高品質音声分析変換合成法
STRAIGHT

河原英紀
ATR 人間情報通信研究所
和歌山大学

平成 13 年 1 月 14 日

[取扱注意：本資料は草稿であり，多くの誤りが含まれている．また，最終稿までにはかなりの変更がある．本資料の印刷および配付は，著者の了解の下でのみ可能である．]

目 次

図 目 次

表 目 次

まえがき

本資料は，高品質音声分析変換合成法として開発された STRAIGHT の全体像をまとめたものである．STRAIGHT の構想，理論的背景，実装方法の詳細，インタフェースの設計，具体的な音声加工への応用にわたって一貫して紹介するのは本資料が初めての試みとなる．

STRAIGHT は，ATR 人間情報通信研究所において音声知覚研究用のツールとして生まれた，高品質音声分析変換合成アルゴリズムである．音声知覚の研究をターゲットとしたため，情報量と計算量を度外視して，理解し易く操作し易い情報表現と，非常に高い品質の実現を同時に追求することとした．研究の主流から外れたこの進め方と，多くの研究協力者を得ることのできる ATR という環境が STRAIGHT の実現を可能にしたと言えよう．

STRAIGHT の最初のアイデアではスペクトル包絡の基本周波数に適応した時間周波数平滑化が主要な部分を占めていたため，Speech Transformation and Representation based on Adaptive Interpolation of weiGHTed spectrogram の略称として『STRAIGHT』という名前を提案した．その後の研究の進展により，現在では STRAIGHT の高い品質が音源情報の精密なモデル化とパラメタ抽出にも大きく依存するものであることが明らかとなっている．しかし，それらの新しい要素においても直線 (STRAIGHT line) が様々な場面で重要な役割を果たしているのは不思議な暗合である．

1996 年 4 月の STRAIGHT の発明から実装までの中心的な仕事は著者によるものである．しかし，STRAIGHT の研究・開発をここまで進めることができたのは，数多くの方々の協力，支援，批判，フィードバックがあってはじめて可能なことであった．ここに，深く感謝したい．それらの中でも，最初のアイデアを認め支援してくれた東倉洋一（ATR 人間情報通信研究所社長：当時，敬称略以下同様），樋口宣男（ATR 音声翻訳研究所室長）の諸氏，著者が和歌山大学に移り ATR の招聘研究員の間支援してくれた一ノ瀬裕（ATR 人間情報通信研究所社長），片桐滋（ATR 人間情報通信研究所室長）の諸氏，これらの期間を通じて研究仲間として議論や実験を一緒に進めてきた勝瀬郁代，津崎実，入野俊夫，Parham Zolfaghari，阿竹義徳（奈良先端大），東山恵祐（奈良先端大），坂野秀樹（奈良先端大），Alain de Cheveigné（仏 CNRS），Roy D. Patterson（英 CNBH），片寄晴弘（和歌山大学），後藤真孝（電総研）の諸氏には特に多くを負っている．また，1997 年 10 月より始まった科学技術振興事業団の戦略的基礎研究支援事業 CREST の『脳を創る』領域の甘利俊一統括には，提案したプロジェクトである「聴覚の情景分析に基づく音響・音声処理システム」を支援して頂いたことにより，STRAIGHT から聴覚の計算理論への展開が可能となったことを深く感謝したい．

平成 13 年 1 月 14 日
河原英紀

第1章　STRAIGHTの概要

この章では，STRAIGHT の概要と背景を簡単に説明し，次の章以降のための導入とする．本章の最後には，STRAIGHT に直接あるいは間接的に関連した文献について，位置付けと概要の紹介をまとめた．また，少し旧い版についてではあるが，STRAIGHT についての一般向けの解説も紹介した．

1.1　はじめに

STRAIGHT は音声知覚研究のためのツールとして開発された．既に膨大な音声研究の蓄積があるところに新しいツールを作成する背景には，一つの基本的な疑問があった．限られた技術による劣悪な合成音声を用いて獲得された音声知覚に関する知見が，実際の音声に関する人間の能力をどの程度反映しているだろうかという疑問である．この疑問に答えるためには，自然の音声と区別できない程に高品質であり，かつ知覚に関連する物理パラメタを自由に精密に制御できるような合成音声を用いた実験を行わなければならない．そのため，自然音声を知覚的に意味の有るパラメタに分析し，変換を行った後に再合成するという分析合成型のシステムを実現することとした[1]．

人間の知覚は高度に非線形なシステムである．このようなシステムのふるまいを，要素的な刺激に対する応答だけに基づいて理解できる保証は無い．それらの要素的な刺激を用いた研究に加え，システムが正常に動作している時の振舞いを定量的に解析することが必要なのである．

音声知覚の研究でそのような研究戦略を取ろうとすると，自然の音声に匹敵する高い品質と自然さを有しながらも，物理的パラメタを自由に精密に制御できる刺激の作成法が必要となる．さらに，これまでの音声知覚に関する膨大な研究の蓄積からの継続性と批判的再検討を可能にするためには，ソースフィルタモデルのように音声生成の生理学・心理学的な構成要素との対応関係が分かりやすいことも重要になる．このような要請から，STRAIGHT は，必然的に channel vocoder[12] 型の構成を採用したのである．ただし，STRAIGHT の構成は，簡単に正弦波モデル (sinusoidal model)[25] として読み替えることもできる[2]．channel vocoder よりも更に音声生成過程に近いモデルを追求する方向も重要な研究課題ではある．しかし，STRAIGHT は，知覚の研究ツールを第一義に狙うため，生成モデルを採用することはしない．

1.1.1　STRAIGHT の履歴

STRAIGHT は最初の発明以来，多くの改造が重ねられて来た．それらの各段階で，技術的内容を発表して来たがその後の改良で捨てられた項目も多い．ここでは，STRAIGHT のスペクトル情報抽出部分，音源情報抽出部分，合成部分のそれぞれの履歴を簡単に説明する．更に詳しい履歴に関しては，論文の位置付けの節で紹介する．

---

[1] 分析合成型のシステムは，波形処理型のシステムに品質ではかなわないという通念がある．しかし，ソースフィルタ型のシステムである実際の人間の発声機構が自然な音声を生み出せるという反証の前には，この通念に説得力は無い．実際，信号の位相情報をほぼ完全に破壊する分析合成型のシステムである STRAIGHT は，十分に波形処理型のシステムと勝負のできる品質を，広範なパラメタ操作の下でも保つことができることを実証している．

[2] 議論を混乱させないために，本資料では，STRAIGHT の sinusoidal model としての読み替えについて詳しく議論することはしない．STRAIGHT が sinusoidal model として読み替え得るという議論は STRAIGHT の最初 [53, 23] から指摘しており，最初の特許でも触れている．知覚研究が目的であったため，STRAIGHT の研究の主流には置いていないが，CREST の枠組みで検討 [40, 41] は進められている．実は，声道共振の減衰と基本周期が同じオーダーとなる低い周波数の領域では，ソースフィルタモデルよりも sinusoidal model と読み替える方が適切なのである．また，高い周波数の領域では，後で述べる『もあれ』の影響もあって，sinusoidal model よりもソースフィルタモデルの方が適切である．更に，細かな議論をするなら，声門の開度に応じた声門インピーダンスの動的な変化による共振の減衰率の変化，音源と声道の相互作用の議論も行わなければならない．しかし，高品質な知覚研究用のツールを作るという立場に立てば，ソースフィルタモデルや sinusoidal model を使って知覚的に同等な音を作ることができるのであれば，これらの影響はモデルのパラメタに押し込めてしまうことで実用的には構わない．無論，モデルのパラメタにそれらのダイナミックなあるいは非線形な性質がどのように影響するかを研究することは，STRAIGHT とは別の研究として重要であろう．そのための分析ツールとして，STRAIGHT の構成要素のために開発された一群のアルゴリズムは非常に有用である．

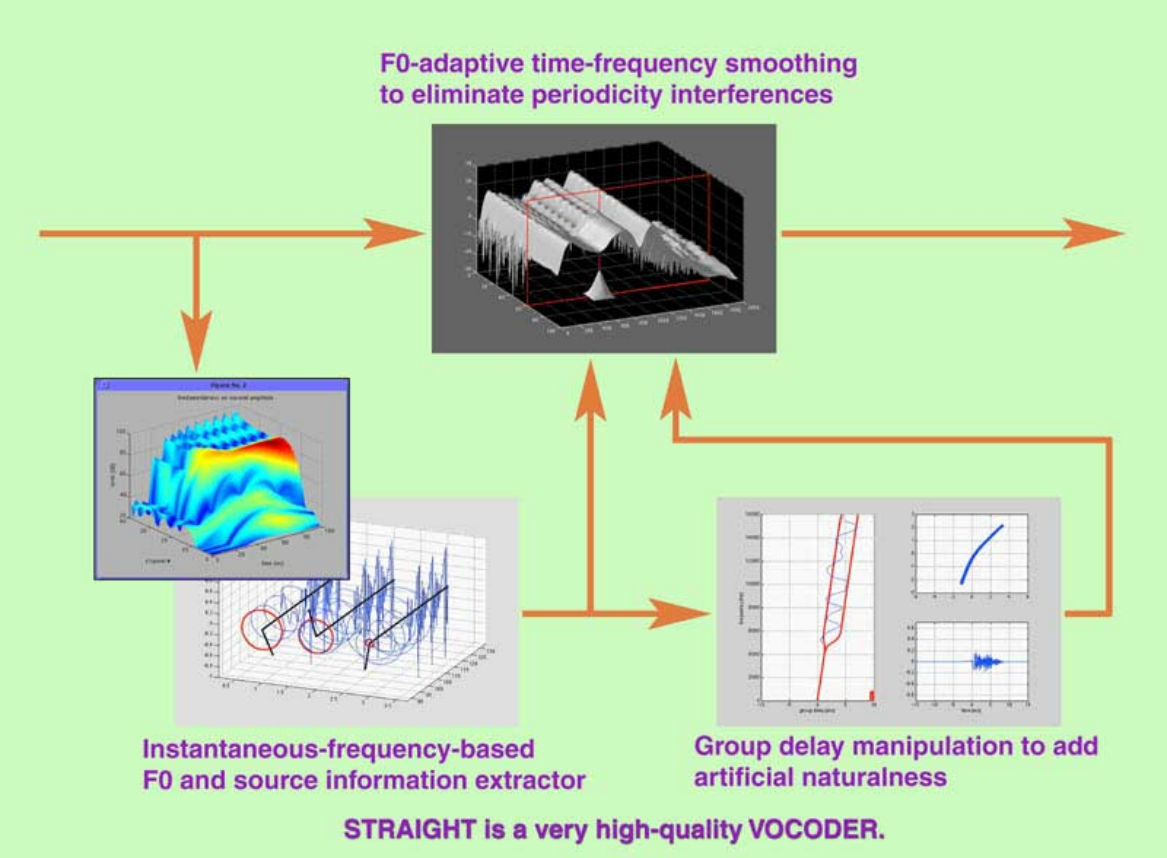

図 1.1: STRAIGHT の構成

スペクトル情報抽出　STRAIGHT の名称の由来となったスペクトル包絡抽出の最初のアイデアでは，時間-周波数領域での spline 基底を用いた二次元平滑化に基づいていた．このアイデアは，その後，時間方向の干渉を軽減した相補的な時間窓の利用による時間平滑化の省略，spline 関数の性質を利用した最適平滑化関数の導出を加えて現在に到っている．

音源情報抽　基本周波数および音源情報の抽出では，最初は変型自己相関法の一つに位置付けられる変形した嵯峨山法から出発した．この方法は，その後，瞬時周波数と『基本波らしさ』という指標を用いた TEMPO indexTEMPO によって置き換えられ，次いで，複数の調波成分を利用した推定値の改良が導入された．さらに，フィルタ中心周波数から出力の瞬時周波数への写像の不動点の性質を利用することで各調波成分の C/N を推定し，それらに基づいて調波情報を統合する現在の方法へと到っている．

合成　音声合成のための駆動音源の群遅延操作は，最小位相インパルス応答を用いた合成器で標本化周波数よりも高い分解能で基本周波数を制御するための機構の拡張機能として実現された．最初の段階では，空間周波数の制御が行われておらず，乱数そのものが用いられていた．ただし，制御を行う周波数領域を指定する周波数加重と群遅延の標準偏差の制御は行われており，1996 年 11 月には，現在と同様に，空間周波数の制御を加えたものとなっている．

1.2　STRAIGHT の構成

STRAIGHT の構成を図 1.1 に示す．図の左側からシステムに入力された音声は、音韻性に係わる成分と、韻律に係わる成分とに分解され、（必要であれば）変換を受けて、再合成される。図の中央上部には、音韻性に係わる成分（スペクトル包絡から構成される時間周波数表現）だけを分離抽出する機構、左下には、韻律等に係わる成分（基本周波数と有声／無声等の情報）を精密に抽出する機構、右下には、再合成した音声に自然さを与えるための音

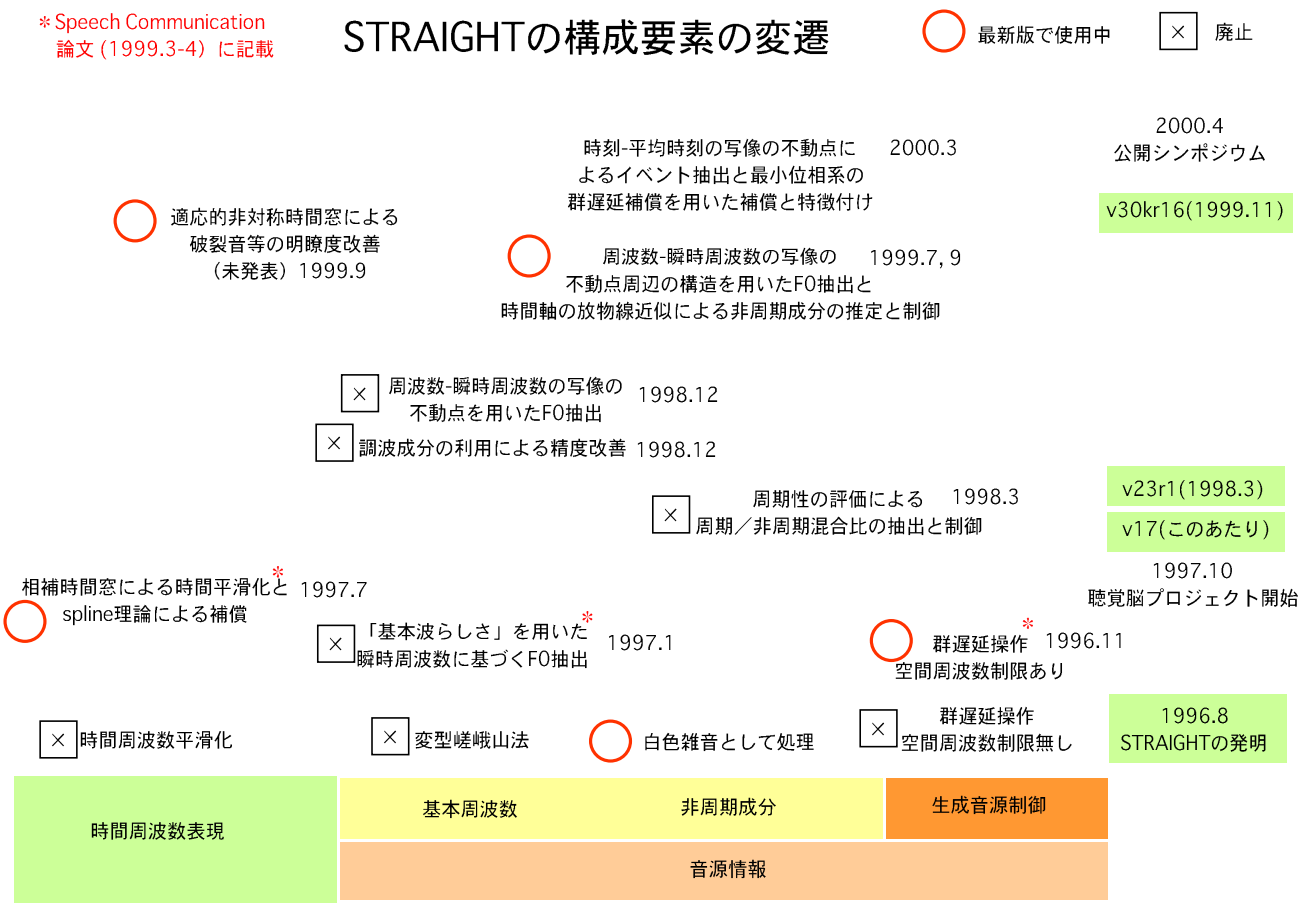

図 1.2: STRAIGHT の履歴と資料との関連

源信号の生成機構（群遅延操作音源）のそれぞれの動作原理を表すイメージを配している[3]。

STRAIGHT のこれらの構成要素は，全て科学技術計算のための強力な環境である Matlab の上に，マシンに依存しない形で実装されている．そのため，Matlab が走るマシンであれば，原則的には無修正で STRAIGHT を走らせることができる．STRAIGHT を走らせるために必要な最小限の Matlab システムの構成は，Matlab 本体と signal processing toolbox である．ここで説明する STRAIGHT の実装では，STRAIGHT の基本的な機能を簡単に使用できるように，Matlab 本体に含まれる GUI（グラフィカル・ユーザ・インタフェース）ツールを用いた GUI が与えられている．しかし，GUI はあくまでも簡便な利用のためのものであり，目的に応じた自由な利用のためには，アルゴリズムの詳細な理解が必要である．そのような目的で STRAIGHT を利用するために必要な Matlab の関数群についての情報も本資料に含まれている．

1.3 構成要素と公開資料・特許

STRAIGHT に関連した多数の資料がある．しかし，これまでは，それぞれの資料や特許がどのように STRAIGHT の現在の実装に関連しているのかを整理した資料はなかった．ここでは，それらを明らかにすることを狙い，STRAIGHT の構成要素と対応させながら，これまでに STRAIGHT に関連して発表してきた資料の位置付けについて説明する．

図 1.2 に STRAIGHT の履歴と主な資料の関連を示す．

[3] これらの要素は、現在も改良・改造が続けられているため，イメージ図の各要素は必ずしも現状の実装を正しく象徴してはいない。

1.3.1　時間周波数表現

STRAIGHT の発端は，音声のような周期的な信号のスペクトログラムに認められる周期性による干渉と包絡成分との分離についてのアイデアである．周期的な駆動を妨害と見ることを止め，時間周波数表現を組織的に標本化する手段であると視点を転換したのである．そうすると,『限られた標本点から元の時間周波数表現をいかにして復元するか』が解くべき問題となる．最初の STRAIGHT では，その問題を，標本点の間を双一次曲面で埋めるというアイデアで解くことを試みた．その具体化に，双一次曲面による補間と等価であり，数値的な頑健性とある種の保存則を満たすことのできる平滑化を用いたことも，重要なアイデアである．具体的には，基本周波数に適応した spline 基底による時間-周波数の平滑化を，非線形変換したスペクトログラムの上で行っている．この，1996 年 4 月頃の発想を特許として先行してまとめるとともに，聴覚・応用音響研究会において発表 [54] し特許出願 [53] した．また，国際的な優先権を主張するために，ICASSP'97 においても報告した [18]．なお，この時点で既に群遅延操作による駆動音源設計のアイデアが導入されている．ただし，群遅延の空間周波数制御という観点は，8 月の時点では十分に意識されてはいない．また，この時点では，基本周波数の抽出方法としては，スペクトルの割算に基づく自己相関法の一変種である嵯峨山による方法 [77] を変形して用いるに止まっていた．

スペクトログラムの計算に用いられる窓関数と，周期性を除去するための適応的平滑化の効果が重なることによる過剰平滑化の問題は，spline 関数の理論を用いることによって，解決された．この発明も特許化と連動して聴覚・音声研究会において報告された [22] また，その後，相補的な時間窓を用いてスペクトルを計算することにより，スペクトル包絡の計算における時間方向の平滑化を省いて計算量を削減することが可能となった [55]．これらは，より整理された形で Speech Communication の論文 [23] としてまとめられている．なおこの論文は，ESCA[4] （現 ISCA[5]）の上部組織である EURASIP[6] の 1998,1999 年度の最優秀論文賞を受賞している．

1.3.2　群遅延操作

合成音声の駆動音源を，通常のパルス列に代えて群遅延を操作することによって合成音に特有のバズ臭さを軽減するアイデアは，最初の STRAIGHT の提案 [54] に既に含まれていた．この群遅延操作の数理的整理と知覚的影響に関しては，最初の提案に続く検討結果が速い時期に聴覚研究会に報告 [56] されている．なお，STRAIGHT の検討にヒントを得て，濱上により，調波成分の位相を制御することで正弦波モデルの合成音声の品質を改善する方法が提案されている [78]．しかし，この方法は，本質的には STRAIGHT での提案の枠を拡大するものではない．また，パルス駆動と正弦波駆動の関係を表面的に誤って理解しており，位相よりも群遅延が本質的で操作し易い量であることも見落としている等の問題がある．

群遅延と音色の知覚の関連については，古くには，位相に関するものとしての Plomp の研究 [30] や電気音響変換システムの特性に関する Blauert による検討 [7] が行われている．STRAIGHT に関しては，津崎らによる一連の研究 [56, 36] と，符号化との関連も考慮した坂野らによる一連の研究 [6, 70] がある．また，Uppenkamph による最近の研究 [38, 37] は，群遅延の知覚が聴覚での情報処理機構についての重要な鍵となる手掛りをもたらすものであることを示唆している．信号の時間的構造と知覚との関連の深い理解は，より効果的な符号化方式の開発に繋がり得る．実際，この方向に向けて，音声の基本周期内部での時間周波数パターンとマスキングの関連に改めて関心が持たれようとしている [31]．

1.3.3　音源情報抽出

音源情報の抽出に関連する部分は，STRAIGHT の版による変化の最も大きい部分である．変型自己相関に関連する量から出発して，瞬時周波数 [8, 9, 11] に基づく二つの方法を提案している．また，時間軸の非線形変換を利用した周期成分と非周期成分の推定も導入されている．更に，次の版では，群遅延に基づく時間的イベントの特徴付けによる拡張が予定されている．

[4] European Speech Communication Association
[5] International Speech Communication Association
[6] The European Association for Speech, Signal and Image Processing

基本波らしさに基づく方法

最初の提案に引き続く早い時期に，合成音声の『よごれ』た印象が従来の基本周波数抽出法に起因するものであることを見い出し，基本波の瞬時周波数を用いた基本周波数抽出方法（TEMPO :Time-domain Excitation extraction based on a Minimum Perturbation Operator）を提案し [49] 特許出願 [47] した．前述の Speech Communication 誌の論文 [23] には，この基本周波数抽出法が用いられている．GUI-STRAIGHT の作成と同時に用意された操作説明書 [17] で紹介されている音源情報の抽出方法も，この TEMPO である．

しかし，TEMPO には，環境騒音により S/N の劣化しやすい基本波成分のみを利用しており，他の調波成分に含まれる基本周波数の情報を有効に利用していない等の弱点があった．そのため，以下に述べる方法に置き換えられており，現在の実装では用いられていない[7]．

混合スペクトルの試み

群遅延操作を用いても，有声／無声という二値的な判断に基づく音源を用いる場合には，鼻子音や母音の終了部分でのバズ音が耳につくという問題が残されていた．これを周期成分と非周期成分に対応するスペクトルの混合としてモデル化しようとした試みを報告している [58]．混合駆動音源を用いるというアイデア自体は，既に数多く報告されているが [16, 4, 13, 33, 39]，周期成分と非周期成分の推定の分散を考慮してスペクトルの推定に反映させるというアイデアは新しいものであった．しかし，この段階での混合スペクトルおよび音源モデルの試みでは他の理論的基盤が未整備だったため，高い周波数領域での『もあれ』の問題等が残っていた[8]．現在の STRAIGHT の実装では，次に説明する調波成分を利用した方法によって置き換えられている．

調波成分の利用

複数の調波成分を用いた基本周波数推定値の改良の試みは，二つのそれぞれ異なった方向から行われた．現在の実装では，調波成分抽出の副産物として，帯域毎の周期性／非周期性に関する情報が求められている．

複数の形状の analyzing wavelet の利用　最初の試みは，wavelet に基づく方法を，計算量を厭わずに，複数回，組織的にサイズを変更した analyzing wavelet を用いて適用することで改良値を求めようとするものであった [20]．この方法は，基本周波数の推定誤差の改善には有効であるものの，計算量の問題と対応点の抽出の難しさから，以下で説明する方法に置き換えられている．

周波数領域での写像の不動点に基づく方法　帯域フィルタの出力の瞬時周波数とフィルタの中心周波数との写像における不動点が正弦波成分の瞬時周波数の推定値を与えることに基づいた一連の方法が提案され [59, 52, 66, 21] それぞれの要素が特許出願 [48, 57] された．なお，周波数領域でこのような特徴的な写像が存在すること自体は，早い時期から気付かれていた [10, 1, 2, 3]．しかし，従来の方法では，正弦波成分の誤差を定量化する鍵となる指標を導くことができず，また適切な時間窓の設計法を欠いていたため，有効な基本周波数抽出法としてまとめられていなかったことを指摘しておく[9]．現在 STRAIGHT に用いられている方法は，写像の不動点近傍での形状と時間変化の情報からそれぞれの調波成分の C/N 比の推定法を与えることで，まず基本波成分の抽出を行い，その初期推定値を複数の調波成分からの情報を利用して改良するという二段階で構成されている [52, 66, 21]．さらに，この基本周波数の情報に基づいて時間軸の非線形伸縮を行うことで，音源の周期成分と非周期成分との推定が行われている[10]．

時間領域での写像の不動点に基づく方法　現在の STRAIGHT の実装に残される群遅延制御に関するマジックナンバーの自動的抽出を目的に，音響中のイベントとその駆動源の情報を抽出する一連の方法を発明し [63, 19, 50] 特

[7] 音声の録音状態が良い場合には，TEMPO も非常に高い品質の基本周波数情報を与える．また，後で説明する周波数領域の不動点を用いた方法も，第一段階では TEMPO の核となるアイデアと同型の機構を新しい情報表現用に再設計して利用している．

[8] GUI-STRAIGHT の作成と同時に用意された操作説明書 [17] では，この混合スペクトルモデルに基づいた説明が行われている．

[9] 音楽のように複数の調波複合音が重なった場合の基本周波数の推定には，瞬時周波数に基づいた複合音の構造のモデルを導入する方法 [71] が有効であろう．この方法は，繰返し演算を含む EM アルゴリズムを利用しているにも関わらず実時間動作するシステムとして実装されていることが注目される．

[10] この周期成分と非周期成分の推定は，1999 年 11 月から STRAIGHT に実装されていた．しかし，幾つかの資料 [52, 21] に断片的に説明があるだけで，まとまった記述としては，本資料のアルゴリズムの章が，初めての紹介である．

表 1.1: STRAIGHT の各版の主な特徴と評価.

項目	内容と説明				
STRAIGHT の版と時期	最初の版 1996 年 8 月	V17 1997 年 9 月	V23 1998 年 3 月	V30 1999 年 7 月	V30kr16 1999 年 11 月
F0 抽出	変型嵯峨山法	TEMPO		不動点	
有声／無声	二値		二値および連続量		連続量
非周期成分	-		帯域毎のピッチ相関		非線形時間軸補正と周期性評価
包絡抽出	時間-周波数平滑化	相補的時間窓，最適平滑化関数，時間領域補正			
群遅延操作	標準偏差，遷移周波数	標準偏差，遷移周波数，空間周波数			
操作性	コマンドライン		グラフィカルユーザインタフェース：GUI		
特許出願	ATR			ATR，科学技術振興事業団	
明瞭度	-			DRT	-
総合的品質	-	評価済	二値音源についてのみ評価済		-

許出願 [46] した.. 将来の版の STRAIGHT では，この方法により求められた音源情報が利用される予定である.

1.3.4 STRAIGHT の評価

STRAIGHT の品質は，何度も繰返し評価されている．しかし，STRAIGHT が常に改訂を繰返していたことと，評価の計画と実行に要する時間の食い違いから，評価された版は，常に数世代前のものであるという問題がある．最初の評価は，情報圧縮への応用を狙ってケプストラム vocoder との比較に基づいて，包絡表現の情報量削減と品質の関連を明らかにすることを狙って行われた [75]．その後も，奈良先端大の鹿野研究室等の協力により，一連の品質評価の研究が行われている [72, 44, 73]．また，これらの総合的な品質評価と並行して問題点の診断を目的とした DRT による評価が行われ，改良項目が明らかにされた [43]．これらの中から最近の評価をまとめたものが，国際会議に報告されている [42].

STRAIGHT の重要な構成要素である音源情報抽出部分については，EGG (Electro GlottoGram) と音声を同時収録したデータベースを用いた客観評価が行われている．TEMPO については，男女各一名づつのデータベースによる評価結果が前述の Speech Communication 誌の論文 [23] に載っている．また，周波数領域での不動点を用いた基本周波数の抽出と時間領域での不動点を用いた駆動情報の抽出においては，男女各 14 名がそれぞれ 30 文章を読み上げたデータベース [66, 67] を用いた評価が行われ，研究会 [63, 51] や国際会議 [19, 5] および論文 [67] において報告された.

様々な開発段階の STRAIGHT の版の概要と評価との関連を表 1.1 に示す．表から明らかなように，音源情報の精密化を実装した部分は未だに評価されていない．これまでの主観評価実験結果によれば，音源情報を二値として扱う限り，V17,V23,V30 の間に有意差は無い．しかし，音源情報の精密化による品質の向上は予備実験から明らかである．現状では，方式として決定すべきパラメタが多いため，これらの精密化の効果の本格的な評価実験は今後の予定のままである.

1.3.5 応用研究

STRAIGHT によるスペクトル包絡と基本周波数の独立の制御を応用した最初の報告は基本周波数と声道寸法を同時に変換した場合の変換の自然性と印象に関するものであった [60]．STRAIGHT の開発段階で評価を依頼した大学等の研究機関においても，様々な研究成果が発表され始めている．それらの幾つかを挙げると，母音の知覚特性 [26]，発話速度の変換による知覚の変化 [76]，音韻の局所的な時間長の変動の知覚特性，視覚と聴覚の時間処理の関連 [34, 69]，話者の変換 [24]，感情の知覚に関連するパラメタの探索等である [68].

1.3.6 解説・紹介

STRAIGHT には，様々のレベルでの紹介／解説がある．比較的初期の解説には，音響学会のもの [62] と音声学会のもの [61] がある．前者は聴覚の情景分析との関連と方式の位置付けを中心に，後者は音声知覚研究への応用の可能性を示すことを中心に書かれている．操作説明書としては，GUI-STRAIGHT の最初の版に対応するものが ATR の Technical Report として用意されている [17]．しかし，この資料は英文であり，現在の版ではかなりの部分が置き換えられている．

1.4 STRAIGHT の応用

前の節で紹介した応用に加え，STRAIGHT には広範な応用の可能性が開けている．まず，当初の設計目的に照らせば，fMRI 等の脳活動の計測手段を利用した研究のための刺激の作成に非常に大きな可能性があるものと考えられる．音声知覚能力の評価と補綴システムの開発研究への応用は，高齢人口の急速な増加を背景として，もう一つの重要な方向であると考えられる．また，もう少し実用的な方向には音声のフォントビジネスとでも呼べるような規則合成のための基盤システムの提供，音声を用いたユーザインタフェースへの感情やパラ言語情報の付与，外国語音声の訓練システム，音声のモーフィング [32] 等による既存コンテンツの音声部分の再利用，ゲーム等のキャラクタへの利用等，多くのの応用可能性があろう．

また，直接には STRAIGHT の応用とは言えないが，Vocoder 型の分析合成システムによって自然音声に匹敵する加工音声が実現できることを実証したことは，音声合成の研究を再度活発化させる効果をもたらしていることを指摘しておきたい．

第2章　STRAIGHTのアルゴリズム

この章では，STRAIGHT を構成する多数のアルゴリズムについて，理論的な背景を含めて説明する．STRAIGHT は，聴覚の計算理論に結び付けることを強く意識しているものの，変換音声の品質を最優先の評価基準としている．そのため，短時間フーリエ変換，時間周波数分析 [11]，wavelet 分析，spline 関数の理論，写像の不動点等の様々な理論やアイデアが無節操に用いられている印象を与えていることは否めない．しかし，最終的に聴覚の計算理論がまとめられれば，それぞれの位置付けが明らかになるはずのものである．ここでは，できるだけ無秩序な印象を与えないよう，それぞれの理論の選択の背景にも触れて説明するように努める．

2.1　はじめに

VOCODER 型のシステムで高品質の変換音声を作成するためには，以下の諸点に配慮することが必要である．

- 『信号の周期性に起因する時間周波数表現の周期的構造を選択的に除去すること.』分析合成だけが目的である場合には，この要請は不要である．元の信号と異なった基本周波数や発話速度，声道形状等に変換する場合，時間周波数表現に元の信号の基本周波数の影響が残っていると品質が劣化する．
- 『なめらかな基本周波数の軌跡が得られること.』分析合成だけが目的である場合には，この要請は不要である．自己相関等に基づく方法では，しばしば男性の低い声を分析したばあい，基本周波数に（例えば）階段状の不連続が生ずる．そのような音声の変換において高い基本周波数に変換すると，ジッタが知覚されて変換音声の品質が劣化する．
- 『標本化周波数の時間分解能に制限されない基本周波数の制御が可能であること.』音声の合成音源にパルスを用いる場合，基本周期の制御の分解能は，標本化周期で制限される．そのため，女性のように基本周波数が高い音声の場合，基本周波数が正確に再現されない問題がある．また，長時間平均が一致するように制御できてもジッタが生じて品質が劣化する．
- 『合成音声特有のバズ音が無いこと』合成音源にパルスを用いた場合，ヘッドフォンによる受聴では特有の『バス』音が耳につき品質が劣化する場合がある．

これらの要請が，現在の STRAIGHT に実装されている以下の３つの重要な構成要素の発明を導いた．それらは，(1) 基本周波数に適応した時間周波数平滑化，(2) 瞬時周波数に基づいた基本周波数の抽出方法，(3) 群遅延操作による駆動音源の作成，である．以下では，これらの構成要素で用いられているアルゴリズムについて説明する．

2.2　周期性に基づく時間周波数表現への干渉の除去

ある窓関数 $w(t)$ を用いて信号 $s(t)$ のスペクトログラムを求めると，信号に周期性がある場合には，時間方向もしくは周波数方向あるいは双方に窓関数と信号波形の干渉により周期的な変動が生ずる．この干渉の影響を除くことがこの節の課題である．

2.2.1 時間周波数平滑化：標本化としての周期性

STRAIGHT は，この周期性に起因する干渉を妨害ととらえず，むしろ音声の情報を表現する仮想的な滑らかな時間-周波数表現からの組織的な標本化によるものであると理解することから始まった[1].

ここで次のようなガウス型の時間窓 $w(t)$ を考えよう．この時間窓は，時間方向の分解能と周波数方向の分解能が同程度に設定してある．以下では，この窓を等方的時間窓と呼ぶことにする．

$$w(t) = \frac{1}{\tau_0} e^{-\pi(t/\tau_0)^2} \tag{2.1}$$

$$W(\omega) = \frac{\tau_0}{\sqrt{2\pi}} e^{-\pi(\omega/\omega_0)^2}\ , \tag{2.2}$$

where ここで $W(\omega)$ は $w(t)$ のフーリエ変換である．時定数 τ_0 の逆数を f_0 と表わし，対応する角周波数を表わすため $\omega_0 = 2\pi f_0$ という記号を用いた.

この等方的な時間窓を用いて周期が τ_0 であるような周期信号を分析することを想定する．得られる時間周波数表現には時間方向にも周波数方向にも周期的な構造が認められる．大まかな言い方をするなら，周波数方向では f_0 毎に，時間方向では τ_0 毎に，現象の背景にある滑らかな時間周波数表現に関する情報が標本化されていると見ることができる．このように解釈すると，周期信号から滑らかな時間周波数表現を求める問題は，標本値からの曲面の復元の問題となる．確率モデルに基づくパラメタ推定の問題と捉えるのではなく，確定的な関数近似の問題として捉えるのである.

問題をこのように捉えて,『元の曲面に関する強いモデルが利用できない場合には，できるだけ局所的な情報から構成することのできる関数を想定する』という一つの合理的な戦略を用いることとする．等間隔な格子に対するそのような関数の一つが spline 関数である．最も少ない格子情報から構成できる連続な spline 関数は 2 次のカーディナル B-spline 関数である．これは，区分的一次関数と呼ばれる関数でもある．STRAIGHT のもう一つの重要なアイデアは，格子点の抽出と補間操作という数値的に脆弱な操作を，2 次の spline 基底関数を用いた平滑化という等価で頑健な操作に置換えたことである[2].

ところで，実際の音声の場合，基本周期 $(\tau_0(t) = 2\pi/\omega_0(t))$ は時間と伴に変化する．従って，分析のための窓関数も基本周期に対して適応的に変化することとなる．混乱の生じない限り，表記を簡潔にするために時間の関数であることを示す (t) を省くこととする.

この段階までの STRAIGHT における基本周波数に適応した平滑化スペクトログラムの計算は，以下のようにまとめることができる.

先に説明したような時間周波数曲面を周期信号によって 2 次元で標本化された情報から復元するための 2 次元の平滑化関数 $h_t(\lambda, \tau)$ は，次のように二つの 2 次のカーディナル B-spline の基底関数を用いて構成される．ここで λ は周波数，τ は時間を表わす.

$$\begin{aligned} h_t(\lambda,\tau) &= \frac{1}{4}(1-|\lambda/\omega_0(t)|)(1-|\tau/\tau_0(t)|) \\ &\quad \text{where}\ \ \omega_0(t) = 2\pi f_0(t),\ \ \text{and} \\ &\quad [-\omega_0(t) \le \lambda \le \omega_0(t),\ -\tau_0(t) \le \tau \le \tau_0(t)] \end{aligned} \tag{2.3}$$

ここで $\omega_0(t) = 2\pi f_0(t)$ であり，$[-\omega_0(t) \le \lambda \le \omega_0(t),\ -\tau_0(t) \le \tau \le \tau_0(t)]$ である.

この平滑化関数を用いることで，基本周波数に適応した平滑化スペクトログラムは，以下のように計算される.

$$S(\omega, t) = g^{-1}\left(\iint_D h_t(\lambda,\tau) g(|F(\omega-\lambda, t-\tau)|^2) d\lambda d\tau \right) \tag{2.4}$$

[1] このアイデアは，図らずも，最近の入野-Patterson による安定化 wavelet-Mellin 変換の位置付けについての議論 [35] を先取りするものであった．これは，この方向に一つの聴覚の計算理論がありそうだという予想を補強するものである．Patterson は，以前から「音の周期は神経系の動作速度と比較すると速過ぎる，周期的信号をストロボスコープのように安定化させてゆっくりとした動きに変えて特徴を捉え易くする機構があるはずだ.」というアイデアにこだわって，数多くのモデルを提案している [28, 29]．STRAIGHT の作者も,「周期性のある有声音の方が同じ母音でも無声音よりも滑らかに豊かに聞こえる．音声スペクトルの確率モデルは非常な成功をおさめているが，この有声音のなめらかな知覚を説明できない点において，何か欠けている視点が残されているはずだ.」という疑問に永年こだわって来た [64]．根底にある問題意識に共通しているものがあるので，到達点が収斂して行くのも当然かも知れない．なお，周期性による推定値のバイアスを，駆動信号の周期性を仮定して補償しようとする試み [14] は，実際の音声が厳密には AR モデルで表現できないため，実音声に適用すると破綻する．STRAIGHT の初期の段階で論文の追試を含めて検討し，実際に破綻することを確認したが，資料としては残していない．現在では，修士の演習課題程度の問題であろう.

[2] 時間周波数の情報が格子点に δ 関数を配置して与えられている場合には等価になるが，そうでない場合には，標本化情報の広がりが問題になる．これについては，最適化平滑化の節で議論する.

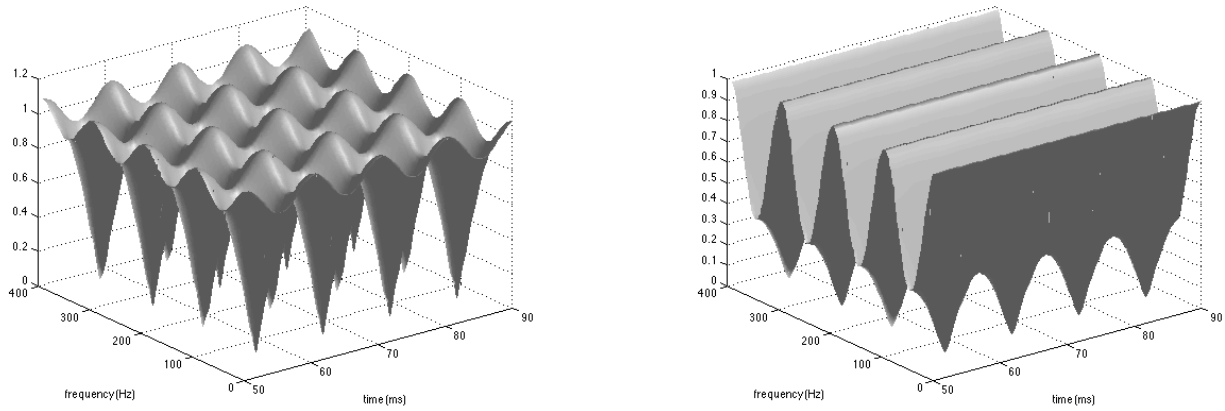

図 2.1: 等方的時間窓によるパルス列のスペクトログラム（左）とピッチ同期処理を導入した場合のスペクトログラム（右）.

ここで D は，平滑化関数 $h_t(\lambda,\tau)$ の定義域を表わす．また，$|F(\omega,t)|^2$ は，先に示した時間窓を用いたスペクトログラムである[3].

式 2.4 では，これまで触れなかった要素 $g(\ )$ が導入されている．この $g(\ )$ は，聴覚の各チャンネルからの情報統合における非線形性を近似するためのものである．非線形性が無い場合には，$g(x)=x$ という恒等写像となる．ラウドネス感覚の 1/3 乗則を $g(x)=x^{1/3}$ として取り入れれば，近似的に知覚されるラウドネスを保存するような平滑化を行うことができる[4].

2.2.2 相補的時間窓と最適平滑化関数

1996 年の STRAIGHT は，文字通り時間周波数領域での二次元の平滑化を行っていた．しかし，計算量が多い割にその効果は顕著ではなく，窓関数と平滑化が重なることによるの問題が残されていた．この問題を解決するために，二つのアイデアが用いられた．一つは時間方向における周期性の干渉の影響の少ない窓の採用であり，他は spline 関数の理論を利用した最適平滑化関数の導出である．以下では，それぞれについて順を追って説明する．

相補的時間窓

等方的時間窓を用いて周期的なパルス列のスペクトログラムを求めると図 2.1 の左の図に示すように，時間と周波数の双方向に規則的に零のあるものが得られる．

ここで，時間方向の周期性による干渉を除くため，等方的な時間窓に次式に示すような基本周期に適応した spline 基底関数 $h(t)$ を畳込んで同期化時間窓 $w_p(t)$ を作成することとする．

$$\begin{aligned} w_p(t) &= e^{-\pi\left(\frac{t}{\eta t_0}\right)^2} \odot h(t/t0) \\ h(t) &= \begin{cases} 1-|t| & |t|<1 \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{2.5}$$

ここで，$\odot$ は畳込を表わしている．この畳込みにより，図 2.1 の右の図に示すように，ピーク部分の干渉は除かれ，谷の部分に周期的な干渉の影響が残るだけとなっている．ここで新たな係数 η を導入した．この値が 1 であれば，ガウス関数の部分は等方的時間窓と同じものとなる．η が大きくなると，時間方向の分解能が低下する代償として隣接調波からの干渉が少なくなり，谷が深くなる．

[3] ここでは一般的なスペクトログラムの定義と整合性を保つために以前の論文のように平方根を用いることを止めた.

[4] この論理は，論文でこの非線形演算を正当化するために用いられたものである．しかし，実際には，平滑化の操作によってピークの鋭さが鈍ってしまうという問題を軽減するために，この非線形性は用いられている．恒等写像を用いた場合には，明らかに合成音声が曇った響きになってしまう.

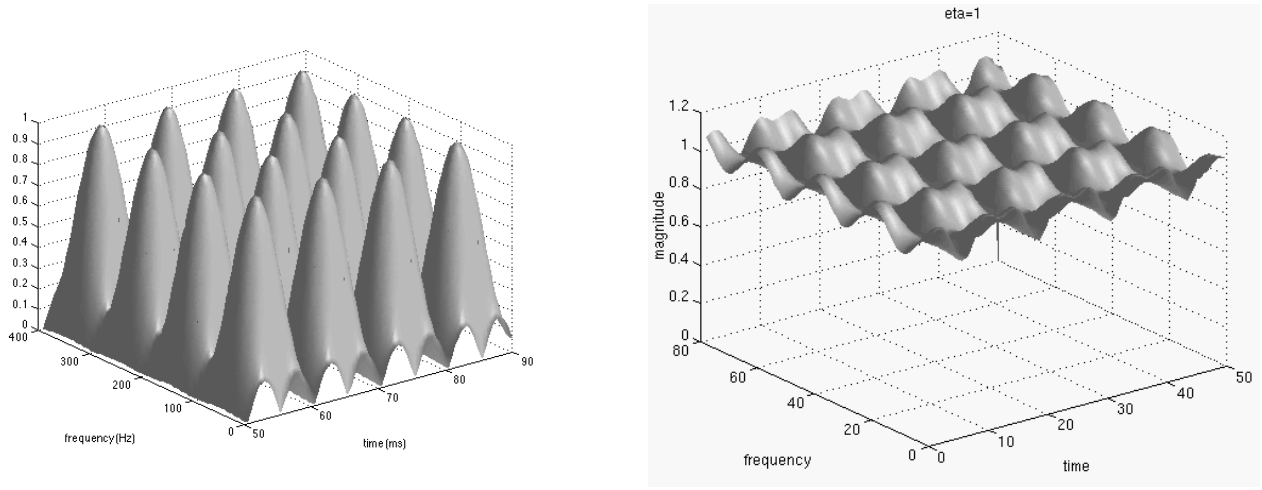

図 2.2: 相補的時間窓によるパルス列のスペクトログラム（左）と $\eta = 1$ の場合に最適に合成されたスペクトログラム（右）.

次に必要となるのは，このスペクトログラムのピークの部分が零となり，谷の部分の干渉による零の部分にピークがあるような相補的なスペクトログラムである．これは，次のように基本周波数の半分の周波数を有する正弦波を掛けた時間窓 $w_c(t)$ を作ることにより実現できる．

$$w_c(t) = w_p(t) \sin\left(\pi \frac{t}{t_0}\right) \ . \tag{2.6}$$

この時間窓を用いてスペクトログラムを求めると，図 2.2 の左の図のように，図 2.1 の右の図の零があった部分にピークのあるスペクトログラムが得られる．

こうして得られた相補的スペクトログラム $P_c^2(\omega, t)$ と同期化スペクトログラム $P_o^2(\omega, t)$ を，最終的な干渉が最少となるように加重 ξ を調整して平滑化スペクトログラムを得る．

$$P_r(\omega, t) = \sqrt{P_o^2(\omega, t) + \xi P_c^2(\omega, t)} \ , \tag{2.7}$$

図 2.2 の右側の図は，$\eta = 1$ の窓を用いた場合の最適なスペクトログラムである．しかし，この図では，干渉は少なくなっているものの，格子状の規則的な構造が認められる．

前に導入した η を用いることで，時間分解能の低下を代償に，干渉の影響を実用的に支障のない程度にすることができる．このようにすると，最適な合成のための加重は η の関数 $\xi\eta$ となる．

$$P_r(\omega, t, \eta) = \sqrt{P_o^2(\omega, t, \eta) + \xi(\eta) P_c^2(\omega, t, \eta)} \ , \tag{2.8}$$

η に対する最適な合成のための加重 $\xi\eta$ の変化を図 2.3 の左側に示す．これは，等間隔のパルス列について求めた合成スペクトログラムの時間方向の標準偏差を最少とするという基準の下で，数値的最適化計算法によって求めた値である．図 2.3 の右側には，$\eta = 1.4$ としてこの最適な混合比（$\xi(1.4) = 0.43$）を用いて合成したスペクトログラムを示す．周波数方向の周期的な構造は残っているが，時間方向の干渉が実用的な意味で取り除かれていることが分かる．現在の STRAIGHT の既定値では，$\eta = 1.4$ が用いられている．

このような時間方向の干渉の影響を軽減したスペクトログラムの導入は，最初の STRAIGHT の定式化で必要であった時間方向の平滑化を不要とする．また，時間方向の平滑化を実装する場合に必要だった，分析フレームの細かな刻みに対する要求も軽減される．これらは，演算量の削減に大きな効果がある．

最適平滑化関数

前の節で説明した特殊な時間窓を利用することによって，時間方向の平滑化の必要は無くなり，必要な処理は周波数方向の一次元の適応的平滑化だけとなった．ここで，これまで無視して来た過剰平滑化の問題を取り上げ，その解決策を提案する．

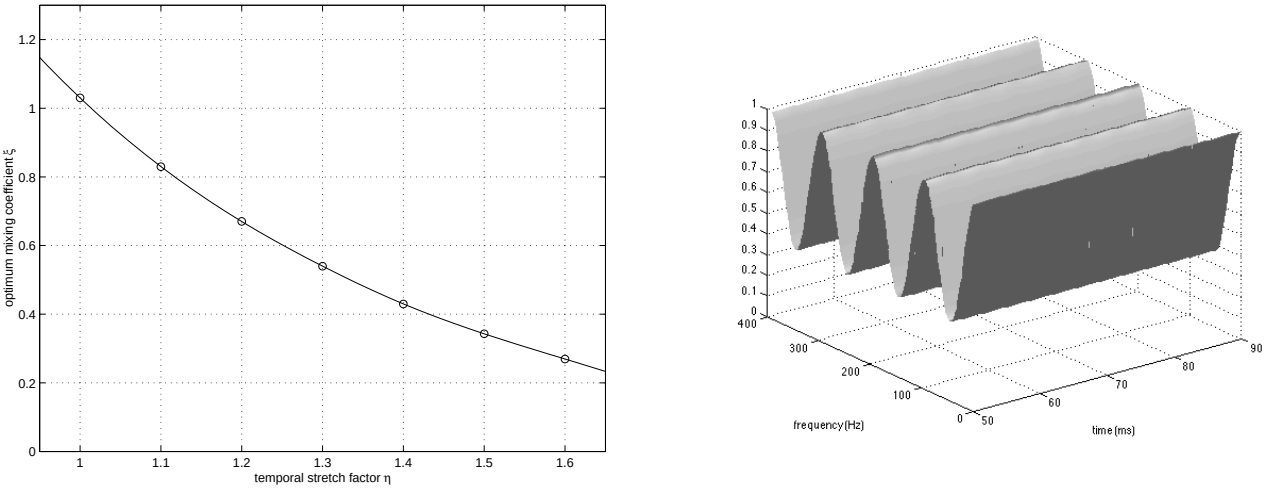

図 2.3: η に対する加重 $\xi(\eta)$ の最適値（左）と $\eta = 1.4$ の場合に最適に合成されたスペクトログラム（右）.

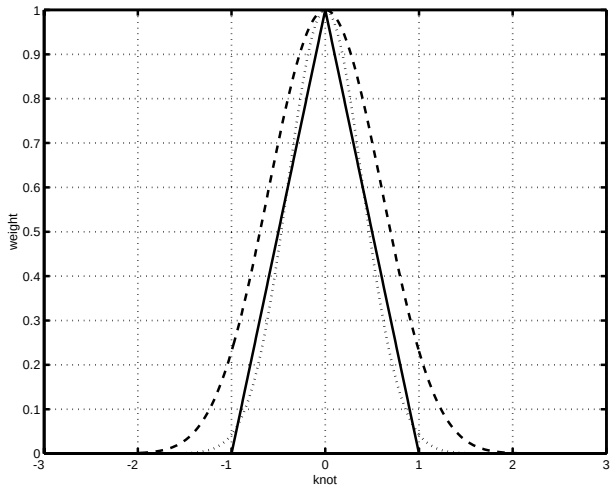

図 2.4: 過剰平滑化の例．平滑化の結果として得られるべき 2 次のカーディナル　B-spline 基底（実線）が，時間窓のフーリエ変換が有する周波数方向の拡がり（点線）と重なることにより，破線で示すように変型してしまう.

STRAIGHT の最初のアイデアでは，格子点で情報が与えられた場合に区分的一次関数による補間と 2 次のカーディナル B-spline 基底による平滑化が等価であるという性質を用いていた．しかし，実際のスペクトログラムにおいては，格子点の情報は，既に窓関数の形状によって定まる時間周波数領域での広がりによってある程度の平滑化が既に行われてしまっている．したがって，そこで更に干渉を除去するための平滑化を行うと，過剰に平滑化が行われてしまうことになる．前の節の時間窓を利用することによって 2 次元の問題から 1 次元の問題に簡単化されても，周波数方向ではやはり過剰平滑化の問題が残ることに変わりはない．この 1 次元での過剰平滑化の様子を，図に示す.

過剰平滑化は，前に求めた窓関数のフーリエ変換 $W(\omega)$ と平滑化関数 $h(\omega)$ を用いて表わすと，それらの畳込みとして次式で求められ関数 $v(\omega)$ が一般には $v(\omega) = h(\omega)$ という関係を満たさないこととして定式化される.

$$v(\omega) = \int_{-\infty}^{\infty} W(\omega - \lambda) h(\lambda) d\lambda \tag{2.9}$$

ここで，spline 補間のふるまいが節点の値だけによって決まることに注意すると，それらは，ω_0 の間隔で取り出した値から構成される離散的な値の間 $v_k = v(k\omega_0), k \in Z$ の問題となる．すると，過剰平滑化の問題を解決するに

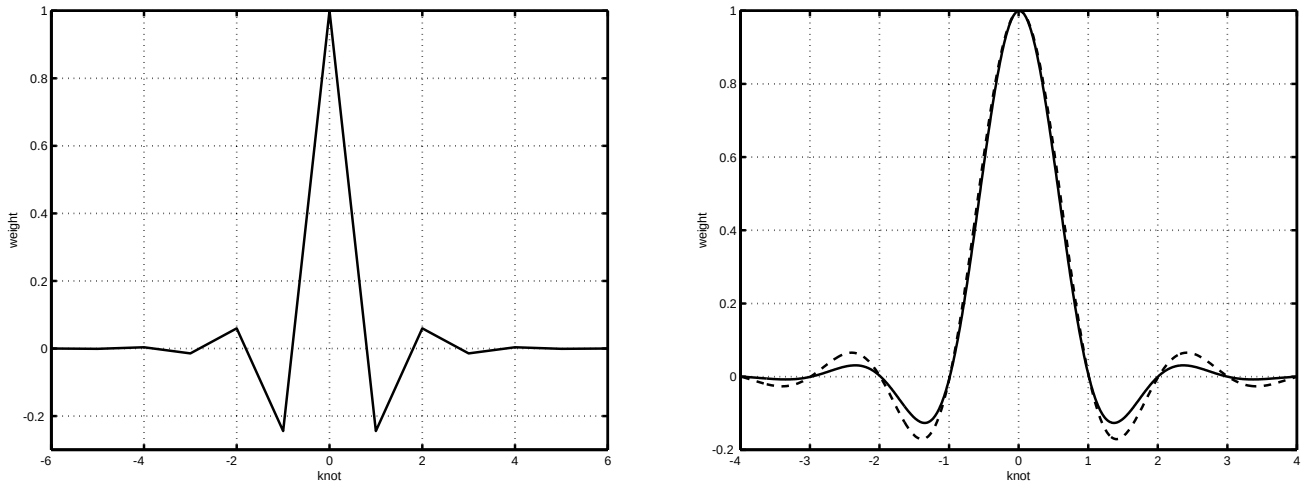

図 2.5: 最適平滑化関数（左）と窓関数と最適平滑化関数の畳込み（右）

は，以下のような性質を持つ係数の組 $c_k, k = -N, \ldots, N$ を求めれば良いことになる．

$$u_l = \sum_{k=-N}^{k=N} c_k v_{l-k} \tag{2.10}$$

where

$$u_l = \begin{cases} 1 & (\ l = 0) \\ 0 & (\text{otherwise}) \end{cases} \tag{2.11}$$

ここで，N は，v_N を実質的に 0 と看做すことができるような適当に大きな整数である．これは，簡単に言ってしまえば，v_k をインパルスに変換する様な逆フィルタを作成することに他ならない．

議論を見易くするために，ベクトル表記を用いることにする．問題は，次を満たす $\mathbf{c}$ を求めることに他ならない．

$$\mathbf{u} = \mathrm{H}\mathbf{c} \tag{2.12}$$

ここで

$$\mathbf{u} = [u_{-M}, u_{-M+1}, \ldots, u_0, \ldots, u_{M-1}, u_M]'$$

$$\mathbf{c} = [c_{-N}, c_{-N+1}, \ldots, c_0, \ldots, c_{N-1}, c_N]'$$

$$\mathrm{H} = \begin{array}{c} \\ -M \\ \vdots \\ k \\ \vdots \\ M \end{array} \begin{array}{c} \begin{array}{ccccc} -N & \cdots & l & \cdots & N \end{array} \\ \begin{pmatrix} v_{-N-M} & \cdots & v_{l-M} & \cdots & v_{N-M} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ v_{k-N} & \vdots & v_{k+l} & \vdots & v_{k+N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ v_{-N+M} & \cdots & v_{l+M} & \cdots & v_{N+M} \end{pmatrix} \end{array}$$

ここで $[\ ]'$ は行列の転置を表わす．この解は，以下のようにして求められる．

$$\mathbf{c} = (\mathrm{H}'\mathrm{H})^{-1}\mathrm{H}'\mathbf{u} \tag{2.13}$$

図 2.5 にこうして求めた最適平滑化関数と，窓関数との畳込みの結果を示す．狙った通り，畳込みの結果は，一個の節点を除き，値が 0 となっていることが分かる．ただし，その代償として，最適平滑化関数の定義域が最初の平滑化関数の 3 倍以上になってしまっている．また，節点以外では，負の応答を示す部分も出て来てしまっている．これらについては，次の節で議論する．

2.2.3 実装に際しての注意

最適平滑化のアイデアは，過剰平滑化の問題を解決するが，離散的な節点のみの制御であるため，新たな問題が入って来てしまう．また，議論を簡単にするために非線形処理をした領域で平滑化を行った効果を無視していた．こ

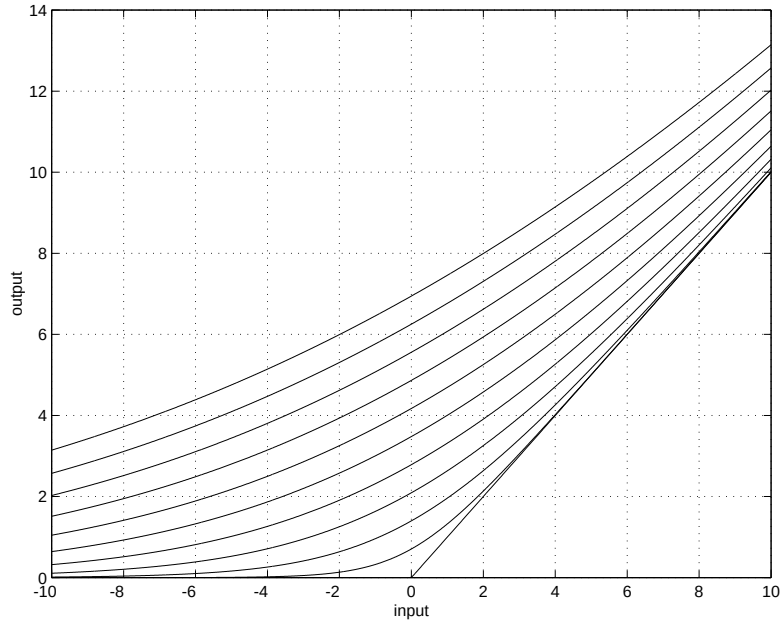

図 2.6: 滑らかな半波整流関数．参考のために，半波整流関数も併せて示す．

れらについて簡単に触れる．

節点以外での振舞　図 2.5 の右側に示されているように，節点の間の値は，ここで提案した方法では制御できない．特に問題となるのは，負の値を示す部分が生ずることである．後で説明するように，STRAIGHT の合成部分では，複素ケプストラムを介して最小位相のインパルス応答を求めている．この段階でスペクトルに負の部分があると，インパルス応答に異常が生ずる．

これを根本的に避ける方法は無い．現実的な対策としては，滑らかな半波整流関数を利用することである．そのような関数 $r(x)$ としては，次のようなものを挙げることができる．

$$r(x) = \beta \log(e^{\frac{x}{\beta}} + 1) \tag{2.14}$$

パラメタ β を 1 から 10 まで 1 刻みで変えたときの $r(x)$ の振舞を図に示す．

実際の音声のパワースペクトルのダイナミックレンジは 60 dB 以上に及ぶことがある．そのため，この滑らかな半波整流関数にパワースペクトルをそのまま入力したのでは，目的とする効果は得られない．STRAIGHT の現在の実装では，周波数軸上で局所的にパワーを正規化した後に最適平滑化関数を畳込み，滑らかな半波整流関数で負の値の処理をした後，正規化に用いたパワーを再び掛けて最終的な平滑化スペクトルを得ている[5]．

非線形処理の影響　平滑化は，式 2.4 に示すように，非線形変換を行った領域において行われる．従って，最適平滑化関数も，窓関数のフーリエ変換を非線形変換したものを対象として求めなければならない．

2.2.4 インパルス応答の時間領域での補正

音声波形の全ての部分が同様に知覚に影響を与える訳では無い．特に，男性のような低い基本周波数の音声の場合，周期中の位置によって，高い周波数成分を有する雑音バーストへのマスキング量が数十 dB のオーダーで変化する[6] ことが注目されている [31]．

[5]実装としては，この通りである．しかし，正規化という処理が人工的であり，場当たり的な印象を与える.. 考え方としては，正規化せずに最適平滑化関数を畳込んだ後に，局所的なレベルに応じて β の値が変化するとした方が見通しが良い．ただし，実現される性能としては，余り変わりは無いと考えられる．

[6]この現象に関する筆者による断片的な検討の報告がある．しかし，16 年前に着手されたこの検討は，人事異動により中断したままである．

逆に見ると，音声波形の中で，知覚に影響するのは大きな振幅を示す部分の周辺であるから，その部分の波形の性質を局所的に加工することで品質を向上できる可能性があることになる．

2.2.5 破裂音のための時間的包絡の補償

STRAIGHT の DRT による評価から，幾つかの問題点が指摘された．その中の一つに，無声破裂音における異聴の増加があった．これは，それ以前にもインフォーマルな形で指摘されていた「/k/等の無声破裂音が甘くなる印象がある」という問題点にも対応する．

現在の版では，この問題の対策として，信号の性質に応じて包絡の計算のための窓の形状を適応的に変化させる方策を取っている．この方策はアドホックなプログラムとして作成されているため，詳細は実装の部分で説明する．

2.3 瞬時周波数に基づいた基本周波数の抽出方法

周期信号の定義は，常に同じ波形が繰返されることを要請している．一方，音声の基本周波数は常に変化し続ける．このような信号に対して周期信号の数学的定義をそのまま当てはめることはできない．瞬時周波数は，このように時間とともに変化する信号を表現する自然な方法である．

時間とともに変化する瞬時角周波数を $\omega_i(t)$ と表わすこととする．また，瞬時振幅を $a(t)$ とする．すると，これらから次のようにして信号 $s(t)$ が作られる．

$$s(t) = a(t)e^{\int_0^t \omega_i(\tau)d\tau + \phi(0)} \tag{2.15}$$

なお，ここで，$a(t)$ と $\omega_i(t)$ は，$s(t)$ と比較すると，十分に遅い速度で変化するものと仮定する．

このような信号が複数個加え合わされて音声信号が出来上がっているものとする．先ほどの瞬時周波数，瞬時振幅と初期位相に成分を表わす添字 k をつけて表わすこととする．

$$s(t) = \sum_{k=1}^{N} a_k(t)e^{\int_0^t \omega_k(\tau)d\tau + \phi_k(0)} \tag{2.16}$$

有声音の場合には，それぞれの $\omega_k(t)$ は，共通の基本（角）周波数 $\omega_0(t)$ と次のような関係にあると仮定する．

$$\omega_k(t) = k\omega_o(t) + \varepsilon_k(t), \quad k = 1, \ldots, N \tag{2.17}$$

ここで $\varepsilon_k(t)$ は，小さなずれを表わすために導入した．

そうすると，信号が与えられた時にそこから共通する成分としての基本周波数 $f_0(t) = \omega_0(t)/2\pi$ を求めることが解くべき問題となる．最初に紹介する TEMPO と呼ばれる方法では，$k = 1$ の成分の瞬時周波数を基本周波数の推定値として用いる．次に紹介する方法では，複数の k に属する調波成分の瞬時周波数を統合して基本周波数を求める．そのため，二番目の方法では，フィルタの中心周波数から瞬時周波数への写像の不動点の性質を利用して適切な統合のための加重を設定する方法を発明している．

2.3.1 TEMPO: 『基本波らしさ』を用いた方法

式 2.16 のモデルを用いると，$k = 1$ の成分の瞬時周波数が求められれば基本周波数の良い推定値が得られることが分かる．しかし，問題がある．第一次の調波成分の選択には，その成分に同調したフィルタが必要であり，そのようなフィルタを用意するためには，基本周波数が予め知られていることが必要であるという，矛盾が含まれているように見えるからである．この見かけ上の矛盾は，次のようにして解決される．

まず，ここでは，十分に密にフィルタを用意すれば，その中には，基本周波数成分だけを含むようなフィルタが必ず見つかると仮定する[7]．すると，問題は，多数の帯域通過フィルタを通過した信号の中から，基本周波数成分だけを含んでいるような最良のフィルタ出力を見つけだす問題となる．

[7]当然生ずる疑問は，基本波成分が欠落した信号への対応である．しかし，現実的には音声では基本周波数成分が欠落することは無い．また，以下で紹介する方法は，適切にフィルタを設計することにより，例えば電話音声のようにそれほど鋭く無い高域通過フィルタによって処理された音声の場合には，十分に基本周波数を抽出することができる．

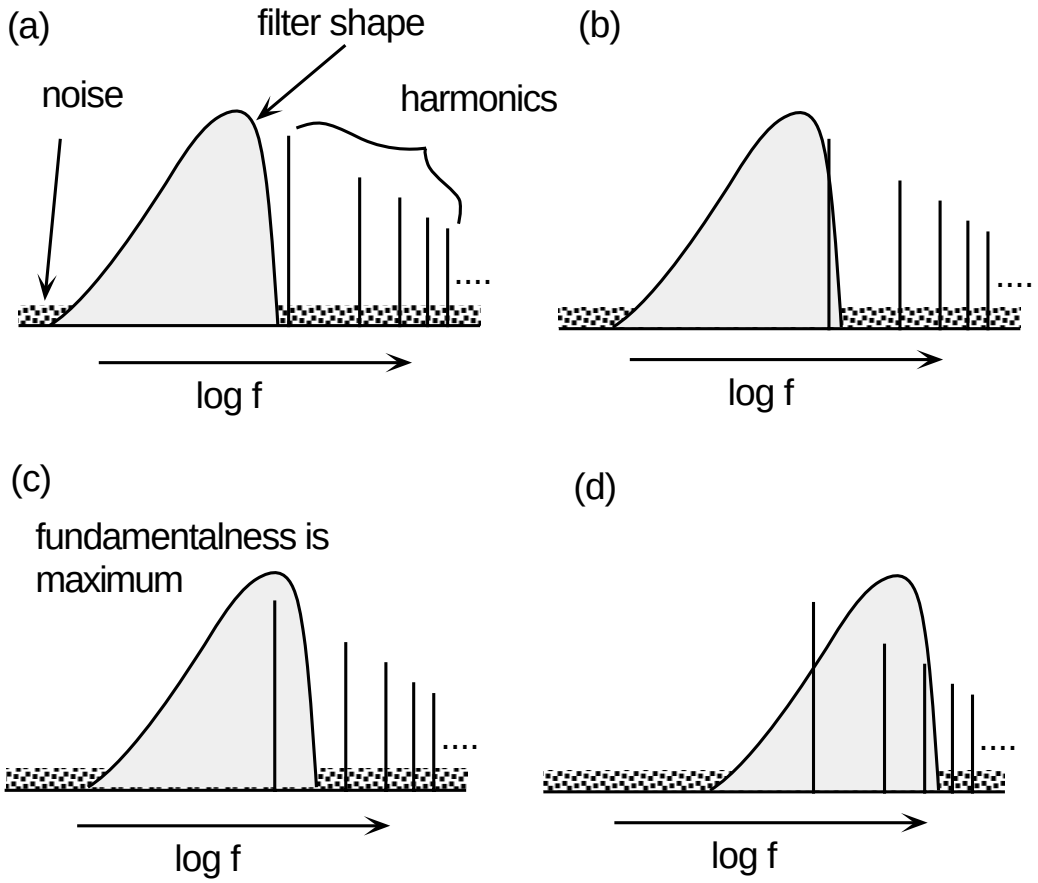

図 2.7: フィルタ設計法と『基本波らしさ』の模式図．(a) フィルタがいずれの調波成分も通過域内に含まない場合．(b) 基本波成分だけが通過域の端に含まれる場合．(c) 基本波成分がフィルタの最適周波数に一致し，しかも他の調波成分が通過域に含まれない場合．(d) 高次の調波成分と隣接する幾つかの調波成分がフィルタの通過域に含まれる場合．

この問題は，二つのアイデアを組み合わせることで解決される．一つは，フィルタの中心周波数が複合音の基本周波数とほぼ同じ場合にだけ出力に含まれる調波成分が一個となるように設計されたフィルタの利用である．もう一つは，フィルタの出力に基本周波数成分だけが含まれる時に最大となるようなフィルタ出力の評価尺度の利用である．

これらのアイデアは，次のようにして実現できる．まず，図 2.7 を用いて定性的に説明する．(a) は，複合音の成分を含まないような帯域を通過帯域とするようなフィルタの場合を表わす．この場合には，通過帯域内部には背景雑音が入るのみである．したがって，出力の瞬時周波数や瞬時振幅は常に変動し，相対的な変動の大きさも大きい．(b) は，通過帯域の一部に調波成分が含まれる状態である．この場合，一部に含まれるという状況が生ずるのは，その成分が基本波成分の場合のみである．この場合，出力の瞬時周波数はその基本波の瞬時周波数にほぼ支配され，瞬時振幅も同様にその成分の振幅にほぼ支配される．しかし，この場合，基本波成分は最大の利得でフィルタを通過する訳ではないので，瞬時周波数や瞬時振幅の変動は最小にはならない．(c) は，フィルタの最適周波数と基本周波数が一致する場合である．この場合は，(b) と同様であるが，基本波成分が最大の利得でフィルタを通過するので，瞬時周波数や瞬時振幅の変動は最小になる．なお，この時，フィルタの通過帯域の中には第二調波等，基本波成分以外の調波成分が入らない形状にフィルタを設計しておく．(d) は，フィルタの通過域に基本波成分以外の調波成分も含まれて来る場合である．(c) で説明したような形のフィルタを用いる場合には，フィルタの中心周波数が基本波の周波数よりも高い値となっている場合が相当する．この場合，通過域内に同時に存在する複数の調波成分の干渉により，出力の瞬時周波数と瞬時振幅には，大きな相対的変動が生ずる．

フィルタは，具体的には，以下のようにして構成できる．等方的な Gabor$w_G(t)$ 関数から出発して，調波成分の

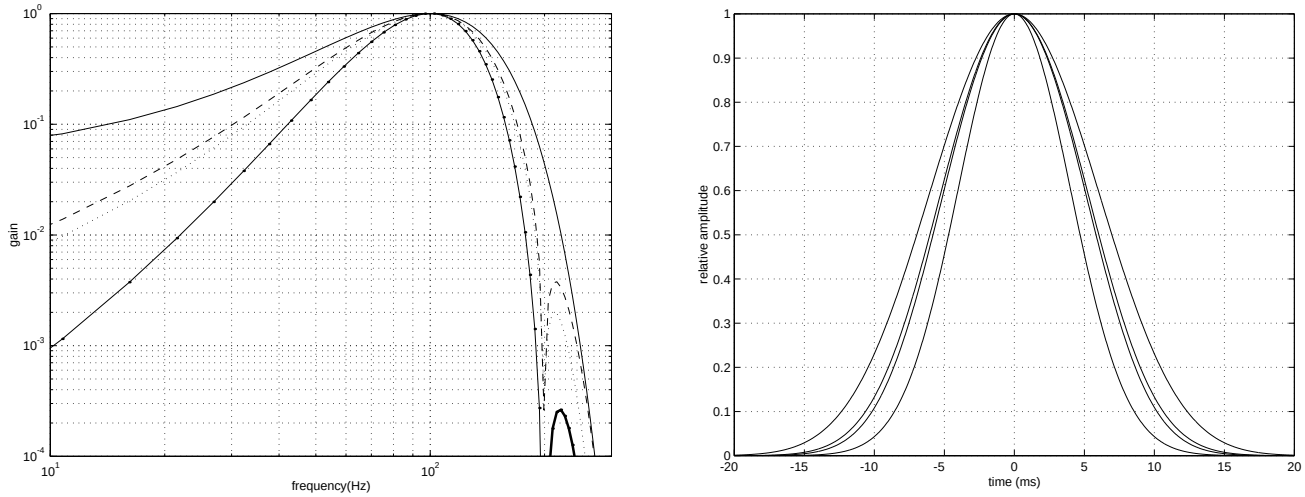

図 2.8: Gabor 関数と派生したフィルタ関数の振幅特性（左）と時間包絡特性（右）．振幅特性は，上から順に Gabor 関数 $w_G(t), w_{G0}(t), w_{G1}(t), w_{G2}(t)$ を表す．時間包絡と特性は，下から順に Gabor 関数 $w_G(t), w_{G0}(t), w_{G1}(t), w_{G2}(t)$ を表す．

位置にゼロを置くことで様々な変型ができる．

$$
\begin{aligned}
w_G(t) &= e^{-\pi \frac{t^2}{\eta t_0^2}} e^{2\pi t} && (2.18)\\
w_{G0}(t) &= w_G(t + \frac{t_0}{4}) - w_G(t - \frac{t_0}{4}) && (2.19)\\
w_{G1}(t) &= w_G(t) \odot h_1(t) && (2.20)\\
w_{G2}(t) &= w_G(t) \odot h_2(t) && (2.21)
\end{aligned}
$$

ここで

$$
\begin{aligned}
h_1(t) &= \begin{cases} 1 & |t| < \frac{t_0}{2} \\ 0 & \text{その他の場合} \end{cases} \\
h_2(t) &= \begin{cases} (1 - \frac{|t|}{t_0}) e^{2\pi t} & |t| < 1 \\ 0 & \text{その他の場合} \end{cases}
\end{aligned}
$$

元となる Gabor 関数 $w_G(t)$ を出発点として，$w_{G0}(t)$ は，原点と 2 次調波の位置に零を置くことを狙って設計した関数であり，副次的に他の偶数調波位置に零が置かれる．$w_{G1}(t)$ は，基本波以外の全ての調波位置に零を置くように設計した関数であり，副次的に，基本波からの距離に反比例した利得の減少が組み込まれている．$w_{G2}(t)$ は，利得の減少の傾向が距離の自乗に反比例するようにした例である．$w_{G0}(t)$ までは，最初の TEMPO の資料と特許に明示的に記述されている．これらから $w_{G2}(t)$ までの一般化は，自然であろう．図 2.8 にこれらの関数の振幅特性と時間包絡特性を示す．ここで，最適周波数は 100 Hz に設定されている．

『基本波らしさ』

これらの関数を用いてフィルタされた信号から瞬時周波数と瞬時振幅を求め，それらを評価する尺度を具体的に構成することが必要である．

まず，あるチャンネル τ_c についてのフィルタ出力 $D(t, \tau_c)$ を求める．

$$
\begin{aligned}
D(t, \tau_c) &= |\tau_c|^{-\frac{1}{2}} \int_{-\infty}^{\infty} s(t) \overline{w_G \left(\frac{t-u}{\tau_c} \right)} du && (2.22)\\
w_G(t) &= g_W(t - 1/4) - g_W(t + 1/4) && (2.23)\\
g_W(t) &= e^{-\pi \left(\frac{t}{\eta} \right)^2} e^{-j2\pi t} && (2.24)
\end{aligned}
$$

この特定のフィルタ出力の瞬時周波数 $\omega(t)$ は，次の式で求められる．

$$\begin{aligned}\omega(t) &= 2f_s \arcsin \frac{|y_d(t)|}{2} \\ y_d(t) &= \frac{D(t+\Delta t/2,\tau_0)}{|D(t+\Delta t/2,\tau_0)|} - \frac{D(t-\Delta t/2,\tau_0)}{|D(t-\Delta t/2,\tau_0)|}\end{aligned} \tag{2.25}$$

実際には，逆三角関数を必要とするこの式よりも，次に示すような，Flanagan による計算法 [15] が適切である．

$$\omega(t,\lambda) = \lambda + \frac{\Re[S(t,\lambda)]\Im\left[\frac{dS(t,\lambda)}{dt}\right] - \Im[S(t,\lambda)]\Re\left[\frac{dS(t,\lambda)}{dt}\right]}{|S(t,\lambda)|^2} \tag{2.26}$$

ここで $\Re[\ ]$ と $\Im[\ ]$ は，それぞれ実部と虚部を表す．

こうして求められた瞬時周波数と瞬時振幅を用いて『基本波らしさ』という尺度を設計することとする．これから設計する尺度は，次のような性質を持つことが望ましい．(1) 絶対的な振幅や周波数に依存しないこと．(2) フィルタ内に基本波成分のみが存在するときに最大値を示すこと．(3) 瞬時周波数のみが変動しても瞬時振幅のみが変動しても『基本波らしさ』の低下として現れること．(4) 一定の率の周波数の変化や振幅の変化で『基本波らしさ』が大きく低下しないこと．

ここでは，まず，(1) から (3) の要請を満たすような尺度として次のように定義される指標 $M(t,\tau_c)$ を構成する．

$$\begin{aligned}M(t,\tau_c) &= -\log\left[\int_\Omega w_e(u)\left(\frac{d|D(t-u,\tau_c)|}{dt}\right)^2 du\right] + \log\left[\int_\Omega w_e(u)|D(t-u,\tau_c)|^2 du\right] \\ &\quad - \log\left[\int_\Omega w_e(u)\left(\frac{d^2\arg(D(t-u,\tau_c))}{dt^2}\right)^2 du\right] \\ &\quad + \log\int_\Omega w_e(u)du + 2\log\tau_c \qquad (2.27) \\ w_e(t) &= e^{-\pi\left(\frac{t}{\alpha\tau_c}\right)^2} \qquad (2.28)\end{aligned}$$

ここで積分区間 $\Omega(\tau_c)$ の長さはフィルタの応答の時定数 τ_c に比例するように設定される．最初の項は，瞬時振幅の時間変化の自乗として AM（Amplitude Modulation）による変動分を捉えており，この成分の強さは次の項のエネルギーで正規化されている．第三の項目は，周波数の時間変化の自乗として FM（Frequency Modulation）による変動部分を捉えている．最後の 2 つの項は，この指標 M を標本化周波数やフィルタの中心周波数ひいては基本周波数に依存しない無次元の量とするための正規化係数である．

ここで，先ほどは除外していた (4) の要請を取り入れることとする．これは，現実の音声では，基本周波数や振幅が一定方向に変化することが普通である状況に対応するためである．ここでは，そのような一次式で表現できるような成分を指標の計算から除去し新たな指標 $M_c(t,\tau_c)$ とする..

$$\begin{aligned}M_c(t,\tau_c) &= -\log\left[\int_\Omega w_e(u)\left(\frac{d|D(t-u,\tau_c)|}{dt} - \mu_{AM}(t,\tau_c)\right)^2 du\right] \\ &\quad - \log\left[\int_\Omega w_e(u)\left(\frac{d^2\arg(D(t-u,\tau_c))}{dt^2} - \mu_{FM}(t,\tau_c)\right)^2 du\right] \\ &\quad + \log\left[\int_\Omega w_e(u)|D(t-u,\tau_c)|^2 du\right] + \log\int_{\Omega(\tau_c)} w_e(u)du \\ &\quad + 2\log\tau_c \qquad (2.29) \\ \mu_{AM}(t,\tau_c) &= \frac{1}{c_f(t,\tau_c)}\int_\Omega w_e(u)\left(\frac{d|D(t-u,\tau_c)|}{dt}\right)du \qquad (2.30) \\ \mu_{FM}(t,\tau_c) &= \frac{1}{c_f(t,\tau_c)}\int_\Omega w_e(u)\left(\frac{d^2\arg(D(t-u,\tau_c))}{dt^2}\right)du \\ c_f(t,\tau_c) &= \int_\Omega w_e(u)du \qquad (2.31)\end{aligned}$$

この指標を用いて基本周波数を抽出するには，まず，各時刻において最大の『基本波らしさ』を示すチャンネル τ_c を選択し，そのフィルタと周辺のフィルタの『基本波らしさ』から『基本波らしさ』が最大値を示すフィルタ位置を放物線補間等の方法で求める．次に，そうして求めたフィルタ位置における瞬時周波数を，フィルタ位置の周辺の瞬時周波数から補間により求める．こうして求めた瞬時周波数を基本周波数の推定値とする．

2.3.2 周波数領域の不動点に基づく方法

『基本波らしさ』に基づく方法は，基本周波数成分のみを利用している．これは，空調雑音や商用電源からの誘導雑音が低い周波数成分を主体としており，基本波成分がそれらの影響を最も受け易いという状況を考えると，一つの弱点である．この問題を回避する方法の一つに，複数の調波成分の瞬時周波数の情報を統合する方法がある．ここでは，統合の合理的な基準を与えるため，それぞれのフィルタ出力に含まれる雑音成分の相対的比率を求める方法を与える．

正弦波成分と不動点

中心周波数が λ である任意の帯域通過フィルタを考える．もしフィルタの通過帯域内に周波数が λ_o であるような顕著な正弦波成分が存在すれば，フィルタ出力の瞬時周波数 ω は，その正弦波成分に支配されて，ほぼその正弦波成分の周波数 ω_c となる．したがって，フィルタの中心周波数が ω_c よりも低い場合には，フィルタ出力の瞬時周波数の方が高くなり，フィルタの中心周波数が ω_c よりも高い場合には，フィルタ出力の瞬時周波数の方が低くなる．すなわち，フィルタの中心周波数 λ からフィルタ出力の瞬時周波数への写像 $\omega(t,\lambda)$ を考えると，この写像は，正弦波の瞬時周波数 ω_c において，不動点を持つことが分かる．このようなフィルタの集まりに複数の正弦波成分からなる複合音が入力されたとすると，それらの成分である正弦波の瞬時周波数の集合 $\Lambda(t)$ は，この写像の次のような不動点の集合として求められる．

$$\Lambda(t) \quad = \quad \{\lambda \mid \omega(t,\lambda)=\lambda,\ \frac{\partial\omega(t,\lambda)}{\partial\lambda}<1\}\ , \tag{2.32}$$

以上の議論は，一般的なものであり，どの調波成分についても成り立つ議論である．しかし，これは，帯域通過フィルタの中に単一の調波成分だけが含まれていることを前提とした議論でもある．この議論が成り立つためには，予め基本周波数が分かっている必要がある．これは,『基本波らしさ』の節で遭遇したものと同じ事態である．この問題を解決するため，キャリア周波数 λ_c に関して等方的（あるいはやや時間軸方向に η 倍だけ伸長した）Gabor 関数 $w(t,\lambda_c)$ と，その周波数の逆数の 2 倍の寸法の 2 次のカーディナルスプライン関数 $h(t,\lambda_c)$ を畳込んで作成した次のような関数 $w_s(t,\lambda_c)$ を用いて wavelet 分析を行うものとする[8].

$$w_s(t,\lambda_c) \quad = \quad (w(t,\lambda_c)\odot h(t,\lambda_c))\,e^{j\lambda_c t}\ , \tag{2.33}$$

$$w(t,\lambda_c) \quad = \quad e^{-\frac{\lambda_c^2 t^2}{4\pi\eta^2}}\ ,$$

$$h(t,\lambda_c) \quad = \quad \max\left\{0,1-\left|\frac{\lambda_c t}{2\pi\eta}\right|\right\}\ , \tag{2.34}$$

『基本波らしさ』の節で議論したのと同じ論理により，このような関数の構成の下では，基本周波数がキャリア周波数と一致する場合に，主要な正弦波成分とそれ以外の成分の比（C/N 比：Carrier to Noise ratio）が最大となる．従って，基本周波数の決定は，不動点の中から C/N 比が最大となるものを選択する問題となる．TEMPO との違いは，まず，この選択論理にある．放物線補間で求められる最大の『基本波らしさ』を与えるフィルタ位置での瞬時周波数としての基本周波数と，正弦波成分の存在位置を示す不動点の中から最大の C/N 比を示すものとして選択された成分の瞬時周波数としての基本周波数のいずれが頑健であるかがまず問われることとなる．

C/N 比と不動点周辺の写像の形状

付録に説明するように，主要な正弦波成分以外による干渉をモデル化すると，不動点における写像の偏導関数を組み合わせて近似的に C/N を求めることができる．ここでは，結果のみを記す．ところで，この C/N 比は，不動

[8] ここで用いる関数は，TEMPO の最初の版では提案されていない．しかし，それを一般化した関数のグループには含まれている．

点の周波数を λ_o としたとき，不動点近傍での写像の幾何学的性質に基づいて以下のようにして求められる $\bar{\sigma}$ を用いて $C/N = 1/\bar{\sigma}$ として近似的に推定することができる．

$$\bar{\sigma}^2(t,\lambda) = \int_{-T_w}^{T_w} |w(\tau,\lambda)|\tilde{\sigma}^2(t-\tau,\lambda)d\tau \tag{2.35}$$

$$\tilde{\sigma}^2(t) = c_a\left(\frac{\partial\omega(t,\lambda)}{\partial\lambda}\right)^2 + c_b\left(\frac{\partial^2\omega(t,\lambda)}{\partial t\partial\lambda}\right)^2 . \tag{2.36}$$

$$c_a = \frac{1}{\int_{-\infty}^{\infty}\left(\lambda_o \left.\frac{dg(\lambda)}{d\lambda}\right|_{\lambda=\lambda_o}\right)^2 d\lambda_o} .$$

$$c_b = \frac{1}{\int_{-\infty}^{\infty}\left(\lambda_o^2 \left.\frac{dg(\lambda)}{d\lambda}\right|_{\lambda=\lambda_o}\right)^2 d\lambda_o} .$$

ここで T_w は，関数 $|w(\tau,\lambda)|$ が実質的に 0 でない範囲を覆うことができるように設定する．

C/N 比に基づく複数調波成分の統合

こうして得られた C/N 比は，誤差の相対的な大きさの推定値を与える．ここでは，それらの各成分の誤差が相互に独立であると仮定して統合し，基本周波数を推定することにする．議論を簡単にするために，それぞれの調波に基づく基本周波数の推定値 $\tilde{\omega}_k$ の誤差は，平均 0，標準偏差 σ_k の正規分布に従うものと仮定する．ここで k は，調波の番号を表す．このように仮定すると，N 個の調波成分の情報を次式に従って統合することで，基本周波数の推定値 $\tilde{\omega}_0$ に含まれる誤差の分散を最も少なくすることができる．

$$\tilde{\omega}_0 = \frac{1}{c_n}\sum_{k=1}^{N}\frac{\tilde{\omega}_k}{\sigma_k} \tag{2.37}$$

$$c_n = \sum_{k=1}^{N}\frac{1}{\sigma_k}$$

2.4 駆動音源情報の抽出

STRAIGHT の品質の問題点 [45, 42] を改善するための試みの一環として，周波数帯域毎に駆動成分の周期性を評価して合成に反映させる仕組みを作ることが必要であった．確定的成分と確率的成分を含む音源の制御の問題は，符号化の分野で既に様々な検討が行われている．しかし，STRAIGHT では，音声のモーフィング等の柔軟な加工への応用を目標の一つとしているため，帯域を分割してそれぞれの帯域毎に二値的な判定を導入するといった既存の方法 [16] をそのまま用いることはできない．また，繰返し演算に基礎を置く，より精密な方法 [39] を用いることは，実時間システムとしての実現を狙うという目標と整合しないので，採用することはできない．さらに，これらの方法では，音声のように基本周波数が急速に変化する場合の高い周波数領域における『もあれ』の問題 [59, 52, 21] への対応策を有していない．そのため，ここでは後者の発想を参考にしながらも，非定常な基本周波数に対応でき，しかも繰返し演算を含まない新しい方法を発明して利用する．ここで実装されている方法そのものについては，アイデアの断片的な記述は幾つかの資料 [52, 21] に載っているものの，具体的な手続きとしてまとめられた資料は，これまでに発表されていない．

2.4.1 スペクトルの上側包絡と下側包絡に基づく周期／非周期比の抽出

厳密に周期的な信号があるとする．簡単のために，この信号を周期的なパルス列であると仮定する．この信号に，非周期的な信号として白色雑音が加わったとする．加えられた白色雑音のレベルが周期的な信号と比較して十分に小さい場合，周期的な信号のレベルと非周期的な信号のレベルは次のようにして求められる．

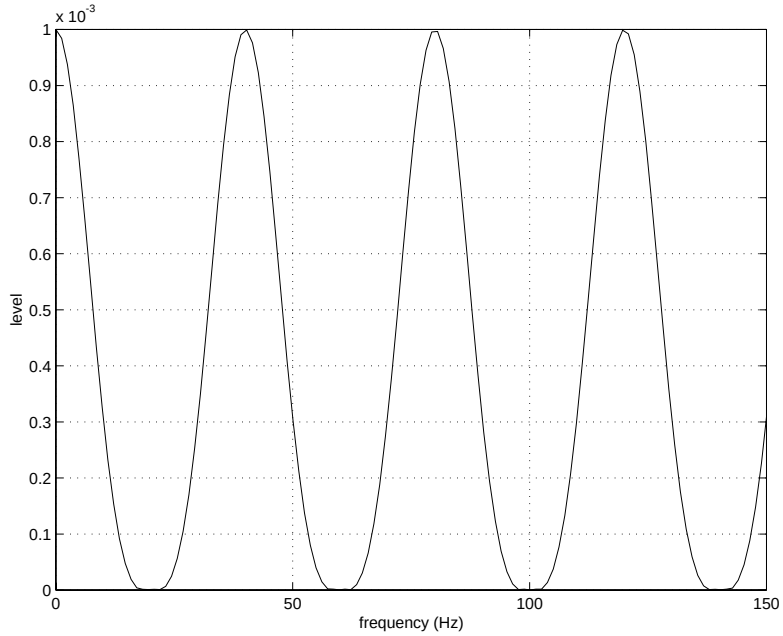

図 2.9: $w_{p2}(t)$ による 40 Hz の周期信号のスペクトル分析の例.

通過帯域が周期信号の基本周波数よりも狭い帯域フィルタを用意する．この，フィルタは，最大の利得が 1 となるように設計されているものとする．そのようなフィルタを調波間の雑音のみが存在する部分に当たるように中心周波数を調整し，出力のパワーを求める．このパワーを帯域フィルタの ERB(等価矩形帯域幅：Effective Rectangular Bandwidth) で正規化すれば，帯域当たりの雑音のパワーが求められる．正弦波成分のパワーは，フィルタの中心周波数を調波成分に一致させた場合の出力のパワーとして求められる．

時間窓の設計

以上のアウトラインは，信号が厳密に周期的な場合のものである．実際には，信号は変化し続けるものであるため，分析に支障のない限りできるだけ局所的にある部分だけを取り出さなければならない．そのための窓関数は，関心の有る時刻を中心として，前後に単調に減少するような形であることが望ましい．一方，周波数方向に関しても満たすべき条件がある．まず，調波成分の間にある雑音成分を抽出することができるためには，帯域フィルタを調波の中央に置いた時，調波位置での利得が零にならなければならない．また，周波数方向でもできるだけ局所化できた方が良い．

このような時間／周波数双方に関わる要請を厳密に満たすには，偏長楕円関数が必要となる．しかし，時間方向の局在性を少し緩めることで，次のような見通しの良い関数 $w_{p1}(t), w_{p2}(t), \ldots$ を利用することができる．信号の周期を t_0 とする．

$$w_{p1}(t) = e^{-\pi\left(\frac{t}{t_0}\right)^2} \odot h_{p1}(t) \tag{2.38}$$

$$w_{p2}(t) = e^{-\pi\left(\frac{t}{t_0}\right)^2} \odot h_{p2}(t) \tag{2.39}$$

$$h_{p1}(t) = \begin{cases} 1 & |t| < t_0 \\ 0 & \text{otherwise} \end{cases}$$

$$h_{p2}(t) = \begin{cases} 1 - \left|\frac{t}{2t_0}\right| & |t| < 2t_0 \\ 0 & \text{otherwise} \end{cases}$$

図 2.9 に $w_{p2}(t)$ を用いて分析した 40 Hz の周期信号のスペクトルの絶対値を示す．狙い通り調波間がほとんど零となっていることが分かる．

上側包絡，下側包絡の推定

こうして求められたスペクトルから周期成分と非周期成分を推定するには，厳密に周期的な信号に雑音が付加された場合には，その周期によって定まるピークの周波数の位置における値を周期成分とし，谷の周波数の位置における値を非周期成分とすれば良い．更に，そうして取り出した値に基づいて繰返し演算によって推定値を改良する方法も提案されている[9]．

しかし，実際の音声では，高い周波数領域においては調波性は破壊されており，ピークと谷が上記の周波数に生じる保証は無い．そのような場合，ピークを結んで作成される上側包絡と谷を結んで作成される下側包絡との大小関係が逆転する場合も生じ得る．この大小関係の逆転の無いことを保証するため，以下の一連の手続きを導入する．

時間軸の非線形伸縮　基本周波数が急速な時間変化を示すと，高い周波数領域では複数の調波間の干渉によって調波構造が破壊される．この問題は，基本周波数が一定の値に見えるような時間軸の上に波形を置き直すことにより，解決することができる．具体的には，新しい時間軸 u を元の時間軸 t の関数 $u(t)$ として表した時，以下の関係を設定すれば良い．

$$u(t) = c_0 \phi^{-1}(t) \tag{2.40}$$

このように設定すると，見かけの瞬時周波数は一定となる[10]．

ケプストラム領域でのリフタ処理　時間軸の非線形伸縮によって見かけの基本周波数が一定となっても，幾つかの問題は残っている．それらは，非周期成分が主要な信号を分析した場合のピークや谷の位置が周期信号のピークや谷の位置と異なることと，周期信号に非周期成分が加わった場合，干渉によって，一つの谷の中に複数の極小値が生ずること等である．

これらの問題を回避して周期成分と非周期成分を簡単な演算で推定するためには，以下の要因についての配慮が必要である．周波数方向では分析に用いた時間窓で定まる分解能以上の細かさで分析することは，無駄である．たとえ，谷の中に複数の極小が生じて見えたとしても，各々が意味があるのではなく，全体で代表させれば良い．実時間性の要請により繰返し計算を用いないため，成分の推定にはかなりの誤りが必然的に存在することとなる．この誤りの影響は，レベルに相対的であり，かつ，非周期成分を過大推定するよりも過小推定した方が品質への悪影響は少ない．

ケプストラムとリフタを用いたケフレンシ上での帯域制限は，これらへの現実的な一つの解を与える．リフタは，想定している基本周期に対応するケフレンシの成分については抑圧せず，基本周期の 2 倍に対応するケフレンシの成分については完全に抑圧し，その間を滑らかに移行する形状とする．

包絡の抽出　時間軸の非線形伸縮とケプストラム領域でのリフタ処理によりスペクトル構造が単純化されたため，ピークと谷を抽出し，それぞれを結ぶことにより，周期成分に対応する上側包絡と非周期成分に対応する下側包絡とを抽出することができる．なお，原点とナイキスト周波数においては，ピークあるいは谷のいずれかのみとなるため，他方については，最近傍の値を利用することとするのが妥当であろう．

数値例

ここでは，以上で説明して来た方法に基づいた具体的な分析の途中経過と結果を例示する．図 2.10 の左側の図に，時間軸の非線形伸縮によって基本周波数を 40 Hz に正規化した母音「ア」のスペクトルを実線により示す．同図中の破線は，ケプストラム領域でのリフタ処理によって平滑化したスペクトルを示す．図 2.10 の右側の図は，同じ母音のケプストラムを示す．正規化した 40 Hz に対応して 25 ms に顕著なピークがある．また，25 ms の整数倍の部分にもピークが見える．

図 2.11 に，こうして求めたケプストラムのピークと谷を単峰化するためのリフタの特性を示す．なお，ここで用いたリフタは以下の式による．

$$\frac{1}{1 + e^{400(t-0.035)}} \tag{2.41}$$

[9]繰返し演算による方法は，前にも指摘したように，実時間性を狙う本方法では採用しない．
[10]現在の版では，局所的な演算だけからこの演算を行うため，基本周波数の変化率から決定される放物線近似を用いている．

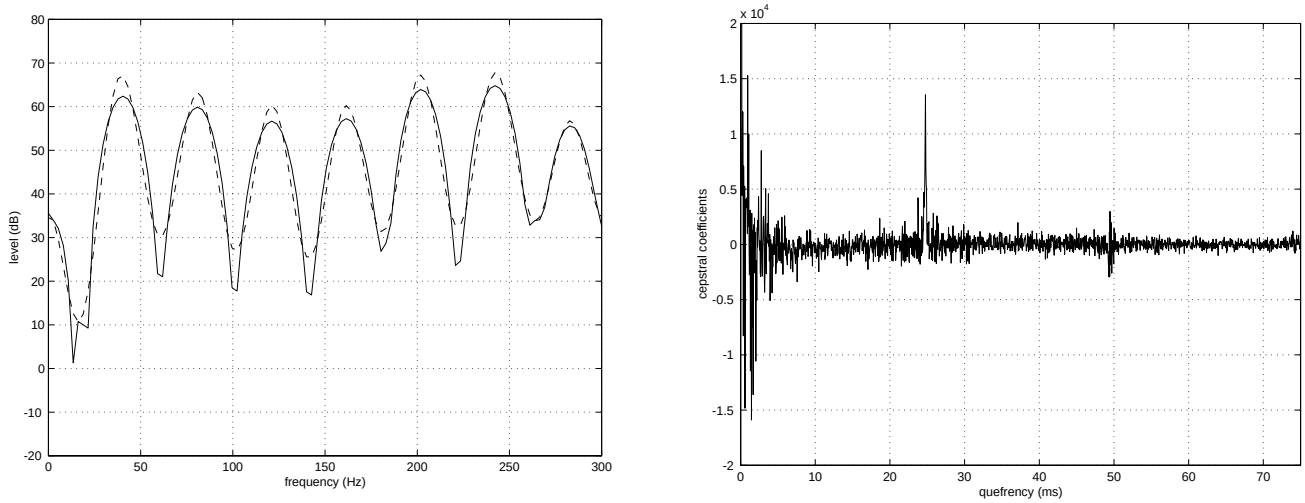

図 2.10: 40 Hz に正規化した母音のスペクトル（左）と，ケプストラム（右）．スペクトルの図中の破線は，平滑化したスペクトル．

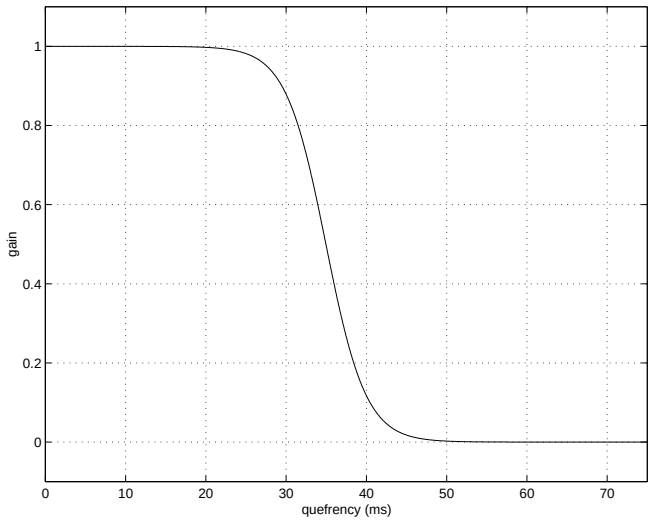

図 2.11: 平滑化のためのリフタ．40 Hz に対応する 25 ms の成分はそのまま通し，その整数倍の成分は完全に抑圧するように設定してある．

図 2.12 の左側に，平滑化されたスペクトルのピークを結んで求められた上側包絡と谷を結んで求められた下側包絡を示す．図 2.12 の右側には，両者の差として求められる非周期成分の割合を示す．これらの表示の縦軸は，尺度に依存しないように dB となっている．

2.5　合成部

これらの分析結果（あるいは，それらを変形したもの）として得られる音声パラメタを用いて，合成部分では，音声を再合成する．各時刻の振幅スペクトル情報は，複素ケプストラムを経て，最小位相のインパルス応答 [27] に変換される．このインパルス応答と音源信号が畳込まれて，再合成音声となる．畳込みには実際は FFT が用いられるため，音源に対する様々な加工もこの段階で加えられる．

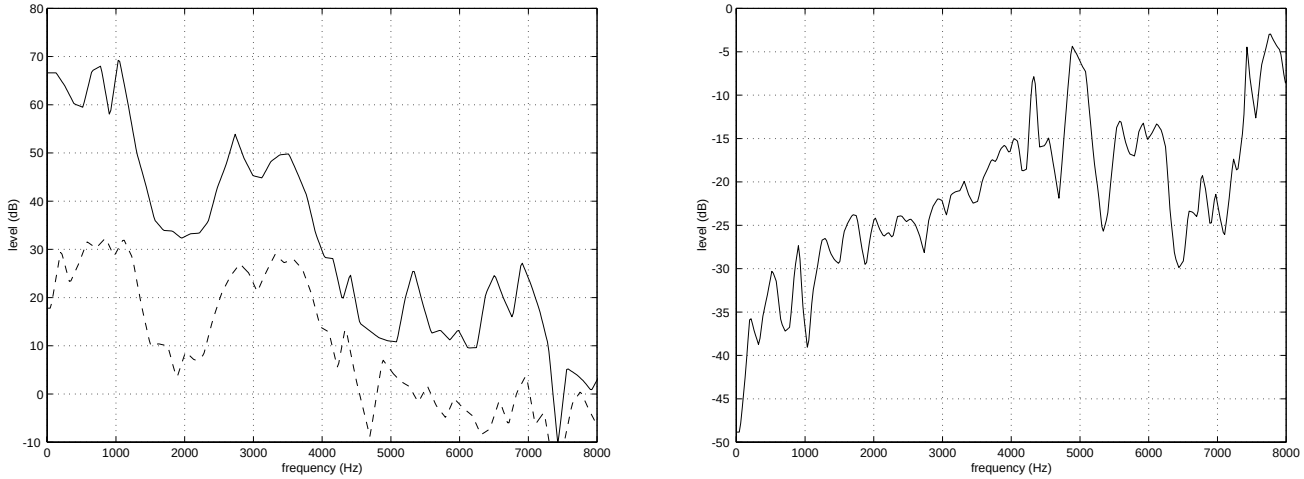

図 2.12: 抽出された上側包絡と下側包絡（左）非周期成分の割合（右）

2.5.1 群遅延操作による駆動音源の作成

群遅延 τ_g は，まず，精密な基本周波数の設定に用いられる．有声音の駆動用のパルスの位置を，標本化周波数から決まるものよりも τ_d だけ後の位置に置くことが必要になったとする．ここで τ_d は，標本化周期よりも短い時間長である．すると，パルスのフーリエ変換の位相 $\Phi_1(\omega)$ を以下のようにすることで，目的は実現できる[11]．

$$\Phi_1(\omega) \quad = \quad -\int_0^{\omega} \tau_d d\lambda \tag{2.42}$$

ここで，基本周波数の精密な制御のための位相制御に加え，パルスのエネルギーの時間的拡散のための制御を加える．ここで位相制御による波形への影響を組織的に調べる．まず，位相を正弦波状に変化する位相の合成として表現する．

$$\Phi_2(\omega) = \exp\left(-j\rho(\omega)\sum_{k\in P}\alpha_k \sin(k\omega)\right) \tag{2.43}$$

ここで，P は，添字の集合を表す整数の集合であり，また．$\rho(\omega)$ は，位相が変動する周波数範囲を指定する関数である．$\rho(\omega)$ には，例えば，次のような sigmoid 関数を用いることができる．

$$\rho(\omega) = \frac{1}{1+\exp(-\frac{\omega-\omega_0}{b_w})} \tag{2.44}$$

このような位相特性の操作を行う目的は，パルスのエネルギーの時間的な広がりの制御である．位相の操作とエネルギーの広がりの関係を調べるため，まず，最初に，この $\rho(\omega)$ が一定値 1 である場合を考える．更に，一つの k に対応する成分のみが存在するものとする．すると，成分の強度がある一定の値以下であるような範囲は，は以下のように Γ 関数を用いて求めることができる．

$$\begin{aligned} |\alpha| \quad &\leq \quad (\varepsilon\Gamma(\beta+1))^{\frac{1}{\beta}}\, 2^{\frac{\beta-1}{\beta}} \\ \beta \quad &= \quad \frac{\Delta t}{k} \end{aligned} \tag{2.45}$$

図 2.13 は，空間周波数成分が単一 $k = 11$ の場合でかつ周波数に依存しない加重 $\rho(\omega) = 1$ の場合の波形，波形の振幅の対数軸表示，群遅延特性を示す．この広がりは，式 2.45 により求められる範囲と良く対応している．

図 2.14 は，同じ群遅延特性に sigmoid 型の周波数加重を掛けた場合を示す．周波数加重の導入は，それぞれのパルス位置を変えないが，各々のパルスは，高域通過フィルタのインパルス応答の形状に変わっている．この場合も，エネルギーの時間的な広がりは，パルス位置が変わらないため，図 2.13 と同様になる．

しかし，このように単一の空間周波数成分を用いた場合には，パルスが特有の音色を持つという問題がある．この問題を避けるため，現在の版の STRAIGHT では，群遅延特性は，帯域制限された白色雑音を用いて設計されて

[11]離散時間システムでの実装に関連する問題については，次の節で議論する．

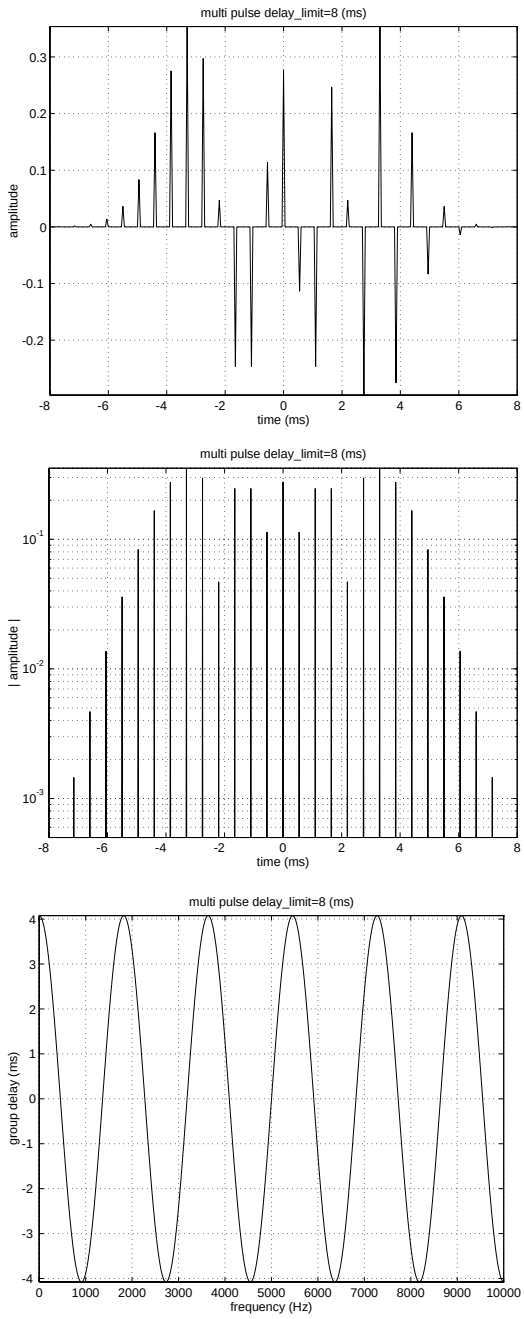

図 2.13: 単一空間周波数成分のみの群遅延によるパルスの時間表現．波形（上），波形振幅の対数表示（中），群遅延（下）．($k = 11, \varepsilon = 10^{-3}, \rho(\omega) \equiv 1$)

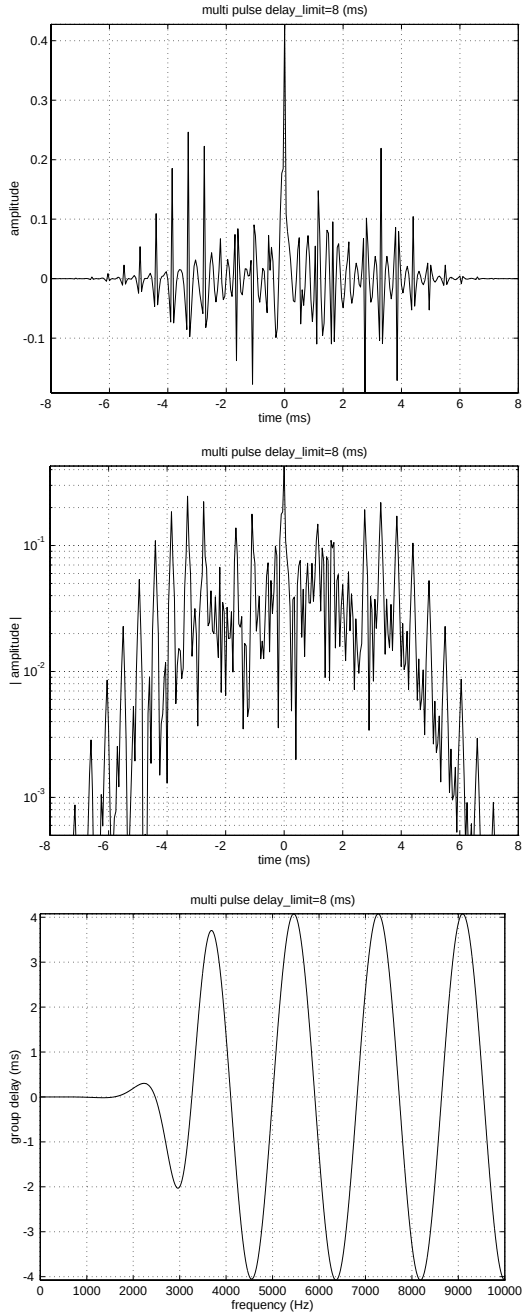

図 2.14: 単一空間周波数成分に周波数加重を加えた群遅延によるパルスの時間表現．波形（上），波形振幅の対数表示（中），群遅延（下）．($k = 11, \varepsilon = 10^{-3}, \rho(\omega)$ は，3kHz を境界とする sigmoid 型．)

いる．ここでは，位相を設計のためのパラメタとして用いていない．目的とする制御量は，時間領域での広がりであり，時間軸上で定義される群遅延で指定する方がより直接的なためである．

まず，周波数軸上の白色ガウス雑音 $n(\omega)$ を作成する．次に，この雑音の時間領域表現 $N(\tau)$ を，空間周波数領域（時間）での加重 $W_s(\tau)$ を用いて，帯域制限し，周波数領域での帯域制限された乱数 $x(\omega)$ を作成する．目的とする群遅延特性は，この乱数を正規化して指定した群遅延の標準偏差 d_g を対象とする周波数領域で持つように $\rho(\omega)$ を用いて周波数加重を行って作成する．これら，一連の段階は，次式にまとめられる．

$$\tau_{g3}(\omega) \quad = \quad \rho(\omega) \frac{d_g x(\omega)}{\sqrt{\frac{1}{2\pi} \int_{-\pi}^{\pi} |x(\omega)|^2 \ d\omega}} \tag{2.46}$$

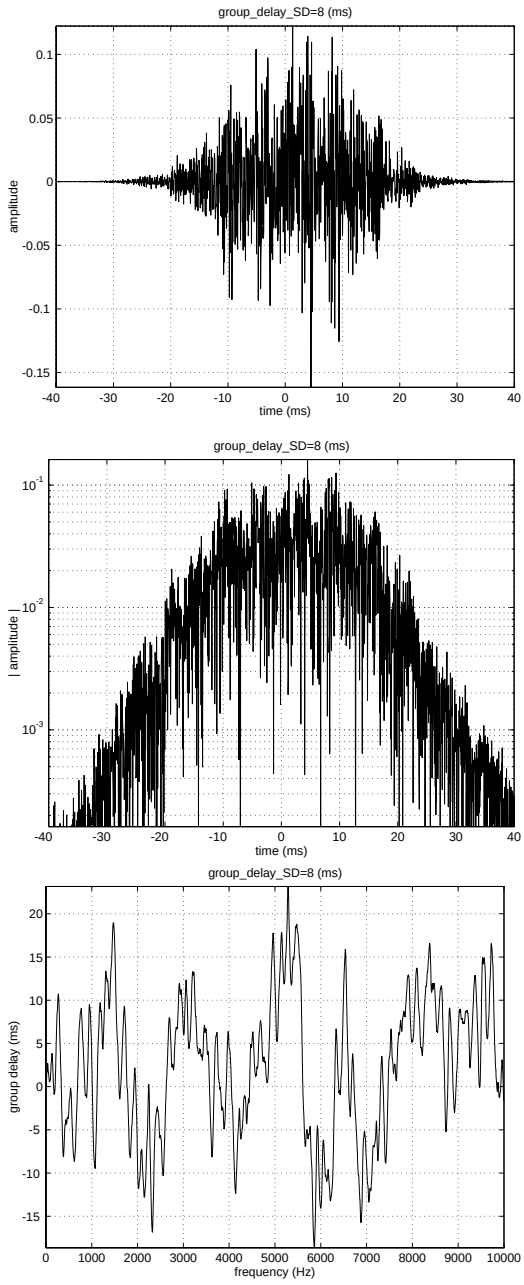

図 2.15: 帯域制限した乱数を群遅延とするパルスの時間表現．波形（上），波形振幅の対数表示（中），群遅延（下）．($d_g = 8(\mathrm{ms}), \rho(\omega) \equiv 1$)

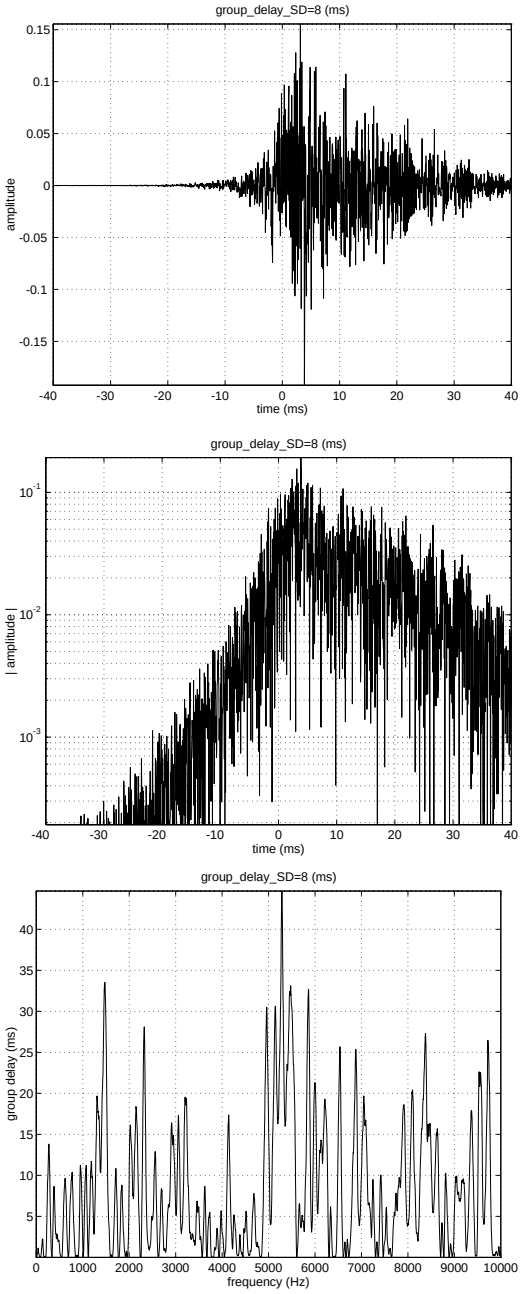

図 2.16: 帯域制限した乱数に非対称性を加えた群遅延を持つパルスの時間表現．波形（上），波形振幅の対数表示（中），群遅延（下）．($d_g = 8(\mathrm{ms}), \rho(\omega) \equiv 1$)

$$
\begin{aligned}
x(\omega) &= F^{-1}\left(W_s(\tau)N(\tau)\right) \\
W_s(\tau) &= |\tau| \exp\left(-\pi(\tau/\tau_{bw})^2\right)
\end{aligned} \tag{2.47}
$$

ここで，$F^{-1}()$ は，逆フーリエ変換を表す．この成分の位相特性 $\Phi_3(\omega)$ は，$\tau_{g3}(\omega)$ を周波数軸上で積分することにより求められる．図 2.15 に，こうして作成されたパルスの例を示す．

通常の基本周波数の範囲では効果が顕著では無いために STRAIGHT には実装されていない操作として，時間的に非対称な群遅延特性がある．位相特性を用いて書けば以下のような操作である．

$$
\Phi_4(\omega) = \exp\left(-j\int_{-\pi}^{\omega} r\left(j\frac{d\log\Phi_3(\lambda)}{d\lambda}\right)d\lambda\right) + c_0 \tag{2.48}
$$

ここで，$r()$ は，適当に滑らかな偶関数である．例を以下に示す．

$$r(x) \quad = \quad x + \beta \log\left(1 + e^{-\frac{2x}{\beta}}\right) \tag{2.49}$$

ここで β は，x のダイナミックレンジに応じて決めるべきパラメタである．図 2.16 に，このようにして非対称性を導入した群遅延特性から作成されたパルスを示す．振幅特性にも同様な非対称性が生じていることが分かる．この非対称性の知覚の詳細に関しては資料 [56, 36, 38] を参照されたい．

2.5.2 駆動音源とスペクトル情報からの音声合成

合成は，駆動音源の更新のタイミング毎に行われる．有声音の場合には，駆動パルスを立てる時刻毎であり，無声音の場合には，一定のフレーム周期毎である．この時刻を t とする．内部に格納されている時間周波数表現は，t に関して連続的であり得るが，合成に際しては，このように離散的な時刻での情報のみが用いられるため，合成に関連するパラメタでは，時間を添字として扱うこととする．

合成音声は，こうして求められた個別の位相特性 $\Phi_1(\omega,t),\ldots,\Phi_4(\omega,t)$ を合成した位相特性 $\Phi_t(\omega)$ と，対応する滑らかなスペクトル情報 $S(t,\omega)$ を用いて，以下のようにして求められる．まず，位相特性 $\Phi_t(\omega)$ を求める．

$$\Phi_t(\omega) = \sum_{k=1}^{4} \Phi_k(\omega,t) \tag{2.50}$$

次に，スペクトル情報 $S(\omega,t)$ から複素ケプストラムを経由して最小位相インパルス応答に対応する周波数表現 $V_t(\omega)$ を求める．

$$\begin{aligned} V_t(\omega) \quad &= \quad \exp\left(\frac{1}{\sqrt{2\pi}}\int_0^\infty h_t(q,t)e^{j\omega q}dq\right) \\ h_t(q) &= \begin{cases} 0 & (q<0) \\ c_t(0) & (q=0) \\ 2c_t(q) & (q>0) \end{cases} \\ \text{and} \qquad c_t(q) &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{-j\omega q}\log S(\omega,t)d\omega \end{aligned} \tag{2.51}$$

$c_t(q)$ はケプストラム，$h_t(q)$ は，それから求められる複素ケプストラムであり，q は，ケフレンシーを表す．

有声音の場合の応答波形 $y_t(t)$ は，この $V_t(\omega)$ と $\Phi_t(\omega)$ の積を逆フーリエ変換して求められる．

$$y_t(t) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} V_t(\omega)e^{j\Phi_t(\omega)}e^{j\omega t}d\omega \tag{2.52}$$

最後に，周期によるエネルギーの変化を補正するための係数 $1/\sqrt{f_0(t)}$ を掛けて，一つの駆動パルスに対応する合成音声波形が求められる．これをそれまでに求められている合成音声波形に加えることで，合成が続けられる．

無声音の場合には，フレームの更新周期に対応する長さだけを切り出した白色ガウス雑音 $n_t(t)$ のフーリエ変換 $N_t(\omega)$ を用いて，以下のように一フレーム分の応答 $y_t(t)$ が計算される．

$$y_t(t) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} N_t(\omega)e^{j\omega t}d\omega \tag{2.53}$$

こうして求められた応答は，それまでの合成音声波形に加えられ，合成が続けられる．

周期／非周期混合音源の制御

現在の STRAIGHT の実装では，周期音源／非周期音源に対応するスペクトルは，それぞれの成分の比率に基づいて補正され，独立に合成された後，時間軸を揃えて混合される．この非周期成分の比率を自動的に群遅延特性の制御パラメタに反映させることにより，更に合成音声の品質を向上させることができるものと考えられる．ただし，そのような改良は，今後の課題である．

第3章　GUI-STRAIGHTのユーザインタフェース

STRAIGHT は，これまでに説明したアルゴリズムの集合である．それぞれの関数は，引数を適切に設定すれば，独立に使用することができ，組み合わせて使用することで音声分析変換合成系を構成することができる．STRAIGHT の開発環境である Matlab[1] 自体，様々なデータ可視化用ツールと対話的な操作を可能とするインタプリタが統合された柔軟なものであり，それだけでも高度な研究環境として利用することが可能である．しかし，必ずしも Matlab についての高度な知識を持たないユーザの研究段階を支援するツールとして有用であるためには，様々なアイデアの試行錯誤をコマンドを介さずに行えるようなインタフェースを用意することが必要であった．また，そのようなユーザインタフェースは，Matlab に関する十分な知識を有するユーザにとっても，億劫さを除くことで試行錯誤を促し開発をスピードアップするという効果がある．

この章では，STRAIGHT の使用を容易なものとするために試験的に実装した GUI をベースとするシステム（GUI-STRAIGHT）について説明する．ここでは，具体的な音声ファイルの分析を例として，具体的なステップを一つ一つ追いながら，インタフェースの内容と操作法を紹介することとする．[2]

3.1　主操作パネル

GUI-STRAIGHT を構成する .m ファイルやサンプル音声は，一つのフォルダーにまとめられている．GUI-STRAIGHT および関連する関数群を使用するためには，path 命令あるいは GUI を用いて，Matlab の実行パスが GUI-STRAIGHT の格納されているパスを含むようにしておく必要がある．

まず，以上の環境を設定しておく．この環境の下で GUI-STRAIGHT を起動するには，Matlab のコマンドウィンドウから

```
straight
```

という命令を入力する．

すると，図 3.1 に示すような主制御インタフェースが表示される．STRAIGHT control panel と書かれた操作パネルの中央上部が主要な操作を行うためのサブパネルである。それぞれの時点で操作可能なボタン以外は操作できないように休止状態:disabled とされている。休止状態になっているボタンは色が淡く表示されている。[3]大まかに言えば，上半分に分析／操作関連の機能，下半分に合成／表示機能が置かれている．以下，順を追いながら説明する．

3.2　音声データの読込み

最初に起動した状態で可能なのは，再初期化と音声の読込みである．ここでは，まず，音声を読込む．

主操作パネルの中央上部のパネルに，音声読み込みのためのボタン (read from file) がある．このボタンを押すことで、図 3.2 に示すような音声ファイルを指定するためのダイアログが現れる。(以下，Mac の例を示す。ここでは，それぞれのプラットフォームにおけるファイル読み込みのためのダイアログが出る．この例では，a125.wav というファイルを指定している．

GUI-STRAIGHT で読込むことができるのは，Windows で良く用いられる WAVE 形式 (.wav)，Macintosh や Amiga の音楽ソフト等で用いられる AIFF 形式 (.aiff, .aif) [4]，16 ビットのリニア PCM の生のバイナリデータ（バ

[1] Mathworks 社（http://www.mathworks.com/）製，国内代理店はサイバネットシステム社（http://www.cybernet.co.jp/）

[2] GUI-STRAIGHT を最初に作成した時には基本周波数の抽出方法に一世代前の TEMPO が用いられていた．この時に用意した英文の説明書 [17] は既に内容の大部分は陳腐化している．

[3] ただし、Matlab 上の実装は試験システムという位置付けであるため，ボタンの休止状態の制御に見落としがある可能性も残っている。

[4] ただし，この実装では単一のセグメントのデータのみがサポートされている．

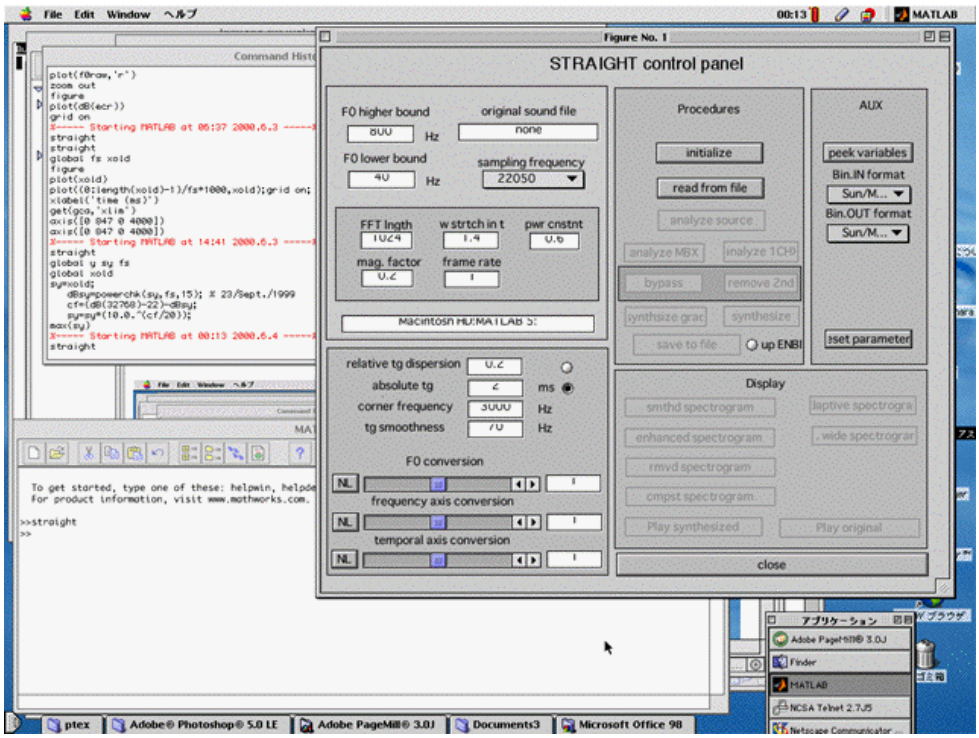

図 3.1: GUI-STRAIGHT の主操作パネル

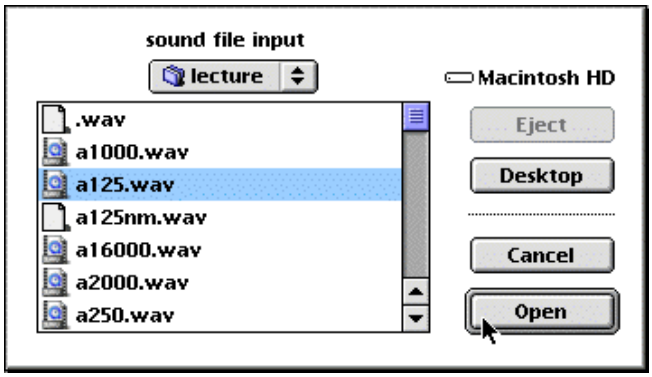

図 3.2: 音声ファイル入力のためのダイアログ（Mac の場合）

イト順は，二通り）の三種類である．ファイルの種類は拡張子で判断しているため，(.wav, .aif, .aiff) 以外は，生のバイナリデータとして解釈される．

3.3　音源情報の分析

音声ファイルを読込むと，GUI-STRAIGHT の主操作パネルは図 3.3 のように変化する．まず，音源情報分析ボタン (analyze source) がアクティブになり，操作可能であることを示す．また，サンプリング周波数の情報を有する音声ファイルの場合には，左上にある分析用のサブパネルの情報が更新される場合がある．

図 3.4 に分析用のサブパネルを示す．音声ファイルのサンプリング周波数情報が大きな場合（例えば 44.1 kHz, 48 kHz 等）には，分析用のパラメタの一つである FFT のサイズが変更される場合がある．なお，生のバイナリファイルの場合には，ファイルにはサンプリング周波数の情報が添付されていないので，ポップアップメニューを用いて設定する必要がある．

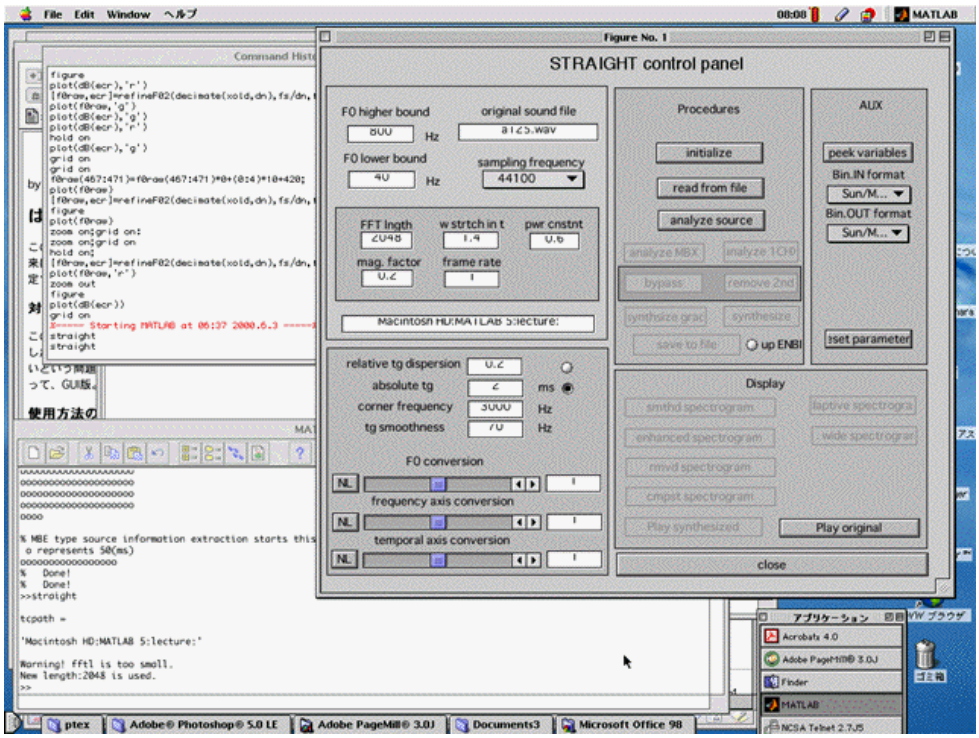

図 3.3: 音声ファイルを読込んだ後の GUI-STRAIGHT の主操作パネル

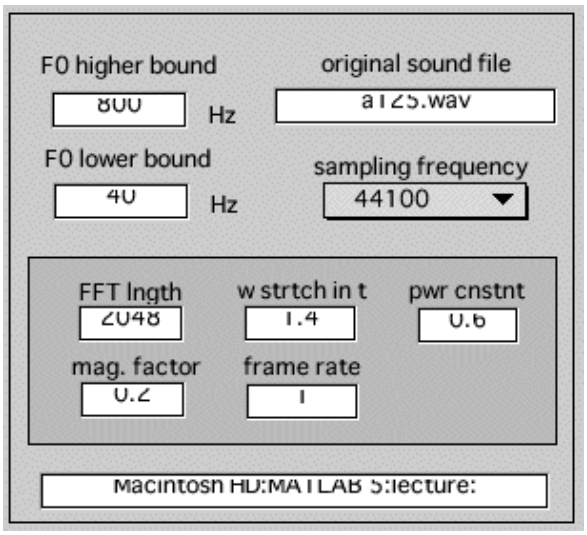

図 3.4: 音声ファイルを読込んだ後の分析サブパネル

分析サブパネルの左上の二つの edit window は，音声の基本周波数の探索範囲を設定するためのものである．ここで既定値として設定されている値は，都木らが放送プログラムの音声を調べた結果 [74] に基づいている．分析対象とする音声の基本周波数に関する事前情報が利用できる場合には，この範囲を狭くすることで分析に要する時間を短縮することができる．サブパネルの中にさらに囲いで区切られている 5 個の edit window は，分析の他のパラメタの操作用である．FFT length は，内部で用いる FFT のサイズを指定する．w stretch in t, power constant, mag. factor は，理論的背景の部分で説明した補正に関連する部分である．様々な聴取試験の結果，既定値が設定されているので，これらを変更することはあまり勧められない．

frame rate は，変更した方が望ましい既定値である．理論的背景で説明したように，現在の STRAIGHT では，相補的時間窓を用いることで，時間方向の平滑化が既に分析の段階で組み込まれている．frame rate を 1 ms とし

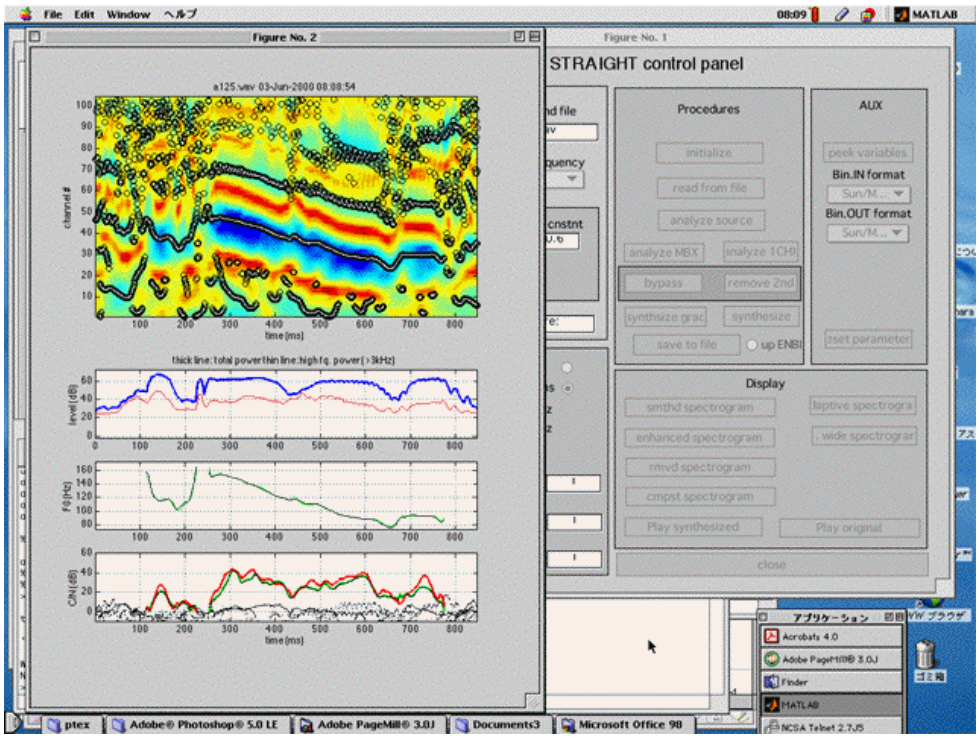

図 3.5: 音源情報抽出処理の修了時の画面

た場合には，時間方向の平滑化も行われるため，過剰平滑化によりやや音質が劣化することが分かっている．この値を 2 ms 以上，5 ms 以下とすることで，品質をそれほど劣化させることなく処理速度，記憶容量を削減することができる．

3.3.1　音源情報の表示

図 3.5 に音源情報抽出が修了した時の画面の様子を示す．画面の左側に，音源情報の表示パネルが見える．ここで用いた音声は、男性が発声した「１２５」という音声で,「ひゃくにじゅうご」と発声している．パネル内のそれぞれのグラフについて，以下，説明する．

図 3.6 に示したマップは、各フィルタの出力を用いて求めたフィルタの中心周波数から出力の瞬時周波数への写像の不動点と、それぞれのフィルタ出力についての主要正弦波成分と背景雑音の比（C/N 比：Carrier to Noise ratio, 通信工学からの借用。）を表わす。不動点は○印で表され，/N 比は疑似カラーで表されている。C/N 比が高い部分を寒色、C/N 比が低い部分を暖色としている。相対的なノイズ分の多さが色で表されていると解釈（イメージを描く）した場合に直感に整合するようにデザインされている。基本周波数は C/N が最も高い不動点として選択される[5]。

マップの縦軸は分析に用いたフィルタの番号 (channel #) を示す。フィルタ番号 n_f とフィルタの中心周波数 f_c の間には、以下の関係がある。

$$f_c = f_l 2^{\frac{n_f - 1}{24}} \tag{3.1}$$

ここで f_l は，基本周波数の探索範囲の下限の周波数を表わす．STRAIGHT では，1 オクターブに 24 のチャンネルを配置しているが，これは音源情報を見易く可視化するための設定である．基本周波数の抽出性能は，1 オクターブあたりのチャンネル数を 6 にしても劣化しないことが確認されている．

図 3.7 は，有声／無声判定のための補助情報として用いられているパワー情報である．画面の青色（図では太線）の線は全帯域でのパワーを示し，赤線（図では細線）の線は 3 kHz 以上の帯域でのパワーを示す．有声／無声の判

[5] この版の STRAIGHT で実装されている選択論理は，不動点に基づく方法の最良の性能を引き出してはいない．最良の性能を実現するための実装については，別の資料で詳細を説明する．

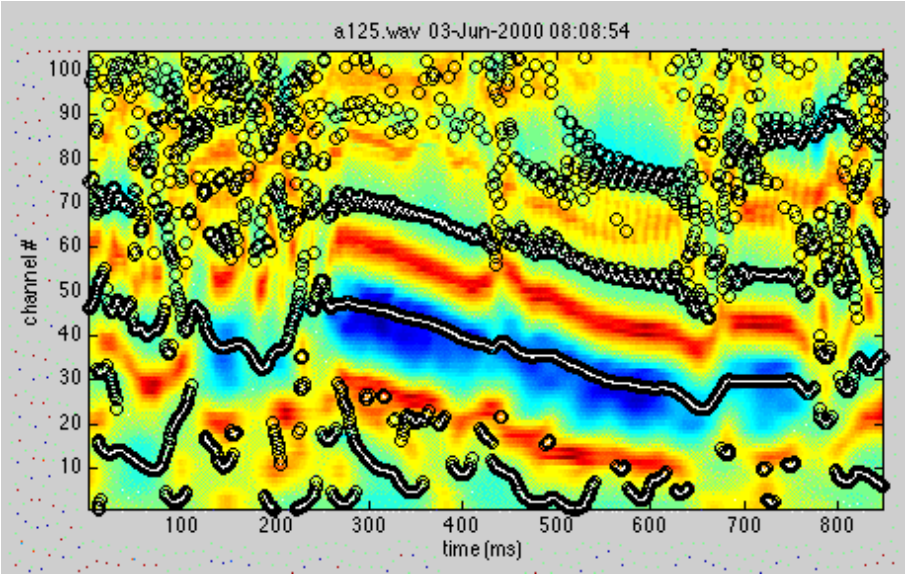

図 3.6: 不動点に基づく基本周波数の抽出と C/N の疑似カラー表示

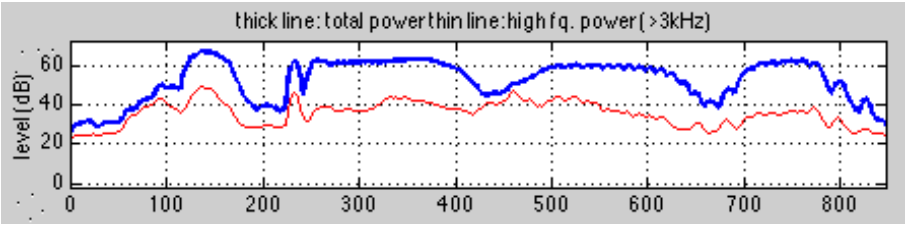

図 3.7: 有声／無声判定のための帯域パワーの表示

定には，これらのパワー情報と不動点の分析で求められる C/N 情報と不動点の連続性等が用いられている[6]．

図 3.8 に，基本周波数の初期推定値と調波性を用いて改良された推定値を示す．緑色の線（資料では淡い線）が最初のマップの不動点から直接求められた基本周波数を表わす。ここでは，各時刻において最大の C/N 比を持つ不動点を選択し、その不動点の属するフィルタの出力の瞬時周波数と隣接するフィルタの瞬時周波数を用いて、実際の不動点の周波数を補間して求めている。黒い線は、その基本周波数の情報を用いて、適応的時間窓を設計し、第三調波成分までに含まれる基本波の情報を用いて求めた基本周波数を表わす。粗い基本周波数の情報に基づいて基本周波数を求めるルーチンは独立の関数として用いることが可能である。詳しくは、次の章で説明する。

図 3.9 は、不動点の場所での C/N 比を表わす。0dB という C/N 比は、抽出された正弦波とそれ以外の背景雑音成分とのレベル差が無いことを示す。緑色の線は、基本周波数として選択された不動点に対応する不動点の C/N 比を表わす。母音の中央部分では、基本波成分は他の不動点よりも遥かに高い C/N 比であることが分かる。赤い線

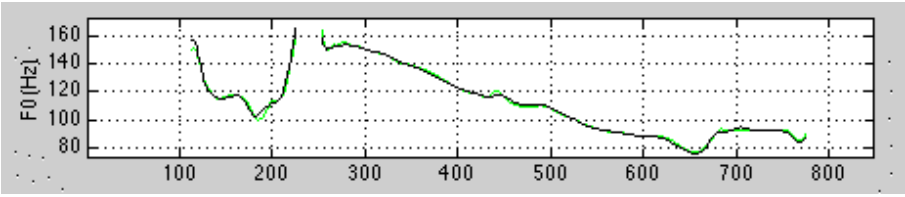

図 3.8: 基本周波数の初期推定値と調波性を用いて改良された推定値

[6] この判定論理も改善の余地の有る実装となっている．

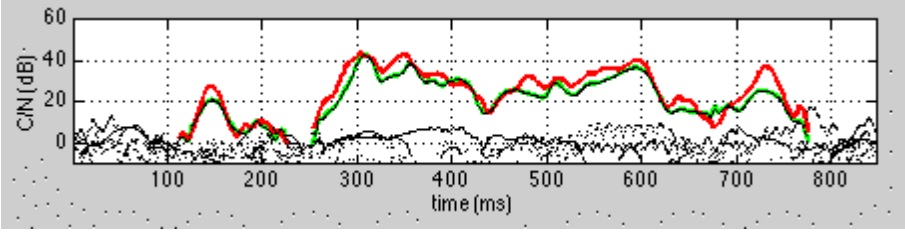

図 3.9: 各不動点の C/N と改良された推定に対応する C/N

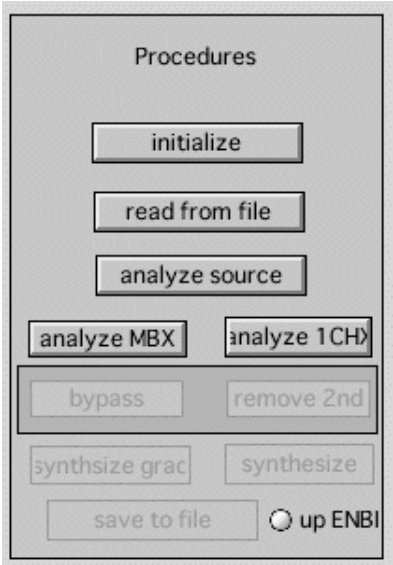

図 3.10: 音源情報抽出後の操作サブパネル

は、複数の調波成分から求められた基本周波数に対応する C/N 比である。緑の線と赤い線は利用している周波数帯域が異なるため、C/N 比の意味は、それぞれで異なっている。

3.4　時間周波数情報の抽出

基本周波数情報が，この段階で求められた。次は、帯域毎の周期性と非周期性を考慮した音源情報の抽出である。この分析は、STRAIGHT 本体のスペクトル分析と併行して行われる。

図 3.10 に音源情報抽出後の操作サブパネルの様子を示す．ここで，新たに二つのボタンが操作可能になっている．ここで用いるのは，左側の'analyze MBX' の方である．右側のボタンは，音源情報を二値的に判定する分析を起動し，既にサポートは行っていない．

'analyze MBX' をクリックすると，平滑化スペクトログラムの計算と周期／非周期情報の分析が開始される．この過程はかなり時間を要するので，進行状況を Matlab の command window に表示するようにしている[7]．

分析が終了すると，周期／非周期に関するデバッグ用の表示を行い，次の図 3.11 に示すような画面となる．図の中央に表示されているグラフはデバッグを目的としたものであり，後の段階で利用される情報を表示している訳ではない．グラフは，各フレームにおける周期性信号の領域と非周期性信号の領域の境界となる周波数を表示している．ここでは，低周波側が周期的，高周波側が非周期的であることが仮定され，sigmoid を当てはめた場合の変曲点を境界と看做している．この段階で得られる周期／非周期情報については，次の章で詳しく説明する．

[7]この段階で，frame rate を 2 ms 以上にしていると，Matlab の command window に『フレーム周期が短過ぎて時間方向の平滑化ができない』と警告が表示される．これは，旧版からの名残りで無視して構わない．むしろこの章の始めに説明したように，時間方向の平滑化を行わない方が音質が良いことが報告されている．

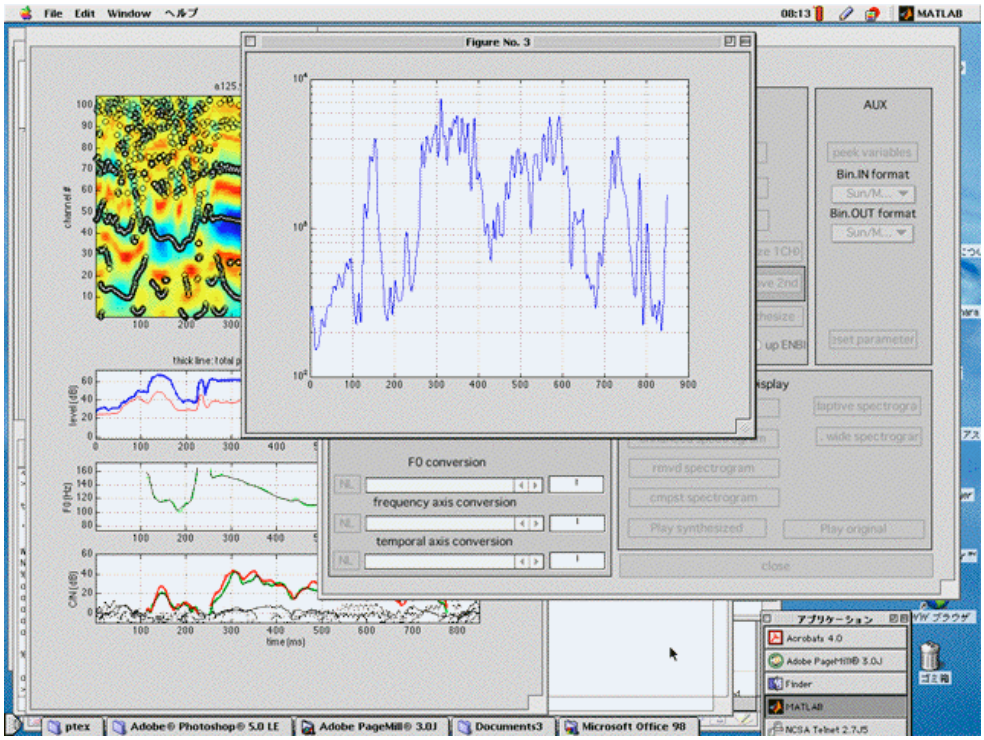

図 3.11: 平滑化スペクトログラム抽出後の全画面

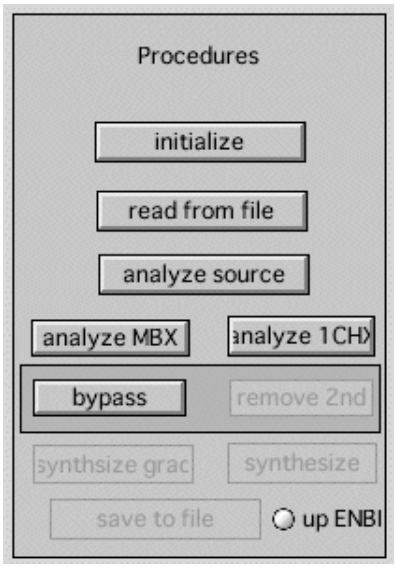

図 3.12: スペクトル情報抽出後の操作サブパネル

この段階で，グラフの下に隠れている操作サブパネルは，以下の図 3.12 のように変化している．ここでは，新しく使用可能になった'bypass' をクリックする[8].

図 3.13 に'bypass' ボタンをクリックした後の操作サブパネルを示す．新たに二つののボタンが使用可能になっている．これでようやく合成が可能な段階となった.

[8]'remove 2nd' はここでは用いない．この操作によって音質が改善される場合は存在するが，どのような機構がこの操作を必要とするようなスペクトル上の二次構造 [58] の原因になっているかが明らかになるまでは，安易に利用しない方が良いと判断した.

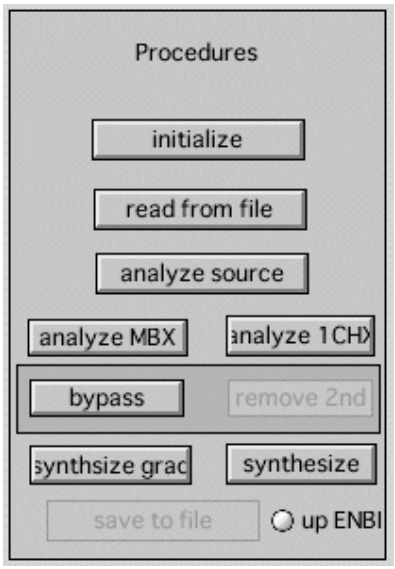

図 3.13: bypass ボタンのクリック後の操作サブパネル

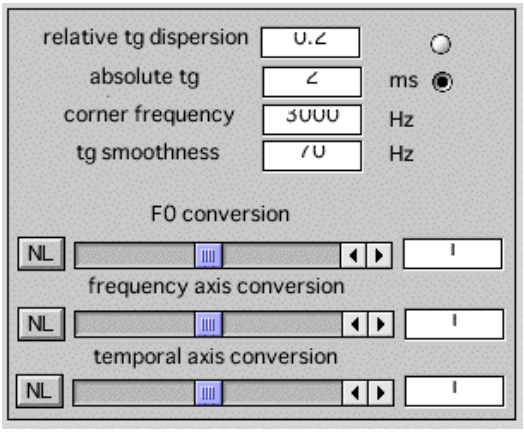

図 3.14: 合成パラメタ操作用のサブパネル

3.5　合成パラメタの操作と再合成

合成には，図 3.13 の’synthesize graded’ のボタンを用いる．他方のボタンは動作するが，既にサポートは打ち切られている．

図 3.14 に合成パラメタの操作用のサブパネルを示す．このパネルで操作できるパラメタとその説明を表 3.1 に示す．最初の４つのパラメタは，合成音源の郡遅延制御に関連するものである．右側のラジオボタンは，郡遅延の標準偏差を基本周期に対する割合で指定するのか，絶対的な標準偏差で指定するのかを選択する．これまでの経験では，絶対的な標準偏差で指定した方が自由に制御でき，自然性も良い傾向がある．パラメタの具体的な制御は，エディット窓の中の数字を書き換えることによって行う．

サブパネル下部には，基本周波数，周波数軸，発話速度の変換用の GUI ツールが配置されている．それぞれは，ボタンとスライダーとエディット window から構成されている．周波数軸用のボタン以外のボタンは有効では無い．また，周波数軸用のボタンも動作するが十分には保守されていないため，ここでは，スライダーとエディット window についてのみ説明する．

スライダーとエディット window で設定されるのは，比例定数である．これらは，どちらか一方を変更すると，もう一方の要素も更新されるように作られている．また，極端な値をエディット　window に入力しても，スライダー可動範囲で示される有効範囲内の数値に制限される．基本周波数の場合は，この GUI ツールで設定される比例定数

表 3.1: 合成パラメタと簡単な説明

パネル表示名	概要
relative tg dispersion	周期に対する郡遅延の標準偏差の割合
absolute tg	群遅延の標準偏差 (ms)
corner frequency	群遅延の境界周波数 (ms)
tg smoothness	郡遅延の空間周波数の上限 (Hz)
F0 conversion	基本周波数の変換率
frequency axis conversion	周波数軸の変換率
tempral axix conversion	発話時間の変換率

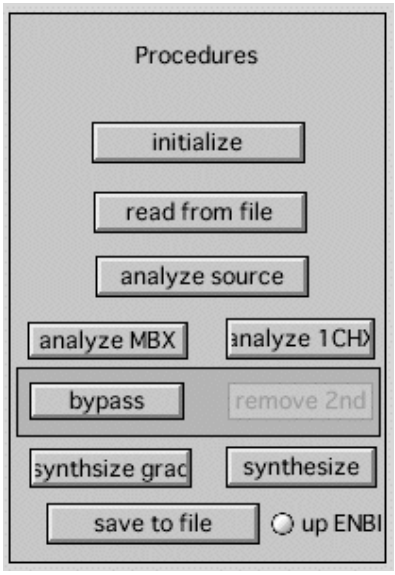

図 3.15: 音声合成が完了した後の主操作サブパネル

は，合成音声の基本周波数と原音声の基本周波数の比である．例えば,「2」という数値を設定すると，合成音声の基本周波数は原音声の基本周波数の二倍となる．

周波数軸の変換率の場合には，設定された比例定数に従って，合成音声の作成に用いられるスペクトルが周波数軸方向に伸長される．具体的には，原音声から求められたスペクトルを例えば $P_O(\omega)$ とし，合成の時に用いられるスペクトルを $P_S(\omega)$ とし，GUI で設定される数値を a とすると，両者の間には次のような関係が設定される．

$$P_S(\omega) = P_O\left(\frac{\omega}{a}\right) \tag{3.2}$$

時間軸の変換では，合成音声の時間長が原音声の時間長に GUI に設定された値を掛けた値となるように合成される．

なお，ここで紹介したように，スライダーに対応づけられているパラメタは，この合成用のサブパネルでは比例的に変化させることができるだけである．しかし，応用に関する部分で説明するように，合成関数を直接呼び出すことによって，非線形の任意の変換が可能である．

3.6 音声の書き出し

音声の合成が完了すると，command window に Done!というメッセージが表示されるとともに，図 3.15 に示すように，主操作サブパネルのファイル書出し用のボタンが使用可能になる．

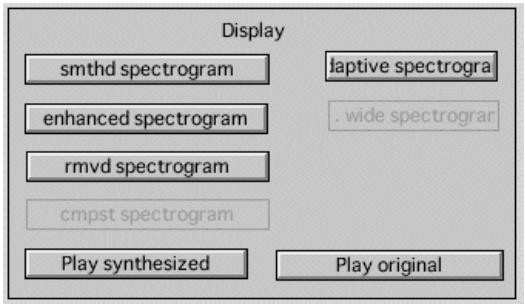

図 3.16: 合成が完了した後の表示サブパネル

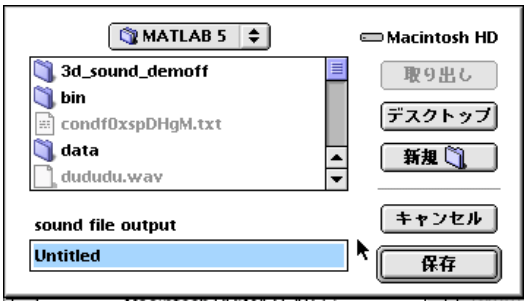

図 3.17: ファイル書出し用のダイアログ

また，同様に，表示用のサブパネルでも図 3.16 に示すように，合成音声出力用のボタンが使用可能になる．ここで,「Play synthesized」と「Play original」というボタンを用いて合成音声と原音声とを比較することができる．

合成音声をチェックして問題が無ければ，以降の利用のために，ファイルに書出すことができる．主操作サブパネルの「save to file」と書かれたボタンをクリックすると，機種に依存したファイル書き出し用のダイアログが表示される．図 3.17 には，Macintosh の場合のダイアログを示す．ここでディレクトリを選択し，ファイル名を指定して音声をファイルに書出すことができる．ファイル名の拡張子として'.wav' を選択した場合には，WAVE 形式でファイルが作成され，'.aif' を選択した場合には，AIFF 形式でファイルが作成される．それ以外の場合には，各機種に特有のバイト順で 16 bit の符号付き整数形式のバイナリでファイルが作成される[9]．

主操作サブパネルの「save to file」の横には,「up ENBL」と書かれたラジオボタンがある．これは，標本化周波数が 32000 Hz 以下である場合に，ファイルに出力する前に標本化周波数が 32000 Hz 以上でかつ 48000 Hz 以下となるようにオーバーサンプリングにより高い標本化周波数に変換する処理を行うことを指定するためのボタンである．このボタンが有効なのは，標本化周波数が 8000, 10000, 11025, 12000, 16000, 20000, 22050, 24000 Hz の場合のみである．任意の標本化周波数には対応していない．

[9]バイト順を明示的に指定する GUI は用意されているが，現在の版では Macintosh の場合にのみ GUI が表示される．このバグは軽易なものなので近々修理される予定である．

第4章　GUI-STRAIGHTの実装

この章では，前章で説明したアルゴリズムがどのようにSTRAIGHTに実装されているかについて，必要に応じて具体的なMatlabのコードを引用しながら説明する．

4.1　実装の概要

前章で説明したSTRAIGHTの開発には，数値演算と科学技術データの可視化環境であるMatlabが用いられている．そのため，全てのアルゴリズムはMatlabの関数として実装され試験されている．ただし，関数の集合だけでは，様々な試行錯誤が困難であるため，GUIを用いたインタフェースを作成した．このGUIを有するシステムを以下ではGUI-STRAIGHTと呼ぶことにする．このGUIも，MatlabのGUI作成環境を用いてプログラムされている．

STRAIGHTの旧版のサブセットがCで実装されているという報告がある．しかし，現状では，公式にMatlabの最新版に整合する版でCに実装されているものは把握されていない．

4.1.1　GUI-STRAIGHTの動作環境

GUI-STRAIGHTは，Matlab v5以上の版の上で動作する．GUI-STRAIGHTの全ては，mファイルとして実装されており，マシンのnative codeで実装されている部分は無い．したがって，Matlabが動いてさえいれば，些細な不具合をのぞき，GUI-STRAIGHTはマシンに依存せずに動作する．GUIを用いない関数の幾つかは，簡単な書き換えを行うことによって，PDSであるovtaveの上でも動作するものと思われる．

現在，GUI-STRAIGHTが動作している環境には，以下のものがある．Apple Macintosh, Windows 95, Windows 98, Windows NT, UNIX (SGI, Sun, Alpha). STRAIGHTは，メモリが贅沢にあることを想定しているので，300 MB以上の実メモリのマシンを用いることを勧める．また，STRAIGHTは計算資源を多く使用するので，できるだけ速いマシンを使うことを勧める．

4.2　GUI-STRAIGHTの構造

GUI-STRAIGHTは，いわゆるswitch board programのスタイルで書かれている．GUIの動作は，分析や合成の開始を指示する命令語をswitch boardプログラムの引き数として渡す形に統一されている．switch boardプログラムでは，渡された命令に応じて，一連の動作を実行する．具体的な演算や処理を行う関数は，GUIに依存せず，それぞれ単独で使用することができるように設計されている．

4.3　各関数の機能

ここでは，内部で利用されている関数を紹介し，それぞれの実装上の問題点についても触れる．

4.3.1　音源情報抽出

現在のSTRAIGHTの版では，基本周波数の抽出に以下の関数が用いられている．それらのプログラム名と機能の概要を表4.1に記す．

表 4.1: 音源情報抽出に関する関数.

関数名	概要
fixpF0VexMltpBG4	基本周波数抽出用関数
f0track5	有声／無声判定関数．チューニング不足.
refineF02	調波成分を利用した基本周波数の更新
aperiodicpart4	上側包絡と下側包絡の抽出
aperiodiccomp	非周期成分比率の計算

表 4.2: 音源情報を抽出するための Matlab 関数の help.

```
Fixed point analysis to extract F0
[f0v,vrv,dfv,nf]=
   fixpF0VexMltpBG4(x,fs,f0floor,nvc,nvo,mu,imgi,shiftm,smp,minm,pc,nc)
x       : input signal
fs      : sampling frequency (Hz)
f0floor : lowest frequency for F0 search
nvc     : total number of filter channels
nvo     : number of channels per octave
mu      : temporal stretching factor
imgi    : image display indicator (1: display image)
shiftm  : frame shift in ms
smp     : smoothing length relative to fc (ratio)
minm    : minimum smoothing length (ms)
pc      : exponent to represent nonlinear summation
nc      : number of harmonic component to use (1,2,3)
```

4.3.2　音源情報抽出における実装の問題

音源情報抽出のための最上位の関数は，fixpF0VexMltpBG4 である．しかし，この関数だけでは合成に必要なパラメタが揃わないため，後処理関数を用いてパラメタを揃えている．本来は，合成関数とペアになる形に整理しておくべきものである．

不動点に基づく方法の本体（fixpF0VexMltpBG4）

基本周波数抽出の第一段階であるこの関数の呼び出し部分は，以下の表 4.2 のようになっている．

wavelet 変換に用いる関数（multanalytFineCSPB）では，Gauss 型の包絡と spline 基底の双方を同じパラメタ η を用いて時間方向に 1.2 倍だけ伸長している．これは，他の調波成分の抑圧という主旨に整合しない処理である．この結果は，spline 処理を導入する代わりに伸長率を増加させるだけで実質的に他の調波成分を抑圧することができる可能性を示唆している．あるいは，低い周波数の位置に意図的に零を置くことに効果があるのかも知れない．

f0track5

音源情報抽出部分での問題点のまとめ

現在の実装では，音源の周期成分／非周期成分の制御のチューニングが不十分である．また，基本周波数抽出部分から有声／無声判定に到る部分のチューニングも不足している．これらは，それぞれの応用に応じてノウハウの

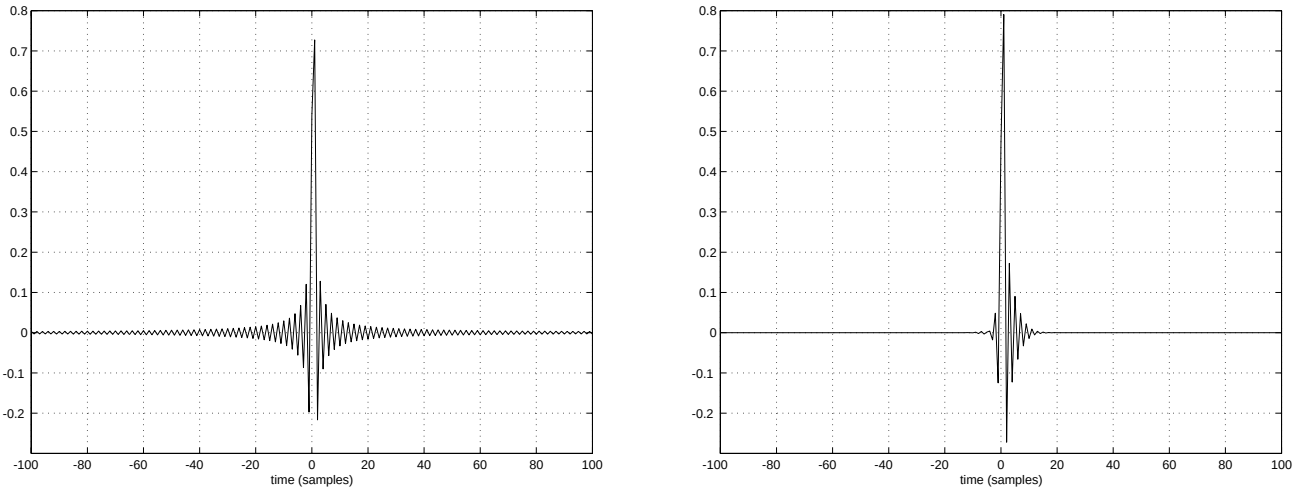

図 4.1: 群遅延を操作して標本化周期以下の遅延を加えたパルス．ナイキスト周波数での不連続がある場合（左）と，滑らかな関数で遷移させた場合（右）を示す．

蓄積によって改良して行くべき部分であろう．

4.3.3 平滑化スペクトル抽出

現在の STRAIGHT の版では，平滑化スペクトルの抽出に以下の関数が用いられている．それらのプログラム名と機能の概要を記す．

4.3.4 平滑化スペクトル抽出における実装の問題

現在の実装では，最適平滑化関数の利用，平滑化を適用する領域への非線形変換，インパルス応答の時間領域での補正等，重複する効果を有する仕掛けが含まれている．現在の既定値は，予備実験の結果に基づいて最良の品質の合成音声を作成できるように決められている．しかし，全体として最良の状態に近いパラメタ設定が実現されているとしても，それぞれの要素は，必ずしも最適な状態である保証は無い．

4.3.5 音声合成

現在の STRAIGHT の版では，音声合成に以下の関数が用いられている．それらのプログラム名と機能の概要を記す．

4.3.6 音声合成における実装の問題

群遅延制御

群遅延を操作する場合，離散時間系に実装するために，ナイキスト周波数で位相に不連続が生ずる場合がある．位相の不連続を残しておくとナイキスト周波数の振動が FFT 長にわたって残るとともに，フレーム周期の設定によっては，接続部分で不連続を発生させる．この様子を図 4.1 の左側の図に示す．この問題は，ナイキスト周波数での不連続を滑らかな関数を用いて接続することで軽減することができる．

実装に際しては，幾つかの条件を考慮する必要がある．滑らかに接続することは，ナイキスト周波数を中心とする部分に局所的に群遅延を加えることでもある．その結果，ナイキスト周波数での振動を局在化させることはできるが，ナイキスト周波数での群遅延の量を中心とした波束が生ずる．実装の条件は，この波束の性質に関するものである．それらを枚挙すると，以下のようにまとめられる．(1) 波束は，パルスに先行しないこと．(2) 波束は，拡

表 4.3: StraightCIv1.m から呼ばれる関数 (1)

関数名	概要
fixpF0VexMltpBG4	基本周波数抽出用関数
f0track5	有声／無声判定関数．チューニング不足．改良の余地大
refineF02	調波成分を利用した基本周波数の更新
straightBodyC03m	現行，スペクトル包絡の抽出
specreshape	時間領域処理によるスペクトル形状の補正
aperiodicpart4	上側包絡と下側包絡の抽出
straightSynthTB06c	現行，音声合成エンジン
aperiodiccomp	非周期成分比率の計算

がりができるだけ狭いこと．(3) 波束の位置は，パルスの位置にできるだけ近いこと．(4) 群遅延の遷移領域ができるだけ狭いこと．

まず，条件 (1) を満たす関数を設計する．滑らかに遷移させるための群遅延関数を $\Psi[k]$ とする．この群遅延関数とパルス位置が標本化の位置からずれる量 τ_d が以下を満たすようにする．

$$2\pi \quad = \quad \sum_{k}^{N} \left[2\pi \frac{\tau_d f s}{N} + \Psi[k] \right] \tag{4.1}$$

これより，条件 (2)，(3) と (4) が矛盾することが直ちに分かる．ナイキスト周波数が可聴周波数でない場合には，問題は無いが，そうでない場合には，これらの矛盾する条件の間の妥協点は，聴覚実験によって決定されるべきである．図 4.1 の右側に実装の一例を示す．

4.4　主要な関数と引数

ここでは，GUI-STRAIGHT のプログラムの構成を説明する．最上位のプログラムである straight.m は，StraightCIv1.m を初期化指令の引数とともに呼ぶ．StraightCIv1.m では，この初期化指令に応じて，最初に GUI を設定するための straightPanel98.m という関数を呼び出し，様々な初期値を設定する．その後は，GUI の操作に応じて関数を起動して行く．それぞれの GUI 要素に対応する動作については，使用例に譲り，ここでは機能と関数の対応を中心に説明する．

表 4.3 は，STRAIGHT の一連の処理に不可欠な関数を挙げる．これらは，適切なパラメタを設定することで，個別に使用することが可能である．表の中でインデントしてある関数は，不可欠ではあるが分析結果の修飾や後処理等，サポートの位置付けにあることを示す．

表 4.4 は，サポートのための関数を示す．これらは，STRAIGHT 専用というよりも一般的なユーティリティー用の関数である．AIFF の入出力用の関数は，Matlab に用意されていなかったために作成した関数である．AIFF の仕様を完全にはサポートしていない．圧縮を行わない単一のセグメントの読み書きのみをサポートしている．また，ProTools[1]等のプログラムとの間で 24 bit 音声データを交換できるように，24 bit までサポートしている．音声出力用の関数は，ローカルマシンに音声出力機能が無い場合にネットワークを介した音声出力機能を利用したり，標本化周波数に自動的に対応する低域フィルタを持たない D/A 変換装置を用いる場合のエイリアシングの問題を避けるためにアップサンプリングを行うために導入した関数である．STRAIGHTv30kr162 の版には，アップサンプリングを行う関数として実装されている．STRAIGHT を走らせるマシンの環境に応じてこの関数を書換えて使用することを想定している．

表 4.5 は，雑多な関数のリストである．これらは，実際に STRAIGHT の中で用いられている．しかし，不可欠ではない．例えば，mktstr.m は，これまでの改版の経緯から残っているが今となっては既存の関数で置き換えが可能である．dB.m と dBpower.m は，プログラムを見易くするために用いられている簡単に定義できる関数である．

[1]ProTools は，Digidesign 社の製品である．同社の web ページ (http://www.digidesign.com/) を参照のこと．

表 4.4: StraightCIv1.m から呼ばれる関数 (2)

関数名	概要
aiffread	AIFF 形式ファイルの読み込み
aiffwrite	AIFF 形式ファイルの書き出し
powerchk	有声部分のパワー計算
straightsound	音声出力関数．カスタマイズ用

表 4.5: StraightCIv1.m から呼ばれる関数 (3)

関数名	概要
bendline	不完全，GUI の実験用
boundmes2	表示用にのみ使用．周期領域と非周期領域の境界
dB	振幅に対する dB の計算
dBpower	パワーに対する dB の計算
getfsfrommenu	GUI 用関数.
mktstr	日付け関数
straightPanel98	GUI
syncgui	GUI 表示と内部状態の同期用

bendline.m と boundmes2.m は，デバッグや予備実験用に残してある関数である．残りは，GUI を動作させるために必要な関数であり，STRAIGHT による分析変換合成そのものには，不要である．

表 4.6 は，現在は動作しないボタンに割り付けられていたり，保守されていない関数を示す．これらは，将来の版では抹消される予定である．これらの関数は，これまでの検討の経緯の記録を兼ねるという意外に残す理由は無い．

表 4.6: StraightCIv1.m から呼ばれる関数 (4) 不要な関数

関数名	概要
BcorrMap	廃棄
bendline	不完全，GUI の実験用
getTrace	廃棄．旧版での interaction 用
rmv2nd	検討途中．スペクトル 2 次構造の除去
straightBodyB04m	旧版，使用せず
straightSynthTB06	旧版，二値的音源による音声合成エンジン
wfromMap4	廃棄，旧版での周期／非周期成分調整用

第5章　GUI-STRAIGHTとMatlab関数を用いた音声の加工例

5.1　はじめに

この章では，STRAIGHT を用いて音声を加工する様々な方法について，具体例を用いながら説明する．まず，最初に GUI-STRAIGHT で用いられている global 変数の具体的利用法の説明から始める．これらの変数を操作し，GUI-STRAIGHT の内部で用いている関数を直接呼び出して使うことにより，試験的に作成した GUI では不可能な自由度の高い変換が可能となる．

5.2　GUI-STRAIGHTのglobal変数の利用

GUI-STRAIGHT で用いられている global 変数の中から合成音声の加工に関連したものを表に示す．これらの変数を Matlab の command window あるいは利用者の関数の中で global として宣言することにより，情報を利用したり変更することが可能となる．以下では，具体例に基づきながら，使用法を明らかにしていく．

ここで紹介する方法では，操作法の説明を目的として冗長になることを厭わずにステップを踏んでいる．それぞれの応用では，ここで紹介した内容をヒントとして，専用の自動化されたプログラムを作成すことが必要となることに注意しておく．

5.3　基本周波数の操作

ここでは，まず簡単な例から始めて，段階的により柔軟な基本周波数の操作方法を説明して行く．

表 5.1: GUI-STRAIGHT の global 変数と簡単な説明

global 変数名	概要
fname	入力ファイル名
xold	入力音声データ（絶対値は 32768 以下）
fs	標本化周波数 (Hz)
f0shiftm	フレーム周期 (ms)
f0raw	基本周波数 (Hz)
nsgram	適応的スペクトログラム（干渉除去以前）
n3sgram	適応的平滑化スペクトログラム
gdbw	群遅延の標準偏差 (ms)
cornf	群遅延の境界周波数 (ms)

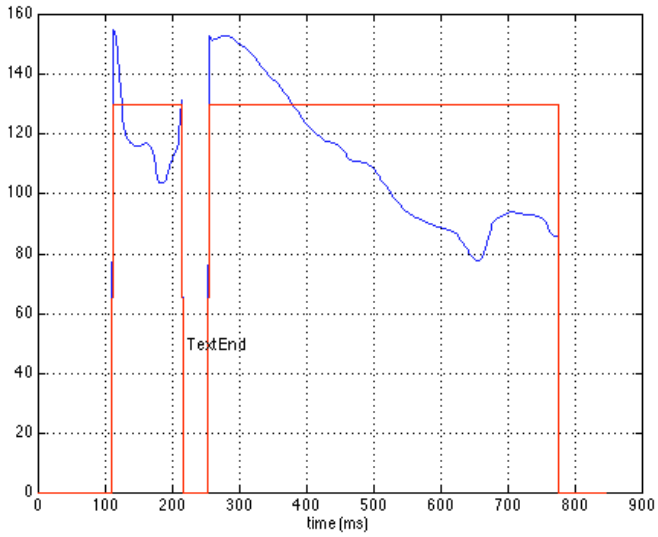

図 5.1: 有声部分の基本周波数を一定にする

5.3.1　簡単な例

一定の基本周波数への置換え

まず，有声部分を一定の基本周波数に置き換える例を挙げる．f0raw では無声部分が周波数 0 で表わされているので，周波数が 0 でない部分を一定の周波数にする．Matlab では，変数の添字として変数の満たすべき条件を書くことができることを利用すると，以上の処理の本体を一行で書くことができる．

STRAIGHT による分析は，終了しているものとする．ここで用いる音声は，前の例と同じく，男性が発声した「125」という数字音声である．

command window から以下を打ち込むことで処理が完了する..

```
global f0raw fs f0shiftm xold fname
f0bak=f0raw;
f0raw(f0bak>0)=f0bak(f0bak>0)*0+130;
tx=(0:length(f0raw)-1)*f0shiftm;
figure
plot(tx,f0bak,tx,f0raw,'r');grid on;
xlabel('time (ms)')
ylabel('fundamental frequency (Hz)');
```

すると，図 5.1 に示すような表示が得られる．図 5.1 の青線（濃色）は，元の基本周波数，赤線（淡色）は，変換された基本周波数を表わす．

ここで，以下の命令を打ち込むことにより，図 5.2 に示すように波形を表示させることができる．

```
global sy
figure
plot((0:length(sy)-1)/fs*1000,sy);grid on;
xlabel('time (ms)')
```

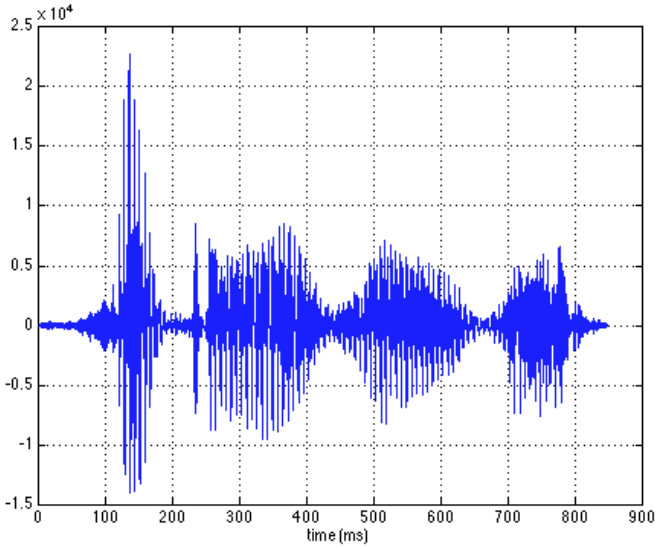

図 5.2: 一定の基本周波数から再合成した音声波形

5.3.2 基本周波数軌跡の局所的変更

次の例では，基本周波数の軌跡の一部分だけを変更する例を示す．ここでは，一定の割合いで基本周波数が増加するような成分を元の基本周波数の軌跡の 300 ms 以降の部分に加える．一定の傾斜は cumsum を利用して作成している．cumsum を用いることで，変更を加える区間の長さと，終端部分での周波数の増加量から簡単に変更量の計算を実装することができる．

以下のような命令を打ち込む．

```
modf=zeros(size(f0bak));
modf(301:400)=cumsum(ones(size(modf(301:400)))*100/100);
f0raw(f0bak>0)=f0bak(f0bak>0)+modf(f0bak>0);
plot(tx,f0raw,'r',tx,f0bak,'b');grid on;
xlabel('time (ms)')
```

これらの命令により，図 5.3 に示すような表示が得られる．変更された基本周波数の軌跡（赤線：淡色）と元の基本周波数の軌跡（青線：濃色）を以下に示す．

5.3.3 手作業による基本周波数情報の修正

精密な知覚実験に STRAIGHT を用いる場合には，基本周波数情報等の音源情報を修正しておく必要がある．ここでは，手作業による修正方法について説明する．

図 5.4 に基本周波数と元の音声信号を時間軸を揃えて重ねて表示する．この表示には，以下の命令を用いた．

```
global xold f0raw fs f0shiftm fname
figure
tx=(0:length(xold)-1)/fs*1000;
tfx=(0:length(f0raw)-1)*f0shiftm;
plot(tx,xold/32768*50+80,tfx,f0raw,'r');grid on;
xlabel('time (ms)');
```

この表示から明らかなように，f0raw の有声区間の判定と，波形から認められる有声区間とは細部に若干の違いが

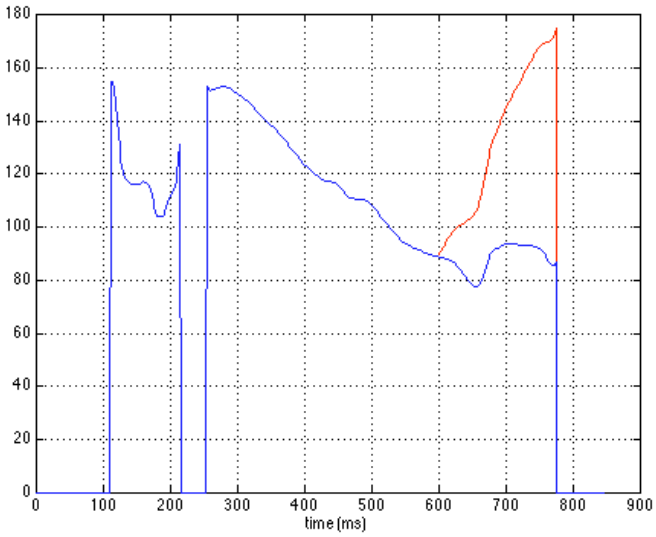

図 5.3: 局所的に線形に増加する成分を加えた基本周波数軌跡

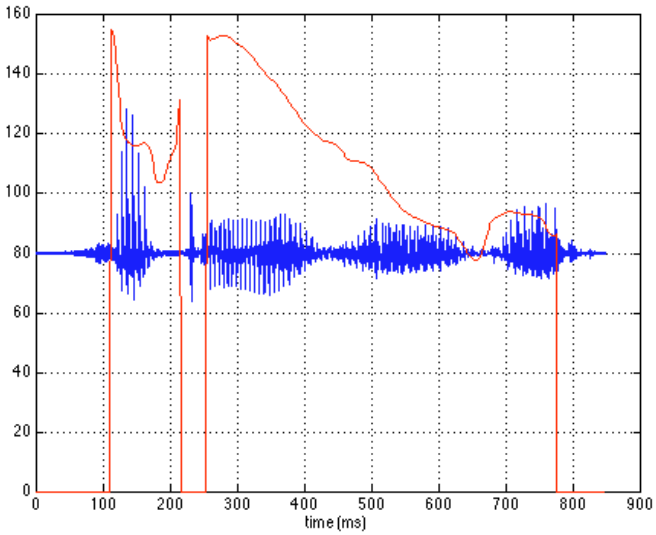

図 5.4: 音声波形と基本周波数情報の同時表示：全体

ある．一般的には，f0raw は，有声側に判断しがちな傾向を持っている[1]．

まず，最初の有声区間の修正から始める．zoom on 命令を打ち込んで，マウスのドラッグにより注目する区間を図 5.5 のように拡大表示する．マウスのドラッグの代わりに axis 命令を用いて，拡大表示する区間を直接指定しても良い．

図 5.5 から明らかなように，f0raw は有声区間を実際よりも広いものと判定している．視察によれば，有声区間の始まりは，118 ms 付近であるし，終了は 180 ms 付近のようである．以下の命令により，それ以外の部分を無声と判断するように f0raw の値を 0 とする．

```
f0bak=f0raw;
```

[1]前にも触れたように，STRAIGHT の実装における有声／無声判定論理は十分にチューニングされていない．素材としての不動点に基づく基本周波数抽出法は，GUI-STRAIGHT での実装よりも性能が高いことが分かっている [51]．プログラムによる自動処理に用いるためには，GUI-STRAIGHT の内蔵関数を流用するのではなく，専用の関数を利用することが望ましい．そのような関数の例を付録に示す．

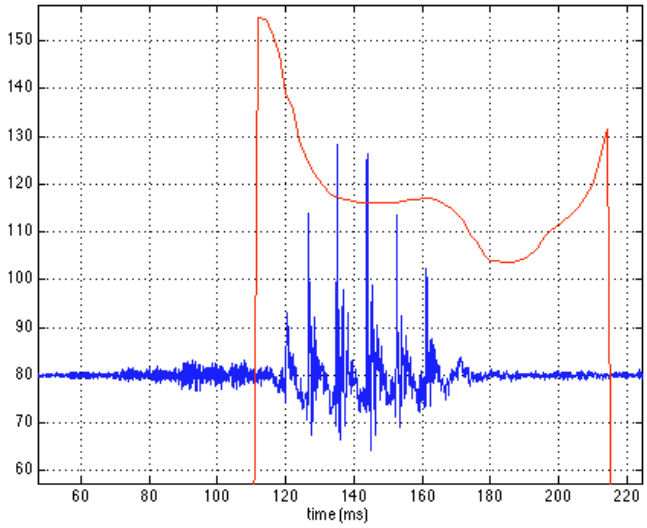

図 5.5: 音声波形と基本周波数情報の同時表示．ここでは，最初の有声部分を拡大表示している．

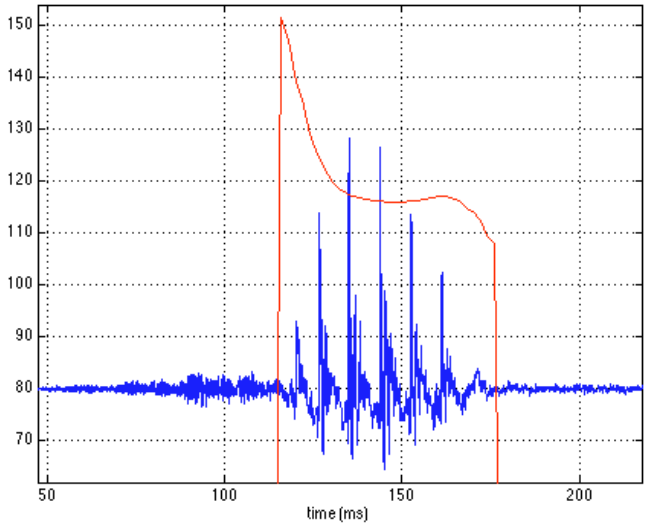

図 5.6: 音声波形と修正を加えた基本周波数情報の同時表示．ここでは，最初の有声部分を拡大表示している．

```
f0raw(50:58)=f0raw(50:58)*0;
f0raw(90:110)=f0raw(90:110)*0;
plot(tx,xold/32768*50+80,tfx,f0raw,'r');grid on;
xlabel('time (ms)');
zoom on
```

この操作によって，図 5.6 に示すように，実際の視察に合うように f0raw が修正される．

次の有声区間では，逆に f0raw は，有声部分を狭く判定してしまっている．図 5.7 に開始部分を示す．/ku.../の部分に相当するが，/k/の破裂の後の母音の開始から少し遅れて有声と判定されている．ここでは，視察により，不足している部分に一定の基本周波数の値を仮に与えることとする．

以下の命令により修正する．

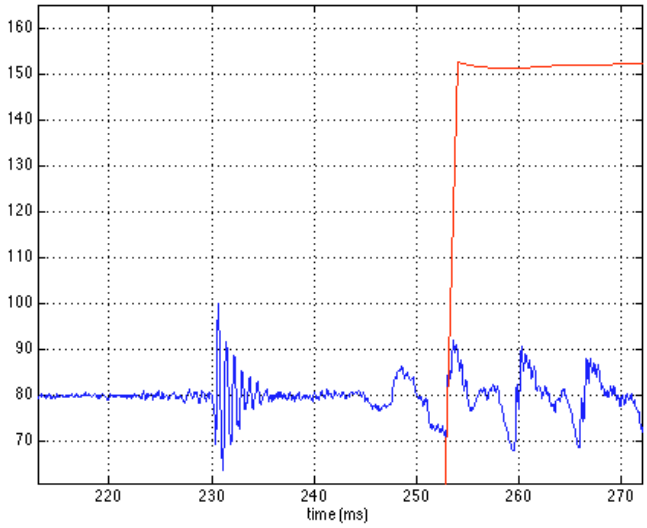

図 5.7: 音声波形と基本周波数情報の同時表示．ここでは，二番目の有声区間の最初の部分を拡大表示している．

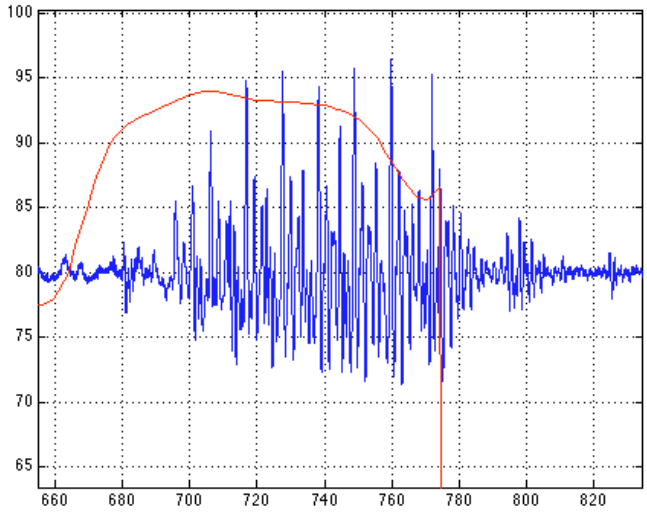

図 5.8: 音声波形と基本周波数情報の同時表示．ここでは，二番目の有声区間の終わり部分を拡大表示している．

```
f0raw(124:127)=f0raw(124:127)*0+160;
```

この区間の最後の部分は，難しい問題を含んでいる．図 5.8 に，この部分の音声波形と基本周波数を示す．図 5.8 から分かるように，780 ms 以降では，音声波形はきちんとした周期性を示していない．声帯振動が停止直前に不安定になっている部分なので，周期性ばかりではなく波形も大きく変わっている．

しかし，無声と判定するよりは，不完全でも有声情報を与えた方が適切であろう．ここでは，急速に基本周波数が低下するような軌跡を以下の命令によって追加することとする．ついでに表示も行う．

```
f0raw(387:405)=f0raw(387:405)*0+ ...
   85-cumsum(ones(size(f0raw(387:405)))+15/(405-387)/f0shiftm);
plot(tx,xold/32768*50+80,tfx,f0raw,'r');grid on;
xlabel('time (ms)');
```

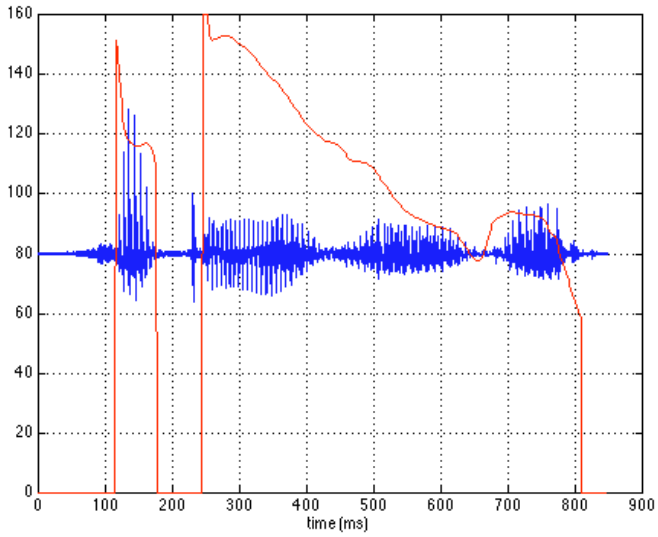

図 5.9: 音声波形と修正が終了した基本周波数情報の同時表示：全体

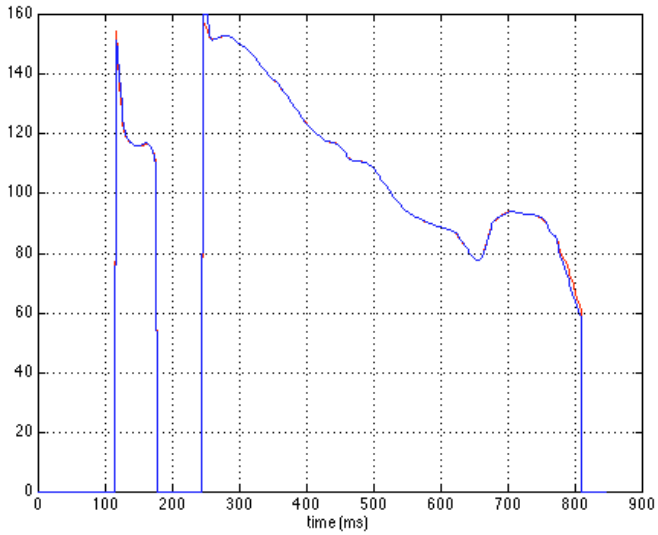

図 5.10: 手作業で修正した基本周波数情報（青線：濃色）と調波情報を用いて改良した推定値（赤線：淡色）

やや複雑に見えるが，cumsum を利用しているので，開始周波数と追加した区間での基本周波数の低下量を直接書込むことができる．

これらの変更の結果を図 5.9 に示す．

こうして修正した基本周波数を初期値とすれば，以下の命令により，第三調波成分までの情報を利用して基本周波数の推定値を改良することができる．

```
dn=floor(fs/(800*3*2));
[f0raw2,ecr]=refineF02(decimate(xold,dn),fs/dn,f0raw,512,1.1,3,f0shiftm,1,length(f0raw));
plot(tfx,f0raw2,'r',tfx,f0raw,'b');grid on;
xlabel('time (ms)');
```

結果を図 5.10 に示す．なお，この修正された基本周波数情報を用いて，'analyze MBX' による分析を再度行うこ

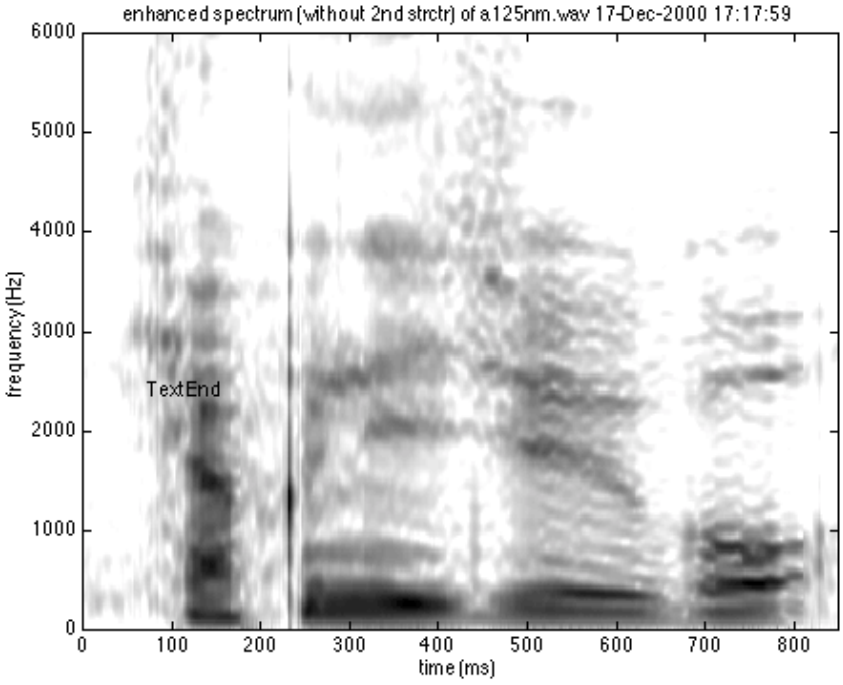

図 5.11: 修正された有声／無声情報と基本周波数情報によって求められた平滑化された時間周波数表現.

とができる．得られた情報のスペクトログラム状の表示を図 5.11 に示す．既定値を用いて無修正のまま求めた表現と比較すると，破裂子音の周辺や単語の末尾の乱れが改善されていることが分かる．ところで，図 5.11 の後半部分では，狭帯域スペクトログラムに見られるような基本周波数に並行して走る横縞のような構造が見える．これは，基本周波数の間隔の構造が見えているのではなく，見えているのは基本周波数の 2 倍の周波数の構造である．この構造は，基本周期の主要なイベントの中間にもう一つのイベントが生ずることによって形成されるものである．この構造を抑圧する手段は既に開発されている [58] が，音質に対する影響と適用基準の検討が済んでいないため，GUI からは外してある．プログラムの中には，その機能を使用するコードが残されているので，効果を試すことは可能である．

このような修正された基本周波数情報を用いることにより，より正確な，平滑化された時間周波数表現を得ることができる．これらの情報を.mat ファイル等に記録しておけば，以降の精密な知覚実験用の変換音声の作成のための基礎として用いることができる．

5.3.4　区分的一次関数による操作

前の節で紹介したアイデアに基づいて，より使い易い基本周波数の操作方法を導入する．

背景となるメカニズムが何らかの関数で近似できればそれらの関数のパラメタを操作することを介して基本周波数のパターンを変換することができる．基本周波数のパターンを変換する一つの方法は，基準となる時点毎の変化量を指定する方法であろう．ここでは最も簡単な関数として，区分的一次関数を用いて，基本周波数の操作量を指定することとする．

まず，最初に操作を加える f0raw のサイズを確認しておく．

```
>>size(f0raw)
ans =
1 424
```

ここでは，frame rate が 2 ms と設定されているため，424 は，846ms の時刻に対応していることに注意しておく．

表 5.2: 基本周波数制御のための操作点と操作量の設定表

時刻 (ms)	操作量 (Hz)
0	-30
400	0
650	30
850	120

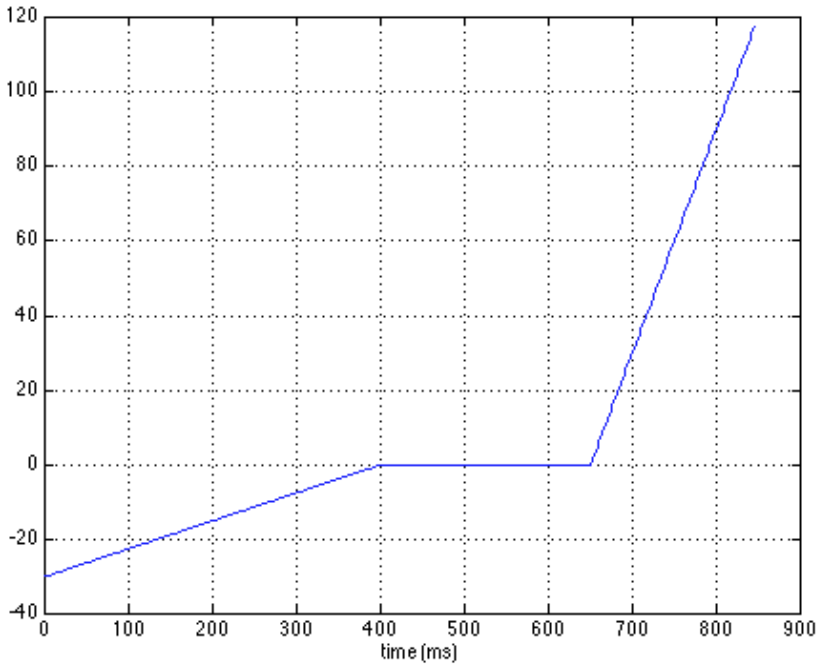

図 5.12: 区分的一次関数として作成した基本周波数操作量.

ここで，以下の表 5.2 の左の欄に示す操作点において，右の欄に示すだけの操作を加え，それぞれの操作点の間での操作量は一次関数で補間するものとする.

これらの操作は，以下の命令によって実行させることができる．まず，操作量のベクトルを作成し表示する.

```
mdf=interp1([0 400 650 700 850],[-30 0 0 30 120],tfx);
plot(tfx,mdf);grid on;
xlabel('time (ms)');
```

作成された操作量を図 5.12 に示す.

この操作量を有声部分にだけ加えて，図 5.13 に示すような変換した基本周波数軌跡を得る．ここで青線（濃色）が元の基本周波数軌跡，赤線（淡色）が変換された基本周波数軌跡を表わす.

5.4 周波数軸の操作

GUI-STRAIGHT の global 変数 fconv は，通常は周波数軸の伸縮量を表すパラメタでスカラー量である．この変数をスカラー量ではなくベクトルとすると，合成用の周波数軸から分析結果の周波数軸への写像の役割を果たすよ

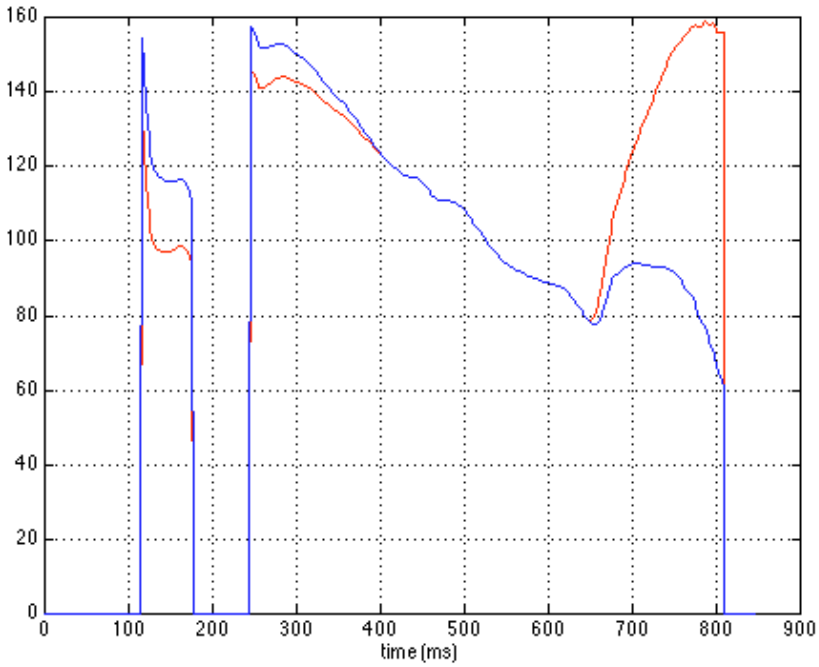

図 5.13: 区分的一次関数の操作量を用いて変換した基本周波数軌跡.

うになる．この方法を用いることで，個人性のような性質を簡単に変換することができる[2]．

5.4.1　環境の設定

ここでは，単独に発声された母音「ア」の変換を例として操作法を説明する．以下の操作に入る前に，音源情報の分析とスペクトル包絡の分析が終わっているものとする．合成用のボタンが使用可能になっていれば，この状態にある．

まず，利用する global 変数を宣言し，GUI-STRAIGHT の内部の情報が利用できるようにしておく．

```
global n3sgram fs fconv
```

今回の操作では，n3sgram, fs, fconv の 3 つの変数だけが対象となる．この段階では，fconv には「1」という数字が入っている．この fconv を写像を表すベクトルとするために，まず，ベクトルの長さとして設定すべき値を確認する．そのためには，スペクトログラムの周波数軸の長さを以下の command により調べれば良い．

```
size(n3sgram)
```

この例では，513 が周波数方向の長さであった．したがって，写像用のベクトルとして，513 要素のものを用意する．周波数軸は，1 番目の要素が 0 Hz に対応し，513 番目の要素が fs/2 に対応する．この例の場合は，22050 Hz に対応する．

5.4.2　指数関数を用いた例

まず，ここでは簡単な対応関係として，次に示されるものを考える．

$$f_{in}(f_{out}) = \frac{f_s}{2}\left(\frac{2f_{out}}{f_s}\right)^{\gamma} \tag{5.1}$$

[2]本格的な個人間の変換には，例えば時間周波数スペクトルの間の写像を求めなければならない [24]．しかし，もし，そのような本格的な写像を求めずに個人性のある程度一貫した変換が可能であれば，非常に有用である．また，このような簡単な変換で，例えば同一人物での話声と歌声の変換が可能であるならば，それも非常に有用である．

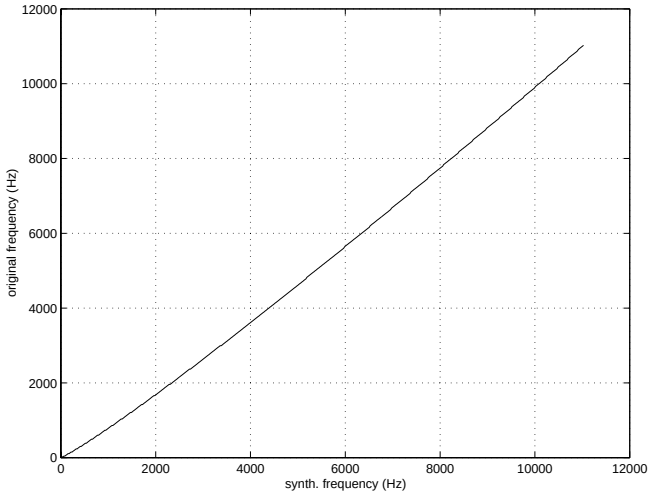

図 5.14: 合成時の周波数軸 f_{out} と分析時の周波数軸 f_{in} の対応関係を表す写像 (fconv).

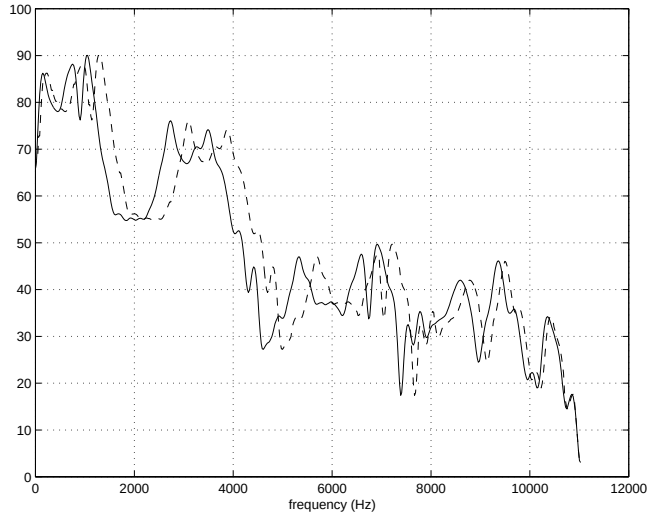

図 5.15: 元の母音「ア」のスペクトル包絡（実線）と，図 5.14 の写像を用いて変換されたスペクトル（破線）.

図 5.14 に $\gamma = 1.1$ とした場合の対応関係を示す．具体的には Matlab の command window で以下の命令を打ち込む.

```
fconv=(((0:512)/512).^1.1)*512+1;
fx=(0:512)/1024*fs;
plot(fx,fx(round(fconv)));grid on;
xlabel('synth. frequency (Hz)')
ylabel('original frequency (Hz)')
```

最後の４行は，グラフを表示するためのものである.

この対応関係は，見かけ上は僅かな変化しか無い．しかし，実際にこの対応関係で音声を変換すると，知覚的には全く別人の声に聞こえる程度の大きさである.

この対応関係を用いることで，母音「ア」のスペクトル包絡を図 5.15 に示すように変換することができる．実線が元のスペクトル包絡，破線が変換されたスペクトル包絡を示す．この非線形変換によってスペクトルのピークの位置が移動していることが分かる.

このグラフの表示には，以下の命令を用いた.

表 5.3: 母音「ア」のフォルマントのデータに基づいて設定した周波数軸の変換表.

分析時の周波数 (Hz)	合成時の周波数 (Hz)
750	720
1050	1310
2740	2500
3480	3800

```
plot(fx,dB(n3sgram(:,100)),'r');grid on;
hold on;plot(fx,dB(n3sgram(round(fconv),100)),'k--');grid on;
xlabel('frequency (Hz)')
```

この命令から分かるように，合成に使われる時間周波数表現は，$S_{SYN}(2\pi f_{out}, t) = S_{IN}(2\pi f_{in}(f_{out}), t)$ となる．ここで，S_{SYN} は，合成に使われる時間周波数表現を表し，S_{IN} は，分析結果として得られた時間周波数表現を表す.

以上のように fconv を設定した後に GUI の「syntheize graded」をクリックすれば，変換された時間周波数表現を用いた音声が合成される.

5.4.3 区分的一次関数を用いた例

もう少し，実際的な例として，フォルマント周波数の対応関係を変換したい場合を取り上げる．表 5.3 に今回分析した母音「ア」のフォルマント周波数と，梅田によるデータ [65] から求められた平均値との関係に基づいて設定した周波数の変換表を示す.

このような表に基づいて周波数の変換を行う場合に最も簡単な方法は，これらの周波数の間を直線で補間することである．具体的には，以下の command を実行させる.

```
tfx=[0 720 1310 2500 3800 fs/2];
ofx=[0 750 1050 2740 3480 fs/2];
foo=((0:512)/512)*fs/2;
fii=interp1(tfx,ofx,foo);
fconv=round(fii/fs*2*512)+1;
plot(fx,fx(round(fconv)));grid on;
xlabel('synth. frequency (Hz)')
ylabel('original frequency (Hz)')
```

これらを実行することにより図 5.16 が得られる．先ほどの例と比較すると，複雑な対応関係となっている.

また，この写像を用いて変換したスペクトル包絡と元のスペクトル包絡を図 5.17 に示す.

5.5 区分的一次関数による時間軸制御

GUI-STRAIGHT には実装されていない補助的関数を用いることにより，再合成音声の時間軸を自由に制御することができる．ここでは，元の音声の基準となる点と再合成音声の時刻との対応関係を定義し，それらを一次関数で結ぶことによって実現する．以下，順を追って説明する.

5.5.1 時間軸の非線形操作用音声合成関数

ここで用いる関数は，GUI-STRAIGHT で用いられている関数に時間軸の非線形制御のためにインタフェースを加えたものである．表 5.4 にその関数のインタフェース部分を示す.

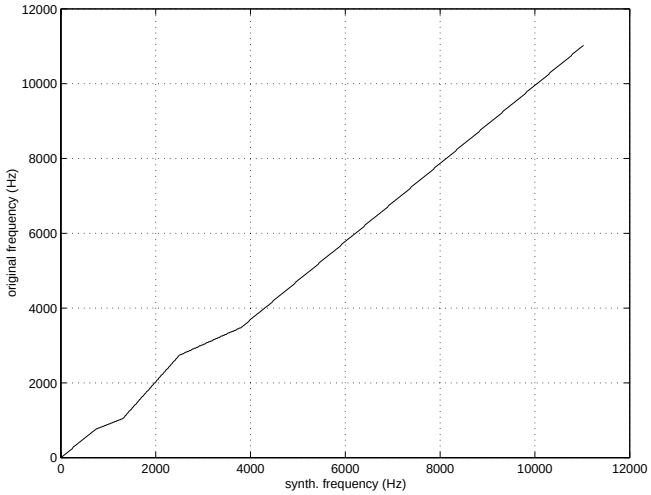

図 5.16: 合成時の周波数軸 f_{out} と分析時の周波数軸 f_{in} の対応関係を表す写像 (fconv)．写像は，表 5.3 に基づく．

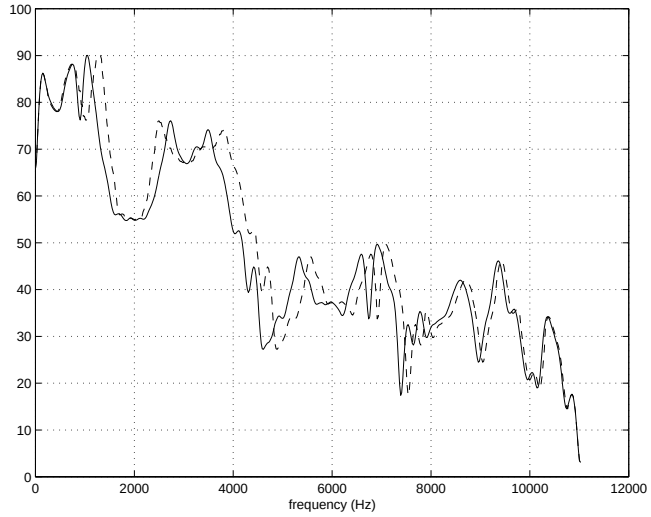

図 5.17: 元の母音「ア」のスペクトル包絡（実線）と，図 5.16 の写像を用いて変換されたスペクトル（破線）．

この関数を用いて時間軸の非線形の変換を行うに先立って，まず，GUI-STRAIGHT を用いて音源情報の分析とスペクトル包絡の分析を済ませておく[3]．その後，音声合成に必要となる global 変数を宣言しておく．

```
global fs f0shiftm f0raw n3sgram pcnv fconv sconv gdbw delfrac delsp cornf delfracind
```

また，この関数のインタフェースに挙がっている”ap”という引数は，非周期成分の比率を表すパラメタであり，GUI-STRAIGHT の内部で計算されている上側スペクトル包絡 (apv) と下側スペクトル包絡 (dpv) から求める必要がある．そのため，それらを global として宣言しておき，一連の命令を入力する．

```
global apv dpv

function ap=aperiodiccomp(apv,dpv,ashift,f0,nshift,fftl);
%[nn,mm]=size(nsgram);
mm=length(f0);
nn=fftl/2+1;
```

[3]分析後に “bypass” ボタンを押すことを忘れると，global 変数である n3sgram に値がセットされない．以下の説明では，global 変数に分析結果の値が入っていることを前提としている．

表 5.4: 時間軸の非線形制御を加えた音声合成関数.

```
function sy=straightSynthTB06ca(n2sgram,f0raw,shiftm,fs, ...
            pcnv,fconv,sconv,gdbw,delfrac,delsp,cornf,delfracind,ap,imap);
%    Straight synthesis with all-pass filter design based on
%       TEMPO analysis result
%   sy=straightSynthTB06c(n2sgram,f0raw,f0var,f0varL,shiftm,fs, ...
%               pcnv,fconv,sconv,gdbw,delfrac,delsp,cornf,delfracind,apv);
%       sy      : synthsized speech
%       n2sgram : amplitude spectrogram
%       f0raw   : pitch pattern (Hz)
%       f0var   : expected F0 variation with fricative modification
%       f0varL  : expected F0 variation
%       shiftm  : frame shift (ms) for spectrogram
%       fs      : sampling freqnency (Hz)
%       pcnv    : pitch stretch factor
%       fconv   : freqnency stretch factor
%       sconv   : speaking duratin stretch factor (overridden if || imap || >1 )
%       gdbw    : finest resolution in group delay (Hz)
%       delfrac : ratio of standard deviation of group delay in terms of F0
%       delsp   : standard deviation of group delay (ms)
%       cornf   : lower corner frequency for phase randomization (Hz)
%       delfracind      : selector of fixed and proportional group delay
%       ap      : aperiodicity measure
%       imap    : arbirtary mapping from new time (sample) to old time (frame)
```

```
[n2,m2]=size(apv);
x=(0:m2-1)'*ashift;
xi=(0:mm-1)'*nshift;
xi=min(max(x),xi);
ap=interp1q(x,(dpv-apv)',xi)';%,'*linear')';
```

なお，”apv”と”dpv”は，固定のフレーム間隔（5 ms）で求められている．

ここで計算した比率 (ap) は，global としては格納されてはいない．GUI-STRAIGHT 内部では，以下のように，呼び出しの引数として関数を介してその都度計算されている．

```
sy=straightSynthTB06c(n3sgram,f0raw,shiftm,fs, ...
   pcnv,fconv,sconv,gdbw,delfrac,delsp,cornf,delfracind, ...
   aperiodiccomp(apv,dpv,5,f0raw,f0shiftm,fftl));
```

5.5.2 GUI を用いない単純な音声合成

時間軸の非線形制御では GUI-STRAIGHT の内部の合成関数を用いることができない．そのため，まず，ここでは GUI のボタンをクリックした時と同じ操作を Matlab の command window から行う方法を紹介する．以下の例でもこれまでと同じく，男性によって発声された「125」という数字音声を用いる．

まず，以下をタイプする．

```
global sy
sy=straightSynthTB06c(n3sgram,f0raw,f0shiftm,fs, ...
  pcnv,fconv,sconv,gdbw,delfrac,delsp,cornf,delfracind, ...
  aperiodiccomp(apv,dpv,5,f0raw,f0shiftm,fftl));
```

ここで，”...”は，Matlab において行の継続を表す．この命令を入力すると，しばらく計算を続けた後，command window に「Done!」と表示される．これで，変数”sy”に合成された音声が格納されたことになる．

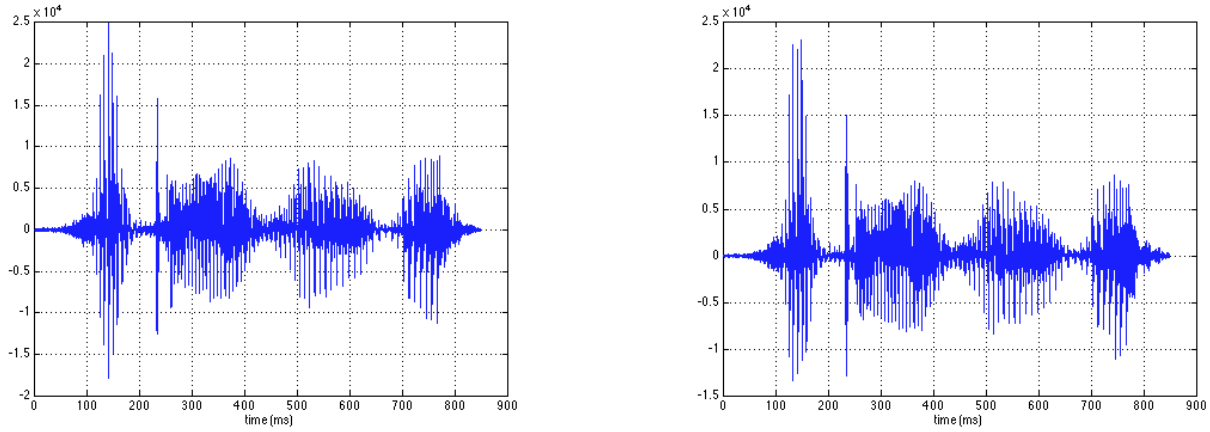

図 5.18: 音声合成関数の出力波形（左）と，正規化された合成音声波形（右）.

表 5.5: 基準時刻と合成音中の対応する時刻の定義

original (ms)	synthesized (ms)
0 to 200	0 to 200
200 to 400	200 to 500
400 to 600	500 to 600

ここまでで合成された音声のレベルは，ほぼ入力と同じになっている．通常の使用では，有声部分のレベルが同じになっていた方が使い易いので，GUI-STRAIGHT では，以下の処理を行って，最大レベルの-22 dB に平均レベルを合わせるようにしている．

```
dBsy=powerchk(sy,fs,15);
cf=(dB(32768)-22)-dBsy;
sy=sy*(10.0.^(cf/20));
```

同様にレベルを合わせたい場合には，これらの命令を Matlab の command window から入力する．図 5.18 に音声合成関数の出力と，正規化後の波形を比較して示す.

5.5.3 区分的一次関数による時間軸変型の指定

これで，Matlab の command window から音声合成を操作する手順が明かとなった．次に，時間軸を非線形に変換する簡単な方法を紹介する．

まず，表 5.5 のように，元の音声の時間軸と，変換された合成音声の時間軸の対応関係が表の形で与えられているものとする．このように指定された場合の最も簡単な対応関係は，対応点の間での時間変化が直線で表されるとするものである．要するに，区分的一次関数を非線形の時間軸の対応関数として用いるのである．

必要となる区分的一次関数 (imap) は，次のような Matlab の命令で作成することができる．

```
ogtm=[0 400 800 847];
trgt=[0 400 600 647];
otm=0:1000/fs:647;
imap=interp1(trgt,ogtm,otm);
figure
plot(imap)
plot((0:length(imap)-1)/fs*1000,imap);grid on
```

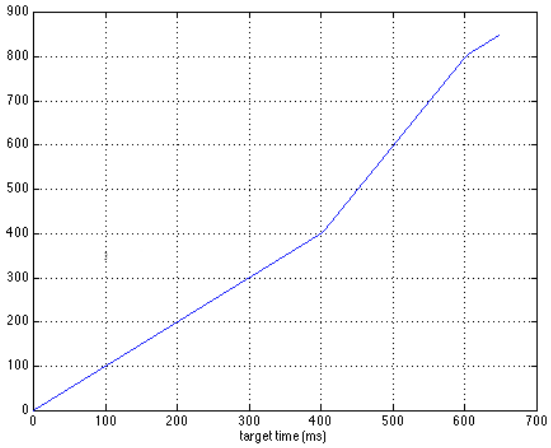

図 5.19: 表 5.5 に基づいて作成された区分的一次関数.

```
xlabel('target time (ms)')
ylabel('original time (ms)')
```

実際には，このようにして作成される”imap”は，標本化周波数で標本化された合成音声の時間軸の各時刻に対応する分析フレームの時刻（単位：ms）からなるベクトルである．図 5.19 に”imap”の内容を示す.

この区分的一次関数を用いて，以下の命令によって時間軸を変換した合成音声が作成される.

```
imap2=imap+1;
sy=straightSynthTB06ca(n3sgram,f0raw,f0shiftm,fs, ...
pcnv,fconv,sconv,gdbw,delfrac,delsp,cornf,delfracind, ...
aperiodiccomp(apv,dpv,5,f0raw,f0shiftm,fftl),imap2);
dBsy=powerchk(sy,fs,15); % 23/Sept./1999
cf=(dB(32768)-22)-dBsy;
sy=sy*(10.0.^(cf/20));
wavwrite(sy/32768,fs,16,'a125nmv1.wav');
```

こうして合成された音声の波形を図 5.20 に示す．指定された区間が短くなっているのが分かる．なお，この例では，ファイルへの書き出しまでを行っている[4].

5.5.4　区分的一次関数計算用の関数

ここで紹介した区分的一次関数による時間軸の変換は良く用いられるので，表から区分的一次関数を計算するための Matlab 関数 (makeimap) としてまとめてある．この関数を用いることにより，同様の処理が以下のように簡単になる.

```
ogtm=[0 400 800 847];
trgt=[0 400 600 647];
imap=makeimap(ogtm,trgt,1);
```

この関数の help を表 5.6 に示す.

[4]GUI-STRAIGHT では，音声情報は LSB（最小桁のビット）が 1 となるような尺度を用いているので，”wavwrite”関数の仕様と整合させるためには，16 bit の符号付き整数の最大の値で正規化する必要があることに注意.

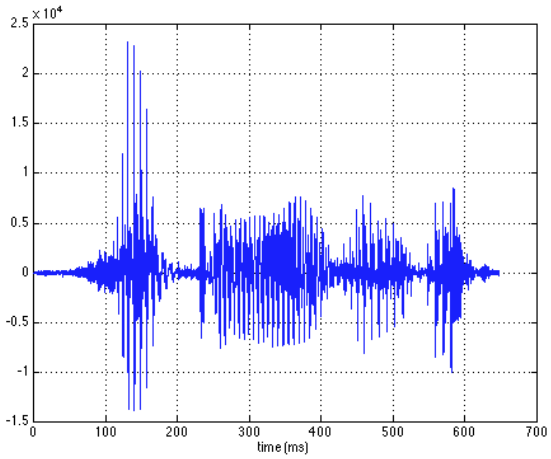

図 5.20: 区分的一次関数により変換された時間軸を用いて合成された音声波形.

表 5.6: 表から区分的一次関数を計算するための Matlab 関数の help.

```
   Piece-wise time axis mapping
   *** WARNING *** Prior to run this routine you have to
           perform GUI STRAIGHT analysis. This routine shares
           global variables
   imap=makeimap(orgpos,synthpos,opts)
input parameters
   orgpos          : list of anchor points of the original speech (ms)
   synthpos        : list of anchor points of the manipulated speech (ms)
   opts            : 0: calculates imap only
                   : 1: calculates imap and re-synthesis
                        This updates re-synthesized speech "sy".
output parameters
   imap            : mapping table from re-synth time (sample) to the
                        original time (frame number)
```

あとがき

STRAIGHT には，現在も改良のための検討が加えられている．この改良の中心は，現在は手動で設定しなければならない再合成用の音源の群遅延パラメタの自動設定に向けたものである．ここでは，イベントの検出と特徴付けが鍵を握ることとなる．それらの発展や，本資料作成後に明らかになったバグ等，本資料に載せきれなかった詳細は，以下のサポートページを参照されたい．ただし，このページは一般には公開されていないため，アクセス権の設定に関しては，個別に相談をお願いしたい

http://www.sys.wakayama-u.ac.jp/~kawahara/STRAIGHTtipse/

連絡先　相談内容に応じて，以下の連絡先を利用されたい．

- [技術的内容]

  住所：〒 640-8510 和歌山市栄谷 930 番地
  　　和歌山大学システム工学部 デザイン情報学科
  発明者：河原英紀
  E-MAIL：kawahara@sys.wakayama-u.ac.jp

- [商用化相談]

  住所：〒 619 — 0288　京都府相楽郡精華町光台 2-2
  （株）国際電気通信基礎技術研究所　開発室
  TEL：(0774) 95-1192
  FAX：(0774) 95-1179
  E-MAIL: deliv@ctr.atr.co.jp

関連資料

参考文献

[1] T. Abe, T. Kobayashi, and S. Imai. Harmonics estimation based on instantaneous frequency and its application to pitch determination. *IEICE Trans. Information and Systems*, Vol. E78-D, No. 9, pp. 1188–1194, 1995.

[2] T. Abe, T. Kobayashi, and S. Imai. Robust pitch estimation with harmonics enhancement in noisy environments based on instantaneous frequency. In *Proc. ICSLP 96*, pp. 1277–1280, Philadelphia, 1996.

[3] T. Abe, T. Kobayashi, and S. Imai. The if spectrogram: A new spectral representation. In *Proc. ASVA-97*, pp. 423–430, Tokyo, 1997.

[4] A. J. Abrantes, J. S. Marques, and I. M. Trancoso. Hybrid sinusoidal modeling of speech without voicing decision. In *Proceedings of Eurospeech 91*, pp. 231–234, Paris, 1991.

[5] Yoshinori Atake, Toshio Irino, Hideki Kawahara, J. Lu, S. Nakamura, and K. Shikano. Robust fundamental frequency estimation using instantaneous frequencies of harmonic components. In *Proc. ICSLP'2000*, PB(2)-26, pp. 907–910, Beijing China, October 2000.

[6] Hideki Banno, Jinlin Lu, Satoshi Nakamura, Kiyohiro Shikano, and Hideki Kawahara. Efficient representation of short-time phase based on group delay. In *Proc. ICASSP'98*, pp. 861–864, Seattle, 1998.

[7] J. Blauert and P. Laws. Group delay distortion in electroacoustical systems. *J. Acoust. Soc. Am.*, Vol. 63, No. 5, pp. 1478–1483, 1978.

[8] Boualem Boashash. Estimating and interpreting the instantaneous frequency of a signal – part 1: Fundamentals. *Proc. of IEEE*, Vol. 80, No. 4, pp. 520–538, 1992.

[9] Boualem Boashash. Estimating and interpreting the instantaneous frequency of a signal – part 2: algorithms and applications. *Proc. of IEEE*, Vol. 80, No. 4, pp. 550–568, 1992.

[10] F. J. Charpentier. Pitch detection using the short-term phase spectrum. *Proceedings of ICASSP'86*, pp. 113–116, 1986.

[11] L. Cohen. *Time-frequency analysis*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[12] H. Dudley. Remaking speech. *J. Acoust. Soc. Am.*, Vol. 11, No. 2, pp. 169–177, 1939.

[13] Thierry Dutoit and Henri Leich. An analysis of the performance of the MBE model when used in the context of a text-to-speech system. In *Proceedings of Eurospeech 93*, pp. 531–534, Berlin, 1993.

[14] A. El-Jaroudi and J. Makhoul. Discrete all-pole modeling. *IEEE Trans.*, Vol. SP-39, pp. 411–423, 1991.

[15] J. L. Flanagan and R. M. Golden. Phase vocoder. *Bell Syst. Tech. J.*, Vol. 45, pp. 1493–1509, 1966.

[16] Daniel W. Griffin and Jae S. Lim. Multiband excitation vocoder. *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. 36, No. 8, pp. 1223–1235, 1988.

[17] Hideki Kawahara. Gui-straight: Getting started. Technical Report TR-H-241, ATR Human Information Processing research laboratory, Kyoto Japan, 1988.

[18] Hideki Kawahara. Speech representation and transformation using adaptive interpolation of weighted spectrum: Vocoder revisited. In *Proceedings of IEEE int. Conf. Acoust., Speech and Signal Processing*, Vol. 2, pp. 1303–1306, Muenich, 1997.

[19] Hideki Kawahara, Yoshinori Atake, and Parham Zolfaghari. Accurate vocal event detection method based on a fixed-point analysis of mapping from time to weighted average group delay. In *Proc. ICSLP'2000*, Beijin China, 2000.

[20] Hideki Kawahara, Alain de Cheveigné, and Roy D. Patterson. An instantaneous-frequency-based pitch extraction method for high-quality speech transformation: revised TEMPO in the STRAIGHT-suite. In *Proc. 5th Int. Conf. on Spoken Language Processing (ICSLP '96)*, Vol. 1, Sudney, 1998.

[21] Hideki Kawahara, Haruhiro Katayose, Alain de Cheveigné, and Roy D. Patterson. Fixed point analysis of frequency to instantaneous frequency mapping for accurate estimation of F0 and periodicity. In *Proc. Eurospeech'99*, Vol. 6, pp. 2781–2784, 1999.

[22] Hideki Kawahara and Ikuyo Masuda. Spline-based approximation of time-frequency representation in straight method. *Technical Report of IEICE*, Vol. SP96-97, pp. 19–24, 1997. [in Japanese].

[23] Hideki Kawahara, Ikuyo Masuda-Katsuse, and Alain de Cheveigné. Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction. *Speech Communication*, Vol. 27, No. 3-4, pp. 187–207, 1999.

[24] Noriyasu Maeda, H. Banno, S. kajita, K. Takeda, and F. Itakura. Speaker conversion through non-linear frequency warpping STRAIGHT spectrum. In *Prod. Eurospeech'99*, Vol. 2, pp. 827–830, Budapest, Hungary, 9 1999.

[25] Robert J. McAulay and Thomas F. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Trans. ASSP*, Vol. 34, pp. 744–754, 1986.

[26] Amy T. Neel. *Factors influencing vowel identification in elderly hearing-impaired listeners*. Ph.D. dissertation, Department of Speech and Hearing Sciences and the Cognitive Science Program, Indiana University, 1998.

[27] A. Oppenheim and R. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, Englewood Cliffs, NJ, 1989.

[28] R. D. Patterson. A pulse ribbon model of monaural phase perception. *J. Acoust. Soc. Am.*, Vol. 82, No. 5, pp. 1560–1586, 1987.

[29] Roy D. Patterson, T. Anderson, and M. Allerhand. The auditory image model as a preprocessor for spoken language. In *Proc. Third ICSLP*, pp. 1395–1398, Yokohama, Japan, 1994.

[30] R. Plomp and H. J. Steeneken. Effects of phase on the timbre of complex tones. *J. Acoust. Soc. Am.*, Vol. 46, pp. 409–421, 1969.

[31] Jan Skoglund and W. Bastiaan Kleijn. On time-frequency masking in voiced speech. *IEEE Trans. on Speech and Audio Processing*, Vol. 8, No. 4, 2000.

[32] Malcolm Slaney, Michele Covell, and Bud Lassiter. Automatic audio morphing. In *Proceedings of IEEE int. Conf. Acoust., Speech and Signal Processing*, pp. 1–4, Atlanta, 1996.

[33] Yannis Stylianou, Jean Laroche, and Eric Moulines. High-quality speech modification based on a harmonic + noise model. In *Proceedings of Eurospeech 95*, pp. 451–454, Madrid, 1995.

[34] Sayoko Takano, Minoru Tsuzaki, and Hiroaki Kato. Perceptual Sensitivity to Temporal Distortion of Visual, Auditory and Bimodal Speech. *Journal of the Acoustical Socitey of Japan(E)*, Vol. 21, No. 1, pp. 41–43, 2000.

[35] Toshio Irino and Roy D. Patterson. Segregating information about the size and shape of the vocal tract using a time-domain auditory model: The Stabilised Wavelet Mellin Transform. *Speech Communication*, 2001. [to be published].

[36] Minoru Tsuzaki and Hideki Kawahara. Discrimination of 'time-stretched' pulse trains with asymmetric group delay patterns. In *Proc. WESTPRAC VII*, Kumamoto, 2000.

[37] Stefan Uppenkamp, Sandra Fobel, and Roy D. Patterson. Temporal integration and the perception of short frequency sweeps. In A. Schick, M. Meis, and C. Reckhardt, editors, *8th Oldenbrug Symposium on Psychological Acoustics*, pp. 353–372, BIS Oldenbrug, 2000.

[38] Stefan Uppenkamp, Roy D. Patterson, A. Rupp, M. Scherg, and T. Dau. The neural basis of the perception of temporal asymmetry in short frequency sweeps. In *12th International Symposium on Hearing*, ISH2000, Mierlo/Eindhoven, 3 2000.

[39] B. Yegnanarayana, C. d'Alessandro, and V. Darsinos. An iterative algorithm for decomposition of speech signals into periodic and aperiodic components. *IEEE Trans. on Speech and Audio Processing*, Vol. 6, No. 1, pp. 1–11, January 1998.

[40] Parham S. Zolfaghari and Hideki Kawahara. Sinusoidal analysis/synthesis on frequency-to-instantaneous mapping. 日本音響学会春季研究発表会, 1-7-3, pp. 203–204, 3 2000.

[41] Parham S. Zolfaghari and Hideki Kawahara. A sinusoidal model based on frequency-to-instantaneous frequency mapping. In *Proc. ICSLP'2000*, PAe(16,17)-J-11, Beijing China, 10 2000.

[42] Parham Zolfaghari, Yoshinori Atake, Kiyohiro Shikano, and Hideki Kawahara. Investigation of analysis and synthesis parameters of STRAIGHT by subjective evaluations. In *Proc. ICSLP'2000*, Ob(12)-F2-07, Beijin China, 2000.

[43] Parham Zolfaghari, 河原英紀. Subjective evaluation of STRAIGHT. 音響学会秋季講演論文集, 第 I 巻, pp. 193–194, 1999.

[44] 阿竹義徳, 河原英紀, 陸金林, 中村哲, 鹿野清宏. STRAIGHT の分析合成方式パラメタの主観評価による検討. 日本音響学会研究発表会講演論文集, 1-7-5, pp. 205–206, 2000.

[45] 阿竹義徳, 陸金林, 中村哲, 鹿野清宏, 河原英紀. STRAIGHT の分析合成方式パラメタの主観評価による検討. 音響学会春季講演論文集, 第 I 巻, pp. 205–206, 2000.

[46] 河原英紀. 駆動信号分析装置. 特許願２０００－５９８６１. 出願者：科学技術振興事業団, 出願：2000 年 3 月, (時間領域の不動点による音源情報抽出の基本特許).

[47] 河原英紀. 信号分析方法. 特許公開平１０－１９７５７５. 出願者：株式会社エイ・ティ・アール人間情報通信研究所, 出願：1997 年 1 月 14 日, (TEMPO の基本特許).

[48] 河原英紀. 信号分析方法. 特許公開２０００－１８１４７２. 出願者：科学技術振興事業団, 出願：1998 年 12 月 10 日, (周波数領域の不動点による基本周波数抽出の基本特許).

[49] 河原英紀, Alain de Cheveigné. 原理的に抽出誤りの存在しないピッチ抽出方法とその評価について. 信学技報, SP96-96, pp. 9–18, 1997.

[50] 河原英紀, Parham Zolfaghari. 群遅延情報を利用した音声の駆動情報の多重解像度分析について. 信学技報, EA2000-35, pp. 63–70, 8 2000.

[51] 河原英紀, Parham Zolfaghari. 不動点に基づく音源情報抽出法の評価について. 聴覚研究会資料, H-2000-80, 2000.

[52] 河原英紀, Parham Zolfaghari, Alain de Cheveigné, Roy D. Patterson. 周波数から瞬時周波数への写像の不動点を用いた音源情報の抽出について. 信学技報, SP99-40, 7 1999.

[53] 河原英紀, 増田郁代. 周期信号変換方法、音変換方法および信号分析方法. 特許公開平１０－９７２８７. 出願者：株式会社エイ・ティ・アール人間情報通信研究所, 出願：1996 年 12 月 24 日, (STRAIGHT の基本特許).

[54] 河原英紀, 増田郁代. 時間周波数領域での補間を用いた音声の変換について. 信学技報, EA96-28, 8 1996.

[55] 河原英紀, 増田郁代, 東山恵祐. 音声分析・変換・合成方法 STRAIGHT-TEMPO における相補的な時間窓の利用について. 信学技報, SP97-32, pp. 21–28, 1997.

[56] 河原英紀, 津崎実, Roy D. Patterson. オールパスフィルタの位相操作による時間構造制御とその知覚への影響について. 聴覚研究会資料, H-96-79, pp. 1–8, 1996.

[57] 河原英紀, 入野俊夫. 音源情報の抽出方法. 特許願平１１－１９２４３７. 出願者：科学技術振興事業団, 株式会社エイ・ティ・アール人間情報通信研究所, 出願：1999 年 7 月, (周波数領域の不動点による音源情報抽出の基本特許).

[58] 河原英紀, 片寄晴弘. 音声分析変換合成法 STRAIGHT における音源情報の精密化について. 信学技報, SP97-112, pp. 31–38, 2 1998.

[59] 河原英紀, 片寄晴弘, Roy D. Patterson, Alain de Cheveigne. 瞬時周波数を用いた基本周波数の高精度の抽出について. 聴覚研究会資料, H-98-116, 12 1998.

[60] 河原英紀, 山田玲子, 久保理恵子. STRAIGHT を用いた音声パラメタの操作による印象の変化について. 聴覚研究会資料, H-97-63, 9 1997.

[61] 河原英紀. 自然性の極めて高い音声分析変換合成法. 音声研究, Vol. 2, No. 2, pp. 28–36, 1998.

[62] 河原英紀. 聴覚の情景分析が生んだ高品質 vocoder: Straight. 日本音響学会誌, Vol. 54, No. 7, pp. 521–526, 1998.

[63] 河原英紀, 阿竹義徳. 音声の群遅延特性に基づく声門閉止等のイベント抽出について. 信学技報, SP99-171, 2000.

[64] 河原英紀, 栃内香次, 永田邦一. 小区間の線形予測分析とその誤差評価. 日本音響学会誌, Vol. 33, No. 9, pp. 470–479, 1977.

[65] 鳥井規子. 日本語ソナグラムについての若干の考察. 通研経過資料, 579 号, 日本電信電話公社電気通信研究所, 1957. （電子情報通信学会編：『聴覚と音声』の書誌情報より）.

[66] 阿竹義徳, 入野俊夫, 河原英紀, 陸金林, 中村哲, 鹿野清宏. 調波成分の瞬時周波数を用いたピッチ推定方法の検討. 信学技報, SP99-170, 3 2000.

[67] 阿竹義徳, 入野俊夫, 河原英紀, 陸金林, 中村哲, 鹿野清宏. 調波成分の瞬時周波数を用いたピッチ推定方法. 電子情報通信学会論文誌, Vol. J83-D-II, No. 11, pp. 2077–2086, 2000.

[68] 櫻庭京子, 今泉敏, 筧一彦. 感情表現が言語的制約に及ぼす影響の発達的検討. 信学技報, SP2000-39, pp. 53–59, 2000.

[69] 高野佐代子, 津崎実, 加藤宏明. 発話の時間構造知覚における視聴覚の情報統合－時間知覚における聴覚優位の再発見. 日本音響学会誌, Vol. 56, No. 10, pp. 683–694, 2000.

[70] 坂野秀樹, 陸金林, 中村哲, 鹿野清宏, 河原英紀. 時間領域平滑化群遅延を用いた短時間位相の効率的表現方法. 電子情報通信学会論文誌. （採録決定）.

[71] 後藤真孝. 音楽音響信号を対象としたメロディーとベースの音高推定. 電子情報通信学会論文誌 D-II, Vol. J84-D-II, No. 1, pp. 12–22, 2001.

[72] 吉岡大祐, 米崎正, 陸金林, 中村哲, 鹿野清宏. ボコーダ型音声分析合成系 STRAIGHT による低ビットレート音声符号化. 電子情報通信学会技術研究報告, SP99-132, pp. 7–12, 2000.

[73] 戸田智基, 坂野秀樹, 梶田将司, 武田一哉, 板倉文忠, 鹿野清宏. 側抑制性重み付けを用いた雑音環境下における STRAIGHT 分析合成系の品質改善. 信学論, Vol. J83-DII, No. 11, pp. 2180–2189, 2000.

[74] 都木徹, 清山信正, 宮坂栄一. 複数の窓幅から得られた自己相関関数を用いる音声基本周波数抽出法. 電子情報通信学会論文誌 A, Vol. J80-A, No. 9, pp. 1341–1350, 1997.

[75] 東山, 陸, 中村, 鹿野, 河原. 4 khz 帯域 STRAIGHT の品質評価と情報圧縮について. 日本音響学会研究発表会講演論文集, 1-2-6, pp. 207–208, 1997.

[76] 内田照久. 音声の発話速度の制御がピッチ感及び話者の性格印象に与える影響. 日本音響学会誌, Vol. 56, No. 6, pp. 00–00, 2000.

[77] 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. 信学全大, 5, p. 263, 1978.

[78] 濱上知樹. 音源波形形状を高調波位相により制御する音声合成方式. 日本音響学会誌, Vol. 54, No. 9, pp. 623–631, 1998.

索 引