UM ESTUDO DE MODELOS BÁSICOS DE PROSÓDIA PARA O PORTUGUÊS BRASILEIRO

Solimar de Souza Silva

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

Prof. Sérgio Lima Netto, Ph.D.

Prof. Abraham Alcaim, Ph.D.

Prof. Márcio Nogueira de Souza, D.Sc.

RIO DE JANEIRO, RJ - BRASIL
MAIO DE 2004

SILVA, SOLIMAR DE SOUZA

Um Estudo de Modelos Básicos de Prosódia para o Português Brasileiro [Rio de Janeiro] 2004

IX, 82 pp., 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia Elétrica, 2004)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1.Síntese de Voz 2.Modelo de Fujisaki 3.Prosódia

I.COPPE/UFRJ II.Título (série)

Agradecimentos

Quando lemos qualquer trabalho acadêmico, costumamos subestimar o esforço empreendido para a sua conclusão. Não me refiro tão somente ao empenho do mestrando, no caso, mas também ao de todas as pessoas que contribuem para criar a infraestrutura necessária.

Infelizmente não posso agradecer aqui a todas as pessoas que ajudaram a criar tal infraestrutura, por motivos práticos. Mas seria injusto se não destacasse algumas delas.

Primeiro, agradeço a todos os meus amigos, em especial aqueles que conheci no curso técnico do CEFET, durante o curso de graduação e no Laboratório de Processamento de Sinais. E à minha ex-namorada, que me proporciou conforto com a sua companhia, nos momentos em que as atribulações da vida pareciam intransponíveis. Eles ajudaram-me a superar um período extremamente conturbado de minha existência.

Meu orientador, Sérgio Lima Netto, foi a pessoa que mais ajudou-me neste trabalho. Ele mostrou o caminho que deveria ser seguido, orientando-me de fato. Efetuou a façanha de manter-me motivado, mitigando parte do meu pessimismo, e as duras críticas que fez (sem perder a ternura jamais) aos vários erros que cometi foram muito importantes.

Também sou grato a todos os professores e alunos do LPS, que justificam a existência do laboratório. Sinto-me honrado de ter compartilhado o ambiente do LPS com pessoas de grande caráter e capacidade.

As idéias, conhecimento e opiniões de vários pesquisadores, com os quais eu tive contato direto ou indireto, alguns deles citados no texto, também foram de grande importância para este trabalho. Todos eles merecem minha gratidão.

Finalmente, não posso deixar de agradecer a ajuda financeira prestada pela CAPES, que foi de grande valia.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

UM ESTUDO DE MODELOS BÁSICOS DE PROSÓDIA PARA O PORTUGUÊS BRASILEIRO

Solimar de Souza Silva

Maio/2004

Orientadores: Sérgio Lima Netto

Programa: Engenharia Elétrica

Algumas das aplicações dos sistemas de conversão texto-fala (TTS) são: auxílio a deficientes visuais e pessoas que possuem disfunções da fala, interfaces de voz de sistemas computadorizados. A prosódia é muito importante para aumentar a naturalidade dos sistemas TTS.

O objetivo principal desta tese é o estudo de modelos básicos de prosódia para sistemas de conversão texto-fala para o português brasileiro. Outro objetivo é estudar as características de vários sintetizadores para o português brasileiro.

É proposto um novo algoritmo para inversão do modelo de Fujisaki, e um módulo de prosódia simplificado é implementado para um sistema de conversão texto-fala baseado em sílabas.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

A STUDY OF BASIC PROSODY MODELS FOR THE BRAZILIAN PORTUGUESE

Solimar de Souza Silva

May/2004

Advisors: Sérgio Lima Netto

Department: Electrical Engineering

Some of the applications of text-to-speech systems (TTS) are: help for visually handicapped and people having speech disfunctions, speech interface of computerized systems. The prosody is very important to increase the naturality of the TTS systems.

The main objective of this thesis is to study basic models of prosody in text-to-speech systems for the brazilian portuguese. Another objective is to study the characteristics of several synthesizers for the brazilian portuguese.

It is proposed a new algorithm of Fujisaki model inversion, and a simplified prosody model is implemented for a text-to-speech system based on syllables.

Sumário

Lista de Figuras

Capítulo 1

Introdução

1.1 Objetivos

O objetivo principal deste trabalho é estudar modelos básicos de prosódia para a conversão texto-fala para a língua portuguesa falada no Brasil. Um sistema de conversão texto-fala (*text-to-speech*, TTS) é um sistema capaz de converter uma representação do texto, na forma de uma seqüência de caracteres (letras, números, sinais de pontuação etc), em voz. Estes sistemas possuem diversas aplicações [1], tais como interfaces de voz, sistemas de leitura automática para deficientes visuais e auxílio a pessoas acometidas por disfunções da fala.

Devido à inerente complexidade do projeto de um sistema TTS, é indispensável dissecar o problema de síntese de voz a partir de texto em problemas menores que podem ser resolvidos satisfatoriamente. Por isso, tais sistemas têm uma organização modular que facilita o projeto.

Além do objetivo anteriormente exposto, também tinha-se em vista a implementação de um módulo simples de prosódia. Este módulo foi inserido no sistema TalkActive, um sistema TTS baseado em sílabas, descrito no Capítulo 4. O módulo faz uma análise lingüística simples do texto e usa o modelo de Fujisaki para gerar o contorno de *pitch* do enunciado a ser sintetizado.

O modelo de Fujisaki, um modelo acústico de entonação, é estudado em maiores detalhes neste trabalho. É um modelo bem simples, baseado nas restrições fisiológicas da laringe e utilizado com sucesso em várias linguagens. Seus parâmetros de entrada têm forte correlação com as informações lingüísticas de alto nível ligadas

ao enunciado.

A complexidade da conversão texto-fala acarreta, além da necessidade de dividir o problema em parcelas menores, outra imposição: a construção de alguns módulos do sistema é extremamente dependente da linguagem alvo. Assim, é necessário fazer restrições rígidas à linguagem alvo do sistema, definindo precisamente a língua, o dialeto e o sotaque. De outra forma, o problema seria intratável com as técnicas atuais. Neste trabalho optou-se pela língua portuguesa falada no Brasil (*português brasileiro*, PB), na cidade do Rio de Janeiro (sotaque carioca).

Outro objetivo do trabalho era realizar uma pesquisa sobre sistemas TTS para o PB. Vários sistemas TTS, comerciais e acadêmicos, foram analisados. Algumas das informações mais importantes obtidas para estes sistemas são apresentadas neste trabalho, em especial na forma de uma tabela que resume as características principais.

O problema da inversão do modelo de Fujisaki, ou seja, da obtenção dos comandos do modelo (entrada) dado o contorno de F0 (saída), é também abordado. É elaborada uma nova forma de visualizar o problema. Todos os algoritmos de inversão propostos outrora faziam uma estimativa inicial dos comandos, que era aperfeiçoada mediante métodos de otimização iterativos. Propõe-se um algoritmo que obtém as amplitudes dos comandos em forma fechada, e que pode aumentar a velocidade de convergência da otimização iterativa, bem como diminuir o erro da aproximação.

1.2 Motivação

Atualmente já existem sistemas TTS que possuem ótima inteligibilidade, mas a voz sintetizada por estes sistemas ainda carece da naturalidade da voz humana, sendo considerada pelo ouvinte como enfadonha, depois de ser ouvida durante algum tempo. A artificialidade da voz sintética é uma conseqüência da inacurada modelagem da prosódia nestes sistemas. Modelos de prosódia mais acurados seriam capazes de gerar voz mais natural.

Isto se explica pelo fato de que a prosódia permite que grande quantidade de informações lingüísticas de alto nível sejam embutidas na voz através da evolução

apropriada de parâmetros acústicos ao longo do tempo, como a freqüência fundamental, a energia ao longo de um enunciado, as durações dos segmentos etc. A voz natural tem uma riqueza lingüística superior à da voz sintética dos sistemas TTS atuais, devido à simplicidade dos modelos de prosódia.

Isto mostra porque a pesquisa hoje se concentra bastante na tarefa de obter modelos de prosódia adequados para um sistema TTS. Tais modelos ainda são incapazes de realizar corretamente a integração das informações lingüísticas para inserí-las na voz. A prosódia é o calcanhar de Aquiles dos sistemas TTS de alta qualidade, em qualquer linguagem.

Devido à dificuldade em derivar regras para realizar o processamento prosódico, uma das direções seguidas nas pesquisas em prosódia para sistemas TTS é a inferência do modelo a partir de corpora de fala anotados. Através de técnicas como, por exemplo, árvores de decisão ou redes neurais, tenta-se encontrar um mapeamento que leve uma representação lingüística de alto nível em uma representação de mais baixo nível. Para isso é necessário, durante a fase de treinamento, fornecer exemplos do mapeamento retirados do corpus. Esta abordagem é conhecida como *top-down*.

Para a construção do módulo de prosódia baseado no modelo de Fujisaki, a solução do problema de inversão tem grande importância, já que encontrar manualmente os comandos do modelo a partir da freqüência fundamental não é uma tarefa trivial. Por outro lado, a automatização do processo de inversão permitiria ter acesso a maior quantidade de dados para treinar um sistema de processamento prosódico, o que é de grande valia no caso da abordagem *top-down*.

Neste trabalho foi construído um módulo simples de prosódia, sem ter acesso a um corpus anotado de fala. Este módulo utiliza o modelo de Fujisaki e algumas informações lingüísticas triviais. Com esta implementação podemos verificar a qualidade suprasegmental do sistema resultante da incorporação de um módulo de prosódia simplificado.

Também foi realizada uma pesquisa sobre sistemas TTS para o PB, que permite que saibamos em que ponto estão os sistemas desenvolvidos para esta língua. Isto nos dá um bom referencial tanto para este como para outros trabalhos em conversão texto-fala baseados no PB.

1.3 Estrutura da tese

O Capítulo 2 descreve alguns dos sistemas de TTS desenvolvidos para o PB e suas características. Este capítulo tem como objetivo mostrar o atual estado da tecnologia de síntese de voz a partir de texto para a língua portuguesa falada no Brasil.

O Capítulo 3 apresenta o conceito de prosódia e descreve modelos de prosódia, que incluem os modelos de duração e de entonação. O modelo de Fujisaki, que descreve o contorno de F0 através de um conjunto de parâmetros com certa significação lingüística, é descrito neste capítulo. Este modelo será usado no TalkActive para obter o contorno de F0 do enunciado a ser sintetizado.

O Capítulo 4 descreve o sistema de conversão texto-fala TalkActive, as experiências realizadas para estudar a modificação dos parâmetros prosódicos e mostra a implementação do modelo de entonação neste sistema, usando o modelo de Fujisaki e informações lingüísticas triviais do texto.

No Capítulo 5 são descritos alguns algoritmos de inversão do modelo de Fujisaki, e é sugerido um algoritmo que se aproveita da relação quadrática entre as amplitudes dos comandos do modelo e o erro da aproximação para obter analiticamente as amplitudes dos comandos que minimizam o erro da aproximação, dados os tempos dos comandos.

Por fim, o Capítulo 6 apresenta as conclusões do trabalho e sugestões de trabalhos futuros.

Capítulo 2

Estado da Arte em TTS para o PB

2.1 Introdução

Neste capítulo incluiremos alguns dos principais sistemas de síntese de voz a partir de texto para o português brasileiro, expondo-se as informações obtidas sobre cada sistema. Desta forma podemos ter uma visão geral da atual tecnologia de síntese de voz para a língua portuguesa falada no Brasil.

O conhecimento obtido com a análise de outros sistemas TTS pode nos mostrar alguns dos problemas encontrados na implementação de tais sistemas e até mesmo soluções adotadas para resolvê-los. Além disso, tal estudo poderá nos indicar as tendências da área, tanto as de pesquisa como também as de implementação e comerciais.

As seções iniciais constituem uma introdução a sistemas TTS, que apresenta os principais conceitos da área de síntese de voz. São analisadas as abordagens para resolver o problema de síntese de voz a partir do texto, a questão da escolha do tamanho das unidades e os fatores que influenciam na naturalidade dos sistemas TTS.

Os sistemas TTS analisados são descritos na Seção 2.3. Além do histórico e da filosofia de alguns sistemas, são vistas algumas características técnicas, como o tamanho das unidades, o algoritmo de modificação prosódica, a presença de processamento prosódico etc. As características dos sistemas são apresentadas, de forma resumida e de fácil visualização, na Seção 2.4, na forma de uma tabela.

2.2 Sistemas TTS

2.2.1 Abordagens usadas em TTS

Os sistemas TTS capazes de sintetizar enunciados arbitrários, os quais podemos chamar de sistemas de *síntese arbitrária* ou sistemas de *vocabulário ilimitado*, são os que despertam maior interesse dos pesquisadores. Deve-se salientar que o problema de síntese de voz pode ser resolvido com relativa facilidade, gerando voz sintética com alta inteligibilidade e naturalidade, quando o domínio da aplicação é restrito. Isto porque nestes casos a voz sintetizada pode ser obtida através da concatenação de palavras ou frases, pois é possível gravar todos os trechos de voz necessários para a síntese, em todos os contextos possíveis. Esta facilidade é garantida devido ao vocabulário limitado e o emprego de alguns poucos contextos [1]. Porém, no caso em que o sistema deve ser capaz de sintetizar qualquer enunciado, gravar todas as palavras da língua em todos os contextos possíveis seria inviável, e portanto outras soluções devem ser adotadas [1].

O problema de síntese arbitrária é resolvido dividindo-o em vários blocos que são implementados independentemente. Esta abordagem é motivada pela concepção do estruturalismo de que o fenômeno da linguagem pode ser descrito em diferentes níveis. Dividir este problema em várias partes é fundamental.

Uma das características mais importantes da linguagem humana é a possibilidade desta de ser segmentada em um número finito de unidades menores com significado que podem ser combinadas para transmitir uma mensagem [2]. Ou seja, apesar da natureza contínua do sinal de voz, podemos estabelecer, em cada língua, um conjunto finito de símbolos que podem ser associados a certos trechos de voz. A partir do momento em que descobrimos um mapeamento razoável que leva os símbolos contidos no texto (grafemas) nas unidades abstratas elementares que compõem uma dada língua (fonemas e informações lingüísticas supra-segmentais) e um que leva as unidades elementares na realização acústica do sinal de voz, resolvemos o problema de síntese de voz a partir do texto. O primeiro mapeamento corresponde à etapa que é conhecida como *processamento lingüístico* enquanto o segundo corresponde à etapa de *processamento de sinais* [3].

Com relação à etapa de processamento de sinais, podemos destacar três

abordagens para resolver o problema de síntese arbitrária: em uma delas, tenta-se construir um modelo matemático razoável do aparelho fonador. Este modelo tem como entradas certos parâmetros articulatórios (como o recuo e a altura da língua, a abertura dos lábios etc) e a sua saída é o sinal de voz. Por isto, esta abordagem é conhecida como *síntese articulatória*. Entre as várias desvantagens desta abordagem, podemos mencionar, por exemplo, que o mapeamento direto (articulatório → acústico) tem alta complexidade computacional, o mapeamento inverso (acústico → articulatório) não é unívoco e não há uma disponibilidade de informações detalhadas a respeito da dinâmica dos articuladores [4].

Devido às dificuldades que surgem neste tipo de abordagem, outras abordagens foram desenvolvidas, nas quais o objetivo não é imitar o funcionamento do aparelho fonador, produzindo assim o sinal de voz, e sim imitar diretamente o sinal de voz a ser produzido, valendo-se de artifícios mais simples. Podemos destacar duas abordagens que seguem esta filosofia: a *síntese por regras* e a *síntese por concatenação*. Na síntese por regras, a idéia é utilizar um conjunto de regras para obter o sinal de voz associado ao texto de entrada. Este conjunto de regras fornecerá a evolução dos parâmetros de uma dada representação do sinal de voz, de acordo com a seqüência de fones relacionada ao texto de entrada. Por exemplo, as regras, em um sintetizador de formantes [5], são usadas para determinar as freqüências dos formantes, suas larguras de banda e os parâmetros do sinal de excitação.Como o sinal de voz é formado por trechos estáveis que são interligados por transições [5][1], as regras devem contemplar a evolução dos parâmetros tanto nos trechos estáveis quanto nos trechos correspondentes às transições. Não obstante, sabe-se que as transições entre fones são extremamentes importantes para a inteligibilidade da fala [6]. Devido a este fato, as regras que modelam as transições devem ser boas o suficiente para que as transições sejam razoavelmente precisas, de modo a não causar uma queda drástica da inteligibilidade. Assim sendo, a obtenção das regras não é uma tarefa trivial [1][3].

A síntese por concatenação resolve este problema através da justaposição de unidades cujas características são tais que preservam as transições entre fones. Nos sistemas de síntese por concatenação, existe um banco de unidades contendo todas as unidades necessárias, que são gravadas de antemão. Esta abordagem nos isenta

de obter as regras associadas às transições, porque que estas fazem parte do banco de unidades (estão contidas dentro das unidades) [3].

2.2.2 Tamanho das unidades

A partir do momento em que se opta pela abordagem de síntese por concatenação, torna-se imperativo decidir quais as unidades que farão parte do banco de unidades. A escolha das unidades é condicionada por vários fatores, como, por exemplo, limites impostos ao tamanho do banco de unidades (capacidade de armazenamento), a quantidade de fenômenos fonéticos que se quer abarcar e a facilidade de construção do banco (gravação e segmentação das unidades). Os primeiros sintetizadores de voz baseados na abordagem concatenativa tinham os fonemas como unidades básicas. O número médio de fonemas em uma língua é da ordem de 40, logo o banco de unidades ocuparia pouco espaço e seria de fácil gravação. Infelizmente, teríamos que gravar um grande número de variantes alofônicas para lidar com a coarticulação e com certos fenômenos supra-segmentais [1][2]. Também devemos notar que a forma como estas unidades são segmentadas força pontos de concatenação nas transições entre fones. Desta maneira, como as transições não são, em geral, corretamente realizadas, pois as regras para sua realização não estão embutidas no sistema, a inteligibilidade conseguida com o uso destas unidades é muito baixa [6]. Os problemas que ocorrem na concatenação de fonemas podem ser amenizados através da escolha de outro tipo de unidade que preserva a transição entre os fones. Uma das unidades mais simples e mais populares que possuem tais características são os difones, que começam na região estável de um fone e terminam na região estável do próximo fone, preservando desta forma a transição entre os dois [1][6]. Por outro lado, os trifones são unidades que contêm o trecho que vai da região estável de um fone (fone 1) até o começo do próximo fone (fone 2), a realização acústica completa do fone 2 e o trecho que vai do final do fone 2 até a região estável do fone que sucede o fone 2 (fone 3), preservando assim as transições entre os fones 1 e 2 e entre os fones 2 e 3 [7]. As unidades conhecidas como polifones englobam os difones, os trifones, e quaisquer unidades que começam na região estável do primeiro fone e terminam na região estável do último fone, contendo as realizações acústicas completas de dois ou mais fones intermediários [7]. Infelizmente, mesmo

que se mantenha as transições intactas, ainda continua existindo o problema da co-articulação que nos obriga a gravar um certo número de variantes alofônicas, onde o número de variantes depende da qualidade desejada. Além disso, considerando todas as possibilidades de combinação de um certo número de fonemas, o número de difones numa língua seria da ordem $40^2 = 1600$, o número de trifones seria da ordem de $40^3 = 64000$ e o de polifones 40^N, onde N é o número de fones da unidade. Na verdade, nem todas as possibilidades podem ocorrer em uma dada linguagem e podemos utilizar poucas variantes alofônicas para combinações raras de fonemas. Os termos difone (ou bifone) e trifone também se aplicam a um único fone inserido em um dado contexto fonético [8][9]. Os difones são fones com um contexto à direita ou à esquerda e os trifones são fones com um contexto à direita e outro à esquerda [9][10]. Estas unidades também podem ser utilizadas em sistemas de síntese de voz, pois as informações de contexto podem ser aproveitadas para restringir as variações espectrais entre os fones, fazendo com que as descontinuidades nas transições entre fones sejam pequenas e facilmente suavizáveis com o emprego de algoritmos de concatenação capazes de modificar a envoltória espectral do sinal de voz. Desta forma, sob um certo ponto de vista, as transições entre fones são preservadas. A vantagem das unidades maiores está calcada na presença de um número menor de pontos de concatenação. Cada ponto de concatenação corresponde à junção de um trecho de voz com outro trecho retirado possivelmente de um contexto fonético distinto, o que acarreta uma possível descontinuidade que diminui a inteligibilidade e a naturalidade. Portanto, quanto menor o número de pontos de concatenação, maior a qualidade do sistema. Em contrapartida, temos a desvantagem da maior quantidade de unidades a serem gravadas e do maior número de contextos a serem considerados, aumentando o tamanho do banco de unidades e dificultando a sua construção.

Pode-se concluir que ganha-se em flexibilidade optando-se pela utilização de unidades de tamanhos diferentes no banco de unidades, o que permite uma maior economia de espaço com o uso de unidades maiores apenas nas realizações altamente articuladas. O inconveniente gerado pela presença de unidades de tamanhos diferentes é o aumento da complexidade da determinação da seqüência de unidades durante a síntese.

Nos sistemas baseados em unidades de tamanho variável, as unidades po-

dem ser determinadas a priori, durante a construção do banco de unidades, ou a posteriori, durante a síntese. No último caso, é criado um grande banco de dados contendo trechos de fala natural devidamente segmentados e etiquetados. O sistema pode, por exemplo, procurar, durante a síntese, as maiores seqüências contíguas de fones que se encaixam na descrição fonológica, diminuindo o número de pontos de concatenação.

Outra escolha possível para o tamanho das unidades está ligada à idéia de usar unidades subfonêmicas, cujas definições não possuem um paralelo direto com as unidades fonológicas. A vantagens de tais unidades é a facilidade de construção do banco, pois elas têm definições acústicas simples e objetivas. Elas teriam como desvantagem a dificuldade em se encontrar um mapeamento que leva a seqüência de unidades fonológicas na seqüência de unidades acústicas.

As sílabas são unidades de difícil definição que são percebidas intuitivamente. Perceptualmente, as sílabas são compostas por *aclives*, *ápices* e *declives*. Os aclives e declives, também conhecidos como *ataque* e *coda*, respectivamente, determinam as fronteiras silábicas, e são trechos do sinal de voz de baixa intensidade, que podem conter consoantes. Os ápices correspondem ao núcleo da sílaba e são formados por sons vocálicos, que possuem alta intensidade, porque correspondem a uma fonação sem constrições na cavidade bucal. As sílabas que terminam com consoantes são denominadas "travadas" enquanto as outras são conhecidas como "abertas" [2]. As fronteiras de uma sílaba são o começo do primeiro fone do ataque e o final do último fone da coda silábica e estes pontos de segmentação podem estar na transição entre fones. Existem unidades conhecidas como semi-sílabas que são obtidas com uma segmentação que ocorre na região estável do núcleo vocálico, sendo então polifones. Embora o mesmo problema já apontado na discussão sobre fonemas, causado pela não preservação das transições, também ocorra aqui, sabe-se que a coarticulação intersilábica é menor que a coarticulação intrasilábica [1]. Por isso, em geral, as transições entre sílabas são menos importantes.

2.2.3 Naturalidade em sistemas TTS

Para estudar a linguagem é necessário compreender que a voz, apesar de ter uma aparência contínua, é dividida em unidades abstratas distintas pelos falantes

da língua e pelos lingüísticos. Essa divisão é baseada na percepção de diferenças de articulação, como o ponto de articulação (vogais, consoantes bilabiais, consoantes labiodentais etc) e modo de articulação (surda ou sonora), entre outros fatores, e até mesmo no esforço articulatório (para diferenciar sílabas átonas das tônicas). Estas diferenças levam à definição de unidades abstratas conhecidas como *fonemas*. São abstratas porque o mesmo fonema pode ter realizações acústicas diferentes, devido à coarticulação, que é o fenômeno no qual a realização de um fonema é dependente do contexto fonético vizinho. A coarticulação aumenta a inteligibilidade da voz, adicionando redundância ao enunciado. Todos os fenômenos fonéticos que ocorrem no nível destas unidades são chamados de *fenômenos segmentais*, e as realizações acústicas dos fonemas são chamadas de segmentos. A inteligibilidade de um sistema de conversão texto-fala depende fundamentalmente da qualidade segmental do sistema, ou seja, de quão perfeita é a produção de cada segmento do enunciado.

Para tal produção, é preciso descobrir o segmento correto a partir de uma transcrição fonética larga ou estreita do texto de entrada, e produzir o segmento através de regras, no caso da síntese por regras, ou seleção do segmento em inventórios de unidades, na síntese por concatenação. A produção do segmento deve levar em consideração o contexto fonético vizinho, ou seja, os fenômenos de coarticulação, que são importantes não só para conferir inteligibilidade, mas também para dar naturalidade à voz sintetizada. Existem outros fenômenos fonéticos cujos efeitos se estendem por vários segmentos, e por isso são conhecidos como *fenômenos supra-segmentais*. Alguns desses fenômenos estão relacionados com peculiaridades de um determinado processo articulatório que provocam um prolongamento temporal que, devido à dinâmica do aparelho fonador, engloba vários segmentos. Como exemplo, poderíamos citar a nasalização. Porém, outros fenômenos supra-segmentais têm uma natureza mais complexa: informações lingüísticas de alto nível são carreadas por uma modulação supra-segmental, que modifica os segmentos de modo a inserir tais informações. A variação da freqüência fundamental ao longo de um enunciado é um exemplo de um desses fenômenos. Alguns parâmetros do sinal de voz têm papéis importantes na modulação de informações lingüísticas. A freqüência fundamental da voz, a duração e a energia dos segmentos são alguns destes parâmetros. Um sistema de síntese de voz a partir de texto deve ser capaz de descobrir, a partir do

texto, qual a evolução temporal da freqüência fundamental do enunciado, e qual a duração e a energia dos segmentos. A inteligibilidade depende muito mais da qualidade segmental que da qualidade supra-segmental. Esta, porém, tem grande influência na naturalidade de um sistema TTS.

2.3 Sintetizadores para o PB

2.3.1 Aiuruetê

O Aiuruetê é um sistema TTS concatenativo para o português brasileiro. O nome do sistema, "Aiuruetê", vem do nome tupi-guarani do papagaio verdadeiro (*amazona aestiva*). O sistema foi desenvolvido através de um esforço conjunto de 5 anos, desde 1994, de lingüistas do LAFAPE (Laboratório de Fonética Acústica e Psicolingüística Experimental) e de engenheiros do LPDF (Laboratório de Processamento Digital da Fala), na UNICAMP [11].

O sistema se baseia na abordagem concatenativa, pois almejava-se alta naturalidade e baixo custo computacional, havendo uma alta restrição na disponibilidade de recursos devido ao caráter acadêmico do projeto. Os módulos e interfaces foram escritos em C++ e a interface entre o aplicativo e o usuário foi implementada em Delphi [11].

O módulo de transcrição grafema-fonema do Aiuruetê possui um pré-processador, responsável por converter as abreviaturas e acrônimos pelo texto equivalente por extenso, e o *Ortofon*, que converte a saída do pré-processador numa seqüência de letras que é uma representação fonética simplificada do texto [11]. Este módulo apresentou uma taxa de erros de 4%, que podem ser tratados por dicionários de exceções [11]. O sistema possui processamento lingüístico, sendo capaz de tratar alternâncias vocálicas semanticamente significantes, caracterizadas por menor ou maior abertura da vogal, através da identificação das classes gramaticais. Por exemplo, ele consegue diferenciar o verbo "secar" na terceira pessoa do singular do presente do indicativo (maior abertura vocálica) do adjetivo "seca" (menor abertura vocálica).

O inventório de unidades tem unidades básicas com tamanhos variando de semi-sílabas a seqüências de 5 segmentos, tendo um total de cerca de 2500 unidades. Estudos fonéticos da língua portuguesa indicam que os difones não são uma boa

escolha como unidades básicas. Como exemplo, podemos citar a nasalização de vogais, que é caracterizada no sistema através do arquifonema nasal /N/, que se segue à representação fonológica da vogal [12]. Alguns estudos indicam que a seqüência VN é foneticamente trifásica: começa numa fase oral contendo uma versão reduzida da vogal, acompanhada por uma fase contendo a vogal completamente nasalizada e terminando com um murmúrio nasal de baixa amplitude. Este e outros exemplos de fenômenos fonéticos sustentam a decomposição semi-silábica das seqüências CVC em uma unidade CV que preserva toda a transição mas possui duração muito curta e uma outra unidade VC contendo a rima da sílaba, sendo bem maior [11].

Um outro problema apontado pela análise fonética do fenômeno de nasalização é o fato de que a duração do murmúrio nasal pode variar de algo negligível a algo considerável, e esta variação pode ser atribuída a fatores de alto nível. Existem outros casos em que não é possível definir o alofone correto a partir de regras simples de contexto. A solução empregada no Aiuruetê foi utilizar o conceito de arquisegmentos, segmentos que não são completamente especificados, na conversão grafema-fonema. A conversão grafema-fonema é implementada em dois níveis: no nível de palavras, gerando uma representação fonológica lexical, e no nível de sentenças, gerando uma representação fonológica pós-lexical.

O conceito de arquisegmento também leva a um princípio simples para criar o inventório de unidades, que é a imposição de que os arquisegmentos não devem ser separados de segmentos adjacentes completamente especificados. Como este princípio levaria a um inventório contendo mais de 10000 unidades, é utilizado um princípio mais fraco no qual os arquisegmentos que constituem núcleo silábico podem ser separados [12].

As unidades foram obtidas embutindo as seqüências desejadas no centro de palavras sem sentido e as seqüências que ocorriam em fronteiras de palavras foram embutidas no interior de seqüências de duas palavras. Essas palavras ou seqüências de duas palavras não eram enunciadas isoladamente, mas no interior de sentenças portadoras [13].

Como o tamanho das unidades é variável, a determinação da seqüência apropriada de unidades a partir da seqüência de fonemas não é trivial. Para determinar a melhor seqüência, vale-se de um algoritmo que toma suas decisões aplicando um

conjunto de regras [13].

O Aiuruetê determina a duração de cada fonema a partir de um modelo de duração baseado no modelo de Campbell[14]. Tal modelo usa unidades de tamanho silábico como unidades básicas de programação rítmica. Uma rede neuronal obtém a duração dessas unidades a partir de uma representação fonológica do texto. Em seguida, a duração de cada fonema é gerada através de um modelo de repartição que segue um princípio de elasticidade forte [15]. Para o contorno de F0, usa-se uma simples declinação declarativa [11].

2.3.2 DeltaTalk

O DeltaTalk é um sistema TTS desenvolvido pela MicroPower Software, uma empresa localizada em São Paulo, que fornece produtos e soluções na área de síntese de voz, tradução, *e-learning* e desenvolvimento Web [16]. Existem duas versões do DeltaTalk, a 1.0 e a 2.0, que custavam respectivamente, R$ 14,90 e R$ 44,00, em 2002. O DeltaTalk 1.0 pode ser obtido na versão *shareware* através pelo site da MicroPower.

A tabela 2.1, com informações obtidas no site da MicroPower Software, ilustra algumas diferenças entre as versões 1.0 e 2.0 do DeltaTalk. Os requisitos de *hardware* mínimos da versão 2.0 são um Pentium 100 com 16 Mb de memória e 30 Mb de espaço livre em disco. As plataformas suportadas são as versões 9X e NT 4.0 (com o Service Pack 4 ou superior) do Microsoft Windows. O DeltaTalk 2.0 pode interfacear com o Virtual Vision 2.0 da MicroPower Software, permitindo que deficientes visuais usem o computador.

O Virtual Vision pode ler textos digitados pelo usuário, em documentos ou páginas da Web, inclusive informando os links disponíveis na página e o tipo e tamanho das fontes usadas. Pode informar os objetos que estão sendo apontados pelo mouse, ler mensagens enviadas por aplicativos, e fornecer detalhes sobre o status dos controles do windows.

O DeltaTalk faz a conversão texto-fala através da concatenação de difones. O sistema faz normalização de texto e possui um dicionário de exceções no qual novas pronúncias podem ser adicionadas pelo usuário. O sistema possui modelamento prosódico, o que significa que deve ser feito algum tipo de processamento

lingüístico de alto nível. (verificar se existe tratamento da alternância vocálica). É possível controlar a velocidade de pronúncia e a tonalidade da voz, bem como editar o contorno de F0 mediante um editor de melodias.

Os inventórios de difones das três vozes do DeltaTalk 2.0 podem ser obtidos através do site do projeto MBROLA [17]. O sistema utiliza o mesmo tipo de técnica de síntese do MBROLA, o algoritmo MBR-PSOLA [3].

Tabela 2.1: Principais diferenças entre as versões 1.0 e 2.0 do DeltaTalk.

Benefícios / Características	DeltaTalk 1.0	DeltaTalk 2.0
Tipos de vozes	1	3
Voz Feminina	não tem	1
Voz Masculina	1	2
SAPI	não	sim
Preparado para deficientes visuais	não	sim
Otimizado para Pentium III	não	sim

Uma das inovações da versão 2.0 do DeltaTalk em relação à versão 1.0 foi a adoção do padrão Microsoft SAPI (*Speech Application Program Interface*). Com a adoção deste padrão, o motor de conversão texto-fala pode ser compartilhado por vários aplicativos compatíveis com a interface, e os aplicativos podem utilizar quaisquer motores que tenham sido instalados. Isto permite uma ampla integração entre aplicativos.

Existem várias ferramentas para facillitar a programação de aplicativos capazes de ter acesso a motores de síntese. Algumas destas ferramentas consistem em componentes, que podem ser adicionados ao programa, que facilitam o uso de recursos da Microsoft SAPI ou da tecnologia Microsoft Agent.

2.3.3 Difones

Este sistema é descrito em [18]. Foi desenvolvido pelo aluno Alex Ribeiro Franco, na UFRJ, sob a orientação do Prof. Márcio Nogueira de Souza. Trata-se de

um sistema TTS concatenativo que usa a técnica LPC e baseou-se em um sistema anterior que concatenava os difones diretamente, no domínio do tempo.

A conversão grafema-fonema é feita com um conjunto de regras de contexto para cada grafema. A partir da seqüência de fonemas pode-se obter a seqüência de difones a serem concatenados. Os valores dos coeficientes LPC não são interpolados nas fronteiras dos segmentos, mas mesmo assim o sistema apresentou resultados melhores em relação ao sistema anterior baseado na concatenação direta dos segmentos.

O sintetizador não possui nenhum processamento lingüístico de alto nível e portanto nenhum modelamento prosódico. Os enunciados são sintetizados com um contorno de *pitch* uniforme e a duração dos segmentos não é alterada. A velocidade da fala pode ser controlada alterando a freqüência de reprodução do sinal, mas modificando também o *pitch*.

A Tabela 2.2 apresenta os resultados obtidos na avaliação da inteligibilidade de palavras e frases deste sintetizador e do sintetizador anterior por concatenação direta. Os resultados mostram que ocorre um aumento drástico da inteligibilidade das frases que não ocorria no sintetizador original.

Tabela 2.2: Comparação entre o sintetizador com difones usando LPC e concatenação direta.

Sintetizador	Palavras (%)	Frases (%)
Difones	67,00	92,00
Anterior	75,75	66,12

2.3.4 Digalo

Digalo [19] é uma divisão da Elan Informatique, uma empresa localizada em Toulouse, França, que é especializada no suprimento de softwares para conversão texto-fala. As empresas costumavam fornecer softwares de conversão texto-fala para várias línguas através de licenças OEM, a preços que não eram razoáveis para o consumidor final. A missão da Digalo é fornecer motores de síntese de voz a preços acessíveis para os desenvolvedores e consumidores finais.

O Digalo não é um software de síntese embutido em um aplicativo, mas um *plug-in* que contém o motor de síntese. Qualquer aplicativo pode ter acesso ao motor

através da interface Microsoft SAPI 4.0 ou da tecnologia Microsoft Agent. Neste último caso, o aplicativo poderá comandar uma personagem animada interativa, parecida com os assistentes do Microsoft Office.

Os motores da Digalo estão disponíveis em 7 línguas, entre elas o português brasileiro. A única plataforma suportada é o Microsoft Windows (9X, 2000 e NT), e o hardware mínimo é um PC com um processador Pentium 133 MHz ou equivalente, com 32 Mb de RAM e de 3 a 10 Mb de espaço livre em disco dependendo da língua do motor de síntese.

No site da Digalo existem links para vários aplicativos (colocar link direto) que podem ser usados com os motores disponibilizados. Dentre os aplicativos, podemos encontrar, por exemplo:

- Relógios falantes;
- Leitores da área de transferência, e-mails e outros documentos;
- Aplicativos que se integram a browsers de web (Internet Explorer, Netscape etc) e *instant messengers* (ICQ, MSN Messenger etc);
- Aplicativos de automação para controlar eletrodomésticos;
- Aplicativos de telefonia para leitura de e-mails pelo telefone, para dizer o nome e o número do assinador chamador etc.

Os motores de síntese possuem dicionários de exceções e abreviaturas que permitem corrigir transcrições fonéticas incorretas de certas palavras. Pode-se obter versões de demonstração dos motores de síntese através do site da Digalo, funcionais por um período de 15 dias. O preço dos motores é de $ 29.00 por língua.

2.3.5 DOSVOX

O DOSVOX [20] é um sistema que permite o uso de computadores por deficientes visuais através da tecnologia de síntese de voz. Foi desenvolvido no NCE (Núcleo de Computação Eletrônica), na UFRJ, sob a supervisão do Prof. José Antônio dos Santos Borges. O DOSVOX possui um sistema operacional que contém os elementos da interface com o usuário, um sistema de síntese de voz e mais de 40

programas. Alguns desses programas são: teste de teclado para decorar a posição das teclas, editor e leitor de textos, caderno de telefones, agenda de compromissos, programas educativos, programas para acesso à Internet etc. Atualmente, mais de 2000 cegos que fazem uso do sistema em todo Brasil. Os responsáveis pelo projeto são:

- Gerência: José Antônio dos Santos Borges
- Programação: Marcelo Pimentel
- Apoio aos Usuários: Renato Costa e Bernard Condorcet
- Centro de Distribuição do DOSVOX: Katia Oliveira
- Projeto Inter-Vox (cegos na Internet): Maria Irene Sá

O DOSVOX foi inicialmente criado para permitir que Marcelo Pimentel, que é deficiente visual, um dos alunos do curso de Computação Gráfica do Prof. José Antônio dos Santos Borges, pudesse usar o computador de forma independente. Antônio fez o sistema de síntese de voz e Marcelo programou a interface com o usuário. Para que Marcelo editasse arquivos, foi desenvolvido um editor simples, inspirado no EDIT do DOS, que foi chamado de EDIVOX. Os dois notaram que o sistema podia ser utilizado por outros cegos, e organizaram um curso na UFRJ, sobre o DOSVOX, para deficientes visuais. O curso permitiu melhorar a interface do sistema graças às sugestões dos usuários. Com o aumento do número de usuários, a administração do projeto se tornou difícil e decidiu-se pela criação de uma micro-empresa que se responsabilizaria pela distribuição do produto, com uma pequena margem de lucro que seria revertida para o desenvolvimento do sistema.

A saída de voz do DOSVOX pode ser através da reprodução de frases ou palavras previamente gravadas ou por meio da concatenação de unidades básicas, no caso de texto irrestrito. Atualmente o DOSVOX possui um servidor de fala que permite que seus programas tenham acesso a motores de síntese instalados no computador e que utilizam a interface Microsoft SAPI. A descrição a seguir se refere ao sistema TTS disponível na distribuição básica.

No caso de texto irrestrito, o sistema realiza uma conversão grafema-fonema e depois concatena as unidades selecionadas a partir da seqüência de fonemas, para

obter o sinal de voz. A concatenação das unidades é direta. O sistema não possui processamento lingüístico nem modelamento prosódico.

A conversão grafema-fonema é realizada por meio de uma espécie de máquina de estados finitos, que equivale a um conjunto de regras de contexto. A representação fonética na saída especifica também se o fonema pertence a uma sílaba tônica. As palavras que não são adequadamente convertidas são tratadas com um dicionário de exceções.

Para evitar a completa perda de inteligibilidade de alguns fonemas, que ocorreria no caso em que fonemas gravados isoladamente ou em contexto fonético diverso fossem concatenados, o sistema possui cerca de 350 unidades, do tipo CV, encontros vocálicos e outras unidades associadas a junções críticas.

2.3.6 Elan TTS

A Elan Informatique [21] é uma empresa especializada em sistemas TTS, fornecendo soluções de síntese de voz para várias aplicações. O motor de síntese para o português brasileiro foi desenvolvido com a colaboração da Fundação CPqD, e é o mesmo motor usado no Digalo. Alguns dos produtos da Elan são:

- Telecomunicações - NETworks CUBE: um sistema operacional para a construção de *clusters* de servidores TTS escravos controlados por um mestre, capaz de controlar 100 servidores. Cada servidor pode fazer 200 conversões texto-fala simultaneamente, o que possibilita a construção de um sistema capaz de realizar 20000 tarefas de conversão texto-fala. As plataformas suportadas são Microsoft Windows, Solaris e Linux. Speech CUBE: um componente de software para arquiteturas cliente-servidor. Um único servidor é capaz de realizar 200 conversões texto-fala simultâneas em um Pentium III 900 MHz. Estes sistemas podem ser utilizados em portais de voz, acesso a bases de dados, leitura de e-mails etc, são disponíveis em 11 línguas, incluindo o português brasileiro, e 20 vozes. Speech UNIT: um sistema completo (hardware + software) para aplicações de segurança, monitoramento remoto, servidores de voz de baixa capacidade etc. Dial & Play: software para aplicações de voice e-mail tendo 7 línguas disponíveis, com reconhecimento automático;

- Multimedia - Speech ENGINE: motor de síntese para aplicações multimídia, disponível em 10 línguas, incluindo o PB. O Speech ENGINE é distribuído com exemplos de código, DLLs e componentes ActiveX para facilitar a programação de aplicativos;

- Assistentes pessoais - Pocket Speech: kit de desenvolvimento de aplicações de síntese de voz para PDAs, disponível em 9 línguas e 17 vozes, incluindo o português brasileiro;

- *Embedded speech* - Implementações específicas para embutir tecnologia de síntese de voz em sistemas eletrônicos (automóveis, eletrodomésticos etc);

- *e-language family* - Algumas ferramentas que podem ser utilizadas para melhorar a qualidade da voz sintetizada dos produtos distribuídos pela Elan. Lexitool: ferramenta para editar os dicionários de exceções e de abreviaturas. Prosel: ferramenta para copiar a prosódia de enunciados naturais para enunciados sintéticos.

2.3.7 Lernout & Hauspie

Os recursos de fala e linguagem da Lernout & Hauspie foram adquiridos pela ScanSoft inc. [22], que fornece um sistema TTS chamado RealSpeak. Este sistema possui motores em 19 línguas, incluindo o português brasileiro, e as plataformas suportadas são: Linux, Solaris, AIX e várias versões do Microsoft Windows. Este sistema possui processamento lingüístico e modelamento prosódico, e é capaz de tratar alternâncias vocálicas. A interface dos motores com os aplicativos é feita através do padrão Microsoft SAPI.

2.3.8 MBROLA

MBROLA [17] é um projeto do laboratório TCTS da Faculté Polytechnique de Mons, na Bélgica, que tem como objetivo obter um conjunto de sintetizadores de voz para o maior número possível de línguas e disponibilizá-los gratuitamente para aplicações não comerciais. Espera-se com isso aumentar a atividade de pesquisa acadêmica em síntese de voz, principalmente na geração de prosódia. Atualmente o

sistema disponibiliza bases de difones para 26 línguas, incluindo o português brasileiro.

O sintetizador MBROLA gera voz sintética por concatenação de difones com o algoritmo MBR-PSOLA [3]. Tem como entrada a seqüência de fonemas e informação prosódica na forma de duração dos fonemas e uma descrição linear em partes do contorno de F0 [17]. Não se trata de um sistema TTS propriamente dito, pois a entrada do sistema não é texto.

Neste algoritmo, as unidades são processadas *offline*, usando a técnica *Multiband Resynthesis* [23], para terem mesmo *pitch* e mesma fase. A fase constante resolve o problema de possíveis descontinuidades de fase causadas pelo posicionamento inconsistente das marcas de *pitch* no PSOLA. Pode-se demonstrar que, se o *pitch* e a fase são os mesmos em todas as unidades, uma interpolação dos sinais no domínio do tempo corresponde a uma interpolação no domínio da freqüência [3]. Com isso, é possível suavizar a transição espectral nas fronteiras das unidades, trabalhando com os sinais diretamente no domínio do tempo. Porém, existe uma perda de naturalidade causada pelo processo de ressíntese [24].

2.3.9 Sílabas

Este sistema TTS foi desenvolvido por André Cardon, na UFRGS, sob orientação do Prof. Philippe O. A. Navaux [25]. Este sistema se baseia na concatenação direta de sílabas. Não possui processamento lingüístico, nem modelamento prosódico.

As sílabas foram gravadas isoladamente, não tendo sido extraídas de palavras nas quais estavam embutidas. Para escolher as unidades a serem concatenadas, o sistema realiza uma conversão grafema-fonema, através de um conjunto de regras de contexto. Isso é necessário, por exemplo, na diferenciação de palavras como “caça” e “casa”, escolhendo as unidades “sa” ([sa]) e “za” ([za]), respectivamente.

Realizou-se um teste de inteligibilidade com 5 ouvintes, aos quais foram apresentadas 50 palavras sintetizadas. A tarefa dos ouvintes era indicar se cada uma das palavras foi ou não compreendida. A inteligibilidade do sistema foi de 90% Não obstante, a pronúncia de uma sílaba isolada é planejada pelo locutor de tal forma a ser compreensível para os ouvintes, mesmo fora de contexto. O mesmo não ocorre

dentro de uma palavra, quando a realização acústica da sílaba é deformada pela sobreposição de gestos articulatórios, perdendo a inteligibilidade fora deste contexto. Deste modo, a falta de coarticulação intersilábica causa uma perda de naturalidade em um sistema baseado em sílabas enunciadas isoladamente.

2.3.10 TalkActive

Este sistema será descrito no Capítulo 4.

2.3.11 TextoFala

O Texto-Fala [26] é um sistema TTS desenvolvido pela Fundação CPqD, em colaboração com a Elan Informatique. O sistema é concatenativo e usa polifones como unidades, bem como a técnica PSOLA para modificar os segmentos [27].

O Texto-Fala e os motores de síntese da Elan são os melhores sistemas TTS entre os sistemas aqui descritos, possuindo boa qualidade segmental e melodia e rítmo razoáveis.

O sistema possui processamento lingüístico e modelamento prosódico. O modelo de entonação consiste em duas etapas: a primeira mapeia o texto numa representação fonológica [28] e a última mapeia esta representação no contorno de F0.

O primeiro mapeamento consiste em um *parsing* que encontra as fronteiras dos constituintes prosódicos e os níveis de acento de cada palavra. São usadas 20 marcações, reunídas em 4 grupos, para especificar o acento da palavra [29]. O segundo mapeamento utiliza estas marcações para consultar um dicionário de contornos de F0, escolhendo a entrada mais similar. Os contornos obtidos do dicionário são ajustados e concatenados para produzir o contorno de F0 final [30].

2.4 Resumo das principais características

A Tabela 2.3 possibilita uma comparação fácil e rápida das principais características dos vários sintetizadores descritos na Seção 2.3.

Tabela 2.3: Comparacção entre os sintetizadores.

Características	Aiuruetê	DeltaTalk	Difones[1]	Digalo	DOSVOX	L&H	Sílabas[2]	TalkActive	TextoFala
Origem	LAFAPE UNICAMP	MicroPower Software	DEL UFRJ	Elan Informatique	NCE UFRJ	Lernout & Hauspie	PPGC UFRGS	DEL UFRJ	Fundação CPqD
Versão		2.0							
Comercial	não	sim	não	sim	não	sim	não	não	sim
Custo		R$ 44,00		$ 29,00					
Normalização de texto	sim	sim	não	sim	não	sim	sim	não	sim
Leitura de e-mails		sim	não	sim	sim		não	não	
Processamento Lingüístico	sim	sim	não	sim	não	sim	não	não	sim
Prosódia	sim	sim	não	sim	não	sim	não	não	sim
Unidade	polifones	difones	difones	polifones	CV e outras		sílabas	sílabas	polifones
Técnica de síntese	TD-PSOLA HNM	MBR- -PSOLA	LPC	PSOLA	Direta		Direta	TD- -PSOLA	PSOLA
SAPI	não	sim	não	sim	não	sim	não	não	
Integração com aplicativos	sim	não	sim	sim	sim	não	não	não	
Referências									

1 - Refere-se ao sintetizador baseado em difones descrito em [18].

2 - Refere-se ao sintetizador baseado em sílabas descrito em [25].

2.5 Conclusão

Neste capítulo são introduzidos alguns conceitos associados ao problema de síntese de voz a partir de texto. São abordadas as diferentes maneiras de resolver este problema, a escolha do tamanho das unidades e os fatores que interferem na naturalidade da voz sintetizada.

O capítulo também descreve vários sistemas TTS para a língua portuguesa falada no Brasil para demonstrar o atual nível de desenvolvimento da tecnologia de síntese para o português brasileiro. Será descrito em detalhes, no Capítulo 4, o sistema TalkActive, em sua versão 3.0. É um sistema baseado em sílabas, no qual será implementado um módulo de prosódia simplificado.

Os sistemas da Fundação CPqD e da Elan Informatique, que compartilham a mesma tecnologia, possuem uma qualidade segmental excepcional, sendo, entre todos os sistemas analisados, os que produzem voz sintética de mais alta qualidade. Mas a voz produzida ainda pode ser considerada enfadonha e artificial, devido à simplicidade do módulo de prosódia. Daí a importância do estudo da geração de prosódia em síntese de voz.

Capítulo 3

Modelos de Prosódia

3.1 Introdução

Este capítulo contém um estudo básico sobre prosódia e mostra a forma como o processamento prosódico pode ser incorporado a um sistema TTS. O conceito de prosódia é muito importante em lingüística. Graças à prosódia, certas informações lingüísticas de alto nível são embutidas na voz, no processo de codifição inconsciente realizado pelo locutor.

A ausência de parte dessa informação lingüística de alto nível dá, à voz sintetizada, um caráter artificial. Por isso é importante uma modelagem apropriada do processo de codificação dessas informações lingüísticas. Para tal é necessário entender o que é prosódia e visualizar as formas como a codificação pode ser implementada em um sistema TTS.

Neste capítulo será apresentado, na Seção 3.2, o conceito de prosódia. Nessa seção serão mostradas as dificuldades enfrentadas para obter uma definição de prosódia que seja adequada para o projeto de sistemas TTS.

Na Seção 3.3 são indicadas as dificuldades para construir modelos de prosódia, como, por exemplo, a dificuldade de fazer analises sintáticas e semânticas de um texto e o fato de que características consideradas segmentais podem carrear informações lingüísticas de alto nível, o que exigiria, a princípio, uma modificação específica das unidades do inventório.

Já na Seção 3.4, as funções elementares da prosódia, coletivamente chamadas de *prosódia de alto nível*, serão conceituadas. Pesquisas em lingüística parecem

mostrar que todo o processo de codificação prosódica realizado pelos locutores pode ser compreendido como sendo formado por apenas duas funções elementares: uma que é responsável pela segmentação de um enunciado em seus constituintes e outra responsável por atribuir diferentes ênfases a diferentes elementos do enunciado.

A prosódia de alto nível se expressa através de variações do contorno de F0 ao longo do tempo, da duração dos segmentos, da energia ao longo do enunciado etc. Estes parâmetros acústicos carregam, na sua evolução temporal, informações lingüísticas de alto nível.

Um modelo de prosódia para um sistema TTS deve, assim, ser capaz de prever a evolução temporal destes parâmetros. Por isso, as Seções 3.5 e 3.6 descrevem modelos de duração e entonação, respectivamente, sendo dada grande ênfase na descrição do modelo de Fujisaki, na Seção 3.6.3.

3.2 Prosódia

A literatura apresenta várias definições de prosódia, como por exemplo [31]:

"**pro.só.dia sf (lat prosodía)**

1 **Gram** Pronúncia correta das palavras, de acordo com a acentuação; acentuação tônica. 2 Parte da Gramática que se ocupa da pronúncia das palavras. 3 **Mús** Adequada ligação das palavras com os acentos melódicos, de modo que as sílabas longas e breves conservem a acentuação que lhes é própria."

E também [32]:

"Prosódia é a parte da fonética que trata da correta acentuação e entonação dos fonemas."

Estas definições, assim como outras, não nos são muito úteis na prática, por serem ora muito específicas, ora muito genéricas.

Em textos sobre processamento de voz, o conceito de prosódia costuma ser apresentado através da enumeração de suas funções na comunicação verbal. Algumas das funções da prosódia são: deixar clara a intenção do locutor (e.g., se ele está afirmando ou perguntando algo), esclarecer a relação sintática entre os elementos frasais, restringir o significado das palavras ou expressões do enunciado, informar

sobre o estado emotivo do locutor etc.

Este tipo de definição apresenta dois grandes inconvenientes: é necessário enumerar exaustivamente todas as funções da prosódia na comunicação verbal para que a definição seja completa, e não nos revela nenhuma pista de como fazer o processamento da prosódia.

De modo geral, a função da prosódia é aumentar a inteligibilidade da mensagem a ser comunicada, pois possui uma função de integração: embute informações lingüísticas de alto nível (morfologia, sintaxe, semântica, pragmática e discurso) e paralingüísticas no sinal de voz [33]. Algumas informações que podem ser embutidas são os significados das palavras ou frases (informação semântica inferida pelo ouvinte através da ênfase dada a palavras ou grupos de palavras, ou pela segmentação do enunciado), a modalidade da frase (informação pragmática que nos diz se a frase é declarativa ou interrogativa, por exemplo), os sentimentos e a personalidade do locutor (informações paralingüísticas) etc [34].

Um dos motivos da baixa naturalidade dos atuais sistemas de conversão texto-fala é o fato de que os modelos de prosódia ainda não são capazes de promover esta integração eficientemente, de forma que tais informações lingüísticas e paralingüísticas não estão prontamente disponíveis ao ouvinte [35].

Certos autores definem prosódia como o conjunto de atributos da voz que não estão associados a segmentos individuais, mas sim a unidades maiores como sílabas, palavras, sentenças e parágrafos [36]. Em outras palavras, os fenômenos prosódicos são confundidos com os fenômenos suprasegmentais, ou seja, com os fenômenos fonológicos que abarcam vários segmentos. No entanto, existem inconvenientes nesta definição, já que certos fenômenos suprasegmentais não são comumente classificados como prosódicos [37], como, por exemplo, a nasalização e a labialização, embora algumas escolas lingüísticas os classifiquem como tais [2].

Sabe-se que a integração supracitada de informações lingüísticas é realizada mediante conveniente modulação dos parâmetros acústicos. Assim, alguns autores definem prosódia como sendo a combinação das variações de certos parâmetros do sinal de voz (a freqüência fundamental, a duração dos segmentos e a energia) durante a fala [38]. Esta última definição está de acordo com a definição de prosódia como sendo o estudo dos elementos que podem ser descritos pela ação dos músculos

respiratórios que aumentam ou diminuem a energia do fluxo de ar, modificando o comprimento, o *pitch* e a intensidade dos diversos trechos do sinal de voz [2]. Esta definição, entretanto, não considera outros parâmetros do sinal de voz, como a distribuição de energia no espectro de freqüências, por exemplo, que podem contribuir na tarefa de integrar informações lingüísticas de alto nível no sinal de voz e que são tipicamente considerados como características segmentais [39].

Os componentes acústicos da prosódia são conhecidos como *prosódia de baixo nível* [40] e a duração dos segmentos, a freqüência fundamental e a energia, seus principais componentes acústicos, são chamados *parâmetros prosódicos* [41]. Devemos fazer uma distinção entre os parâmetros encarados do ponto de vista da produção e do ponto de vista da percepção. Podemos citar como exemplo o volume de um segmento, que não é exatamente igual à energia, pois depende também da variação da freqüência fundamental no interior do segmento e de sua duração [3][15].

A prosódia ajuda a compreender a mensagem porque a variação dos parâmetros prosódicos confere uma estruturação ao enunciado que depende de fatores lingüísticos de alto nível [7][40]. Por esta razão, as informações lingüísticas são codificadas nas variações dos parâmetros prosódicos. Tal estruturação é construída através de duas funções: uma função de segmentação dos enunciados, e uma função que promove ou rebaixa palavras ou grupos de palavras, estabelecendo entre elas uma oposição do tipo fraco/forte [40][36]. Estas funções, conhecidas como *prosódia de alto nível* [40], serão discutidas na Seção 3.4.

Neste texto, o termo prosódia será definido como o conjunto dos atributos do sinal de voz que promovem a integração de certas informações lingüísticas e paralingüísticas, ajudando o locutor a melhor comunicar a mensagem.

3.3 Modelagem da prosódia

A palavra "modelo" neste texto assume um significado amplo. Aqui, "modelo de prosódia" significa qualquer forma de descrição do mapeamento que leva o conjunto de informações lingüísticas e paralingüísticas a serem transmitidas na comunicação, na evolução dos parâmetros prosódicos.

Um dos problemas com relação à modelagem da prosódia é o fato de que

outras características acústicas além dos parâmetros prosódicos podem influenciar no processo de integração de informações lingüísticas de alto nível. Por exemplo, tais informações podem ser codificadas através da duração do murmúrio nasal nas vogais nasais [39]. Isso leva à questão da possibilidade de processar em paralelo a prosódia e os segmentos [42]. Alguns sistemas de síntese baseados em seleção de unidades em grandes bases de dados utilizam anotações prosódicas para obter a seqüência de unidades a serem concatenadas, reduzindo com isto a necessidade de modificações prosódicas [43]. Tais sistemas fazem uma espécie de processamento conjunto da prosódia e dos segmentos.

A tendência atual da pesquisa em prosódia é a adoção de modelos estatísticos derivados de *corpora* de fala [39][35]. Um dos problemas dos modelos de prosódia que não são baseados nesta abordagem (*top-down*) e sim em regras (*bottom-up*) é que a inadequação de alguma regra pode causar uma performance muito ruim em alguns casos, enquanto que nos modelos estatísticos tenta-se extrair um comportamento médio, ajudando a evitar erros grosseiros [44].

Os modelos estatísticos devem ser obtidos de um *corpus* de fala. Porém, o locutor se adapta, modificando sua voz de acordo com o ambiente, de forma a otimizar a comunicação da mensagem [42]. Assim, torna-se necessário, a fim de obter alta naturalidade, que o *corpus* do qual são obtidos os modelos de prosódia seja criado para o mesmo domínio de aplicação (estilo de fala, relação sinal/ruído do ambiente etc) da voz a ser sintetizada. Os modelos obtidos não garantirão naturalidade fora do domínio de aplicação.

Outro problema da modelagem prosódica em sistemas de conversão texto-fala é a dificuldade em fazer análises lingüísticas profundas a partir do texto [42]. Mas a modelagem e geração de prosódia neutra (sem emoção) pode ser feita, geralmente com pouca perda de naturalidade, valendo-se de uma análise sintática superficial [3].

Neste trabalho são feitas as seguintes considerações:

1. A prosódia e os segmentos podem ser processados separadamente sem degradações perceptíveis da naturalidade;

2. Somente a modificação dos parâmetros prosódicos já é suficiente para garantir boa naturalidade;

3. Serão apenas considerados os principais correlatos acústicos da prosódia, ou seja, os parâmetros prosódicos encarados sob o ponto de vista da produção: freqüência fundamental, duração observada e energia. A obtenção destes parâmetros a partir do sinal de voz é mais direta, em contraste com os principais correlatos perceptuais, ou seja, os parâmetros prosódicos encarados do ponto de vista da percepção: altura, duração percebida e volume. É mister utilizar um modelo perceptual para obter estes últimos;

4. Os parâmetros prosódicos possuem ordens diferentes de importância perceptual. Entende-se aqui que a importância maior está associada à maior correlação entre a variação de determinado parâmetro prosódico e a percepção de certas características. As variações de F0 são os melhores correlatos para indicar prosódia não neutra; por outro lado, o contorno de F0 junto com a duração dos segmentos são os melhores correlatos para detectar prominência relacionada com o discurso (foco) [45]. Considerar-se-á que variações da energia não são muito importantes. Logo, não serão analisados modelos para descrever a variação de energia.

3.4 Prosódia de alto nível

Os estudos de prosódia supõe que o sinal de voz tem uma certa estrutura hierárquica composta de *domínios prosódicos* [45]. Investigações para descobrir as regras associadas à melodia, ao ritmo da fala e a certos fenômenos fonológicos como *sandhi* externo confirmam a validade desta suposição [45]. Informações lingüísticas e paralingüísticas são embutidas no sinal de voz graças a esta estruturação, mediante duas funções que segmentam a voz e enfatizam ou não palavras ou grupos de palavras.

A função responsável pela segmentação gera segmentos contíguos que são conhecidos como *constituintes prosódicos*. Encontrar as fronteiras entre constituintes prosódicos é um dos problemas que devem ser resolvidos para dar naturalidade ao texto. Admite-se que, em geral, uma análise sintática superficial é suficiente para localizar estas fronteiras no caso de prosódia neutra [3], mas outras informações lingüísticas de alto nível, como, por exemplo, semântica e discurso, podem ser im-

portantes para a localização da posição das fronteiras. Podemos admitir também a existência de vários tipos de fronteiras entre constituintes prosódicos, e o tipo de fronteira seria determinado pelas informações lingüísticas de alto nível [3].

A função que estabelece entre as palavras uma relação do tipo fraco/forte é a *acentuação*. Toda palavra polissilábica da língua portuguesa, pronunciada isoladamente, possui uma sílaba que é pronunciada com maior "esforço" que as outras e que é conhecida como *sílaba tônica*. A ênfase adicional empregada na pronúncia de uma sílaba é chamada *acento*. A posição da sílaba tônica pode ser importante para distinguir vocábulos, como, por exemplo, nas palavras "sabia", "sábia" e "sabiá". Para determinar qual das sílabas é a tônica basta fazer uma busca em um dicionário, e por isso este tipo de acento é chamado de *acento léxico*. Em português, o acento é realizado através de variações de *pitch*, volume e duração, devendo receber a denominação de *acento de intensidade* (*stress accent*) em oposição ao *acento tônico* (*pitch accent*), que é realizado principalmente com variações do *pitch*.

Além do acento léxico, podemos identificar a existência do *acento frasal*, que ocorre quando as palavras são pronunciadas no contexto de uma frase. A sílaba que recebe o acento frasal nem sempre é a mesma que recebe o acento léxico.

Dentro de uma frase, algumas palavras do constituinte prosódico são mais importantes que outras. Por exemplo, verbos e substantivos geralmente são mais importantes que preposições e conjunções. As palavras que têm significado fora do contexto frasal (e.g., os substantivos) são conhecidas como *palavras de conteúdo* e são pronunciadas com mais ênfase que as palavras que não posuem significado fora da frase e que são conhecidas como *palavras de função* (e.g., as preposições). Os diferentes graus de importância das palavras em uma frase são os responsáveis pela diferença de ênfase na pronúncia, provocando o aparecimento de diferentes graus de acentuação.

A Figura 3.1 ilustra a posição da análise da prosódia de alto nível na cadeia de processamento prosódico. A análise lingüística de alto nível obtém, a partir do texto, as informações discursivas, pragmáticas, semânticas e sintáticas necessárias para determinar os acentos e as fronteiras de constituintes. O resultado da análise prosódica pode consistir, por exemplo, na seqüência de fonemas com marcações das fronteiras das sílabas e da indicação das fronteiras entre constituintes e dos níveis de

acento de cada sílaba. Esta saída é utilizada pelos modelos de duração e entonação para obter a duração de cada segmento e o contorno de F0, respectivamente.

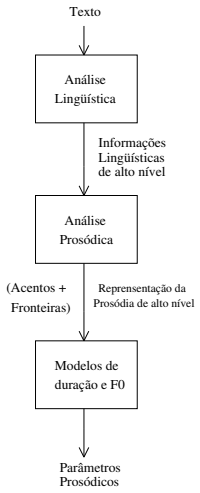


Figura 3.1: Diagrama ilustrando o processamento prosódico.

3.5 Modelos de duração

Os modelos de duração são responsáveis pela tarefa de mapear as informações lingüísticas nas durações de cada segmento. Um dos problemas na obtenção da duração dos segmentos é o fato de que não é possível definir exatamente onde começa e termina um determinado segmento, devido à coarticulação. Não obstante, a posição da fronteira entre dois fones pode ser determinada de forma bastante consistente. A maior dificuldade está em fazer a marcação das fronteiras automaticamente, se quisermos obter modelos estatísticos através de uma grande base de dados. A marcação das fronteiras deve ser feita manualmente, a não ser que se disponha de um sistema de reconhecimento de voz capaz de realizar uma segmentação automática.

Como a duração depende de um grande número de fatores, isto pode criar uma situação de esparsividade de dados, mesmo que a base de dados seja grande [46], já que é muito provável ocorrer em um enunciado uma combinação rara de fatores. Como diversos fatores interagem entre si, regiões do espaço de descrição do segmento podem conter poucos ou nenhum ponto [46]. Desta forma, a duração prevista através de modelos baseados em um dicionário de duração ou de modelos obtidos por *clustering* não hierárquico se torna imprecisa e inacurada, ou o modelo

pode ser incapaz de prever a duração. Assim, é imperativa a utilização de um modelo capaz de generalização, que seja capaz de interpolação no espaço de descrição do segmento, aproveitando-se da propriedade de *invariância direcional* dos fatores que influenciam na duração.

A invariância direcional é uma propriedade da duração dos segmentos fonéticos que nos permite afirmar que, se um fator modifica a duração em um sentido, aumentando ou diminuindo a duração, para uma certa combinação dos outros fatores, então o sentido de mudança será sempre o mesmo, independente dos outros fatores.

A seguir serão descritos sucintamente alguns modelos de duração. Os modelos descritos são os modelos baseados em dicionários, em *clustering* não hierárquico, o modelo de duração de Klatt, modelos de soma de produtos, o modelo de Campbell e modelos baseados em árvore de decisão.

3.5.1 Modelo baseados em dicionários

Seja $\mathbf{f} = \{f_1, f_2, \ldots, f_N\}$ um vetor do espaço de descrição do fone, onde f_i representa um fator que modifica a duração (e.g., identidade do fone, contexto fonético, tonicidade etc). Em um modelo baseado em dicionários de duração, a duração prevista para um dado fone de um enunciado a ser sintetizado é a média das durações de todos os fones do conjunto de treinamento que têm a mesma combinação de fatores $\mathbf{f}$ do fone do enunciado. Este modelo é bastante sensível ao problema de esparsividade, pois, caso a combinação de fatores não esteja presente no conjunto de treinamento, é impossível prever a duração do fone (ou tentar uma estimativa baseando-se na duração média da combinação de fatores mais próxima). Além disso, se o número de fones com a mesma combinação de fatores for pequeno, a média pode não ser uma estimativa significativa da duração.

3.5.2 Modelos baseados em *clustering* não hierárquico

Os vetores $\mathbf{f}$ dos fones do conjunto de treinamento, são agrupados em células através de um processo de *clustering* não hierárquico. Neste modelo, mesmo que o número de células não seja pequeno, ou que o algoritmo de *clustering* não cometa erros na alocação das células, fones com combinações diferentes de fatores poderão ser colocados na mesma célula, possivelmente aumentando o erro da estimativa.

Também podem ocorrer problemas de esparsividade de dados neste modelo. Pode-se tentar obter a duração de um fone, cuja combinação de fatores não está no conjunto de treinamento, usando a duração média da célula mais próxima.

3.5.3 Modelo de Klatt

O modelo de Klatt faz parte da classe dos modelos multiplicativos. Nos modelos multiplicativos, a duração $d(\mathbf{f})$ de um fone é uma função de várias variáveis (combinação de fatores), definida como o produto de várias funções de uma única variável ($F_i(f_i)$), cada uma destas associada a um dado fator que influencia na duração do fone, como vemos na equação (3.1).

$$d(\mathbf{f}) = \prod_{i=1}^{N} F_i(f_i) \tag{3.1}$$

No modelo de Klatt, assume-se que todos os fones possuem uma duração inerente e uma duração mínima. A duração de um fone não pode ser reduzida para um valor abaixo da duração mínima. A duração mínima pode ser obtida através da duração inerente [15] ou verificando a partir de que valor de duração os resultados da síntese são desagradáveis [7]. Este modelo gera as durações dos segmentos através da equação (3.2)

$$d(p) = \left(\prod_{i=1}^{N} K_{c_i} \right) (d_{inh}(p) - d_{min}(p)) + d_{min}(p) \tag{3.2}$$

Onde $d(p)$ é a duração do fone p sujeito aos fatores contextuais c_i, $i = 1, ..., N$, K_{c_i} são coeficientes relacionados aos fatores contextuais, $d_{inh}(p)$ é a duração inerente do segmento e $d_{min}(p)$ é a duração mínima do fone [46]. Os valores dos coeficientes K_{c_i} estão associados a um conjunto de regras [7], por exemplo, se a palavra que contém o fone é uma palavra de conteúdo, a sua duração é aumentada por um fator $K_{content} = 1.2$ [7]. De acordo com Klatt, existem sete fatores relevantes na determinação da duração: acentuação silábica, ênfase da palavra, fonemas anteriores, fonemas posteriores, posição na frase, posição dentro da palavra e identidade do segmento [46].

3.5.4 Modelos de soma de produtos

Os modelos de soma de produtos exibem melhores resultados que os modelos multiplicativos, pois capturam, melhor que aqueles, o fenômeno de invariância direcional e a interação entre os vários fatores. A duração de um segmento é dada pela equação (3.3).

$$d(\mathbf{f}) = \sum_{j} \prod_{i} F_{i,j}(f_i) \tag{3.3}$$

3.5.5 Modelo de Campbell

O modelo de Campbell assume que a duração dos segmentos depende das interações entre características de nível segmental (modo e ponto de articulação e contexto fonético) e as de nível suprasegmental (processos operando a nível de sílaba ou acima) [14].

Para separar tais características, a duração das sílabas é calculada usando fatores contextuais de nível suprasegmental e informações mínimas a nível segmental, como o número de fones na sílaba e o tipo de núcleo silábico (por exemplo, se é um *schwa*, um ditongo etc). A duração das sílabas é obtida a partir dos fatores contextuais usando uma rede neural.

A duração de cada segmento é calculada posteriormente, a partir das distribuições dos fones. Como o logaritmo da duração das sílabas possui uma distribuição aproximadamente normal quando são separadas nas classes associadas às sílabas tônicas e átonas, isso sugere o uso do logaritmo da duração ao invés da duração. A duração d_i de cada segmento é obtida através de um princípio de elasticidade forte, que afirma que todos os segmentos que compõe uma sílaba são expandidos ou contraídos do mesmo fator k. O valor do fator de expansão/contração k é obtido a partir da equação (3.4)

$$s_i = \sum_{i=1}^{n} e^{(\mu_i + k\sigma_i)} \tag{3.4}$$

onde s_i é a duração da sílaba, n é o número de fones que a sílaba contém, e μ_i e σ_i são a média e o desvio padrão do logaritmo da duração do i-ésimo segmento, respectivamente. A duração de cada segmento é dada por $d_i = e^{(\mu_i + k\sigma_i)}$.

3.5.6 Modelos baseados em árvores de decisão

Os modelos baseados em árvores de decisão particionam o espaço de descrição do fone. O algoritmo de construção da árvore de decisão procura diminuir a variância da duração nas partições do espaço de descrição. Desta forma, o algoritmo seleciona automaticamente quais os melhores fatores para classificar os fones de acordo com a duração.

A duração prevista para o fone é obtida percorrendo a árvore até atingir um nó folha. Uma desvantagem deste modelo em relação aos modelos multiplicativos ou de soma de produtos é o fato de que a generalização do modelo não é boa.

3.6 Modelos de entonação

Os modelos de entonação são responsáveis em mapear as informações lingüísticas no contorno de F0. A Figura 3.2 é um diagrama de blocos que ilustra os principais blocos de um modelo de entonação. Primeiro, a prosódia de alto nível é mapeada em uma representação abstrata do contorno de F0 através de uma análise entoacional. Depois, o modelo lingüístico faz o mapeamento desta representação abstrata do contorno de F0, que será chamada de *representação fonológica*, em uma representação paramétrica do contorno de F0, que será chamada de *representação fonética*. Esta representação fonética é a entrada do modelo acústico, que a converte no contorno de F0 final.

O principal problema na obtenção do contorno de F0 é a sua grande variabilidade, não só entre locutores mas também para um só locutor. A variabilidade entre locutores sugere a obtenção de um modelo de entonação específico para cada locutor, de preferência o mesmo locutor que gravou o inventório de unidades, no caso de um sistema concatenativo de síntese de voz. Outro problema é o fato de que os elementos do conjunto imagem do mapeamento realizado pelo modelo de entonação não são escalares (como no caso da duração) ou vetores, mas sim curvas. Esta última característica é o que sugere a introdução de um modelo acústico, como pode ser visto na Figura 3.2. Além disso, assim como ocorre com a duração, um grande número de fatores interagem influenciando no contorno de F0 [40].

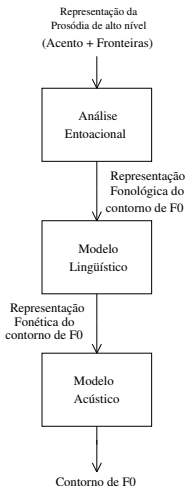


Figura 3.2: Modelos de entonação.

3.6.1 Modelos acústicos

Os modelos acústicos não passam de parametrizações do contorno de F0, que visam diminuir a dimensionalidade da curva, obtendo um pequeno número de parâmetros que descrevem um conjunto de vários pontos do contorno. Por isto, estas parametrizações são aproximações da curva original e também são conhecidas como *estilizações* do contorno de F0 [3]. Além da redução da dimensionalidade, os modelos acústicos podem ajudar na separação das componentes macroprosódica e microprosódica da entonação [47].

Alguns exemplos de modelos acústicos são:

1. Parâmetros RFC - O nome RFC vem de *rise/fall/connection*. Os eventos significativos do contorno de F0 são aproximados por quatro parâmetros. Os eventos são divididos em subidas e descidas que são aproximadas por parábolas. Os parâmetros são a amplitude e a duração da subida e a amplitude e a duração da descida. Os eventos são interconectados por retas [48][3];

2. Parâmetros Tilt - Esta é uma representação mais compacta que a RFC. Os eventos são aproximados por três parâmetros: duração, amplitude e *tilt*, onde este último indica a forma geral da curva. Estes parâmetros podem ser obtidos a partir dos parâmetros RFC [49];

3. *Splines* quadráticas - Uma função *spline* é uma seqüência contínua de polinômios de grau n, cujas derivadas até ordem $(n-1)$, inclusive, são contínuas.

Esta representação, com *splines* quadráticas, exige três parâmetros [47][3];

4. Linear - Cada evento interessante do contorno de F0 é aproximado por uma seqüência de segmentos de retas [3];

5. PaIntE - Os eventos significativos do contorno de F0 são aproximados por uma soma de duas sigmóides. Esta representação usa seis parâmetros [50].

Outro modelo acústico muito conhecido é o modelo de Fujisaki, que será apresentado em maiores detalhes na Seção 3.6.3.

3.6.2 Modelos lingüísticos

Os modelos lingüísticos são os responsáveis pelo mapeamento da representação abstrata do contorno de F0 na representação paramétrica associada a um modelo acústico. Um dos modelos lingüísticos mais conhecidos é o modelo de seqüência de tons (TSM - *tone sequence model*) de Pierrehumbert, que descreve o contorno de F0 como uma seqüência de tons relativos [3]. Este modelo foi a base do sistema de anotação prosódica ToBI (*Tones and Break Indices*) [3]. Outro exemplo de modelo lingüístico é o RFC (*rise/fall/connection*), que faz uma descrição do contorno de F0 como sendo uma seqüência de subidas, descidas e conexões [48].

Assumindo, em relação ao modelo de Fujisaki, que os comandos de frase têm alguma relação com as fronteiras de constituintes prosódicos, e que os comandos de acento têm alguma relação com a acentuação na frase, podemos observar que os parâmetros do modelo de Fujisaki são uma ótima representação fonológica dos enunciados. Desta maneira, torna-se relativamente simples construir um modelo lingüístico baseado no modelo de Fujisaki.

3.6.3 Modelo de Fujisaki

O modelo de Fujisaki é um modelo acústico baseado em propriedades fisiológicas da laringe [3][34]. A descrição do contorno de F0 é feita através de eventos discretos que são a entrada de filtros lineares de segunda ordem criticamente amortecidos [3]. As equações (3.5), (3.6) e (3.7) descrevem o modelo.

A Figura 3.3 apresenta um diagrama do modelo. Como podemos ver na figura, este é um modelo de sobreposição, que descreve o contorno de F0 de uma

sentença como a soma das respostas dos filtros mencionados. Neste trabalho não será levado em conta o mecanismo de oscilação glotal, com o intuito de simplificar o modelo, sem perda de generalidade.

$$\begin{aligned} \ln F_0(t) &= \ln F_b + \sum_{i=1}^{N_p} A_{p_i} G_p(t - T_{p_i}) + \\ &\quad + \sum_{j=1}^{N_a} A_{aj} \left[G_a(t - T_{a1_j}) - G_a(t - T_{a2_j}) \right] \end{aligned} \tag{3.5}$$

onde $G_p(t)$ e $G_a(t)$ são dados por

$$G_p(t) = \begin{cases} \alpha^2 t e^{-\alpha t}, & \text{for } t \geq 0 \\ 0, & \text{for } t < 0 \end{cases} \tag{3.6}$$

$$G_a(t) = \begin{cases} \min\left[1 - (1 + \beta t)e^{-\beta t}, \gamma\right], & \text{for } t \geq 0 \\ 0, & \text{for } t < 0 \end{cases} \tag{3.7}$$

Todas as variáveis associadas ao modelo estão resumidas na Tabela 3.1. Os parâmetros α, β e γ são parâmetros dos filtros, e portanto estão relacionados às características intrínsecas do aparelho fonador de um dado locutor. $G_p(t)$ representa o controle de frase e $G_a(t)$ representa o controle de acento. Os outros parâmetros (F_b, A_{p_k}, T_{p_k}, A_{ak}, T_{a1k} e T_{a2k}) estariam associados aos comandos que o cérebro do locutor enviaria ao seu aparelho fonador. F_b é a freqüência de base que é adicionada às saídas dos filtros, e é a freqüência para a qual tende a saída do modelo no caso em que se pára de aplicar comandos na entrada. Os parâmetros A_{p_k} e T_{p_k} indicam a amplitude e o instante de aplicação de cada comando de frase, respectivamente. Na equação (3.5), N_p é o número de comandos de frase. Já os parâmetros A_{ak}, T_{a1k} e T_{a2k} indicam a amplitude e o tempo de início e fim de cada comando de acento, respectivamente. Na equação (3.5), N_a é o número de comandos de acento.

No modelo de Fujisaki, quanto maior o número de comandos de acento e de frase, melhor a aproximação do contorno de F0. Por isso, devemos saber de antemão o número de comandos para ter uma descrição lingüisticamente significante. O número de comandos pode ser obtido através de uma análise lingüística dos textos associados aos enunciados, se os textos forem disponíveis [34]. De posse

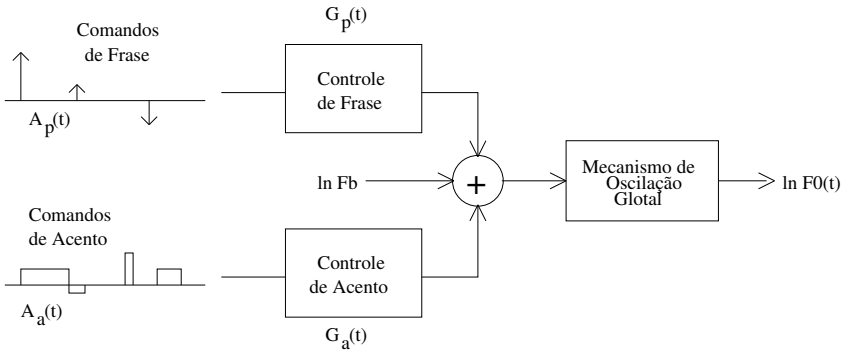


Figura 3.3: Modelo de Fujisaki.

do número de comandos, a extração dos parâmetros pode ser feita através de um processo de otimização que busca minimizar a diferença entre o contorno original e a aproximação [34][51].

3.7 Conclusão

Este capítulo apresentou o conceito de prosódia e teceu comentários sobre o problema de obter um modelo de prosódia. Implementar processamento prosódico em sistemas TTS é complicado, principalmente pelo fato de que ainda é difícil fazer análises lingüísticas detalhadas do texto.

Como a prosódia se manifesta principalmente através da evolução de certas propriedades acústicas do enunciado, como a freqüência fundamental ao longo do tempo e a duração dos segmentos, um sistema TTS deve ser capaz de modificar estas propriedades para poder introduzir prosódia no enunciado. São apresentados exemplos de modelos de duração e de entonação que tentam prever esses parâmetros no enunciado a ser sintetizado.

O modelo de Fujisaki é descrito em detalhes, pois foi o modelo de entonação escolhido neste trabalho. Este modelo tem o conveniente de que a entrada apresenta uma forte correspondência com as funções elementares da prosódia descritas em 3.4. Isso sugere que o modelo lingüístico, que converte a representação fonológica da entonação (posição e tipo das fronteiras entre constituintes e posição e tipo de acentos) na sua representação fonética (amplitudes e posições temporais dos comandos do modelo), seria bastante simples.

Tabela 3.1: Variáveis associadas ao modelo de Fujisaki.

F_b:	freqüência base
N_p:	número de comandos de frase
N_a:	número de comandos de acento
A_{pk}:	amplitude do k-ésimo comando de frase
A_{ak}:	amplitude do k-ésimo comando de acento
T_{pk}:	posição do k-ésimo comando de frase
T_{a1k}:	início do k-ésimo comando de acento
T_{a2k}:	final do k-ésimo comando de acento
$G_p(t)$:	resposta ao impulso do mecanismo de controle de frase
$G_a(t)$:	resposta ao impulso do mecanismo de controle de acento
α:	freqüência natural do mecanismo de frase
β:	freqüência natural do mecanismo de acento
γ:	nível máximo do componente de acento

Capítulo 4

TalkActive e Implementação do Modelo de Prosódia

4.1 Introdução

Este capítulo descreve o sistema TTS TalkActive, bem como algumas experiências realizadas e a forma como as modificações prosódicas são implementadas.

Primeiro será apresentado o sistema TalkActive na Seção 4.2, já que é neste sistema que será implementado o módulo de prosódia. A seção comenta todas as etapas de construção do sistema e suas características principais.

A Seção 4.3 descreve as experiências realizadas que tinham como objetivo verificar a viabilidade da imposição de prosódia pela modificação dos parâmetros prosódicos, e, em especial, permitir a avaliação da qualidade do enunciado sintético resultante da modificação desses parâmetros no TalkActive.

Já a Seção 4.4 descreve o algoritmo utilizado para a modificação prosódica. Vê-se nesta seção a forma como o algoritmo PSOLA é aplicado para modificar a duração de cada sílaba separadamente e impor um contorno de F0 arbitrário. Um modelo de prosódia simplificado é proposto para gerar o contorno de F0 a partir do texto.

4.2 TalkActive

O TalkActive é um sistema TTS para o português brasileiro (sotaque carioca), descrito mais detalhamente em [52], desenvolvido na UFRJ. Este sistema se baseia na abordagem concatenativa e tem as sílabas como unidades básicas.

É inviável gravar todas as realizações de todas as unidades em todos os contextos possíveis [1]. Por isso, durante a síntese por concatenação, é preciso, em alguns casos, modificar as unidades para o contexto em que serão inseridas. Tal modificação é a tarefa do algoritmo de concatenação escolhido, o TD-PSOLA (*Time Domain - Pitch Synchronous Overlap and Add*) [53]. Este algoritmo tem uma implementação bem simples e é o que resulta em voz sintetizada com maior naturalidade em relação às outras alternativas populares, como o LPC e o HNM [24].

Tendo escolhido a abordagem concatenativa e o tipo de unidade, isto nos levava a um dos principais problemas de um sintetizador deste tipo: a construção do banco de unidades. Para tal, fez-se uma pesquisa para listar todas as sílabas existentes na língua portuguesa, valendo-se de um dicionário. Esta pesquisa resultou numa lista contendo cerca de 2000 sílabas. O critério usado na escolha das sílabas foi ao mesmo tempo ortográfico e fonético. Por exemplo, as sílabas "a" e "ha" são consideradas diferentes, embora a realização fonética seja a mesma, e a sílaba "ra" nas palavras "rato" e "arara" são consideradas diferentes, pois no primeiro caso trata-se de uma fricativa velar e no segundo, um *tap* alveolar (no sotaque carioca). Por outro lado, não são diferenciadas sílabas nas posições tônica ou átona. Como exemplo de problema no inventório inicial de unidades podemos citar sílabas como "do" (/do/) que seria ouvida como /du/, se retirada da palavra "medo", devido ao fenômeno de redução vocálica.

Cada sílaba foi retirada de uma palavra que faz parte do léxico da língua, e que foi enunciada isoladamente e de forma natural, não estando inserida em um contexto frasal.

Após a gravação das palavras, as sílabas foram extraídas através de um procedimento semi-manual usando uma ferramenta de recorte, desenvolvida no ambiente Matlab e cuja interface é mostrada na Figura 4.1.

As sílabas recortadas passaram por uma etapa de pós-processamento, na qual foram obtidas as marcas de *pitch* indispensáveis ao algoritmo PSOLA, e na qual as

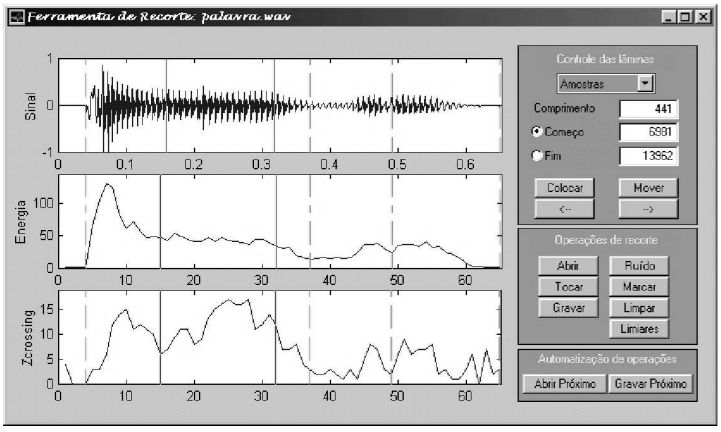


Figura 4.1: Interface do programa de recorte.

amostras nas extremidades da unidade, antes da primeira marca inclusive e depois da última exclusive, foram removidas para garantir continuidade de fase na concatenação [54]. A extração de pitch foi realizada com o algoritmo de autocorrelação com *center clipping*, a decisão de sonoridade se baseou na energia e na taxa de cruzamento pelo zero, e as marcas foram inseridas de forma a se posicionar na amostra de maior amplitude dentro de um ciclo, que é a posição considerada mais confiável [55].

A conversão do texto na seqüência de unidades é realizada através de um módulo de silabificação, que faz a separação do texto em sílabas. Este módulo consiste em um tipo de máquina de estados que decide sobre ações a serem tomadas a partir da seqüência de grafemas de entrada, como, por exemplo, juntar o grafema corrente na sílaba anteriormente formada para formar uma nova sílaba, iniciar nova sílaba ou separar o último grafema da última sílaba para formar nova sílaba com o grafema corrente. A máquina de estados decide a ação a ser tomada a partir de uma tabela contendo todas as seqüências de dois grafemas e a ação associada a cada uma delas.

Outro módulo é encarregado de, a partir da seqüência de sílabas, gerar a seqüência de unidades a serem concatenadas. Este módulo deve descobrir, através do contexto da sílaba ortográfica, qual a unidade do banco a ser selecionada. Como exemplo, podemos considerar as realizações do grafema "r" nas palavras "rato" e "arara". Apesar de ter a mesma realização ortográfica, "ra", as realizações fonéticas são distintas, correspondendo a unidades diferentes no inventório. Na versão 3.0, a

seleção da unidade é feita no sistema da forma mais simples, escolhendo a primeira unidade associada a uma certa realização ortográfica da sílaba.

Devido às exceções às regras de silabificação, é necessário um banco de exceções contendo palavras que não obedecem às regras e suas respectivas silabificações. Por exemplo, com as atuais regras, e sem que seja adicionada ao banco de exceções, a palavra "matéria" seria dividida em sílabas da seguinte forma: "ma-té-ri-a".

Foi acrescentado ao sistema um módulo de normalização de texto, que retorna um texto pré-processado no qual os números e as abreviaturas são escritos por extenso (por exemplo, "32" é escrito como "trinta e dois" e "Dr." é escrito como "doutor"), moedas e datas são interpretados ("R$ 20,00" torna-se "vinte reais" e "20/04" torna-se "vinte de abril") e os acrônimos são substituídos por uma seqüência de grafemas que corresponde à sua pronúncia (por exemplo, "IEEE" é escrito como "i três é").

Ainda na versão 3.0, o sistema não possui processamento lingüístico de alto nível nem modelamento prosódico. Os enunciados são sintetizados com um contorno de *pitch* uniforme. O *pitch* e a velocidade da fala podem ser alterados com a técnica PSOLA, mas qualquer alteração do pitch altera também a velocidade.

Todo o sistema foi programado em C++, usando o Visual C++ 6.0. A interface com o usuário foi implementada usando componentes MFC (*Microsoft Foundations Classes*). A Figura 4.2 ilustra a interface do TalkActive. O uso de programação orientada a objetos provou ser útil na implementação do sistema devido à modularidade do projeto, permitindo uma melhor organização do código.

Foram realizados testes de inteligibilidade e preferência para avaliar a qualidade do sistema. Os testes de inteligibilidade medem o número de erros na compreensão de palavras e frases, enquanto os testes de preferência compararam o TalkActive com vários outros sintetizadores, acadêmicos e comerciais. Os testes de inteligibilidade são os mesmos descritos em [18].

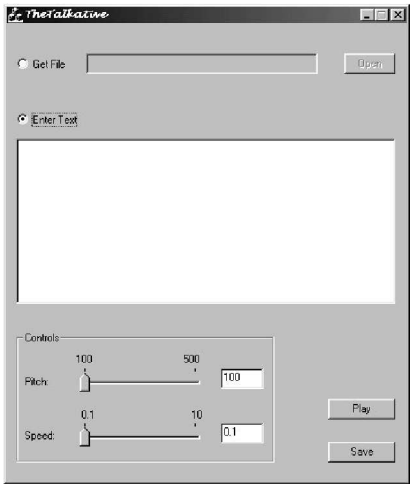


Figura 4.2: Interface do TalkActive.

4.3 Experiências

As experiências serão descritas mais detalhadamente a seguir. Elas tinham como objetivo mostrar:

- Como a percepção do enunciado é alterada com a estilização do contorno de F0. As experiências sugerem que mesmo estilizações lineares grosseiras do contorno de F0 ainda preservam grande parte das informações lingüísticas de alto nível, além de resultarem em enunciados extremamente naturais;

- Quão importantes a freqüência fundamental e a duração são para carrear informações lingüísticas de alto nível. Experiências para transformar enunciados de diferentes modos sentenciais (interrogativo para declarativo), modificando a freqüência fundamental e a duração, sugerem que a modificação destes parâmetros é suficiente para conferir boa naturalidade, se a qualidade segmental é alta e se sabemos como modificar tais parâmetros;

- A diferença entre modificar a duração de cada fone que constitue a sílaba separadamente ou modificar a sílaba inteira através de uma expansão ou compressão linear. Foi verificado que, embora a modificação independente das durações de cada fone confira uma qualidade melhor, as experiências sugerem que a diferença de qualidade é pequena;

- O quanto melhora a qualidade do enunciado sintetizado através da concatenação de sílabas da base de dados do TalkActive, se utilizamos as mesmas

durações de sílaba e o mesmo contorno de F0 de um enunciado natural. Com esta experiência chega-se à conclusão de que a qualidade segmental deve antes ser muito boa para que a modificação prosódica possa garantir alta qualidade.

4.3.1 Estilização linear do contorno de F0

Realizaram-se experiências em que o contorno original de F0 foi substituído por um contorno mais simples, que pode ser descrito como a concatenação de segmentos de reta. A modificação do contorno de F0 foi feita através do programa PRAAT, utilizando o algoritmo PSOLA. Dois níveis de estilização linear foram utilizadas: uma estilização grosseira, onde o contorno é aproximado por um pequeno número de segmentos de reta, procurando, em alguns casos, seguir apenas a declinação da F0, e uma estilização mais acurada, onde tenta-se aproximar movimentos mais localizados do contorno de F0, normalmente associados à acentuação.

As Figuras 4.3 e 4.5 são exemplos da estilização grosseira do contorno de F0, para as frases "A matéria do jornal foi bastante discutida" e "O meu chefe foi almoçar com o presidente da empresa", nos modos sentenciais declarativo e interrogativo, respectivamente. No caso da estilização grosseira, percebe-se uma diferença enorme entre a entonação dos enunciados estilizados e dos originais. Além disso, o enunciado estilizado não possui uma entonação natural.

Já no caso da estilização mais acurada do contorno de F0, exemplificada nas Figuras 4.4 e 4.6, embora seja possível perceber diferenças entre a entonação do enunciado original e do estilizado, as diferenças são muito pequenas, e, em geral, não se percebem mudanças no modo sentencial ou em outras características lingüísticas ou paralingüísticas do enunciado. Por outro lado, o enunciado estilizado possui uma entonação bastante natural na maioria dos casos. Em alguns casos, a diferença pode ser explicada pelos artifícios introduzidos pelo algoritmo de modificação prosódica.

Foi constatado que as diferenças entre os enunciados estilizados e originais são maiores nos modos sentenciais exclamativo e interrogativo. No caso do modo interrogativo, a realização do contorno de F0 no final do enunciado deve ser bem acurada para não modificar o modo sentencial.

Esta experiência demonstra que, no caso de frases declarativas, e até no caso de outros modos sentenciais, existe uma certa tolerância com relação à síntese do

contorno de F0, mesmo que se pretenda uma entonação extremamente natural.

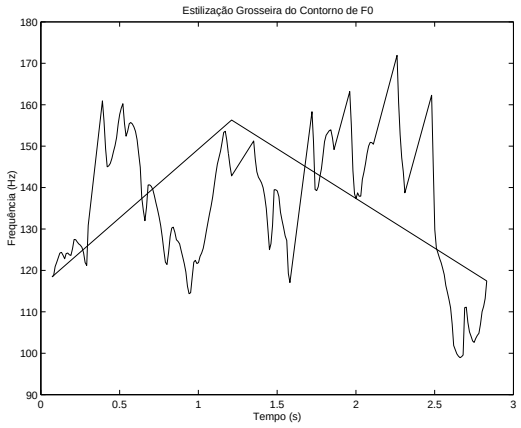


Figura 4.3: Estilização linear grosseira do contorno de F0 da frase "A matéria do jornal foi bastante discutida", em modo sentencial declarativo.

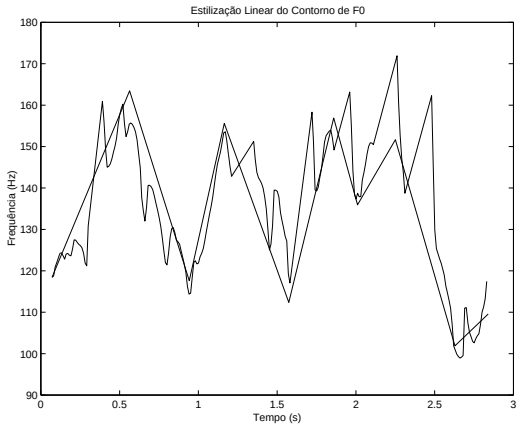


Figura 4.4: Estilização linear mais acurada do contorno de F0 da frase "A matéria do jornal foi bastante discutida", em modo sentencial declarativo.

4.3.2 Modificação do modo sentencial de enunciados naturais

Uma das experiências realizadas tinha como objetivo transferir parâmetros prosódicos (F0 e duração) de um enunciado em um modo sentencial para outro de modo sentencial diverso. Como o TalkActive utiliza sílabas como unidades, foi usada a duração das sílabas do enunciado. Novamente a modificação do enunciado foi feita com o PRAAT, utilizando PSOLA. Um *script* de Matlab foi escrito para, a partir de um arquivo no formato TEXTGRID, do PRAAT, com a descrição das fronteiras

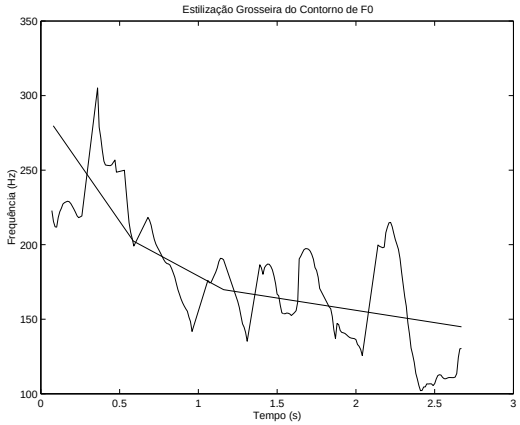


Figura 4.5: Estilização linear grosseira do contorno de F0 da frase "O meu chefe foi almoçar com o presidente da empresa", em modo sentencial declarativo.

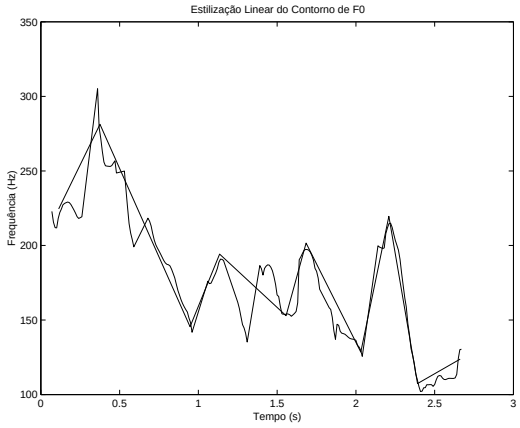


Figura 4.6: Estilização linear mais acurada do contorno de F0 da frase "O meu chefe foi almoçar com o presidente da empresa", em modo sentencial declarativo.

do segmento no sinal de voz, gerar um arquivo que é utilizado pelo PRAAT para modificar a duração dos segmentos, de forma a promover o alinhamento.

Para possibilitar a transferência do contorno de F0, é necessário alinhar as sílabas, implicando na necessidade de modificar também a duração das sílabas. Na maioria dos casos, o enunciado modificado ficou completamente indistinguível do enunciado de referência (enunciado do qual foram extraídos os parâmetros prosódicos). Nos casos em que isto não ocorreu, foi devido aos artifícios do algoritmo de modificação prosódica e também à falta de alinhamento dos fones dentro da sílaba. Para verificar esta última hipótese, em um dos enunciados em que perceberam-se diferenças, foi feito o alinhamento a nível de fones, minimizando bastante a diferença entre os enunciados modificado e de referência.

4.3.3 Modificação de parâmetros prosódicos no TalkActive

Para ter uma idéia da qualidade do enunciado sintetizado pelo TalkActive, com o atual inventório de unidades, caso a geração do contorno de F0 e a estimativa da duração das sílabas seja perfeita, o sistema foi modificado para aceitar um arquivo contendo as marcas de *pitch* associadas a cada sílaba de um dado enunciado. Um *script* de Matlab foi escrito para obter as marcas de *pitch* a partir dos arquivos nos formatos PITCHTIER e TEXTGRID, gerados a partir do PRAAT.

O enunciado "A matéria do jornal foi bastante discutida", em vários modos sentenciais, foi empregado na experiência. Foi constatado que a qualidade do enunciado não é muito boa, mesmo com prosódia perfeita, devido à baixa qualidade segmental do sistema.

4.4 Algoritmo de modificação prosódica

O algoritmo de modificação prosódica utilizado baseia-se no TD-PSOLA (*time domain - pitch synchronous overlap-and-add*) [53]. No PSOLA, são obtidos sinais de curta duração através da multiplicação do sinal de voz por janelas de Hamming centradas nas marcas de *pitch*, durante a etapa de análise. Esta etapa nos fornece uma representação intermediária do sinal de voz na forma de marcas de *pitch* associadas a sinais de curta duração. A modificação do *pitch* é realizada por meio de mudanças das posições das marcas, e a duração dos segmentos é modificada através da inserção ou remoção de marcas de *pitch* [53].

Quando o contorno de F0 é uniforme, a modificação do *pitch* e do tempo total do enunciado é relativamente trivial. Para modificar o *pitch*, forçamos as novas posições das marcas de *pitch* a serem espaçadas de um período igual ao correspondente à freqüência fundamental desejada. Para modificar a duração foi elaborado um procedimento no qual a última marca de *pitch* do enunciado é forçada a coincidir com a última marca de *pitch* desejada, através de um mapeamento linear. Depois, para cada marca de *pitch* desejada, encontra-se a marca de *pitch* original que, após o mapeamento linear, está mais próxima da marca de *pitch* desejada. O sinal de curta duração associado a esta marca de *pitch* original é copiado para a representação intermediária do sinal a ser sintetizado e é associado à marca de *pitch* desejada.

Este procedimento pode ser reformulado para permitir a imposição de um contorno de F0 arbitrário e a contração/expansão temporal linear de um número qualquer de segmentos contíguos do sinal de voz. As marcas de *pitch* desejadas são obtidas considerando o contorno de F0 desejado e as durações dos segmentos, através do seguinte algoritmo:

```
int  nmt = 0;   // número total de marcas
long marca = 1; // posição da primeira marca do enunciado

// Faz um loop em todos os segmentos do sinal
// de voz (podem ser fones, difones ou sílabas).
// A variável nseg contém o número de segmentos.
for (c = 0; c < nseg; c++) {

   int nm = 0; // número de marcas de pitch do segmento

   // Conta número de marcas, calcula e armazena
   // as marcas deste segmento. A variável bound[c]
   // contém a posição da última amostra do
   // (c+1)-ésimo segmento.
   while (marca < bound[c]) {

      // GetPitch é uma função que retorna o valor de
      // F0 (em Hertz) para o tempo t (em segundos).
      // FS é a frequência de amostragem do sinal de voz.
      nm++;
      double t = (double)(marca-1)/(double)FS;
      double pitch = GetPitch(t);
      seg[c].pmark[(nm - 1) + nmt] = marca;
      marca += (long) rint((double)FS/(double)pitch);

   }
```

```
   seg[c].np = nm; // número de marcas de pitch no segmento
   nmt += nm; // número total de marcas de pitch

}
// A partir deste ponto, a variável seg contém as marcas
// de pitch de cada segmento.
```

Para associar as marcas de *pitch* originais com as marcas desejadas, faz-se um mapeamento linear em cada segmento. Sejam c_d e f_d variáveis que nos dão as marcas desejadas de começo e fim, respectivamente, para um segmento, e c_o e f_o as marcas originais para o mesmo segmento, do começo e do fim, respectivamente. Queremos encontrar um mapeamento $f : x \rightarrow y$ com as seguintes restrições: $c_d \rightarrow c_o$ e $f_d \rightarrow f_o$. Um mapeamento possível é o mapeamento linear $y = f(x) = \alpha x + \beta$. Os valores de α e β, para cada segmento, que fazem o mapeamento obedecer as restrições descritas são:

$$\alpha = \frac{c_o - f_o}{c_d - f_d} \tag{4.1}$$

$$\beta = \frac{c_d f_o - c_o f_d}{c_d - f_d} \tag{4.2}$$

O seguinte procedimento é utilizado para obter a relação entre as marcas desejadas e as originais. Após encontrar qual marca original corresponde à marca desejada, o sinal de curta duração associado à marca original é copiado para a representação intermediária desejada.

```
// As marcas de pitch originais de cada segmento
// são colocadas no vetor pmark
int p = 0;
for (int i = 0; i < nseg; i++)
   for (int j = 0; j < seg_orig[i].np; j++)
      pmark[p++] = seg_orig[i].pmark[j];

int npt_orig = p; // número total de marcas de pitch originais

// Realiza mapeamento linear em partes
```

```
p = 0;
for (int i = 0; i < nseg; i++) {

   int np = seg[i].np;           // número de marcas desejadas
   int np_orig = seg_orig[i].np; // número de marcas originais

   // Marcas de pitch do começo e fim dos segmentos
   double cd = (double) seg[i].pmark[0];
   double co = (double) seg_orig[i].pmark[0];
   double fd = (double) seg[i].pmark[np - 1];
   double fo = (double) seg_orig[i].pmark[np_orig - 1];

   // Cálculo de alfa e beta
   double den     = cd - fd;
   double numalfa = co - fo;
   double numbeta = cd * fo - co * fd;

   double alfa = numalfa/den;
   double beta = numbeta/den;

   // Aplicando transformação linear
   for (int j = 0; j < np; j++)
      trans[p++] = alfa * (double)seg[i].pmark[j] + beta;

}

int npt = p; // número total de marcas de pitch desejadas

// Primeiro sinal de curta duração é associado à primeira
// marca de pitch. A função CopySTSignal(i, j) copia o
// sinal de curta duração da representação intermediária
// original, associado à (i+1)-ésima marca de pitch, para
```

```
// representação intermediária do sinal a ser sintetizado,
// associando-o à (j+1)-ésima marca de pitch.
CopySTSignal(0, 0);

// Associa marcas de pitch originais a marcas de pitch
// desejadas, e copia os sinais de curta duração
for (p = 1; p < npt; p++) {

   // minimo é inicializado com o maior
   // valor possível que pode assumir
   double minimo = (double) pmark[npt_orig-1];
   int index = 0; // índice da marca mais próxima

   // Busca marca original mais próxima da desejada
   // (após a transformação linear)
   for (int i = 1; i < npt_orig; i++) {
      double dif = fabs((double) pmark[i] - trans[p]);
      if (dif < minimo) {
         minimo = dif;
         index = i;
      }
   }
   // Copia sinal de curta duração da marca original mais
   // próxima ((index+i)-ésima marca) para a representação
   // intermediária do sinal a ser sintetizada, associan-
   // do-o à (p+1)-ésima marca.
   CopySTSignal(p, index);

}
```

4.5 Módulo simplificado e resultados obtidos

O módulo de prosódia implementado no TalkActive apenas altera o contorno de *pitch*, não alterando a duração original das sílabas nem as amplitudes. O contorno de F0 é previsto pelo modelo de Fujisaki, usando informações lingüísticas de fácil determinação. Os parâmetros prosódicos são alterados utilizando o algoritmo descrito na Seção 4.4. Assume-se que cada enunciado possui um único comando de frase, que é aplicado no tempo $t = -\frac{1}{\alpha}$, de forma que o máximo da componente de frase ocorra em $t = 0$. O começo e o final de um comando de acento coincidem com o começo e o final das sílabas tônicas (detectando acento léxico ao invés do acento frasal). A sílaba tônica é detectada através de um conjunto simples de regras, não havendo um banco de exceções para determinar o acento léxico de palavras que não obedeçam às regras.

A motivação para usar um modelo tão simples está no fato de que a determinação da posição dos comandos de frase exigiria uma análise léxico-morfológica de complicada implementação, e, em alguns casos, até uma análise sintática ou semântica. Isto porque o mais simples dos algoritmos para determinar as fronteiras entre constituintes prosódicos precisa descobrir a localização dos verbos na frase [3]. O acento léxico é empregado no lugar do acento frasal já que não foram encontradas referências sobre a detecção do acento frasal em frases no PB.

As amplitudes dos comandos de frase e de acento são constantes, independendo de fatores como modo sentencial, comprimento das palavras, posição da sílaba tônica em relação às fronteiras de constituintes prosódicos etc. As constantes do modelo de Fujisaki são inicializadas com os valores $\alpha = 3$, $\beta = 20$ e $\gamma = 0.9$. Em um modelo mais acurado, espera-se que as amplitudes dos comandos sejam obtidos através de através de um mapeamento que leve informações lingüísticas de alto nível nestas amplitudes.

Mesmo com um modelo tão simples, percebeu-se um grande aumento da naturalidade dos enunciados sintetizados com este módulo de prosódia em relação aos sintetizados com *pitch* uniforme. Ainda assim, devido à baixa qualidade segmental do sistema, pode ser que um enunciado sintetizado com *pitch* uniforme apresente maior inteligibilidade que os enunciados em que a entonação é modificada com a ajuda do modelo de Fujisaki, já que pouca informação relevante é adicionada ao

enunciado pelo módulo de prosódia.

4.6 Conclusão

Este capítulo descreve o sistema TalkActive, as experiências realizadas, o algoritmo de modificação prosódica implementado no TalkActive e o modelo de prosódia simplificado para gerar o contorno de F0 a partir do texto.

As experiências realizadas nos levam à conclusão de que não se deve esperar uma qualidade muito boa na síntese de voz, mesmo com geração perfeita de prosódia, se não houver uma alta qualidade segmental (correta seleção das unidades e um inventório de unidades bem construído). Além disso, o algoritmo de modificação prosódica pode introduzir artifícios no sinal de voz.

Apesar disso, observou-se que, em geral, não é necessário prever a duração de cada fone para conseguir alta naturalidade, mas apenas a duração das sílabas. E que mesmo contornos de F0 relativamente simples podem gerar uma entonação extremamente natural, principalmente no caso de sentenças declarativas. Isto nos mostra que não é tão importante obter valores precisos da freqüência fundamental para termos um enunciado sintetizado de alta naturalidade.

O algoritmo de modificação prosódica apresentado na Seção 4.4 é baseado no PSOLA, e os segmentos são sílabas, ao invés de fones. O algoritmo modifica separadamente a duração das sílabas, impondo um contorno de F0 arbitrário. O módulo de prosódia construído é bem simples, usando somente a informação sobre a localização das sílabas tônicas, e amplitudes constantes para os comandos do modelo de Fujisaki. Mesmo com a simplicidade deste módulo, percebemos uma grande diferença de qualidade entre a voz sintetizada com este módulo e com F0 uniforme.

Capítulo 5

Inversão do Modelo de Fujisaki

5.1 Introdução

O modelo de Fujisaki é um modelo muito simples, que gera o contorno de F0 a partir de uma entrada formada por um conjunto de comandos de frase e de acento. A entrada do modelo tem motivação lingüística, podendo-se associar os comandos do modelo às funções elementares da prosódia. O modelo é baseado nas restrições fisiológicas do aparelho fonador, que são as mesmas em todos os seres humanos, o que resulta em independência da linguagem. Por isso tem sido empregado com sucesso em várias linguagens [56].

Este capítulo aborda o problema de inversão do modelo de Fujisaki, que é a determinação dos comandos de frase e acento (entrada) a partir do contorno de F0 que deve ser gerado pelo modelo (saída). Este é um problema complicado, devido à característica de sobreposição exibida pelo modelo.

Tais algoritmos de inversão determinam os comandos do modelo encarando a inversão como um problema de otimização de uma função objetivo não linear. A Seção 5.2 apresenta um resumo dos algoritmos descritos na literatura.

O método analítico proposto na Seção 5.3 determina as amplitudes dos comandos do modelo de Fujisaki, baseando-se em um desenvolvimento analítico do problema geral de otimização que resulta em uma solução em forma fechada para os parâmetros de amplitude.

A Seção 5.4 apresenta um algoritmo no qual as posições dos comandos são obtidas mediante a detecção de pontos críticos do contorno e subseqüente otimização

através de métodos iterativos, tal como é feito nos outros algoritmos de inversão.

Parte da importância deste novo método analítico está no fato de que ele separa a estimativa das amplitudes dos comandos da estimativa das posições. A determinação das amplitudes é um problema de otimização quadrática (considerando que as posições dos comandos são conhecidas de antemão), passível de solução analítica. Já a determinação das posições dos comandos é um problema complexo de otimização não linear, que não tem uma solução analítica geral.

Por outro lado, de um ponto de vista mais prático, a estimativa em forma fechada das amplitudes dos comandos do modelo permite, em geral, uma convergência mais rápida da otimização iterativa. Isto é verificado na Seção 5.5, que apresenta os resultados obtidos para o tempo de processamento e o erro da aproximação do contorno de F0 para diversas variações dos algoritmos de inversão.

5.2 Algoritmos de inversão

O problema de inversão do modelo de Fujisaki é aquele no qual tentamos encontrar os comandos de frase e acento (que são as entradas do modelo) e os valores de α, β e γ (que determinam a resposta dos mecanismos de controle de frase e acento), de modo a gerar uma certa curva de F0 na saída do modelo. No entanto, como estas últimas variáveis são praticamente constantes em qualquer enunciado de um dado locutor, o problema se reduz ao de encontrar os comandos do modelo para gerar uma dada curva de F0 na saída [57].

Muitos algoritmos descritos na literatura não têm acesso a nenhuma informação lingüística de alto nível associada ao contorno de F0. Em uma abordagem *top-down*, pode ser mais conveniente empregar um algoritmo totalmente automático que não requer dados adicionais além do contorno de F0. Porém, a complexidade da otimização sugere o uso de informações lingüísticas de alto nível para orientar o algoritmo, no sentido de encontrar os tempos dos comandos do modelo. Alguns algoritmos usam tais informações para restringir a busca pelos parâmetros [58] [59]. O algoritmo proposto na Seção 5.4 não se vale de nenhuma informação lingüística de alto nível.

Se as saídas dos mecanismos de controle, ilustrados na Figura 3.3, forem

consideradas separadamente, notamos que os componentes de frase e acento possuem mínimos locais (no caso do componente de frase) e mínimos e máximos locais (no caso do componente de acento) que correspondem aos instantes de aplicação do comando de frase, ou aos instantes inicial e final do comando de acento. Porém, devido à sobreposição dos comandos, os instantes dos pontos críticos do contorno de F0, suavizado para eliminar a microprosódia, podem não corresponder exatamente aos tempos dos comandos. Podemos, por outro lado, usar os pontos críticos para detectar a posição aproximada dos comandos.

A maioria dos algoritmos usa o fato de que componentes de variação lenta do contorno de F0 estão relacionadas aos comandos de frase e componentes de variação rápida estão relacionadas aos comandos de acento. Esta propriedade do contorno de F0 foi explorada para detectar fronteiras de constituintes prosódicos em [60]. Graças a ela, algoritmos relativamente simples conseguem separar, em geral, os pontos críticos do contorno associados aos comandos de frase daqueles associados aos comandos de acento.

Como o erro de aproximação do contorno é uma função não linear dos tempos dos comandos, a determinação automática dos parâmetros do modelo de Fujisaki é um problema difícil, geralmente tratado como um problema de otimização resolvido por técnica iterativas [57] [61] [62] [63].

De modo geral, os algoritmos possuem as seguintes etapas:

1. Pré-processamento – Nesta etapa são eliminadas as pequenas flutuações do contorno devidas à microprosódia. Isto pode ser feito através de um filtro passa-baixas, de mediana, ou por estilização do contorno por splines [57] [61] [63]. Nesta etapa também são corrigidos erros grosseiros do ADP [57];

2. Estimativa inicial – É encontrada uma estimativa inicial dos parâmetros do modelo, observando pontos críticos do contorno de F0 suavizado. A separação entre fraseamento e acentuação é feita através de um filtro ou fazendo a busca em intervalos de comprimentos diferentes;

3. Otimização – A estimativa inicial é modificada por meio de um procedimento iterativo para minimizar a função objetivo $F(P)$, levando a uma solução que se espera que seja mais próxima da solução ideal.

As técnicas mencionadas a seguir são usadas em alguns dos algoritmos, e são bastante singulares, no sentido de só serem encontradas em apenas um ou em poucos algoritmos descritos na literatura:

1. Uso de informações lingüísticas de alto nível para orientar a busca dos instantes dos comandos [58] [59];

2. Como os comandos têm uma dependência da esquerda para a direita, podemos estimar os comandos através de um procedimento *left-to-right*, e proceder à detecção de eventos prosódicos: seja P_i um vetor que contém os comandos estimados no intervalo de tempo $0 < t < t_i$. Se o contorno de F0 previsto pelo modelo, $\hat{F}_0(t, P_i)$, difere substancialmente do contorno de F0, $F_0(t)$, para $t > t_j$, isso significa que ocorreu um evento prosódico não previsto, no instante $t = t_j$, que não pode ser explicado apenas com os comandos em P_i. Portanto, um ou mais comandos devem ser adicionados ao vetor P_i [64];

3. Transformada wavelet para separar o fraseamento da acentuação, já que as bandas de freqüência dos componentes de frase e acento são muito próximas e as respostas dos mecanismos de controle são sinais não estacionários [65];

4. Etapa de otimização usando estratégia evolucionária ao invés de um algoritmo de otimização do tipo *hill-climbing* [65]. A dependência da esquerda para a direita pode ser usada para melhorar a otimização;

5. Otimização usando outra função objetivo, ao invés do erro médio quadrático. Um exemplo seria o FRF (*F0 reliability field*). A etapa de otimização deverá, então, maximizar o FRF médio [62].

O algoritmo proposto na Seção 5.4 não se aproveita de nenhuma dessas idéias. A novidade do algoritmo é o procedimento analítico para determinar as amplitudes dos comandos do modelo, sabendo-se a posição dos comandos. Porém, as idéias supracitadas podem ajudar a aperfeiçoar as estimativas das posições dos comandos.

5.3 Determinação analítica das amplitudes

Se $F_0(t)$ é o contorno de F0 de referência obtido por um algoritmo de determinação de *pitch* (ADP), e $\hat{F}_0(t)$ é o contorno de F0 do modelo, podemos afirmar que

$$\ln F_0(t) = \ln \hat{F}_0(t, P) + e(t, P), \tag{5.1}$$

onde $e(t, P)$, com $t \in [0, T]$, é o erro de estimação devido à inacurácia do modelo, e P é o vetor de parâmetros do modelo, dado por

$$\begin{aligned} P = \Big[& A_{p_1} \dots A_{p_{N_p}}, T_{p_1} \dots T_{p_{N_p}}, A_{a_1} \dots A_{a_{N_a}}, \\ & T_{a1_1} \dots T_{a1_{N_a}}, T_{a2_1} \dots T_{a2_{N_a}}, F_b \Big]^T. \end{aligned} \tag{5.2}$$

Para estimar P, devemos usar um procedimento de análise por síntese para minimizar o valor médio quadrático do erro de estimação dado por

$$F(P) = \frac{1}{T} \int_0^T e^2(t, P) dt. \tag{5.3}$$

Desta maneira, a extração de parâmetros do modelo de Fujisaki poderá ser considerada como um problema de otimização no qual pretende-se determinar P que minimiza $F(P)$. Como não temos uma expressão analítica para $F_0(t)$, mas apenas estimativas de $F_0(t)$ amostradas nos tempos $t_n = nT_{pitch}$, $n = 0, 1, \dots, m-1$, onde T_{pitch} é o intervalo entre cada estimativa de $F_0(t)$ e m é o número de estimativas de $F_0(t_n)$ geradas pelo ADP. Assim, $F(P)$ pode ser aproximado, no domínio do tempo discreto, por

$$F(P) \approx \varepsilon^2 = \frac{1}{m} \left(\mathbf{f_0} - \hat{\mathbf{f}}_\mathbf{0} \right)^T \left(\mathbf{f_0} - \hat{\mathbf{f}}_\mathbf{0} \right), \tag{5.4}$$

onde m é o número total de amostras de tempo e

$$\mathbf{f_0} = [\ \ln F_0(t_0) \quad \ln F_0(t_1) \quad \dots \quad \ln F_0(t_{m-1})\ ]^T, \tag{5.5}$$

$$\hat{\mathbf{f}}_\mathbf{0} = [\ \ln \hat{F}_0(t_0, P) \quad \ln \hat{F}_0(t_1, P) \quad \dots \quad \ln \hat{F}_0(t_{m-1}, P)\ ]^T. \tag{5.6}$$

Definindo os vetores auxiliares

$$\mathbf{A_p} = [\ A_{p_1} \quad A_{p_2} \quad \dots \quad A_{p_{N_p}}\ ]^T, \tag{5.7}$$

$$\mathbf{A_a} = [\ A_{a1} \quad A_{a2} \quad \dots \quad A_{aN_a}\ ]^T, \tag{5.8}$$

$$\mathbf{A} = \left[\ \mathbf{A_p}^T \ \mid \ \mathbf{A_a}^T\ \right]^T, \tag{5.9}$$

$$\mathbf{u} = [\ 1 \quad 1 \quad \dots \quad 1\ ]_{m \times 1}, \tag{5.10}$$

e as matrizes auxiliares

$$\mathbf{G_p} = \begin{bmatrix} G_{p_{1,1}} & G_{p_{1,2}} & \dots & G_{p_{1,N_p}} \\ G_{p_{2,1}} & G_{p_{2,2}} & \dots & G_{p_{2,N_p}} \\ \vdots & \vdots & \ddots & \vdots \\ G_{p_{m,1}} & G_{p_{m,2}} & \dots & G_{p_{m,N_p}} \end{bmatrix}, \tag{5.11}$$

$$\mathbf{G_a} = \begin{bmatrix} G_{a1,1} & G_{a1,2} & \dots & G_{a1,N_a} \\ G_{a2,1} & G_{a2,2} & \dots & G_{a2,N_a} \\ \vdots & \vdots & \ddots & \vdots \\ G_{am,1} & G_{am,2} & \dots & G_{am,N_a} \end{bmatrix}, \tag{5.12}$$

$$\mathbf{G} = \left[\begin{array}{c|c} \mathbf{G_p} & \mathbf{G_a} \end{array} \right], \tag{5.13}$$

onde

$$G_{p_{i,j}} = G_p(t_{i-1} - T_{p_j}), \tag{5.14}$$

$$G_{ai,j} = G_a(t_{i-1} - T_{a1_j}) - G_a(t_{i-1} - T_{a2_j}), \tag{5.15}$$

então a equação (3.5) pode ser reescrita como

$$\hat{\mathbf{f_0}} = (\ln F_b)\mathbf{u} + \mathbf{GA}. \tag{5.16}$$

A minimização de ε^2, definido na equação (5.4), é um problema de otimização complicado, devido à sua relação não linear com respeito a T_{p_k}, T_{a1k}, e T_{a2k}. Entretanto, quando analisamos a relação entre ε^2 e A_{p_k}, A_{ak}, e F_b, podemos verificar que a norma do erro é estritamente convexa, apresentando então um único mínimo local. Para que seja possível uma solução analítica, nós consideraremos o seguinte problema: achar os parâmetros A_{p_k}, A_{ak}, e F_b que minimizam ε^2, quando os parâmetros T_{p_k}, T_{a1k}, e T_{a2k} são dados. Para resolver analiticamente este novo problema, consideremos as derivadas de ε^2 em relação a $\mathbf{A}$ e $(\ln F_b)$:

$$\frac{\partial \varepsilon^2}{\partial \mathbf{A}} = \frac{2}{m}\mathbf{G}^T\{\mathbf{f}_0 - [(\ln F_b)\mathbf{u} + \mathbf{GA}]\} = \frac{2}{m}\mathbf{G}^T\mathbf{e}, \tag{5.17}$$

$$\frac{\partial \varepsilon^2}{\partial (\ln F_b)} = \frac{2}{m}\mathbf{u}^T\{\mathbf{f}_0 - [(\ln F_b)\mathbf{u} + \mathbf{GA}]\} = \frac{2}{m}\mathbf{u}^T\mathbf{e}, \tag{5.18}$$

onde o vetor $\mathbf{e}$ é o erro da aproximação, definido como

$$\mathbf{e} = \begin{bmatrix} e_1 & e_2 & \dots & e_m \end{bmatrix}^T, \tag{5.19}$$

onde $e_i = \ln \hat{F}_0(t_{i-1}, P) - \ln F_0(t_{i-1})$.

Igualando as derivadas a zero, temos a seguinte solução em forma fechada do problema:

$$\ln F_b = \frac{\mathbf{u}^T\mathbf{f}_0 - \mathbf{u}^T\mathbf{G}\left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{f}_0}{m - \mathbf{u}^T\mathbf{G}\left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{u}}, \tag{5.20}$$

$$\mathbf{A} = \left(\mathbf{G}^T\mathbf{G}\right)^{-1}\left[\mathbf{G}^T\mathbf{f}_0 - (\ln F_b)\mathbf{G}^T\mathbf{u}\right]. \tag{5.21}$$

Propõe-se uma modificação da etapa de otimização, extraindo os parâmetros de amplitude analiticamente, como descrito na próxima seção. O procedimento torna-se mais acurado e menos intenso computacionalmente quando se tenta extrair todos os parâmetros do modelo de Fujisaki através de otimização iterativa, após o uso do procedimento analítico.

5.4 Algoritmo proposto

Estas expressões analíticas podem ser empregadas por qualquer algoritmo de inversão do modelo de Fujisaki descrito na literatura. Por exemplo, aqui serão descritas versões modificadas do algoritmo de Rossi et al. [63] incorporando o procedimento analítico delineado acima para determinar a amplitude inicial dos comandos. Tal algoritmo gera os melhores resultados de estimação, na maioria dos experimentos, como demonstrado na seção seguinte.

Etapa 1: As regiões surdas do contorno de F0 são interpoladas e o contorno resultante é suavizado por um filtro de médias móveis.

Etapa 2: Procuram-se no contorno pré-processado por pontos dominantes no intervalo de tempo $T = 1\ s$. Estes pontos são usados com uma estimativa inicial da posição dos comandos de frase.

Etapa 3: As amplitudes dos comandos de frase, $\mathbf{A_p}$, e a freqüência base, F_b, são determinados assumindo a ausência de comandos de acento. Desta forma, o contorno de F0 gerado pelo modelo é

$$\hat{\mathbf{f_0}} = (\ln F_b)\mathbf{u} + \mathbf{G_p}\mathbf{A_p}, \tag{5.22}$$

com $\mathbf{A_a}$ sendo um vetor nulo. Os parâmetros $\mathbf{A_p}$ e F_b que minimizam ε_p^2 neste caso são dados pelas equations (5.20) e (5.21), substituindo $\mathbf{G}$ por $\mathbf{G_p}$ e $\mathbf{A}$ por $\mathbf{A_p}$, respectivamente. Isto é devido ao fato de que as equações (5.22) e (5.16)

têm a mesma estrutura. O algoritmo original em [63] determina as amplitudes usando o método descrito em [59], que é uma busca simples um espaço de solução grosseiramente discretizado.

Etapa 4: Determinados $\mathbf{A_p}$ e F_b, a resposta dos comandos de frase pode ser reconstruída e subtraída do contorno de F0 suavizado, resultando em um resíduo, que corresponde (no caso de uma reconstrução perfeita) à resposta dos comandos de acento.

Etapa 5: Buscam-se por pontos dominantes no intervalo de tempo Ta, no resíduo ($Ta = 50\ ms$ deu bons resultados em todos os testes), que corresponderão às estimativas iniciais das posições de começo e final dos comandos de acento.

Etapa 6: O algoritmo original em [63] determinaria as amplitudes $\mathbf{A_a}$ dos comandos de acento usando o procedimento de busca em [59]. Na versão modificada, empregamos um procedimento analítico que tem como objetivo minimizar o funcional $J(A_{ak})$ definido como

$$J(A_{ak}) = \frac{1}{T}\int_0^T e_a^2(t)dt, \tag{5.23}$$

onde

$$e_a(t) = r(t) - \sum_{k=1}^{N_a} A_{ak}[G_a(t - T_{a1k}) - G_a(t - T_{a2k})]. \tag{5.24}$$

A segunda parcela do segundo membro da equação representa a saída do mecanismo de controle de acento do modelo de Fujisaki e $r(t)$ é o resíduo, definido como

$$r(t) = F_0(t) - \hat{F}_{0parcial}(t), \tag{5.25}$$

sendo $\hat{F}_{0parcial}(t) = \left\{\hat{F}_0(t)|A_{ak} = 0, \forall k\right\}$ a saída do modelo assumindo que não há comandos de acento na entrada, ou seja, devida apenas aos comandos de frase. Para formular o problema em forma discreta, definimos o vetor de resíduo

$$\mathbf{r} = [\ r(t_0) \quad r(t_1) \quad \ldots \quad r(t_{m-1})\ ]^T. \tag{5.26}$$

Desta forma, considerando que a resposta dos comandos de acento é dada por $\mathbf{G_aA_a}$, queremos achar $\mathbf{A_a}$ que minimiza o funcional $J(A_{ak})$. Este funcional é aproximado por ε_a^2, definido como

$$J(A_{ak}) \approx \varepsilon_a^2 \ = \ \frac{1}{m}\left(\mathbf{r} - \mathbf{G_aA_a}\right)^T\left(\mathbf{r} - \mathbf{G_aA_a}\right). \tag{5.27}$$

Já que a derivada de ε_a^2 em relação a $\mathbf{A_a}$ é

$$\frac{\partial \varepsilon_a^2}{\partial \mathbf{A_a}} = \frac{-2\mathbf{G_a}^T}{m}\left(\mathbf{r} - \mathbf{G_a}\mathbf{A_a}\right), \tag{5.28}$$

então a função objetivo ε_a^2 é minimizada por

$$\mathbf{A_a} = \left[\mathbf{G_a}^T\mathbf{G_a}\right]^{-1}\mathbf{G_a}^T\mathbf{r}. \tag{5.29}$$

Etapa 7: Finalmente, uma busca iterativa é realizada para melhorar a estimativa inicial do início e final dos comandos.

5.5 Resultados

Nesta seção, aplicamos o procedimento proposto para determinação analítica das amplitudes dos comandos do modelo de Fujisaki em alguns dos algoritmos de inversão previamente apresentados na literatura. Foram comparadas as performances de quatro algoritmos distintos: Algoritmo I and Algoritmo III são os descritos em [61] e [63], respectivamente, ambos, entretanto, usando a mesma etapa de pré-processamento (tal como descrita em [63]), por questão de uniformidade. Algoritmo II e Algoritmo IV são as versões modificadas de Algoritmo I e Algoritmo III, respectivamente, incorporando o cálculo analítico das amplitudes dos comandos.

Os resultados da extração dos comandos de frase e acento de várias sentenças enunciadas língua portuguesa (embora os algoritmos sejam adequados para qualquer linguagem) estão resumidos nas Tabelas 5.3 e 5.4. As frases utilizadas são as listadas na Tabela 5.1. Estas frases enunciadas em diversos modos sentenciais estão numeradas na Tabela 5.2. Na Tabela 5.3, números em negrito indicam o menor erro médio quadrático atingido com a sentença dada. Pode-se notar, comparando os valores da função objetivo ε^2 em cada caso, como o processo analítico melhora consistentemente a performance dos algoritmos de estimação. Também podemos observar, da Tabela 5.4, como o procedimento aumenta a velocidade de convergência do algoritmo de inversão na maioria dos casos.

As Figuras 5.1, 5.3 e 5.5 mostram os contornos de F0 obtidos pelos quatro algoritmos para as Sentenças 6, 1 e 2, respectivamente. Nesta figura, a linha sólida representa o contorno de F0 suavizado ideal, a linha tracejada indica o contorno estimado para cada algoritmo original (Algoritmo I nas Figuras 5.1(a), 5.3(a) e

Tabela 5.1: Frases.

F1:	A matéria do jornal foi bastante discutida.
F2:	O meu chefe foi almoçar com o presidente da empresa.
F3:	Brasília foi construída por Juscelino.

Tabela 5.2: Enunciados utilizados na experiência.

Número da Sentença	Frase	Modo Sentencial
1	F1	Declarativa
2	F2	Declarativa
3	F3	Declarativa
4	F1	Interrogativa
5	F2	Interrogativa
6	F3	Interrogativa
7	F1	Exclamativa
8	F2	Exclamativa
9	F3	Exclamativa

5.5(a) e Algoritmo III nas Figuras 5.1(b), 5.3(b) e 5.5(b)), e a linha pontilhada representa o contorno representado pelos algoritmos modificados (Algoritmo II nas Figuras 5.1(a), 5.3(a) e 5.5(a) e Algoritmo IV nas Figuras 5.1(b), 5.3(b) e 5.5(b)) com cálculo analítico das amplitudes dos comandos. As Figuras 5.2, 5.4 e 5.6 ilustram os comandos de Fujisaki determinados por cada algoritmo. Tanto as Figuras 5.1, 5.3 e 5.5 como as Figuras 5.2, 5.4 e 5.6 dão uma idéia qualitativa sobre como o cálculo analítico proposto das amplitudes dos comandos resulta em melhores resultados. Tal melhoramento é muito certamente devido a uma estimativa inicial mais precisa dos parâmetros, evitando mínimos locais não apropriados.

Tabela 5.3: Valor final da função objetivo ε^2 obtida por algoritmos de estimação distintos com/sem cálculo analítico das amplitudes. Números em negrito representam o melhor resultado para uma dada sentença.

Sentença	Algoritmo I	Algoritmo II	Algoritmo III	Algoritmo IV
1	31.133×10^{-3}	6.685×10^{-3}	10.278×10^{-3}	$\mathbf{0.279 \times 10^{-3}}$
2	77.270×10^{-3}	7.236×10^{-3}	6.449×10^{-3}	$\mathbf{1.783 \times 10^{-3}}$
3	22.443×10^{-3}	7.790×10^{-3}	5.556×10^{-3}	$\mathbf{2.030 \times 10^{-3}}$
4	33.650×10^{-3}	35.805×10^{-3}	63.920×10^{-3}	$\mathbf{4.301 \times 10^{-3}}$
5	56.049×10^{-3}	26.411×10^{-3}	34.181×10^{-3}	$\mathbf{1.347 \times 10^{-3}}$
6	53.661×10^{-3}	12.750×10^{-3}	77.881×10^{-3}	$\mathbf{8.980 \times 10^{-3}}$
7	26.840×10^{-3}	14.178×10^{-3}	3.370×10^{-3}	$\mathbf{2.067 \times 10^{-3}}$
8	9.478×10^{-3}	16.318×10^{-3}	59.643×10^{-3}	$\mathbf{6.401 \times 10^{-3}}$
9	77.665×10^{-3}	26.805×10^{-3}	$\mathbf{7.331 \times 10^{-3}}$	9.063×10^{-3}

5.6 Conclusão

Este capítulo apresenta o problema de inversão do modelo de Fujisaki. Este problema tem sido alvo de pesquisas, e neste capítulo é apresentado um novo método que resolve parcialmente o problema, analiticamente.

Na Seção 5.2 é feito um resumo de alguns algoritmos de inversão, apresentando as principais idéias empregadas para obter os comandos a partir do contorno de F0, como a detecção de pontos críticos da curva de F0 para determinar as posições dos comandos e separação dos componentes de acento e de frase, considerando que o componente de frase está relacionado com as variações lentas na curva de F0 e o de acento com as varições rápidas. Vemos que os algoritmos de inversão possuem três etapas: pré-processamento, estimativa inicial e otimização. São citadas técnicas singulares utilizadas em alguns algoritmos, que podem melhorar a estimativa da posição dos comandos.

A Seção 5.3 mostra o procedimento analítico para obter as amplitudes dos comandos analiticamente, tendo-se a posição dos mesmos. É apresentado, na Seção 5.4 um novo algoritmo para estimação automática dos parâmetros de amplitude

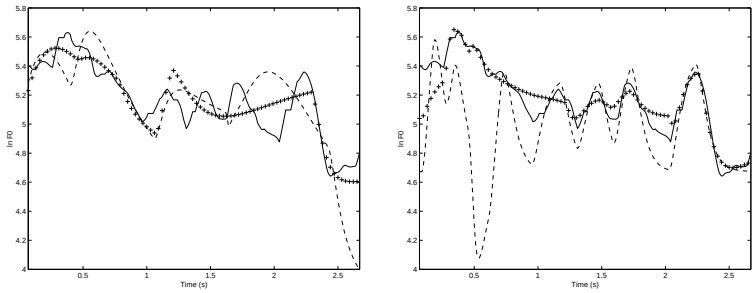
Tabela 5.4: Tempo de CPU em segundos para diferentes algoritmos de estimação com/sem cálculo analítico das amplitudes. Números em negrito indicam os melhores resultados para uma dada sentença.

Sentença	Algoritmo I	Algoritmo II	Algoritmo III	Algoritmo IV
1	**1.18**	12.34	2.00	1.76
2	**0.73**	2.93	1.59	1.59
3	0.66	**0.51**	1.83	5.57
4	0.90	**0.33**	1.74	0.54
5	0.61	**0.30**	1.27	0.86
6	**0.51**	3.14	1.01	0.68
7	**0.67**	1.28	4.71	2.16
8	0.94	**0.36**	2.88	1.15
9	8.78	**0.51**	1.32	5.83

do modelo de Fujisaki. O algoritmo proposto usa o procedimento analítico para determinar as amplitudes dos comandos do modelo. As posições dos comandos são estimativas inicialmente usando pontos críticos da curva, e são melhoradas passo a passo através de um procedimento de otimização iterativo.

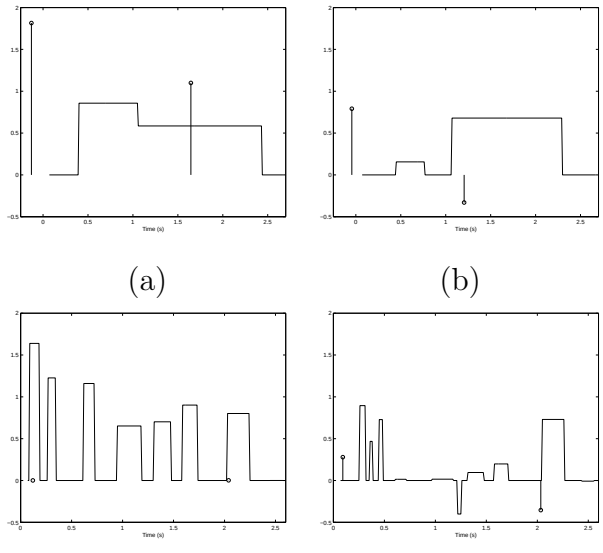
O procedimento analítico é adicionado a dois algoritmos de inversão, e a performance dos algoritmos é medida através do erro médio quadrático e do tempo de convergência da otimização. Os resultados são apresentados na Seção 5.5. O algoritmo, na maioria dos casos, resulta em um erro médio quadrático menor e reduzido custo computacional.

Além de auxiliar na convergência da otimização, o método proposto separa o problema de determinação das amplitudes do problema de determinar a posição dos comandos. O método faz com que o problema de inversão se reduza ao problema de encontrar as posições dos comandos. As experiências com algoritmos de inversão mostram que obter as posições não é trivial, e sugere-se o uso das técnicas listadas na Seção 5.2 para facilitar a obtenção dos instantes dos comandos.



(a) (b)

Figura 5.1: Contorno ideal (linha sólida) e contornos estimados para a Sentença 6: (a) Algoritmo I (linha tracejada) e Algoritmo II ('x'); (b) Algoritmo III (linha tracejada) e Algoritmo IV ('x').



(a) (b)

(c) (d)

Figura 5.2: Comandos do modelo de Fujisaki extraídos para a Sentença 6: (a) Algoritmo I; (b) Algoritmo II; (c) Algoritmo III; (d) Algoritmo IV.

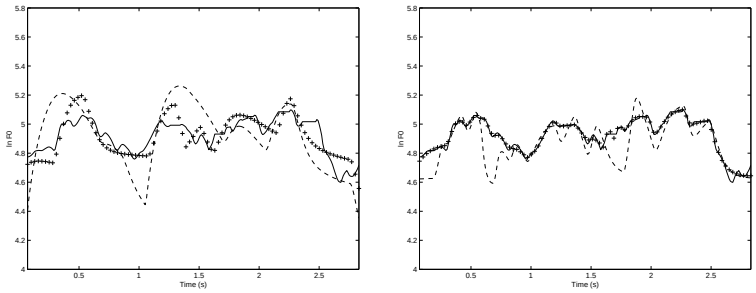


Figura 5.3: Contorno ideal (linha sólida) e contornos estimados para a Sentença 1: (a) Algoritmo I (linha tracejada) e Algoritmo II ('x'); (b) Algoritmo III (linha tracejada) e Algoritmo IV ('x').

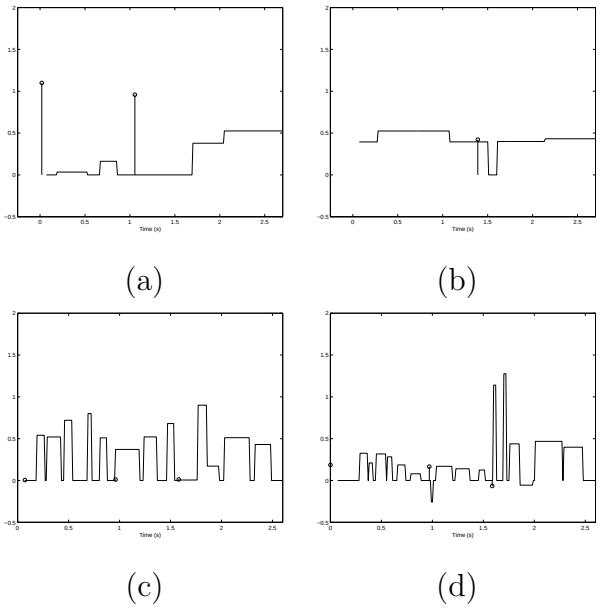


Figura 5.4: Comandos do modelo de Fujisaki extraídos para a Sentença 1: (a) Algoritmo I; (b) Algoritmo II; (c) Algoritmo III; (d) Algoritmo IV.

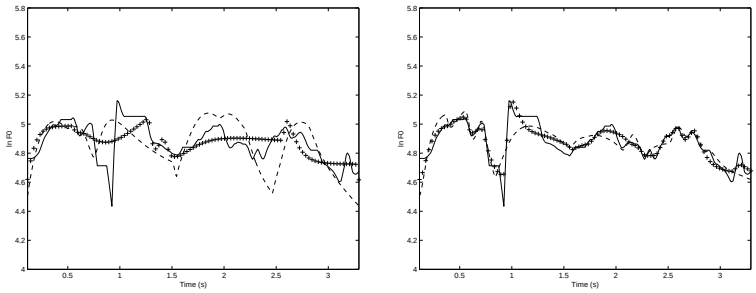


Figura 5.5: Contorno ideal (linha sólida) e contornos estimados para a Sentença 2: (a) Algoritmo I (linha tracejada) e Algoritmo II ('x'); (b) Algoritmo III (linha tracejada) e Algoritmo IV ('x').

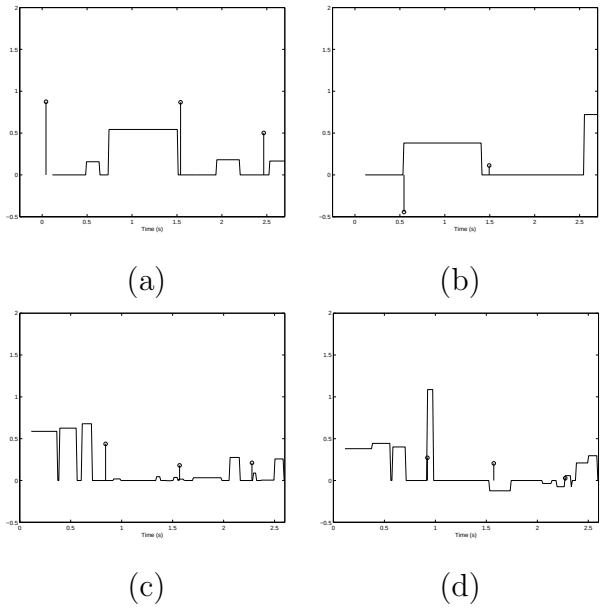


Figura 5.6: Comandos do modelo de Fujisaki extraídos para a Sentença 2: (a) Algoritmo I; (b) Algoritmo II; (c) Algoritmo III; (d) Algoritmo IV.

Capítulo 6

Conclusão

6.1 Conclusões finais

Neste trabalho foi feito um estudo dos sistemas TTS para o PB. A alta qualidade segmental é muito boa nos melhores sistemas comerciais disponíveis, e a voz em geral pode ser considerada bastante natural em algumas sentenças curtas. Ainda assim é percebida como artificial na maioria dos casos, devido à inacurácia dos modelos de prosódia. A constatação da artificialidade da voz sintética é clara principalmente no caso de textos longos.

Uma das dificuldades enfrentadas na modelagem da prosódia é a análise lingüística automática, que ainda não é muito confiável para informações lingüísticas de alto nível. Atualmente são empregadas, em geral, nos modelos de prosódia, apenas propriedades lingüísticas de fácil obtenção. Isto limita em muito a quantidade de informações lingüísticas que podem ser embutidas no enunciado sintetizado.

Também devemos considerar que o processo de integração das informações lingüísticas através de convenientes modificações dos parâmetros prosódicos não é facilmente modelado. Há uma tendência atual de que a modelagem seja realizada através de processos automáticos que inferem o modelo através de um conjunto de treinamento formado por exemplos do mapeamento.

Embora as informações lingüísticas de alto nível sejam incorporadas na voz através da modificação, às vezes sutil, de várias características acústicas, experiências realizadas pelo autor mostram que a modificação da duração dos fones e da freqüência fundamental ao longo do enunciado são suficientes para introduzir tais informações

no sinal de voz, sem perda de naturalidade.

Estas experiências mostram que, em geral, não é necessário prever a duração de cada fone do enunciado, mas somente a duração das sílabas. Outrossim, contornos de F0 extremamente simples (como as estilizações lineares de contornos naturais) podem gerar uma entonação bastante natural. Estes fatos nos indicam que temos grande flexibilidade na geração de prosódia.

Entretanto, foi constatado, com a implementação do algoritmo de modificação prosódica no TalkActive, que a imposição de durações para as sílabas e de um contorno de F0, ambos derivados de um enunciado natural, não melhoram significativamente a qualidade dos enunciados sintetizados. Isto acontece devido à baixa qualidade segmental do sistema. Os artifícios introduzidos pelo algoritmo de modificação prosódica são outra fonte de perda de qualidade da voz sintetizada.

Foi escolhido o modelo de Fujisaki neste trabalho, pois tal modelo tem parâmetros de entrada que podem ser relacionados com a prosódia de alto nível: a função de segmentação pode ser associada aos comandos de frase, e a função que cria uma oposição do tipo fraco/forte entre as palavras, aos comandos de acento. Esta correspondência lingüística facilita a obtenção do mapeamento do texto nos parâmetros prosódicos.

É implementado um módulo de prosódia extremamente simples, usando apenas a acentuação léxica para gerar os comandos do modelo de Fujisaki. Mesmo com um módulo tão simples, percebemos uma diferença significativa entre o enunciado com contorno de F0 uniforme e aquele que é sintetizado usando o modelo de Fujisaki, com um comando de frase no começo do enunciado e comandos de acento cujos começo e fim coincidem com o começo e fim das sílabas tônicas.

Uma das maiores contribuições do trabalho é um estudo dos algoritmos de inversão do modelo de Fujisaki. Graças à característica de sobreposição do modelo, a inversão é um problema de otimização complicado. O trabalho apresenta um novo procedimento para determinar as amplitudes dos comandos do modelo analiticamente. São apresentados os resultados de testes que demonstram que o tempo de processamento e o erro da aproximação são, geralmente, menores quando o procedimento analítico é utilizado antes da fase de otimização de algoritmos de inversão.

6.2 Construção de um módulo de prosódia

Neste trabalho foi implementado um módulo de prosódia, usando o modelo de Fujisaki e análise lingüística simples, detectando apenas a posição do acento léxico. Mesmo este módulo simplificado já representa um grande diferencial em relação à síntese com F0 uniforme. Entretanto, a qualidade ainda pode ser aumentada. Temos a seguir uma lista das etapas que devem ser concluídas para melhorar um módulo de prosódia baseado no modelo de Fujisaki:

1. Construção de um corpus anotado, com a anotação lingüística de alto nível (sintaxe e morfologia, por exemplo), baixo nível (segmentação fonética), e prosódica dos enunciados (usando ToBI, por exemplo). Este corpus seria usado para obter, através de um procedimento de treinamento, o mapeamento que leva o texto nas informações prosódicas;

2. Desenvolver um sistema de análise sintático-morfológica. Este sistema mapeia o texto em informação lingüísticas de alto nível. Sabe-se que as fronteiras de constituintes prosódicos estão relacionadas com a sintaxe do enunciado [45]. Logo, para obter estas fronteiras automaticamente, um módulo de análise sintática é necessário, a princípio. Por outro lado, como o algoritmo mais simples de determinação de fronteiras de constituintes prosódicos, o *chinks 'n' chunks* [3], só precisa do conhecimento sobre classes gramaticais para descobrir se uma palavra é de contexto ou função, então uma busca em um léxico e a detecção de sufixos podem ser suficientes, em alguns casos, para obter as características prosódicas do enunciado. O uso de informações lingüísticas de fácil obtenção é importante, já que a análise sintático-morfológica automática não é confiável;

3. Obter, através do corpus anotado, o mapeamento que leva as informações lingüísticas de alto nível (como a derivada da análise sintático-morfológica) na prosódia do enunciado (as fronteiras de constituintes prosódicos e acentuação);

4. Associar a prosódia de alto nível com os parâmetros do modelo de Fujisaki. Estes parâmetros poderiam ser obtidos a partir do contorno de F0 manualmente ou através de um procedimento automático ou semi-automático de inversão.

A princípio, podemos dizer que as fronteiras entre constituintes prosódicos estão relacionadas com as posições dos comandos de frase, e a acentuação está relacionada com as posições dos comandos de acentos. Entretanto, é possível que esta relação não seja tão direta;

5. É preciso obter, também, o mapeamento que retorna as posições e amplitudes dos comandos do modelo de Fujisaki a partir das informações prosódicas de alto nível, derivadas do texto.

6.3 Sugestões de trabalhos futuros

O presente trabalho sugere várias linhas de pesquisa. Por exemplo, não é analisada em detalhes a relação entre os comandos do modelo de Fujisaki e a prosódia de alto nível. Portanto, revestir-se-ia de grande importância qualquer pesquisa no sentido de comprovar a relação direta entre a entrada do modelo e as funções de segmentação e de acentuação da prosódia.

Melhorar a detecção dos instantes em que são aplicados os comandos do modelo também é importante, pois o aumento da confiabilidade dos algoritmos de inversão ensejaria a construção de um corpus anotado contendo os comandos do modelo, extraídos automaticamente. Se houver uma relação simples entre as posições e amplitudes dos comandos e as posições e tipos de fronteiras ou acentos em um sistema de anotação prosódica, como o ToBI, um algoritmo de inversão confiável possibilitaria a anotação prosódica automática dos enunciados.

Podemos destacar as seguintes possibilidades de trabalhos futuros:

1. Aumentar a confiabilidade dos algoritmos de inversão do modelo de Fujisaki. Uma das principais dificuldades da inversão é a determinação dos instantes dos comandos, já que isto se reduz a um problema de otimização não linear com vários mínimos locais. Foram sugeridas, no Capítulo 5, algumas formas de melhorar a extração dos parâmetros do modelo, baseadas em outros algoritmos proposto na literatura;

2. Construção de um corpus anotado, contendo anotações, pelo menos, do texto dos enunciados, da segmentação fonética e dos comandos do modelo de Fujisaki.

A construção deste corpus será relativamente simples tendo à disposição um algoritmo de segmentação fonética automática confiável, bem como um algoritmo de inversão confiável. Pode-se utilizar a disponibilidade do texto para orientar estes algoritmos;

3. A partir de um corpus anotado, podemos derivar algoritmos para determinar as fronteiras de constituintes prosódicos e a acentuação, a partir do texto. Existindo uma relação direta entre a posição dos comandos do modelo de Fujisaki e a posição das fronteiras e a acentuação, os comandos do modelo, extraídos com um algoritmo de inversão, poderão ser usados no lugar de uma anotação prosódica manual para obter, mediante treinamento, o mapeamento que leva texto na prosódia de alto nível;

4. É possível que exista uma relação simples entre as amplitudes dos comandos e os tipos de fronteiras ou acentos na anotação prosódica usando ToBI, por exemplo. Neste caso, poderíamos fazer a anotação prosódica automática através de algoritmos de inversão do modelo de Fujisaki;

5. A segmentação fonética automática torna possível determinar em que posição do texto são aplicados os comandos do modelo de Fujisaki. Também possibilita determinar a duração dos fones e das sílabas, o que serviria para derivar um modelo de duração através de treinamento. Existe uma tendência de encontrar algoritmos capazes de fazer a segmentação automática através de técnicas simples, como a detecção do ataque de plosivas, detecção da sonoridade (se o segmento é surdo ou sonoro), detecção de fricativas etc. Por outro lado, estes detectores podem ser usados para melhorar a performance de um algoritmo de segmentação baseado em HMM;

6. A segmentação fonética automática a partir de texto conhecido necessitaria de um módulo de conversão grafema-fonema, que deveria ser implementado.

Referências Bibliográficas

[1] WITTEN, I. H., *Principles of Computer Speech*. Academic Press, Inc, 1982.

[2] CALLOU, D., LEITE, Y., *Iniciação à Fonética e à Fonologia*. Jorge Zahar Editor Ltda, 1990.

[3] DUTOIT, T., *An Introduction to Text-to-Speech Synthesis*. Kluwer Academic Publishers, 1996.

[4] PRADO, P. P. L. d., "Sintetizador Articulatório de Voz: Mapeamento Acústico / Articulatório". In: *11o Simpósio Brasileiro de Telecomunicações*, v. II, Natal, RN, Setembro 1993.

[5] DELLERI, J. R., PROAKIS JR., J. G., HANSEN, J. H. L., *Discrete-time processing of speech signals*. Macmillan Publishing Company, 1993.

[6] HARRIS, C., "A study of the building blocks of speech", *Journal of the Acoustic Society of America*, v. 25, pp. 962–969, 1953.

[7] SILVA, C. H. d., *Modelamento Prosódico para Conversão Texto-Fala do Português Falado no Brasil*. M.Sc. dissertation, Universidade Estadual de Campinas (UNICAMP), Dezembro de 1995.

[8] RABINER, L., JUANG, B.-H., JUANG, B.-H., *Fundamentals of Speech Recognition*. 1 ed. Pearson Education POD, 1993.

[9] DONOVAN, R., *Trainable Speech Synthesis*. Ph.D. dissertation, Cambridge Univ. Eng. Dept., June 1996.

[10] YOUNG, S., et al., "HTK Manual", Entropic, 1996.

[11] BARBOSA, P. A., et al., "Aiuruetê: A High-Quality Concatenative Text-to-Speech System for Brazilian Portuguese with Demisyllabic Analysis-Based Units and a Hierarchical Model of Rhythm Production". In: *Proceedings of Eurospeech*, v. 5, pp. 2059–2062, 1999.

[12] ALBANO, E. C., MOREIRA, A. A., "Archisegment-based Letter-to-Phone Conversion for Concatenative Speech Synthesis in Portuguese". In: *Proc. ICSLP '96*, v. 3, pp. 1708–1711, Philadelphia, PA, 1996.

[13] SIMÕES, F. O., *Implementação de um Sistema de Conversão Texto-Fala para o Português do Brasil*. M.Sc. dissertation, Universidade Estadual de Campinas (UNICAMP), maio 1999.

[14] CAMPBELL, N. W., "Syllable-based segmental duration". In: G. Bailly, C. B., Sawallis, T. R. (eds.), *Talking Machines: Theories, Models, and Designs*, Elsevier Science Publishers B.V., 1992.

[15] BARBOSA, P. A., "Revelar a estrutura rítmica de uma língua construindo máquinas falantes: pela integração de ciência e tecnologia de fala". In: *Estudos de Prosódia*, IEL/UNICAMP, Editora da UNICAMP, pp. 21–52, 1999.

[16] MicroPower Software, http://www.micropower.com.br.

[17] Projeto MBROLA, http://tcts.fpms.ac.be/synthesis/mbrola.html.

[18] FRANCO, A. R., "Sintetizador Paramétrico de Voz para a Língua Portuguesa", Escola de Engenharia da UFRJ, Departamento de Eletrônica, 1999, Projeto Final.

[19] DIGALO, http://www.digalo.com.

[20] DOSVOX, http://caec.nce.ufrj.br/ dosvox/index.html.

[21] Elan Informatique, http://www.elantts.com/accueil.html.

[22] ScanSoft inc., http://www.scansoft.com.

[23] GRIFFIN, D. W., LIM, J. S., "Multiband Excitation Vocoder", *IEEE Transactions on Acoustic Speech and Signal Processing*, v. 36, n. 8, pp. 1223–1235, August 1988.

[24] DUTOIT, T., "High Quality Text-To-Speech Synthesis: A Comparison of Four Candidate Algorithms", *Proceedings of the ICASSP*, pp. 565–568, 19-22 april 1994.

[25] CARDON, A., *Um Sistema de Síntese de Fala através da Concatenação de Sílabas*. M.Sc. dissertation, Programa de Pós-Graduação em Computação da UFRGS, 2000.

[26] TEXTO-FALA, http://www.cpqd.br/produtos/textofala/.

[27] NAGLE, E., Comunicação Pessoal.

[28] QUENÉ, H., KAGER, R., "The derivation of prosody for text-to-speech from prosodic sentence structure", *Computer Speech and Language*, , n. 6, pp. 77–98, 1992.

[29] SILVA, C. H. d., et al., "Parsing Prosódico para Modelamento da Prosódia na Conversão Texto-Fala do Português". XV Simpósio Brasileiro de Telecomunicações, 1997.

[30] SILVA, C. H. d., et al., "F0 Generation in a Text-to-Speech System, Using a Database of Natural F0 Patterns". ITS'98, São Paulo, Brasil, 1998.

[31] Dicionário Michaelis online, http://www.uol.com.br/michaelis/.

[32] BECHARA, E., *Moderna Gramática Portuguesa*. Editora Lucerna, Rio de Janeiro, 2001.

[33] DOGIL, G., MÖBIUS, B., "Towards a model of target oriented production of prosody". In: *European Conference on Speech Communication and Technology*, v. 1, pp. 665–668, Aalborg, Denmark, 2001.

[34] MIXDORFF, H., *Intonation Patterns of German - Model-based Quantitative Analysis and Synthesis of F0 contours*. Ph.D. dissertation, Technische Universität Dresden, Maio de 1998.

[35] FACKRELL, J., et al., "Prosodic Variation with Text Type". In: *Proceedings of ICSLP*, 2000.

[36] KLEIJN, W. B., PALIWAL, K. K., *Speech Coding and Synthesis*. Elsevier Science B.V., 1995.

[37] LEHISTE, I., *Suprasegmentals*. Cambridge Mass.: MIT Press, 1970.

[38] FEBRER, M., et al., "Aneto: A Tool for Prosody Analysis of Speech". In: *First COST-G6 Workshop on Digital Audio Effects (DAFX98)*, Barcelona, Spain, November 1998.

[39] BAILLY, G., AUBERGÉ, V., "Phonetic and phonological representations for intonation". In: Jan P. H. van Santen, Richard W. Sproat, J. P. O., Hirschberg, J. (eds.), *Progress in Speech Synthesis*, Springer Verlag, New York, 1996.

[40] UMEDA, N., "Linguistic Rules for Text-to-Speech Synthesis", *Proceedings of the IEEE*, v. 64, n. 4, pp. 443–451, april 1976.

[41] LADEFOGED, P., *Elements of Acoustic Phonetics*. 2 ed. University of Chicago Press, 1996.

[42] BAILLY, G., "No future for comprehensive models of intonation?" In: Y. Sagisaka, N. C., Higuchi, N. (eds.), *Computing prosody: Computational models for processing spontaneous speech*, Springer Verlag, pp. 157–164, 1997.

[43] BULYKO, I., OSTENDORF, M., "Joint prosody prediction andunit selection for concatenative speech synthesis". In: *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, v. II, pp. 781–784, Salt Lake City, Utah, 2001.

[44] FREITAS, D., Comunicação Pessoal.

[45] FROTA, S., *Prosody and Focus in European Portuguese*. Ph.D. dissertation, Faculdade de Letras da Universidade de Lisboa, 1998.

[46] SANTEN, J. P. H. v., "Deriving text-to-speech durations from natural speech". In: G. Bailly, C. B., Sawallis, T. R. (eds.), *Talking Machines: Theories, Models, and Designs*, Elsevier Science Publishers B.V., pp. 275–285, 1992.

[47] HIRST, D., ESPESSER, R., "Automatic Modelling of Fundamental Frequency using a Quadratic Spline Function", *Travaux de l'Institut de Phonétique d'Aix*, v. 15, pp. 75–85, 1993.

[48] TAYLOR, P., "The rise/fall/connection model of intonation", *Speech Communication*, v. 15, pp. 169, 1994.

[49] TAYLOR, P., "Analysis and Synthesis of Intonation using the Tilt Model", *Journal of the Acoustical Society of America*, v. 107, n. 3, pp. 1697–1714, 2000.

[50] MÖHLER, G., CONKIE, A., "Third International Workshop on Speech Synthesis". Jenolan Caves, Australia, 1998.

[51] FUJISAKI, H., LJUNGQVIST, M., MURATA, H., "Analysis and modeling of word accent and sentence intonation in swedish", *Proceedings of the ICASSP*, v. II, pp. 211–214, 1993.

[52] SILVA, S. d. S., "Sistema de Conversão Texto-Fala Usando Unidades Silábicas", Escola de Engenharia da UFRJ, Departamento de Eletrônica, dezembro 2001, Projeto Final.

[53] MOULINES, E., CHARPENTIER, F., "Pitch Synchronous Waveform Processing Techniques for Text-to-Speech Synthesis using Diphones", *Speech Communication*, , n. 9, pp. 453–467, 1990.

[54] SANTOS, G. F. M. d., "Síntese de Voz Síncrona com o Período Fundamental Baseado na Concatenação de Difones", Faculdade de Engenharia da Universidade do Porto, Departamento de Engenharia Eletrotécnica e de Computadores, 1999.

[55] HAMON, C., MOULINES, E., CHARPENTIER, F., "A Diphone Synthesis System Based on Time-Domain Modifications of Speech", *Proceedings of the ICASSP*, pp. 238–241, 1989.

[56] H. FUJISAKI, K. H., "Analysis of voice fundamental frequency contours for declarative sentences of Japanese", *Journal of Acoustic Society Japan*, v. 5, n. 4, 1984.

[57] FUJISAKI, H., NARUSAWA, S., "Automatic extraction of model parameters from fundamental frequency contours of speech". In: *Proc. 2nd Plenary Meeting and Symp. Prosody and Speech Processing*, pp. 133–138, Tokyo, Japan, January 2002.

[58] FUJISAKI, H., OHNO, S., "Prosodic parameterization of spoken Japanese based on a model of the generation process of F0 contours". In: *Proc. Int. Conf. Spoken Languages Processing*, v. 4, pp. 2439–2442, Philadelphia, PA, 1996.

[59] GUTIÈRREZ-ARRIOLA, J., et al., "New rule-based and data-driven strategy to incorporate Fujisaki's F0 model to a text-to-speech in Castillian Spanish". In: *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Salt Lake City, Utah, 2001.

[60] SAKURAI, A., HIROSE, K., "Detection of phrase boundaries in Japanese by low-pass filtering of fundamental frequency contours". In: *Proc. Int. Conf. Spoken Languages Processing*, Sydney, Australia, 1998.

[61] MIXDORFF, H., "A novel approach to the fully automatic extraction of Fujisaki model parameters". In: *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 1281–1284, Istanbul, Turkey, 2000.

[62] NAKAI, M., SHIMODAIRA, H., "The use of F0 reliability function for prosodic command analysis on F0 contour generation model". In: *Proc. Int. Conf. Spoken Languages Processing*, Sydney, Australia, 1998.

[63] ROSSI, P. S., PALMIERI, F., CUTUGNO, F., "A method for automatic extraction of Fujisaki-model parameters". In: *Proc. Speech Prosody*, pp. 615–618, Aix-en-Provence, France, April 2002.

[64] GEOFFROIS, E., "A pitch contour analysis guided by prosodic event detection". In: *Proc. European Conf. Speech Communication and Technology*, v. 2, pp. 793–796, 1993.

[65] KRUSCHKE, H., KOCH, A., "Parameter extraction of a quantitative intonation model with wavelet analysis and evolutionary optimization". In: *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Hong Kong, 2003.