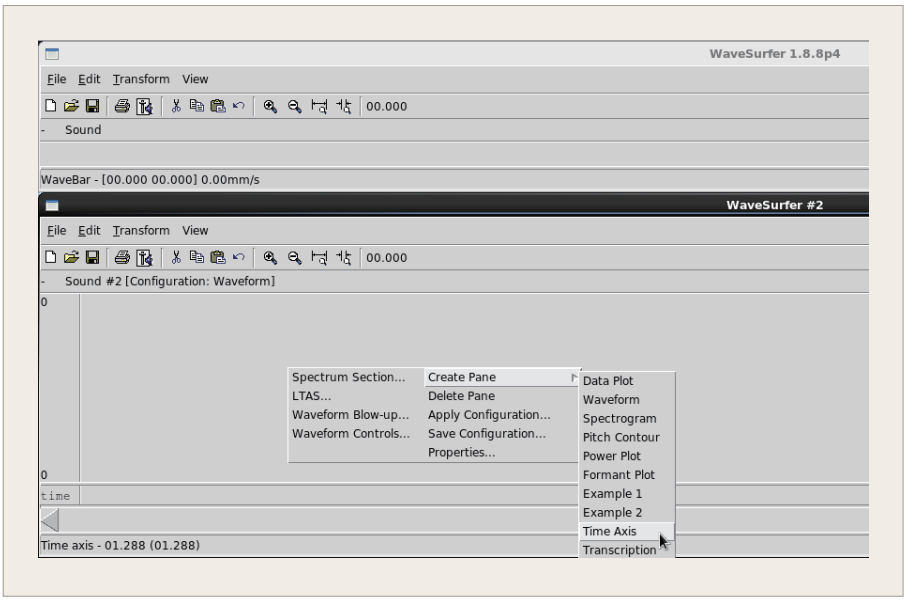


図 3.18: 時間目盛を表示する．

(j) WaveSurfer の録音ボタンを押すと収録が始まります．画面に音声波形がリアルタイムに表示されます．自分の名前（姓（family name）または名（given name）あるいはそれに相当する呼び名）を 5 回録音してください．普通に会話をするときの自然な感じで発声してください．単語の間は 1 秒程度置くようにしてください．音量の調整はマイクの位置でも調整することができます．波形の振幅が範囲（−32768〜+32767）を超えてしまうと（図 3.19），歪んだ音声になってしまいます．歪んだ音声は音声分析・認識のデータとして用いることができません．今回は練習として 5 回発声しますが，実際に分析に用いるのは 5 つの発話のうちの 1 つです．一般に音声収録において，最初の方の発声は不安定で，中間辺りで発声された単語がサンプルとして適当な場合が多いです．2 番目以降から比較的明瞭発音のものを 1 つ選んでください．

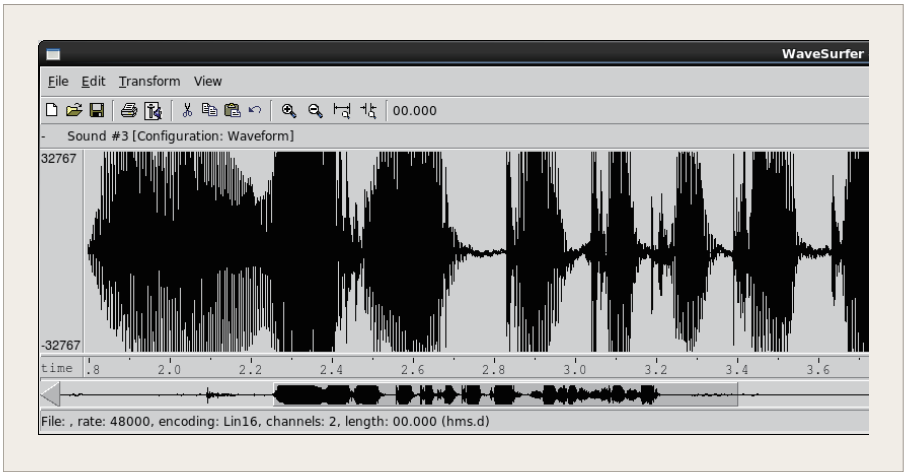


図 3.19: 入力レベルが過大で最大値を越えた部分は音声が歪んでしまう．

(k) 停止ボタンを押して，録音を終えます．WaveSurfer は録音した音声波形全体をバッファに蓄えています．

(l) 再生ボタンを押して，一度全体を聞きましょう．どうでしたか？ 録音された音声が妥当かどうか判断できない場合は，スタッフに聞いてください．

(m) 録音した音声を編集します．まずは，再生ボタンを押して，一度全体を聴きましょう．5 つの中から良い発声と思うものを 1 つ選んで保存します．音声の一部だけを選んで聴きたい場合は，マウスの左ボタンを押しながら範囲を指定し，再生ボタンを押します（図 3.20）．

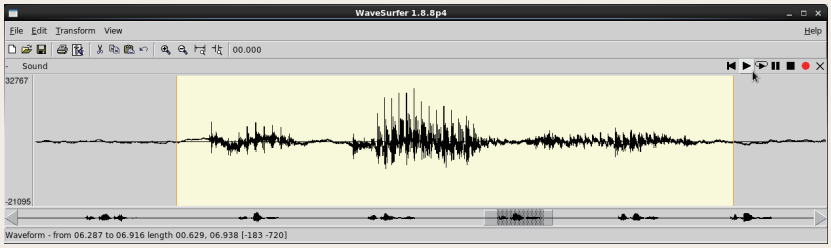
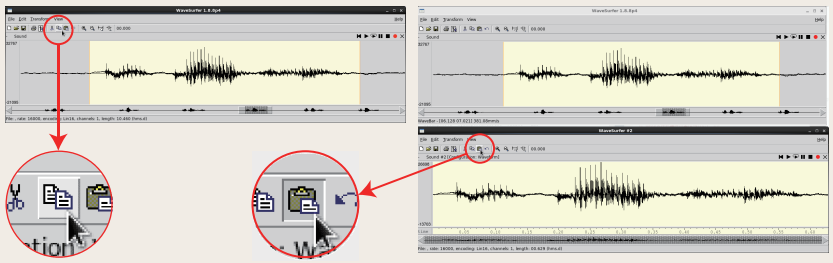


図 3.20: 音声の一部を選択する．

次に，「File」メニューの「New」を選択して新しい窓をつくり，そこに 5 つの発話がある元のファイルから選択した 1 つの発話の波形をコピー＆ペーストします（図 3.21）．音声の前後に 100ms（0.1s）の余白を付けてください．横軸の表示倍率によって目盛の時間幅が異なるので，間違えないように十分注意してください．余白が多すぎると，スペクトル表示の結果を見たときに，音の特徴やスペクトルの変化を観察しにくくなります．



(a) 波形を選択してコピーボタンを押す　(b) 新しい窓でペーストボタンを押す

図 3.21: 選択した音声をコピーして，新しい窓（バッファ）に貼り付ける．

(n) 編集が済んだ録音データをファイルに保存します．メニューから「File」→「Save As」を選択し，ファイル名を入力します．「Files of type:」は「MS Wav Files (*.wav,*.WAV)」を指定してください．ファイルを保存するディレクトリは`~/asr/wrecog/wav` とし，ファイル名は ‘‘名前のローマ字表記`.wav`’’ とします（図 3.22）．以下では，録音した音声データファイルを `takagi.wav` として説明します．

図 3.22: WAV 形式で音声波形を保存する．保存するディレクトリは ~/asr/wrecog/wav．拡張子.wav を付け忘れないように．

(o) WAV 形式で音声データで保存されていることを確認するため，端末で，play wav/takagi.wav と入力し，録音した音声が再生されることを確認してください．play コマンドを実行すると，ヘッドセットから音声が聞こえ，

```
[~/asr/wrecog]% play wav/takagi.wav
play WARN alsa: can't encode 0-bit Unknown or not applicable

wav/takagi.wav:

File Size: 28.8k     Bit Rate: 256k
Encoding: Signed PCM
Channels: 1 @ 16-bit
Samplerate: 16000Hz
Replaygain: off
  Duration: 00:00:00.90

In:100%  00:00:00.90 [00:00:00.00] Out:14.4k [ -=====|====- ]     Clip:0
Done.
[~/asr/wrecog]%
```

というように端末に表示されます．“Encoding:”(波形値記録形式),”Channels:”（チャネル数および量子化ビット数),”Samplerate:”（サンプリング周波数）が上記の例と同じでなければなりません．“Duration” は音声データの長さです．この際に表示される “rec WARN alsa: can't encode 0-bit Unknown or not applicable” のメッセージは無視して結構です．

2. スペクトル分析

保存した音声のフィルタバンク分析および MFCC 分析を行い，音声波形と並べて図として表示します．そのために drawspec というコマンドを使います．端末で，

```
[~/asr/wrecog]% drawspec wav/takagi.wav
```

と入力すると，fb と mfccf でこの音声ファイルを分析し，その結果を音声波形とともに gnuplot を用いて，図 3.23 のように表示します．

この図を PNG ファイルとして保存するときは，

```
[~/asr/wrecog]% drawspec wav/takagi.wav png takagi.png
```

と入力してください[7]．図 3.23 のような出力が得られます．ただし，フィルタバンクの出力値は対数を取っています．PNG ファイルのファイル名は，この例のように「発話内容.png」とします．

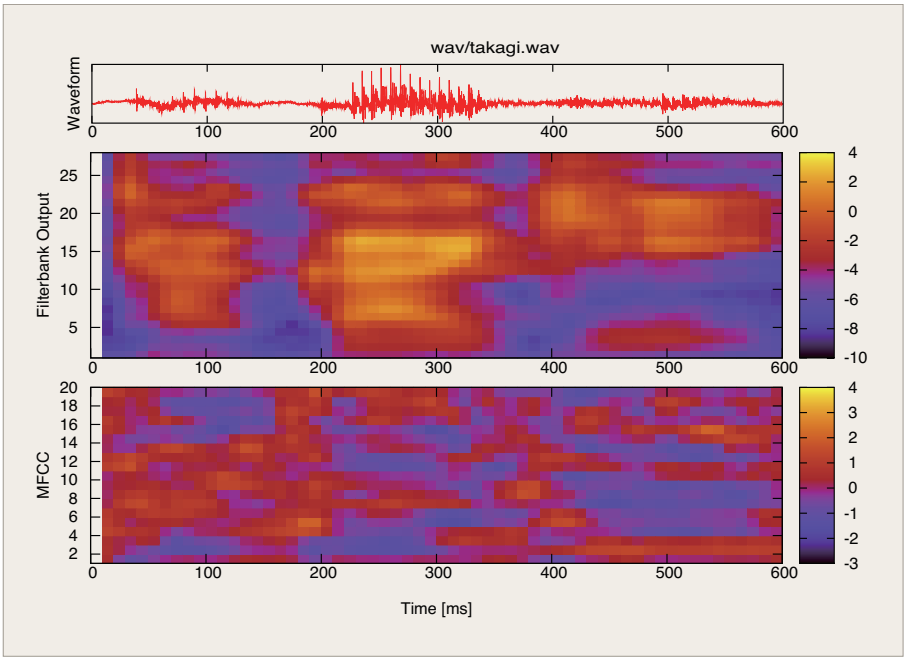


図 3.23: drawspec の出力例．発話は/takagi/．上から，音声波形，フィルタバンク出力，メル周波数ケプストラム係数（MFCC）．横軸は時間 [ms]．

スペクトル分析の結果の図の中段に描かれるフィルタバンク出力に現れる音声のスペクトルパターンを見てみましょう．縦軸はフィルタバンクチャネル番号です．フィルタバンク出力にはメル周波数軸で分析した声道調音フィルタ特性の概形が得られます（3.6 節，3.8 節）．母音のフォルマント（2.2.1 節）に対応するチャネルの出力値は大きいので，明るい色で表示されています．

図 3.23 は/takagi/という音声データの 50ms〜130ms 付近と 210ms〜330ms 付近の母音/a/の区間では，第 6，第 13，および第 17 チャネル付近に明るい領域があります．母音/a/のフォルマントがここに現れているのです．この音声データの 460ms〜580ms 付近は母音/i/です．明るい領域は第 3，第 16，および第 22 チャネル付近にあります．母音/a/とはフォルマントの位置が異なることが分かります．

暗い色で表示されているのはエネルギーが少ない周波数の領域です．無声子音の区間や無音声の区間に対応していることが分かります．

自分の音声データに含まれている音素の種類や位置とフィルタバンク出力のパターンの対応を観察してください．

[7]drawspec はシェルスクリプトです．他の形式で保存したい場合は自分で改造してください．

3.10.4 レポート（第 1 週）

1. 実習課題の目的
2. プログラミングおよびスペクトル分析
   (a) `ad2fb.c` および `fb2mfcc.c` の穴埋め部分.
   (b) 自分の名前の分析結果（図 3.23 の形式の `drawspec` の出力）．第 2 章を参考にして，メルフィルタバンク出力について上記の例のような所見を，できる範囲で良いので，述べてください．メルフィルタバンクのチャネルとその中心周波数の対応は以下のとおりです.

channel	center frequency [Hz]	channel	center frequency [Hz]
1	64	15	1877
2	133	16	2111
3	208	17	2367
4	291	18	2645
5	381	19	2949
6	479	20	3280
7	586	21	3641
8	703	22	4035
9	830	23	4465
10	969	24	4934
11	1120	25	5446
12	1286	26	6004
13	1466	27	6612
14	1663	28	7276

3. 考察
4. 一般事項
   (a) 本日のポイントは何であったか？
   (b) 良くわかったこと
   (c) わからなかったこと
   (d) 要望
   (e) 感想，その他 (USB 接続のヘッドセットを使用した場合はその型番を記録すること)