TR-IT-0120

フレーム同期型 SSS-LR における アクセント句境界尤度の利用

Use of Accent Phrase Boundary Likelihood in a Frame Synchronous SSS-LR

中井 満 Mitsuru Nakai
シンガー ハラルド Harald Singer
匂坂芳典 Yoshinori Sagisaka

1995.06.30

概要

従来のフレーム同期型 SSS-LR では同一の音素履歴における異なる構文仮説は同じスコアを共有していた。そこで、本報告では、アクセント句境界尤度計算、および、SSS-LR による連続音声認識をフレーム同期に処理することによって、構文仮説を制御する手法を提案する。

句境界尤度の計算は One Stage DP によるテンプレート整合法を用いてフレーム毎に句境界仮説を立て、100ms 窓幅で仮説を検証することで、スコアを出力する。

フレーム同期型 SSS-LR 認識部では、音素履歴の LR 状態がアクセント句に還元するアクションとそれ以外のアクションの両方をもつときには、履歴を2分割し、アクセント句に還元する側に句境界尤度を与える。

評価用テストセット SL2 を用いた実験では音素認識率、単語認識率ともに向上する結果が得られた。

目次

図目次

表目次

第 1 章

序論

1　研究の背景と目的

連続音声の認識は音韻情報により認識された音素列を文法規則によって文節、文章を構成し、一意化することである。近年、連続音声認識の対象は自然な対話へと移行しつつあるが、より自然な対話は発話スタイルや発話速度の種類が多様になり、これまでの音響処理系や言語解析系では認識が非常に困難になるものと推測される。

そこで我々は言語解析系での負荷を低くするためにはアクセント句境界情報等の支援が不可欠であると考え、これまでに韻律特徴量から句境界位置の推定を行なう手法について報告してきた [1]。

本報告では従来の連続音声認識の処理系に韻律処理系を加えたシステムについて検討する。

2　句境界情報を利用した連続音声認識

韻律情報を用いた連続音声認識や理解についての研究は数多く行なわれており、構文推定 [2]、単語検出 [3] などにおいて有効であることが報告されている。いずれの報告においても、韻律処理の基本はアクセント句境界や文節境界の推定である。

2.1　アクセント句境界の検出

韻律構造を利用したアクセント句境界の検出ではアクセント句境界が F_0 パターン (基本周波数パターン) の谷間として比較的明瞭に現れるという理由から、これまでに F_0 パターンを用いた様々なアプローチが試みられている。例えば、局所的な特徴 (F_0 パターンの谷間、F_0 パターンの局所変化、境界の時間的な間隔など) を数量化して直接的に句境界を推定する手法 [4][5] や F_0 パターンの生成モデルに基づく分析合成手法 [6] などが提案されている。

我々はアクセント句境界の検出をアクセントパターン列の認識に置き換えた間接的な検出法である「F_0 パターン連続整合法」を提案した [7][8]。この手法は、アクセント句の F_0 パターンの形状は少数個のクラスタに分類できるという仮定、並びに一つの発話はクラスタの代表パターン (テンプレート) の連接で表現されるという仮定に基礎をおいている。類似した手法に F_0 パターンをアクセント型別に HMM で表現する高橋らの手法 [9] があり、これはアクセント型に関する知識を分類基準として与える手法である。いずれの手法も F_0 パターンに関する生成モデルを必要としないという特徴があり、観測された F_0 パターンによるデータ駆動型の手法である。一方、F_0 生成モデルを用いた手法には、AbS 合成による分析に基づくパラメータの推定を補助的に用いた今野らの句境界検出法 [10] があり、句境界の仮説を

立てた上で指令を発生させ、生成したF_0パターンと観測されたF_0パターンの比較から仮説を検証している。また、我々もF_0パターン連続整合法を発展させ、F_0テンプレートをF_0生成モデルの指令で表現することにより、F_0推定誤りの影響を受けにくいテンプレート、F_0生成モデルを考慮したパタン連続整合を提案した。しかし、これらの手法ではアクセント句境界を二値で表現し、ある程度の境界のずれを容認しているため、検出結果をそのまま音声認識に使用するのは困難であった。

それに対し、アクセント句境界らしさ (句境界尤度) を確率的に表現する試みがいくつか行なわれている。花沢らの手法 [11] は、我々の手法と同様にアクセント句をクラスタで学習する手法であるが、テンプレートをクラスタ毎にHMMで学習することによって句境界尤度を数値化している。また、大川らの手法 [12][3] ではフレーム毎に韻律特徴ベクトルをVQコード化し、句境界との関係を統計的に扱うことにより、確率的に表現している。だが、これらの手法では韻律特徴抽出や句境界尤度計算がフレーム同期で処理できないため、プリプロセッサ、あるいはポストプロセッサとして利用されている。

そこで本報告ではフレーム同期型SSS-LRを認識システムの対象とし、フレーム同期に句境界尤度を供給する手法について検討する。

2.2 フレーム同期型連続音声認識

SSS-LR 連続音声認識システムをフレーム同期により効果的に探索する手法については門前らの報告 [13] がある。フレーム同期型の手法は音素同期型とは異なり、文法的な条件を考慮せず、入力音声のフレームの終了とともに探索が終了するため、最終時刻における音素系列が文法的に受理されるという保証はない。そこで本報告では句境界尤度によってLR構文解析の状態を制御する手法について検討する。

第 2 章

アクセント句境界尤度を用いたフレーム同期型連続音声認識

1　システム概略

図 2.1にフレーム同期システムの概略を示す。12kHz の入力音声を 10ms(120 ポイント) 毎のパワーの大きさによりポーズ区間を除去し、発声の開始を検出する。このとき、最初の 120 ポイントを第 0 フレームとし、中心ポイント時刻を第 0 フレームの時刻とする。ピッチ推定 (512 ポイント FFT) では窓幅の半分 (256 ポイント) が入力された時点で第 0 フレームが計算され、以降、10ms 間隔でシフトする。また、デルタピッチ計算は 200ms の三角窓を使用し、窓幅の半分である 10 フレーム相当のピッチが推定された時点から計算を開始する。句境界の推定にはデルタピッチを使用し、フレーム毎に句境界仮説を立てながら 100ms の遅延で句境界尤度を出力する。このとき、最初 (0 番目) の出力は第 0 フレームの始端時刻における句境界尤度となる。一方、句境界尤度計算のプロセスと並行して、音響パラメータ計

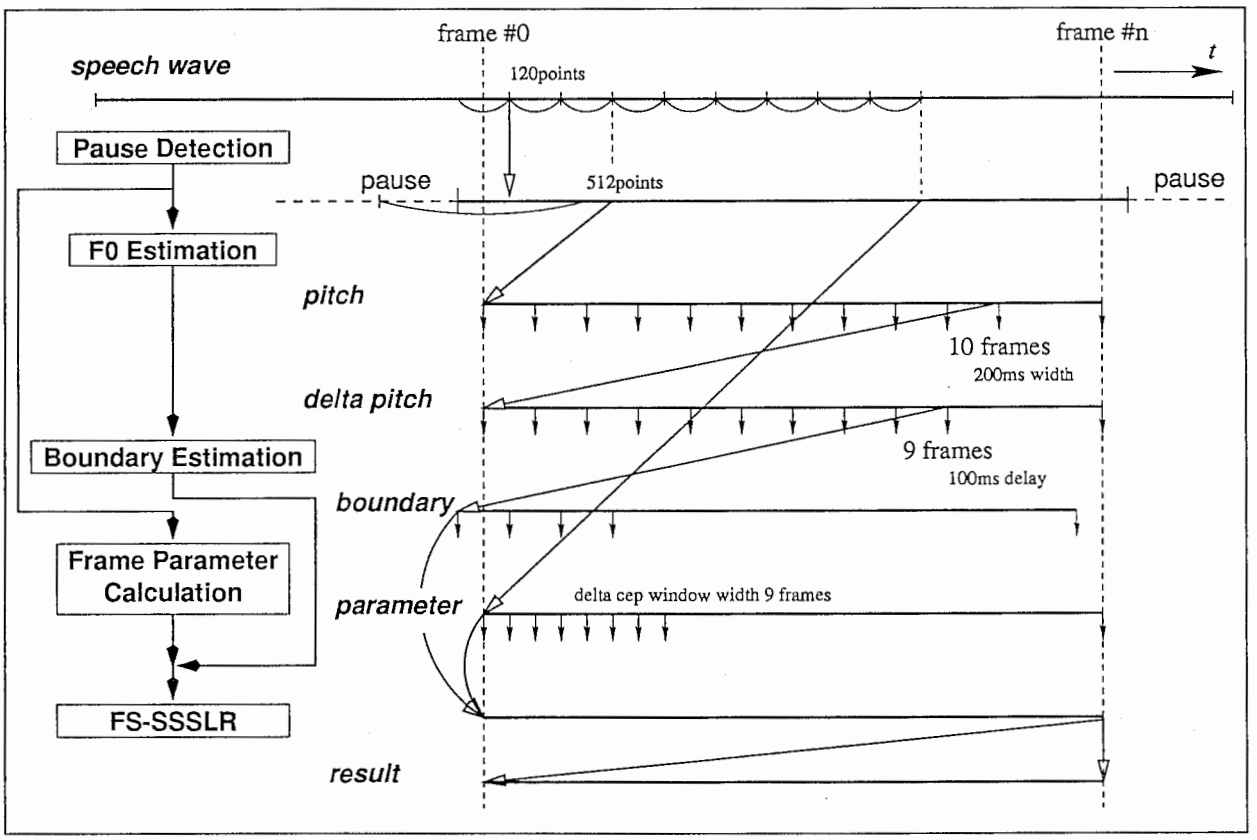

図 2.1: 句境界尤度計算部と連続音声認識部のフレーム同期

算及びフレームの音響尤度が計算される。以上の句境界尤度と音響尤度の2系統をSSS-LR連続音声認識システムに入力し、フレームの同期をとる。最終的に終了フレームにおいて文法的に受理された候補のうち、最も尤度の高い音素系列を出力する。

2　フレーム同期型句境界尤度計算

句境界尤度の計算にはこれまでに我々が提案してきた「F_0 パターン連続整合法」を使用する。本手法は代表的なアクセント句を表現した F_0 テンプレートと入力音声の F_0 パターンの One Stage DP を基本としているのでフレーム同期処理には適している。しかし、これまでの報告では句境界候補の出力は音声入力の終了と同時であり、最終フレームまでの累積二乗誤差が最小なテンプレート系列であった。そこで今回は One Stage DP 上でフレーム毎に立てられた句境界仮説について、微小時間経過後の入力パターンとの歪みの変化をもって仮説の妥当性を検証し、句境界尤度を出力する。フレーム同期で入力 F_0 パターンをトレースする類似した手法に Geoffrois の手法 [14] があるが、扱っている仮説は F_0 生成モデルのパラメータのフレーズ指令とアクセント指令であり、指令の位置からは句境界を推定することはできない。しかし、我々の提案する F_0 テンプレートはこれらの指令と句境界の相対的な位置関係を統計的にモデル化したものであるので、テンプレートの接続を仮定することで句境界の仮説を立てることが可能である。

まず、システムの学習時には、視察及び聴取により決定したアクセント句の F_0 パターンを半自動的に抽出 [15] された F_0 生成モデルの指令パラメータによってモデル化し、クラスタリングの手法で分類することによってテンプレートを作成する。

句境界の自動検出においては、入力音声の F_0 パターンとテンプレートとの One Stage DP 整合を行い、各フレームの句境界仮説が真である場合の累積二乗誤差と偽である場合の累積二乗誤差の差を計算し、句境界尤度として出力する。

2.1　ピッチ抽出

ピッチの抽出にはラグ窓法 [16] を使用する。この手法はラグ窓の調整によって倍ピッチの抑制が可能で、我々は12kHzの入力音声に対し、512ポイントで処理する場合に、男性話者では100ポイント幅、女性話者では200ポイント幅を使用していた。当然、女性話者で男性の窓幅を使用した場合には半ピッチ誤りに陥いり易い。そこで、不特定話者を対象とする場合には、窓幅は200ポイントとし、デルタピッチの計算処理においてピッチエラーを修正する。

図2.2はピッチおよびデルタピッチの抽出例であり、ピッチパターンの縦軸は対数周波数値、横軸は分析フレームである。破線で仕切った区間は適当な間隔で文意がある程度まとまるように区切ったもので、厳密にアクセント句や文節とは一致していない。

デルタピッチの計算では半ピッチ、倍ピッチの誤りを吸収するべく、回帰分析とピッチ修正を繰り返す。図に四角で示された区間のピッチ系列からデルタピッチを計算するとき、中心ピッチの重みが最も高くなる三角窓にピッチ信頼度 (ピッチ周期による自己相関値) を乗じて正規化した窓を使用し、回帰分析を行なう。1回目の分析で分析区間のピッチパターンが $y = ax + b$ で直線近似されたとき、区間のそれぞれのピッチについて直線から n 倍ピッチ $(\log n)$、$\frac{1}{n}$ ピッチ $(-\log n)$ のずれがあるとき、それぞれ $-\log n, \log n$ だけピッチの値を修正し、デルタピッチを再計算する。この処理を数回繰り返す。この手法では分析区間のピッチが全て半ピッチ誤りになる場合もありうるが、デルタピッチを計算する上では問題は無い。分析が次フレームにシフトしたときは、新たに修正前のピッチから同様の処理をする。

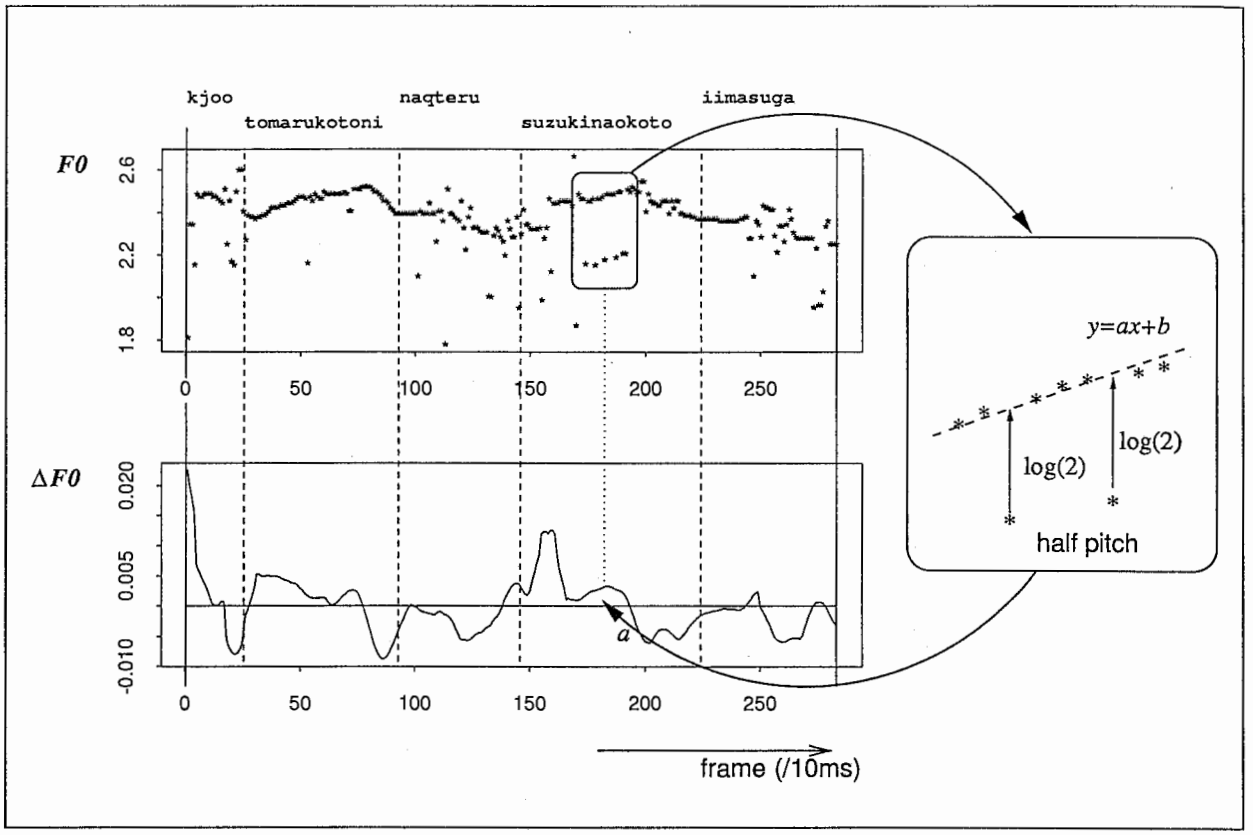

図 2.2: ピッチおよびデルタピッチの抽出

アルゴリズム

```
while( 入力 ){
    /* 入力音声 */
    for( i = 0; i < 分析幅 − 分析シフト; i++ )
        入力音声 [i] = 入力音声 [i+ 分析シフト];
    for( i = 分析幅 − 分析シフト; i < 分析幅; i++ )
        入力音声 [i] = 読み込み;
    /* ピッチ */
    for( i = 0; i < デルタピッチ分析幅 - 1; i++ )
        ピッチ [i] = ピッチ [i+1];
    ピッチ [i] = ラグ窓法 ( 入力音声 );
    /* デルタピッチ */
    for( i = 0; i < デルタピッチ分析幅; i++ )
        修正ピッチ [i] = ピッチ [i];
    デルタピッチ = デルタピッチ計算 ( 修正ピッチ );
    for( i = 0; i < 修正回数; i++ ){
        ピッチ修正 ( 修正ピッチ );
```

```
        デルタピッチ = デルタピッチ計算 ( 修正ピッチ );
    }
    出力 ( デルタピッチ );
}
```

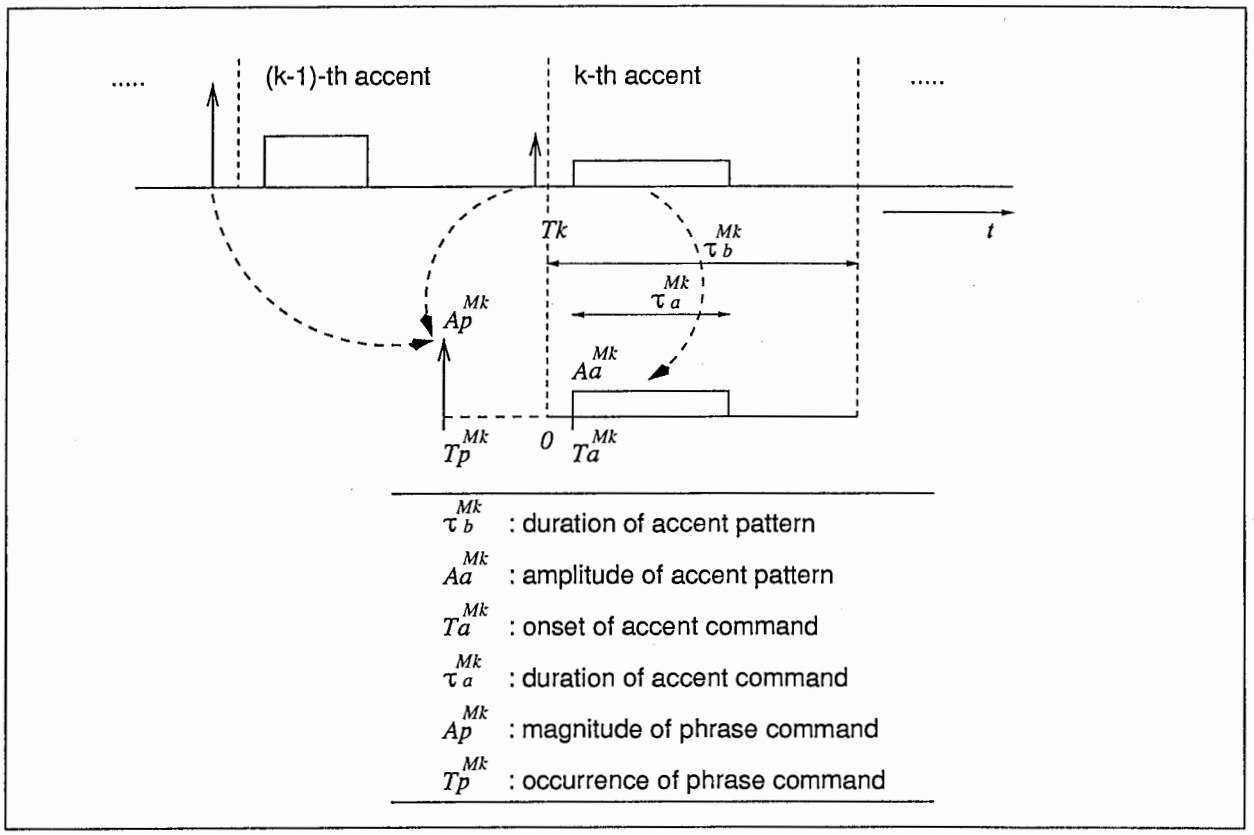

図 2.3: アクセント句のモデル化

2.2 F_0 テンプレートの学習

(2.2.1) アクセント句のモデル化

藤崎らの提案する F_0 生成モデルでは、F_0 パターンは句頭から句末にかけて緩やかに下降するフレーズ成分と、アクセント句に対応するアクセント成分との和として捉えられ、対数 F_0 周波数は時刻 t の関数として

$$\ln F_0(t) = \ln F_{\min} + \sum_{i=1}^{I} A_{p_i} G_{p_i}(t - T_{p_i}) + \sum_{j=1}^{J} A_{a_j} \left\{ G_{a_j}(t - T_{a_j}) - G_{a_j}(t - (T_{a_j} + \tau_{a_j})) \right\} \quad (2.1)$$

により与えられる。ここで $F_{\min}$ は話者に依存した最低基本周波数、I, J は 1 つの発話におけるフレーズ数およびアクセント数、A_{p_i}, A_{a_j} は i 番目のフレーズ指令および j 番目のアクセント指令の大きさ、T_{p_i} は i 番目のフレーズ指令の発生点、T_{a_j}, τ_{a_j} は j 番目のアクセント指令の開始点及び継続時間である。また $G_{p_i}(t), G_{a_j}(t)$ はそれぞれフレーズ制御機構のインパルス応答関数、アクセント制御機構のステップ応答関数であり、α_i, β_j をそれぞれの固有角周波数とすれば

$$G_{p_i}(t) = \begin{cases} \alpha_i^2 t e^{-\alpha_i t}, & t \geq 0, \\ 0, & \text{otherwise} \end{cases} \quad (2.2)$$

$$G_{a_j}(t) = \begin{cases} \min[1 - (1 + \beta_j t) e^{-\beta_j t}, \theta_j], & t \geq 0, \\ 0, & \text{otherwise} \end{cases} \quad (2.3)$$

である。ただし、θ_j は $G_{a_j}(t)$ の上限値 (およそ 0.9) である。

本手法のアクセント句のモデル化には、このアクセント指令・フレーズ指令を用い、個々のアクセント句を図2.3に示された指令パラメータ

$$\mathcal{M}_k = \{T_p^{\mathcal{M}_k}, A_p^{\mathcal{M}_k}, T_a^{\mathcal{M}_k}, \tau_a^{\mathcal{M}_k}, A_a^{\mathcal{M}_k}, \tau_b^{\mathcal{M}_k}\} \tag{2.4}$$

によって表現する。当該アクセント句のモデル化に使用する指令は当該アクセント指令と当該アクセント句の開始前までに発生したフレーズ指令である。アクセント成分は時刻 T_{a_j} の正のステップ応答と τ_{a_j} 後の負の等しい大きさのステップ応答によって打ち消し合うことから、後続するアクセント句の F_0 パターンに与える影響は大きくなないと考えられるので、先行アクセント指令はモデルの要素に加えず、

$$T_a^{\mathcal{M}_k} = T_{a_k} - T_k, \tag{2.5}$$
$$A_a^{\mathcal{M}_k} = A_{a_k}, \tag{2.6}$$
$$\tau_a^{\mathcal{M}_k} = \tau_{a_k} \tag{2.7}$$

の3つのパラメータでアクセント成分を表す。ここで T_k は当該アクセント句の開始時刻である。一方、フレーズ成分は、$k'(\leq k)$ 個の先行するフレーズ成分の和により

$$T_p^{\mathcal{M}_k} = \frac{\sum_{i=1}^{k'} A_{p_i}(T_{p_i} - T_k)e^{\alpha(T_{p_i}-T_k)}}{\sum_{i=1}^{k'} A_{p_i} e^{\alpha(T_{p_i}-T_k)}}, \tag{2.8}$$
$$A_p^{\mathcal{M}_k} = \frac{\sum_{i=1}^{k'} A_{p_i} e^{\alpha(T_{p_i}-T_k)}}{e^{\alpha T_p^{\mathcal{M}_k}}} \tag{2.9}$$

で計算される。ただし、固有角周波数 α, β については文献[15]より、それぞれ 3.0, 20.0 を使用した。これらの値は話者、発話様式の違いによる差[17]が他のパラメータと比較して非常に小さいので固定したことによる影響はほとんど無い。

(2.2.2) アクセントモデルのクラスタリング

モデル化したアクセント句をLBG法を用いて分類し、テンプレートを作成する。句境界検出時のパターン整合において対数 F_0 パターンの二乗誤差を基準とすることから、分類にはモデルから生成される対数 F_0 パターンを用いる。

まず、式(2.1)に基づいて、個々のモデル化アクセント句 $\mathcal{M}_j$ を対数 F_0 パターン

$$\hat{P}_j = (\hat{p}_{j1}, \cdots, \hat{p}_{ji}, \cdots, \hat{p}_{jL}) \tag{2.10}$$

に変換する。ここで、$\hat{p}_{ji}$ は対数 F_0 値の時系列であり、L はパターン長を揃えるための固定値である。このとき、2つのパターン $\hat{P}_j, \hat{P}_k$ 間の距離を

$$D(\hat{P}_j, \hat{P}_k) = \sum_{i=1}^{L} (\hat{p}_{ji} - \hat{p}_{ki})^2 \tag{2.11}$$

と定義する。この距離尺度を基準にクラスタリングを行なうことにより、クラスタ数 K のときのテンプレートの集合

$$\mathcal{R} = \{\mathcal{R}_0, \mathcal{R}_1, \cdots, \mathcal{R}_{K-1}\} \tag{2.12}$$

を求める。個々のテンプレートはクラスタの重心パターンを近似するパラメータ集合

$$\mathcal{R}_k = \{T_p^{\mathcal{R}_k}, A_p^{\mathcal{R}_k}, T_a^{\mathcal{R}_k}, \tau_a^{\mathcal{R}_k}, A_a^{\mathcal{R}_k}\} \tag{2.13}$$

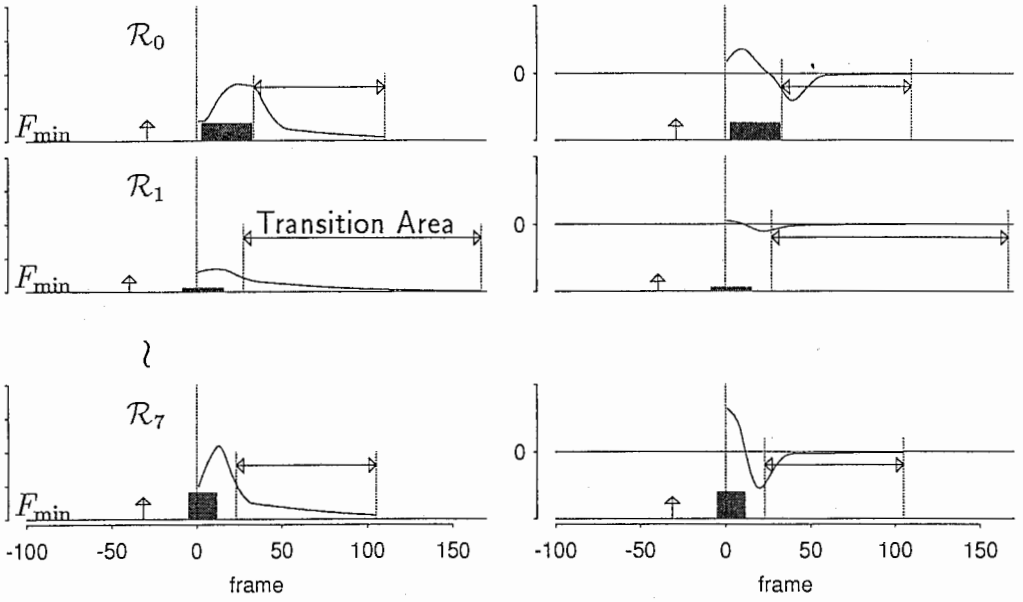

図 2.4: ΔF_0 テンプレート

であり、k 番目のクラスタに属するアクセント句の個数を N_k とすると

$$T_p^{\mathcal{R}_k} = \frac{\sum_{i=1}^{N_k} A_p^{\mathcal{M}_i} T_p^{\mathcal{M}_i} e^{\alpha T_p^{\mathcal{M}_i}}}{\sum_{i=1}^{N_k} A_p^{\mathcal{M}_i} e^{\alpha T_p^{\mathcal{M}_i}}}, \tag{2.14}$$

$$A_p^{\mathcal{R}_k} = \frac{\sum_{i=1}^{N_k} A_p^{\mathcal{M}_i} e^{\alpha T_p^{\mathcal{M}_i}}}{N_k e^{\alpha T_p^{\mathcal{R}_k}}}, \tag{2.15}$$

$$T_a^{\mathcal{R}_k} = \frac{\sum_{i=1}^{N_k} T_a^{\mathcal{M}_i}}{N_k}, \tag{2.16}$$

$$\tau_a^{\mathcal{R}_k} = \frac{\sum_{i=1}^{N_k} \tau_a^{\mathcal{M}_i}}{N_k}, \tag{2.17}$$

$$A_a^{\mathcal{R}_k} = \frac{\int_{T_a^{\mathcal{R}_k}}^{T_a^{\mathcal{R}_k}+\tau_a^{\mathcal{R}_k}} \sum_{i=1}^{N_k} f_i(t)dt}{N_k \tau_a^{\mathcal{R}_k}}, \tag{2.18}$$

$$f_i(t) = \begin{cases} A_a^{\mathcal{M}_i}, & T_a^{\mathcal{M}_i} \le t \le T_a^{\mathcal{M}_i} + \tau_a^{\mathcal{M}_i}, \\ 0, & \text{otherwise} \end{cases}$$

のように計算される。これらのラメータより生成される F_0 パターンを F_0 テンプレートと呼ぶことにする。

図 2.4はクラスタ数が 8 の場合の分類結果の例で、左側は F_0 テンプレートを、右側は回帰係数による ΔF_0 テンプレートを示している。横軸は時間軸 (1 frame = 10ms) であり、アクセント句の開始点は時刻 0 である。また、縦軸は指令の大きさであり、矢印 (↑) はフレーズの指令、長方形はアクセント指令のタイミングと大きさを表している。アクセント指令の終了後の矢印の区間 (↔) はテンプレートの終端可能な区間を表している。

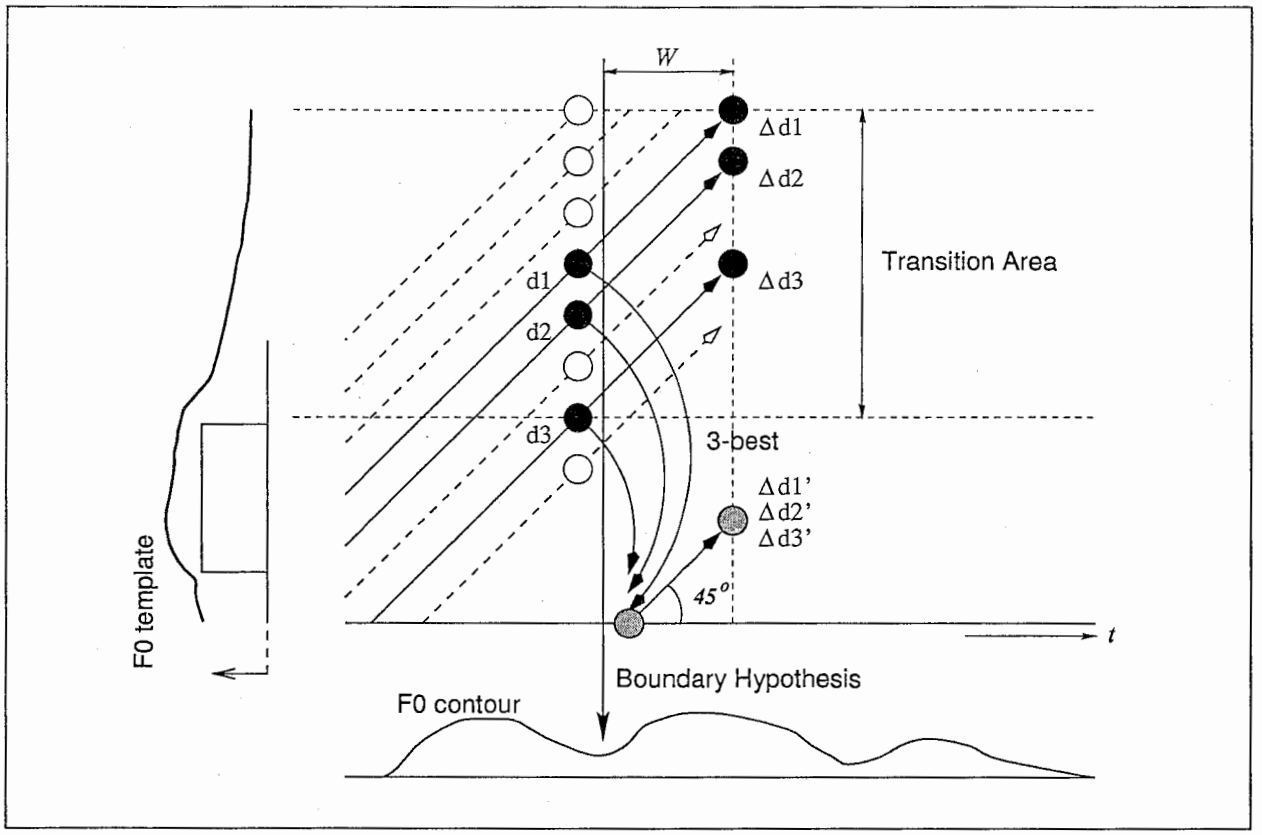

図 2.5: One Stage DP によるアクセント句境界推定

2.3 アクセント句境界尤度計算

入力音声の ΔF_0 パターンと ΔF_0 テンプレートの One Stage DP により句境界を推定する。One Stage DP による句境界検出法の詳細については文献 [1] を参照してもらいたい。アクセント句境界は ΔF_0 テンプレートの接続フレームであるので、各フレームにおける新たな F_0 テンプレートの生起が句境界仮説となる。

図 2.5はテンプレートが 1 つの場合の句境界尤度計算の考え方を示したものである[1]。遷移可能な区間幅 J における、それぞれの格子上の点が N-best 候補を保持しているとすると、1 つ前のフレームから遷移しうる候補は $J \times N$ 個であり、それぞれの候補はその時刻までのコスト (対数周波数による累積歪み)d_n を持つ。これらの候補が当該フレームにおいて新たなテンプレートの始点へと遷移するとき、遷移コストとしてテンプレート間の遷移確率の対数の絶対値を加える。(図の場合はテンプレート数 =1 なので遷移コストは全て等しい。) このとき $J \times N$ 個の候補のうち最もコストの小さい N 個の候補を新たなテンプレートの始点の候補として残す。これらの句境界仮説となった候補が W 後のフレームまでに増加したコストが

$$\Delta d'_n \qquad (n = 1 \cdots N), \tag{2.19}$$

句境界仮説の遷移元となった候補が遷移を行なわずに句を継続した場合のコストの増加分が

$$\Delta d_n \qquad (n = 1 \cdots N) \tag{2.20}$$

[1](注意) 図は F_0 テンプレートの例であるが、実際には ΔF_0 テンプレートを使用している。

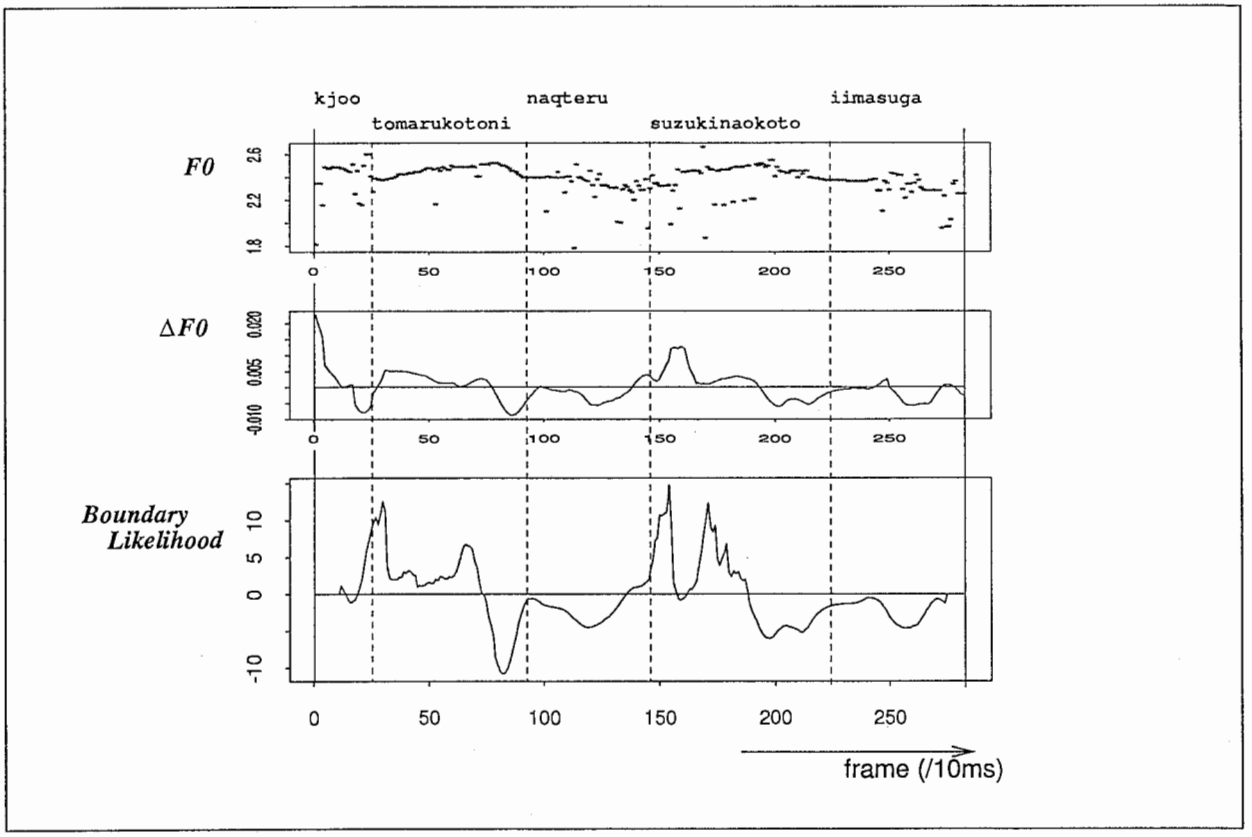

図 2.6: アクセント句境界尤度

となる場合、W 前の時刻の句境界仮説が立てられたフレームの句境界尤度を

$$\frac{1}{N}(\sum_{n=1}^{N}\Delta d_n - \sum_{n=1}^{N}\Delta d'_n) \tag{2.21}$$

として定義する。

もしテンプレート数が K 個であれば、各フレームの終端可能な候補の総数は $\sum_{k=1}^{K} J_k \times N$ となり、各テンプレート毎に N 個の候補を選択して句境界仮説を立てることになる。このため、テンプレート毎に句境界尤度が求まり、多次元尤度情報として出力することができるが、本報告ではこのうち最大の尤度のみを出力するものとする。なお、フレームの進行とともに増加する候補数は1フレームあたり $K \times N$ 個であるが、同時にテンプレートの最大長を超えて刈られる候補も $K \times N$ 個存在するので、候補の総数には上限がある。ただし、処理効率を上げるために候補の許容数 (ビーム幅) を制限して枝刈りを行う。

図2.6に実際に推定された句境界尤度を示す。入力パターンは ΔF_0 パターンであり、参照したテンプレートも ΔF_0 テンプレートである。というのも、F_0 テンプレートでは、話者に依存する $F_{\min}$ の値の推定が必要であり、$F_{\min}$ の推定を誤れば、句境界尤度も大きく異なる。現段階ではフレーム同期に $F_{\min}$ 推定するのは不可能であると考えている。

アルゴリズム

候補構造 {

親候補;

状態;

```
    テンプレート;
    スコア;
}
cand, new_cand : 候補
CANDS : 全候補の集合
NEW_CANDS : 新たに生成された候補の集合
NEW_CANDS' : 新たに生成された候補中から選択された N-best 候補の集合
/* 初期化 */
CANDS = NULL;
for( i = 0; i < テンプレート数; i++ ){
    new_cand の生成;
    new_cand. スコア = 開始コスト;
    new_cand. 親候補 = NULL;
    new_cand. 状態 = 0;
    new_cand. テンプレート = i;
    new_cand を CANDS に追加;
}
while( 入力デルタピッチ ){
    /* 句境界仮説候補の作成 */
    NEW_CANDS' = NULL;
    for( i = 0; i < テンプレート数; i++ ){
        遷移先テンプレート = i;
        NEW_CANDS = NULL;
        for( cand ∈ CANDS ){
            遷移元テンプレート = cand. テンプレート;
            遷移開始 = 遷移元テンプレートの遷移区間の開始状態;
            遷移終了 = 遷移元テンプレートの遷移区間の終了状態;
            if( 遷移開始 ≤ cand. 状態 ≤ 遷移終了 ){
                遷移確率 = prob( 遷移先テンプレート | 遷移元テンプレート );
                遷移コスト = −log(遷移確率);
                new_cand の生成;
                new_cand. スコア = cand. スコア + 遷移コスト;
                new_cand. 親候補 = cand;
```

```
            new_cand.状態 = 0;
            new_cand.テンプレート = 遷移先テンプレート;
            new_cand を NEW_CANDS に追加;
        }
    }
    NEW_CANDS 中の N-best 候補を選択し NEW_CANDS' に追加;
}
/* 傾き 1 (線形 45 度) の shift */
for( cand ∈ CANDS ){
    cand.状態 ++;
    if( cand.状態 > cand.テンプレートの最大長 )
        cand を消去;
}
/* 当該フレームにおける句境界仮説候補を候補集合に追加 */
NEW_CANDS' を CANDS に追加;
/* 当該フレームにおけるコストの追加 */
for( cand ∈ CANDS ){
    整合テンプレートの値 = cand.テンプレート [cand.状態];
    歪み = ( 入力 − 整合テンプレートの値 )^2;
    cand.スコア += 歪み;
}
/* W フレーム前に立てた句境界仮説の尤度計算 */
for( i = 0 i < テンプレート数; i++ )
    スコア [i] = 0;
for( cand ∈ CANDS ){
    if( cand.状態 == W ){
        スコア [cand.テンプレート番号] += cand の親グリッドのスコア増加分 −
          gird のスコア増加分;
        候補数 [i]++;
    }
}
for( i = 0 i < テンプレート数; i++ ){
    句境界尤度 = −HUGE;
    スコア [i] /= 候補数 [i];
    if( スコア [i] > 句境界尤度 )
        句境界尤度 = スコア [i];
```

```
    }
    /* 句境界尤度の出力 */
    出力 ( 句境界尤度 );
    /* 候補の枝刈り */
    枝刈り ( CANDS );
}
```

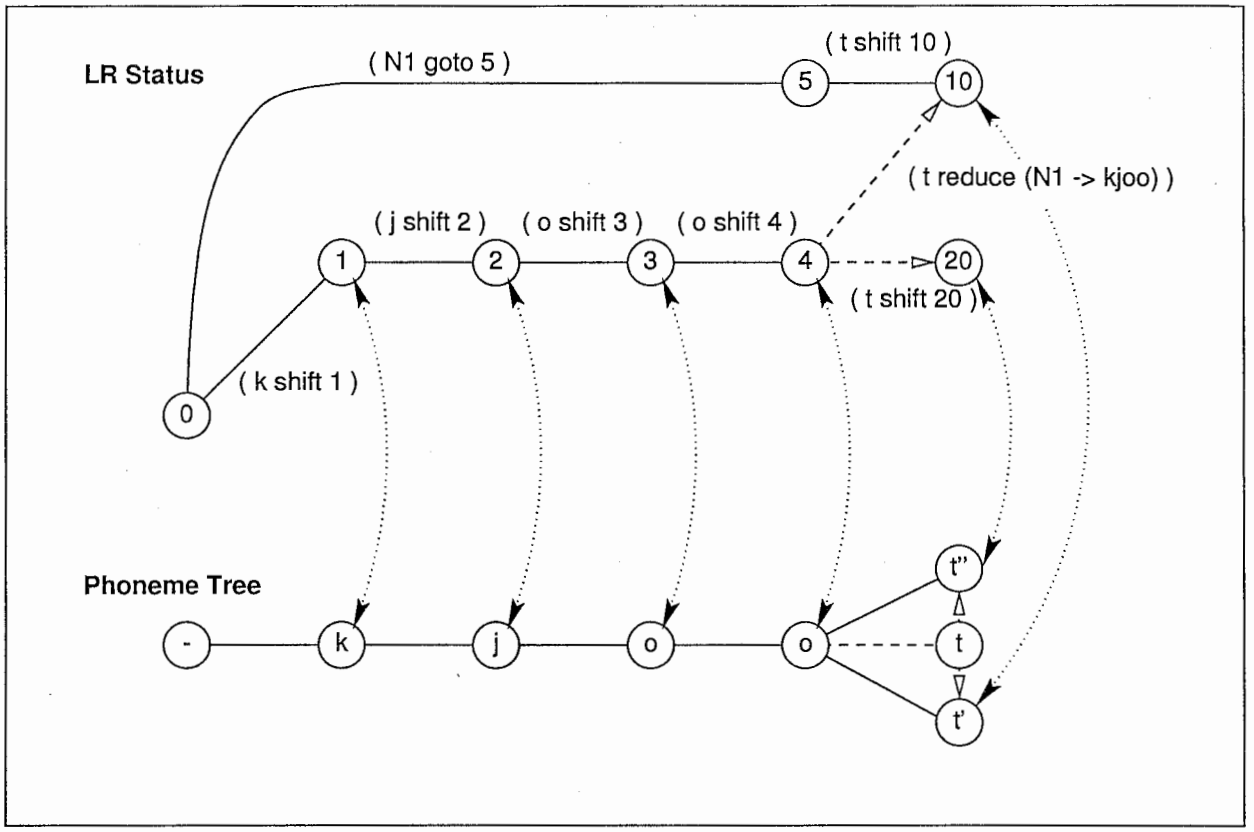

図 2.7: LR状態による音素履歴木の分割

3 フレーム同期型連続音声認識

フレーム同期型SSS-LRの詳細については門前らの報告[18]を参照して戴き、ここでは句境界尤度を利用した変更点について述べる。フレーム同期型SSS-LRでは、ある分析時刻における各々の文候補は1つの音素履歴を持ち、1つの音素履歴はそれが解釈し得る複数の構文仮説を持っている。文候補のスコアとなる音響尤度は音素履歴によってのみ決定され、同一の履歴に対しては異なる構文仮説であろうと、音響尤度が異なることは無い。一方、韻律情報は構文推定などに用いられるように、構文仮説の制御が可能である。つまり、句境界尤度のような韻律尤度を同一の音素履歴中、韻律的に妥当と思われる構文仮説に対し、スコアとして与えることができる。

3.1 LR状態による音素履歴の分割

図2.7は異なるLR状態に対し、句境界尤度を与えるための処理である。One Pass Viterbi上での文候補は音素履歴と1対1に対応しているので、異なる尤度を与えるためには履歴木を分割することになる。現在の韻律尤度はアクセント句境界尤度のみであるので、同一の音素履歴においてスコアが異なる状態はアクセント句にReduceする場合と、しない場合の2通りのみを考える。分割のアルゴリズムは以下のようになる。

アルゴリズム

音素列 k,j,o,o が認識されている状態で、次の音素 t を予測した時刻に

1. (アクセント句 $->$ kjoo) となる規則によってReduceする状態が存在するとき

表 2.1: 句構造

<start>	↔	<_start>
<_start>	↔	q1 <ss> q2
<ss>	↔	<mjphrase>
<mjphrase>	↔	<mjphrase> <minphrase>
<mjphrase>	↔	<minphrase> pau
<mjphrase>	↔	<minphrase>
<minphrase>	↔	<phrase>
<minphrase>	↔	<phrase> < 助詞 >
<phrase>	↔	< 名詞句 >
<phrase>	↔	< 動詞句 >
<phrase>	↔	< 副詞句 >
<phrase>	↔	< 連体詞句 >
<phrase>	↔	< 感動詞 >
<phrase>	↔	< 間投詞 >
<phrase>	↔	< 一部の間投詞 > <phrase>
<phrase>	↔	< 形容詞 >
<phrase>	↔	< 接続詞 >
<phrase>	↔	< 連体詞 >

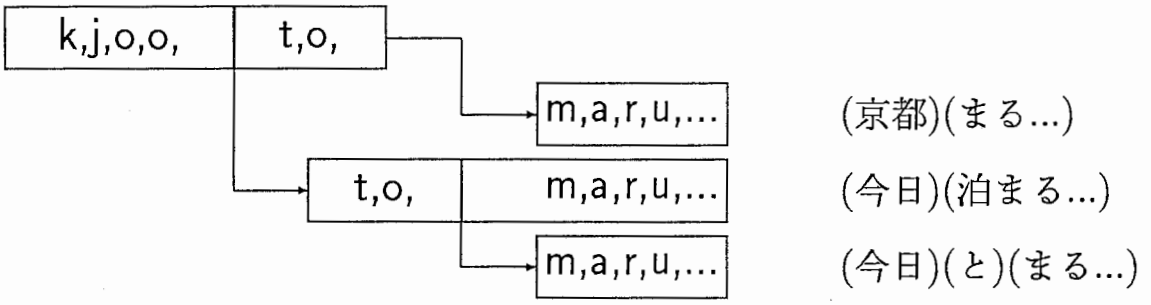

図 2.8: 音素履歴木の例

- アクセント句である kjoo の後に続く音素 t 用の履歴 t' を分割し、その時刻の句境界尤度を与える。
- アクセント句ではない kjoo の後に続く音素 t 用の履歴 t" を分割する。

2. (アクセント句 − > kjoo) となる規則が存在しないとき
   - kjoo の後に続く音素 t は分割しない。

アクセント句の文法規則を記述するには、相応の知識と資料と時間を要するので、今のところ表 2.1のような仮の規則を使用している。ただし、間投詞のうちの短い単語がしばしば句と認識されるのを避けるため「あ (a)」「ああ (a a)」「あっ (a q)」「う (u)」「うう (u u)」「え (e)」「えと (e t o)」「お (o)」「と (t o)」「ん (N)」などは < 一部の間投詞 > とした。

この規則では音素列

k,j,o,o,t,o,m,a,r,u,$\cdots$

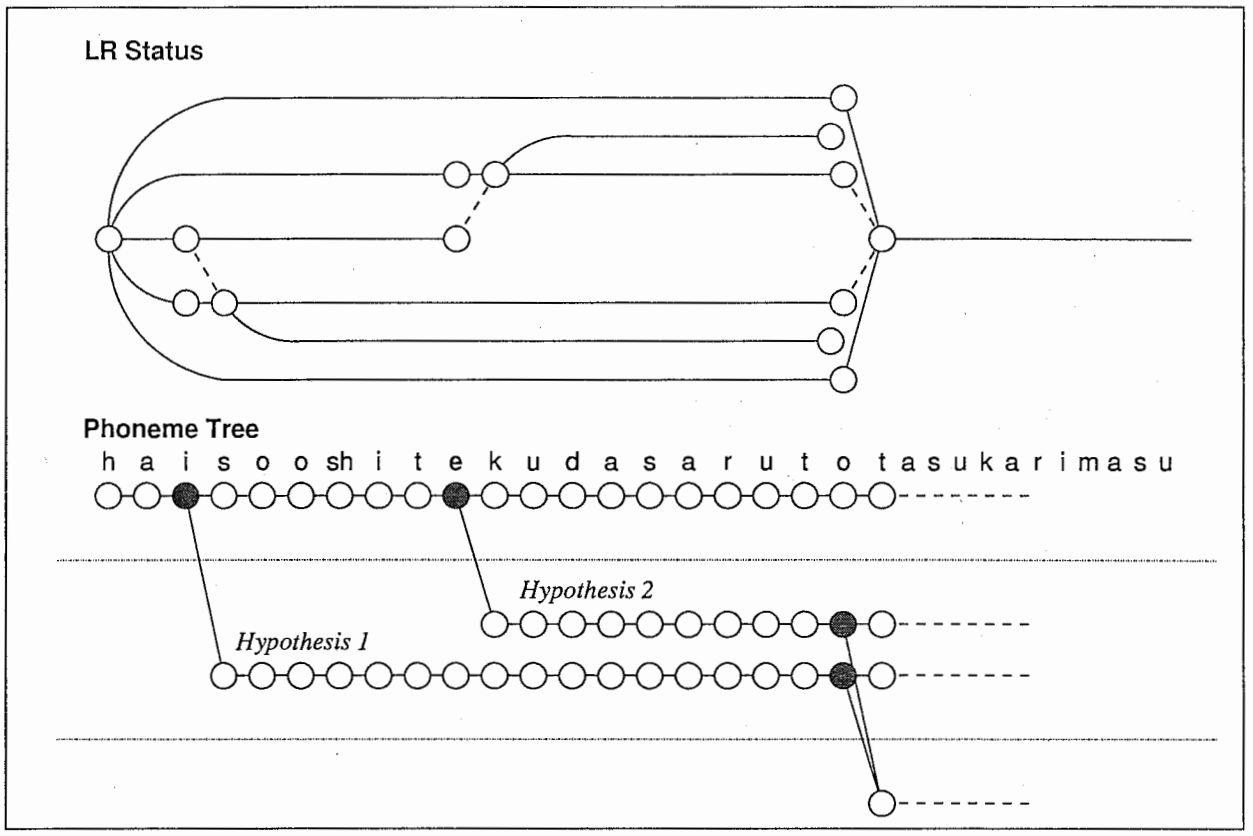

図 2.9: LR 状態による音素履歴木の統合

が認識された時に構文として、図2.8に示される3通りが解釈された場合、問題になるのは2番目の構文である。なぜなら、名詞 kjoo の時刻で一度句に Reduce し、さらに 助詞 to が認識された時刻でもう一度句に Reduce することになるので、非常に句境界尤度が高くなってしまうからである。そこでアクセント句の時間長制御として、一度アクセント句として句境界尤度が与えられた仮説には、最小句境界間隔に相当すると思われる一定時間の間は句境界尤度を与えないものとする。このとき、先のアルゴリズムにおいて、

音素列 k,j,o,o が認識されている状態で、次の音素 t を予測した時刻に

1. (アクセント句 $->$ kjoo) となる規則によって Reduce する状態が存在するとき
   - アクセント句である kjoo の後に続きながら句境界尤度の加算されない場合を考慮して、句を履歴として持たない t" も句に Reduce する LR 状態も持つ。

などの変更が必要となる。

3.2 LR 状態による音素履歴の統合[2]

なお、このように履歴木を分割していくと等しい音素系列の枝が広がっていくため、枝刈りのビーム幅内で展開できる可能性が減少してしまう。したがって途中で分岐した同一音素履歴が同じ時刻に再び句境界となり得る場合には統合する処理が必要である。例えば図

[2]本レポートの実験では時間の都合により行なわれていない。

2.9は

h,a,i,s,o,o,sh,i,t,e,k,u,d,a,s,a,r,u,t,o,t,a,s,u,k,a,r,i,m,a,s,u

の入力に対して、

「はい、そうして下さると助かります。」

「配送して下さると助かります。」

の2種類の意味にとれる文章であるが、途中で同じLR状態になったときには句境界尤度の高い方を残すことで候補を減らすことが可能である。ただし、2つの候補はその後のReduceによって別々の状態に再分岐する可能性があるので、最適性は保証されない。

第 3 章

連続音声認識実験

1　音声資料

実験評価資料

音声認識評価用デベロップメントテストセット SL2 より、1 対話 (TAS12010・FYUYO・女性・8 発声) を使用して実験評価を行なう。

評価方法は音素認識率、音素認識精度、形態素認識率、形態素認識精度とし、それぞれ

$$\text{音素(形態素)認識率} = \text{正解数}/\text{音素(形態素)数}$$

$$\text{音素(形態素)精度} = (\text{正解数} - \text{挿入数})/\text{音素(形態素)数}$$

として定義する。

2　実験条件

音声波形切り出し

音声パワーの閾値からおおよその発声区間を切り出すことは可能であるが、ポーズ検出誤差の影響を評価・考察から省略できるように予め視察によって検出されているポーズ情報を使用する。(使用したポーズ情報は .LBL ファイル、もしくは .TRS ファイルのもの。)

ポーズに挟まれた発声区間の前後 30ms を余分に切り出し、発声開始 -30ms ～ 発声終了 +30ms を 1 発声単位とする。ただし、発声終了後、60ms 以内 に次の発声が開始される場合はポーズ区間の中間時刻で分割した。

ピッチ抽出

ピッチ抽出	ラグ窓法
FFT	512 ポイント (42.7ms)
分析シフト	120 ポイント (10.0ms)(= 1 フレーム)
ピッチ探索範囲	50Hz ～ 500Hz
ラグ窓幅	200 ポイント (16.7ms)
デルタピッチ計算	回帰分析
窓 (三角窓 + 信頼度) 幅	21 フレーム
再計算	5 回

句境界尤度計算

分析シフト	10.0ms
整合	デルタピッチのみ ($F_{\min}$ 不要)
テンプレート数	8個
N-best 数	10位
境界尤度計算窓幅	+100.0ms
ビーム幅	2000
遷移コスト係数	1 (量的な意味は無い)
最小句境界間隔 (*1)	100.0ms

(*1) 本実験で使用するテンプレートは朗読音声で学習したものであるため、発声速度が遅く、アクセント句境界間隔が長い。そのため、テンプレートが整合できる最小のアクセント句は学習結果より160ms以上となり、自然発声では検出できない境界が生じる。そのような句境界の検出を可能とするため、今回はテンプレートの終端可能な長さの最小値を100msにまで下げた。

F_0 テンプレート

… ATR連続音声データベース 音韻バランス503文章より、話者 MHT, MSH, MTK の発声の中から藤崎モデルパラメータを付与した合計565文章を選択、LBG法で最大16クラスタまで学習した。

テプレート遷移頻度

… F_0 テンプレートの学習に使用された565文章中の3046個のアクセント句をテンプレートで量子化したときのテンプレート遷移頻度を使用。

また、対象実験として視察句境界を中心に正規分布で確率を与えた時系列関数「視察句境界尤度」を用いた実験も行なった。

フレームパラメータ計算

分析シフト	5.0ms
ハミング窓	20ms
特徴量	1 ～ 16次LPCケプストラム
	1 ～ 16次ΔLPCケプストラム
	log パワー
	Δ log パワー

フレーム同期 SSS-LR

分析シフト	5.0ms
ビーム幅	3000
ローカルビーム幅	20
最小句境界間隔	100.0ms
句境界尤度係数	variable

ただし、音素履歴の統合を用いたプログラムは時間の都合で間に合わなかったので、今回の実験では同一音素履歴の分割のみ行なっている。

文法

次の 2 種類の文法を使用した。

文法 1 (制約の緩い文法)

文献 [19] の Grammar-III の品詞を細分化したもの。単語辞書は音声認識評価用デベロップメントテストセットより、話題の偏らない testset SL1 (TAS22001, TAS12009, TCC22074, TCC22094, TDS32007, TGS12001, THS12002, TRS12001, TSS12002)、および実験評価資料を含む testset SL2 (TAS12007, TAS12010, TAS12013, TAS12026, TAS22021, TAS22032, TAS32007, TAS32009, TAS32015, TAS32016) の合計 19 会話より作成。音素パープレキシティ約 9。単語パープレキシティ 約 300。

文法 2 (testset SL1 より作成した文法)

音声認識評価用デベロップメントテストセットの SL1 (12 対話) から作成した文法 [20]。音素パープレキティ 5.67。単語パープレキシティ 158.99。これに実験評価資料 TAS12010 の単語を加え、表 2.1 と同様な句構造規則を追加した。

HMnet

話者 HMT の 5240 単語にりより 400 状態 5 混合のモデルを作成。さらに 1 状態 10 混合のポーズモデルを付加。話者 FTK を重畳して再学習したものを使用する。発話様式の適応あり。

3　実験結果

表 3.1: 認識精度

会話 ID	文法	尤度係数	音素数	認識率	(精度)	形態素数	認識率	(精度)
TAS12010		句境界尤度無し (音素履歴の分割無し)						
		—		*75.56*	*(67.29)*		*44.94*	*(5.62)*
		視察句境界尤度						
		10000.0		74.06	(63.53)		41.57	(-1.12)
		20000.0		72.18	(62.41)		33.71	(-5.62)
	1	40000.0	266	74.06	(62.78)	89	40.45	(-6.74)
		80000.0		**78.57**	(59.77)		**52.81**	(-5.62)
		(※)		**79.32**	(62.78)		**55.06**	(3.37)
		自動推定句境界尤度						
		10000.0		72.93	(54.14)		37.08	(-11.24)
		20000.0		74.44	(63.91)		41.57	(-8.99)
		40000.0		75.19	(60.90)		41.57	(-20.22)
		80000.0		71.74	(50.00)		**45.45**	(-38.96)
		(※)		**78.57**	**(69.12)**		**52.81**	(-4.49)
		句境界尤度無し (音素履歴の分割無し)						
		—		*62.41*	*(58.27)*		*41.57*	*(19.10)*
		視察句境界尤度						
		10000.0		**77.07**	**(68.80)**		**50.56**	**(21.35)**
		20000.0		**73.68**	**(64.66)**		**47.19**	(16.85)
	2	40000.0	266	**71.05**	**(62.78)**	89	**46.07**	(10.11)
		80000.0		**77.07**	**(67.29)**		**52.81**	(13.48)
		(※)		**78.20**	**(69.92)**		**53.94**	**(22.47)**
		自動推定句境界尤度						
		10000.0		**72.93**	**(66.17)**		**46.07**	(17.98)
		20000.0		**71.43**	**(62.41)**		38.20	(3.37)
		40000.0		**71.09**	**(59.72)**		**42.86**	(-1.43)
		80000.0		**63.91**	**(43.61)**		28.09	(-26.97)
		(※)		**76.32**	**(65.41)**		**51.69**	(8.99)

(※) 各々の会話文について尤度係数に関わり無く最大認識率のものを累積

4　実験考察

以上の実験結果より次のことが言える。

- 一般に音素モデルが話者に適応していないような条件のもとでは音素認識率が低いため、文法の制約が厳しいと誤った解へと導かれ易い。また、フレーム同期型 SSS-LR では最終時刻で文法的に受理できる候補が無い場合が生じるという問題点がある。このため文法 1 では形態素認識精度が低くても、最終時刻まで認識可能なため、音素認識率は高いが、文法 2 では受理されなっかた部分を脱落として扱うと認識率は低くなる。アクセント句境界尤度を与えた場合には、認識途中までの受理し易い候補にスコアが加算されることになるため、最終時刻で文法が完結する可能性が高くなる。例えば、以下の文

```
LAB: chjoqto tsugoo de tomare  na ku na q ta node     kjaNseru      shi ta i     N de su keredomo
REC: kjuu to tsugoo de toma ru na    donata  Nde ee hi uN seN zhjuu shi ta i Nde N    hi keredomo
```

は、上段が正解形態素系列、下段が認識形態素系列である。句境界尤度を与えない実験では文章の前半部分まで認識して、後半は受理されなかったが、句境界尤度を与えることで最終時刻までの認識が可能になった。このため音素、形態素ともに脱落が減少した。

- アクセント句境界尤度により音素認識率、形態素認識率ともに数値の上では向上した。しかし、形態素認識率 50% 前後では文章としては十分では無く、上の例のように意味のとれないものが多い。これはテストセット SL1 に依存した文法にアクセント句規則を追加したことに原因があるようである。参考までに文法 2 にアクセント句規則を追加しない実験も試みたが、ほとんどが文法的に完了せず、途中時刻までの近似解しか出力されなかった。今回使用したアクセント句規則は評価資料のアクセント句が全て受理されるように作成したものであるが、助詞や間投詞などの短い単語の扱いがかなり不適当である。間投詞などは単独で句になる可能性があるので、下の表のように挿入になり易い。

挿入の多い単語

	間投詞 (う)(u)	間投詞 (んー)(N)	挿入総数
文法 1 尤度無し	4	4	35
自動推定	6	7	51
文法 2 尤度無し	5	1	20
自動推定	6	3	38

- 視察で作成された句境界尤度を使用した場合には句境界尤度の係数が 10000.0 のとき最も認識精度が高く、個々の文章についても 1000.0 のときの認識精度が高い。一方、自動推定句境界尤度はスケールが正規化されていないので、尤度係数の最適化ができず、各々の文章毎に最適な尤度係数が異なるという問題がある。

- 句境界尤度を自動推定する場合には、発声が短いものでも、各フレームでなんらかの数値を出力するため、以下のように認識に悪影響を及ぼすこともある。

句境界尤度無し

```
LAB:   waka ri ma shi ta
REC: u waka ri ma shi ta
```

句境界尤度有り (文法 2・尤度係数 40000.0)

```
LAB:        waka ri ma shi ta
REC: u a ku a    ri ma shi ta
```

- 今回の実験は時間の都合で十分にできなかったため、句境界尤度の有効性について議論するには不十分であろう。今後、さまざまな音素モデル、文法規則を用いて検討していきたい。

第 4 章

結論

1　まとめ

本報告では、アクセント句境界尤度計算、および SSS-LR による連続音声認識をフレーム同期に処理することによって、構文仮説を制御する手法を提案した。

句境界尤度の計算は我々が従来提案してきた One Stage DP によるテンプレート整合法を用いることにより、リアルタイム、フレーム同期で出力することが可能になった。

フレーム同期型 SSS-LR 認識部では、音素履歴の LR 状態がアクセント句に還元するアクションとそれ以外のアクションの両方をもつときには、履歴を 2 分割し、アクセント句に還元する側に句境界尤度を与えることにより構文仮説の制御を行なった。

2　今後の課題

2.1　句境界尤度について

今回の実習では フレーム同期型 SSS-LR での句境界尤度の利用の方に重点を置いたので、句境界尤度の精度については検討する余裕が無かった。ただ、認識実験結果では視察句境界尤度と比較して、精度が落ちることが確認されているので、まだまだ改善の必要があるであろう。

- 実験考察でも述べたが、まず出力される尤度のスケールが正規化されていないので、境界尤度係数が最適化できないという問題がある。

- また、テンプレートの学習には朗読発話を使用していたが、より正確を期すためには自然発話で学習する必要がある。ただし、これまでの句境界検出の研究結果から、あまり依存性は無いと思われる。

- 出力は 1 次元の句境界尤度であったが、異なるアクセント型の句境界に対して同じ尤度を与えるのは不適当である。つまり、アクセント型を考慮した方法、例えば複数のテンプレートに対応した多次元の尤度なども検討する必要がある。

- 今回の手法では一方向であったが、SSS-LR の状態を出力とし、韻律処理系で構文仮説を検証して、スコアを与えるなどの双方向のシステムも検討したい。

2.2 SSS-LR 連続音声認識について

- まず、適切なアクセント句規則の文法を記述しなければならない。今回の実験では音素認識率がかなり低い音素モデルに対し、曖昧性の高い文法を使用する結果となった。

- また、同一の音素履歴を分割することで、限られたビーム幅内での異なる音素履歴の展開の可能性を低めてしまったので、分割された履歴間での枝刈りの処理も必要である。この点については第3.2節の音素履歴の統合について検討する予定である。

謝辞

研究の機会を与えて戴いた北陸先端科学技術大学院大学 下平博 助教授、ATR 音声翻訳通信研究所 山崎泰弘 社長に深く感謝致します。研究を進めるにあたり、SSS-LR 連続音声認識システムに関して清水徹 研究員、林輝昭 研究員、音響尤度計算に関して別府智彦 研究員、韻律特徴抽出に関して平井俊男 研究員よりいろいろと御意見戴いたことを心から感謝致します。最後に研究環境を整えて戴いた TSG の皆様、並びに ATR 音声翻訳通信研究所の皆様に感謝致します。

参考文献

[1] 中井満, シンガーハラルド, 匂坂芳典. “アクセントモデルを用いた F_0 クラスタリングによる句境界検出”. *ATR Technical report*, Vol. TR-IT-0068, , (1994–09).

[2] 大平栄二, 小松昭男, 市川熹. “韻律情報を用いた音声会話文の文構造推定方式”. 信学論 (A), Vol. **J72-A**, 1, pp. 23–31, (1989-01).

[3] 大川茂樹, 瀬戸守, 小林哲則, 白井克彦. “韻律情報と音韻情報を用いた連続音声中の単語検出”. 平 4 秋音講論 I, 2–Q–2, pp. 175–176, (1992-10).

[4] 浮田輝彦, 中川聖一, 坂井利之. “日本語算術文の音声認識におけるピッチパターンの利用”. 信学論 (D), Vol. **J63-D**, pp. 954–961, (1980-11).

[5] 鈴木良弥, 関口芳広, 重永実. “日本語連続音声認識のための韻律情報を利用した句境界の抽出”. 信学論 (D-II), Vol. **J72-DII**, 10, pp. 1606–1617, (1989-10).

[6] H. Fujisaki, K. Hirose, and H. Lei. “Prosody and Syntax in Spoken Sentences of Standard Chinese”. In *ICSLP-92*, pp. 433–436, (1992).

[7] 下平博, 木村正行, 嵯峨山茂樹. “ピッチパタン連続整合による連続音声のセメンテーション”. 信学技報, **SP90**-72, (1990).

[8] M. Nakai and H. Shimodaira. “Accent Phrase Segmentation by Finding N-best Sequences of Pitch Pattern Templates”. In *ICSLP-94*, (1994–09 予定).

[9] 高橋敏, 松永昭一. “統計的韻律モデルによる連続音声の句境界検出”. 信学技報, **SP90**-71, pp. 25–31, (1990).

[10] 今野博之, 広瀬啓吉. “韻律情報を利用した連続音声認識における句境界の検出”. 平 5 音講論 I, 1–8–24, pp. 47–48, (1993-10).

[11] 花沢利行, 阿部芳春, 中島邦男. “意味主導型音声理解システムのための文節スポッティングの検討”. 平 6 音講論 I, 1–Q–14, pp. 169–170, (1994-10).

[12] 遠藤隆, 瀬戸守, 大川茂樹, 渡辺一, 小林哲則, 白井克彦. “韻律情報を用いた句境界の推定”. 平 4 春音講論 I, 3–1–13, pp. 95–96, (1992-03).

[13] 門前聖康, 大関雅和, 好田正紀. “確率文脈自由文法を用いた HMM-LR 文節音声認識における Viterbi best-first サーチの検討”. 平 6 秋信学会, D–390, p. 398, (1994–09).

[14] E. Geoffrois. “Estimation of Prosodic Events from Japanese F_0 Contours”. 信学技報, **SP93**-24, pp. 1–8, (1993-06).

[15] 平井俊男, 岩橋直人, Hélène Valbret, 樋口宣男, 匂坂芳典. “統計的手法による基本周波数パタンの制御”. 平5秋音講論 I, 2–8–3, pp. 225–226, (1993–10).

[16] 嵯峨山茂樹, 古井貞熙. “ラグ窓を用いたピッチ抽出の一方法”. 昭53信学総全大 1235, (1978–03).

[17] 藤崎博也, 廣瀬啓吉, 高橋登, 杉藤美代子. “共通語のイントネーションの音響音声学的特徴と方言の影響”. 音声研資, **S83**-36, pp. 277–284, (1983–10).

[18] 門前聖康, シンガーハラルド, 松永昭一. “フレーム同期型 sss-lr による連続音声認識”. *ATR Technical report*, Vol. TR-IT-0051, , (1994–04).

[19] 清水徹, 松永昭一. “語順の制約による探索空間の削減効果”. 平7春音講論 I, 3–P–9, pp. 179–180, (1995–03).

[20] 竹沢寿幸, 田代敏久, 衛藤純司. “部分木を単位とする音声認識用日本語文法”. *ATR Technical report*, Vol. TR-IT-0110, , (1995–04).

付録 A

テンプレートのパラメータ計算について (導出)

フレーズ指令

2つの異なるフレーズ指令 p_1, p_2 を

$$p_1 = A_{p_1}\alpha(t - T_{p_1})e^{-\alpha(t-T_{p_1})}$$

$$p_2 = A_{p_2}\alpha(t - T_{p_2})e^{-\alpha(t-T_{p_2})}$$

としたとき、$p_3 = (p_1 + p_2)/2$ となるフレーズ指令 p_3 を求める。

$$\begin{aligned}
2A_{p_3}\alpha(t - T_{p_3})e^{-\alpha(t-T_{p_3})} &= A_{p_1}\alpha(t - T_{p_1})e^{-\alpha(t-T_{p_1})} + A_{p_2}\alpha(t - T_{p_2})e^{-\alpha(t-T_{p_2})} \\
2A_{p_3}(t - T_{p_3})e^{\alpha T_{p_3}} &= A_{p_1}(t - T_{p_1})e^{\alpha T_{p_1}} + A_{p_2}(t - T_{p_2})e^{\alpha T_{p_2}} \\
(2A_{p_3}e^{\alpha T_{p_3}} - A_{p_1}e^{\alpha T_{p_1}} - A_{p_2}e^{\alpha T_{p_2}})t &- (2A_{p_3}T_{p_3}e^{\alpha T_{p_3}} - A_{p_1}T_{p_1}e^{\alpha T_{p_1}} - A_{p_2}T_{p_2}e^{\alpha T_{p_2}}) = 0
\end{aligned}$$

全ての t について成り立つとき、第1項より、

$$A_{p_3} = \frac{A_{p_1}e^{\alpha T_{p_1}} + A_{p_2}e^{\alpha T_{p_2}}}{2e^{\alpha T_{p_3}}} \cdots (1)$$

第2項より、

$$\begin{aligned}
2A_{p_3}T_{p_3}e^{\alpha T_{p_3}} &= A_{p_1}T_{p_1}e^{\alpha T_{p_1}} + A_{p_2}T_{p_2}e^{\alpha T_{p_2}} \\
2\frac{A_{p_1}e^{\alpha T_{p_1}} + A_{p_2}e^{\alpha T_{p_2}}}{2e^{\alpha T_{p_3}}}T_{p_3}e^{\alpha T_{p_3}} &= A_{p_1}T_{p_1}e^{\alpha T_{p_1}} + A_{p_2}T_{p_2}e^{\alpha T_{p_2}} \\
(A_{p_1}e^{\alpha T_{p_1}} + A_{p_2}e^{\alpha T_{p_2}})T_{p_3} &= A_{p_1}T_{p_1}e^{\alpha T_{p_1}} + A_{p_2}T_{p_2}e^{\alpha T_{p_2}} \\
T_{p_3} &= \frac{A_{p_1}T_{p_1}e^{\alpha T_{p_1}} + A_{p_2}T_{p_2}e^{\alpha T_{p_2}}}{A_{p_1}e^{\alpha T_{p_1}} + A_{p_2}e^{\alpha T_{p_2}}} \cdots (2)
\end{aligned}$$

従って任意の N 個のフレーズ指令の平均パラメータは以下のようになる。

$$T_p = \frac{\sum_{i=1}^{N} A_{p_i}T_{p_i}e^{\alpha T_{p_i}}}{\sum_{i=1}^{N} A_{p_i}e^{\alpha T_{p_i}}}$$

$$A_p = \frac{\sum_{i=1}^{N} A_{p_i}e^{\alpha T_{p_i}}}{Ne^{\alpha T_p}}$$

アクセント指令

仮に2つの異なるアクセント指令 a_1, a_2 が

$$T_{a_2} = T_{a_1} + \tau_{a_1}$$

$$A_{a_2} = A_{a_1}$$

の関係にあるとき、$a_3 = (a_1 + a_2)/2$ は、

$$T_{a_3} = T_{a_1}$$

$$\tau_{a_3} = \tau_{a_1} + \tau_{a_2}$$

$$A_{a_3} = A_{a_1} = A_{a_2}$$

となる。これは時刻 T_{a_1} で on して τ_{a_1} 後に off するアクセント成分と時刻 $T_{a_1} + \tau_{a_1}$ で on して τ_{a_2} 後に off するアクセント成分の平均パターンが T_{a_1} で on して $\tau_{a_1} + \tau_{a_2}$ 後に off するパターン(振幅 1/2)に等しいことを意味するが、代表パタンをモデルのパラメータで表現する場合に不適当である。そこで、ここではタイミングを重視して、

$$T_{a_3} = \frac{T_{a_1} + T_{a_2}}{2}$$

$$\tau_{a_3} = \frac{\tau_{a_1} + \tau_{a2}}{2}$$

とする。このときの指令の大きさは

$$A_{a_3} = \frac{\int_{T_{a_3}}^{T_{a_3}+\tau_{a_3}} f_1(t) + f_2(t)dt}{2\tau_{a_3}}$$

$$f_1(t) = \begin{cases} A_{a_1} & T_{a_1} \leq t \leq T_{a_1} + \tau_{a_1} \\ 0 & \text{otherwise} \end{cases}$$

$$\vdots$$

$$f_2(t) = \begin{cases} A_{a_2} & T_{a_2} \leq t \leq T_{a_2} + \tau_{a_2} \\ 0 & \text{otherwise} \end{cases}$$

で近似する。従って任意の N 個のアクセント指令の平均パラメータは以下のようになる。

$$T_a = \frac{\sum_{i=1}^{N} T_{a_i}}{N}$$

$$\tau_a = \frac{\sum_{i=1}^{N} \tau_{a_i}}{N}$$

$$A_a = \frac{\int_{T_a}^{T_a+\tau_a} \sum_{i=1}^{N} f_i(t)dt}{N\tau_a}$$

$$f_i(t) = \begin{cases} A_{a_i} & T_{a_i} \leq t \leq T_{a_i} + \tau_{a_i} \\ 0 & \text{otherwise} \end{cases}$$