DNN テキスト音声合成の自然化に向けたアクセント修正 GUI の作成

written by mrK

概要

近年の深層学習の発展により台頭した DNN に基づく統計的音声合成は，日本語の自然なアクセントを完全に再現するのは難しいという問題がある．そこで，音声分析合成システム WORLD を用いて，テキストから合成した音声から音高を司る基本周波数を抽出し，各モーラ (日本語では平仮名に相当) の音高を母音の基本周波数に均一化した際の合成音声をもとの合成音声と比較した．その結果，合成音声のアクセント修正が実現された一方で，モーラ間に生じた不連続性による不自然な接続箇所が見られた．モーラ間の不自然な接続を解消するために Savitzky-Golay 法による基本周波数の平滑化を行い，これにより合成音声の自然化に成功した．また，テキストから音声を合成し，アクセントの修正・平滑化を実装した GUI を作成した．今後は時系列データの学習に有効な RNN を用いたより自然な合成音声を検討する．

1 序論

1.1 本研究の背景

我々人間が生活する上で欠かせないものが言葉であり，言葉を発するためには音声が必要不可欠である．近年では AI の発達により Amazon Echo[1], Google Nest[2] などのスマートスピーカーが普及しているが，スピーカーから発される音声には音声合成技術が用いられている．音声合成技術は 1990 年頃を境に発展を遂げ，ルールベースのフォルマント音声合成からコーパスベースの素片選択型音声合成が主流となったが，現在では深層学習の出現により DNN 音声合成をはじめとする高音質な音声合成が実現されている [3]．特に，統計的音声合成は波形を統計モデル化することで声質を制御しやすい合成手法として有名であるが，人間が話す言葉の自然なアクセントを完全に再現するのは難しい．一方，音声分析合成システムである WORLD[4](D4C edition [5]) は音声波形からパラメータを推定したり，既存のパラメータから音声を合成したりすることが可能である．これらを踏まえて，WORLD を用いて DNN 音声合成の改善を考えたことが本研究の背景である．

1.2 本研究の目的

本研究の目的は，DNN を用いた統計的テキスト音声合成を実装し，テキストから合成した音声を自然なアクセントに修正することである．そのために WORLD を用いてアクセントと密接に関わる基本周波数を抽出し，音高を変化させてからパラメータを再合成する．また，基本周波数を手動で変化させて直感的な音高調節を実現する GUI を作成する．

2　統計的音声合成

2.1　統計モデルに基づく音声合成

統計的音声合成 (Statistical Speech Synthesis) とは，テキストと音声の関係を表す統計モデルからサンプリングすることで音声を合成する手法である (図 2.1)[6]．確率変数の依存関係を表したグラフィカルモデルを図 2.2 に示す．ここで，テキストの集合を W，音声波形の集合を X，未知のテキストを w，統計モデルのパラメータを λ，生成される音声波形を x とする．

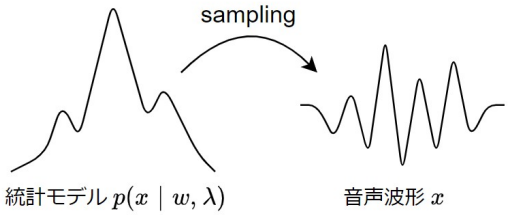


図 2.1: 統計的音声合成におけるサンプリング

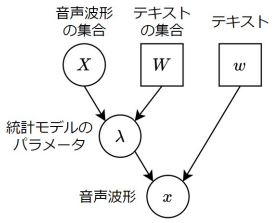


図 2.2: 統計的音声合成のグラフィカルモデル

上記の考え方に基づき，音声波形 x が従う予測分布 p を式 2.1 に示す．

$$
\begin{aligned}
x \sim p(x|w, W, X) &= \int p(x, \lambda|w, W, X) \quad (\because \text{周辺化}) \\
&= \int p(\lambda|w, W, X)\, p(x|w, W, X, \lambda)\, \mathrm{d}\lambda \quad (\because p(a, b) = p(b)\, p(a|b)) \\
&= \int p(\lambda|W, X)\, p(x|w, \lambda)\, \mathrm{d}\lambda \quad (\because \text{図 2.2の依存関係})
\end{aligned} \tag{2.1}
$$

式 2.1 を解析的に解くのは難しいため，点推定を用いて式 2.2 と近似できる．

$$
p(x|w, W, X) \simeq p(x|w, \hat{\lambda}) \quad \text{where} \quad \hat{\lambda} = \underset{\lambda}{\operatorname{argmax}}\, p(\lambda|W, X) \tag{2.2}
$$

近似式 2.2 に基づく音声合成手法を一貫学習といい (図 2.3)，一貫学習を用いた音声の直接的な生成は難しい [6]．

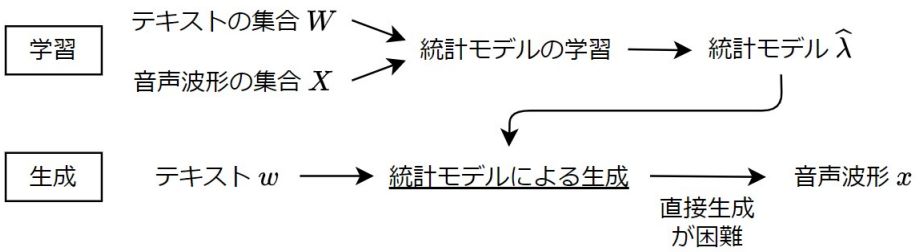


図 2.3: 統計モデルに基づく音声合成のフロー

2.2 統計的パラメトリック音声合成

統計的音声合成において，式 2.2 で表される一貫学習による音声合成は難しいため，部分的な問題に分割して音声合成を容易にした手法を統計的パラメトリック音声合成 (Statistical Parametric Speech Synthesis) という [6]．グラフィカルモデルを図 2.4 に示す．ここで，テキスト，言語特徴量，音響特徴量，音声波形の集合をそれぞれ W, L, O, X，任意のテキスト，言語特徴量，音響特徴量をそれぞれ w, l, o，生成される音声波形を x とする．また，テキストから言語特徴量を予測するモデル，言語特徴量から音響特徴量を予測するモデル，音響特徴量から音声波形を生成するモデルのパラメータをそれぞれ $\lambda_{\mathrm{L}}, \lambda_{\mathrm{A}}, \lambda_{\mathrm{V}}$ とする．

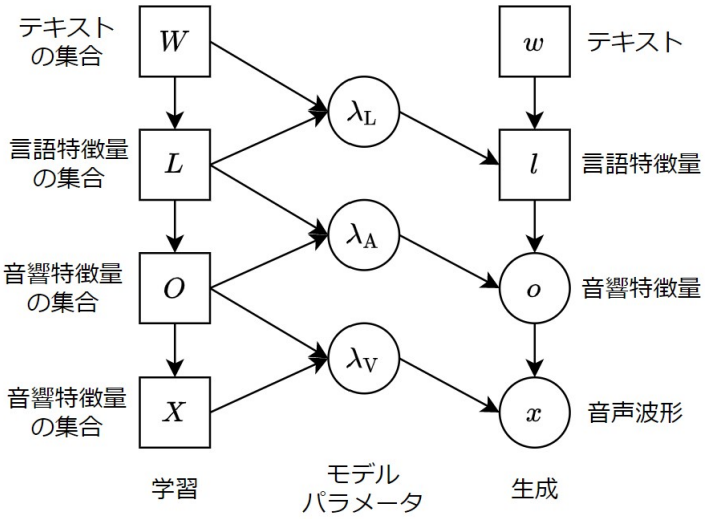


図 2.4: 統計的パラメトリック音声合成のグラフィカルモデル

上記の考え方に基づき，予測分布 p を式 2.3 に示す．

$$p(x, l, o|w, \lambda) = p(l|w, \lambda_{\mathrm{L}}) \cdot p(o|l, \lambda_{\mathrm{A}}) \cdot p(x|o, \lambda_{\mathrm{V}}) \tag{2.3}$$

式 2.3 のうち，特に λ_{A} は音質を左右する重要な要素であるから，以降では λ_{A} のみ考え，$\lambda_{\mathrm{L}}, \lambda_{\mathrm{V}}$ は簡単のため省略する．式 2.2 と同様の考え方により，学習部は

$$\begin{aligned}
\hat{\lambda}_{\mathrm{A}} &= \underset{\lambda_{\mathrm{A}}}{\operatorname{argmax}}\, p(\lambda_{\mathrm{A}}|W, X) \\
&= \underset{\lambda_{\mathrm{A}}}{\operatorname{argmax}}\, \frac{p(\lambda_{\mathrm{A}}, X|W)}{p(X)} \\
&= \underset{\lambda_{\mathrm{A}}}{\operatorname{argmax}}\, p(\lambda_{\mathrm{A}}, X|W) \quad (\because p(X)\text{は}\lambda_{\mathrm{A}}\text{に非依存}) \\
&= \underset{\lambda_{\mathrm{A}}}{\operatorname{argmax}}\, p(\lambda_{\mathrm{A}}) p(X|W, \lambda_{\mathrm{A}}) \\
&= \underset{\lambda_{\mathrm{A}}}{\operatorname{argmax}}\, p(\lambda_{\mathrm{A}}) \cdot \int \sum_{L} p(L|W) p(O|L, \lambda_{\mathrm{A}}) p(X|O)\, \mathrm{d}O
\end{aligned} \tag{2.4}$$

また，生成部は

$$
\begin{aligned}
x \sim p(x|w, \hat{\lambda}_{\mathrm{A}}) &= \int \sum_{l} p(x, l, o|w, \hat{\lambda}_{\mathrm{A}}) \, \mathrm{d}o \\
&= \int \sum_{l} p(l|w) p(o|l, \hat{\lambda}_{\mathrm{A}}) p(x|o) \, \mathrm{d}o
\end{aligned} \tag{2.5}
$$

式 2.4, 2.5 に含まれる積分を解くことは難しいため，点推定を行うことで近似的に解くことができる．学習部は

$$
\left\{ \hat{\lambda}_{\mathrm{A}}, \hat{O}, \hat{L} \right\} = \left\{ \underset{\lambda_{\mathrm{A}}, O, L}{\operatorname{argmax}} \, p(\lambda_{\mathrm{A}}) p(L|W) p(O|L, \lambda_{\mathrm{A}}) p(X|O) \right\}
$$

$$
\therefore \begin{cases}
\hat{L} = \underset{L}{\operatorname{argmax}} \, p(L|W) \\
\hat{O} = \underset{O}{\operatorname{argmax}} \, p(X|O) \\
\hat{\lambda}_{\mathrm{A}} = \underset{\lambda}{\operatorname{argmax}} \, p(\hat{O}|\hat{L}, \lambda_{\mathrm{A}})
\end{cases}
$$

であり，生成部は

$$
\left\{ \hat{x}, \hat{o}, \hat{l} \right\} = \left\{ \underset{x, o, l}{\operatorname{argmax}} \, p(l|w) p(o|l, \lambda_{\mathrm{A}}) p(x|o) \right\}
$$

$$
\therefore \begin{cases}
\hat{l} = \underset{l}{\operatorname{argmax}} \, p(l|w) \\
\hat{o} = \underset{o}{\operatorname{argmax}} \, p(o|\hat{l}, \hat{\lambda}_{\mathrm{A}}) \\
\hat{x} = \underset{x}{\operatorname{argmax}} \, p(x|\hat{o})
\end{cases}
$$

と表せる．統計的パラメトリック音声合成のフローを図 2.5 に示す．

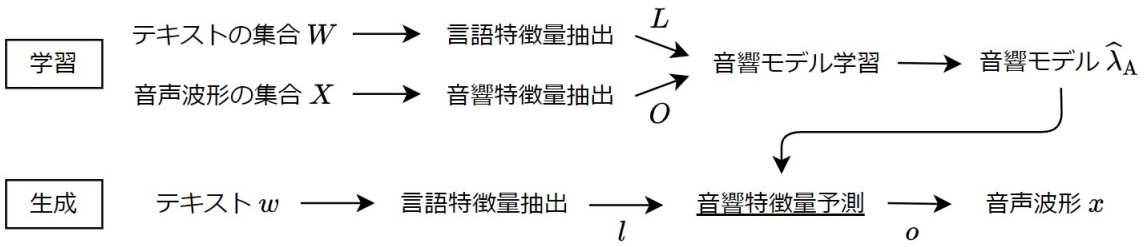


図 2.5: 統計的パラメトリック音声合成のフロー

3　DNN 音声合成

3.1　DNN テキスト音声合成

DNN テキスト音声合成とは，統計的パラメトリック音声合成の音響モデル $\hat{\lambda}_{\mathrm{A}}$ に DNN(Deep Neural Network) を用いる手法である．ただし，DNN 音声合成では，系列長が異なる特徴量間の変換を直接扱うのは難しいため，言語構成単位の言語特徴量からフレーム単位の言語特徴量への変換と，フレーム単位の言語特徴量から音響特徴量への変換の 2 つの手順を DNN でモデル化する必要がある [6]．前者の変換モデルを継続長モデル，後者の変換モデルを音響モデルという．DNN 音声合成のフローを図 3.1 に示す．ここで，図 3.1 中の赤色部分はその実装に DNN を用いることを意味する．

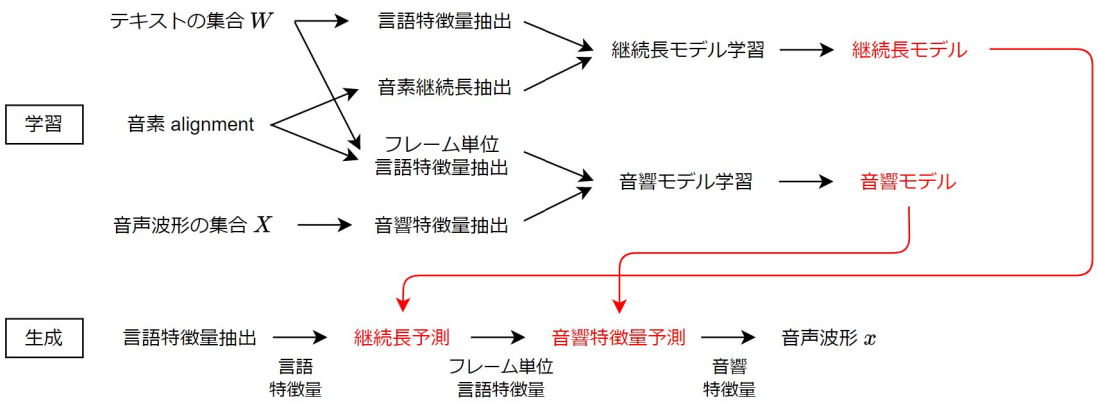


図 3.1: DNN 音声合成のフロー

3.2　言語特徴量の抽出

言語特徴量とは，音声言語の最小単位である音素 (e.g. /a/, /k/) や，母音を中心とした音の塊であるモーラ (e.g.「あ」,「か」) などの音声情報を数値化したものである．言語特徴量の抽出フローを図 3.2 に示す．

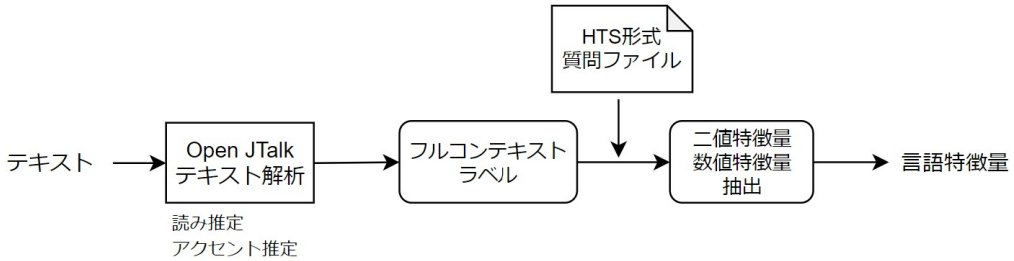


図 3.2: 言語特徴量の抽出フロー

まず，HMM(Hidden Markov Model) 音声合成に基づく日本語音声合成システム Open JTalk[7] を用いてフルコンテキストラベルを抽出する．フルコンテキストラベルは音素単位でその音節，語，アクセントなどに関する情報が正規表現で記述されている [8]．ただし，フルコンテキストラベルには音素アラインメント (各音素の開始・終了時刻の集合) が含まれていないため，汎用大語彙連続音声認識エンジン Julius[11] の音素セグメンテーションキットを用いて音素アラインメントをラベルに付加する必要がある．次に HMM/DNN 音声合成ソフトウェア HTS[9] で用いられる形式の質問ファイルを用いて，フルコンテキストラベルから二値特徴量と数値特徴量を抽出する．ここで，二値特徴量は「該当音素は/A/か」などの True/False で回答される質問により得られる．一方，数値特徴量は「アクセント核と当該モーラの位置の差」などの整数で回答される質問により得られる．最後に，HTS および DNN 音声合成のツールキット Merlin[10] を用いて，先程得られた特徴量を数値表現にすることで言語特徴量を抽出する．

3.3 音響特徴量の抽出

音響特徴量とは，WORLD ボコーダを用いて音声波形から得られる音声パラメータ表現を結合したものである．音声特徴量の抽出フローを図 3.3 に示す．

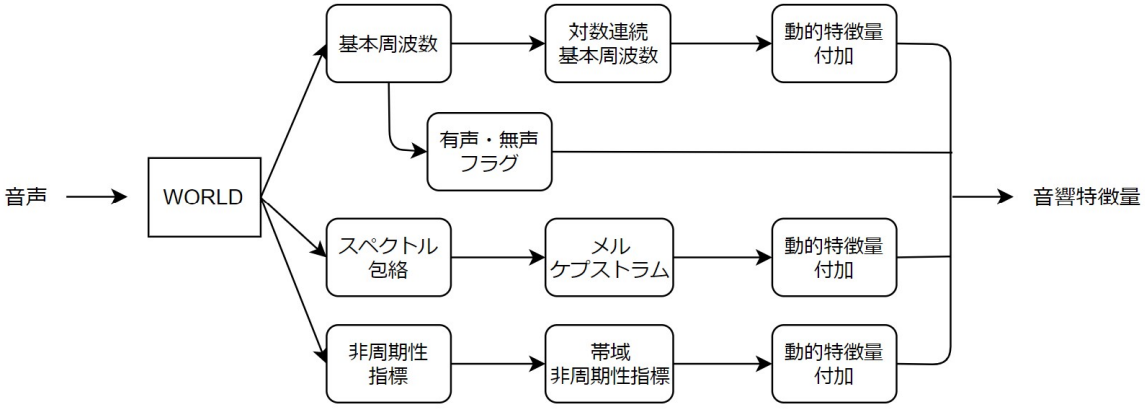


図 3.3: 音声特徴量の抽出フロー

3.3.1 ソース・フィルタモデル

人間が発する音声の生成過程にソース・フィルタモデル [3, 6] という考え方がある (図 3.4)．これは，声帯により生成される音源 (ソース) が声道を通ることで音色付けされる (フィルタされる) という考え方である．まず，声帯を開閉させて空気を振動させることで音高を生成する．このときの声門開閉の周期の逆数を基本周波数という．次に，ソースが口から発されるまでの間に声道を通過することで声の音色が付加される．この声質を決定する声道フィルタの振幅特性をスペクトル包絡という．この 2 つのパラメータの畳み込みによって音声の周波数特性が得られる．また，声道振動の揺らぎなどの非周期的な雑音の占める割合を表す非周期性指標も生成される音声に関わる．

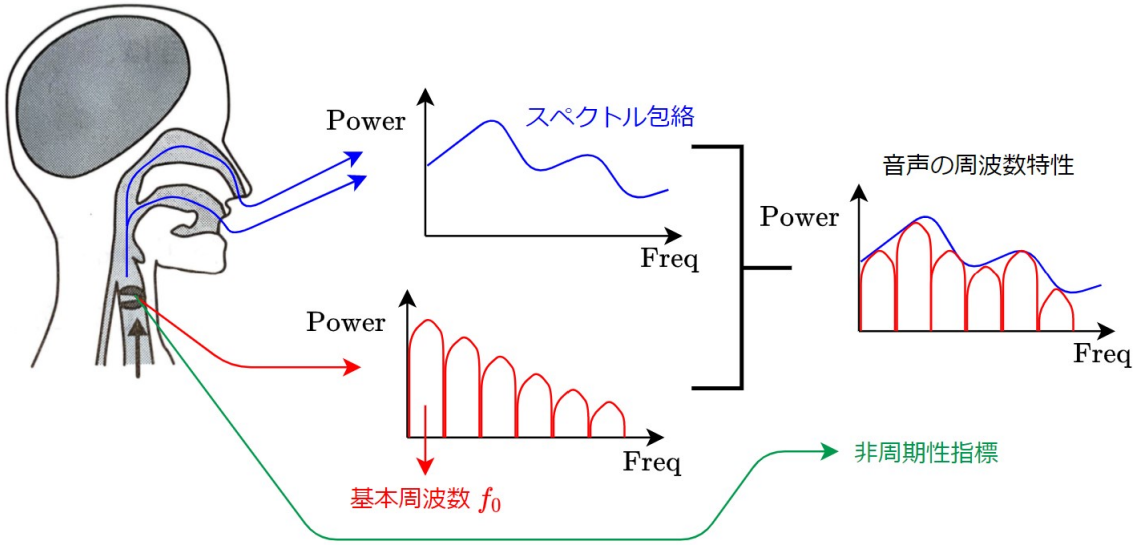


図 3.4: ソース・フィルタモデルと音声パラメータ

3.3.2 対数連続基本周波数

WORLD の基本周波数推定法 DIO を用いて基本周波数を抽出した後，人間の音高知覚が対数的であることを利用して対数化する．ただし，無声音における基本周波数は 0Hz であり，$\log 0$ は未定義であることに注意する．得られた対数基本周波数は無声区間で非連続であるが，その非連続性を DNN 音声合成で扱うのは難しい．そこで有声区間の対数基本周波数を用いて無声区間の対数基本周波数を線形補間する手法が採られる [12]．これにより対数連続基本周波数が求まる．

3.3.3 メルケプストラム

メルケプストラムとは，スペクトル包絡を人間の音高知覚に基づいたメル尺度で近似するパラメータ表現である．WORLD のスペクトル包絡推定法 CheapTrick を用いてスペクトル包絡を抽出した後，周波数軸をメル周波数軸に伸縮したものをケプストラム (フーリエ変換の絶対値の自然対数を逆フーリエ変換したもの) に変換することで得られる．メルケプストラムはスペクトル包絡の次元を約 $\frac{1}{17}$ に圧縮したパラメータ [6] であるため，情報を保持したままの次元数削減に有効である．

3.3.4 帯域非周期性指標

WORLD では，群遅延 (位相特性の 1 回微分) に基づく非周期性指標推定法 D4C が実装されている．D4C は，15kHz 未満の帯域に対して 3kHz ごとに帯域別の非周期性指標を計算し，スペクトル包絡と同じ次元数に展開する [6]．展開したものを帯域ごとの非周期性指標に再圧縮することで帯域非周期性指標が得られる．

3.3.5 動的特徴量

音響特徴量は前後のコンテキストの影響を変化するため，静的特徴量の時間変化を表す動的特徴量を求める必要がある．音声特徴量系列を $o = \{o_1, o_2, \cdots, o_M\}$ とすると，m 番目のフレームにおける 1 次の動的特徴量 Δo_m，2 次の動的特徴量 $\Delta^2 o_m$ は式 3.1, 3.2 で表される [6].

$$\Delta o_m = -0.5 o_{m-1} + 0.5 o_{m+1} \tag{3.1}$$

$$\Delta^2 o_m = o_{m-1} - 2 o_m + o_{m+1} \tag{3.2}$$

上式を用いて，連続対数基本周波数，メルケプストラム，帯域非周期性指標の 3 つの音声パラメータに対して動的特徴量を求める．

3.4 合成音声の生成

DNN により学習させた継続長モデルと音響モデルを用いて，テキストから合成音声を生成することが可能となる．合成音声の生成フローを図 3.5 に示す．図 3.5 は「現実」というテキストから音声を合成する際の手順を表す．テキストを音素に分割した後，継続長モデルにより音素継続長を推定し，フレーム単位に拡張してから音響モデルにより音響特徴量を推定する．ここで推定された音響特徴量は次元数削減のために変形したものであるから，もとのパラメータ表現に戻す必要がある．このアルゴリズムを MLPG(Maximum Likelihood Parameter Generation) といい，式 3.3 で表される [13]．ここで，o は音響特徴量行列，O は Δ, Δ^2 を含む音響特徴量行列，W は式 3.1, 3.2 に基づく o から Δ, Δ^2 を含む行列への変換行列，Σ は Δ, Δ^2 の共分散行列である．これより計算したパラメータを WORLD により合成することで音声が生成できる．

$$o = \left\{ \left(W^\top \Sigma^{-1} W \right)^{-1} W^\top \Sigma^{-1} \right\} O \tag{3.3}$$

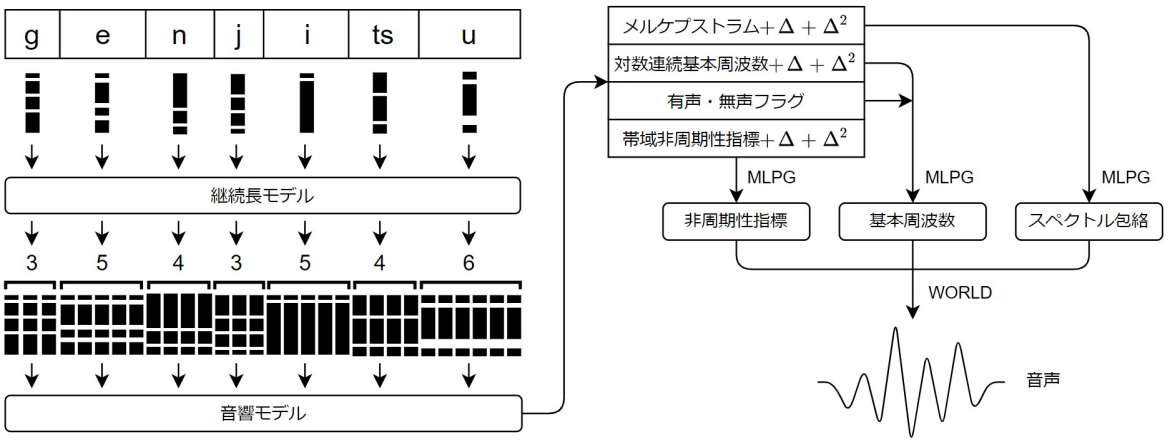


図 3.5: 合成音声の生成フロー

4　DNN を用いた音声合成実験

4.1　実験条件

本研究では，継続長モデルと音響モデルの学習用コーパスに JSUT(Japanese speech corpus of Saruwatari-lab., University of Tokyo)[14] の basic5000 を用いた．ただし，JSUT コーパスには音素アラインメントが同梱されていないため，Julius を用いて推定した音素アラインメントを利用した．また，basic5000 に含まれる 5000 文のうち，Open JTalk の読み推定に明らかな誤りがある発話が 1567 文存在したため，それらはフルコンテキストラベルに誤りがあるとみなし学習データから排除した．本研究における音声合成の実験条件を表 4.1 に示す．

表 4.1: 音声合成の実験条件

統計モデル		
学習音声コーパス	JSUT コーパス basic5000 女性話者 1 名 日本語テキスト発声 3433 文	
サンプリング周波数	48kHz (学習時は 16kHz にダウンサンプリング)	
学習データ	3133 文	
分析条件	サンプリング周波数: 16kHz，フレームシフト: 5ms， メルケプストラム: 40 次元，正規化手法: 標準化	
DNN 構造		
	継続長モデル	音響モデル
入力層	325 ノード	329 ノード
中間層	64 ノード × 2 層	256 ノード × 2 層
出力層	1 ノード	127 ノード
学習条件		
最適化手法	Adam	
学習係数	0.001	
バッチサイズ	32	
エポック数	30	
学習係数の減衰係数	0.5	
学習係数の減衰間隔	10 イテレーション	
検証条件		
検証データ	200 文	
評価条件		
評価データ	100 文	

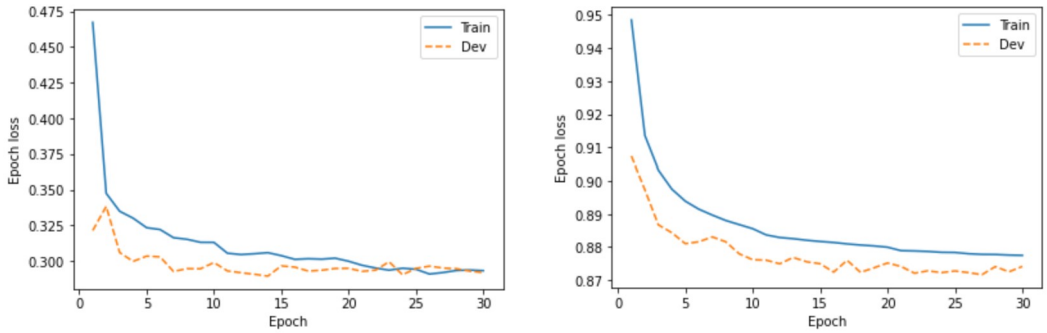
4.2 実験方法

表 4.1 の条件下で，図 3.1 に基づいて継続長モデルおよび音響モデルを学習した際の平均二乗誤差 (MSE: Mean Squared Error) の推移を観察した．ただし，MSE は式 4.1 で表される誤差関数であり，y は正解値，$\hat{y}$ は予測値である．その後，学習した継続長モデルおよび音響モデルを用いて，例として評価データに含まれる発話 BASIC5000_5000「あと 30 分の猶予が与えられた」に対して，基本周波数，スペクトル包絡，非周期性指標を推定し，それらを合成した音声のスペクトログラムを自然音声と比較した．

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{4.1}$$

4.3 実験結果

継続長モデルおよび音響モデルを学習した際の MSE の推移を図 4.1 に示す．また，発話 BASIC5000_5000 に対して推定した基本周波数，スペクトル包絡，非周期性指標をそれぞれ図 4.2, 4.3, 4.4 に示す．さらに，WORLD を用いてパラメータから合成した音声のスペクトログラムを図 4.5 に示す．



(a) 継続長モデル学習時の MSE の推移　　**(b)** 音響モデル学習時の MSE の推移

図 4.1: 統計モデル学習時の MSE の推移

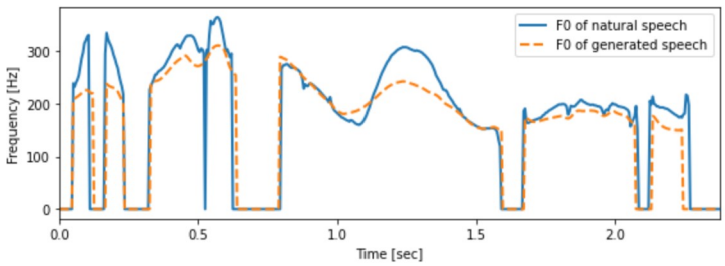


図 4.2:「あと 30 分の猶予が与えられた」に対する基本周波数の正解値と推定値の比較

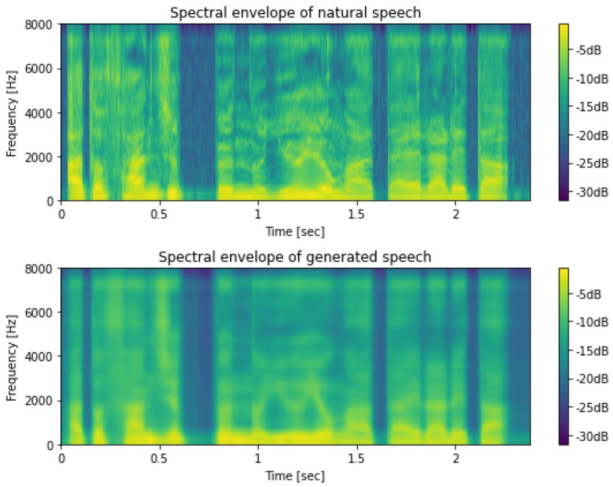


図 4.3:「あと 30 分の猶予が与えられた」に対するスペクトル包絡の正解値と推定値の比較

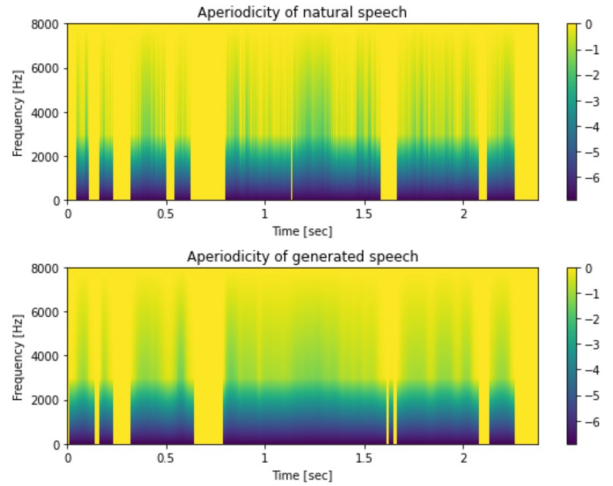


図 4.4:「あと 30 分の猶予が与えられた」に対する非周期性指標の正解値と推定値の比較

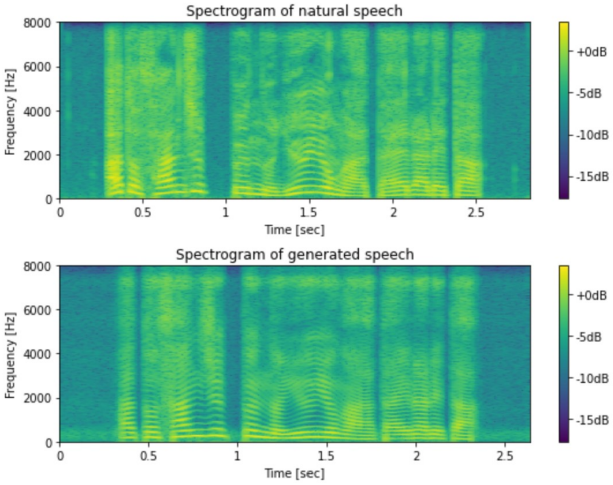


図 4.5:「あと 30 分の猶予が与えられた」に対する自然音声と合成音声のスペクトログラムの比較

4.4 考察

図 4.1 より，統計モデルの学習時の MSE はエポックが進行するにつれて減少したため，モデルを上手く学習できたと考えられる．図 4.2 において，基本周波数の推定値は正解値より小さいことが分かるが，全体的な傾向は再現できたと考えられる．図 4.3，4.4 において，推定値は正解値の概形を捉えているが特に高周波数帯でぼやけていることが分かる．これより，推定値は過剰な平滑化により微細な構造を失っていると考えられる．図 4.5 より，合成された音声のスペクトログラムは過剰な平滑化の影響を受けたが，概形は自然音声のものと類似していることが分かる．合成音声は自然音声と比較して抑揚が若干失われていたが，日本語として十分自然な音声であった．抑揚の消失は先述した基本周波数の縮小に起因すると考えられる．

5 合成音声の自然化に向けたアクセント修正

5.1 予備実験

5.1.1 実験方法

日本語は「母音のみ」,「子音 + 母音」，あるいは「ん」からなるモーラ (平仮名) の組み合わせで出来ており，単語のアクセントは母音または「ん」で決定される．この考えに基づき，モーラの基本周波数をモーラに含まれる母音の基本周波数で均一化しても音声は大きく変化しないという仮説を立て，予備実験を行った．例として，発話 BASIC5000_5000「あと 30 分の猶予が与えられた」の基本周波数を WORLD により抽出し，各モーラの基本周波数を母音の音素/a/, /i/, /u/, /e/, /o/または「ん」を表す音素/N/の基本周波数で均一化した後，WORLD により音声を再合成した．ただし，本実験では WORLD の基本周波数推定法 DIO の代わりに，時間はかかるが高精度な推定法である Harvest を用いた．

5.1.2 実験結果と考察

発話 BASIC5000_5000 に対して，各モーラの基本周波数を/a/, /i/, /u/, /e/, /o/, /N/の基本周波数で均一化した後の基本周波数を図 5.1 に示す．図 4.2 が示す合成音声の基本周波数の推定値は曲線を描いた一方，図 5.1 では例えば「と」に相当する音素/to/が 230Hz で，「じゅ」に相当する音素/ju/が 280Hz で均一化された．

再合成後の音声は滑らかさが少し失われたが，声質やアクセントの概形は変化しなかった．再合成後の音声における発話の滑らかさが少々失われた理由として，モーラの音高を強制的に音素/a/, /i/, /u/, /e/, /o/, /N/の音高に統一したために，モーラ間の連続性が失われたからであると考えられる．

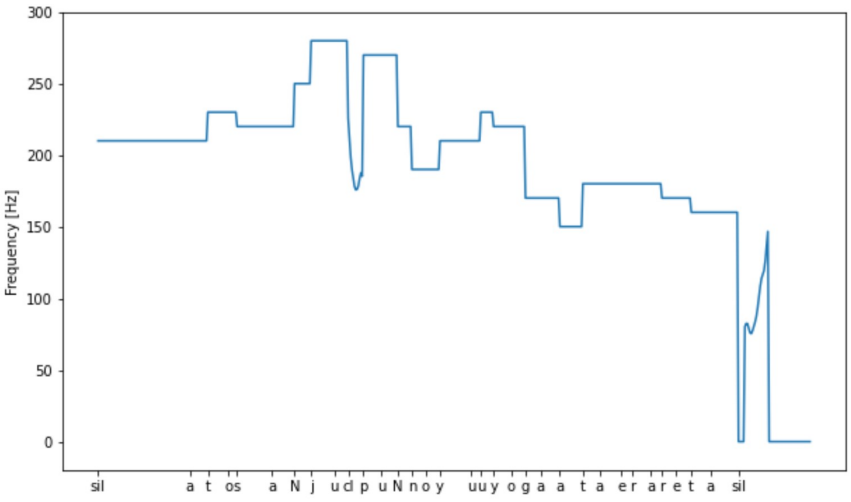


図 5.1:「あと 30 分の猶予が与えられた」の音高統一後の基本周波数

5.2 実験

5.2.1 実験方法

任意のテキストに対して DNN に基づく合成音声を生成し，予備実験と同様に各モーラの基本周波数を/a/, /i/, /u/, /e/, /o/, /N/の基本周波数で均一化した．その後，Python の GUI ライブラリである PySimpleGUI[15] を用いて，各モーラの基本周波数を手動で修正できるようなスライダーをモーラごとに割り当て，修正後にボタンを押すことで基本周波数を修正した後の音声を合成可能な GUI を作成した．

5.2.2 実験結果

作成した GUI の各ウィジェットの役割を表 5.1 に，概形を図 5.2 に示す．また，「テキスト音声合成」というテキストを入力し，アクセントを修正する前後の基本周波数を図 5.3 に示す．

表 5.1: アクセント修正 GUI の各要素の役割

ウィジェットの種類	説明テキスト	役割
TextBox	Text you want to synthesize:	合成対象のテキスト
Button	Extract Features	基本周波数の抽出
Slider	—	モーラの基本周波数の修正
Button	Play Original	DNN に基づく合成音声再生
Button	Play Modified	修正後の基本周波数での合成音声再生
Button	Save Modified	修正後の基本周波数での合成音声保存

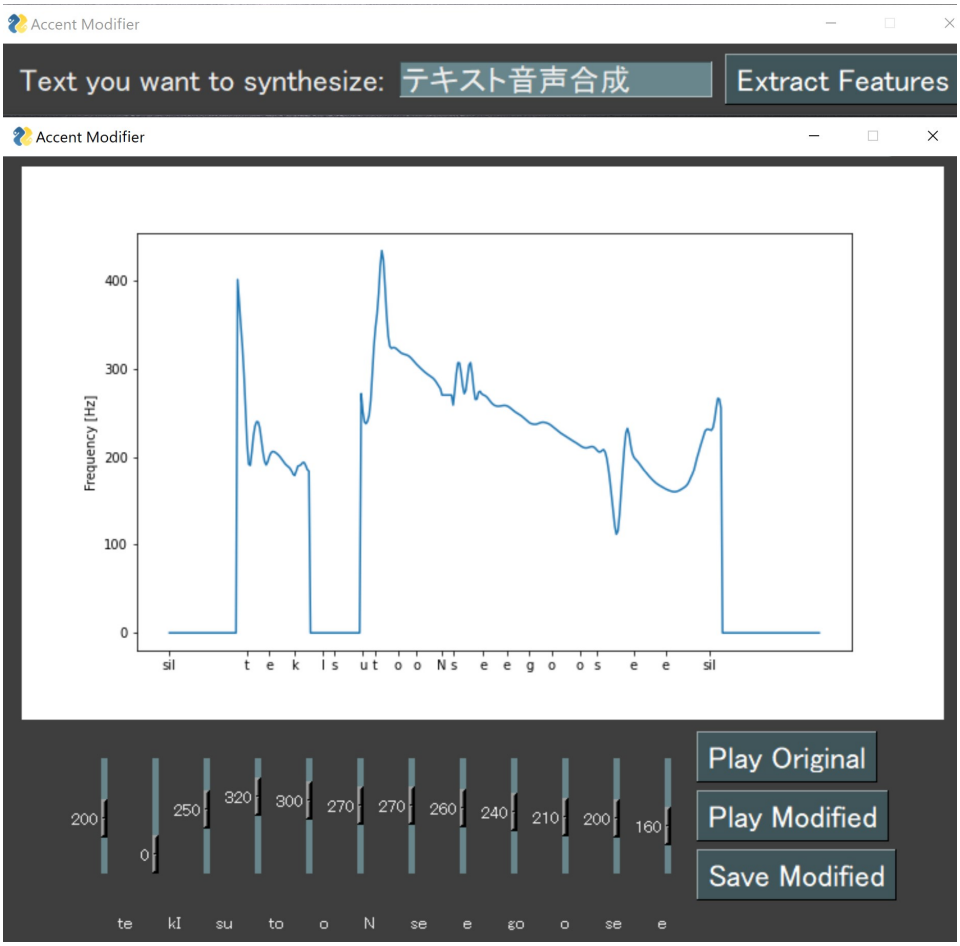
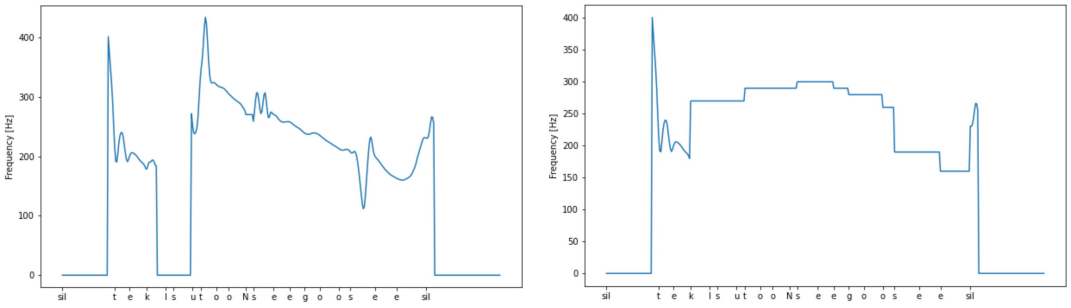


図 5.2: アクセント修正 GUI の概形



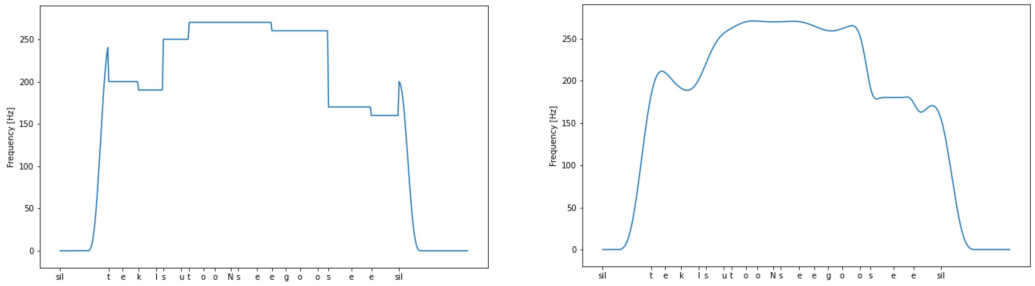
(a) 調節前の基本周波数　　**(b)** 調節後の基本周波数

図 5.3: アクセント修正前後の基本周波数

5.2.3 考察

アクセント修正 GUI のスライダーを手動で操作することで，図 5.3a を図 5.3b に変換できた．これによりアクセント修正を施した合成音声を得られた．非音声区間/sil/の基本周波数は合成音声に影響しないため，値が有声区間と大きく離れていても音質とは無関係であると考えられる．

また，発話のアクセント修正は実現できたが，モーラ間の連続性が失われたために不自然な音声接続箇所が存在した．そこで，同様の手法でアクセントを修正した後，修正後の基本周波数を平滑化することを考える．本考察では時系列データを平滑化する手法として Savitzky-Golay(SG) 法 [16] を用いた．ただし，窓の長さは 11，多項式次数は 3 とした．なお，基本周波数に SG フィルタを適用するボタン Smoothing を GUI に追加実装した．「テキスト音声合成」というテキストのアクセントを修正した直後と，SG フィルタを 10 回適用した後の基本周波数の比較を図 5.4 に示す．図 5.4a で見られるモーラ間の不連続性が，図 5.4b では平滑化されて解消されたことが分かる．これに伴い，合成音声に見られた不自然な発話箇所が消え，自然な音声へと変換された．



(a) 平滑化前の基本周波数　　**(b)** 平滑化後の基本周波数

図 5.4: 平滑化前後の基本周波数の比較

6 結論

本研究では，DNN を用いた統計的テキスト音声合成を実装した．また，テキストから合成した音声から WORLD を用いて基本周波数を抽出・修正することで，再合成後の音声のアクセント修正を実現した．さらに，アクセント修正時に生じたモーラ間の不連続性を解消するために Savitzky-Golay 法による基本周波数の平滑化を行い，自然な音声を生成することに成功した．今後は，時系列データに扱う際に最適な RNN(Recurrent Neural Network) を用いるとより自然な合成音声が期待されるので，その検討の余地がある．

参考文献

[1] Amazon, ”Echo & Alexa 通販”, https://www.amazon.co.jp/b?ie=UTF8&node=5364343051, 2022 (viewed 2022/01/23)

[2] Google, ”Google Nest のスマートスピーカーとディスプレイ”, https://store.google.com/jp/magazine/compare_nest_speakers_displays?hl=ja, 2022 (viewed 2022/01/23)

[3] 猿渡洋・高道慎之介, ” 音声合成・変換 その 1”, 東京大学大学院 2018 年度信号処理論特論第 7 回講義資料, https://www.sp.ipc.i.u-tokyo.ac.jp/~saruwatari/SP-Grad2018_07.pdf, 2018 (viewed 2022/01/23)

[4] M. Morise, F. Yokomori, and K. Ozawa, ”WORLD: a vocoder-based high-quality speech synthesis system for real-time applications,” IEICE transactions on information and systems, vol.E99-D, no.7, pp.1877-1884, 2016

[5] M. Morise, ”D4C, a band-aperiodicity estimator for high-quality speech synthesis,” Speech Communication, vol.84, pp.57-65, Nov. 2016

[6] 山本龍一・高道慎之介, Python で学ぶ音声合成, インプレス社, 2021

[7] Open Jtalk, ”Open Jtalk - HMM-based Text-to-Speech System -”, https://open-jtalk.sp.nitech.ac.jp/, 2018 (viewed 2022/01/23)

[8] 山本龍一, ” 日本語のフルコンテキストラベルの形式”, https://github.com/r9y9/ttslearn/blob/master/docs/appendix.pdf, 2021 (viewed 2022/01/23)

[9] H. working group, ”HMM/DNN-based Speech Synthesis System (HTS)”, http://hts.sp.nitech.ac.jp/, 2021 (viewed 2022/01/23)

[10] The Centre for Speech Technology Research (CSTR), ”Merlin: The Neural Network (NN) based Speech Synthesis System”, https://github.com/CSTR-Edinburgh/merlin, 2019 (viewed 2022/01/23)

[11] Akinobu Lee, et al., ”Julius: Open-Source Large Vocabulary Continuous Speech Recognition Engine”, https://github.com/julius-speech/julius, 2021 (viewed 2022/01/23)

[12] K. Yu and S. Young, ”Continuous F0 Modelling for HMM based Statistical Parametric Speech Synthesis”, IEEE Transactions on Audio, Speech, and Language Processing, vol.19, no.5, pp.4, 2011

[13] @ryo_he_0, Qiita, ”MLPG: Maximum Likelihood Parameter Generation とは”, https://qiita.com/ryo_he_0/items/49e009f207fb36ba7528, 2021 (viewed 2022/01/23)

[14] R. Sonobe, S. Takamichi and H. Saruwatari, ”JSUT corpus: free large-scale Japanese speech corpus for end-to-end speech synthesis,” arXiv preprint, 1711.00354, 2017.

[15] PySimpleGUI Tech LLC, ”PySimpleGUI”, https://pysimplegui.readthedocs.io/en/latest/, 2022 (viewed 2022/01/23)

[16] A. Savitzky, M. J. E. Golay, Smoothing and Differentiation of Data by Simplified Least Squares Procedures. Analytical Chemistry, 36 (8), pp.1627-1639, 1964