UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ELETRÔNICA

# SISTEMA DE CONVERSÃO TEXTO-FALA USANDO UNIDADES SILÁBICAS

Autor: Solimar de Souza Silva

Orientador: Fernando Gil Vianna Resende Junior

Orientador: Sérgio Lima Netto

Examinador: Márcio Nogueira de Souza

DEL
Março de 2001

# Agradecimentos

Ao meu pai e à minha mãe, por ajudarem a me tornar o que eu sou. Sem eles eu não existiria, e nem este projeto final.

Aos professores Fernando Gil Vianna Resende Junior e Sérgio Lima Netto, por terem me aceito como aluno de iniciação científica, e pela paciência que tiveram comigo desde então.

Ao aluno Verochile da Silva Junior, por ter feito parte do trabalho que eu deveria ter feito.

Solimar de Souza Silva

# Resumo

Este projeto consiste no desenvolvimento de ferramentas para a obtenção da base de dados de um sistema TTS concatenativo baseado em sílabas, bem como do módulo de concatenação do sistema, que se utiliza do algoritmo TD-PSOLA.

A sugestão de sílabas como unidades fundamenta-se no fato de que unidades maiores que o difone levam a uma quantidade menor de pontos de concatenação, aumentando, a princípio, a qualidade segmental do sistema em relação à qualidade de um sistema TTS baseado em difones.

Os resultados obtidos validam esta dedução. Também foram feitos testes que mostram a importância da utilização de um bom algoritmo de concatenação num sistema TTS concatenativo.

**Palavras-chave:**

Síntese de voz, TTS, TD-PSOLA, sílabas.

# Sumário

# Capítulo 1

# Introdução

Um sistema de conversão texto-fala é capaz de converter um texto arbitrário na representação acústica equivalente. Existem várias aplicações para um sistema TTS, como, por exemplo, interfaces de voz e auxílio a deficientes visuais.

Este trabalho descreve algumas etapas do projeto de um sistema de conversão texto-fala (text-to-speech, TTS) para a língua portuguesa falada no Brasil que usa a abordagem concatenativa e uma base de dados formada por sílabas. Serão descritas neste trabalho a ferramenta de recorte usada para a obtenção das unidades, as ferramentas e a implementação do módulo de concatenação do sistema, e os testes comparativos para se medir a qualidade do sistema obtido.

Neste capítulo, será apresentado um resumo de algumas das etapas de construção do sistema e do conteúdo dos próximos capítulos. A Seção 1.1 descreve as abordagens usadas em sistemas TTS. A Seção 1.2 apresenta a filosofia geral dos sistemas baseados na abordagem concatenativa, e na Seção 1.3 o sistema implementado é descrito a partir de um diagrama de blocos.

## 1.1 Abordagens usadas em um sistema TTS

As estratégias usadas para resolver o problema de converter texto em voz podem ser divididas em duas classes: síntese por regras e síntese por concatenação [1].

Na primeira, o sinal de voz é descrito por um conjunto de regras que são utilizadas para a composição da realização acústica dos enunciados. Na segunda, a informação necessária para a síntese é armazenada na forma de amostras numa base de dados de segmentos.

A qualidade segmental de um sistema TTS baseado na primeira estratégia

depende não só do modelo escolhido e da qualidade das amostras gravadas, mas também das regras encontradas. Encontrar as regras para um sintetizador deste tipo é um processo complicado e trabalhoso, pois as regras tem que ser ajustadas (por tentativa e erro) para obter voz sintética de alta qualidade [2].

Usando-se a segunda estratégia, torna-se mais simples obter um sistema TTS de alta qualidade segmental. Neste caso, a qualidade depende principalmente do algoritmo de concatenação usado, da qualidade das amostras gravadas e das unidades escolhidas [2].

## 1.2 Sistemas TTS concatenativos

Num sistema TTS concatenativo, a realização acústica dos enunciados é obtida a partir da concatenação das unidades contidas na base de segmentos.

Foi observado que a qualidade segmental de uma sentença sintetizada, ou seja, sua inteligibilidade, depende muito da preservação das transições entre fones [3][2]. Este fato levou à sugestão de unidades conhecidas como *difones*, que são formadas pelo trecho que vai da região central de um fone à região central do fone seguinte, preservando as transições [3].

Não é possível, entretanto, sintetizar certos aglomerados consonantais de forma convincente usando difones [4] (por exemplo: 'gra', 'bla'). Por outro lado, apesar dos difones exigirem pouco espaço de armazenamento, a atual tecnologia de armazenamento possibilita a utilização de unidades maiores.

Unidades maiores que difones podem apresentar melhor qualidade segmental, devido à presença de uma menor quantidade de pontos de concatenação. Neste projeto, optou-se pela escolha de sílabas, que são unidades que começam e terminam em regiões do sinal de voz em que a energia cai bastante [2]. A vantagem obtida na escolha destas unidades é a melhor qualidade segmental em relação aos difones, não só porque há menor quantidade de pontos de concatenação, mas também pelo fato de que a concatenação é menos crítica nas regiões em que a energia é mais baixa.

Existem, porém, alguns problemas com relação à utilização de sílabas que serão abordados no Capítulo 2.

Por fim, deve-se ressaltar que, devido às diferenças de contexto e à necessidade de introduzir prosódia no enunciado sintetizado, são usados algoritmos de concatenação para suavizar a transição de uma unidade para outra. Isto é feito através da modificação dos parâmetros nas fronteiras dos segmentos. A necessidade de introduzir prosódia no enunciado sintetizado é outra razão para serem usados estes algoritmos. O Capítulo 4 descreve o algoritmo de concatenação usado neste projeto, o TD-PSOLA. O TD-PSOLA foi escolhi-

do como algoritmo de concatenação por ser um dos mais conhecidos, o mais simples de implementar, e por resultar em boa qualidade segmental e boa naturalidade [2][5].

## 1.3 Sistema implementado

A Figura 1.1 contém um diagrama de blocos que ilustra as diversas etapas de desenvolvimento do sistema de conversão texto-fala. Segue uma descrição resumida dos blocos.

- Listagem das palavras que contêm as sílabas: este bloco representa a etapa de listagem de todas as sílabas da língua portuguesa contendo, para cada uma das sílabas, um exemplo de palavra que a contém. Esta etapa foi realizada por Rosana Costa de Oliveira [6], aluna da faculdade de letras da UFRJ.

- Gravação das palavras: as palavras listadas na etapa anterior foram gravadas em mídia digital. A gravação das palavras foi feita pelo Prof. Sérgio Lima Netto.

- Recorte das sílabas: as sílabas foram recortadas das palavras gravadas com a utilização de uma ferramenta de recorte manual. A etapa de recorte das sílabas foi realizada pelo aluno Verochile da Silva Junior.

- Implementação da ferramenta de recorte: nesta etapa, uma ferramenta de recorte foi desenvolvida para auxiliar no recorte das unidades. Esta ferramenta é descrita no Capítulo 2.

- Análise V/UV, pitch e pós-processamento: é necessário estimar alguns parâmetros que são utilizados na etapa de concatenação elaborada. Isto é feito nesta etapa. Esta etapa é descrita em maiores detalhes no Capítulo 3.

- Implementação do programa de extração das marcas de pitch: este bloco representa o desenvolvimento de um programa para a estimação dos parâmetros necessários para a concatenação elaborada.

- Banco de sílabas e marcas de pitch: nesta etapa, as sílabas pós-processadas e os parâmetros estimados são armazenados em uma base de dados para posterior utilização.

- Separação em sílabas e Busca no banco: o sistema de conversão texto-fala deve converter o texto em uma seqüência de unidades para o módulo de concatenação. Esta é a função destes dois blocos. Estes blocos não foram implementados.

- Concatenação simples: este bloco faz a concatenação direta das sílabas, sem nenhum processamento adicional.

- Concatenação elaborada: neste bloco, o sinal obtido na concatenação simples é modificado usando o algoritmo de concatenação descrito no Capítulo 4.

- Voz sintetizada: neste bloco, o sinal obtido na concatenação elaborada é convertido em sua realização acústica. Este bloco não foi implementado.

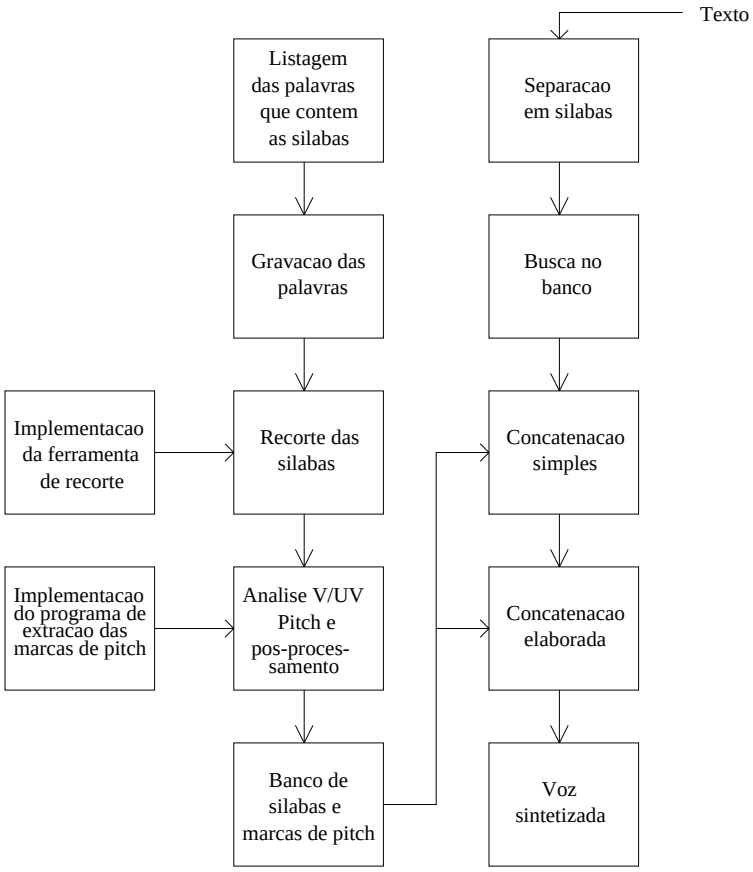


Figura 1.1: Desenvolvimento do sistema de conversão texto-fala.

## 1.4 Conclusão

Neste capítulo foi apresentada a filosofia de um sistema de síntese concatenativo. A utilização de sílabas e do algoritmo TD-PSOLA foi justificada pela possibilidade de maior qualidade segmental em relação a um sistema baseado em difones e devido à alta qualidade de voz permitida pelo algoritmo de concatenação. Foram descritas sucintamente as diversas etapas que compõem o sistema de conversão texto-fala.

# Capítulo 2

# Ferramenta de Recorte

## 2.1 Introdução

### 2.1.1 Recorte das unidades

Para a obtenção da base de dados, foram executadas as seguintes etapas:

1. Criação de uma lista contendo todas as sílabas da língua portuguesa e algumas palavras em que elas aparecem. Isto foi feito através de uma busca exaustiva em um dicionário. A lista das sílabas, contendo exemplos de palavras em que uma destas sílabas aparecem, é apresentada no Apêndice A.

2. Gravação, em mídia digital, das palavras em que ocorrem as realizações acústicas das sílabas.

3. Desenvolvimento de uma ferramenta de recorte. Inicialmente, desejava-se encontrar algum procedimento automático simples para o recorte das sílabas, o que não foi possível. A ferramenta auxilia no recorte manual das unidades.

4. Recorte manual das unidades. Foi utilizada a ferramenta de recorte descrita neste capítulo.

5. Obtenção das marcas de pitch para cada unidade da base de dados. Essa etapa é fundamental para o algoritmo de concatenação usado no sistema e é descrita em maiores detalhes no capítulo 3.

Deve ser notado o fato de que a definição de sílaba dada anteriormente é bastante subjetiva. De fato, não há uma referência que nos indique que a

energia caiu o bastante na fronteira das sílabas para que possa ser feita a segmentação num ponto. Por ser baseada em parâmetros acústicos, chamaremos este conceito de sílaba de "sílaba acústica". Por outro lado, é conveniente ter em mente o conceito de "sílabas gramaticais", que seriam as unidades que as pessoas que falam a língua portuguesa identificam normalmente como sílabas.

A ferramenta de recorte aqui descrita foi desenvolvida para facilitar o recorte dos segmentos silábicos a serem usados num sistema de conversão texto-fala. O programa foi desenvolvido usando a linguagem de programação do Matlab, e usando a ferramenta GUIDE [7] do Matlab para projetar a interface. Como esta é uma linguagem interpretada, o programa só pode ser executado de dentro do Matlab.

A ferramenta de recorte desenvolvida, em sua versão final, permite visualizar a forma de onda do sinal de fala, além da energia e da taxa de cruzamento pelo zero. Permite também reproduzir o som associado a uma determinada região selecionada ou gravá-lo em um arquivo.

As ferramentas de segmentação/transcrição normalmente armazenam as informações com relação à posição das fronteiras das regiões e as transcrições fonéticas em arquivos separados, ou no mesmo arquivo que contém as amostras do sinal de voz, podendo utilizar um formato padrão de arquivo, como o formato XML [8] ou o NIST sphere, por exemplo. Isto não só permite que a qualidade da segmentação e da transcrição possa ser avaliada posteriormente, como também a utilização das informações de segmentação e transcrição no treinamento de sistemas de reconhecimento de voz.

Esta, porém, não se trata de uma ferramenta de transcrição [9], ou seja, ela não permite associar regiões da forma de onda com a sua respectiva transcrição fonética. A ferramenta permite apenas a segmentação de um sinal de voz, através da separação de uma região contígua do sinal de voz das demais. Embora esta ferramenta não seja tão flexível ou poderosa quanto uma ferramenta de transcrição, experiências anteriores do autor com recorte de segmentos para um sintetizador por concatenação (sem a utilização da ferramenta) mostram que ela facilita bastante a obtenção dos segmentos.

## 2.2 Descrição da interface

A Figura 2.1 mostra a interface gráfica do programa de recorte. As funções das barras verticais vermelhas e verdes serão comentadas posteriormente.

Esta interface consiste em três eixos, à esquerda, onde são plotados os gráficos da forma de onda no tempo, da energia e da taxa de cruzamento

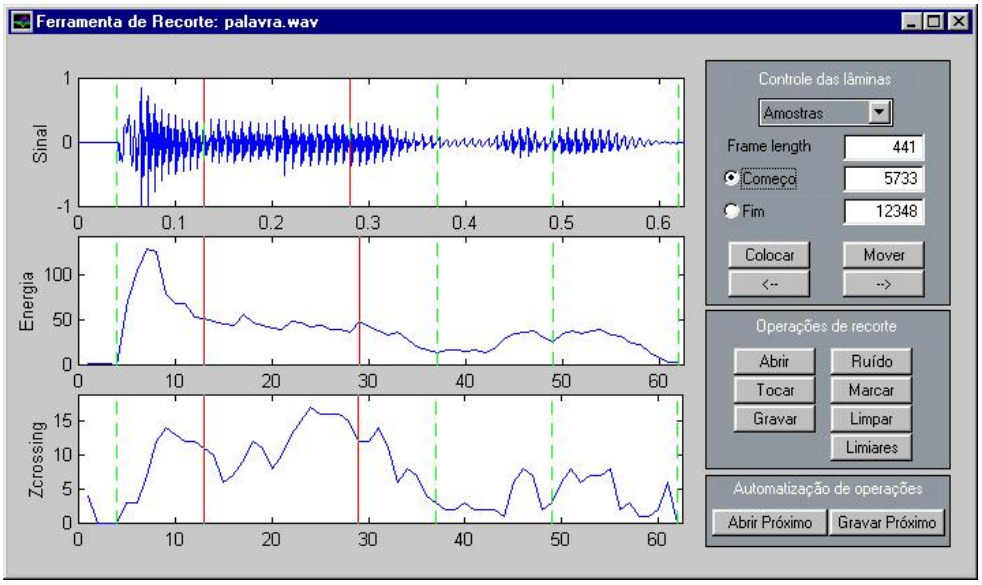


Figura 2.1: Interface gráfica do programa de recorte.

pelo zero, respectivamente, de cima para baixo. No gráfico da forma de onda no tempo o eixo das abscissas está em segundos, o nos outros gráficos, em quadros. A direita da janela estão três grupos de funções: "Controle das lâminas", "Operações de recorte", "Automatização de operações", descritos a seguir.

### 2.2.1 Controle das lâminas

As lâminas são as barras verticais vermelhas (contínuas) que estão relacionadas às amostras inicial e final dos segmentos a serem recortados. O quadro "Controle das lâminas" possui elementos de interface gráfica que permitem ao usuário inserir e controlar a posição das lâminas.

O quadro "Controle das lâminas" contém os seguintes elementos:

- Um menu "pop-up" que permite escolher o tipo de unidade (amostra ou quadro) a ser usada para a especificação da posição das lâminas nas caixas de edição.
- Uma caixa de edição na qual pode ser especificado o comprimento do quadro utilizado no cálculo da energia e da taxa de cruzamento pelo zero, em amostras.
- Dois "radio buttons", para selecionar a lâmina que será movimentada, a da esquerda (início) ou a da direita (fim do segmento).

- Duas caixas de edição que mostram as posições das lâminas, em quadros ou amostras, e que permitem movimentar as lâminas.
- Um botão "Colocar", que serve para inserir as duas lâminas.
- Os botões "Mover" e "$<--$", "$-->$", que servem para mover a lâmina anteriormente selecionada.

As lâminas são colocadas no gráfico com o botão "Colocar". As lâminas devem ser inseridas com o mouse clicando na posição em que se deseja colocá-la, no eixo que mostra a forma de onda no tempo. Para isto, o usuário pode se valer dos gráficos de energia e de taxa de cruzamento pelo zero, para fazer uma primeira estimativa do posicionamento das lâminas.

A primeira lâmina inserida representa a amostra inicial e a segunda representa a amostra final. Assim que as lâminas são inseridas, suas posições (em amostras ou quadros) são mostradas nas caixas de edição correspondentes. Em seguida, as lâminas podem ser movimentadas uma de cada vez. A lâmina que será movimentada é aquela selecionada através dos "radio buttons" e pode ser movimentada das seguintes formas:

- Usando o botão "Mover". A nova posição da lâmina devem ser escolhida usando o mouse, clicando-se na nova posição, no eixo que mostra a forma de onda no tempo.
- Usando as botões "$<--$" e "$-->$".
- Digitando-se a posição da lâmina na caixa de edição (em quadros ou amostras). Neste caso, a lâmina movimentada é aquela correspondente à caixa de edição usada.

### 2.2.2 Operações de recorte

Este quadro possui sete botões, organizados em duas colunas. Na coluna da esquerda, estão os botões associados diretamente com as operações de arquivos: "Abrir", "Tocar" e "Gravar". Na coluna da direita estão os botões associados com as "marcações".

As marcações, representadas nos gráficos como linhas verdes tracejadas, estão relacionadas com posições obtidas através de uma modificação do algoritmo de Rabiner-Sambur para encontrar os extremos de uma palavra isolada [10]. A rotina usada para obter os extremos foi retirada de um programa do aluno Ailton Dias Santana Junior.

As operações de arquivo permitidas são:

- O botão "Abrir" mostra uma caixa de diálogo através da qual podemos selecionar um arquivo. Depois de selecionado, este arquivo é carregado para a memória e os gráficos da forma de onda no tempo, energia e taxa de cruzamento pelo zero são automaticamente mostrados.

- O botão "Tocar" reproduz o som correspondente às amostras da região selecionada com as lâminas. Se as lâminas não foram colocadas, o arquivo inteiro é reproduzido.

- O botão "Gravar" abre uma caixa de diálogo na qual podemos digitar o nome do arquivo de saída. Depois de entrar com o nome do arquivo de saída, um arquivo com este nome é criado, no formato RIFF, contendo as amostras da região selecionada com as lâminas. Se as lâminas não foram colocadas, todas as amostras são gravadas.

Os botões associados às marcações são:

- "Ruído": Este botão permite a leitura de um arquivo que contém amostras do ruído de fundo, para obtenção da média e variância da energia e da taxa de cruzamento pelo zero. Ele mostra uma caixa de diálogo a partir da qual podemos escolher o arquivo a ser lido.

- "Marcar": Obtém a posição dos extremos, desenha as marcações no gráfico e cria uma janela na qual a posição de cada marcação é determinada, em amostras. Caso um arquivo com as amostras do ruído de fundo não tenha sido carregado anteriormente, o programa apresenta uma mensagem informando esta condição.

- "Limpar": Apaga as marcações desenhadas.

- "Limiares": Permite modificar os coeficientes que multiplicam a média e a variância da energia e da taxa de cruzamento pelo zero, de forma a alterar os limiares.

### 2.2.3 Automatização de operações

Este quadro tem os seguintes elementos:

- O botão "Abrir Próximo" - Abre o arquivo que sucede o último arquivo aberto na listagem do subdiretório deste.

- O botão “Gravar Próximo” - Grava um arquivo no formato RIFF que contém as amostras da região selecionada pelas lâminas. O nome do arquivo será o último nome usado na gravação usando o botão “Gravar”, sem a extensão, seguido de um hífen e um número, e da extensão “.wav”. O número começa de 1 (na primeira gravação com o botão “Gravar Próximo”) e é incrementado toda vez que um arquivo é salvo com o botão “Gravar Próximo”. Toda vez que um arquivo é salvo, uma caixa de mensagem é apresentada informando o nome do arquivo salvo.

## 2.3 Exemplo de recorte

Nesta seção, todas as etapas necessárias para o recorte de um segmento de voz serão descritas e ilustradas, passo a passo.

A primeira coisa que devemos fazer para recortar um segmento é abrir um arquivo que contenha o segmento. Para isto, apertamos o botão “Abrir” no quadro “Operações de recorte”. Isto faz aparecer a caixa de diálogo ilustrada na Figura 2.2, a partir da qual podemos escolher o arquivo. Neste caso, foi selecionado o arquivo “questão.wav”.


Figura 2.2: Caixa de diálogo para abrir arquivo.

Depois de selecionado o arquivo, os gráficos da forma de onda no tempo e da energia e taxa de cruzamento pelo zero são plotados, como mostra a Figura 2.3. O nome do arquivo aberto é mostrado na barra de título da janela.

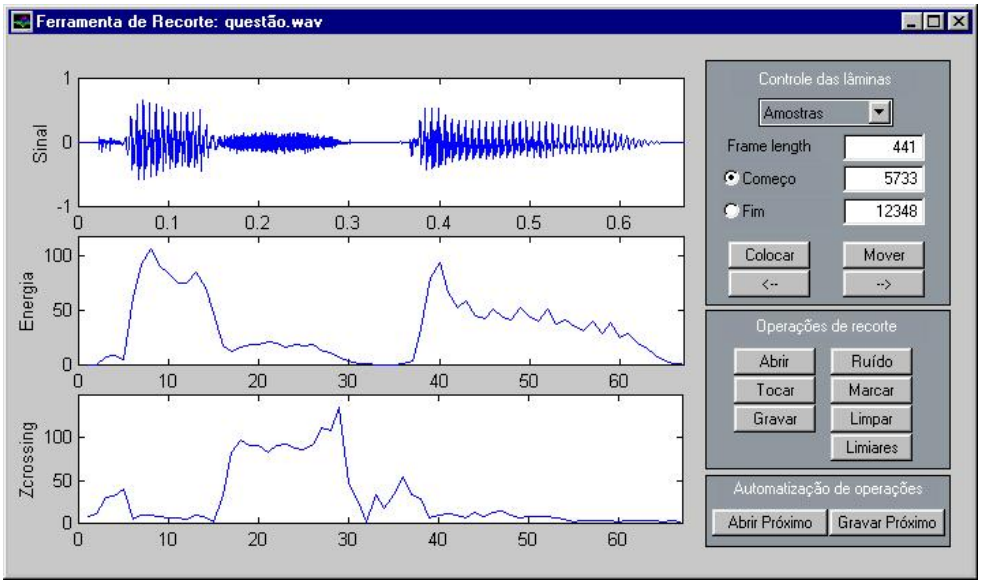


Figura 2.3: Arquivo "questão.wav" aberto pelo programa.

Para inserir as marcações, teríamos que carregar um arquivo contendo as amostras do ruído de fundo. Fazemos isto apertando o botão "Ruído", no quadro "Operações de recorte". Aparecerá então a caixa de diálogo mostrada na Figura 2.4. Neste caso, o arquivo "noise.wav" foi selecionado.

Agora, apertando o botão "Marcar" no quadro "Operações de recorte", as marcações são colocadas no gráfico. Uma janela adicional é criada. Esta janela contém a posição, em amostras, das marcações de início e fim de cada segmento encontrados. Neste caso, as Figuras 2.5 e 2.6 ilustram a janela principal e a criada, respectivamente. Para retirar as marcações, basta apertar o botão "Limpar" do quadro "Operações de recorte".

Os limiares utilizados para obtenção dos extremos podem ser modificados usando-se o botão "Limiares" do quadro "Operações de recorte". Apertando este botão, uma caixa de diálogo aparece, permitindo modificar os coeficientes multiplicadores das médias e variâncias da energia e da taxa de cruzamento pelo zero. A Figura 2.7 mostra a caixa de diálogo apresentada.

Para inserir as lâminas, devemos apertar o botão "Colocar" do quadro "Controle de lâminas". A forma do ponteiro de mouse é modificada para a forma de uma cruz, e clicando num ponto qualquer do gráfico da forma de onda no tempo, inserimos a lâmina que indica o começo da região. Logo a seguir, devemos clicar em outro ponto para inserir a lâmina que indica o final da região. As lâminas serão desenhadas e as posições destas serão mostradas nas caixas de edição no quadro "Controle de lâminas". A Figura 2.8 ilus-

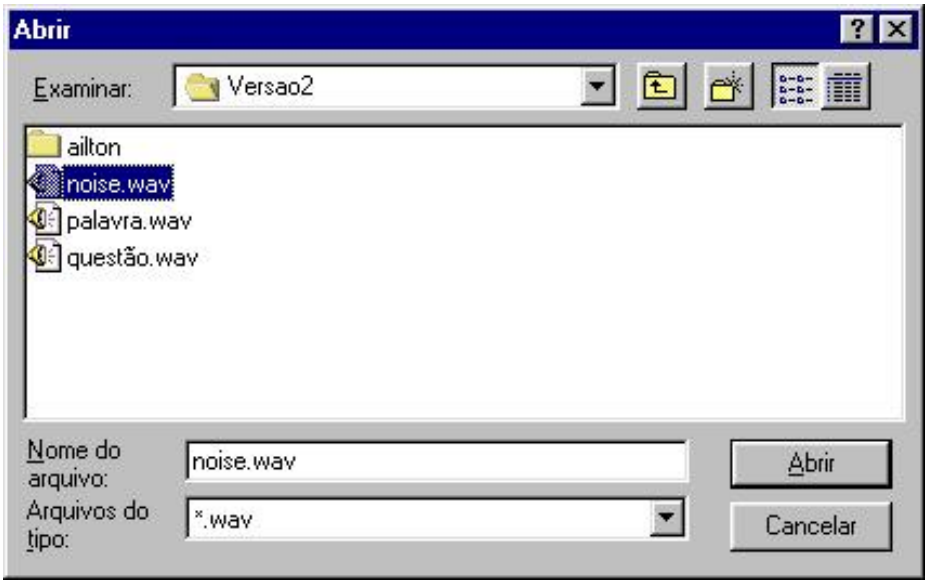


Figura 2.4: Caixa de diálogo para carregar o arquivo contendo amostras de ruído.

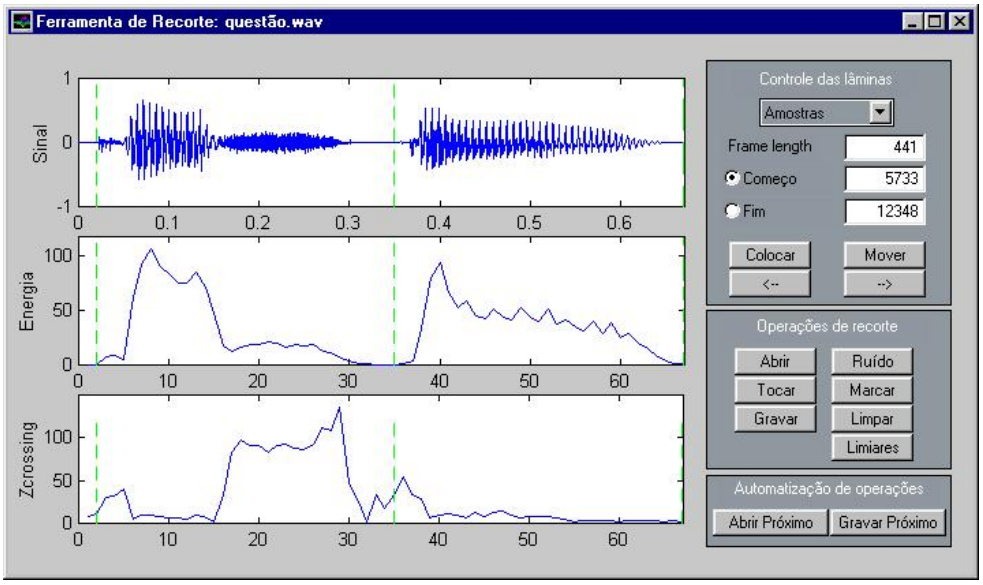


Figura 2.5: Janela principal depois de apertar o botão “Marcar”.


Figura 2.6: Janela criada mostrando as posições das marcações.

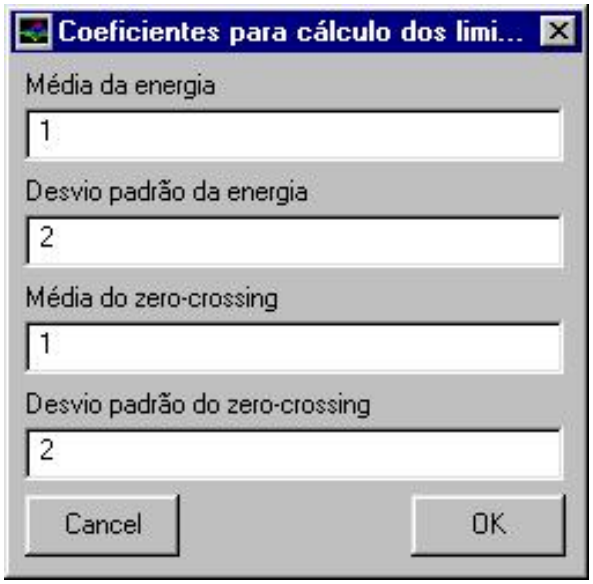


Figura 2.7: Modificação dos coeficientes para cálculo dos limiares para marcação.

tra as lâminas inseridas no gráfico para recortar o 's' da palavra "questão". Apertando o botão "Tocar" do quadro "Operações de recorte", ouviremos apenas o som associado a esta parte selecionada do gráfico.

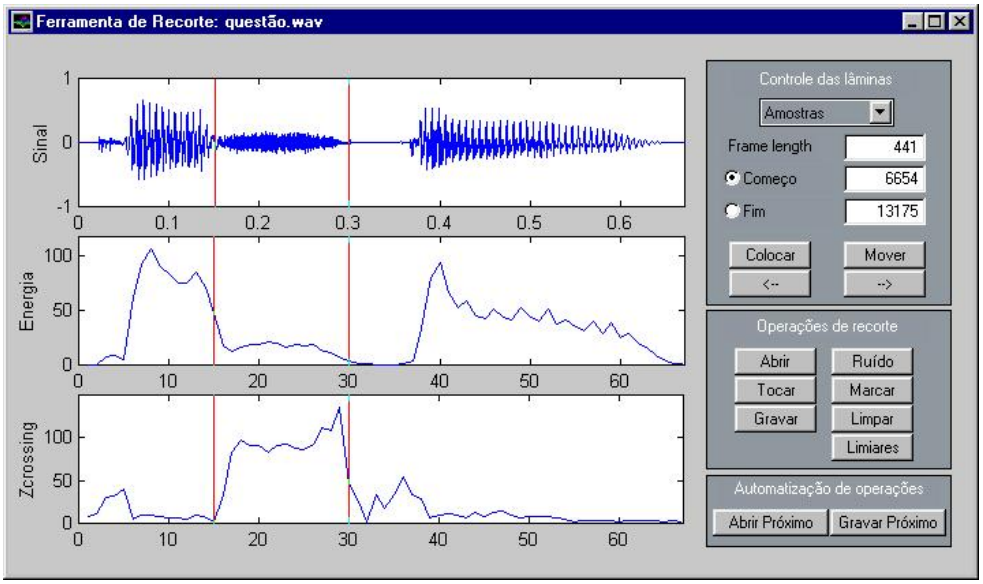


Figura 2.8: Lâminas inseridas para recortar o 's' de "questão".

Podemos movimentar as lâminas usando os botões "Mover", "$<--$", "$-->$" e as caixas de edição das lâminas, localizadas no quadro "Controle de lâminas". A Figura 2.9 mostra a lâmina de final após ter sido movimentada usando o botão "Mover".

Para gravar um arquivo que conterá apenas as amostras do 's' recortado da palavra, usamos o botão "Gravar" no quadro "Operações de recorte". A caixa de diálogo ilustrada na Figura 2.10 aparecerá, onde podemos digitar um nome para o arquivo a ser gravado.

## 2.4 Descrição do código

Como já foi dito anteriormente, o código foi escrito na linguagem do Matlab, com a ajuda do GUIDE. O programa é executado rodando o script recorte.m. O programa de recorte é formado pelos seguintes arquivos:

- cparm.m - Segmenta o arquivo em quadros, e retorna a energia e a taxa de cruzamento pelo zero dos quadros num vetor.
- energia.m - Calcula energia de um quadro.

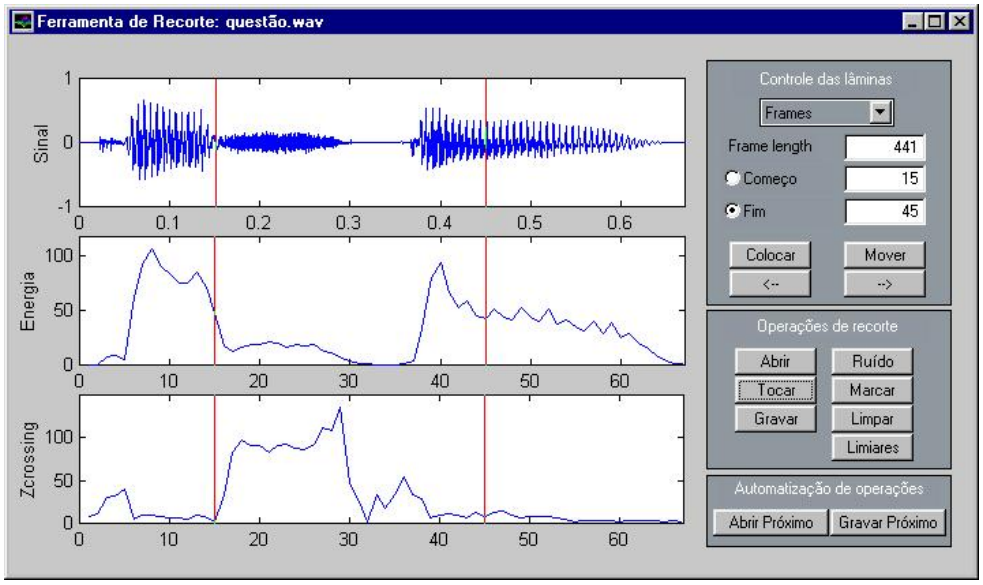


Figura 2.9: Lâmina de final movimentada usando o botão "Mover".

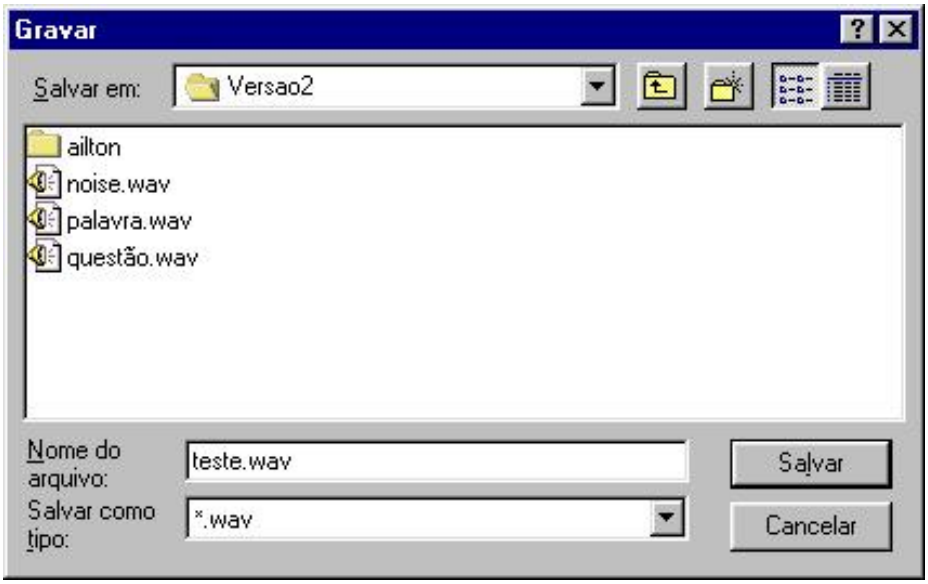


Figura 2.10: Caixa de diálogo para gravar arquivo.

- zcross.m - Calcula a taxa de cruzamento pelo zero de um quadro.
- xycatch.m - A função *ginput* um pouco modificada.
- recorte.m e recorte.mat - Arquivos criados pelo GUIDE. Contêm o código da interface.
- select.m - Chamada (*callback*) dos elementos da interface.

A chamada *CreateFcn* da janela principal inicializa várias variáveis que são utilizadas durante a execução das chamadas dos elementos da interface (botões, caixas de edição etc). A função select, contida em select.m, é usada como chamada para todos os elementos da interface. Isto é feito da seguinte maneira: a função select obtém a propriedade "Tag" do elemento da interface que chamou a função. Dependendo do valor desta propriedade, uma função conveniente, contida no arquivo select.m, é chamada. Esta função é escolhida a partir do valor desta propriedade através de uma estrutura de decisão de opções múltiplas (do tipo *switch-case*). Por exemplo, se o botão "Abrir" for pressionado, tendo a propriedade "Tag" o valor "Abrir", será chamada a função "Abre" contida no arquivo select.m.

Para que um novo elemento da interface seja adicionado, basta inserí-lo usando o GUIDE, modificar a propriedade "Callback" do elemento para "select;" e colocar um valor apropriado para a propriedade "Tag". Depois, deve-se adicionar à função select uma opção adicional, para comparar o valor da propriedade "Tag" com o valor escolhido para este elemento e chamar a função apropriada.

## 2.5 Conclusão

Este capítulo descreveu a ferramenta de recorte usada para obtenção das unidades. Experiências mostram que, apesar de sua simplicidade, a ferramenta realmente facilita o recorte das unidades. Foi apontada a maior flexibilidade das ferramentas de transcrição, que permitiriam até mesmo a fácil utilização da ferramenta no treinamento de sistemas de reconhecimento.

Já que o recorte foi manual, não há uma consistência na forma como foram escolhidos os pontos de segmentação. Em alguns casos, a energia não caía muito na fronteira entre sílabas gramaticais, tornando difícil a implementação de métodos automáticos simples para a segmentação, e fazendo com que a escolha do ponto de segmentação fosse muito subjetiva. Os pontos de segmentação foram sempre escolhidos nas fronteiras das sílabas gramaticais, mesmo quando a energia não caía muito entre elas.

Num sistema de conversão texto-fala, há também o problema de análise da "string" de entrada do sistema de conversão texto-fala, que ocorre em qualquer sistema desse tipo. Como podemos observar, tomando como exemplo a palavra "terra", a análise não consiste numa simples separação silábica usando as regras gramaticais normais (o que resultaria em "ter-ra").

Outro problema é o fato de que a análise deve levar em conta que uma determinada seqüência de grafemas que é mapeada para uma sílaba pode ter várias realizações acústicas possíveis. O mapeamento é do tipo um para muitos.

# Capítulo 3

# Obtenção das marcas de pitch

## 3.1 Introdução

O PSOLA é um algoritmo de análise/modificação/síntese que permite a modificação do pitch e da duração do sinal de voz [1]. Este algoritmo exige, durante a fase de análise, o conhecimento das chamadas "marcas de pitch" [11], que são marcações ao longo do sinal de voz usadas pelo algoritmo, e cuja obtenção será descrita neste capítulo.

Torna-se necessário, para a obtenção automática das marcas de pitch, estimar o contorno de pitch do sinal de voz, bem como o modo de excitação. Isto é feito através da utilização de algoritmos de determinação de pitch (ADP) e de algoritmos de determinação de sonoridade (ADS).

A importância de tais algoritmos não se restringe ao algoritmo de análise/ síntese abordado. Estes algoritmos são utilizados em aplicações de praticamente todas as áreas de processamento de voz. Alguns exemplos de aplicações que geralmente se utilizam destes algoritmos são os vocoders [12] (área de codificação) e aplicações de conversão de voz [13] (área de síntese).

Grande parte do estudo de técnicas de determinação de pitch e sonoridade foi feito durante as décadas de 1960 e 1970, quando foram desenvolvidos vários ADP's e ADS's, cada um possuindo vantagens e desvantagens em relação aos outros [14]. Mesmo os algoritmos mais novos e sofisticados se baseiam nos mesmos princípios dos algoritmos desenvolvidos neste período.

Existem poucos estudos sobre a performance dos primeiros ADP's e ADS's. Alguns destes estudos serão citados como referência ao longo do texto. O autor deste texto teve a oportunidade de verificar a escassez destes estudos, e de comprovar informalmente alguns dos resultados relatados na literatura. O trabalho de Hess [15] é, provavelmente, a referência mais completa sobre métodos de extração de pitch e de decisão de sonoridade.

Este capítulo tem como objetivo descrever a implementação de uma rotina capaz de obter automaticamente as marcas de pitch exigidas na fase de análise do PSOLA. A Seção 3.2 mostra a relação entre o mecanismo de produção de voz e os parâmetros a serem analisados, e apresenta as definições de pitch e modo de excitação. Na Seção 3.3 serão apresentados os vários tipos de ADS's, alguns algoritmos implementados e os resultados obtidos. A Seção 3.4 segue a mesma ordem de apresentação da Seção 3.3, mas referindo-se aos ADP's, e acrescentando-se subseções que tratam da segmentação do sinal de voz, erros de estimação de pitch e algoritmos de suavização do contorno de pitch. A Seção 3.5 mostra como os algoritmos descritos nas seções anteriores são combinados para a obtenção das marcas de pitch.

## 3.2 Princípios e definições básicas

### 3.2.1 Mecanismo de produção de voz

Parte do aparelho fonador humano pode ser imaginado como um sistema de cavidades ressonantes que é percorrido por um fluxo de ar gerado nos pulmões. Este sistema de cavidades ressonantes é chamado de trato vocal. O fluxo de ar que atravessa o trato vocal é produzido por um sistema que aqui será chamado de fonte de excitação, e que é composto dos pulmões e da traquéia.

Os estudos do mecanismo de produção de voz sugerem a possibilidade de modelagens separadas para o trato vocal e para a fonte de excitação [16]. Será feita a seguir uma breve discussão sobre o mecanismo pelo qual é produzido o fluxo de ar que atravessa o trato vocal, já que os algoritmos que serão descritos neste capítulo medem parâmetros dos modelos usados para a fonte de excitação.

Uma das partes mais importantes do aparelho fonador é a laringe. A laringe pode ser definida como um anel cartilaginoso situado na parte superior da traquéia [17]. Do ponto de vista da produção de voz, o papel da laringe é modificar o fluxo de ar vindo dos pulmões, gerando um fluxo de ar periódico no trato vocal. A criação de um fluxo de ar periódico é feita na laringe por intermédio das *cordas vocais*. As cordas vocais estão ligadas a um sistema de cartilagens que permite modificar o seu posicionamento, e com isso modificar o som produzido pelo aparelho fonador. A abertura existente entre as cordas vocais é chamada de *glote* [17].

Quando as cordas vocais estão separadas, significando que a glote está aberta, e são assim mantidas enquanto o ar vindo dos pulmões passa pela laringe, ocorre a produção de sons conhecidos como *surdos*. Estes sons são

caracterizados por sua falta de periodicidade. O sinal que sai da laringe (excitação glotal, doravante chamada apenas de excitação), nesta situação, é chamado de excitação *surda*.

As cordas vocais podem também estar juntas em toda a sua extensão (glote fechada). Como, inicialmente, o ar vindo dos pulmões encontra a glote fechada, a pressão abaixo da glote aumenta, até o ponto em que é forçada a separação das cordas vocais (abertura da glote). Após a separação, quando a pressão acima da glote se iguala à pressão abaixo da glote, as cordas vocais se juntam novamente devido à sua elasticidade. Os sons produzidos por este posicionamento das cordas vocais são chamados de *sonoros*. Estes sons são caracterizados por sua periodicidade. Neste caso, a excitação é chamada de excitação *sonora*.

Existem outros tipos de excitação associados a outras posições das cordas vocais. Quando a glote não está completamente fechada, mas ainda assim posicionada de forma a permitir a vibração das cordas vocais, a excitação pode ser composta de uma mistura de excitações sonoras e surdas. Este sinal é chamado de *excitação mista*.

### 3.2.2 Modo de excitação

O modo de excitação pode ser definido como a presença de excitação sonora e/ou surda [14]. O problema de determinar o modo de excitação é resolvido por algoritmos que serão aqui referidos como algoritmos de determinação de vozeamento (ADV). A importância dos ADV's na implementação do PSOLA reside no fato de que a estratégia utilizada para a obtenção das marcas de pitch é diferente, dependendo se o trecho é sonoro ou não. Além disso, só faz sentido modificar o pitch de trechos sonoros. Tentativas de modificação do pitch de trechos surdos não são desejáveis, pois o método de modificação de pitch do PSOLA introduz ruído tonal nestes trechos [11].

### 3.2.3 Pitch

Em trechos sonoros, a periodicidade da excitação fornece uma percepção de altura (característica que distingue sons graves de agudos). Esta percepção de altura está relacionada com um parâmetro conhecido como pitch. Devido à existência de inúmeras definições de pitch, pode haver uma certa confusão quando da utilização deste termo.

Hess [14] dá várias definições de pitch, baseadas nos pontos de vista da produção, do processamento de sinais e da percepção. O significado original do termo pitch está associado ao ponto de vista da percepção. Sob este ponto de vista, o pitch seria a freqüência da senóide que evoca a mesma sensação

de altura do sinal de voz. Esta definição só se aplica a um sinal de voz estacionário de longa duração, pois é necessário algum tempo para um ser humano poder comparar a sua percepção de altura de uma senóide pura com a de um sinal de voz. Por isso, esta definição não tem muita importância prática na implementação de técnicas automáticas para determinação do pitch.

Do ponto de vista da produção, o pitch estaria relacionado com o período de um ciclo individual da excitação. Sob o ponto de vista de processamento de sinais, o pitch pode ser visto como a freqüência ou período fundamental do sinal de voz (a freqüência fundamental às vezes é representada como F0 na literatura). Os ADP's que são descritos na Seção 3.4 utilizam-se de definições de pitch derivadas destes pontos de vista.

É uma tarefa difícil extrair automaticamente o pitch usando a definição baseada no ponto de vista da percepção por dois motivos. Primeiro, porque a definição só se aplica a sons de longa duração. Segundo, porque a definição exigiria, a priori, intervenção humana no processo de extração de pitch ou no treinamento de um sistema que pudesse realizar a tarefa automaticamente, a não ser que tenhamos um bom modelo para a percepção do pitch.

Para um som complexo de longa duração, as definições de pitch derivadas dos pontos de vista de produção e de processamento de sinais costumam concordar com a definição derivada do ponto de vista da percepção.

## 3.3 Algoritmos de Determinação de Sonoridade

### 3.3.1 Definição e classificação dos ADS's

A tarefa de um ADS é, dado um segmento de voz, classificá-lo como sonoro ou surdo (reconhecer o modo de excitação). Se o segmento é caracterizado pelo fato de não apresentar excitação sonora nem surda, ele pode ser classificado como *pausa* ou *silêncio*. No caso da presença das duas excitações, teremos um segmento com excitação mista (o caso, por exemplo, de fricativas sonoras).

Para a obtenção de marcas de pitch, não é útil definir as classes silêncio e excitação mista. No caso do PSOLA, o importante é saber quais trechos do sinal de voz têm pitch, e portanto podem ser modificados [11]. Daqui em diante, um segmento de silêncio será classificado como surdo e um segmento de excitação mista será classificado como sonoro.

O fato de que alguns algoritmos de modificação de voz exigem uma decisão binária de sonoridade é um dos responsáveis por algumas das distorções

encontradas nestes algoritmos. Um exemplo são os zumbidos encontrados durante a síntese de fricativas sonoras usando a codificação por predição linear (*linear prediction coding*, LPC).

Hess [14] apresenta uma classificação dos ADS's e descreve alguns deles, dividindo-os também em três categorias principais:

- Algoritmos de análise com limiares simples, usando poucos parâmetros básicos.
- Algoritmos baseados em métodos de reconhecimento de padrões.
- Algoritmos integrados para determinação conjunta de sonoridade e pitch.

### 3.3.2 Parâmetros e espaço de padrões

O problema de decisão de sonoridade é um problema típico de *classificação*. Classificação é a associação de cada padrão no espaço de padrões a uma classe. Esta associação do padrão à classe é feita pelo *classificador*.

Qualquer algoritmo de decisão de sonoridade (como qualquer método de reconhecimento de padrões) consiste em uma extração de parâmetros acompanhada de uma classificação. Logo, o problema de decisão de sonoridade pode ser resolvido se forem extraídos parâmetros que caracterizem bem o modo de excitação.

Os ADS's valem-se de qualquer parâmetro que possa ser calculado independente do tipo de sinal (ou seja, independente do modo de excitação e articulação) [14]. Alguns destes parâmetros são [16][4]:

- Energia: Um estimador para a energia de curta duração de um segmento de $N$ amostra do sinal de voz $s(n)$, terminando na amostra $m$, é

$$E_s(m) = \sum_{n=m-N+1}^{m} s^2(n) \tag{3.1}$$

- Taxa de cruzamento pelo zero: Um estimador para a taxa de cruzamento de curta duração de um segmento de $N$ amostras do sinal de voz $s(n)$, terminando na amostra $m$, é

$$ZCR_s(m) = \frac{1}{N} \sum_{n=m-N+1}^{m} \frac{|sgn\{s(n)\} - sgn\{s(n-1)\}|}{2} w(m-n) \tag{3.2}$$

- Razão entre as autocorrelações com atraso $\tau = 1$ e com atraso $\tau = 0$, definida por

$$a = \frac{R(1)}{R(0)} \tag{3.3}$$

onde $R(\tau)$ é a estimativa da autocorrelação do trecho do sinal de voz analisado, com atraso (*lag*) $\tau$. Um estimador para a autocorrelação é apresentado na seção 3.4.3. Se houver alta correlação entre as amostras, que é o que ocorre no caso de um trecho sonoro, tem-se $a \approx 1$. Porém, $a \approx 0$, no caso de um trecho surdo, devido à baixa correlação entre amostras neste caso.

- Razão entre a energia do sinal de voz em diferentes sub-bandas: Sejam $E_{baixa}$ a estimativa para a energia de um segmento do sinal de voz filtrado por um filtro passa-baixas com freqüência de corte $f_c$ e $E_{alta}$ a estimativa da energia para o mesmo segmento filtrado por um filtro passa-altas também com freqüência de corte $f_c$. Tem-se então que a razão $\beta$ entre estas duas estimativas

$$\beta = \frac{E_{baixa}}{E_{alta}} \tag{3.4}$$

nos indica se o trecho é sonoro, se o valor estiver acima de um limiar, ou surdo, se estiver abaixo.

A seguir é apresentado um exemplo de um ADS simples usando um único parâmetro (razão de autocorrelação calculada a partir da equação 3.3) e um limiar de decisão. Neste ADS, escolhe-se um limiar entre 0 e 1. Se a razão estiver abaixo do limiar, o trecho é surdo, caso contrário, é sonoro. Note que o espaço de padrões aqui é unidimensional. É razoável pensar que aumentando a dimensão do espaço de padrões podemos aumentar a separação entre as classes, facilitando a tarefa do classificador.

A energia e a taxa de cruzamento pelo zero podem, juntas, nos indicar se um trecho do sinal é sonoro ou não [16]. Um trecho sonoro possui, em geral, alta energia, em comparação com um trecho surdo, e a energia dos trechos classificados como silêncio é ainda menor. A taxa de cruzamento pelo zero, no caso de uma fricativa, é maior que a de um trecho sonoro, enquanto que a taxa para o silêncio está entre a taxa das fricativas e a dos sons sonoros.

Então, através da energia e da taxa de cruzamento pelo zero, é possível não só identificar os trechos sonoros e surdos, como também separá-los dos

trechos de silêncio. Aqui, teríamos um espaço de padrões bidimensional, o que significa, a priori, que haverá maior separação entre as classes. Podemos encontrar uma superfície que separa as classes formadas por padrões de um conjunto de treinamento, de forma a minimizar o erro de classificação.

Para avaliar a eficiência de ADS's baseados nesta abordagem, foram obtidos gráficos que mostram padrões sonoros e surdos no espaço de padrões. Foram determinados os centróides das classes, e o espaço de padrões foi dividido pela reta mediatriz dos dois centróides. O centróide de uma classe é obtido através da média aritmética de todos os padrões que pertencem à classe.

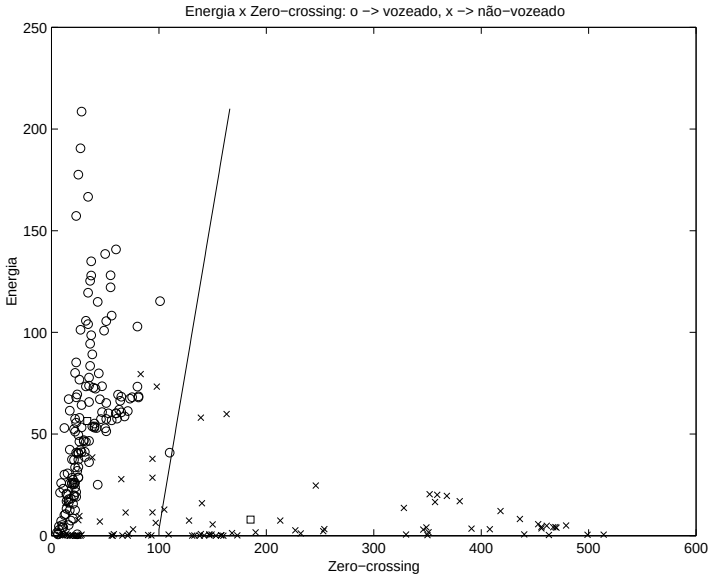


Figura 3.1: Espaço de padrões, usando como parâmetros a energia e a taxa de cruzamento pelo zero. Os padrões foram obtidos usando uma segmentação com janela retangular de 40 ms de comprimento e intervalo entre janelas de 10 ms.

A Figura 3.1 usa como parâmetros a energia e a taxa de cruzamento pelo zero. A decisão de sonoridade foi feita manualmente, através da observação da forma de onda no tempo. Os padrões foram obtidos através de uma segmentação usando janela retangular de comprimento 40 ms e um intervalo entre as janelas de análise de 10 ms. Os padrões '**o**' representam trechos sonoros e os '**x**' representam trechos surdos. A reta é a mediatriz dos centróides, representados por quadrados.

Foi verificado que existem segmentos do sinal que estão na transição de um trecho sonoro para um surdo e vice-versa. Estes segmentos, a priori,

não podem ser corretamente classificados, pois encontram-se na fronteira da superfície separadora, também chamada de *zona de confusão*. Muitos dos erros de classificação ocorrem nestas transições.

A Figura 3.2 usa como parâmetros a energia e a distância média entre cruzamentos pelo zero. A distância entre cruzamentos pelo zero é o intervalo de tempo entre instantes adjacentes em que o sinal de voz passa de positivo para negativo ou vice-versa. A configuração dos padrões no espaço de padrões quando se usam estes parâmetros é radial, e os padrões que correspondem aos trechos surdos estão mais próximos da origem. A distribuição radial dos parâmetros sugere que outro tipo de divisão do espaço de padrões seria mais eficiente, como, por exemplo, dividir as classes usando circunferências em vez de retas.

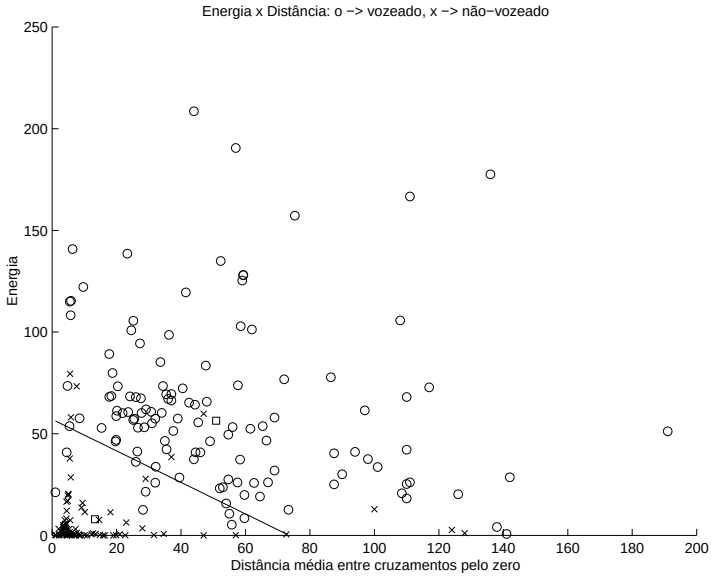


Figura 3.2: Espaço de padrões, usando como parâmetros a energia e a distância média entre cruzamentos pelo zero.

De fato, foi verificado experimentalmente que um dos maiores problemas em um ADS está na escolha dos limiares. Por exemplo, é fácil ver que utilizar limiares fixos não é uma boa idéia, pois uma pessoa pode falar mais alto ou mais baixo, e o nível de ruído de fundo pode variar.

### 3.3.3 ADS's implementados

Nesta seção serão apresentados alguns ADS's implementados e a suas performances. Alguns ADS's utilizam limiares que foram obtidos a partir um

conjunto de treinamento formado pelas sílabas 'a1', 'de358', 'foi549', 'gre685', 'lhan', 'ques1283' e 'sa1416' da base de dados. Para avaliar a performance dos ADS's foi usado um conjunto de teste formado pelas sílabas 'ão', 'di', 'mui995', 'nha', 'pre1218', 'sa1417' e 'ver1673'. Cada quadro obtido através de segmentação das sílabas foi classificado manualmente como sonoro ou surdo. Assim, é possível avaliar a performance dos algoritmos comparando a classificação dos ADS's com a classificação manual. Qualquer classificação do ADS que não concorda com a classificação manual é considerada um erro de decisão. Os limiares obtidos por treinamento foram escolhidos de forma a minimizar o erro de decisão no conjunto de treinamento.

Os seguintes ADS's foram implementados, neste trabalho:

1. O ADS de Savic & Benicasa [18]: Neste algoritmo, o limiar de energia é dado por

$$E_{thres} = 0.6\{\frac{1}{M}\sum_{i=1}^{M} 10\log_{10} E_i\} \tag{3.5}$$

onde $M$ é o número de segmentos, $E_i$ é a energia do i-ésimo segmento, calculada a partir de (3.1). Já o limiar para a taxa de cruzamento pelo zero é determinado por

$$ZCR_{thres} = N\frac{2480}{Fs} \tag{3.6}$$

onde $N$ é o número de amostras no segmento e $Fs$ é a freqüência de amostragem.

Para o limiar de taxa de cruzamento pelo zero. Se o log da energia de um segmento é maior que o limiar de energia ($10\log_{10} E_i > E_{thres}$) e a taxa de cruzamento pelo zero é menor ou igual ao limiar ($ZCR_i \leq ZCR_{thres}$), o segmento é classificado como sonoro.

2. Algoritmo baseado em um único parâmetro definido como

$$\alpha = \frac{E_i}{ZCR_i} \tag{3.7}$$

onde $E_i$ é a energia do segmento calculada a partir de (3.1) e $ZCR_i$ é a taxa de cruzamento pelo zero, calculada a partir de (3.2). Para valores de $\alpha$ acima de um limiar, o segmento é classificado como sonoro. A Figura 3.3 mostra o número de erros de classificação em função do limiar escolhido, para padrões dentro do conjunto de treinamento. O

limiar escolhido é o limiar associado ao menor números de erros de decisão.

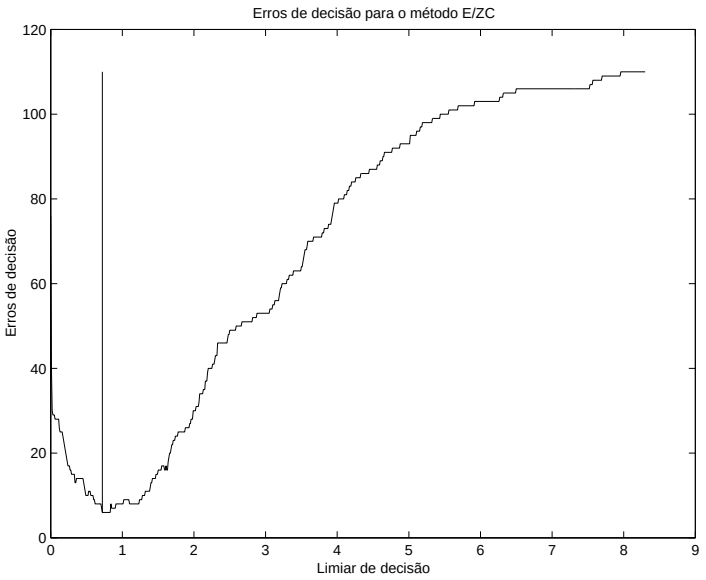


Figura 3.3: Número de erros de classificação em função do limiar para o algoritmo 2. O mínimo ocorre para $\alpha_{thres} = 0,720085$.

3. Algoritmo descrito por Rosenberg et al. [19], baseado no cepstrum. O cepstrum é uma transformação aplicada ao sinal que será abordada em maiores na seção 3.4.3. Em trechos sonoros, o cepstrum apresenta um pico proeminente associado à freqüência fundamental do sinal. De uma forma simplificada, podemos dizer que a altura do pico mais alto do cepstrum pode ser utilizada para determinar se o trecho é sonoro ou surdo.

   Se o pico do cepstrum excede um limiar, ou se não excede mas a taxa de cruzamento pelo zero está abaixo de um dado valor, o segmento é classificado como sonoro. Caso contrário, é classificado como surdo.

   Os limiares para o pico do cepstrum e a taxa de cruzamento pelo zero foram encontrados a partir do mesmo critério de minimização do número de erros de classificação dentro do conjunto de treinamento usado no algoritmo 2. As Figuras 3.4 e 3.5 ilustram o número de erros de classificação em função do limiar escolhido.

   Na verdade, a função erro é uma função de duas variáveis independentes: o limiar para o cepstrum e o limiar para taxa de cruzamento pelo zero. Admite-se aqui que o mínimo da superfície de erro gerada por

todas as possibilidades de limiares está próximo do ponto encontrado através das buscas separadas dos mínimos.

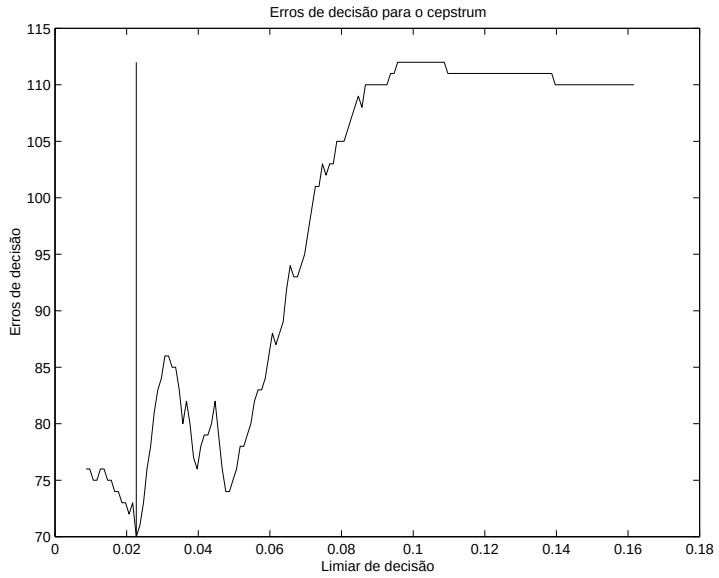


Figura 3.4: Número de erros de classificação em função do limiar para o algoritmo 3. O parâmetro extraído é valor do pico do cepstrum. O mínimo ocorre para $Pico_{thres} = 0,022685$.

4. Algoritmo descrito por Rosenberg et al. [19], baseado na autocorrelação. Neste algoritmo, estima-se a autocorrelação de um segmento de voz e se o máximo normalizado (valor no pico encontrado dividido pelo valor com atraso $\tau = 0$) da autocorrelação excede 0.3, o segmento é classificado como sonoro. Caso contrário, o segmento é surdo.

As tabelas abaixo mostram os resultados encontrados na avaliação dos algoritmos descritos. A primeira tabela apresenta os resultados obtidos para o conjunto de treinamento. Já a segunda tabela apresenta os resultados para o conjunto de teste. Na tabela, $\alpha$ é a taxa de acertos, $\beta$ é a taxa de erros V→UV (segmentos sonoros classificados como surdos), e $\gamma$ é a taxa de erros UV→V (segmentos surdos classificados como sonoros).

Foi constatado que os algoritmos implementados que fazem a decisão de sonoridade através de transformações que ressaltam a periodicidade do sinal (algoritmos 3 e 4) têm uma performance insatisfatória, como é indicado na literatura [14]. O algoritmo 2, que usa um limiar fixo obtido a partir do conjunto de treinamento, tem uma queda acentuada de performance quando utilizado no conjunto de teste, se comparado com o algoritmo 1, que calcula

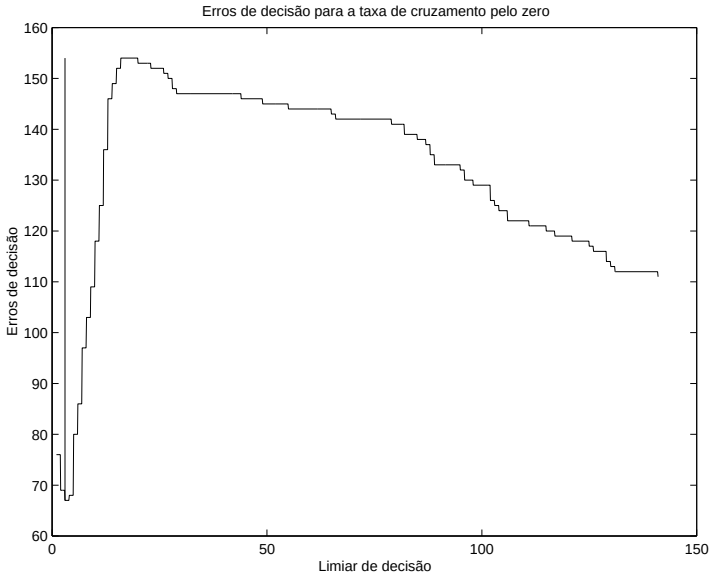


Figura 3.5: Número de erros de classificação em função do limiar para o algoritmo 3. O parâmetro extraído é a taxa de cruzamento pelo zero. O mínimo ocorre para $ZCR_{thres} = 3$.

Tabela 3.1: Performance no conjunto de treinamento. $\alpha$ é a taxa de acertos, $\beta$ é a taxa de erros V→UV e $\gamma$ é a taxa de erros UV→V.

| Algoritmo | $\alpha$ (%) | $\beta$ (%) | $\gamma$ (%) |
|---|---|---|---|
| 1 | 93.09 | 1.60 | 5.31 |
| 2 | 96.81 | 1.06 | 2.13 |
| 3 | 59.57 | 1.60 | 38.83 |
| 4 | 85.11 | 4.26 | 10.63 |

Tabela 3.2: Performance no conjunto de teste. $\alpha$ é a taxa de acertos, $\beta$ é a taxa de erros V→UV e $\gamma$ é a taxa de erros UV→V.

| Algoritmo | $\alpha$ (%) | $\beta$ (%) | $\gamma$ (%) |
|---|---|---|---|
| 1 | 85.88 | 9.41 | 4.71 |
| 2 | 77.06 | 17.06 | 5.88 |
| 3 | 68.82 | 2.94 | 28.24 |
| 4 | 79.42 | 8.82 | 11.76 |

os limiares a partir do sinal a ser analisado. O algoritmo 3 apresenta um aumento de performance no conjunto de teste, o que poderia ser explicado pela forma como os limiares foram encontrados, já que a hipótese simplificadora de independência dos limiares não é razoável. Os algoritmos não apresentaram boa performance nos testes, o que sugere que os espaços de padrões usados não eram adequados ou os classificadores eram insuficientes. No entanto, o algoritmo 2 foi escolhido por apresentar uma performance satisfatória quando os limiares são escolhidos aproveitando-se do fato de que toda sílaba possui um núcleo vocálico. O limiar, neste caso, é uma porcentagem do valor máximo do parâmetro obtido no núcleo vocálico.

## 3.4 Algoritmos de Determinação de Pitch

### 3.4.1 Classificação dos ADP's

Hess [14] divide os ADP's em duas classes: os que operam no domínio do tempo (ADP's no domínio do tempo) e os que exigem que uma transformação seja realizada num sinal de curta duração (ADP's de análise de curta duração). As duas classes de ADP's serão descritas a seguir.

- ADP's no domínio do tempo

  A saída do extrator básico de um ADP no domínio do tempo é uma seqüência de "marcas de pitch" que delimitam a fronteira de ciclos de pitch [14]. Daí, poderíamos concluir que os ADP's no domínio do tempo são uma escolha natural para a obtenção das marcas de pitch necessárias ao PSOLA.

  Em [20], é descrito um algoritmo para obtenção das marcas de pitch que pode ser classificado como um ADP no domínio do tempo. Este algoritmo, porém, não possibilita o grau de automatização desejado. Devido ao tamanho da base de dados (em torno de 1800 unidades), é desejável que o algoritmo de obtenção das marcas de pitch seja capaz de obtê-las de forma completamente automática, sem supervisão humana. Neste capítulo será descrita uma técnica para a obtenção das marcas de pitch que atende a esta especificação.

  Um dos problemas dos ADP's no domínio do tempo está associado ao fato de que estes algoritmos são mais sensíveis a degradações locais do sinal de voz [14], sendo menos confiáveis, nestas situações, que os ADP's baseados em análise de curta duração. Existem vários ADP's no domínio do tempo, alguns utilizando técnicas bastante sofisticadas [14].

Como a apresentação destes algoritmos foge do escopo deste capítulo, novamente o trabalho de Hess [15] é indicado como referência.

- ADP's de análise de curta duração

  Estes ADP's operam fazendo com que um trecho do sinal de voz seja transformado para uma representação mais conveniente que a representação no domínio do tempo, para ressaltar a periodicidade do sinal.

  Para a aplicação, neste projeto, a que se destinam estes algoritmos (obten-ção das marcas de pitch) existem algumas desvantagens nesta abordagem:

  – Estes ADP's retornam como saída o valor do pitch em determinados instantes de tempo. Como é perdida a relação de fase entre o sinal original e o sinal transformado, torna-se necessário fazer uma escolha da primeira marca de pitch num trecho sonoro, sendo as outras obtidas a partir do pitch calculado. Dependendo de como a primeira marca é escolhida, esta pode não ser sincronizada com um certo evento de um ciclo da excitação. Pode não haver, por exemplo, sincronismo com o fechamento da glote. Como veremos adiante, os instantes de fechamento da glote são as melhores posições para a colocação das marcas de pitch.

  – Estes ADP's exigem que o sinal seja segmentado, de tal forma que um segmento possua vários ciclos de pitch (assim o ADP é capaz de realçar a periodicidade do sinal através de uma transformação conveniente). Se houver variação do pitch de um ciclo para outro, a estimativa do ADP não corresponde ao valor exato do pitch, mas sim a uma espécie de média dos valores de pitch em cada ciclo contido no segmento.

  Um ADP baseado nesta abordagem foi escolhido para o desenvolvimento de um programa capaz de obter as marcas de pitch, devido à facilidade de, com este tipo de ADP, desenvolver rotinas automáticas. Alguns ADP's baseados nesta abordagem foram implementados. Descrições destes ADP's serão apresentadas nesta seção, assim como os resultados obtidos.

### 3.4.2 Segmentação do sinal de voz

Existem alguns detalhes sobre a segmentação que devem ser observados para que os ADP's de análise de curta duração funcionem corretamente.

Para fazer a segmentação do sinal, multiplica-se o sinal por uma seqüência de janelas deslocadas no tempo. Temos as seguintes possibilidades de escolha, com relação à segmentação: podemos escolher o comprimento da janela de análise, a amostra em que a janela começa (ou termina) e o tipo de janela utilizado (retangular, Hamming etc).

Os ADP's implementados foram testados usando o seguinte esquema de segmentação: todas as janelas tinham o mesmo comprimento e o intervalo entre o começo de uma janela e o começo da próxima era constante. Todas as janelas eram do mesmo tipo.

Como colocado anteriormente, os ADPs de análise de curta duração exigem que a janela de análise seja grande o suficiente para que se detecte alguma periodicidade. É preciso que a análise do sinal seja em "banda estreita", para termos resolução espectral suficiente para fazer a estimativa do pitch. É razoável pensar (e isto foi verificado por experiências realizadas) que sejam necessários pelo menos dois ciclos de pitch para que os ADPs de análise de curta duração funcionem confiavelmente. No caso de um ADP que se baseia na autocorrelação do resíduo de predição linear, o comprimento ótimo da janela de análise é de três períodos de pitch [21]. Como já foi observado, erros podem ocorrer se o parâmetro a ser medido (pitch) não for suficientemente constante dentro da janela de análise. A janela de análise deve ser suficientemente pequena para atender a esta condição. Às vezes não é possível atender simultaneamente a esta condição e à condição de análise em banda estreita, e então o algoritmo falha [14].

Como o valor do pitch varia ao longo de um enunciado, e uma boa estimativa do contorno de pitch deve acompanhar estas variações. Para isto, é necessário que o espaçamento entre as janelas de análise (intervalo entre o começo de uma janela e o começo da próxima) seja pequeno o suficiente. Esta escolha vai determinar a resolução temporal na estimativa do contorno de pitch, ou seja, o intervalo de tempo entre duas estimativas do pitch. Como não se deve esperar uma grande variação do pitch no intervalo de tempo correspondente a um ciclo de pitch, não seria econômico, do ponto de vista computacional, usar um espaçamento menor que um período fundamental.

### 3.4.3 Descrição dos ADP's implementados

1. Autocorrelação

   Uma estimativa da autocorrelação de curta duração de um segmento de $N$ amostras do sinal $s(n)$, terminando na amostra $m$, pode ser obtida com o seguinte estimador [16]:

$$r_s(\eta; m) = \frac{1}{N} \sum_{n=m-N+1+|\eta|}^{m} s(n)s(n-|\eta|)w(m-n) \qquad (3.8)$$

onde $\eta$ é o intervalo (*lag*) da autocorrelação e $w(n)$ é a janela de análise.

Se $s(n)$ é um sinal periódico, para valores de $\eta$ iguais a múltiplos inteiros do período, a função $r_s(\eta; m)$ apresentará máximos locais. Assumindo a quasi-periodicidade de um pequeno trecho sonoro de fala, a autocorrelação deste trecho deverá apresentar picos para intervalos iguais a múltiplos inteiros do período de pitch. Assim, o primeiro máximo local relevante da autocorrelação corresponde à freqüência fundamental do sinal de voz. A Figura 3.6 ilustra uma estimativa da autocorrelação para um trecho sonoro de um sinal de voz.

Podem ocorrer erros na estimativa do pitch no caso em que o sinal de voz tem muita energia concentrada em torno do primeiro formante. Os erros ocorrem devido ao aparecimento de picos na autocorrelação na freqüência do primeiro formante [16]. Este problema pode ser resolvido através de um pré-processamento não-linear no sinal de voz, antes de ser feito o cálculo da autocorrelação. Uma das técnicas utilizadas é o *center-clipping*, que consiste no seguinte mapeamento:

$$C\left|s(n)\right| = \begin{cases} s(n) - C^+ \ , & s(n) > C^+ \\ 0 \ , & C^- \leq s(n) \leq C^+ \\ s(n) - C^- \ , & s(n) < C^- \end{cases} \qquad (3.9)$$

onde os limites de "clipping", $C^+$ e $C^-$, são ajustados, tipicamente, para 30% do máximo absoluto do sinal [16]. Este mapeamento diminui a influência do primeiro formante sobre a estimativa do pitch [22]. A Figura 3.7 mostra a autocorrelação de um trecho sonoro depois de ter sido aplicado este mapeamento.

2. AMDF (*Average Magnitude Difference Function*)

   Uma estimativa da AMDF de curta duração de um segmento de $N$ amostras do sinal de voz $s(n)$, terminando na amostra $m$ pode ser obtida usando o seguinte estimador [16]:

$$\Delta M_s(\eta; m) = \frac{1}{N} \sum_{n=m-N+1}^{N} \left|s(n) - s(n-\eta)\right| w(m-n) \qquad (3.10)$$

   onde $\eta$ é o intervalo da AMDF e $w(n)$ é a janela de análise.

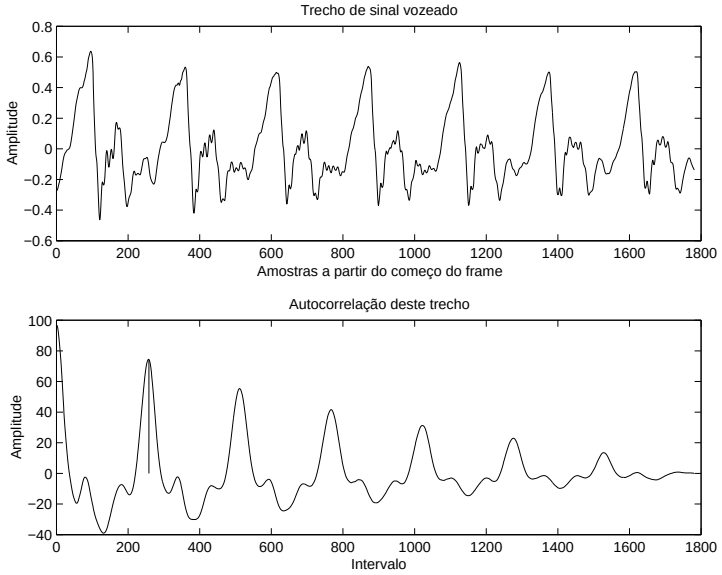


Figura 3.6: Autocorrelação de um trecho sonoro. O pico desejado ocorre no intervalo 258, que corresponde a uma F0 de 170,93 Hz.

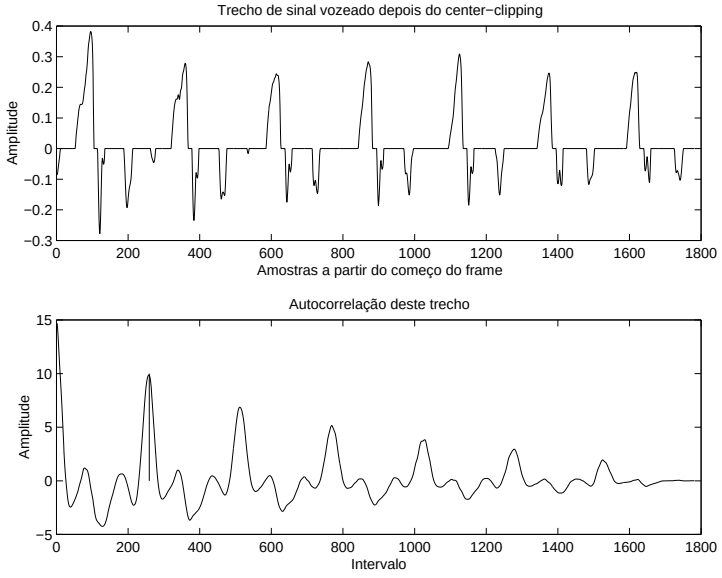


Figura 3.7: Autocorrelação de um trecho sonoro, depois de aplicado *center-clipping*. O pico desejado ocorre no intervalo 259, correspondendo a uma F0 de 170,27 Hz.

A idéia da AMDF é a mesma da autocorrelação: aproveitar-se da semelhança entre um sinal periódico e uma cópia sua com um atraso igual a um múltiplo do seu período. Em intervalos iguais a múltiplos inteiros do período de pitch, a AMDF exibe vales. A vantagem da AMDF sobre a autocorrelação é o fato de que em certos processadores uma operação de adição ou subtração seria muito mais rápida que uma multiplicação. No entanto, os processadores modernos são otimizados para multiplicações. Portanto a utilização da AMDF não representaria um ganho computacional num microcomputador atual. A Figura 3.8 ilustra o comportamento da AMDF em um trecho sonoro.

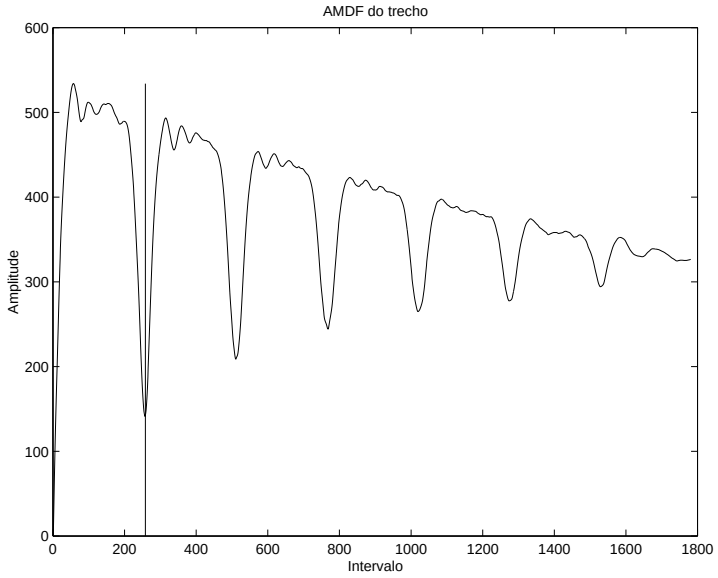


Figura 3.8: AMDF de um trecho sonoro. O vale desejado ocorre no intervalo 259, que corresponde a uma F0 de 170,27 Hz.

3. Método Cepstral

Assumindo a validade do modelo fonte-filtro de produção de voz, o sinal de voz pode ser obtido convoluindo a resposta impulsional do trato vocal com a excitação. A análise cepstral é uma técnica que permite, a partir do sinal de voz, separar a resposta impulsional do trato vocal da excitação (deconvolução). Faz parte de um conjunto de métodos conhecido como *processamento homomórfico* [23].

O cepstrum real de curta duração de um segmento do sinal $s(n)$, terminando na amostra $m$, pode ser obtido com o seguinte estimador [16]:

$$c_s(n;m) = IDTFT(\log|DTFT(s(n)w(m-n))|) \tag{3.11}$$

onde IDTFT é a transformada inversa de Fourier discreta no tempo, DTFT é a transformada de Fourier discreta no tempo e $w(n)$ é a janela de análise.

Os valores no domínio do cepstrum são conhecidos como *qüefrências*. Baixas qüefrências correspondem a componentes de variação lenta no espectro do sinal de voz, sendo, portanto, uma aproximação do cepstrum da resposta do trato vocal. As altas qüefrências correspondem a componentes de variação rápida no espectro, sendo uma aproximação para o cepstrum da excitação.

O cepstrum pode ser utilizado para a determinação do pitch através da busca do primeiro pico na região do cepstrum que corresponde à excitação. A qüefrência deste pico corresponde ao período de pitch. O cepstrum de um trecho de voz sonoro é ilustrado na Figura 3.9.

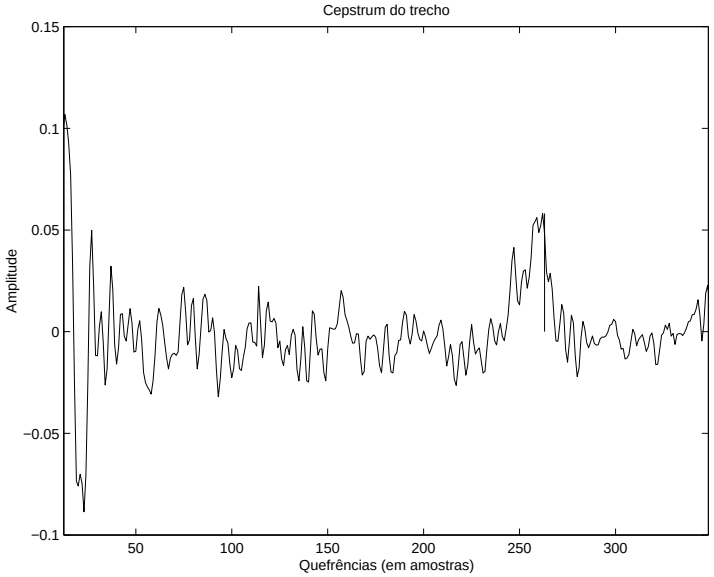


Figura 3.9: Cepstrum real de um trecho sonoro. O pico ocorre na qüefrência 263, que corresponde a uma F0 de 167,68 Hz.

4. SIFT (*Simple Inverse Filtering Technique*)

Um inconveniente da técnica de autocorrelação, que já foi comentado anteriormente, está no fato de que se o sinal de voz tiver muita energia em torno da freqüência do primeiro formante (F1), podem ocorrer erros

na estimativa, sendo a freqüência do primeiro formante confundida com o pitch [4][16].

A utilização do *center-clipping* no método de autocorrelação reduz esse tipo de erro. O *center-clipping* age como uma espécie de branqueamento no sinal de voz, deixando o espectro do sinal mais plano [16] e retirando um pouco da influência dos formantes na estimativa do pitch.

Outra forma de evitar este problema é utilizando-se da técnica de predição linear para fazer o branqueamento. Isso é feito no algoritmo SIFT, cujo diagrama de blocos é apresentado na Figura 3.10.

Neste algoritmo, é feita a filtragem inversa do sinal de voz, ou seja, encontra-se um filtro auto-regressivo que representa o trato vocal e o sinal de voz é introduzido no filtro inverso para obter o sinal de excitação na entrada do filtro (que é o resíduo da predição linear). Daí o nome SIFT, dado ao algoritmo. Depois, é obtida a autocorrelação do resíduo.

Os resultados obtidos com o SIFT são ilustrados na Figura 3.11.

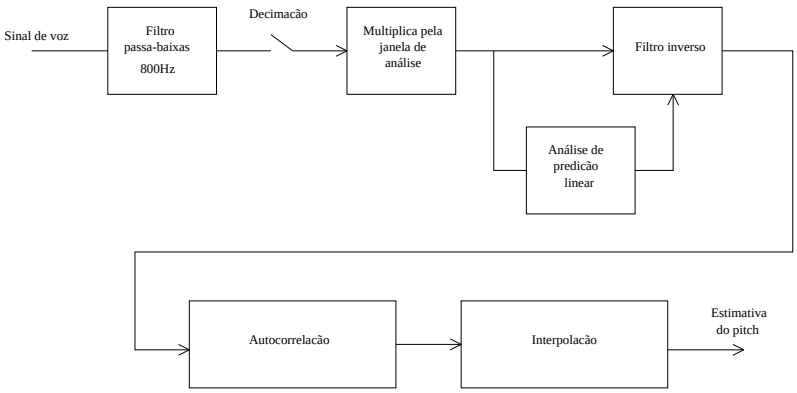


Figura 3.10: Diagrama de blocos do SIFT.

5. HPS (*Harmonic Product Spectrum*)

A técnica HPS utiliza-se do fato de que, se a DTFT do sinal for calculada com resolução suficiente, os harmônicos da excitação serão automaticamente visíveis no espectro calculado. A transformação a ser calculada para evidenciar a periodicidade na DTFT é [16]:

$$P(\omega; m) = \prod_{r=1}^{R} S(r\omega; m) \tag{3.12}$$

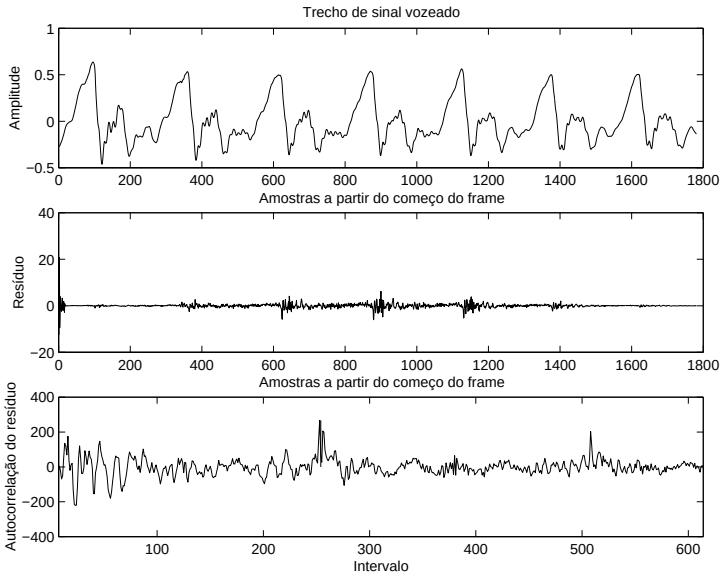


Figura 3.11: Trecho de um sinal de voz, resíduo da predição linear e autocorrelação do resíduo. O pico desejado da autocorrelação ocorre no intervalo 254, que corresponde a uma F0 de 173,62 Hz.

onde $m$ é a amostra em que o segmento termina, e $R$ tem um valor pequeno (tipicamente 5). $S(\omega; m)$ é uma estimativa do espectro de curta duração do segmento do sinal de voz. Em outras palavras, o HPS é o produto de cópias comprimidas (ao longo do eixo das freqüências) do espectro original. Se o domínio do espectro original for comprimido usando fator 2, o segundo harmônico do sinal cairá na posição da fundamental. Se for comprimido usando fator 3, o terceiro harmônico cairá na posição da fundamental. Como sempre haverá um harmônico que será mapeado para a posição da fundamental, o produto das cópias comprimidas terá um pico na freqüência fundamental. As Figuras 3.12 e 3.13 ilustram o funcionamento deste ADP.

### 3.4.4 Escolha do ADP

Verificou-se que, dos ADP's implementados, os que apresentaram melhores resultados foram os ADP's baseados na autocorrelação, na AMDF e no cepstrum. O HPS não tem uma boa resolução em freqüência em comparações com os outros, além de ser computacionalmente complexo. No caso do SIFT, a autocorrelação do resíduo de predição linear é muito ruidosa em comparação com a autocorrelação do sinal de voz, dificultando a localização

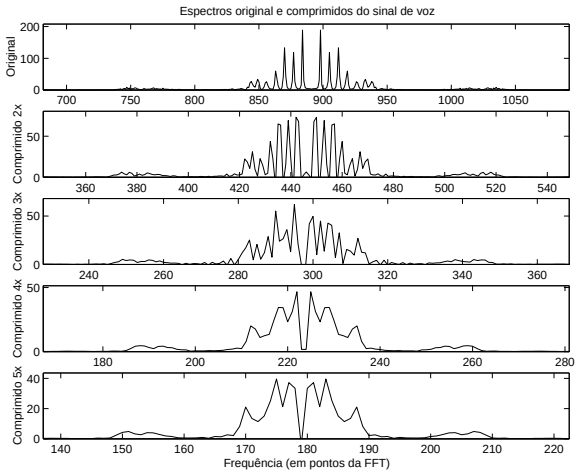


Figura 3.12: Espectro original do sinal e espectros comprimidos.

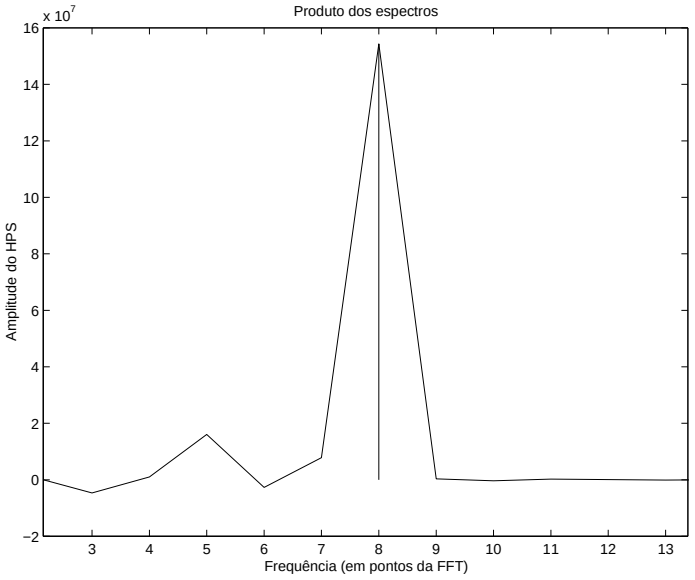


Figura 3.13: Trecho do sinal e HPS. O pico desejado do HPS ocorre em 8, correspondendo a uma F0 de 198,09 Hz. A resolução da estimativa de F0 é de 24,76 Hz.

do pico relevante. E o SIFT tem um custo computacional maior.

No caso do cepstrum, é extraído o logaritmo do espectro do sinal, enquanto que na autocorrelação o espectro é elevado ao quadrado. O achatamento do espectro no cepstrum, causado pelo logaritmo, reduz a relação sinal/ruído, fazendo com que a estimativa do pitch obtida com o cepstrum seja mais ruidosa que a estimativa feita com a autocorrelação [14]. Esse fenômeno foi verificado em experiências feitas com os ADP's.

Como a AMDF e a autocorrelação calculada pela definição têm um custo computacional da ordem de $n^2$, as únicas alternativas razoáveis são os métodos baseados no cepstrum e na autocorrelação calculada a partir da FFT, que têm custo computacional da ordem de $n \log n$.

### 3.4.5 Erros na estimação do pitch

Esta seção descreve alguns erros cometidos na estimativa do pitch e como eles podem ser corrigidos ou diminuídos.

- Erros grosseiros: são falhas drásticas na determinação do pitch, que fazem com que a estimativa do algoritmo seja muito diferente do valor "exato" da medida. Podem ser causados por um primeiro formante forte, mudanças rápidas nos parâmetros do trato vocal, performance insuficiente do algoritmo etc. Estes erros podem ser amenizados usando-se métodos para suavizar o contorno de pitch encontrado [14].

- Erros causados por inacurácias na medida: são pequenos desvios do valor "exato" do pitch. Podem ser causados por inacurácias na determinação, por exemplo, da posição de um pico, por inacurácias intrínsecas causadas, por exemplo, pela quantização do sinal no tempo, ou por pequenas flutuações na voz [14].

  Este é um exemplo de como pequenas flutuações na voz podem causar erros de análise: em um ADP de análise de curta duração, uma janela de análise deve conter vários ciclos de pitch. Se o pitch varia de um ciclo para outro, o ADP não tenta estimar o valor "exato" do pitch no ciclo em questão, mas sim uma espécie de média do pitch de todos os ciclos incluídos pela janela. Uma forma de diminuir estes erros é analisando trechos de voz cujo contorno de pitch não varia muito. Para isso, o locutor deve falar tentando manter o pitch o mais constante possível [2]. Além disso, pode-se também utilizar uma janela que seja apenas grande o suficiente para conter, digamos, dois ciclos somente do sinal de voz. Isto exigiria uma segmentação adaptativa. Esta é uma das razões pela qual é interessante, nas fases de construção de um banco de dados

para síntese de fala concatenativa, que a gravação das unidades seja feita com o mínimo de entonação possível [2]. Esta medida diminuiria os erros de análise.

Estes erros podem ser atenuados com a utilização de algoritmos de suavização linear.

- Erros de sonoridade: se houver algum erro de decisão de sonoridade, o ADP pode tentar estimar o pitch de um trecho surdo, causando um erro grosseiro na estimativa [14]. Na verdade o erro é cometido pelo ADS e não pelo ADP. Usando um bom ADS é possível evitar uma grande quantidade destes erros.

### 3.4.6 Algoritmos de suavização

Já foi comentado, em outras seções, que erros grosseiros na estimativa de pitch podem ser minimizados com o uso de técnicas de suavização não-linear. Sambur et al. [24] propuseram um método de suavização consistindo em um filtro de mediana seguido de um filtro linear. O filtro de mediana é capaz de diminuir os erros grosseiros na extração do contorno de pitch, enquanto que o filtro linear diminui os erros devidos às inacurácias de medida (que podem ser interpretados como um ruído sobreposto ao contorno de pitch).

Existem alguns algoritmos de suavização que se utilizam de programação dinâmica para encontrar um contorno de pitch ótimo sob um certo sentido. A idéia é encontrar um contorno de pitch que maximiza uma função objetivo. Esta função objetivo é tal que impõe certas restrições na variação do pitch de um frame para outro, fazendo com que grandes variações do pitch sejam penalizadas [13].

Uma comparação é feita, na figura 3.14, entre o contorno de pitch suavizado e o não suavizado, para o enunciado "A questão foi retomada no congresso". O algoritmo implementado procura no contorno de pitch estimativas que estão 10% acima (ou abaixo) da média das quatro estimativas ao redor desta. Encontrando uma estimativa com esta propriedade, ele susbtitui a estimativa pela média das quatro estimativas circundantes.

Por exemplo, vamos supor que $fpitch$ é um vetor que contém as estimativas de pitch encontradas por um ADP. Para cada estimativa $fpitch(i)$, o algoritmo verifica se ela satisfaz à propriedade

$$fpitch(i) > media(i) + 0.1media(i) \tag{3.13}$$

onde

$$media(i) = \frac{\sum_{j=-2}^{2} fpitch(i+j)}{4} \tag{3.14}$$

Então, a estimativa original $fpitch(i)$ é substituída pela média, ou seja,

$$fpitch(i)|_{novo} = media(i) \tag{3.15}$$

Depois, o algoritmo procura estimativas que possuem a propriedade

$$fpitch(i) < media(i) - 0.1media(i) \tag{3.16}$$

e faz

$$fpitch(i)|_{novo} = media(i) \tag{3.17}$$

ao encontrar uma estimativa que satisfaz esta propriedade.

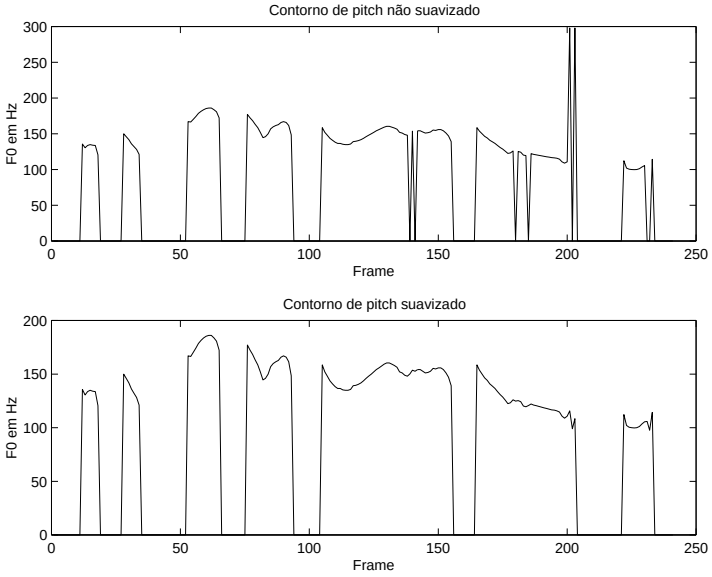


Figura 3.14: Comparação entre contornos de pitch suavizado e não suavizado para o enunciado "A questão foi retomada no congresso".

## 3.5 Marcas de pitch

As marcas de pitch são marcações ao longo do sinal de voz que indicam um evento específico no ciclo de pitch nos trechos sonoros, ou marcações com um espaçamento arbitrário nos trechos surdos. É importante que o

posicionamento das marcas de pitch seja consistente em todas as unidades da base de dados. Se as marcas de pitch não forem posicionadas de forma consistente em cada uma das unidades, ou seja, no mesmo instante relativo dentro de um ciclo, ocorrerão descasamentos de fase no sinal concatenado [2].

A bibliografia da área sugere que a melhor posição para a colocação das marcas de pitch é no instante de fechamento glotal [2][11][5][25][26]. Porém, o problema de estimar o instante de fechamento glotal a partir do sinal de voz é difícil [2][16][5]. Foram usados, por esse motivo, pontos de referência mais facilmente indentificáveis por processos automáticos como, por exemplo, picos na forma de onda no tempo. Algumas das obras da lista de referências sugerem esta solução [2][26].

Neste projeto, foi desenvolvido um programa em C que faz a extração de pitch das sílabas da base de dados. Este programa usa o algoritmo de autocorrelação com *center-clipping*, sem suavização do contorno de pitch estimado. A decisão de não utilizar suavização foi feita depois da constatação de que o algoritmo descrito na seção 3.4.6 não funcionava muito bem em alguns casos. A decisão de sonoridade usa o segundo algoritmo descrito na Seção 3.3.3.

A partir do pitch e da forma de onda no tempo, um script de Matlab obtém as marcas de pitch. O algoritmo do script procura posicionar as marcas de pitch da seguinte forma: procura-se a amostra de maior amplitude do sinal que se encontra em um trecho sonoro, e a partir dela propaga as outras marcas de pitch usando o valor de pitch local. As marcas de pitch propagadas são deslocadas para a amostra de maior amplitude dentro de uma janela de busca em volta da marca propagada.

Foi adotada a técnica descrita em [20] de eliminar as amostras dos segmentos de voz antes da primeira marca de pitch e depois da última marca, com o objetivo de garantir a continuidade de fase após a concatenação. Nessa etapa, foi obtida uma nova base de dados com as extremidades retiradas, e foi ainda gerado um banco de dados contendo as marcas de pitch de cada arquivo de sílaba.

## 3.6 Conclusão

Neste capítulo foram apresentados os conceitos de pitch e de modos de excitação, bem como sua importância, e os algoritmos para determiná-los. Foi constatada a importância da decisão de sonoridade e a dificuldade de implementação de um bom ADS. Para que o algoritmo fosse automático, foi escolhido um ADP baseado na autocorrelação, posicionando as marcas de

pitch na amostra de maior amplitude dentro de um ciclo. O seguinte ADP para esta aplicação: autocorrelação com *center-clipping*, calculada a partir da FFT.

Para garantir a continuidade de fase no sinal concatenado, as extremidades das sílabas antes da primeira marca de pitch e depois da última foram retiradas. Isso exigiu uma regravação da base de dados.

# Capítulo 4

# TD-PSOLA

## 4.1 Introdução

Como já foi comentado anteriormente, qualquer algoritmo capaz de modificar o pitch e a duração do sinal de voz pode ser utilizado, a priori, como um algoritmo de concatenação em um sistema TTS baseado na abordagem concatenativa. O algoritmo TD-PSOLA (*time domain - pitch synchronous overLap and add*) é um algoritmo capaz de modificar a freqüência fundamental e a duração do sinal de voz.

Este capítulo apresenta o algoritmo TD-PSOLA. Na Seção 4.2 é feita uma apresentação geral dos algoritmos de concatenação usados em sistemas TTS, e de que forma eles desempenham sua tarefa. Na Seção 4.3, alguns dos algoritmos mais conhecidos são descritos de forma sucinta. A Seção 4.4 descreve o algoritmo TD-PSOLA em detalhes. A Seção 4.5 apresenta uma interpretação da etapa de modificação do algoritmo no domínio da freqüência. A Seção 4.6 apresenta algumas variantes do PSOLA e suas características e a Seção 4.7 descreve a implementação do algoritmo.

## 4.2 Algoritmos de concatenação

Como já foi explicado anteriormente, sistemas TTS baseados na abordagem concatenativa sintetizam voz a partir da junção de segmentos de voz retirados de um corpus linguístico previamente gravado. Estes segmentos geralmente são retirados de palavras diferentes, e podem ocorrer em contextos diferentes. Esta diferença de contexto faz com que ocorra, freqüentemente, a situação na qual os parâmetros do sinal de voz não evoluem, ao longo do tempo, de forma suave de uma unidade para outra. Este descasamento dos parâmetros do sinal de voz na fronteira das unidades diminui a

inteligibilidade e naturalidade da voz sintetizada.

A função dos algoritmos de concatenação é providenciar a evolução temporal suave dos parâmetros. Para isso, é necessário que os algoritmos utilizados permitam a modificação destes parâmetros. Por isso, todo algoritmo que permite modificar os parâmetros do sinal de voz é um candidato potencial a algoritmo de concatenação.

Outra função importante delegada aos algoritmos de concatenação é a introdução de prosódia no sinal sintetizado. Para que isto seja possível, o algoritmo deve ser capaz de modificar o contorno de pitch e a duração do sinal de voz. Por isso, em toda descrição de um algoritmo de concatenação, a forma como estes parâmetros são modificados é sempre explicitada.

## 4.3 Exemplos de algoritmos de concatenação

Os algoritmos de concatenação são algoritmos de análise/síntese do sinal de voz, nos quais temos uma representação intermediária conveniente para a alteração dos parâmetros do sinal de voz. Então, podemos chamá-los também de algoritmos de análise/modificação/síntese do sinal de voz. Esta seção descreve alguns algoritmos de concatenação. O objetivo é apresentar uma visão das principais características destes algoritmos, possibilitando posteriores referências a estes algoritmos ou comparações destes algoritmos com o TD-PSOLA.

O termo "modelo" é usado para indicar a forma como o sinal de voz é representado (representação intermediária). Já o termo "algoritmo" é usado para indicar os procedimentos que convertem o sinal, representado no domínio do tempo, na sua representação intermediária (análise), os que modificam a representação intermediária (modificação), e os que convertem a representação intermediária para um sinal no domínio do tempo (síntese).

### 4.3.1 LPC (*Linear Prediction Coding*)

O algoritmo LPC é baseado em um modelo de produção de voz, e permite que o pitch e a duração sejam modificados. Sendo assim, este algoritmo pode ser utilizado como um algoritmo de concatenação. Além disso, ele ainda permite modificar a envoltória espectral do sinal, o que pode ser útil para eliminar descasamentos espectrais nas fronteiras entre unidades.

Como já foi visto anteriormente, o aparelho fonador pode ser dividido em duas partes, se admitimos a validade do modelo fonte-filtro: a fonte de excitação e o trato vocal. No modelo LPC, o sinal produzido pela fonte de excitação é modelado como um trem de impulsos periódicos ou um ruído

branco, dependendo se o trecho é sonoro ou surdo, respectivamente. Já o trato vocal é modelado como um filtro digital $V(z)$ (equação 4.1) do tipo autorregressivo:

$$V(z) = \sum_{i=0}^{p} a_i z^{-i}, \text{ com } a_0 = 1 \tag{4.1}$$

onde $a_i$ são os coeficientes do filtro e $p$ é a sua ordem.

O filtro digital deve ser obtido para trechos de voz nos quais os coeficientes do filtro possam ser considerados praticamente constantes. O sinal de voz deve, então, ser dividido em vários segmentos nos quais é praticamente estacionário. O pitch pode ser alterado através da modificação do período do trem de impulsos, quando o trecho é sonoro. A duração pode ser modificada inserindo ou retirando segmentos.

Como o trato vocal é representado por um modelo auto-regressivo, o modelo LPC não permite uma boa representação do trato vocal quando existem fortes anti-ressonâncias, como no caso dos sons nasalizados. Outro problema do modelo é a representação extremamente simplista do sinal de excitação. Isto provoca o aparecimento de zumbidos em trechos surdos ou com excitação mista, ou rouquidão nos trechos sonoros, devido ao caráter binário da decisão de vozeamento ou erros do ADS.

### 4.3.2 HNM (*Harmonic + Noise Model*)

O modelo HNM, ou modelo híbrido harmônico/estocástico (H/S), é um modelo baseado em uma abordagem fenomenológica, isto é, não é baseado em referências diretas ao mecanismo fisiológico de fonação [1]. Ele resolve o problema da representação simplificada do sinal de excitação através da separação do sinal de voz em duas componentes: uma determinística e outra estocástica [2]. O problema é resolvido porque o algoritmo baseado neste modelo possui procedimentos de modificação diferentes para cada um dos componentes. Ele usa representações espectrais para cada componente, que serão descritas a seguir.

O sinal original é modelado como a soma da componente determinística $\hat{s}(t)$ com a estocástica $e(t)$, de modo que podemos escrever

$$s(t) = \hat{s}(t) + e(t) \tag{4.2}$$

A componente determinística é modelada como uma soma de componentes harmônicas:

$$\hat{s}(t) = \sum_{k=-K(t)}^{K(t)} A_k(t) e^{jkt\omega_0(t)} \tag{4.3}$$

onde $K(t)$ é o número de harmônicos da excitação, $A_k(t)$ são os coeficientes de cada uma das exponenciais e $\omega_0(t)$ é a freqüência fundamental.

A componente estocástica é modelada como a saída de um filtro só de pólos multiplicada por uma função de envoltória de energia. Esta representação é motivada pela estrutura temporal encontrada na componente estocástica [25]:

$$e(t) = w(t)[A(t, Z) * n(t)] \tag{4.4}$$

onde $w(t)$ é a envoltória, $A(t, Z)$ é um filtro só de pólos normalizado e $n(t)$ é um ruído gaussiano. O símbolo * representa a operação de convolução.

Este algoritmo permite modificar o pitch e a duração do sinal de voz. Também é possível modificar a envoltória espectral devido à sua natureza paramétrica. Segundo a literatura, o fato do modelo ser paramétrico contribui para a perda de naturalidade [5].

Como já foi mencionado, problemas encontrados em outros algoritmos são resolvidos no HNM com o tratamento independente das componentes determinística e estocástica. No entanto, a complexidade computacional é muito grande em relação a outros algoritmos [5]. Maiores detalhes sobre este algoritmo podem ser encontrados nas referências [2][25].

## 4.4 Descrição do TD-PSOLA

Assim como o HNM, o TD-PSOLA também é um algoritmo baseado em uma abordagem fenomenológica. Mas, ao contrário do HNM, ele usa uma representação no domínio do tempo, ao invés da espectral. Este é o motivo pelo qual é às vezes chamado de "modelo nulo" [5]. Esta seção apresenta uma descrição das etapas de análise, síntese e modificação do sinal de voz usando o TD-PSOLA.

### 4.4.1 Modificações no domínio do tempo

Antes de descrever o algoritmo, será apresentada uma discussão sobre a possibilidade de alterar parâmetros como o pitch e a duração usando algoritmos que atuam no domínio do tempo. Tais algoritmos se baseiam na observação de que a forma de onda do sinal de voz apresenta uma quase periodicidade nos trechos sonoros.

Esta quase periodicidade pode ser interpretada da seguinte maneira: o sinal produzido pela fonte de excitação pode ser modelado como um trem de impulsos, como no modelo LPC. Admitindo a validade do modelo fonte-filtro, o sinal produzido pelo aparelho fonador é a resposta impulsional do trato vocal convolucionada com o trem de impulsos, como mostrado na Figura 4.1.

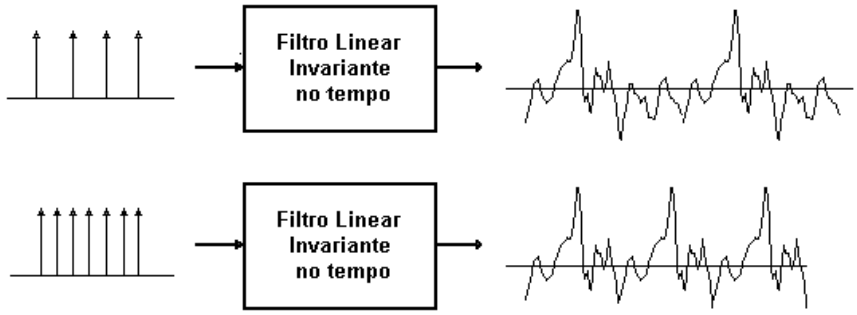


Figura 4.1: Interpretação da característica de quase periodicidade em trechos sonoros, sugerindo uma forma de alterar o pitch no domínio do tempo.

Daí podemos ver porque o sinal de voz em um trecho sonoro é uma seqüência de sinais de curta duração (que correspondem à resposta impulsional do trato vocal) praticamente idênticos, e que ocorrem com um período igual ao período do trem de impulsos. Mesmo que a excitação não seja considerada como um trem de impulsos, o mesmo raciocínio pode ser aplicado, se ele for periódico, como é ilustrado na Figura 4.2.

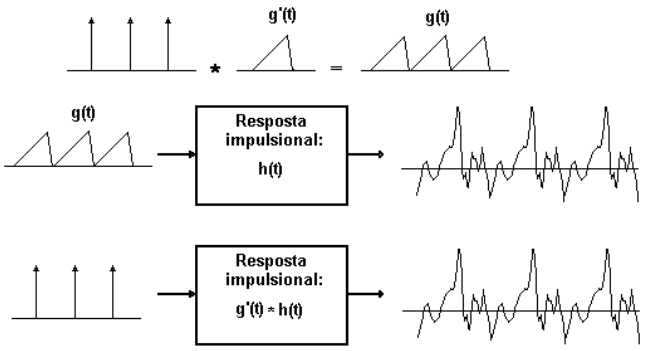


Figura 4.2: A interpretação é válida mesmo que a excitação não seja um trem de impulsos.

Esta discussão sobre a natureza do sinal de voz em trechos sonoros nos sugere uma forma de modificar a freqüência fundamental e a duração usando um algoritmo que atua no domínio do tempo. De fato, se alterarmos a distância entre os trechos que se repetem, alteraremos o pitch (isto é ilustrado na Figura 4.1), e se retirarmos ou inserirmos alguns trechos, alteraremos a duração. Assim, como a excitação pode ser arbitrária e não existe uma tentativa de aproximar a resposta impulsional do trato vocal por um filtro digital, é possível fazer modificações no sinal de voz que praticamente não alteram a qualidade da voz.

### 4.4.2 Análise

A análise consiste em obter uma representação intermediária do sinal de voz que consiste em sinais de curta duração, cada um deles associado a uma marca de pitch. Estes sinais são obtidos da seguinte forma: o sinal de voz é multiplicado por uma janela centrada em uma marca de pitch e com um comprimento de duas a quatro vezes o período de pitch local, ou seja, aproximadamente 50% a 75% de sobreposição entre janelas consecutivas. As janelas mais utilizadas são as de Hanning e Hamming. Esta operação é feita para cada marca de pitch obtida. A etapa de análise, então, leva a uma seqüência de sinais de curta duração, cada um deles associado a uma marca de pitch, e que podem ser descritos por

$$s_i(n) = s(n)w(n - iT_{0i}) \tag{4.5}$$

onde $s_i(n)$ é o sinal de curta duração, $s(n)$ é o sinal original, $w(n)$ é a janela de análise e $T_{0i}$ é o período local de pitch.

Para ilustrar o procedimento, a Figura 4.3 mostra o sinal original, uma das marcas de pitch e uma janela de análise centrada nesta marca de pitch, bem como o sinal de curta duração obtido para esta marca de pitch.

### 4.4.3 Síntese

Na etapa de síntese, o sinal sintetizado é obtido através de um procedimento OLA (*overlap-add*, ou soma com sobreposição): os sinais de curta duração são posicionados de tal forma que suas amostras centrais coincidam com as suas marcas de pitch e então são somados. A Figura 4.4 ilustra este procedimento, que é chamado de OLA simplificado.

De uma forma geral, o sinal obtido na etapa de síntese é:

$$\tilde{s}(n) = \sum_{i=-\infty}^{\infty} s_i(n - i(T_i - T_{0i})) \tag{4.6}$$

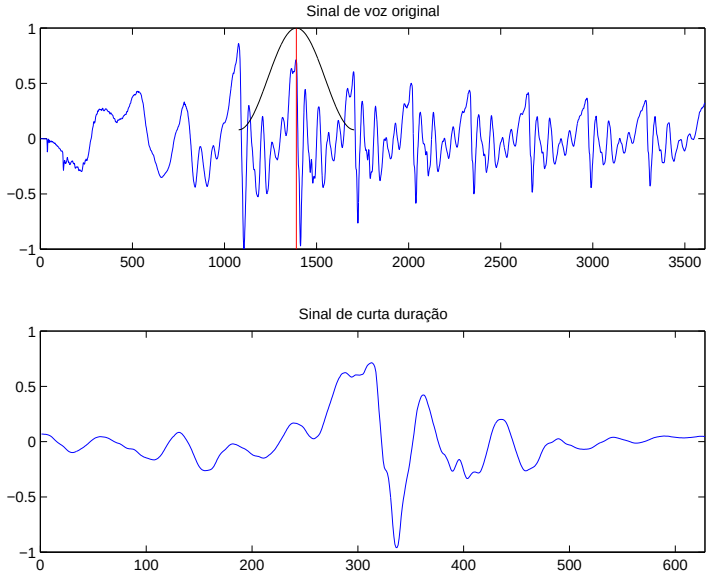


Figura 4.3: Sinal original com uma janela de análise centrada numa marca de pitch ($n = 1390$) e sinal de curta duração extraído.

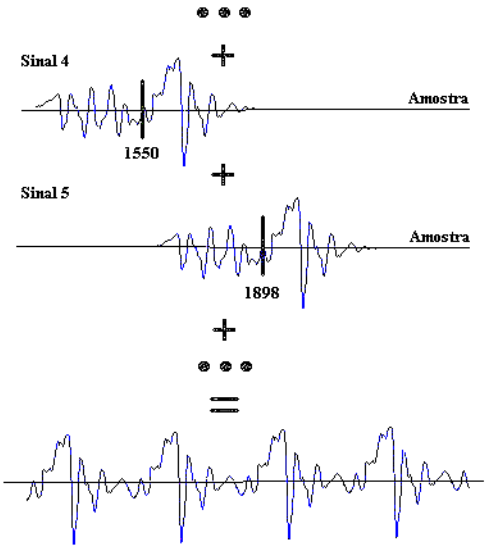


Figura 4.4: Procedimento de OLA simplificado.

onde $\tilde{s}(n)$ é o sinal sintetizado, $s_i(n)$ é o sinal de curta duração, $T_i$ é o novo período de pitch e $T_{0i}$ é o período de pitch original.

Consideremos, no entanto, a situação em que o sinal de voz é analisado e sintetizado, sem ser modificado. Neste caso, $T_i = T_{0i}$, e o procedimento de análise/síntese nos daria o sinal

$$\tilde{s}(n) = \sum_{i=-\infty}^{\infty} s(n)w(n - iT_{0i}) \tag{4.7}$$

que pode ser reescrito como

$$\tilde{s}(n) = s(n)\hat{w}(n) \approx Ks(n) \tag{4.8}$$

onde $\hat{w}(n)$ é uma função de ponderação periódica e $K$ é uma constante arbitrária. Ou seja, o sinal sintetizado teria igual a mesma forma do sinal original, que é o que desejamos, pois não houve modificação da representação intermediária. Isto é válido assumindo que o espectro de $\hat{w}(n)$ é suficientemente estreito, porque, neste caso, o espectro de $\hat{w}(n)$ poderia ser aproximado por um impulso, e o sinal no domínio do tempo poderia ser aproximado por uma constante.

Entretanto, é possível que para determinados fatores de modificações de pitch a aproximação utilizada na equação (4.8) não seja tão boa. Para compensar isto, existem sugestões de procedimentos de OLA [11][27] que utilizam uma soma ponderada. Como estes procedimentos não foram utilizados neste projeto, e como o aumento da qualidade é pequeno quando são utilizados [2], eles não serão descritos aqui.

### 4.4.4 Modificação

Na etapa de modificação, o pitch e duração são modificados através da alteração da representação intermediária.

Para alterar o pitch, as posições das marcas de pitch são alteradas de modo a corresponderem às posições associadas ao novo contorno de pitch. Por exemplo, para aumentar o pitch as marcas de pitch devem ser aproximadas de forma que a distância entre elas corresponda ao novo período de pitch. Por outro lado, para diminuir o pitch, as marcas de pitch devem ser separadas. Deve-se levar em conta o fato de que a alteração do pitch provoca naturalmente uma alteração na duração do sinal, que deve ser compensada, se necessário. A Figura 4.5 ilustra um mapeamento de marcas de pitch para alteração do pitch. A linha horizontal de cima representa o domínio temporal no sinal original, e a linha horizontal de baixo representa o domínio temporal no sinal sintetizado. Os traços verticais nas linhas horizontais representam

as marcas de pitch, e as linhas que ligam as marcas de pitch do sinal original às do sinal sintetizado representam o mapeamento.

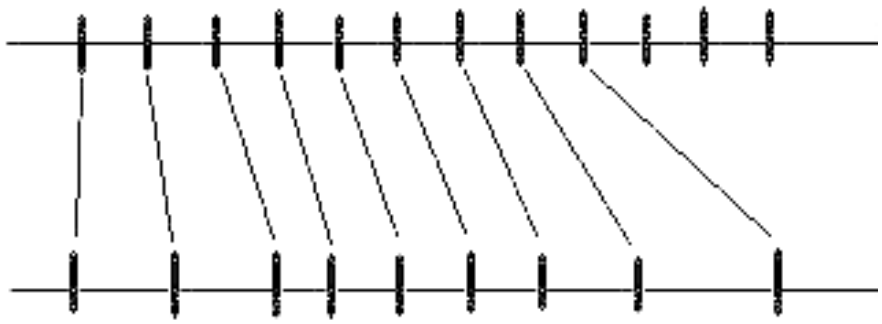

Figura 4.5: Representação de um mapeamento para modificar o contorno de pitch. Repare que a duração também é alterada.

Para alterar a duração, alguns sinais de curta duração da representação intermediária são repetidos ou removidos, para aumentar ou diminuir a duração, respectivamente. O número de sinais inseridos ou removidos depende do fator de modificação da duração desejado. A Figura 4.6 ilustra um mapeamento genérico de marcas de pitch para alterar o pitch e a duração do sinal de voz. As mesmas convenções utilizadas na figura anterior são válidas.

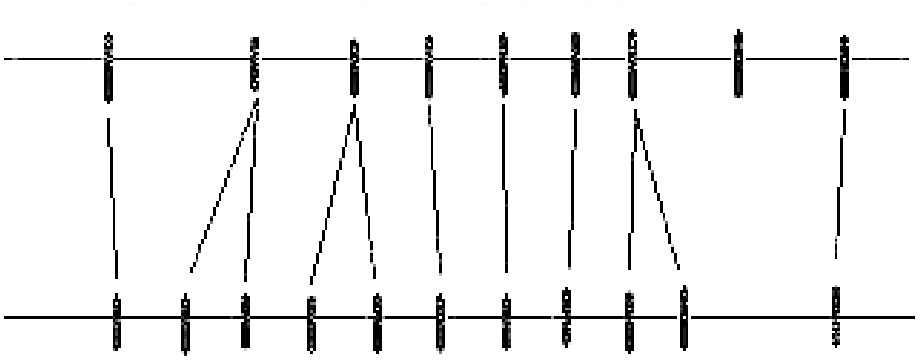

Figura 4.6: Um mapeamento genérico no qual o pitch e a duração são alterados.

## 4.5 Interpretação da etapa de modificação

Esta seção apresenta uma análise do efeito da etapa de modificação no domínio da freqüência. Como será visto, esta análise é interessante pois permite uma melhor compreensão das possíveis distorções causadas pela modificação. No artigo de Moulines e Charpentier [11], esta análise é apresentada de uma forma detalhada.

O sinal de voz $s(n)$ pode, nos trechos sonoros, ser dividido em uma componente determinística periódica $d(n)$ e em uma componente estocástica $e(n)$ que leva em conta variações do sinal entre os ciclos:

$$s(n) = d(n) + e(n) \tag{4.9}$$

Um dos objetivo da análise é descobrir o efeito da modificação do pitch sobre a componente estocástica do sinal. As seguintes suposições são feitas para facilitar a análise:

- A componente determinística $d(n)$ é periódica, de período $P$.
- As marcas de pitch de análise estão localizadas nos instantes

$$t_n = nP,\ n = 1, 2, ... \tag{4.10}$$

- O fator de modificação de pitch $\beta$ é constante e igual ao fator de modificação de tempo, de forma que as marcas de pitch de síntese serão

$$\tilde{t}_n = n\beta P \tag{4.11}$$

- É utilizado o procedimento de OLA simplificado e as janelas satisfazem à propriedade

$$\sum_{i=-\infty}^{\infty} w(n - iT_0) = K \tag{4.12}$$

que justifica a aproximação realizada na equação (4.8).

Podemos ver que o procedimento de OLA simplificado faz com que a componente determinística do sinal sintetizado possa ser interpretada como a convolução de um trem de impulsos de período $P$ por um sinal protótipo que é obtido pela multiplicação da janela de análise por um ciclo da componente determinística do sinal original, isto é:

$$\tilde{d}(n) = \delta_t(n, P) * x_0(n) \tag{4.13}$$

onde $x_0(n) = w(n)d(n)$, sendo $w(n)$ a janela de análise, e $\delta_t(n, P)$ é o trem de impulsos separados por um intervalo $P$.

Pensando no domínio da freqüência, vemos que o espectro do sinal sintetizado $\tilde{D}(\omega)$ é o produto de um trem de impulsos, separados por intervalos de freqüência $1/P$, pelo espectro do sinal protótipo $X_0(\omega)$, ou seja

$$\tilde{D}(\omega) = \delta_t(\omega, 1/P)X_0(\omega) \tag{4.14}$$

Como o sinal protótipo é o produto da janela pela componente determinística, o espectro do sinal protótipo é a convolução do espectro da janela $W(\omega)$ pelo espectro da componente determinística $D(\omega)$, isto é:

$$X_0(\omega) = W(\omega) * D(\omega) \tag{4.15}$$

Assim, o espectro do sinal protótipo é a *envoltória espectral* do sinal sintetizado. A envoltória espectral do sinal sintetizado depende da resolução em freqüência da análise, ou seja, do comprimento da janela de análise.

As situações de análise em banda estreita e em banda larga serão analisadas a seguir.

### 4.5.1 Condição de análise em banda estreita

No caso de uma janela de Hamming ou Hanning, a condição de análise em banda estreita é satisfeita se o comprimento da janela é maior que quatro vezes o período local de pitch [11]. Nesta condição, temos uma boa resolução espectral, como pode ser visto na Figura 4.7, que ilustra o sinal original, um trecho deste sinal obtido por janelamento, e o espectro do sinal de curta duração, onde podemos observar bem os harmônicos da excitação (estrutura fina).

É esperado que um algoritmo capaz de modificar o pitch modifique apenas a posição dos harmônicos da excitação, deixando a envoltória espectral do sinal sintetizado intacta. No entanto, no caso do TD-PSOLA em condição de análise em banda estreita, a envoltória espectral do sinal sintetizado é quase uma cópia do espectro do sinal original. Isto pode ser demonstrado através da equação (4.15), imaginando o caso extremo de análise em banda estreita, que é quando $W(\omega)$ é um impulso.

Assim, em certas freqüências, a amplitude da envoltória espectral do sinal sintetizado será menor que a amplitude da envoltória espectral original. Daí, podemos concluir que haverá atenuações seletivas em freqüência no sinal

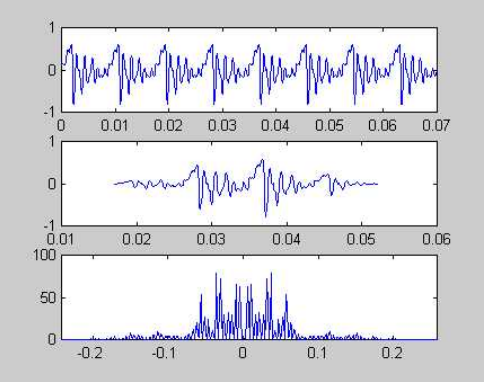


Figura 4.7: Análise sob a condição de banda estreita. É possível visualizar a estrutura fina (harmônicos da excitação), devido à alta resolução espectral.

sintetizado. Estas atenuações manifestar-se-ão como ecos com um atraso igual ao período de pitch original [11]. Essa distorção ocorre mesmo que os fatores de modificação de pitch sejam pequenos.

Como estas atenuações seletivas também ocorrem na componente estocástica, podemos interpretar a componente estocástica sintetizada como a saída de um *comb filter* tendo como entrada a componente estocástica original. A distorção resultante é percebida como um "ruído tonal" (assobio) [11].

### 4.5.2 Condição de análise em banda larga

A condição de análise em banda larga é satisfeita para janelas com comprimento menor que duas vezes o período local de pitch, para uma janela de Hamming ou Hanning [11].

Para esta condição, como não há resolução espectral suficiente para visualizar a estrutura fina, a envoltória espectral do sinal sintetizado é uma estimativa suavizada da envoltória do sinal de voz. Isto pode ser visto na Figura 4.8.

Como a envoltória espectral do sinal sintetizado é praticamente a mesma envoltória do sinal original, o problema de atenuação seletiva visto no caso da análise de banda estreita não ocorre. Porém, ocorre um aumento da largura de banda dos formantes devido à convolução do espectro da janela pelo espectro do sinal de voz.

No artigo de Moulines e Charpentier [11] é feita a sugestão de que, no caso de análise em banda larga, a distorção mínima de fase ocorre quando a janela de análise é sincronizada com o instante de fechamento da glote. Experiências

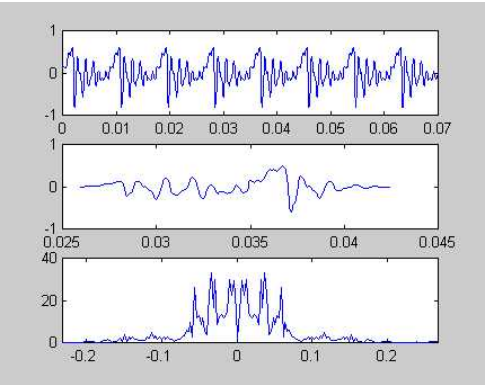


Figura 4.8: Análise sob a condição de banda larga. Por causa da baixa resolução de análise, obtemos uma estimativa suavizada da envoltória espectral do sinal original.

demonstraram que a voz sintetizada era percebida como "rouca" quando o deslocamento da janela em relação ao instante de fechamento da glote era maior que 30% do período de pitch.

## 4.6 Variantes do PSOLA

Nesta Seção serão descritas sucintamente as seguintes variantes do PSOLA: LP-PSOLA [11], FD-PSOLA [11] e MBR-PSOLA [2].

### 4.6.1 LP-PSOLA (*Linear Prediction PSOLA*)

A idéia do LP-PSOLA é armazenar a base de dados na forma de coeficientes de predição linear, juntamente com o resíduo da predição. É possível, usando versões codificadas do resíduo (a partir da abordagem multipulso ou de um *codebook*), diminuir a capacidade de armazenamento necessária para a base de dados.

No caso do LP-PSOLA, as etapas de análise, síntese e modificação não operam sobre a forma de onda no tempo, mas sobre o resíduo. Existe uma flexibilidade adicional que advém da representação usando LPC, pois podemos modificar a envoltória espectral do sinal através da modificação dos coeficientes de predição. Além disto, no LP-PSOLA não ocorre o problema de alargamento dos formantes na análise em banda larga porque a informação associada à envoltória espectral não passa pelas etapas de análise, modificação e síntese do PSOLA. É necessário que a análise seja em banda larga para preservar o achatamento espectral do sinal de excitação.

### 4.6.2 FD-PSOLA (*Frequency Domain PSOLA*)

Como foi visto na Seção 4.5, a etapa de modificação do TD-PSOLA provoca alterações indesejáveis na componente estocástica do sinal de voz.

No caso da análise em banda estreita, ocorrem distorções que se manisfestam como ecos. Uma forma de eliminar este tipo de problema é também modificando os sinais de curta duração antes da etapa de síntese, e não só as posições das marcas de pitch, como no TD-PSOLA.

No FD-PSOLA, os sinais de curta duração têm uma representação espectral formada por uma componente que é a *envoltória espectral* do sinal e por outra que é a *excitação* (como no modelo fonte-filtro). Essa representação é obtida da seguinte maneira: primeiro é calculada a DFT do sinal original multiplicado por janelas de análise que começam em cada marca de pitch. A envoltória espectral é estimada usando LPC ou análise cepstral. O sinal de excitação é obtido dividindo a DFT pela envoltória espectral encontrada. A modificação do pitch é feita mudando o espaçamento entre os harmônicos da excitação. A síntese consiste em converter o espectro modificado para o domínio do tempo através da IDFT, e fazer a soma com sobreposição. As janelas devem ter um comprimento grande, de forma que a análise seja em banda estreita, pois devemos ter resolução espectral para resolver os harmônicos da excitação. O problema de atenuação seletiva de harmônicos na condição de análise em banda estreita não ocorre no FD-PSOLA, porque a envoltória espectral é conservada.

Um outro conveniente do FD-PSOLA é a flexibilidade para alterações mais complexas no sinal de voz, já que podemos alterar, independentemente, as componentes de envoltória e excitação.

### 4.6.3 MBR-PSOLA (*MultiBand Resynthesis PSOLA*)

O TD-PSOLA apresenta alguns problemas com relação à possibilidade de descontinuidades de fase causadas por posicionamento não consistente das marcas de pitch e descontinuidades na envoltória espectral devido à sua natureza não paramétrica.

Uma solução para este problema é ressintetizar a base de dados com pitch constante e fase constante. Isto pode ser feito usando o modelo harmônico/estocástico, descrito brevemente na Seção 4.3. Se a fase nos trechos sonoros é constante, posicionar consistentemente as marcas de pitch deixa de ser um problema complicado.

Como o mapeamento da representação no domínio do tempo de um sinal para a sua representação espectral é linear, podemos concluir que a interpolação linear temporal dos segmentos da base de dados ressintetizada é

equivalente à interpolação espectral. Assim, o problema de descontinuidades espectrais na concatenação é resolvido com uma simples interpolação linear no domínio do tempo.

## 4.7 Implementação da classe PSOLA

O algoritmo TD-PSOLA foi implementado como uma classe em C++. Esta seção descreve os métodos e propriedades da classe.

### 4.7.1 Propriedades da classe

O algoritmo TD-PSOLA converte o sinal de voz, na fase de análise, em uma representação intermediária que consiste de uma seqüência de sinais de curta duração associados a marcas de pitch. Isto implica na necessidade das seguintes variáveis:

- Um vetor contendo as marcas de pitch (*pmarks).*
- Uma variável que que armazena o número de marcas de pitch (*np).*
- Um vetor de estruturas (*st*) com os seguintes campos:
  - Um vetor que contém as amostras dos sinais de curta duração (*segment).*
  - Uma variável que indica o número de amostras do sinal de curta duração (*size).*

### 4.7.2 Métodos da classe

#### Métodos públicos

A seguir é apresentada uma lista dos métodos públicos implementados.

- *analise*: Este método recebe o sinal de voz e a partir dele obtém a representação intermediária. Este é o protótipo da função:

  *void analise(WaveT& s, char *fmarks, double mu);*

  onde *s* é um objeto da classe *WaveT* que contém o sinal a ser modificado, *fmarks* é uma string contendo o nome do arquivo com as marcas de pitch, e *mu* é o fator de multiplicação da janela.

- *sintese*: É um método que sintetiza o sinal de voz a partir da representação intermediária. Este é o protótipo da função:

  *void sintese(WaveT& sint);*

  onde *sint* é um objeto da classe *WaveT* que armazenará o sinal sintetizado.

- *modifica*: Este método modifica o contorno de pitch do sinal para um contorno uniforme. O protótipo é:

  *void modifica(double pitch, double fs);*

  onde *pitch* é o novo valor de pitch, em Hz, e *fs* é a freqüência de amostragem em Hz.

- Métodos para obter o número de sinais de curta duração (*getnp*), e para obter ou ajustar o valor de uma marca de pitch (*getpm* e *setpm*). Estes métodos podem ser usados para fazer uma modificação arbitrária do sinal de voz. Os protótipos são:

  *int getnp(void);*
  *int getpm(int p);*
  *void setpm(int p, int mark);*

  onde *p* é o índice da marca de pitch no vetor de marcas de pitch, e *mark* é o seu valor.

### Métodos privados

Estes são os métodos privados implementados:

- *getstsignal*: Extrai do sinal original um sinal de curta duração associado a uma determinada marca de pitch. Chamado por *analise*.

- *hamming*: Retorna uma janela de Hamming. É chamado por *getstsignal*.

- *readmarks*: Lê as marcas de pitch do arquivo que as contém. É chamado por *analise*.

## 4.8 Conclusão

Neste capítulo foram descritos sucintamente alguns algoritmos de concatenação. O algoritmo TD-PSOLA foi descrito detalhadamente, e foram analisadas algumas distorções que podem ocorrer nas condições de análise em banda estreita e em banda larga. Foram apresentadas algumas variantes que possuem a vantagem de resolver alguns dos problemas apresentados pelo TD-PSOLA. Finalmente, foi apresentada a interface da classe, escrita em C++, que implementa o algoritmo.

# Capítulo 5

# Desempenho do sistema

## 5.1 Introdução

O objetivo deste capítulo é avaliar o módulo de concatenação através das seguintes comparações:

- Comparação entre os resultados obtidos com a base de sílabas e os obtidos a partir de dois sintetizadores baseados em difones. Estes foram desenvolvidos por alunos orientados pelo Prof. Márcio Nogueira de Souza. Um deles usa concatenação direta das unidades no domínio do tempo e o outro usa o modelo LPC.
- Comparação entre os resultados obtidos a partir da concatenação direta das sílabas com os obtidos a partir da concatenação usando o algoritmo TD-PSOLA.

## 5.2 Avaliação da qualidade de sistemas TTS

Não existe uma forma de definir um padrão de referência que permita a avaliação objetiva de enunciados gerados por um sistema de síntese. Simplesmente comparar uma seqüência de parâmetros extraídos de um enunciado sintetizado com a de um enunciado natural é inviável por causa da qualidade muito maior do enunciado natural e da imensa quantidade de fatores que afetam a qualidade do enunciado sintetizado. Além disso, a enorme variabilidade da voz humana faz com que enunciados com seqüências de parâmetros muito diferentes sejam muito parecidas perceptualmente.

O método descrito pelo aluno Alex Ribeiro Franco em seu projeto final [28] será usado para a avaliação da qualidade segmental do módulo de concatenação, ou seja, da sua inteligibilidade. Este método também será descrito

Tabela 5.1: Resultados obtidos para os vários sintetizadores

| **Sintetizador** | **Palavras** | **Frases** |
|---|---|---|
| Difones | 75,75% | 88,12% |
| Difones com LPC | 67% | 92% |
| Sílabas | | |
| Sílabas com TD-PSOLA | | |

aqui, por conveniência. Um dos motivos pelo qual este método foi escolhido é porque isso permitirá uma comparação direta com os sintetizadores baseados em difones, pois os resultados obtidos com esses sintetizadores usando este método foram documentados.

## 5.3 Descrição do método de avaliação e resultados

O método de avaliação consiste na utilização de palavras foneticamente balanceadas e de frases que as contenham. Os enunciados sintetizados são apresentados uma única vez aos ouvintes sem que eles tenham conhecimento prévio do material. O grau de entendimento de cada enunciado é computado para cada ouvinte.

Para o teste com palavras, é computado um ponto para cada palavra compreendida. Para o teste com frases, é computado um ponto se houver compreensão total de frase e meio ponto se houver compreensão parcial, ou seja, se mais de 70% da frase for entendida.

As palavras e frases usadas são listadas no Apêndice B. A tabela 5.1 demonstra os resultados obtidos, em termos de porcentagem de enunciados compreendidos.

Os enunciados usando difones e LPC e aqueles usando sílabas e TD-PSOLA foram gerados impondo um contorno de pitch constante, e sem se preocupar com a imposição de uma duração para os segmentos.

## 5.4 Conclusões

Neste capítulo foi descrito o método de avaliação do módulo de concatenação e os resultados obtidos. Observou-se que a qualidade segmental aumenta quando usamos sílabas no lugar de difones. Verificou-se também um

aumento da qualidade segmental quando utilizado o algoritmo TD-PSOLA para impor um contorno de pitch uniforme nos enunciados.

# Capítulo 6

# Conclusões

## 6.1 Conclusões Finais

Este projeto final descreveu a implementação tanto das ferramentas para obtenção da base de dados para um sistema TTS como do módulo de concatenação do sistema, e os resultados da avaliação da qualidade segmental do módulo de concatenação.

Os resultados mostram que há um aumento da qualidade segmental em relação aos difones quando se utilizam sílabas. No entanto, a menor qualidade segmental dos sintetizadores baseados em difones pode estar bastante relacionada a erros do módulo de transcrição fonética.

Também foi observado que o uso do algoritmo TD-PSOLA aumenta a inteligibilidade dos enunciados em relação à concatenação direta. Uma hipótese levantada pelo autor é de que a descontinuidade de pitch nas fronteiras dos segmentos funcione como um fator de distração, tirando a atenção do ouvinte.

O autor levanta a possibilidade de que uma maior qualidade segmental poderia ser obtida se a escolha das sílabas levasse em conta o contexto fonético. Considerações de contexto fonético, que são importantes devido à coarticulação entre sílabas, não foram feitas durante a construção da base de dados.

Durante a realização do projeto, o autor constatou a importância de ferramentas de manipulação de corpora linguísticos, como as ferramentas de anotação linguística. Essas ferramentas facilitam a análise do corpus e o intercâmbio de informações.

## 6.2 Sugestões de trabalhos futuros

- Transformar a ferramenta de recorte numa ferramenta de anotação. O autor teve algumas dificuldades na utilização de algumas ferramentas de transcrição disponíveis publicamente. Uma ferramenta de transcrição não só facilitaria o recorte das unidades como também facilitaria outras tarefas com a base de dados. Um exemplo seria a facilidade de avaliar um ADS, se fossem anotadas as decisões de sonoridade obtidas manualmente através de observação de parâmetros do sinal e da forma de onda do sinal.

- Melhorar a extração de pitch. Os principais problemas na extração de pitch são a decisão de sonoridade e a não utilização de um método de suavização do contorno de pitch. No caso da obtenção de marcas de pitch, teríamos ainda que usar algum método para encontrar o instante de fechamento glotal.

- Utilização de informações de contexto para auxiliar na segmentação. As sílabas foram obtidas sem nenhuma preocupação com o seu contexto fonético. No entanto, testes mostram que a coarticulação entre sílabas ainda é grande o suficiente para que nos preocupemos com os fones vizinhos. Além disso, algumas sílabas foram retiradas de palavras nas quais eram sílabas tônicas, e apresentam problemas quando utilizadas como sílabas átonas.

- Pode-se utilizar alguma técnica de compressão para diminuir o tamanho da base de dados. Uma sugestão simples é diminuir a taxa de amostragem de 44.1 kHz para 22.05 kHz, por exemplo. Esta modificação não deve alterar significativamente a qualidade segmental, mas reduziria pela metade o tamanho da base de dados.

# Apêndice A

# Base de dados

# Apêndice B

# Lista de Palavras e Frases

**Lista de Palavras**

1. Abacate
2. Morango
3. Carro
4. Zumbi
5. Salada
6. Didática
7. Gruta
8. Filho
9. Palhaço
10. Garoto
11. Canhão
12. Violeta
13. Jarro
14. Lapiseira
15. Notícia
16. Xícara
17. Lacrado
18. Plástico
19. Caderno
20. Profissão

**Lista de Frases**

1. A menina está bonita.
2. O professor deu uma bela aula hoje.
3. Todos gostaram da apresentação dos músicos.
4. A eleição para prefeito é amanhã.
5. A matéria do jornal foi bastante discutida.
6. O auditório ficou lotado para a comunicação do reitor.
7. Vila Lobos foi um dos melhores compositores de todos os tempos.
8. O super homem é um dos primeiros heróis dos quadrinhos.
9. O meu chefe foi almoçar com o presidente da empresa.
10. A minha avó cozinha divinamente bem.
11. A minha casa entrará em obras na próxima semana.
12. A professora foi elogiada por todos.
13. Todos ficaram surpresos com a sua humildade.
14. A música do nordeste está cada vez ganhando mais força.
15. O Brasil está entrando numa nova fase.
16. Brasília foi construída por Juscelino.
17. O futebol realmente é a paixão dos brasileiros.
18. Entre os esportes, o tênis e o golfe se destacam como os mais elegantes.
19. A Lua gira em torno da Terra.
20. Parece que agora tudo vai bem.

# Referências Bibliográficas


[1] T. Dutoit. A short introduction to text-to-speech synthesis. http://tcts.fpms.ac.be/synthesis/introtts.html.

[2] T. Dutoit. An introduction to text-to-speech synthesis. *Kluwer Academic Publishers*.

[3] A. L. Liberman. Minimal rules for synthesizing speech. *Journal of the Acoustical Society of America*, pages 1490–1499, 1959.

[4] I. H. Witten. Principles of computer speech. *Academic Press, Inc*, 1982.

[5] T. Dutoit. Text-to-speech synthesis: a comparison of four candidate algorithms. *Proceedings of the ICASSP*, I:565–568, 1994.

[6] R. C. de Oliveira. Comunicação pessoal.

[7] Building GUI's with Matlab, version 5. Documentação do Matlab.

[8] XML. http://www.w3.org/TR/ & http://www.oasis-open.org/cover/xml.html.

[9] Tools and formats for linguistic annotations. http://www.ldc.upenn.edu/annotation/.

[10] L. R. Rabiner & M. R. Sambur. An algorithm for determining the endpoints of isolated utterances. *The Bell Sytem Technical Journal*, 54(2), February 1975.

[11] E. Moulines & F. Charpentier. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication*, (9):453–467, 1990.

[12] H. Dudley. The vocoder. *Bell Labs Rec.*, 17:122–126, 1939.

[13] L. Schwardt. Voice conversion: an investigation. Master's thesis, University of Stellenbosch, December 1997.

[14] W. J. Hess. Advances in speech signal processing. *Marcel Dekker Inc. NY, Bassel, Hong Kong*. Chapter Pitch and Voicing Determination.

[15] W. J. Hess. Pitch determination of speech signals - algorithms and devices. *Springer-Verlag. Berlin*, 1983.

[16] J. R. Deller, Jr J. G. Proakis & J. H. L. Hansen. Discrete-time processing of speech signals. *Macmillan Publishing Company*, 1993.

[17] D. Callou & Y. Leite. Iniciação à fonética e à fonologia. *Jorge Zahar Editor Ltda*, 1990.

[18] D. S. Benincasa & M. I. Savic. Voicing state determination of co-channel speech. *Proceedings of the ICASSP*, pages 1021–1024, 1998.

[19] L. R. Rabiner, M. J. Cheng, A. E. Rosenberg & C. A. McGonegal. A comparative performance study of several pitch detection algorithms. *IEEE Transactions on Acoustic, Speech, and Signal Processing*, 24(5), October 1976.

[20] G. F. M. dos Santos. Síntese de voz síncrona com o período fundamental baseado na concatenação de difones. 1999. Seminário/projeto Ramo automação, controlo e instrumentação.

[21] H. Fujisaki, K. Hirose & K. Shimizu. A new system for reliable pitch extraction of speech. *Proceedings of the ICASSP*, 1986.

[22] M. M. Sondhi. New methods of pitch extraction. *IEEE Transactions on Audio and Electroacoustics*, 16:262–268, June 1968.

[23] A. V. Oppenheim & R. W. Schafer. Homomorphic analysis of speech. *IEEE Transactions on Audio and Electroacoustics*, 16:221–226, June 1968.

[24] L. R. Rabiner, M. R. Sambur & C. E. Schmidt. Applications of nonlinear smoothing algorithm to speech processing. *IEEE Transactions on Acoustic, Speech, and Signal Processing*, ASSP-23:552–557, 1975.

[25] J. Laroche, Y. Silyanou & E. Moulines. HNS: Speech modification based on a harmonic + noise model. *Proceedings of the ICASSP*, II:550–553, 1993.

[26] C. Hamon, E. Moulines & F. Charpentier. A diphone synthesis system based on time-domain modifications fo speech. *Proceedings of the ICASSP*, pages 238–241, 1989.

[27] J. B. Allen. Short term spectral analysis, synthesis, and modification by discrete Fourier transform. *IEEE Transactions on Acoustic, Speech, and Signal Processing*, ASSP-23:235–238, June 1977.

[28] A. R. Franco. Sintetizador paramétrico de voz para a língua portuguesa. 1999. Projeto Final.