Síntese e Reconhecimento da Fala Humana

Rumiko Oishi Stolfi

Trabalho Final de
Mestrado Profissional em Computação

Instituto de Computação
Universidade Estadual de Campinas

Síntese e Reconhecimento da Fala Humana

Rumiko Oishi Stolfi

Defendida em 31 de outubro de 2006

Banca Examinadora:

- **Prof. Dr. Fábio Violaro (Orientador)**
  Faculdade de Engenharia Elétrica e de Computação - UNICAMP
- **Prof. Dr. Carlos Alberto Ynoguti**
  Instituto Nacional de Telecomunicações
- **Prof. Dr. Neucimar Jerônimo Leite**
  Instituto de Computação - UNICAMP
- **Prof. Dr. Alexandre Xavier Falcão (Suplente)**
  Instituto de Computação - UNICAMP

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC-UNICAMP

Bibliotecária: Miriam Cristina Alves – CRB8a / 5094

Stolfi, Rumiko Oishi

St68s Síntese e reconhecimento da fala humana / Rumiko Oishi Stolfi – Campinas, [S.P.:s.n], 2006.

Orientadores: Fábio Violaro, Anamaria Gomide.

Trabalho final (mestrado profissional) – Universidade Estadual de Campinas, Instituto de Computação.

1. Sistemas de processamento da fala. 2. Processamento de sinais. 3. Reconhecimento automático da voz. 4. Síntese da voz. I. Violato, Fábio. II. Gomide, Anamaria. III. Universidade Estadual de Campinas, Instituto de Computação. IV. Título.

Título em inglês: Synthesis and recognition of human speech

Palavras-chave em inglês (keywords): 1. Speech processing systems. 2. Signal processing. 3. Automatic speech recognition. 4. Voice synthesis.

Área de concentração: Engenharia de Computação

Titulação: Mestre em Ciência da Computação

Banca examinadora: Prof. Dr. Fábio Violaro (FEEC-UNICAMP)
Prof. Dr. Carlos Alberto Ynoguti (INATEL)
Prof. Dr. Neucimar Jerônimo Leite (IC-UNICAMP)
Prof. Dr. Alexandre Xavier Falcão (IC-UNICAMP)

Data da defesa: 31/10/2006

Programa de Pós-Graduação: Mestrado Profissional em Engenharia de Computação

Síntese e Reconhecimento da Fala Humana

Este exemplar correponde à redação final do Trabalho Final, devidamente corrigido e defendido por **Rumiko Oishi Stolfi** e aprovado pela Banca Examinadora.

Campinas, SP, 31 de outubro de 2006

Prof. Dr. Fábio Violaro
Orientador

Profa. Dra. Anamaria Gomide
Co-orientadora

Trabalho Final apresentado ao Curso de Pós-Graduação em Ciência da Computação da Universidade Estadual de Campinas como requisito parcial para a obtenção do título de Mestre em Ciência da Computação, na área de Engenharia da Computação.

Síntese e Reconhecimento da Fala Humana

Rumiko Oishi Stolfi

Trabalho Final Escrito defendido e aprovado em 31 de outubro de 2006, pela Banca Examinadora composta por:

Prof. Dr. Fábio Violaro (Orientador)
Faculdade de Engenharia Elétrica e de Computação - UNICAMP

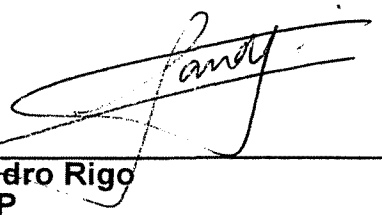
Prof. Dr. Carlos Alberto Ynoguti
Instituto Nacional de Telecomunicações

Prof. Dr. Neucimar Jerônimo Leite
Instituto de Computação - UNICAMP

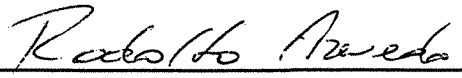
TERMO DE APROVAÇÃO

Trabalho Final Escrito defendido e aprovado em 28 de Agosto de 2006, pela Banca Examinadora composta pelos Professores Doutores:

Prof. Dr. Omar Carvalho Branquinho
PUC - Campinas



Prof. Dr. Sandro Rigo
IC - UNICAMP



Prof. Dr. Rodolfo Jardim de Azevedo
IC - UNICAMP

Dedico este trabalho aos meus pais, Massakazu e Miyako.

Meus agradecimentos:

à minha mãe, pela paciência, apoio e companhia;

ao meu esposo Jorge, pelo incentivo persistente, pelos valiosos esclarecimentos sobre processamento de sinais, e pelo incomensurável apoio durante a elaboração desse trabalho: na confecção da maioria dos gráficos, na disponibilização das bibliotecas de leitura e escrita de arquivos de áudio e na formatação do texto em LaTeX;

ao Prof. Fábio, meu orientador, que incansávelmente sempre esteve disposto a ceder o seu tempo, auxiliando-me com valiosas sugestões;

aos coordenadores do MP, em especial ao Prof. Alexandre, pela sua paciência e compreensão, e pela concessão de bolsa parcial;

à Profa. Anamaria Gomide, por estar sempre disposta a me socorrer;

aos analistas do IC Carlos Froldi e Éric Ostroski, por instalarem os softwares de que precisei para desenvolver meu projeto.

Resumo

O objetivo deste trabalho é apresentar uma revisão dos principais conceitos e métodos envolvidos na síntese, processamento e reconhecimento da fala humana por computador. Estas tecnologias têm inúmeras aplicações, que têm aumentado substancialmente nos últimos anos com a popularização de equipamentos de comunicação portáteis (celulares, *laptops*, *palmtops*) e a universalização da Internet.

A primeira parte deste trabalho é uma revisão dos conceitos básicos de processamento de sinais, incluindo transformada de Fourier, espectro de potência e espectrograma, filtros, digitalização de sinais, e o teorema de Nyquist.

A segunda parte descreve as principais características da fala humana, os mecanismos envolvidos em sua produção e percepção, e o conceito de fone (unidade lingüística de som). Nessa parte também descrevemos brevemente as principais técnicas para a conversão ortográfica-fonética, para a síntese de fala a partir da descrição fonética, e para o reconhecimento da fala natural.

A terceira parte descreve um projeto prático que desenvolvemos para consolidar os conhecimentos adquiridos neste mestrado: um programa que gera canções populares japonesas a partir de uma descrição textual da letra e música, usando o método de síntese concatenativa.

No final do trabalho listamos também alguns softwares disponíveis (livres e comerciais) para síntese e reconhecimento de fala.

Abstract

The goal of this dissertation is to review the main concepts relating to the synthesis, processing, and recognition of human speech by computer. These technologies have many applications, which have increased substantially in recent years after the spread of portable communication equipment (mobile phones, laptops, palmtops) and the universal access to the Internet.

The first part of this work is a revision of fundamental concepts of signal processing, including the Fourier transform, power spectrum and spectrogram, filters, signal digitalization, and Nyquist's theorem.

The second part describes the main characteristics of human speech, the mechanisms involved in its production and perception, and the concept of phone (linguistic unit of sound). In this part we also briefly describe the main techniques used for orthographic-phonetic transcription, for speech synthesis from a phonetic description, and for the recognition of natural speech.

The third part describes a practical project we developed to consolidate the knowledge acquired in our Masters studies: a program that generates Japanese popular songs from a textual description of the lyrics and music, using the concatenative synthesis method.

At the end of this dissertation, we list some available software products (free and commercial) for speech synthesis and speech recognition.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Nosso objetivo neste trabalho é apresentar uma revisão dos principais princípios envolvidos na síntese, processamento e reconhecimento da fala humana por computador. Descrevemos também o projeto experimental — um sintetizador de canções populares japonesas — que desenvolvemos no decorrer de nossos estudos.

A necessidade da interação do homem com a máquina através da fala já era evidente desde o início da era da computação [20]. A figura 1.1 dá uma idéia das expectativas no final da década de 1950 [6, 7].

Figura 1.1: Comunicação homem-máquina por interface de voz, na visão de Carl Barks (1958).

Em 1968, no filme *2001 — Uma Odisséia no Espaço*, Arthur C. Clarke e Stanley Kubrick [13, 14] idealizaram o computador HAL 9000 como sendo capaz de conversar. Esse filme fez acreditar que comunicação verbal entre o homem e o computador não só era possível, mas seria realidade muito em breve.

Entretanto, essas previsões se mostraram otimistas. É verdade que a tecnologia de síntese de fala avançou consideravelmente, a tal ponto que a fala artificial hoje é quase indistinguível da fala natural. Contudo o reconhecimento da fala humana ainda tem um longo caminho à frente, devido à complexidade da linguagem natural.

As pesquisas mal começam a unir síntese com reconhecimento de fala, objetivando aplicações como tradução em tempo real e interfaces amigáveis homem-computador. A popularização de equipamentos de comunicação portáteis (celulares, *laptops*, *palmtops*) e a universalização da Internet aumentaram consideravelmente o potencial de aplicação destas tecnologias.

1.1 Estrutura da monografia

O restante deste trabalho está dividido em três partes. A parte I é uma revisão dos principais conceitos de processamento de sinais:

- **Capítulo 2**: Apresenta os principais elementos da teoria de sinais contínuos, incluindo os conceitos de expansão, deslocamento, convolução, e sinais periódicos
- **Capítulo 3**: Revê os conceitos de transformada de Fourier, espectro de potência e espectrograma para sinais analógicos, e o conceito de função de janelamento.
- **Capítulo 4**: Revê o conceito de filtro de sinais e descreve os principais tipos de filtros analógicos.

- **Capítulo 5**: Trata do processamento digital de sinais, introduzindo os conceitos de amostragem, quantização, e reconstrução. Descreve as principais condições para digitalização de qualidade, incluindo o teorema da amostragem de Nyquist. Introduz os conceitos de transformada discreta de Fourier e transformada Z.

A parte II apresenta as características da fala humana, e descreve os principais métodos para conversão texto-fala e reconhecimento da fala natural:

- **Capítulo 6**: Discorre sobre a natureza do som em geral e da fala humana em particular, descrevendo os órgãos responsáveis pela produção da fala (trato vocal), sua captação (sistema auditivo), e os mecanismos físicos correspondentes.

- **Capítulo 7**: Introduz o conceito de fone (unidade elementar da fala) e suas características analíticas, incluindo espectro dos sons primordiais principais e dos principais tipos de fones. Introduz o conceito de formantes (picos no espectro de potência que caracterizam certos fones).

- **Capítulo 8**: Descreve as principais características e aplicações de sistemas de conversão texto-fala, incluindo sua estrutura geral e as principais dificuldades e soluções. Apresenta uma relação dos diferentes fones da língua portuguesa.

- **Capítulo 9**: Apresenta as principais aplicações e tecnologias para síntese da fala humana: síntese concatenativa, síntese por formantes, e simulação articulatória. Descreve em particular o modelo fonte-filtro baseado em formantes (bancos de ressonadores) e em filtros de predição linear (LPC). Descreve também a técnica PSOLA para concatenação suave de segmentos de fala e sua variação de duração e freqüência de "pitch".

- **Capítulo 10**: Enumera as principais aplicações para sistemas de reconhecimento da fala humana, e classifica as mesmas segundo vários atributos. Descreve brevemente as principais dificuldades do problema e algumas tecnologias utilizadas para sua solução, como redes neurais artificiais e cadeias de Markov.

A parte III descreve um projeto prático desenvolvido para consolidar os conhecimentos adquiridos neste mestrado:

- **Capítulo 11**: Descreve o sistema que implementamos, batizado `karacat`, que sintetiza canções populares japonesas, usando o modelo de síntese concatenativa.

Finalmente, nos **Apêndices**, listamos alguns softwares disponíveis (livres e comerciais) para síntese e reconhecimento de fala.

Parte I

Elementos de Processamento de Sinais

Capítulo 2

Análise de sinais

2.1 Sinais analógicos

O som é uma deformação de um meio elástico (por exemplo, uma variação da densidade e pressão do ar, ou da tensão e deformação de um sólido) que se propaga na forma de ondas. Veja a figura 2.1.

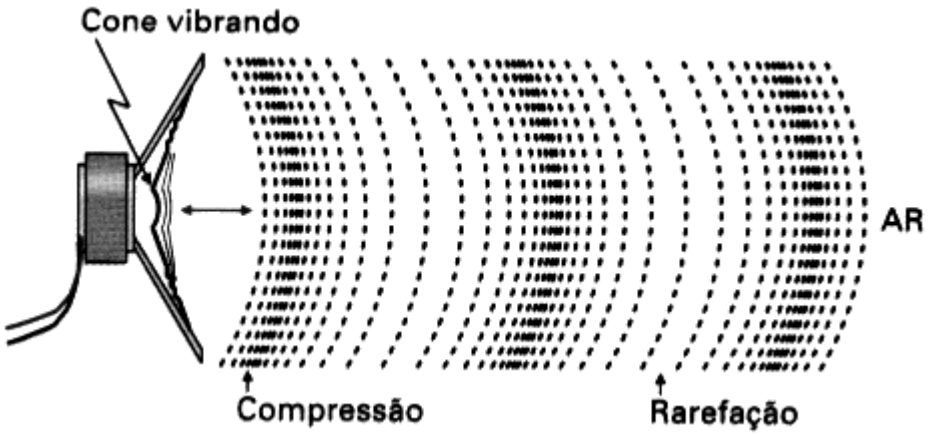


Figura 2.1: Propagação das ondas sonoras pelo ar.

Uma grandeza física que varia com o tempo, como a pressão do ar em um determinado ponto de uma onda sonora, pode ser descrita por um *sinal analógico*: uma função real s (pressão, corrente, tensão, deslocamento, etc.) de uma variável real t (tempo) com as seguintes características: (1) é uma função contínua, e (2) em qualquer intervalo de tempo, a integral do quadrado dessa função é finita.

Estas propriedades valem naturalmente para o som, pois: (1) a pressão varia de forma contínua, uma vez que as partes móveis da fonte sonora não podem se mover a velocidade infinita; e (2) a integral do quadrado da pressão é proporcional à energia emitida na forma de som, que é necessariamente finita.

2.2 Operações com sinais

Sinais podem ser matematicamente combinados com as operações de soma, subtração, produto, etc. Nesses casos entende-se que a operação é aplicada a valores tomados no mesmo instante. Por exemplo, a soma de um sinal f e um sinal g é um sinal $h = f + g$ tal que $h(t) = f(t) + g(t)$ para todo instante t.

Outras operações com sinais, importantes para processamento de som, são a *amplificação*, o *deslocamento*, a *expansão*, e a *convolução*.

2.2.1 Amplificação ou atenuação

A *amplificação* ou *atenuação* de um sinal f por um fator real α produz um sinal g tal que $g(t) = \alpha f(t)$ para todo t. Obviamente, quando $\alpha = 1$ o resultado é o próprio sinal f, e quando $\alpha = 0$ o resultado é o sinal nulo (que vale zero para todo instante t).

Esta operação multiplica a amplitude do sinal por $|\alpha|$. O nome *amplificação* é geralmente usado quando $\alpha > 1$, e *atenuação* quando $|\alpha| < 1$.

2.2.2 Deslocamento

O *deslocamento* de um sinal f por um tempo fixo τ produz um sinal g tal que $g(t+\tau) = f(t)$ para todo instante t. Isso equivale a dizer que $g(t) = f(t-\tau)$ para todo instante t. Ou seja, o resultado g é igual ao sinal f, exceto que atrasado pelo tempo τ (ou adiantado, se τ é negativo).

2.2.3 Expansão ou contração

A *expansão* ou *contração* de um sinal f por um fator real $\alpha \neq 0$ produz um sinal g tal que $g(\alpha t) = f(t)$ para todo instante t. Isso equivale a dizer que $g(t) = f(t/\alpha)$ para todo instante t. O nome *expansão* é mais apropriado quando $\alpha > 1$, e *contração* quando $\alpha < 1$.

2.2.4 Convolução

A *convolução* de duas funções f e g, escrita $f * g$, é definida pela fórmula

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t-\tau)\, d\tau \tag{2.1}$$

ou seja, o valor da função $f * g$ num instante t é uma combinação linear dos valores de f em todos os instantes τ, ponderados pelos valores $g(t-\tau)$. Mostra-se que a convolução é comutativa ($f * g = g * f$) e associativa ($f * (g * h) = (f * g) * h$).

O elemento-identidade da convolução é a *função impulso unitário* ou *função de Dirac*, denotada por δ. Por definição, $\delta * f = f$ para qualquer sinal f. Decorre desta definição que $\delta(t)$ é zero para todo $t \neq 0$, mas tem integral unitária em qualquer intervalo que contenha $t = 0$. Portanto δ não é propriamente uma função real, mas pode ser entendida como o limite de uma seqüência de funções reais contínuas $f_1, f_2, \ldots, f_n$, onde cada f_i tem integral unitária, e é nula fora de um intervalo J_i, que contém 0 e cuja largura tende a zero.

2.3 Sinais periódicos

Dizemos que um sinal analógico s é *periódico* se ele se repete indefinidamente: $s(t+T) = s(t)$, para algum $T > 0$ e para todo t.

O menor valor positivo T que satisfaz esta condição é chamado de *período fundamental* do sinal, e qualquer trecho do sinal com duração T é um *ciclo*. A *freqüência fundamental* de um sinal periódico é o número $f = 1/T$ de períodos fundamentais (ou ciclos) por unidade de tempo. A unidade SI de freqüência, 1 ciclo por segundo, é denominada *hertz* e abreviada Hz.

2.3.1 Senóides

Os exemplos clássicos de sinais periódicos são as funções seno e cosseno ($\operatorname{sen} t$ e $\cos t$), que tem período 2π. Elas são casos particulares de *senóides*, funções da forma

$$s(t) = M \operatorname{sen}(2\pi f t - \theta) \tag{2.2}$$

onde M, θ, e f são números reais arbitrários. O parâmetro M é a *amplitude*(o valor máximo) da senóide, e f é sua *freqüência*. O parâmetro θ é o *deslocamento de fase* da senóide. Veja a figura 2.2.

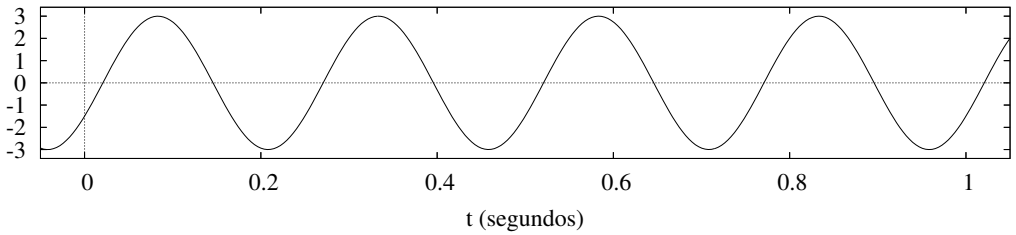


Figura 2.2: Uma senóide de freqüência $f = 4$Hz, deslocamento de fase $\theta = \pi/6$, e amplitude $M = 3$.

Em particular, a função cosseno $\cos(2\pi f t) = \operatorname{sen}(2\pi f t + \pi/2)$ é uma senóide com freqüência f, amplitude 1, e deslocamento de fase $\pi/2$. Mais genericamente, a função (2.2) pode ser escrita

também como uma combinação linear de $\text{sen}(t)$ e $\cos(t)$, contraídos pelo fator $1/(2\pi f)$:

$$M \,\text{sen}(2\pi f t - \theta) = A \cos 2\pi f t + B \,\text{sen}\, 2\pi f t \tag{2.3}$$

onde $A = -M \,\text{sen}\,\theta$ e $B = M \cos\theta$ (e portanto $M = \sqrt{A^2 + B^2}$).

Se f é zero, a função (2.2) tem valor constante $A = -M \,\text{sen}\,\theta$; caso contrário ela é uma função periódica, com freqüência fundamental f e período fundamental $T = 1/f$. Deve-se observar que uma senóide de amplitude M, freqüência f e deslocamento de fase θ também pode ser vista como tendo amplitude $-M$, freqüência $-f$ e deslocamento de fase $-\theta$.

Capítulo 3

A transformada de Fourier

Uma ferramenta essencial para o estudo de sinais analógicos é a teoria de Fourier, cujos conceitos principais descrevemos a seguir. Omitiremos detalhes e demonstrações, que podem ser encontradas em qualquer livro texto sobre o assunto [10].

3.1 Decomposição em senóides

A teoria de Fourier diz que todo sinal analógico, não necessariamente periódico, pode ser analisado como uma combinação linear de infinitas senóides de todas as freqüências possíveis, positivas ou nulas [10]. Usando a fórmula (2.3), esta afirmação equivale a dizer que, para todo sinal $s(t)$, existem funções $A(f)$ e $B(f)$ tais que

$$s(t) = \int_0^\infty (A(f)\cos 2\pi ft + B(f)\operatorname{sen} 2\pi ft)\, df \tag{3.1}$$

Os fatores $A(f)$ e $B(f)$ representam as amplitudes dos sinais $\cos 2\pi ft$ e $\operatorname{sen} 2\pi ft$, respectivamente, que, na análise de Fourier, contribuem para o sinal s. Cada senóide $A(f)\cos 2\pi ft + B(f)\operatorname{sen} 2\pi ft$ é, por definição, *a componente de s com freqüência f*.

3.2 Análise de Fourier complexa

As fórmulas da análise de Fourier ficam muito mais simples se trabalharmos com números complexos.

Definimos uma *senóide complexa* como sendo qualquer função da forma $Ce^{\mathbf{i}2\pi ft}$, onde C é algum número complexo, f algum número real (a freqüência), e $\mathbf{i}$ a unidade imaginária, $\mathbf{i} = \sqrt{-1}$. Veja a figura 3.1. O significado desta fórmula é dado pela *identidade de Euler*:

$$e^{\mathbf{i}\theta} = \cos\theta + \mathbf{i}\,\mathrm{sen}\,\theta \tag{3.2}$$

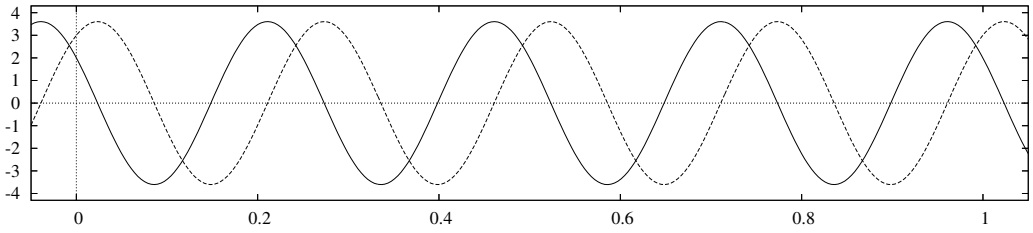


Figura 3.1: Uma senóide complexa com freqüência $f = 4$Hz e amplitude complexa $C = 2 + 3\mathbf{i}$. A parte real da função é a linha cheia, a parte imaginária é a linha tracejada.

Esta identidade permite escrever qualquer senóide de freqüência f como a soma de duas senóides complexas:

$$A(f)\cos 2\pi ft + B(f)\,\mathrm{sen}\,2\pi ft \;=\; S(f)e^{\mathbf{i}2\pi ft} + S(-f)e^{\mathbf{i}2\pi(-f)t} \tag{3.3}$$

onde

$$S(f) = \begin{cases} \frac{1}{2}(A(f) - \mathbf{i}B(f)) & \text{se } f > 0, \\ \frac{1}{2}(A(-f) + \mathbf{i}B(-f)) & \text{se } f < 0. \end{cases} \tag{3.4}$$

Portanto, podemos re-escrever a equação (3.1) como

$$s(t) = \int_{-\infty}^{\infty} S(f)e^{\mathbf{i}2\pi ft}\,df \tag{3.5}$$

Ou seja, todo sinal pode ser analisado como uma combinação linear de senóides complexas $e^{\mathrm{i}2\pi ft}$, de todas as freqüências possíveis (positivas e negativas), cada qual com determinado coeficiente $S(f)$.

3.3 Transformada de Fourier

Verifica-se que a função S da fórmula (3.5) pode ser calculada pela fórmula

$$S(f) = \int_{-\infty}^{\infty} s(t)e^{-\mathrm{i}2\pi ft}\,dt \tag{3.6}$$

A função S é chamada de *transformada de Fourier* do sinal s. A fórmula (3.5), que recupera a função original s a partir da transformada S, é chamada de *transformada inversa de Fourier*.

A teoria de Fourier nos permite representar o mesmo sinal físico de duas maneiras, no *domínio do tempo* (a função s) e no *domínio da freqüência* (a função S). A transformada de Fourier e sua inversa realizam a passagem de um domínio para o outro.

Cada operação com sinais realizável num domínio possui uma operação equivalente no outro domínio. Porém, certas operações são visualizadas ou mesmo efetuadas mais facilmente num domínio do que no outro.

3.4 Propriedades da transformada de Fourier

Seguem-se algumas propriedades importantes da transformada de Fourier. Sejam s, u, v sinais analógicos com transformadas S, U, V, e sejam α, β constantes reais.

Linearidade: Se $s(t) = \alpha u(t) + \beta v(t)$ para todo t, então $S(f) = \alpha U(f) + \beta V(f)$, e vice-versa. Ou seja, a transformada de Fourier (e sua inversa) são operações lineares.

Expansão/Contração: Se $s(t) = u(\alpha t)$, então $S(f) = U(f/\alpha)/\,|\alpha|$. Ou seja, se o sinal é encolhido no tempo, sua transformada expande em freqüência e diminui em amplitude.

Deslocamento: Se $s(t) = u(t - \alpha)$, então $S(f) = e^{-\mathbf{i}2\pi\alpha f}U(f)$. Ou seja, o deslocamento de um sinal no tempo não altera o módulo $|S(f)|$ de sua transformada, mas apenas altera o deslocamento de fase de cada componente, proporcionalmente à sua freqüência.

Teorema da energia (de Rayleigh): Para todo sinal s, tem-se

$$\int_{-\infty}^{+\infty} |s(t)|^2 \, dt = \int_{-\infty}^{+\infty} |S(f)|^2 \, df \tag{3.7}$$

Ou seja, a energia total do sinal pode ser calculada pela mesma fórmula (integral do quadrado da função), tanto no domínio do tempo, quanto no domínio da freqüência.

Produto/Convolução: Se $s(t) = u(t)v(t)$ para todo t, então $S = U * V$. Se $s = u * v$, então $S(f) = U(f)V(f)$, para todo f. Ou seja, a convolução de duas funções no domínio do tempo equivale ao produto ponto a ponto no domínio da freqüência, e vice-versa.

3.5 Espectro de potência

O *espectro de densidade de potência* de um sinal s é a função

$$\hat{S}(f) = |S(f)|^2 + |S(-f)|^2 \tag{3.8}$$

definida para $f \geq 0$, onde S é a transformada de Fourier de s. Informalmente, o valor de $\hat{S}(f)$ é a energia das componentes do sinal s que possuem freqüência $\pm f$. Veja a figura 3.2(d).

Vale observar que muitos autores preferem trabalhar com o *espectro bilateral de potência*, $\tilde{S}(f) = |S(f)|^2$, definido para todo f real, positivo e negativo. Veja a figura 3.2(c). Entretanto, o espectro bilateral de um sinal analógico real é sempre simétrico ($|S(-f)|^2 = |S(f)|^2$), pois nesse caso $S(-f)$ é o conjugado complexo de $S(f)$.

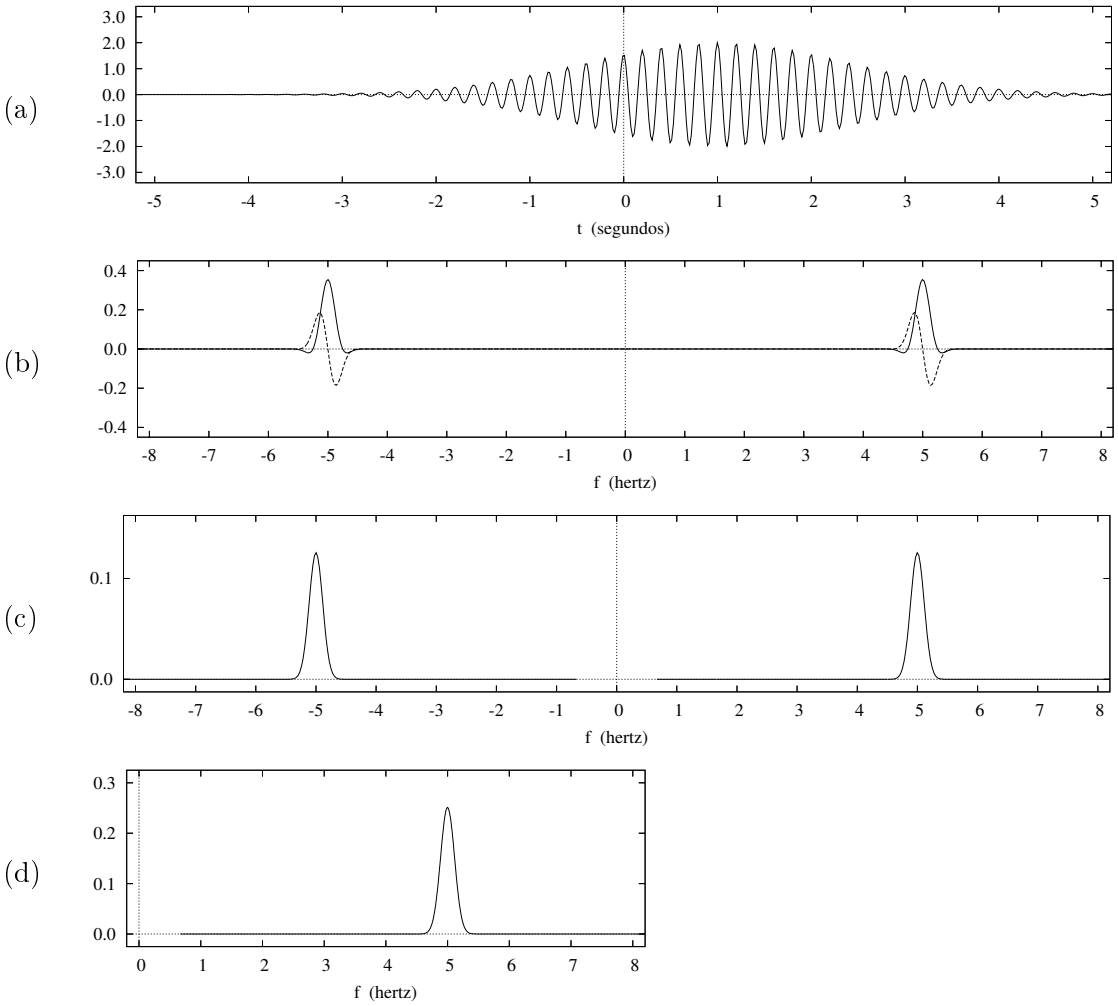


Figura 3.2: Transformada de Fourier e espectro de potência. (a) o sinal analógico $s(t) = 2\cos(10\pi(t-1))\exp(-(t-1)^2/4)$; (b) sua transformada de Fourier $S(f) = (\sqrt{\pi}/5)(e^{-2\pi^2(f+5)^2} + e^{-2\pi^2(f-5)^2})(e^{-\mathbf{i}2\pi f})$; (c) seu espectro bilateral de potência $\tilde{S}(f) = (\pi/25)(e^{-2\pi^2(f+5)^2} + e^{-2\pi^2(f-5)^2})^2$; (d) seu espectro (unilateral) de potência $\hat{S}(f) = (2\pi/25)(e^{-2\pi^2(f+5)^2} + e^{-2\pi^2(f-5)^2})^2$.

3.6 Espectrograma

O *espectrograma* é uma representação de um sinal analógico intermediária entre o domínio do tempo e o domínio da freqüência. Para construir o espectrograma de um sinal s, escolhe-se uma *função de janelamento* h. Esta função deve ser um sinal analógico cujo valor $h(t)$ é positivo quando t está dentro de determinado intervalo $(-a, +a)$, e zero para todo t fora

desse intervalo. Com esta escolha, o espectrograma de s é a função $\hat{S}$ de duas variáveis t, f, definida por

$$\hat{S}(t,f) = |S(t,f)|^2 + |S(t,-f)|^2 \tag{3.9}$$

onde

$$S(t,f) = \int_{-\infty}^{\infty} s(t+u)h(u)e^{-\mathbf{i}2\pi f u}\,du \tag{3.10}$$

Ou seja, para cada instante t, constrói-se um sinal que é um "extrato" de s, restrito ao intervalo de tempo $[t-a, t+a]$ e deslocado de modo a colocar o centro desse intervalo no instante 0. Isto é, constrói-se a função r tal que $r(u) = s(t+u)h(u)$ para todo u. Seja então R a transformada de Fourier do sinal r, e $\hat{R}$ seu espectro de potência. Temos então que $S(t,f) = R(f)$ e $\hat{S}(t,f) = \hat{R}(f)$ para cada freqüência f.

O valor de $\hat{S}(\tau, f)$ mede portanto a energia das componentes de freqüência $\pm f$ que estão presentes no trecho do sinal s restrito ao intervalo de tempo $[\tau - a, \tau + a]$. Veja a figura 3.3.

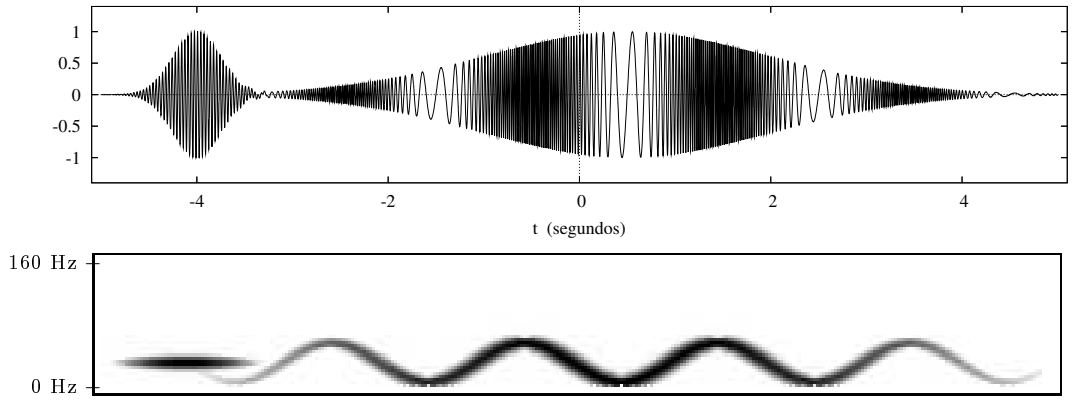


Figura 3.3: Espectrograma. Gráfico de um sinal analógico s (no alto) e seu espectrograma $\hat{S}(t,f)$, representado como uma imagem bidimensional, onde o eixo horizontal é o tempo t, e o eixo vertical é a freqüência f. Tons mais escuros representam valores maiores de $\hat{S}(t,f)$.

O espectrograma é uma ferramenta muito útil na análise de sinais cujo espectro de potência é bem característico quando analisado em trechos curtos, mas varia bastante em escalas de tempo maiores. Como veremos no capítulo 7, a voz humana tem essas características.

3.7 Funções de janelamento

A função de janelamento $h(t)$ mais simples é a *janela retangular*, que vale 1 se $-a < t < a$, e 0 caso contrário. Veja a figura 3.4.

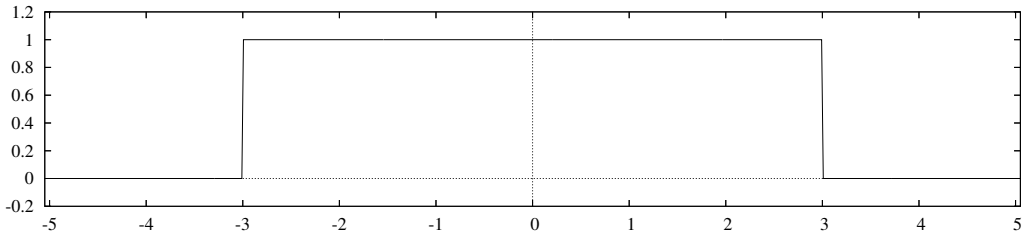


Figura 3.4: Função de janelamento retangular para o intervalo $[-3, +3]$.

Esta função é pouco usada na construção de espectrogramas, pois o produto $s(t)h(t-\tau)$ geralmente tem descontinuidades quando $t = \tau - a$ e $t = \tau + a$, que introduzem detalhes espúrios no espectrograma. Várias outras funções de janelamento podem ser encontradas na literatura: Gauss, Hamming, Hann, Bartlett, Bartlett-Hann, Nuttall, Kaiser, Blackman, Blackman-Nuttall, Blackman-Harris, Welch, e Parzen [72].

A função de Hann (popularmente, mas incorretamente, chamada "Hanning") é definida pela fórmula $h(t) = (1 + \cos(\pi t/a))/2$. Ela é muito usada, pois é fácil de implementar e produz espectrogramas de boa qualidade. Veja a figura 3.5.

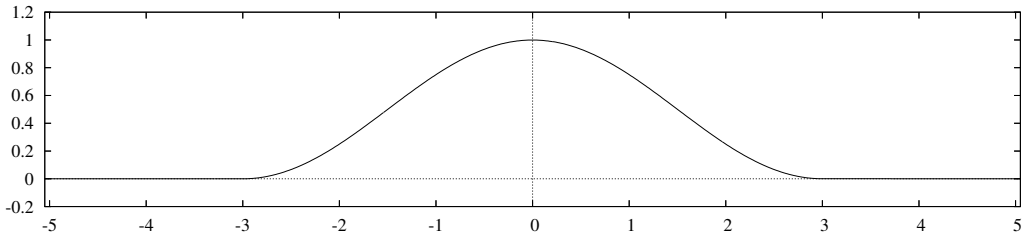


Figura 3.5: Função de janelamento de Hann para o intervalo $[-a, +a] = [-3, +3]$.

Capítulo 4

Filtros

Um *filtro*, na definição mais geral do termo, é um dispositivo que recebe um sinal s e devolve uma versão modificada s' do mesmo.

Na verdade, qualquer meio físico de transmissão (como uma parede de concreto ou madeira, um fio elétrico, ou mesmo o ar) sempre introduz alguma mudança não trivial no sinal, e portanto pode ser considerado um filtro.

Os filtros de interesse em áudio e telecomunicações são normalmente empregados para ressaltar, atenuar ou suprimir certas componentes do sinal, dependendo da freqüência. Eles são geralmente circuitos eletrônicos, mas há muitos exemplos importantes de filtros mecânicos, como por exemplo os ressonadores e cavidades de instrumentos musicais. Um filtro mecânico muito importante para este trabalho é o *trato vocal* (seção 6.4.1), que modifica os sons produzidos na laringe.

4.1 Filtros lineares e invariantes com o tempo

Dizemos que um filtro é *linear* se o sinal de saída depende de maneira linear do sinal de entrada. Isto é, para qualquer α e β constantes, se a entrada s produz a saída s', e a entrada

r produz a saída r', a entrada $\alpha s + \beta r$ deve produzir a saída $\alpha s' + \beta r'$.

Dizemos que um filtro é *invariante com o tempo* (ou apenas *invariante*) se o único efeito de um atraso arbitrário do sinal de entrada é um atraso igual na saída, ou seja, se a entrada s produz a saída s', e o sinal r é tal que $r(t) = s(t - \tau)$, para algum τ e para todo t, então a entrada r deve produzir o sinal r' tal que $r'(t) = s'(t - \tau)$.

4.2 Filtros para sinais complexos

Para estudar o efeito de filtros à luz da teoria de Fourier, é necessário definir seu efeito quando a entrada é um sinal complexo $s(t) + \mathbf{i}r(t)$ (uma função complexa da variável real t), possivelmente produzindo na saída outro sinal complexo, $s'(t) + \mathbf{i}r'(t)$. Para tanto, basta usar a seguinte regra: se o sinal real de entrada s produz a saída real s', então o sinal imaginário $\mathbf{i}s$ produz, por definição, a saída imaginária $\mathbf{i}s'$. Verifica-se que, com esta regra, um filtro que é linear e invariante para sinais reais também o é para sinais complexos.

4.3 Função de transferência

Demonstra-se que um filtro real, linear e invariante no tempo, quando alimentado com uma senóide $A\,\mathrm{sen}(2\pi ft - \theta)$, produz sempre outra senóide $A'\,\mathrm{sen}(2\pi ft - \theta')$; que pode diferir da entrada em amplitude e deslocamento de fase, mas *tem sempre a mesma freqüência.*

A mesma propriedade vale quando trabalhamos com exponenciais complexas. Mais precisamente, se a entrada de um filtro linear e invariante for a senóide complexa $e^{\mathbf{i}2\pi ft}$, de amplitude 1, a saída será outra senóide complexa $H(f)e^{\mathbf{i}2\pi ft}$, com a mesma freqüência f. Ou seja, o filtro pode apenas multiplicar o sinal por um número complexo arbitrário $H(f)$ — o que pode afetar seu módulo e seu deslocamento de fase, mas não sua freqüência.

O número $H(f)$ geralmente depende da freqüência f. Por linearidade, se a entrada for

uma senóide complexa geral $Ae^{\mathrm{i}2\pi ft}$, a saída será $H(f)Ae^{\mathrm{i}2\pi ft}$. Portanto, se conhecermos o valor de $H(f)$ para toda freqüência f, podemos determinar a saída s' para qualquer sinal de entrada s. Basta decompor s em suas componentes senoidais complexas, aplicar o filtro a cada uma delas, e combinar as senóides complexas resultantes. Ou seja, a transformada de Fourier S' da saída s' está relacionada à transformada S de s pela fórmula

$$S'(f) = H(f)S(f) \tag{4.1}$$

Concluímos portanto que a função H, chamada *função de transferência*, descreve completamente o efeito de um filtro linear e invariante no tempo, por mais complicado que ele seja, para qualquer sinal de entrada.

A função $H(f)$ é a transformada de Fourier da *resposta impulsiva* $h(t)$ do filtro, que é o sinal observado na saída do filtro quando a entrada é a função impulso $\delta(t)$ de Dirac.

Verifica-se que a grande maioria dos filtros, naturais e artificiais, é linear e invariante no tempo, pelo menos aproximadamente, desde que a amplitude do sinal não seja excessiva. Por outro lado, todo filtro físico deixa de ser linear quando o sinal excede um certo limite.

Deste ponto em diante, vamos supor implicitamente que todos os filtros são lineares e invariantes no tempo.

4.4 Filtros importantes

Entre os filtros mais importantes em acústica, estão os filtros *passa-baixas*, *passa-altas* e *passa-banda*, e os *ressonadores*.

Passa-baixas: é um filtro que permite a passagem sem alteração das componentes senoidais de baixa freqüência, mas elimina (ou reduz significativamente) as componentes com freqüências maiores que um determinado valor $f_{\max}$, denominado a *freqüência de corte*.

No filtro passa-baixas ideal, a função de transferência H é tal que $|H(f)| = 1$ quando $f < f_{\max}$, e $|H(f)| = 0$ quando $f > f_{\max}$. Porém, este tipo de filtro não pode ser realizado fisicamente; portanto os filtros passa-baixas usados na prática satisfazem estas condições apenas de maneira aproximada. Veja a figura 4.1.

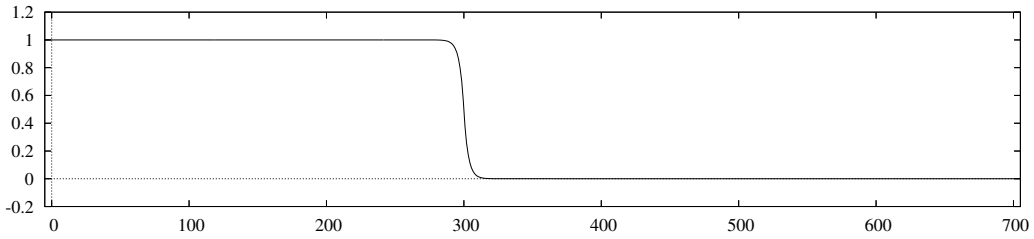


Figura 4.1: Função de transferência típica de um filtro passa-baixas com $f_{\max} = 300$ Hz.

Como veremos na seção 5.3.1, uma aplicação importante de filtros passa-baixas é a eliminação das componentes com freqüências altas antes da digitalização de um sinal. Outra aplicação é separar os sons graves de um sinal de áudio para alimentá-los a um alto-falante especializado (*woofer*). Na verdade, por limitações físicas, todo transdutor ou circuito eletrônico é incapaz de acompanhar senóides com freqüências acima de um certo valor. Portanto, pode-se supor que todo sistema físico inclui um filtro passa-baixas.

Passa-altas: Este filtro funciona de maneira complementar a um filtro passa-baixas, ou seja, ele elimina as componentes com freqüências menores que uma certa freqüência de corte $f_{\min}$, deixando passar inalteradas aquelas com freqüências maiores que $f_{\min}$. Veja a figura 4.2.

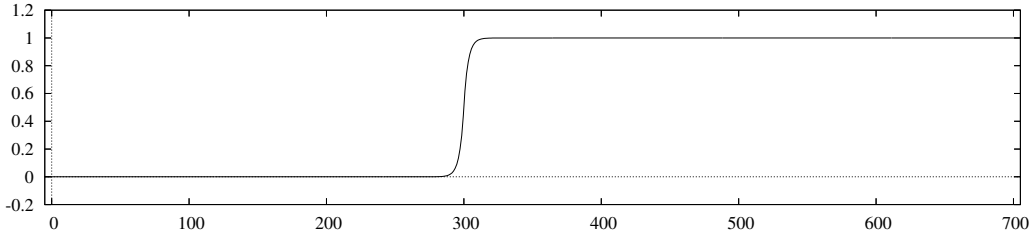


Figura 4.2: Função de transferência típica de um filtro passa-altas com $f_{\min} = 300$ Hz.

Uma aplicação de filtros passa-altas em acústica é eliminar componentes com freqüências menores que 20Hz (inaudíveis) antes da digitalização. Outra aplicação é separar os sons agudos para alimentá-los a um alto-falante especializado (*tweeter*).

Passa-banda: Um *filtro passa-banda* permite a passagem apenas de freqüências f dentro de uma determinada faixa, $f_{\min} < f < f_{\max}$. Ele combina os efeitos de um filtro passa-altas com corte $f_{\min}$ e um filtro passa-baixas com corte $f_{\max}$. Veja a figura 4.3.

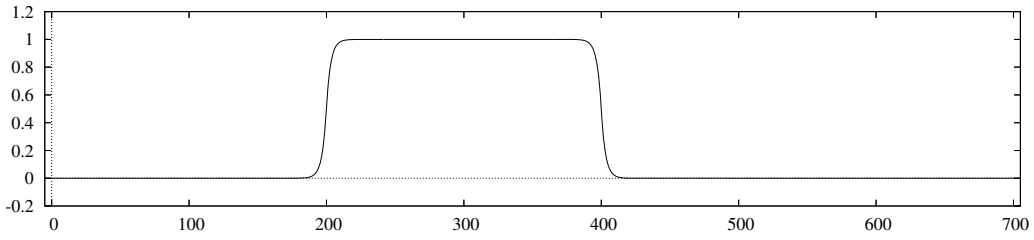


Figura 4.3: Função de transferência típica de um filtro passa-banda com $f_{\min} = 200$ Hz, $f_{\max} = 400$ Hz.

Como veremos na seção 6.3.1, o sistema auditivo humano inclui implicitamente um filtro passa-banda, cujas freqüências de corte são aproximadamente $f_{\min} = 20$ Hz e $f_{\max} = 20.000$ Hz.

Ressonador: é um caso especial de filtro passa-banda que possui $f_{\max}$ próximo a $f_{\min}$, de modo que preserva apenas as componentes com freqüência próxima a $f_{\text{med}} = (f_{\min} + f_{\max})/2$. Veja a figura 4.4.

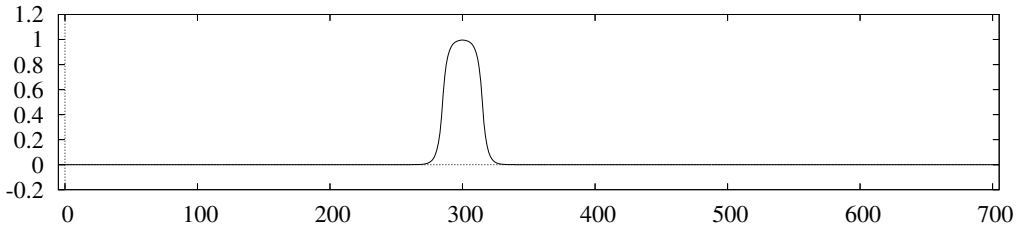


Figura 4.4: Função de transferência típica de um ressonador com $f_{\text{med}} = 300$ Hz.

Ressonadores são componentes importantes de instrumentos musicais. Por exemplo, cada tubo de um órgão é construído para ressonar na freqüência de uma determinada nota musical. No ser humano, a laringe funciona como um ressonador que, pelo seu alongamento ou contração, ajuda a controlar a freqüência de vibração das pregas vocais.

Anti-ressonador (*notch filter*): é um filtro que tem efeito complementar ao de um ressonador, ou seja, elimina as componentes de um sinal dentro de uma estreita faixa de freqüências, deixando passar todas as outras sem alteração. Veja a figura 4.5.

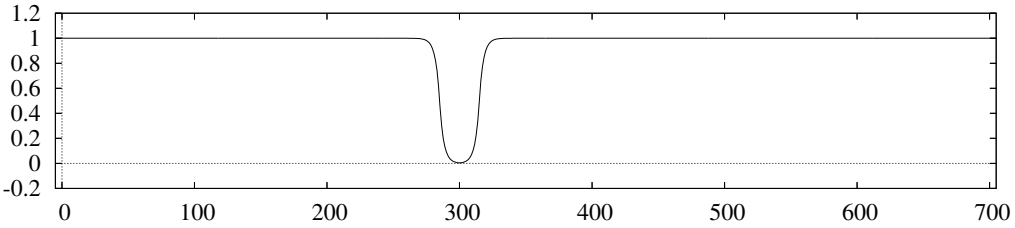


Figura 4.5: Função de transferência típica de um anti-ressonador com $f_{\text{med}} = 300$ Hz.

Anti-ressonadores são usados em sistemas de som, por exemplo, para eliminar *microfonia* (apitos causados pela realimentação do som do alto-falante no microfone).

Capítulo 5

Processamento digital de sinais

5.1 Introdução

Teoricamente, um sinal analógico existe desde $t = -\infty$ até $t = +\infty$, assume infinitos valores de amplitude em qualquer intervalo de tempo, e esses valores podem ser infinitamente próximos uns dos outros. Uma vez que computadores não conseguem armazenar ou manipular quantidades infinitas de dados, o sinal, para ser processado, precisa ser *digitalizado* — aproximado por uma coleção discreta de valores que possa ser codificada com um número finito de bits (zeros e uns).

A seqüência de valores $s_0, s_1, \ldots, s_{n-1}$ resultante desse processo é chamada de *sinal digital*, e cada valor s_i é uma *amostra digital* do sinal $s(t)$.

O processo inverso à digitalização é a *reconstrução* do sinal analógico $s(t)$ a partir do sinal digital $s_0, s_1, \ldots, s_{n-1}$. Esta reconstrução é necessária principalmente para que sons armazenados ou processados em forma digital possam ser tocados num alto-falante e ouvidos.

É óbvio que há uma infinidade de sinais analógicos distintos que produzem os mesmos valores depois de amostrado e quantizados. Portanto, o resultado da reconstrução, em geral,

é um sinal $s'(t)$ diferente do sinal $s(t)$ original — mas, espera-se, suficientemente similar para a aplicação considerada.

5.2 Digitalização

O processo de digitalização envolve três conceitos: recorte, amostragem e quantização.

O *recorte* de um sinal de áudio consiste simplesmente em limitar o tempo a um intervalo finito. Isto deve ser feito de preferência em instantes onde o sinal é nulo, pois caso contrário o salto repentino no valor é percebido com um estalo. Quando isso não é possível, pode-se usar uma função de janelamento, similar às descrita na seção 3.7, para "ligar" e "desligar" suavemente o sinal. Veja a figura 5.1(a–c). Um sinal de longa duração é freqüentemente recortado em uma série de segmentos de duração fixa, que são processados separadamente.

A *amostragem* consiste em substituir uma função de variável real $s(t)$ por uma seqüência finita de *amostras* — valores $s(t_0), s(t_1), \ldots, s(t_{n-1})$ medidos em instantes discretos $t_0, t_1, \ldots, t_{n-1}$ dentro do intervalo de recorte. Quase sempre os instantes são igualmente espaçados, por exemplo a cada 10^{-4} segundos. Veja a figura 5.1(d). O número de amostras por segundo é chamado *freqüência de amostragem*.

A *quantização* consiste em reduzir cada número real $s(t_i)$ a um valor s_i escolhido dentre um conjunto finito de valores possíveis — por exemplo, $\{-1,5\,, -1,2\,, -0,9\,\ldots, +1,2, +1,5\}$. Veja a figura 5.1(e). Um dispositivo que implementa este passo é chamado de *conversor analógico-digital* ou *conversor A-D*.

5.3 Condições para boa amostragem

No processo de digitalização e reconstrução, deve-se tomar cuidado para que o resultado $s'(t)$ reproduza o sinal analógico original $s(t)$ com fidelidade aceitável. Examinamos a seguir as

principais considerações sobre a amostragem relevantes para esse objetivo.

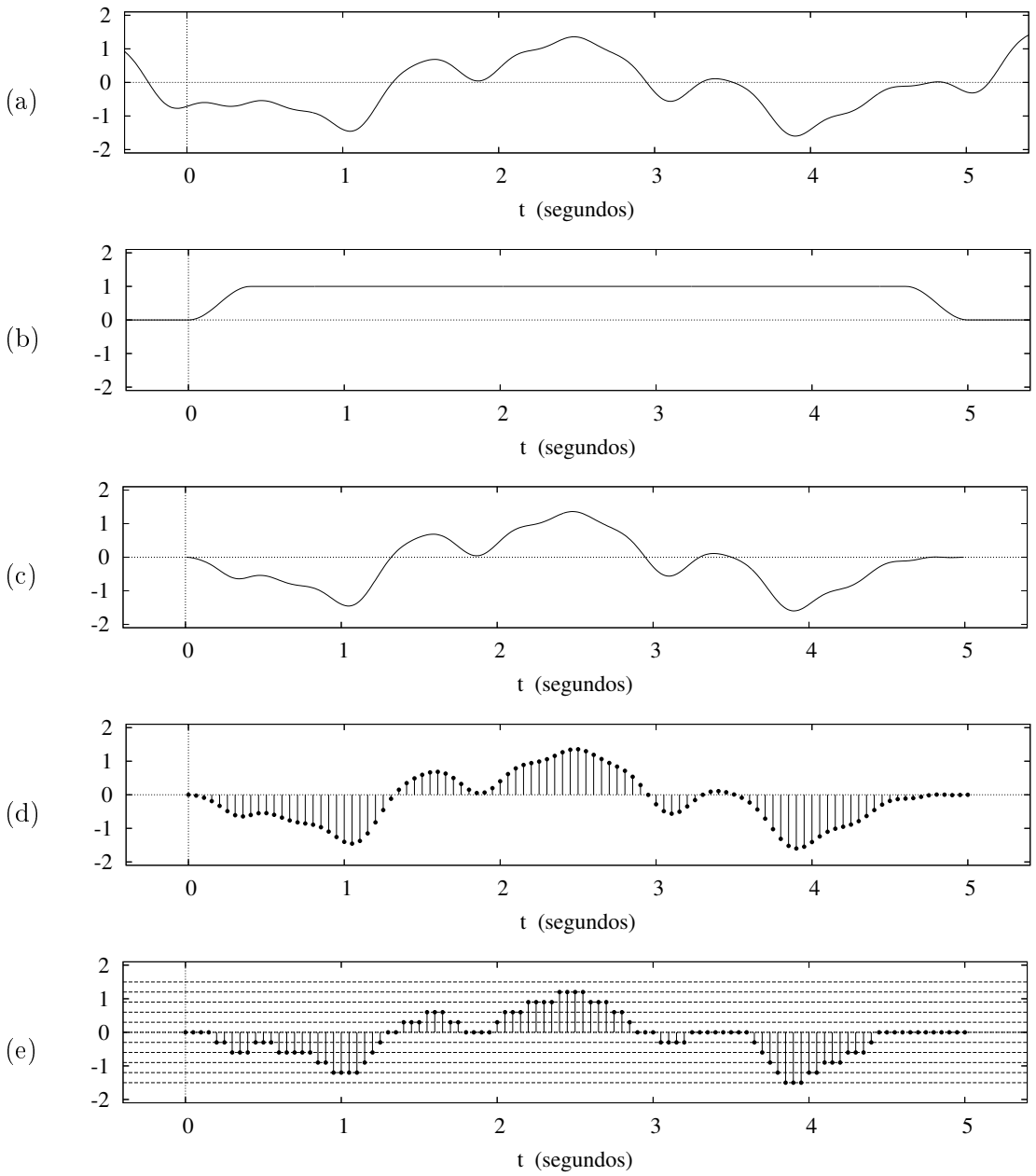


Figura 5.1: Digitalização de um sinal analógico. (a) Gráfico do sinal contínuo e entrada. (b) Uma função de janelamento para recorte suave. (c) O sinal recortado. (d) O sinal amostrado a cada 50 ms. (e) O sinal quantizado para 11 níveis igualmente espaçados entre $-1,5$ e $+1,5$.

5.3.1 Teorema da amostragem de Nyquist

O resultado mais importante para uma boa amostragem é o *Teorema de Nyquist*:

> *O sinal original s pode ser reconstruído exatamente a partir das amostras $s(t_i)$ se a freqüência de amostragem for maior que o dobro da maior freqüência das componentes presentes em s.*

Ou seja, se a freqüência de amostragem é f_*, a reconstrução perfeita é possível se $S(f) = 0$ para todo f com $|f| \geq f_*/2$.

Por outro lado, se as freqüências das componentes presentes em S cobrirem um intervalo $[-f_{\max}, +f_{\max}]$ onde $f_{\max} \geq f_*/2$, a reconstrução perfeita é impossível. Isto porque o sinal pode conter componentes $c(t) = e^{\mathrm{i}2\pi ft}$, de freqüência f, e $d(t) = e^{\mathrm{i}2\pi(f-f_*)t}$, de freqüência $f - f_*$, que produzem a mesma seqüência de amostras. Essa confusão (*aliasing*) entre as duas componentes implica que as amostras não contém informação suficiente para reconstruir o sinal original.

5.3.2 Pré-Filtragem

Em vista do teorema de Nyquist, conclui-se que, para garantir a reconstrução correta de um sinal arbitrário, é necessário remover as suas componentes com freqüências maiores ou iguais a $f_*/2$ *antes* da amostragem. Ou seja, o sinal deve passar por um filtro passa-baixas antes de ser alimentado ao conversor A-D.

Uma vez que o ouvido humano é sensível a freqüências entre 20 Hz e 20 kHz, a amostragem com freqüência f_* pouco maior que 40 kHz é adequada mesmo para os ouvidos mais exigentes. Por essa razão, em CDs de áudio comerciais o som é amostrado a 44.100 Hz. Verifica-se que a eliminação das componentes acima de 22.050 Hz não produz efeitos perceptíveis.

Por outro lado, verifica-se que as componentes da fala humana até 4kHz são geralmente

suficientes para compreensão de qualquer língua. Portanto, uma amostragem f_* de 8kHz é considerada suficiente para telefonia fixa e telefones celulares.

5.4 Condições para boa quantização

Na conversão de amostras reais $s(t_i)$ para valores discretos s_i, cada amostra $s(t_i)$ deve ser substituída por um valor s_i, dentro de um conjunto finito V de valores permissíveis. Geralmente V é *simétrico*: isto é, se o valor v pertence a V, então $-v$ também pertence.

Nesse processo ocorre um *erro de quantização* $e_i = s(t_i) - s_i$. O sinal reconstruído s' contém portanto um sinal indesejado $e(t)$, o *ruído de quantização*, que é o resultado da reconstrução da seqüência $e_0, e_1, \ldots, e_{n-1}$. Em sistemas de som, este ruído geralmente é percebido como um chiado sobreposto ao som original.

Os seguintes aspectos são importantes na quantização: o *espaçamento* dos valores, seu *alcance* e o *número de bits*.

5.4.1 Espaçamento dos valores

A grosso modo, o volume do ruído $e(t)$ é proporcional ao espaçamento entre os elementos de V. Por exemplo, se um valor v' de V e seu vizinho mais próximo v'' diferem em 1 mV, o erro de quantização, para amostras entre esses dois valores, será no máximo ± 0,5 mV. Portanto, para reduzir o erro $e(t)$, deve-se reduzir o espaçamento entre os valores de V.

5.4.2 Alcance dos valores

Os elementos de V também devem cobrir o intervalo de todos os valores que podem ocorrer no sinal s. Ou seja, o valor de $|s(t)|$ não deve exceder o valor $v_{\max} = \max\{\,|v| : v \in V\,\}$, o

alcance (*range*, em inglês) do quantizador. Caso contrário, a diferença $|s(t_i)| - v_{\max}$, com sinal apropriado, irá para o erro e_i.

Em sinais de som, o resultado desta condição de *sobrecarga* (*overload* ou *overflow*) é bastante desagradável. Para evitar que esta condição ocorra, é desejável que o valor de $v_{\max}$ seja o maior possível. Felizmente, uma vez que tanto a voz humana quanto transdutores (microfones e alto-falantes) têm potência limitada, os sinais de voz geralmente estão limitados a um intervalo $[-s_{\max}, +s_{\max}]$ conhecido.

5.4.3 Número de bits

Para codificar cada amostra digital, precisamos usar pelo menos $\log_2 |V|$ bits, arredondado para cima, onde $|V|$ é o número de valores distintos em V. Dito de outra forma, se cada amostra é codificada em b bits, o conjunto V terá no máximo 2^b valores distintos. Assim, por exemplo, se V é o conjunto dos inteiros entre -127 e 127, cada amostra necessita de 8 bits.

Junto com a freqüência de amostragem, este parâmetro determina a quantidade de dados que podem ser armazenados em qualquer meio digital (memória, disco rígido, CD, DVD, etc.), transmitidos (via cabo, Internet, telefone, etc.), e processados (por computadores ou dispositivos digitais especializados). Portanto, é desejável que este parâmetro seja o menor possível. Por outro lado, para compatibilidade com computadores e sistemas de transmissão de dados digitais, é comum fixar o número de bits por amostra em alguma potência de 2.

5.4.4 Escolha dos valores

A escolha mais natural para o conjunto V são os múltiplos inteiros de um valor fixo d; ou seja, $V = \{\, id : -\kappa \le i \le +\kappa \,\}$, sendo κ e i inteiros. Este esquema é chamado de *codificação linear*, e é o mais simples de processar e analisar.

Entretanto, em aplicações onde a economia de bits é importante (como telecomunicações

e gravadores portáteis), utiliza-se na prática um conjunto de valores V cujo espaçamento não é uniforme. A justificativa para esta *codificação não linear* é que, quanto mais intenso o sinal sonoro, mais tolerante é o ouvido humano ao ruído gerado pelos erros de quantização. Portanto, para se obter um determinado padrão de qualidade subjetiva, os valores de V próximos a zero precisam ter espaçamento menor que os valores mais afastados de zero.

Assim, por exemplo, o padrão de quantização conhecido como *lei* μ (*mu-law*), desenvolvido pelos Laboratórios Bell [18], usa os valores $V = \{v_i : -\kappa \leq i \leq \kappa\}$, definidos pela fórmula

$$v_i = \operatorname{sgn}(i)\frac{(1+\mu)^{|i/\kappa|} - 1}{1+\mu} s_{\max} \tag{5.1}$$

onde o inteiro κ é um parâmetro do modelo, e $\mu = 2\kappa+1$. Este padrão é usado para telefonia nos EUA e Japão, com 8 bits por amostra, $\kappa = 127$, e $\mu = 255$. Veja a figura 5.2.

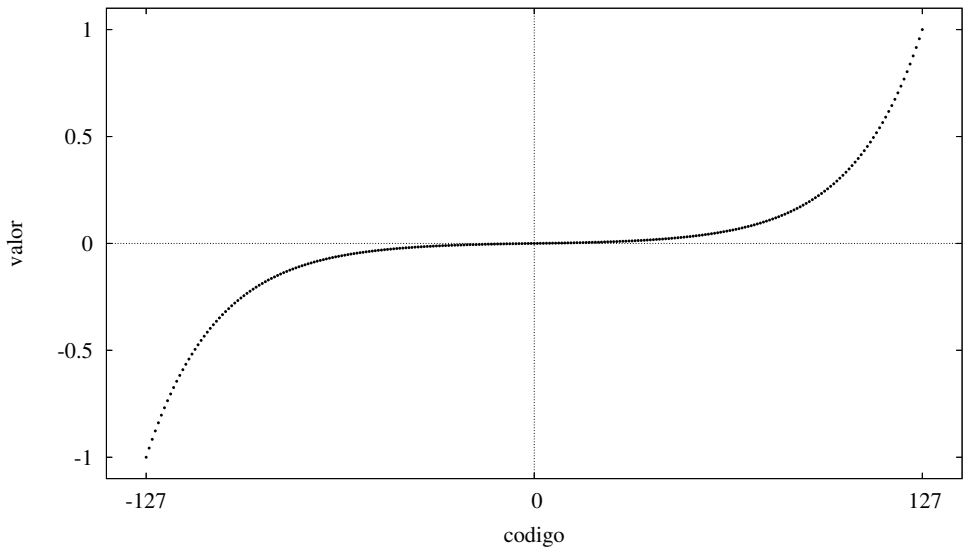


Figura 5.2: A correspondência entre o código numérico i e o respectivo valor do sinal v_i no esquema de codificação "lei μ" para 8 bits ($\mu = 255$).

Os padrões de telefonia da Europa e do Brasil especificam um sistema bastante semelhante, conhecido por *lei A* (*A-law*) [17]. Com qualquer dos dois sistemas, 8 bits por amostra são adequados para compreensão das palavras, desde que o volume geral da voz seja devidamente ajustado. A qualidade subjetiva do som reconstruído é similar à obtida por codificação linear com 13 bits por amostra.

Para aplicações mais exigentes (como gravação e transmissão de músicas, trilhas sonoras de filmes e televisão, etc.), 8 bits por amostra não são suficientes, mesmo com codificação não-linear. O padrão atual de codificação para CDs de áudio usa 16 bits por amostra, com codificação linear.

5.4.5 Digitalização como filtragem

Como veremos na seção 5.4.6, a estrutura concreta de um sistema eletrônico para conversão analógico-digital é determinada por limitações da física dos dispositivos usados. Matematicamente, porém, o processo de digitalização pode ser descrito como um filtro não-linear, composto de três módulos. O primeiro módulo é o filtro passa-baixas, que elimina componentes indesejadas. O segundo módulo, que representa a amostragem do sinal, multiplica o sinal filtrado por um *trem de impulsos de Dirac*, uma função Ш definida por

$$\text{Ш}(t) = \sum_{i=-\infty}^{+\infty} \delta(t - t_i) \tag{5.2}$$

onde t_i são os instantes de amostragem. O terceiro módulo representa os erros de quantização e_i introduzidos pelo conversor A/D, e cujo efeito é somar a cada impulso $s(t_i)\delta(t-t_i)$ o termo adicional $e_i\delta(t - t_i)$. Veja a figura 5.3.

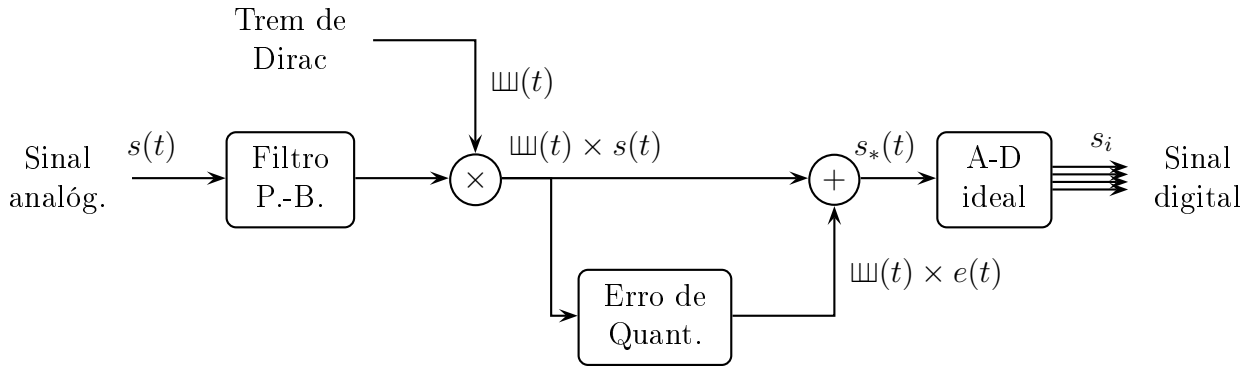


Figura 5.3: Esquema conceitual da digitalização vista como filtragem.

Conceitualmente, a saída deste *filtro de digitalização* é o trem de impulsos

$$s_*(t) = \sum_{i=-\infty}^{+\infty} s_i \delta(t - t_i) \tag{5.3}$$

cujas amplitudes são as amostras digitalizadas $s_0, s_1, \ldots$. Estes valores estão no conjunto V, e portanto o sinal $s_*(t)$ pode ser precisamente codificado em formato digital (bits) sem nenhuma perda ou alteração.

Este modelo tem o propósito de explicitar as três alterações efetivamente sofridas pelo sinal na digitalização, e separá-las da conversão de formato (analógico para binário) propriamente dita, que não afeta o sinal. Em particular, este modelo mostra que transformada de Fourier do sinal digitalizado s_* é

$$S_*(f) = (S(f)B(f) + E(f)) * \text{Ш}(f) \tag{5.4}$$

onde $S(f)$ é a transformada do sinal original s, $B(f)$ é a função de transferência do filtro passa-baixas, $\text{Ш}(f)$ é a transformada do trem de impulsos $\text{Ш}(t)$, e $E(f)$ é é a transformada de um sinal e tal que $e(t_i) = e_i$.

5.4.6 Digitalização na prática

Na prática, um dispositivo para digitalização de sinais geralmente consiste de um circuito analógico de amostragem e estabilização (*sample-and-hold*) seguido do conversor analógico-digital propriamente dito, como ilustrado na figura 5.4.

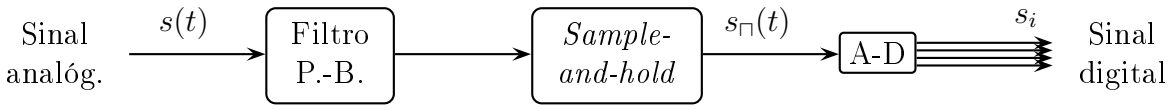


Figura 5.4: Esquema de blocos de um conversor analógico-digital típico.

O circuito *sample-and-hold* amostra o sinal de entrada s a cada instante t_i, e reproduz esse valor no sinal de saída $s_\sqcap$ até o próximo instante de amostragem; ou seja, $s_\sqcap(t) = s(t_i)$ durante

cada intervalo de t_i a t_{i+1}. O gráfico do sinal de saída $s_\sqcap(t)$ é portanto uma seqüência de degraus que acompanham aproximadamente o sinal de entrada. Veja a figura 5.5.

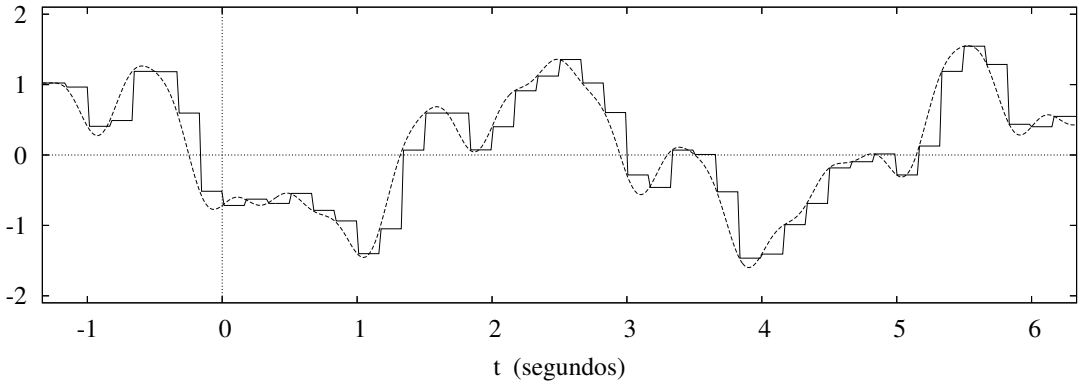


Figura 5.5: Um sinal analógico (linha tracejada) e a saída do circuito *sample-and-hold* (linha cheia).

O papel do circuito *sample-and-hold* é manter o sinal analógico estável até que o conversor A-D consiga determinar a representação binária do valor $s(t_i)$. Nesse momento, os sinais elétricos binários que representam os bits desse valor são "lidos" pelo computador ou outro sistema digital, e o processo todo se repete com a próxima amostra.

Na prática, é impossível construir um dispositivo *sample-and-hold* capaz de medir exatamente o sinal de entrada s no instante t_i apenas. Um circuito fisicamente realizável consegue apenas obter uma média aproximada dos valores de s nas vizinhanças de t_i.

A saída s' do circuito *sample-and-hold* pode ser escrita como

$$s'(t) = (s \times \text{Ш}) * \sqcap(t/p) \tag{5.5}$$

onde $\sqcap$ é o pulso retangular de altura 1 e duração 1.

5.5 Reconstrução

É óbvio que a reconstrução somente pode ser feita dentro do intervalo de recorte $[a, b]$. O primeiro passo da reconstrução é converter cada número digital s_i para uma representação analógica $s(t_i)$. Esta conversão é efetuada por um *conversor digital-analógico* ou *conversor D-A*. Feito isso, é necessário *interpolar* esses valores, ou seja, definir $s(t)$ para os demais instantes t.

5.5.1 Reconstrução como filtragem

Matematicamente, o processo de reconstrução pode ser concebido como uma seqüência de duas etapas separadas. Na primeira etapa, a seqüência de números $s_0, s_1, \ldots$ é convertida num sinal s_* que consiste de uma seqüência de pulsos de Dirac, onde o i-ésimo pulso ocorre no instante t_i e tem intensidade s_i:

$$s_*(t) = \sum_i s_i \, \delta(t - t_i) \tag{5.6}$$

Na segunda etapa, o sinal s_* passa por algum filtro "suavizador" (o *filtro de reconstrução*) que converte os impulsos num sinal contínuo Veja a figura 5.6.

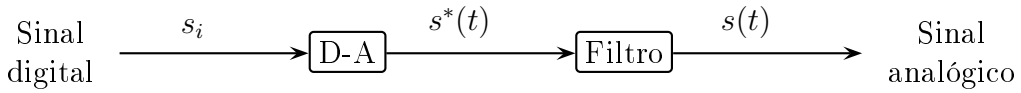


Figura 5.6: Esquema da reconstrução de um sinal digital vista como filtragem.

Esta interpretação é vantajosa sempre que o processo de reconstrução é linear e invariante com o tempo, pois permite descrever precisa e sucintamente o efeito do mesmo pela função de transferência $R(f)$ do filtro usado na segunda etapa. Ela é importante também para o projeto do filtro suavizador. Idealmente, o sistema de reconstrução (figura 5.6) deveria desfazer o efeito do sistema de digitalização (figura 5.3), na medida do possível.

Uma vez que, nesta abordagem, os conversores A-D e D-A se cancelam perfeitamente, eles podem ser ignorados. Segue-se que a escolha do filtro de reconstrução R depende do filtro passa-baixas B usado antes da digitalização, e das propriedade estatísticas do sinal de erro de quantização $e(t)$. Por exemplo, se B é um filtro passa-baixas ideal (retangular) com $f_{\max}$ igual à freqüência de Nyquist, verifica-se que o filtro R deve ser esse mesmo filtro. Nesse caso, o sinal reconstruído é dado pela fórmula

$$s(t) = \sum_i s_i \operatorname{sinc}\left(\frac{t - t_i}{p}\right) \tag{5.7}$$

onde p é o passo de amostragem, e sinc é a função definida por

$$\operatorname{sinc}(t) = \frac{\operatorname{sen} \pi t}{\pi t} \tag{5.8}$$

Veja a figura 5.7.

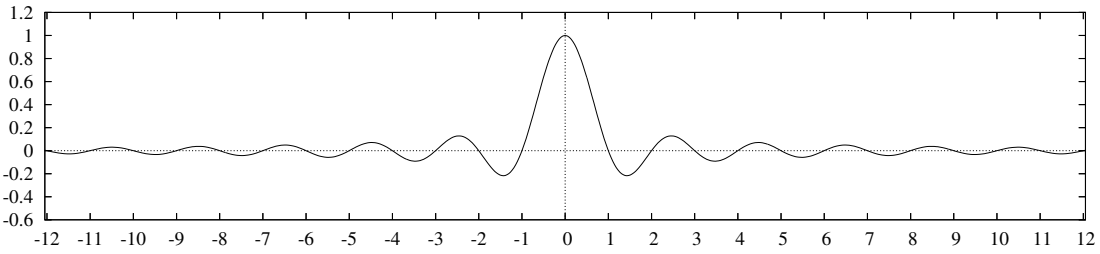


Figura 5.7: A função sinc(t).

É importante notar que a fórmula de reconstrução (5.7) vale apenas quando o filtro passa-baixas B do digitalizador é o filtro retangular ideal, e os erros de quantização são desprezíveis. Se B tem uma transição gradual nas vizinhanças de $f_{\max}$ (como sempre ocorre na prática), ou se os erros de quantização são significativos, a reconstrução deve deve usar um filtro R específico — o que implica em substituir a função sinc na fórmula (5.7) por alguma outra função.

5.5.2 Reconstrução na prática

Na prática, o circuito eletrônico que faz a conversão digital-analógica produz uma aproximação por degraus do sinal discreto, ilustrada na figura 5.8, cujo valor em cada instante é a amostra mais recente. Ou seja, a saída do conversor D-A é um sinal $s_\sqcap$ tal que $s_\sqcap(t) = s_i$, onde i é o inteiro tal que $t_i \leq t \leq t_{i+1}$.

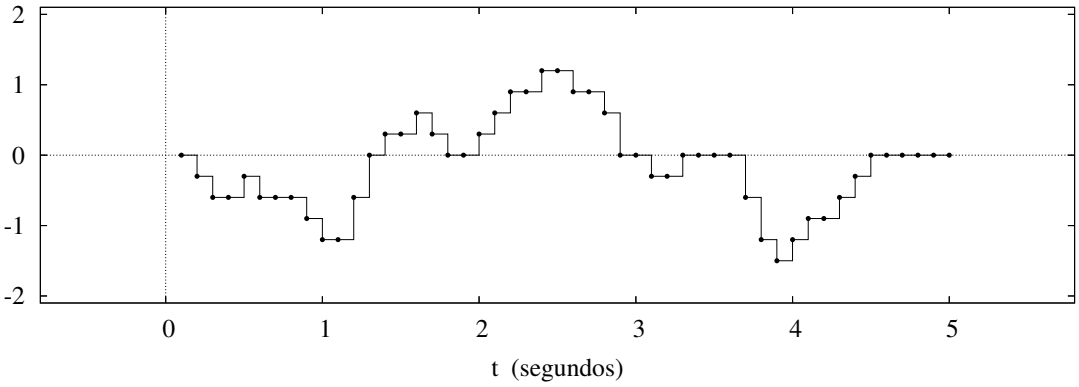


Figura 5.8: Um sinal discreto (pontos) e sua reconstrução retangular (linhas).

Em situações normais, a diferença entre a saída bruta $s_\sqcap$ do conversor D-A e o sinal original s não pode ser ignorada. A mudança abrupta do sinal em cada instante t_i introduz componentes com freqüências altas, que tornam o som bastante desagradável. Portanto, o sinal $s_\sqcap$ ainda precisa passar por um filtro de suavização R', que elimina essas transições.

O sinal $s_\sqcap$ pode ser analisado como o resultado da passagem do trem de impulsos $s_*(t)$ do esquema conceitual (figura 5.3) por um *filtro de extensão* X, que transforma cada impulso de Dirac $\delta(t - t_i)$ em um pulso retangular $\sqcap((t - t_i)/p)$, onde p é o passo de amostragem. Verifica-se que a função de transferência deste filtro é

$$X(f) = p\,\mathrm{sinc}(pf) = \frac{\mathrm{sen}(\pi f p)}{\pi f} \quad \text{para} \quad -\frac{f_*}{2} \leq f \leq +\frac{f_*}{2} \tag{5.9}$$

Portanto, o filtro de suavização R', que substitui o filtro de reconstrução R do esquema conceitual (figura 5.6), deve ter função de transferência $R'(f) = R(f)/X(f)$.

Na verdade, nenhum circuito fisicamente realizável consegue fazer uma transição instantânea entre um degrau e outro. Portanto, na prática o filtro X embutido no conversor D-A difere da fórmula (5.9), especialmente nas freqüências mais altas. Este detalhe deve ser levado em conta no projeto do filtro R'.

5.6 Análise de Fourier discreta

5.6.1 Série de Fourier

Verifica-se, na teoria de Fourier, que um sinal *periódico* s pode ser representado como uma *soma* infinita de senóides, todas com o mesmo período de s [10]. Ou seja

$$s(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty}[a_k cos(2\pi f_k t) + b_k sen(2\pi f_k t)] \tag{5.10}$$

onde $f_k = k/T$.

Comparando a fórmula geral (3.5) da análise de Fourier com a fórmula (5.10), observamos que a integral sobre a freqüência f foi substituída por uma somatória onde aparecem apenas termo a_0 (o *valor médio* do sinal) e as senóides cujas freqüências f_k são múltiplos inteiros da freqüência $f = 1/T$ do sinal, a *freqüência fundamental*.

A senóide de freqüência $f_k = kf$ é o *k-ésimo harmônico* ou *harmônico de ordem k* do sinal s. O primeiro harmônico, de freqüência f_1, é a *componente fundamental* do sinal. (A freqüência f_1 é denotada f_0 ou F_0 por muitos autores.)

Os coeficientes a_k, $k = 0, 1, 2, ...$ e b_k, $k = 1, 2, ...$ da fórmula (5.10) podem ser obtidos pelas fórmulas

$$a_k = \frac{2}{T}\int_0^T s(t)\cos(2\pi k f t)\,dt \qquad b_k = \frac{2}{T}\int_0^T s(t)\,\mathrm{sen}(2\pi k f t)\,dt \tag{5.11}$$

5.6.2 Série de Fourier complexa

Assim como no caso contínuo, as séries de Fourier podem ser simplificadas trabalhando-se com os números complexos. Especificamente, todo sinal periódico, s real ou complexo, pode ser representado como soma de senóides complexas:

$$s(t) = \sum_{k=-\infty}^{\infty} S_k e^{\mathrm{i}2\pi kft} \qquad \text{onde} \qquad S_k = \frac{1}{T}\int_{-T/2}^{+T/2} s(t) e^{-\mathrm{i}2\pi kft}\, dt \tag{5.12}$$

Temos portanto que a série de Fourier de uma função periódica *contínua* $s(t)$, como a ilustrada na figura 5.9(a), é uma seqüência *discreta* de números complexos S_k, como ilustrado nas figuras 5.9(b,c).

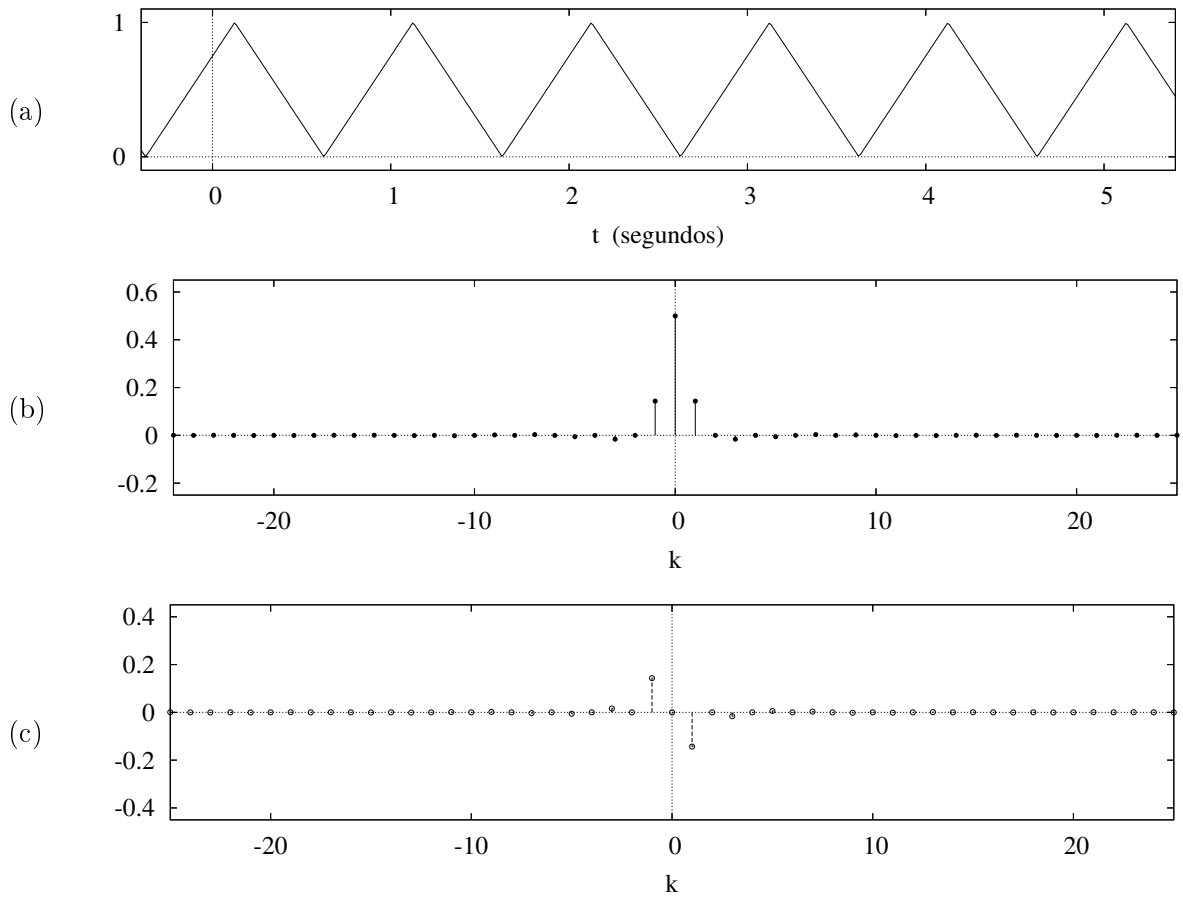


Figura 5.9: Série de Fourier. (a) Uma função periódica $s(t)$. (b,c) Partes real e imaginária de alguns coeficientes S_k de sua série de Fourier complexa.

5.7 Transformada discreta de Fourier

Se o sinal s é conhecido apenas em n instantes $t_0, ..., t_{n-1}$ igualmente espaçados ao longo do período, a série de Fourier (5.12) pode ser reduzida aos n primeiros termos

$$s_i = \sum_{k=0}^{n-1} S_k e^{\mathbf{i}2\pi ki/n} \qquad \text{onde} \qquad S_k = \frac{1}{n}\sum_{i=0}^{n-1} s_i e^{-\mathbf{i}2\pi ki/n} \tag{5.13}$$

A fórmula de S_k é chamada de *transformada discreta de Fourier*, e a fórmula que recontrói as amostras s_k é sua transformada *inversa* [47]. Veja a figura 5.10.

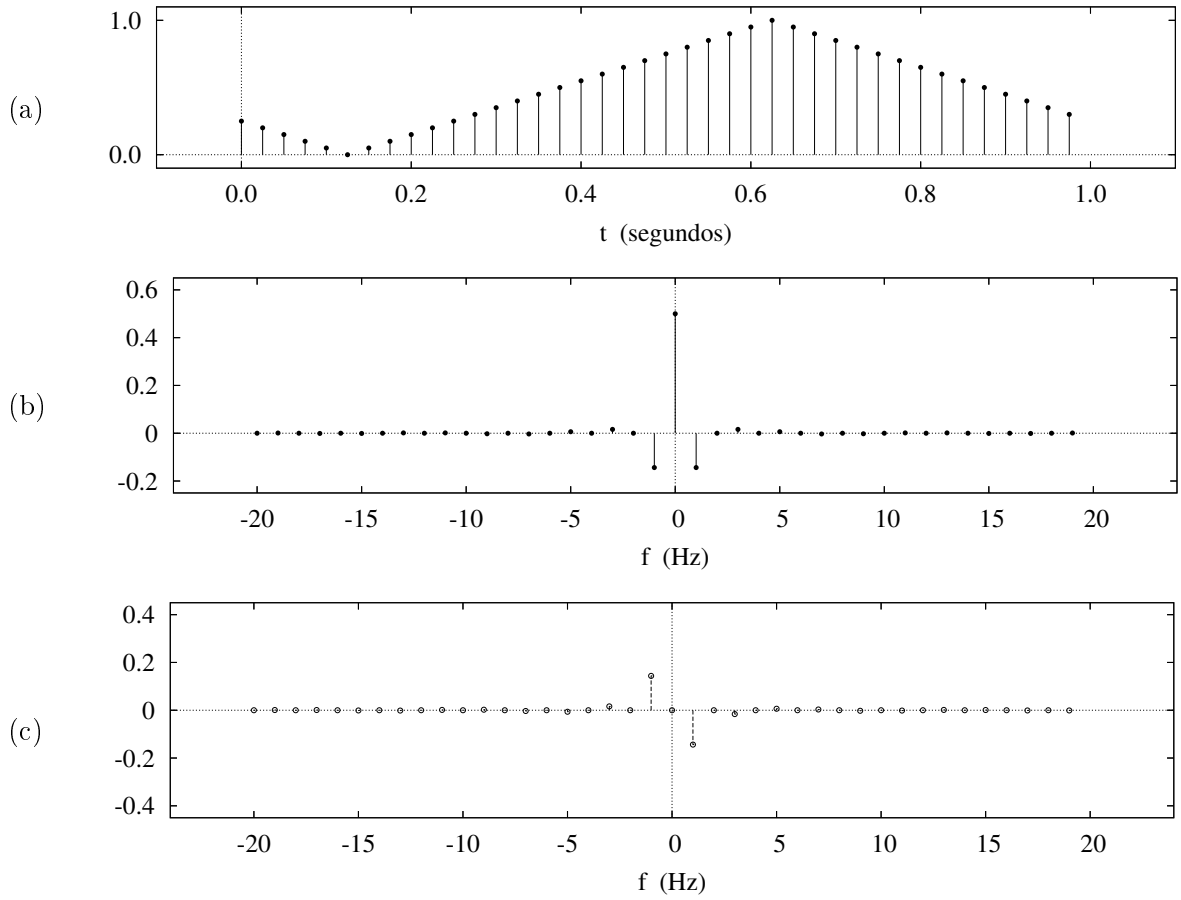


Figura 5.10: Série discreta de Fourier. (a) Uma função periódica, representada por $n = 40$ amostras s_i no seu período. (b,c) Partes real e imaginária dos coeficientes S_k de sua série discreta de Fourier.

Como o sinal s é periódico, podemos supor que as amostras s_i se repetem com período n; isto é, $s_{i+n} = s_i$ para todo inteiro i. Verifica-se que a mesma propriedade vale para os coeficientes S_k; isto é, $S_{k+n} = S_k$ para todo inteiro k.

A transformada discreta de Fourier pode ser aplicada mesmo quando o sinal não é periódico, ou quando seu período não coincide com o intervalo de recorte. Veja a figura 5.11.

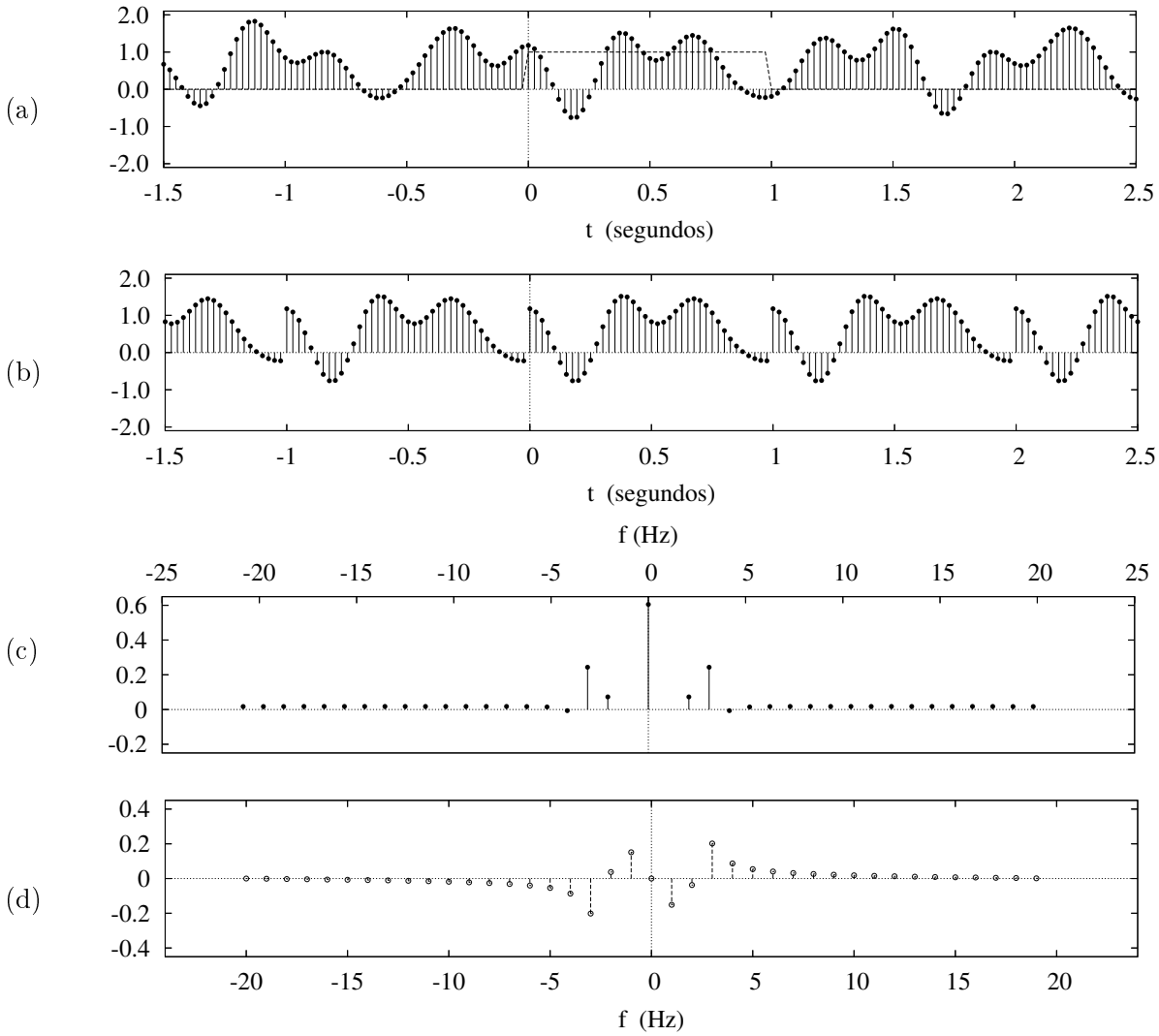


Figura 5.11: TDF de um sinal não periódico com janelamento retangular. (a) O sinal original. (b) O sinal recortado no intervalo $[0, 1]$ e repetido. (c,d) As partes reais e imaginárias de sua transformada discreta de Fourier.

Usar as fórmulas (5.13) nesses casos eqivale a repetir idefinidamente as n amostras dadas; ou seja, supor que o sinal tem período np onde p é o passo de amostragem.

Esta prática geralmente cria um salto entre as amostras s_{n-1} e s_0, que introduz componentes espúrias na transformada — como as componentes acima de 5 Hz na figura 5.11(c,d). Para reduzir este defeito, usa-se geralmente uma função de janelamento suave, como as vistas na seção 3.7 para recortar o sinal sem produzir saltos. Veja a figura 5.12.

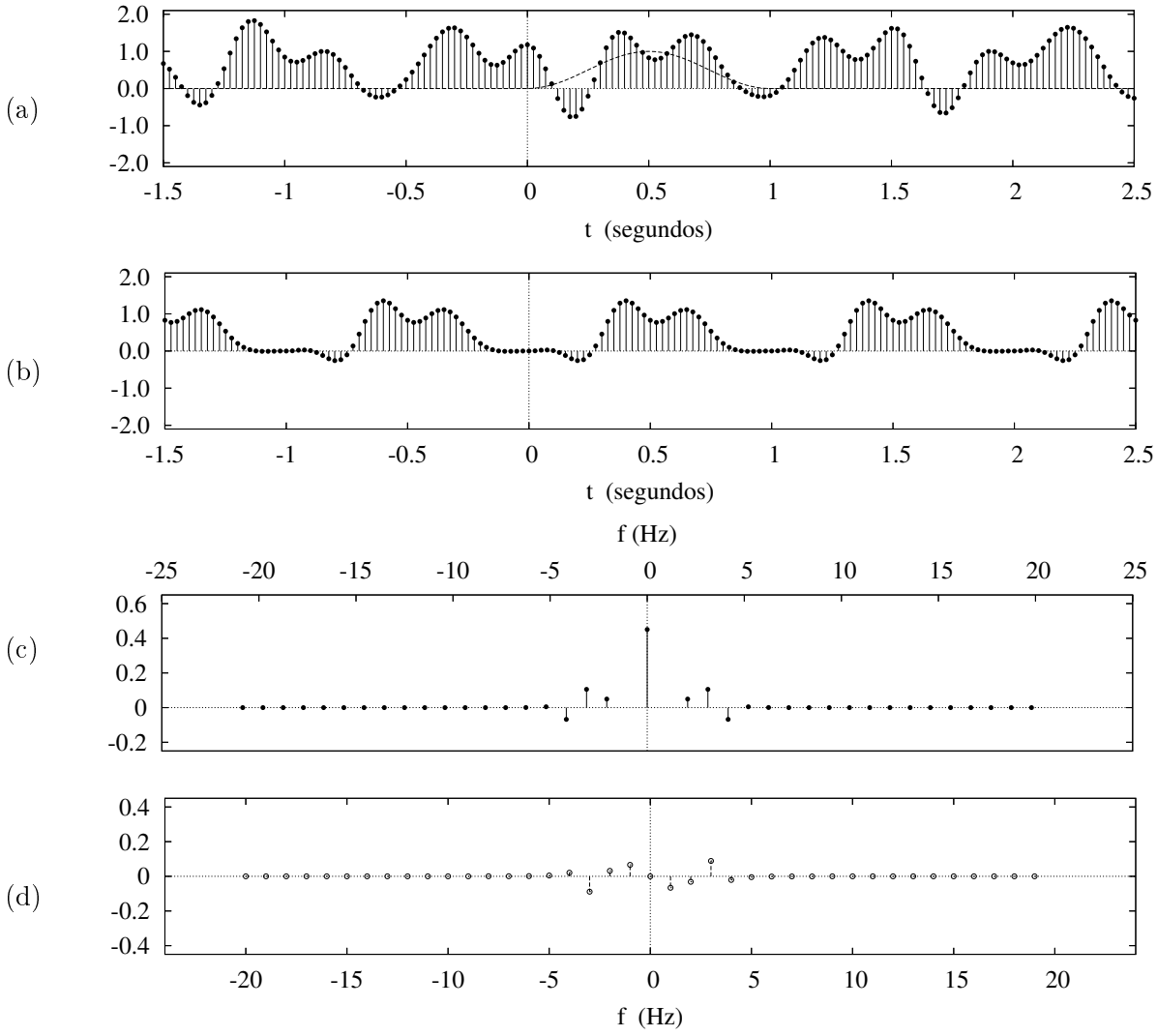


Figura 5.12: TDF de um sinal não periódico com janelamento de Hann. (a) O sinal original. (b) O sinal recortado no intervalo $[0, 1]$ com janelamento de Hann e repetido. (c,d) Sua transformada discreta de Fourier. Compare com a figura 5.11.

5.7.1 Transformada rápida de Fourier

O cálculo da transformada discreta de Fourier pela fórmula (5.13) requer um número de operações proporcional a n^2. Na prática, utiliza-se o algoritmo conhecido como *transformada rápida de Fourier* (*Fast Fourier Transform*, FFT) [10], que calcula as fórmulas (5.13) em tempo proporcional a $n \log_2 n$.

5.8 Transformada Z

Apesar da importância física da transformada de Fourier, a análise de sinais discretos geralmente fica muito mais simples quando se trabalha com a *transformada Z* ou *transformação direta* [47, 10].

A transformada Z de um sinal discreto x é uma função complexa S de variável complexa, definida pela fórmula

$$S(z) = \sum_{i=-\infty}^{\infty} s_i z^{-i} \tag{5.14}$$

Nesta fórmula, entende-se que z varia sobre todo o plano dos números complexos.

A série infinita nem sempre converge. Porém, se a seqüência tem duração finita, com amostras s_0, s_1, ..., s_{n-1}, a transformada Z passa a ser uma série finita

$$S(z) = \sum_{i=0}^{n-1} s_i z^{-i} \tag{5.15}$$

Note-se que a fórmula (5.15) é um polinômio em $1/z$, com coeficientes s_0, s_1, ..., s_{n-1}.

Pode-se verificar que, se tomarmos

$$z_k = e^{2\pi \mathbf{i}/n} = (\cos(2\pi k/n) + \mathbf{i}\,\mathrm{sen}(2\pi k/n)) \tag{5.16}$$

a equação (5.15) fica

$$S(z_k) = \sum_{i=0}^{n-1} s_i e^{-2\pi \mathbf{i} k i/n}$$

que é essencialmente (a menos de um fator de escala) a fórmula (5.13), a transformada discreta de Fourier. Ou seja, os coeficientes da TDF de um sinal s estão contidos na sua transformada Z $S(z)$; mais precisamente, nos valores de $S(z)$ em certos pontos z com $|z| = 1$ (isto é, no círculo unitário do plano complexo). Portanto, podemos dizer que a transformada Z é uma generalização da transformada discreta de Fourier.

5.8.1 Propriedades da transformada Z

A transformada Z possui muitas propriedades úteis, análogas às propriedades da transformada de Fourier (veja seção 3.4). Por exemplo, suponha que r é uma cópia do sinal s, atrasada de p amostras (ou seja, $r_i = s_{i-p}$ para todo i), e que $R(z)$ é sua transformada Z. Nesse caso, observa-se que

$$R(z) = z^{-p} S(z) \tag{5.17}$$

Em particular, define-se a *convolução discreta* $h = f \circledast g$ de duas seqüências $f_0, f_1, \ldots, f_{m-1}$ e $g_0, g_1, \ldots, g_{n-1}$ pela fórmula

$$h_i = \sum_{k=-\infty}^{+\infty} f_k g_{i-k} \tag{5.18}$$

onde se entende que elementos não-existentes valem 0. Nesse caso, pode-se verificar que $H(z) = F(z)G(z)$. Isto é, a transformada Z da convolução é o produto das transformadas Z das duas seqüências.

5.9 Filtros digitais

Um *filtro digital* é um dispositivo, processo ou algoritmo que recebe um sinal digitalizado como entrada, e devolve outro sinal digitalizado como saída. Muitos dos conceitos que vimos para filtros analógicos (capítulo 4) podem ser aplicados para filtros digitais. Em particular, pode-se definir o conceito de filtro digital *linear e invariante no tempo*, e verifica-se que o efeito de tal filtro é completamente descrito por uma função complexa $F(z)$, a transformada

Z da saída produzida quando a entrada é um *impulso discreto* — o sinal discreto δ que tem $\delta_0 = 1$ e $\delta_i = 0$ para todo $i \neq 0$.

5.9.1 Filtro de predição linear

O *filtro de predição linear* é um tipo de filtro digital muito importante em inúmeras aplicações, especialmente para processamento de sinais de voz. Nesse tipo de filtro, cada amostra s'_i do sinal digital de saída é calculada somando-se o sinal de entrada a uma *predição linear* — uma combinação linear de um número finito p de amostras de saída anteriores:

$$s'_i = s_i + \sum_{k=1}^{p} a_k s'_{i-k} \tag{5.19}$$

Os parâmetros a_k, $k = 1, \ldots, p$ são chamados de *coeficientes de predição linear*.

Esta abordagem foi desenvovida no final da década de 1960, inicialmente para compressão de sinais de voz em telecomunicações. Nessa aplicação, ela é chamada de *codificação por predição linear*, em inglês *linear predictive coding* [52, 23]. Por essa razão, este tipo de filtro é universalmente chamado de *filtro LPC*.

O efeito deste filtro pode ser determinado comparando-se a transformada Z do sinal de entrada, $S(z)$, com a do sinal de saída,

$$S'(z) = S(z) + \sum_{k=1}^{p} a_k S'(z) z^{-k} \tag{5.20}$$

Da equação (5.20), e da fórmula (5.17) para a transformada Z de um sinal deslocado, tiramos que

$$S'(z) \;=\; S(z) + \sum_{k=1}^{p} a_k S'(z) z^{-k} \;=\; S(z) + S'(z) A(z) \tag{5.21}$$

onde

$$A(z) = \sum_{k=1}^{p} a_k z^{-k} \tag{5.22}$$

Concluímos da equação (5.22) que

$$S'(z) \;=\; S(z)\frac{1}{1-A(z)} \tag{5.23}$$

Ou seja, a função de transferência Z do filtro é $1/(1-A(z))$.

Parte II

A Fala Humana

Capítulo 6

Som, Audição e Fala

6.1 Natureza do som

Como mencionado no capítulo 2, o som é uma deformação de um meio elástico que se propaga na forma de ondas. Na fala humana, o meio elástico é geralmente o ar, e a deformação consiste de variações de densidade e pressão.

6.1.1 Fontes sonoras

Uma fonte geradora de som geralmente consiste de um fornecedor primário de energia, de um elemento produtor do som (um corpo que vibra) e de elementos que modificam e direcionam o som (ressonadores).

Por exemplo, num violão, o elemento primário é a mão que pinça a corda, o elemento vibrante é a corda e a caixa do violão seria o ressonador. Numa caixa de som, o elemento primário é a bobina do alto-falante, o elemento vibrante é o cone flexível acionado pela bobina, e o restante da caixa funciona como ressonador.

6.1.2 Amplitude, potência e intensidade

A *amplitude* do som é definida como a variação máxima de pressão a partir da pressão normal do meio. A onda sonora carrega energia mecânica, portanto outra medida da "intensidade" do som é a potência por unidade de área perpendicular à direção de propagação da onda (W/m^2). Esta grandeza é proporcional ao quadrado da amplitude do som.

A intensidade *relativa* de dois sons é geralmente medida pelo número de *decibéis* (dB). Se o primeiro som tem potência P_1 e o segundo tem potência P_2, a intensidade do segundo é por definição $10 \log_{10} \frac{P_2}{P_1}$ decibéis em relação ao primeiro.

Em um som *puro* ou *senoidal*, a pressão varia com o tempo segundo uma senóide (vide seção 2.3.1). Portanto, um som puro pode ser caracterizado pela sua freqüência, sua amplitude e sua fase.

6.2 Processamento de som

Até a década de 1950, o processamento e a transmissão do som eram feitos inteiramente de maneira analógica, isto é, por circuitos elétricos e/ou dispositivos mecânicos, onde o som era representado por correntes elétricas, movimentos de peças, flexão de membranas, etc., que variavam de maneira *contínua* com a pressão do ar.

Muitos circuitos e dispositivos analógicos ainda são usados hoje em dia. Por exemplo, um *microfone* é um dispositivo analógico que converte variações de pressão em variações de tensão (ou corrente) elétrica; e um *alto-falante* é um dispositivo que faz a conversão contrária. Microfones e alto-falantes são casos particulares de *transdutores*. Outros exemplos incluem as cabeças de leitura e gravação em gravadores de fita magnética, e as cabeças de *pick-up* para discos de vinil. A maioria dos amplificadores e equalizadores encontrados em sistemas caseiros de áudio ainda fazem o processamento do sinal elétrico de maneira analógica.

6.3 Sistema auditivo humano

Percebemos sons no ar principalmente através do nosso *sistema auditivo*, que consiste de três partes: *ouvido externo*, *médio* e *interno* (figura 6.1). O ouvido externo inclui a orelha, o canal auditivo e o tímpano. O ouvido médio é uma cavidade na qual se localizam três pequenos ossos (*ossículos*) interligados: o *martelo*, a *bigorna* e o *estribo*, que fazem o acoplamento mecânico entre o tímpano e o ouvido interno. O ouvido interno é constituído por três *canais semi-circulares* (órgãos importantes para o sentido de equilíbrio, mas sem papel na audição) e pela cóclea (onde o som passa ao sistema nervoso) [20, p.142].

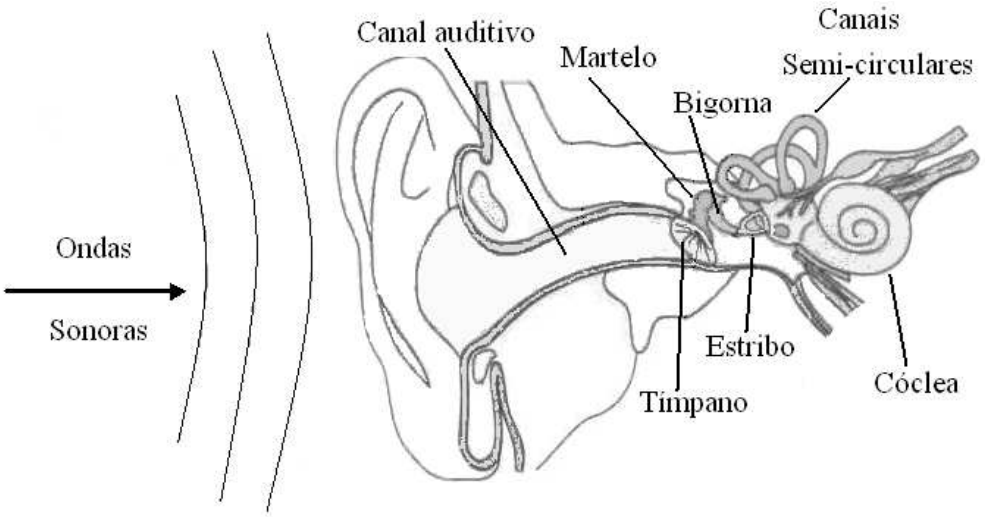


Figura 6.1: O sistema auditivo humano.

6.3.1 Percepção do som

A cóclea é um tubo ósseo enrolado em espiral, dividido longitudinalmente por uma membrana em dois canais principais paralelos cheios de um líquido.

As ondas sonoras percorrem o canal auditivo pressionando o tímpano, que é uma membrana semelhante à pele de um tambor, causando vibrações. Essas vibrações movimentam

os ossículos, que as transformam em movimentos do líquido na cóclea.

A membrana da cóclea tem duas camadas, separadas por milhares de *micro-pelos*. Os micro-pelos estão inseridos em células da membrana, as *células capilares*, e estas estão ligadas às fibras do *nervo auditivo*, o principal nervo entre o cérebro e o sistema auditivo.

Qualquer movimento do líquido coclear flexiona os micro-pelos e ativa as células capilares que então emitem sinais elétricos. Desta forma, as vibrações mecânicas do som são transformadas em sinais elétricos, que são transmitidos ao cérebro.

A cóclea é a parte do sistema auditivo responsável por distinguir sons de diferentes freqüências. As freqüências maiores são detectadas na parte mais próxima ao estribo, e as menores na outra ponta.

O sistema auditivo humano normalmente consegue captar sons de potência entre 10^{-6} Watt/cm^2 (nível definido como 0 dB em acústica), e os sons que produzem dor e podem causar lesões, acima de 120 dB, ou seja 10^{12} (um trilhão) de vezes mais intensos em potência do que o mínimo.

Na verdade, a intensidade mínima perceptível depende da freqüência. O limiar é 0 dB apenas para sons de aproximadamente 3000 Hz, mas é maior para sons com freqüência abaixo ou acima desse valor. Sons com freqüência abaixo de 20 Hz ou acima de 20.000 Hz, com qualquer amplitude, são inaudíveis [46].

Alguns animais, como cães e morcegos, conseguem ouvir sons de freqüências bem maiores. Sons acima de 20.000 Hz (*ultra-sons*) são usados em alguns processos industriais, como limpeza de peças e solda de plásticos, e em medicina como uma alternativa a raios-X.

6.4 Produção da voz humana

O ser humano instintivamente se comunica com seus semelhantes por meio de uma grande variedade de sons, a *fala humana*. Veja a figura 6.2.

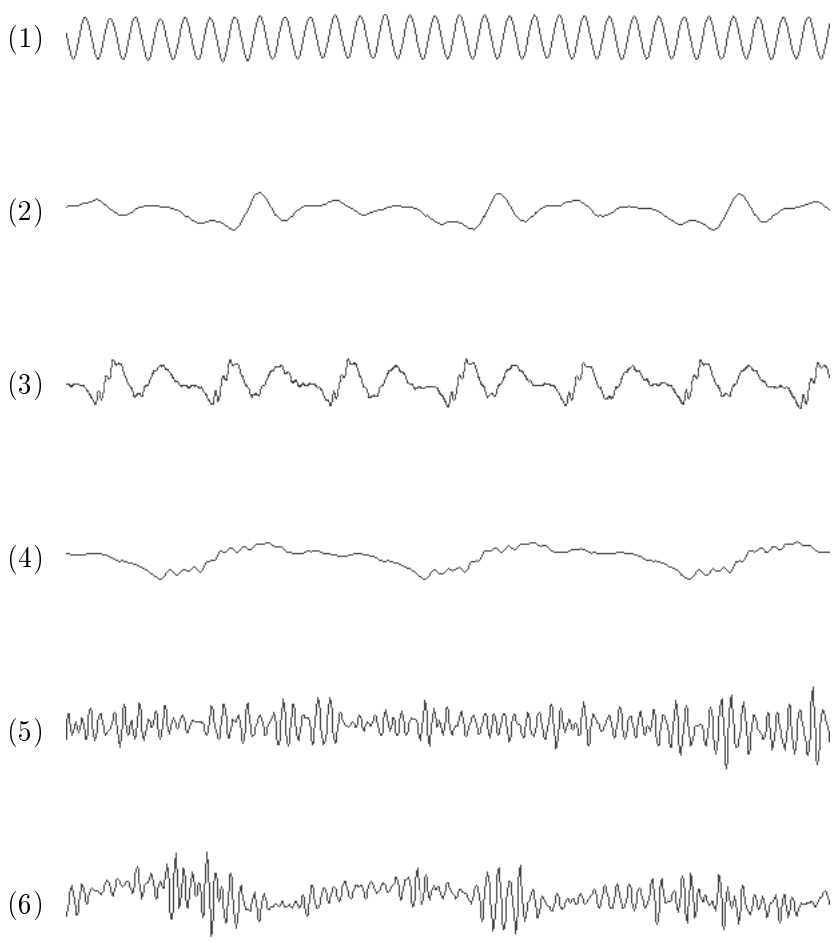


Figura 6.2: Variação da pressão em função do tempo, para vários sons produzidos pelo homem. (1) Assobio; (2) a vogal 'a'; (3) a vogal 'e'; (4) o som 'm'; (5) o som 'ch'; (6) o som 'j'. A duração total de cada gráfico é aproximadamente 18 ms.

6.4.1 O trato vocal

Os sons da fala são produzidos principalmente pelo fluxo do ar originado dos pulmões através de um conjunto de órgãos chamado de *trato vocal*, que vai desde a *glote* (laringe) até os lábios, incluindo a cavidade nasal [20, p.124]. Veja a figura 6.3.

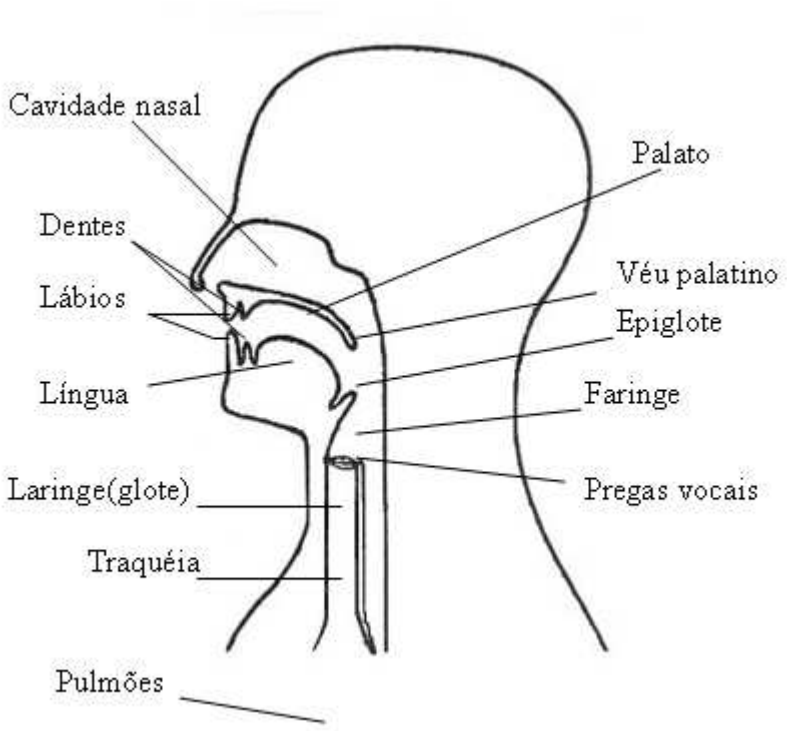


Figura 6.3: Visão seccionada da cabeça mostrando o trato vocal.

6.4.2 As pregas vocais

O órgão principal do trato vocal são as *pregas vocais* — duas membranas localizadas na laringe, que podem ser tensionadas ou relaxadas por músculos sob controle do cérebro.

Quando as pregas estão tensionadas, o ar precisa forçar a passagem pela abertura entre elas, fazendo-as vibrar. A freqüência da vibração geralmente varia de 80 Hz a 500 Hz para homens, e de 200 a 1100 Hz para mulheres, dependendo da tensão aplicada e da pessoa [46].

Esta vibração das pregas vocais ocorre em todos os sons da fala ditos *sonoros*, que, na língua portuguesa, incluem as vogais e certas consoantes como '*b*', '*v*', '*d*', '*g*', '*z*', '*j*', '*m*', '*n*', '*nh*', '*l*', '*lh*', '*r*', e '*rr*'.

Por outro lado, quando as pregas vocais estão relaxadas, o ar passa livremente pela glote, e elas não vibram. Esta é a situação quando respiramos sem falar, ou quando emitimos sons ditos *não sonoros*, que incluem as consoantes portuguesas '*p*', '*f*', '*t*', '*k*', '*s*', e '*ch*', ou a consoante inglesa '*h*'.

6.4.3 Articulação

Os outros órgãos do trato vocal também desempenham papéis importantes na produção da a fala humana. Modificando o formato do trato vocal — movimentando a língua, o véu palatino e a mandíbula, ou abrindo e fechando os lábios e a arcada dentária — o ser humano cosnegue modificar os sons produzidos pelas pregas vocais de várias maneiras, e gerar sons adicionais. Este processo, chamado de *articulação*, é o responsável pela grande variedade de sons usados em qualquer língua.

Por exemplo, o som '*s*' é produzido forçando-se o ar a passar por uma abertura estreita entre a língua e os alvéolos dentais (a parte saliente do céu da boca, logo atrás dos dentes). O som '*p*' é produzido fechando os lábios e o véu palatino, e depois abrindo os lábios e deixando o ar sair de repente. O som '*m*' é produzido vibrando as pregas vocais com os lábios fechados e o véu palatino abaixado, de modo que o ar passa pela cavidade nasal. E assim por diante.

6.4.4 Fonemas e Fones

A fala de cada pessoa, num determinado idioma, pode ser decomposta numa seqüência de *sons elementares* distintos ou *fones*, extraídos de um repertório de tamanho relativamente limitado (algumas dezenas de elementos). Estes fones são concatenados numa ordem específica para

formar as *palavras* do idioma.

Na verdade, o conjunto de fones de uma língua varia de região para região e mesmo de pessoa para pessoa. Por esse motivo, lingüistas geralmente trabalham com uma unidade mais abstrata de fala, o *fonema*, que representa um conjunto de fones que os falantes da língua reconhecem como equivalentes. Assim, diferentes pessoas podem realizar o mesmo fonema por fones diferentes, sem que isso prejudique a compreensão mútua. (Por exemplo, o fonema [t] da palavra *quente* é pronunciado de maneira bem diferente por cariocas e paulistas.) O conjunto de fonemas e suas realizações admissíveis, assim como o conjunto de palavras e seus significados, foram estabelecidos pela evolução histórica de cada idioma.

Durante a fala, o formato do trato vocal muda relativamente devagar, comparado com as vibrações das pregas vocais. A duração mínima de um fone é determinada pelo tempo necessário para que os nervos e músculos consigam modificar a articulação, que é da ordem de 50 milissegundos. Isto significa que a taxa de emissão de fones é limitada, por volta de 20 fones por segundo no máximo.

Algumas mudanças do trato vocal podem ocorrer em tempo menor, da ordem de 30 ms, se não forem controladas diretamente pelo cérebro. Este é o caso, por exemplo, das oscilações do diafragma que produzem o canto *vibrato*, ou das oscilações da língua que produzem o som '*rr*' da palavra *carro*. Veja a figura 6.4.

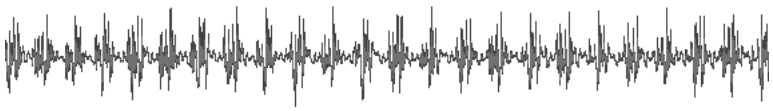


Figura 6.4: Forma de onda do som '*rr*' (R alveolar vibrado) do português, pronunciado de maneira contínua. Cada pulso (correspondente a uma oscilação da ponta da língua) dura aproximadamente 30 ms.

O tempo de articulação pode ser menor que o normal também no caso de mudanças devido a movimentos simultâneos de partes independentes do trato vocal, por exemplo nas transições entre fones como /ia/, /pr/ ou /st/, etc. Veja a figura 6.5.

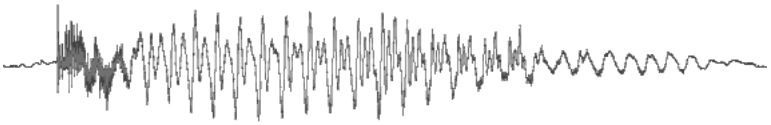


Figura 6.5: Forma de onda da palavra *tia*. A duração total do gráfico é 310 ms.

O trato vocal pode produzir muitos sons que não são usados por nenhum (ou quase nenhum) idioma, mas podem ser importantes na comunicação. Um exemplo bem conhecido é o *assobio*. Trata-se de uma vibração produzida pela passagem do ar entre os lábios e a língua, (por um mecanismo semelhante ao que ocorre em instrumentos de sopro como a flauta doce e o órgão).

6.5 Os fones da língua portuguesa

No caso da língua portuguesa, os fones correspondem vagamente às letras do alfabeto. Assim por exemplo, a vocalização da palavra *bebi* pode ser decomposta em quatro trechos correspondentes às letras "*b*","*e*", "*b*", e "*i*". Verifica-se que os trechos correspondentes aos dois "*b*"s, apesar de diferentes se comparados amostra por amostra, são bastantes semelhantes em certas características (formantes, duração, amplitude, etc.). A substituição de um desses trechos pelo outro, se feita com um pouco de cuidado para evitar saltos, não muda a palavra, e não altera significativamente a qualidade da pronúncia percebida pelos ouvintes. Por outro lado, se trocarmos outros trechos (por exemplo, "*b*" com "*i*", ou "*e*" com "*i*"), o resultado será outra palavra — ou, no mínimo, uma pronúncia muito incorreta da palavra.

Nem sempre os fones correspondem exatamente às letras da escrita. Na vocalização da palavra *abracadabra*, por exemplo, os sons correspondentes ao primeiro e ao quarto "*a*" são equivalentes, e o mesmo vale para os sons do segundo, do terceiro e do quarto "*a*"; porém, estes dois grupos de sons são significativamente diferentes em seu espectro e volume, e não podem ser trocados entre si. Considera-se portanto que estes dois grupos são realizações de dois fones distintos — respectivamente, "a pleno" e "a reduzido".

Fones são convencionalmente escritos entre barras, /a/, /ch/, etc. A tabela 6.1 mostra os fones da língua portuguesa, segundo o grupo LAFAPE do Instituto de Estudos da Linguagem da UNICAMP [3, 2].

Fone	Exemplo		Fone	Exemplo		Fone	Exemplo	
Vogais plenas								
/a/	lata	latA	/i/	pipa	pipA	/e/	medo	medO
/eh/	sela	sehlA	/o/	bolo	bolO	/oh/	bola	bohlA
/u/	mula	mulA						
Vogais reduzidas								
/A/	casa	kazA	/E/	trôpego	tRopEgO	/I/	rápido	rapIdO
/O/	pérola	pehROlA	/U/	glóbulo	gLohbUlO			
Vogais nasais								
/aN/	maçã	masaN	/AN/	ímã	imAN	/eN/	senta	seNtA
/EN/	hífen	ifEN	/iN/	tinta	tiNtA	/IN/	ínterim	iNterIN
/oN/	tonta	toNtA	/ON/	mórmon	mohRmON	/uN/	um	uN
/UN/	fórum	fohrUN						
Consoantes plenas								
/p/	pata	patA	/b/	bala	balA	/t/	tapa	tapA
/d/	data	datA	/k/	casa	kazA	/g/	gota	gotA
/f/	faca	fakA	/v/	vela	vehlA	/s/	sela	sehlA
/z/	zona	zonA	/sh/	cheque	shehkE	/zh/	joga	zhohgA
/m/	mola	mohlA	/n/	nada	nadA	/r/	rato	ratO
/l/	lata	latA	/nh/	vinho	vinhO	/lh/	molho	molhO
Consoantes reduzidas								
/S/	estar	eStaR	/R/	prato	pRatO	/L/	placa	pLakA
/B/	submete	suBmehtE	/D/	admira	aDmirA	/G/	magneto	maGnehtO
/K/	aspecto	aSpehKtO	/M/	amnésia	aMnehziA	/P/	adepto	adehPtO
Combinações especiais								
/Ks/	fixo	fiKsO						

Tabela 6.1: Os fones da língua portuguesa, na classificação do LAFAPE/IEL/UNICAMP. A notação usada para os fones (e para a transcrição fônica de cada palavra-exemplo) é a do sistema Ortofon.

6.6 Características perceptuais da voz humana

As principais características perceptuais da fala que o ser humano consegue variar, controlando o diafragma e os músculos do trato vocal, são:

Volume. O *volume* ou *intensidade* da voz é a principal qualidade que distingue sons altos ou fortes de sons baixos ou fracos — por exemplo, a fala normal da fala gritada. A grosso modo, ela corresponde à potência da onda sonora, e é determinada principalmente pela quantidade de ar expelida pelos pulmões por segundo.

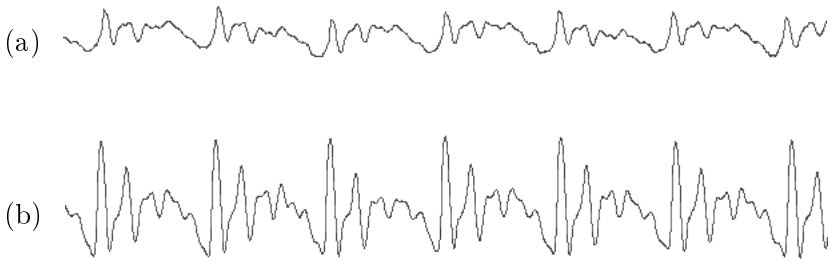


Figura 6.6: Forma de onda da vogal /a/ pronunciada com volumes diferentes. (a) Volume baixo; (b) volume alto.

Altura. A *altura* da voz (ou *pitch* em inglês) é a qualidade que distingue sons agudos de sons graves. A grosso modo ela corresponde à freqüência de vibração das pregas vocais, que é determinada principalmente pela sua tensão: quanto mais tensas, maior a freqüência, e mais agudo é o som.

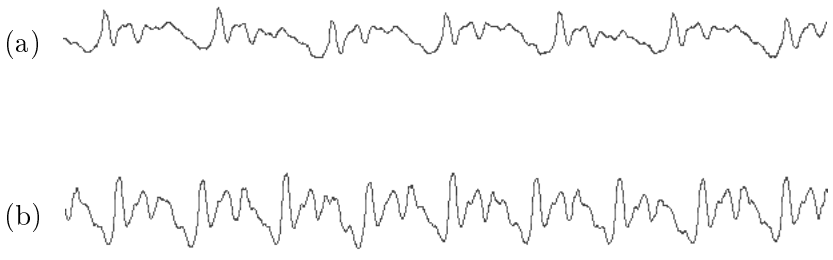


Figura 6.7: Forma de onda da vogal /a/ pronunciada em duas alturas diferentes. (a) Mais grave; (b) mais aguda.

A altura também depende do tamanho das pregas: quanto maiores, menor a freqüência, e mais grave o som. Este fator é a principal causa da diferença entre as alturas naturais da voz infantil, da voz feminina e da voz masculina.

Em certos idiomas, como o chinês e o suaíli, há fones que se diferenciam apenas pela altura da voz. Na língua portuguesa, a altura é usada principalmente para expressar interrogação. Por exemplo, a diferença entre a frase "você vai." (afirmativa) e "voce vai?" (interrogativa) é que a altura da voz aumenta no final da segunda frase.

Deve-se tomar cuidado com possível confusão entre os termos do idioma português referentes a volume e altura. Note-se que *som alto* e *som baixo* dizem respeito a volume (forte e fraco, respectivamente), enquanto que *tom alto* e *tom baixo* dizem respeito a altura (agudo e grave). Note-se também que a classificação de cantores em *baixo*, *barítono*, *tenor*, *alto*, *contralto* e *soprano* tem a ver com altura, e não com volume.

Timbre. O *timbre* é uma característica difícil de descrever, que permite diferenciar sons de mesma altura e volume. De modo geral, tem a ver com a forma do gráfico de cada onda sonora. É esta característica que permite identificar as pessoas pela voz, ou distinguir instrumentos como oboé, violão e violino. Na fala humana, o timbre é determinado principalmente pela forma do trato vocal. Veja a figura 6.8.

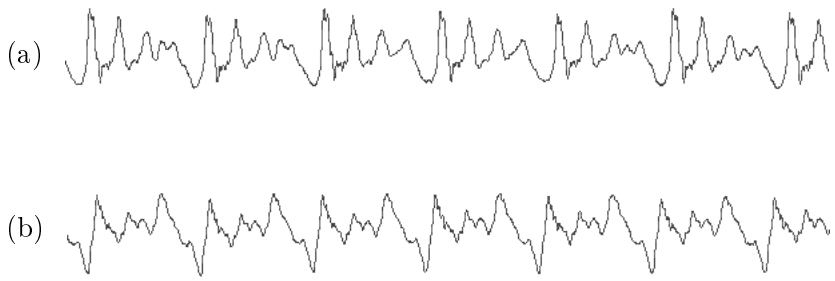


Figura 6.8: Forma de onda da vogal /a/ pronunciada na mesma altura por duas pessoas diferentes. (a) Mulher; (b) homem.

O timbre também depende muito da posição de seus órgãos móveis, e é qualidade que distingue as diferentes vogais de um idioma. Veja a figura 6.9.

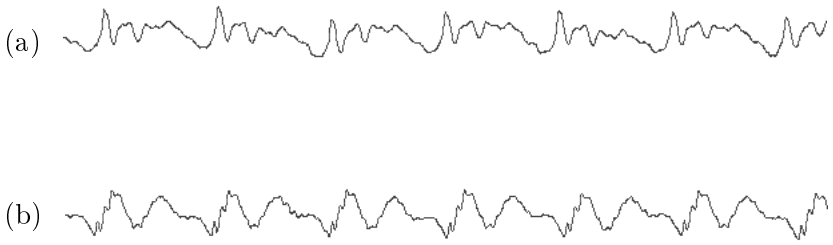


Figura 6.9: Forma de onda de vogais diferentes pronunciadas pela mesma pessoa na mesma altura. (a) Vogal /a/; (b) vogal /e/.

Duração. A *duração* de um som é a característica que distingue sons curtos ou breves de sons longos ou prolongados, ou seja, é o tempo decorrido entre o início e o fim do som. Na fala humana, a duração de certos fones é fixa, enquanto que a de outros sons pode ser controlada pelo falante, mantendo a articulação fixa pelo tempo desejado. No primeiro caso estão, por exemplo, certas consoantes como /p/, /t/, /b/, etc. No segundo caso estão todas as vogais e algumas consoantes como /m/, /l/, /s/, /ch/, etc.

No idioma português (e em muitos outros), a mesma seqüência de fones pode ser reconhecida como duas palavras distintas, dependendo da duração com que cada fone é pronunciado. Por exemplo, a diferença essencial entre *dúvida* (substantivo) *duvida* (verbo) é a duração relativa das vogais /u/ e /i/.

Em certas línguas, a duração dos fones é ainda mais importante. Por exemplo, em japonês a palavra *obasan* (primeiro /a/ curto) significa "tia", enquanto que *obāsan* (primeiro /a/ longo) significa "avó" [69]. Veja a figura 6.10.

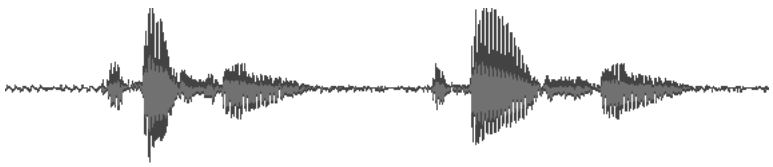


Figura 6.10: Forma de onda das palavras do idioma japonês *obasan* e *obāsan*. A duração total do gráfico é 2 segundos. A sílaba /ba/ dura aproximadamente 100 ms na primeira palavra, e 200 ms na segunda.

Capítulo 7

O espectro da fala humana

Neste capítulo descrevemos as principais características do espectro da fala humana. Uma descrição mais detalhada pode ser encontrada no livro de I. H. Witten [74].

De modo geral, os fones de qualquer idioma podem ser distinguidos pelo seu espectrograma. Na verdade, a cóclea pode ser entendida como um dispositivo que calcula o espectrograma do som que chega ao ouvido. A distinção é bastante nítida no caso de fones sustentáveis, e um pouco mais sutil no caso de plosivos.

Os pesquisadores B. Forbes e E. R. Pike do King's College de Londres afirmaram que as ferramentas matemáticas clássicas, como a teoria de Fourier, não são suficientes para descrever a formação de sons pelo trato vocal; e propõem usar ferramentas da mecânica quântica para esse fim [57]. Porém, este trabalho ainda é muito especulativo.

7.1 Sons primordiais

7.1.1 Voz laringeal

Como vimos na seção 6.4.2, a principal fonte de som da voz humana é a vibração das pregas vocais causada pela passagem do fluxo do ar dos pulmões. Pesquisas intensivas revelaram que as pregas vocais produzem uma seqüência de pulsos aproximadamente triangulares, como ilustrado na figura 7.1(a,b). O espectro deste som é bastante "rico," apresentando componentes significativas cujas freqüências são múltiplos da freqüência f dos pulsos, até alguns kHz. Veja a figura 7.1(b).

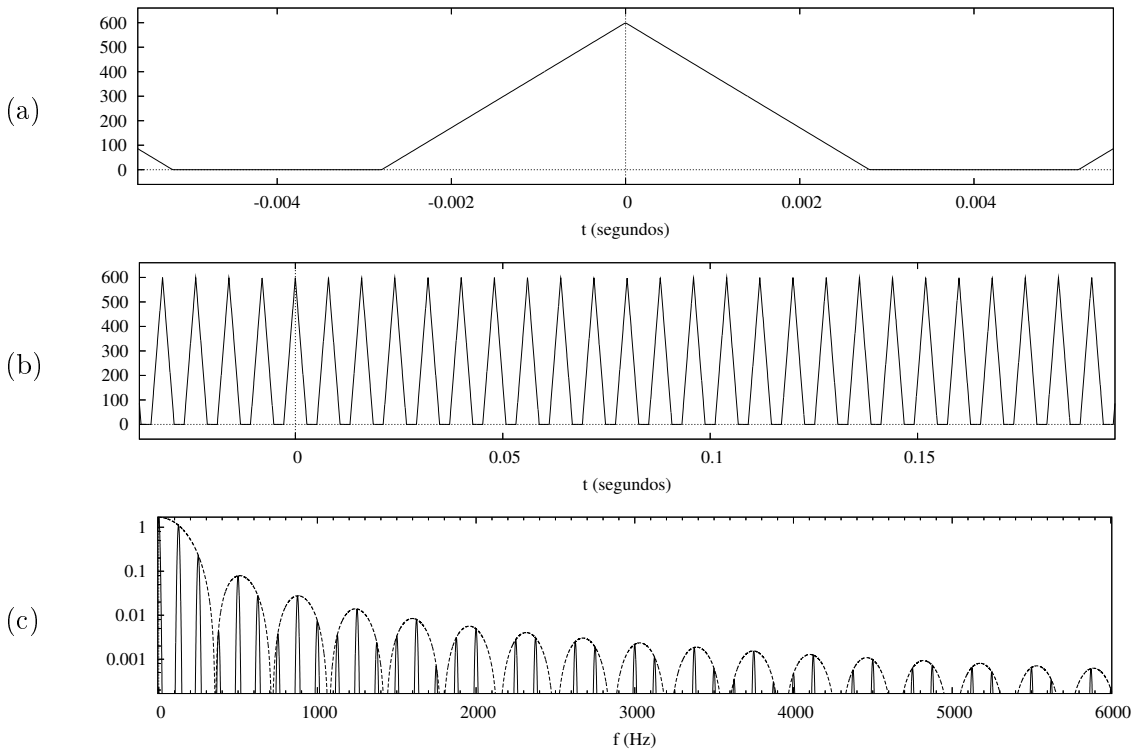


Figura 7.1: O som "primordial" produzido pelas pregas vocais. A freqüência fundamental é $f_1 = 125$ Hz. (a) gráfico idealizado do fluxo de ar (cm^3/sec) de um pulso; (b) gráfico idealizado de uma seqüência de pulsos; (c) o espectro de potência do trem de pulsos (linha cheia) e de um pulso glotal isolado (linha tracejada).

7.1.2 Sons fricativos

Outra fonte importante de som na voz humana é a turbulência gerada pelo fluxo do ar por aberturas estreitas formadas entre várias partes do trato vocal posteriores à glote. Este tipo de fonte ocorre em vários fones fricativos como /s/, /f/, etc. Ao contrário do som das pregas vocais, esses sons são aleatórios (não periódicos) e seu espectro mostra que a potência está espalhada sobre um grande intervalo de freqüências. Veja a figura 7.2.

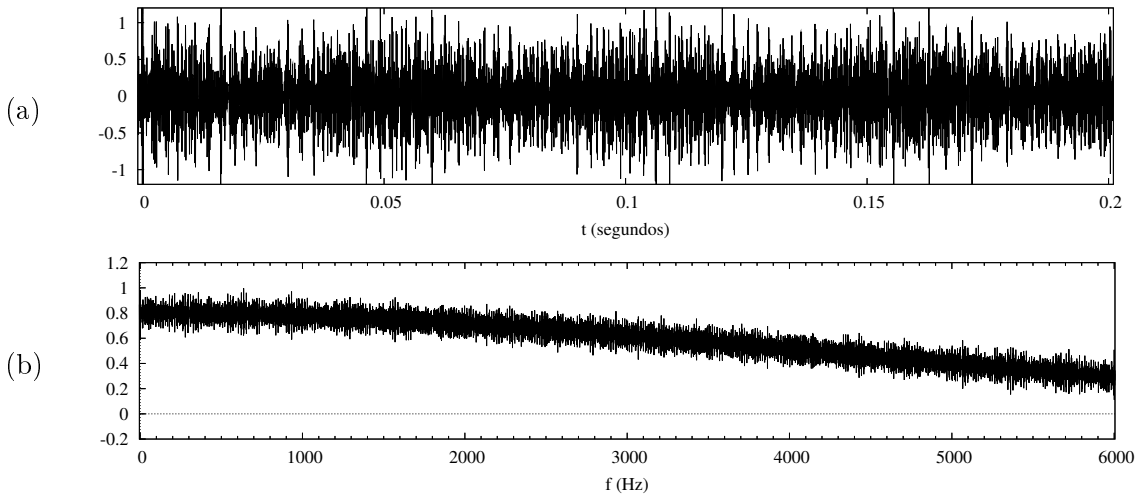


Figura 7.2: Som primordial de fones fricativos. (a) gráfico típico da pressão; (b) o espectro de potência correspondente.

7.1.3 Plosivos

Os sons *plosivos* como /t/, /p/, /k/, são caracterizados por uma interrupção temporária do fluxo do ar seguida de um pulso de pressão em que o ar represado é liberado de uma vez. As diferenças entre os vários fones plosivos são devidas principalmente à sua filtragem por diferentes partes do trato vocal, dependendo da articulação. Veja a figura 7.3.

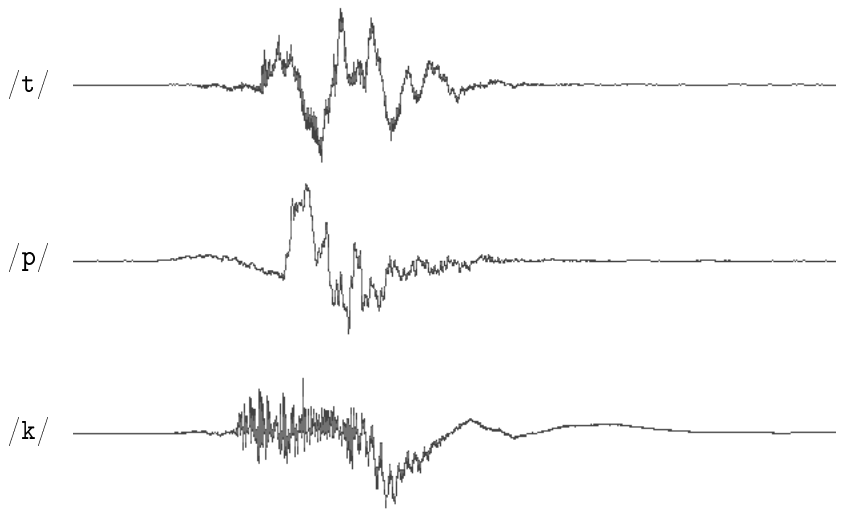


Figura 7.3: Gráficos da pressão para os sons plosivos /t/, /p/, /k/.

Certos fones, como /z/, /j/, /v/ do português, usam duas ou mais fontes de sons primordiais (pregas vocais e turbulência) simultaneamente.

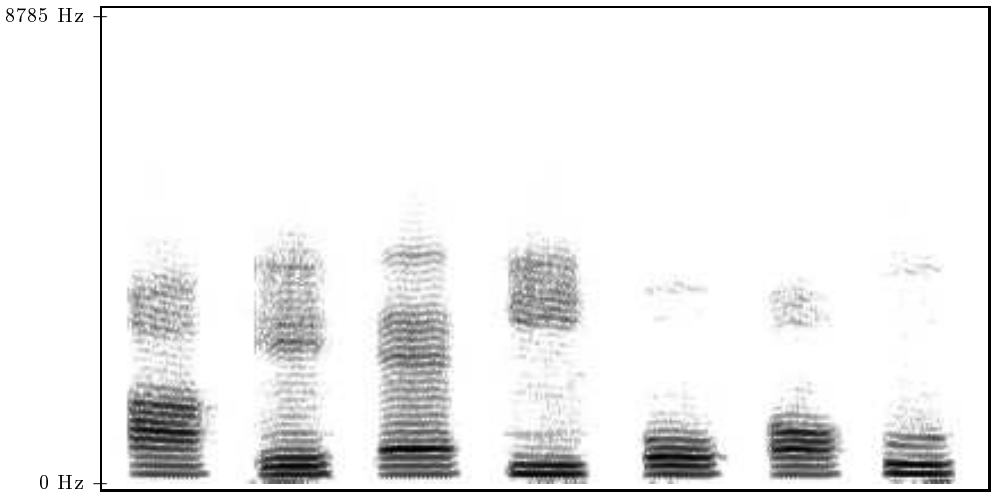
7.1.4 Vibrantes

O fone /rr/ do português, presente nas palavras *carro*, *guerra*, etc., é pronunciado em alguns dialetos como uma sucessão de dois ou mais sons semelhantes ao /r/ simples. Os movimentos da língua necessários para cada /r/ não são produzidos pelos músculos, mas pela própria corrente de ar, que faz a língua vibrar com uma freqüência de alguns hertz.

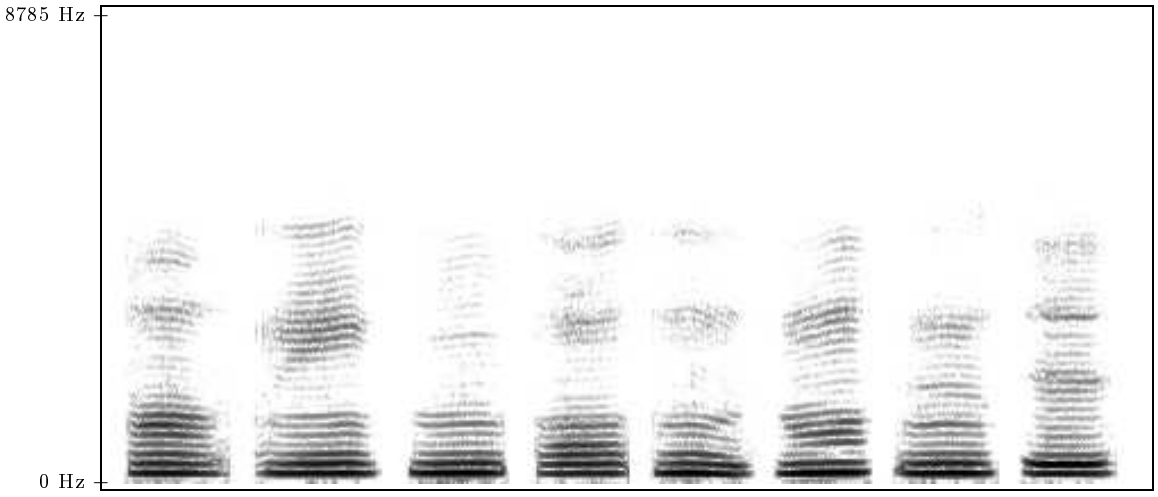
7.2 Formantes

Os sons primordiais são modificados pelo trato vocal que funciona como um filtro. O timbre da voz depende da função de transferência $H(f)$ deste filtro, que por sua vez depende da articulação corrente. Para os sons sustentáveis, a *assinatura* (conjunto de propriedades essenciais que diferem um fone de outro) consiste de concentrações de energias no espectro,

chamadas *formantes*. Os formantes correspondem a ressonâncias do trato vocal, isto é, a valores máximos de sua função de transferência $|H(f)|$. Veja as figuras 7.4 e 7.5.

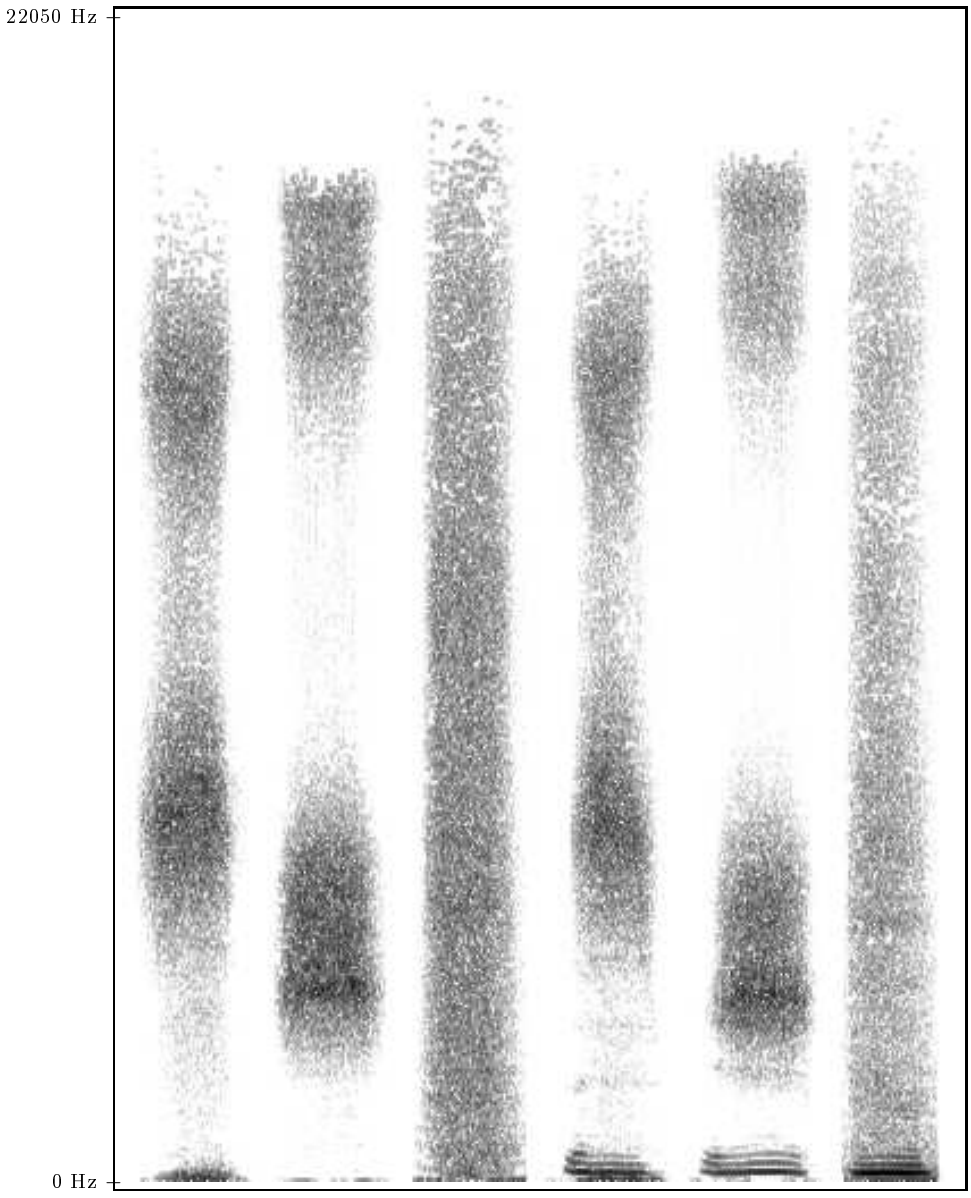


/á/, /ê/, /é/, /i/, /ô/, /ó/, /u/



/ã/, /ẽ/, /ĩ/, /õ/, /ũ/, /m/, /n/, /l/

Figura 7.4: Espectrogramas das vogais, sons nasais e sons laterais do português. Os riscos finos mais escuros são os harmônicos da freqüência da vibração das pregas vocais. Os formantes são as faixas horizontais mais escuras. Os sons /ẽ/, /ĩ/, /ũ/ ocorrem em palavras como "*vento*", "*sinto*", "*junto*".



/s/, /ch/, /f/, /z/, /j/, /v/

Figura 7.5: Espectrogramas dos sons fricativos do português. Note a ausência de sinal glotal nos sons surdos /s/, /ch/ e /f/.

Os formantes podem ser observados tanto na voz normal quanto na *voz sussurrada* ou *faringeal*. Nesta última, o som primordial das pregas vocais é substituído por um fricativo produzido forçando o ar pela epiglote (faringe) quase fechada. Praticamente todo som da fala normal tem um som correspondente na fala sussurrada, com a mesma articulação.

Os primeiros sintetizadores elétricos — como por exemplo o Stewart (1922), descrito na seção 9.3.2 — utilizavam apenas os dois formantes principais (de maior energia) das vogais. Em 1932, Obata e Teshima descobriram o terceiro formante, que é importante para a qualidade e clareza da fala. Os três primeiros formantes são geralmente considerados suficientes para fala sintética inteligível [34].

Parte III

Processamento de Fala

Capítulo 8

Conversão texto-fala

A síntese de fala a partir de um texto, ou *conversão texto-fala* (*text-to-speech synthesis*, TTS) aceita como entrada um arquivo texto e produz sua vocalização como um sinal de áudio.

8.1 Aplicações

Há inúmeras aplicações para a tecnologia de conversão texto-fala [36, 74]. Dentre as mais importantes, podemos citar as seguintes:

Mensagens por telefone: Uma aplicação importante e bem estabelecida de sistemas de conversão texto-fala é a leitura de mensagens de correio eletrônico por telefone. Para essa aplicação, não é necessário que a voz sintetizada soe natural, apenas que o texto seja reproduzido com clareza e fidelidade.

Leitura durante trabalho: Esta tecnologia também é usada para permitir a leitura de documentos e dados durante a execução de tarefas que exigem atenção visual, como manutenção de equipamentos, condução de veículos, cirurgias, vigilância e segurança, etc.

Deficientes visuais: Outra aplicação importante desta tecnologia são as máquinas de leitura para cegos. Nesses equipamentos, *scanners* e algoritmos de reconhecimento de caracteres são usados para transformar material impresso em textos digitais, que são convertidos para voz. Esta tecnologia também permite que deficientes visuais tenham acesso à Internet. Um inconveniente ainda a ser superado é que o leitor cego não tem idéia do tamanho do texto de uma página WWW. Para resolver esse problema, seria conveniente que cada página contivesse informações sobre seu tamanho, ou sobre o tempo previsto de leitura.

Educação: Conversores texto-fala podem ser úteis para vários fins educacionais, como o ensino de alfabetização e de línguas, difusão de literatura, apoio a estudantes disléxicos, etc.

8.2 Estrutura

Um sistema de conversão texto-fala é constituído de duas fases principais. A primeira fase, descrita neste capítulo, é a *transcrição fônica*, onde a seqüência de letras e símbolos é analisada segundo as regras da ortografia de linguagem e transformada numa seqüência de símbolos que denotam sons elementares da linguagem. A segunda fase, descrita no capítulo 9, consiste na geração do sinal acústico a partir dessa seqüência de símbolos. Essas duas fases são às vezes denominadas *síntese de alto nível* e *síntese de baixo nível*. Veja a figura 8.1.

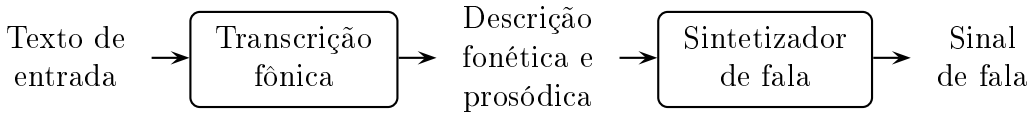


Figura 8.1: Esquema simplificado de um sistema de conversão texto-fala.

A síntese de baixo nível será tratada no capítulo 9; no restante deste capítulo, trataremos apenas da transcrição fônica. Esta etapa geralmente consiste de vários módulos, como *pré-processador* (discutido na seção 8.2.1), *conversor ortográfico-fonético* (seção 8.2.2), e *processador prosódico* (seção 8.2.3) [68].

8.2.1 Pré-processador

O primeiro passo para a transformação de um texto numa seqüência de fones é a passagem por um *pré-processador* ou *normalizador*, que re-escreve números, abreviações, símbolos, etc. em sua forma por extenso. Assim, por exemplo, o pré-processador substitui *32* por *trinta e dois*, *Sr.* por *senhor*, *R$ 5,20* por *cinco reais e vinte centavos*, e assim por diante. A tarefa do pré-processador é bastante difícil, e pode exigir análise gramatical do texto. Por exemplo: A frase *32 ovos* deve ser substituída por *trinta e dois ovos*; já *32 caixas* deve ser substituída por *trinta e duas caixas*.

8.2.2 Conversor ortográfico-fonético

O *conversor ortográfico-fonético* é o módulo que recebe o texto normalizado (contendo apenas palavras por extenso) e produz a seqüência de fones correspondente. Os conversores ortográfico-fonéticos podem variar bastante quanto à abrangência de detalhes ortográficos e contextuais, e quanto à riqueza de detalhes da descrição fônica produzida.

Como os exemplos acima mostram, mesmo nas línguas que usam escrita alfabética, há uma distância significativa entre a escrita "correta" ou "oficial" (*ortografia*) das palavras, e a seqüência de fones correspondente (*fônica*). A transformação da primeira na segunda envolve, em primeiro lugar, as regras gerais de pronúncia. Na ortografia portuguesa, por exemplo, valem as regras "*ç*" → /s/, "*qui*" → /k//i/, "*qua*" → /k//U//a/etc.

Porém, há muitas palavras cuja pronúncia não pode ser deduzida por regras; por exemplo, temos *exato* → /e//z//a//t//O/, mas *exu* –> /e//sh//u/. Para tratar esses casos, o conversor precisa ter um dicionário de exceções.

O dicionário não é suficiente para a conversão de palavras homógrafas e não homófonas — como *piloto*, que pode ser verbo (pronunciado /p//I//l//oh//t//O/) ou substantivo (pronunciado /p//I//l//o//t//O/). Para distinguir esses dois casos, é necessário fazer uma

análise gramatical do texto. Outros exemplos desse tipo são *molho*, *tomo*, *pelo*, etc.

Para certas palavras, pode ser necessária uma análise do sentido (*semântica*) do texto. Por exemplo, o substantivo *sede* pode ser pronunciado /s//e//d//E/ ou /s//eh//d//E/), dependendo do significado. Infelizmente, a escolha muitas vezes depende de raciocínios lógicos, informações do contexto e conhecimentos gerais que ainda estão muito além do alcance de sistemas de inteligência artificial. Considere, por exemplo, a frase "*A empresa de refrigerantes preocupa-se com sua sede*".

8.2.3 Processador prosódico

Para uma leitura "natural" e agradável de um texto, é necessário levar em conta a *prosódia* — variações de atributos como velocidade, volume e altura, que exprimem informações gerais sobre o conteúdo do texto e sobre o estado emocional do autor — tais como ênfase, dúvida, indignação, questionamento, etc. A prosódia também é importante para auxiliar o ouvinte a separar as sentenças e perceber a estrutura sintática do texto.

Alguns atributos prosódicos podem estar indicados explicitamente no texto. Por exemplo, parênteses são geralmente usados para marcar texto que deve ser pronunciado com volume reduzido, enquanto que letras em itálico ou sublinhado geralmente indicam maior volume e/ou altura de voz. O sinal "*?*" indica que a sentença anterior deve ser pronunciada com altura de voz crescente, para expressar interrogação. Outros sinais de pontuação (vírgula, ponto, ponto-e-vírgula, dois-pontos) indicam pausas maiores ou menores.

Para reproduzir esses efeitos, o texto normalizado deve passar também por um *processador prosódico*, que extrai informações sobre a duração ou entonação dos fones, a partir dos sinais de pontuação e do contexto sintático de cada palavra. Estas informações geralmente são *supra-segmentais*, isto é, aplicam-se a trechos de palavras ou de sentenças como um todo, e não a fones individuais [42]. Entretanto, para sua realização na fala sintetizada, estas informações precisam ser distribuídas entre os fones do trecho afetado.

8.3 O conversor *Natural Voices*

Um exemplo de sistema comercial moderno de conversão texto-fala é o software *Natural Voices* da Lucent Technologies (2001). O usuário pode acelerar ou desacelerar o ritmo da saída, e escolher o timbre (homem, mulher, criança). Há versões para vários idiomas, incluindo inglês, espanhol, italiano, alemão, russo, romeno, chinês e japonês.

O sistema *Natural Voices* descende da linha de pesquisa em síntese da fala dos Laboratórios Bell da AT&T, que começou com os sistemas *Vocoder* e *Voder* no final da década de 1930. O primeiro TTS completo foi lançado no mercado em 1973. Este era baseado no modelo articulatório desenvolvido por C. Coker [32]. A versão corrente utiliza um modelo concatenativo de polifones. Segundo S. Lemmetty [34], o sistema tem uma estrutura modular, consistindo de uma *pipeline* com 13 estágios, o que permite que cada módulo seja melhorado independentemente. Veja a figura figura 8.2.

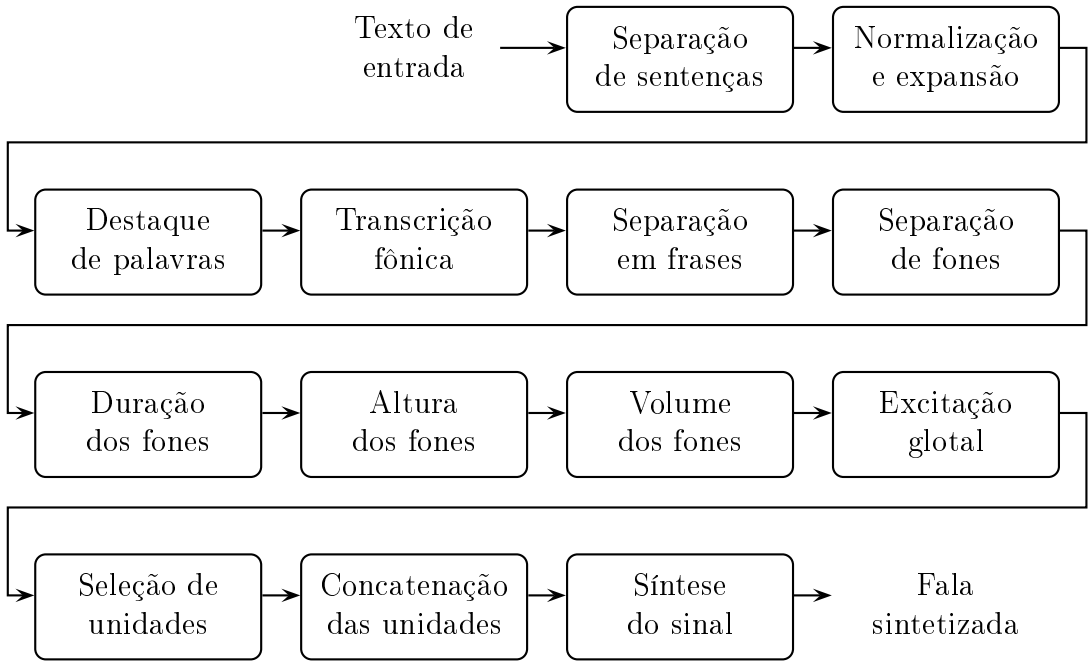


Figura 8.2: O sistema *Natural Voices* da Lucent Technologies.

Os módulos executam as seguintes funções: (1) separação do texto em sentenças; (2) a normalização do texto (expansão dos números, abreviações, etc.) e algumas análises gramaticais; (3) identificação das palavras que devem ser destacadas na sentença; (4) transcrição fônica e resolução de palavras homógrafas; (5) divisão de sentenças em unidades de entonação; (6) separação de fones; (7) determinação da duração de cada fone; (8) variação da altura da voz; (9) escolha do volume de cada fone; (10) determinação fração de abertura das pregas vocais e outros parâmetros da fonte glotal; (11) seleção das unidades a serem concatenadas; (12) concatenação das unidades; e (13) síntese do sinal de voz.

8.4 O conversor *Aiuruetê*

O sistema *Aiuruetê* é um conversor texto-fala para a língua portuguesa do Brasil, desenvolvido em 1999 na UNICAMP. Ele combina o conversor ortográfico-fonético *Ortofon* do Instituto de Estudos da Linguagem (IEL), com um sintetizador de fala da Faculdade de Engenharia Elétrica e de Computação (FEEC) [56, 5]. Veja a figura 8.3.

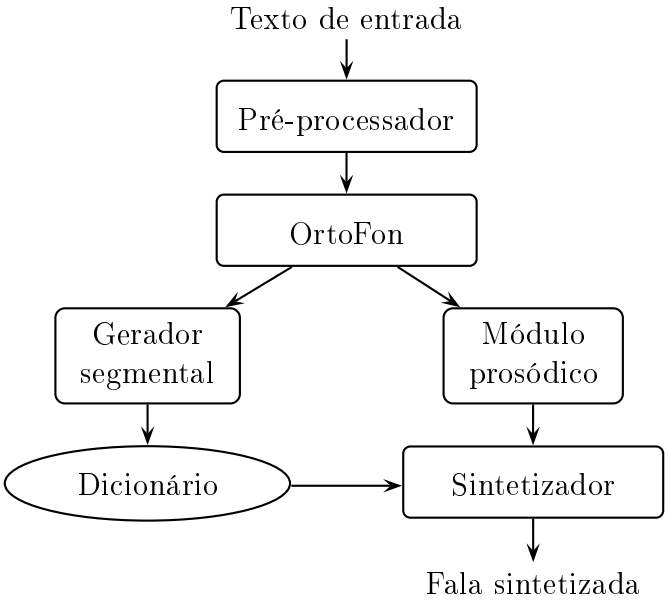


Figura 8.3: Esquema do conversor texto-fala *Aiuruetê*.

8.4.1 O conversor ortográfico-fonético *Ortofon*

O conversor ortográfico-fonético usado no sistema *Aiuruetê* é o *Ortofon*, desenvolvido pelo Instituto de Estudos da linguagem (IEL) da UNICAMP [3, 56]. Ele utiliza uma notação conveniente para os fones da língua portuguesa, já apresentada na tabela 6.1, que dispensa as barras entre os fones. Veja a figura 8.4:

(a)
```
O Prof. Pardal da UNICAMP não comprou 36 ovos
por R$ 5,20.
```

(b)
```
o professor pardal da unicamp não comprou
trinta e seis ovos por cinco reais e vinte
centavos.
```

(c)
```
o pRofesoR paRdaU da unikaNpI naNU koNpRoU
tRiNtA e sEIS ohvOS pOR siNkO reaIS e viNtE
seNtavOS.
```

Figura 8.4: Exemplo da transcrição fonética do sistema *Aiuruetê*. (a) Um texto de entrada. (b) Saída típica do pré-processador. (c) Saída típica do conversor ortográfico-fonético *Ortofon*.

8.5 Histórico

O primeiro sistema completo de conversão texto-fala para a língua inglesa foi desenvolvido em 1968 por Noriko Umeda e outros do Laboratório Eletro-Técnico do Japão [32]. A fala era inteligível, porém muito monótona e bem longe da qualidade dos sistemas atuais.

Outro marco importante nesta área foi a criação em 1976 de uma máquina de leitura para deficientes visuais por R. Kurzweil [33]. A máquina aceitava qualquer texto impresso ou escaneado, e produzia na saída a versão sonora do mesmo. Por causa do seu alto preço (30.000 dólares), essas máquinas não foram muito difundidas; elas eram encontradas apenas nas bibliotecas das grandes cidades.

Em 1979, J. Allen, M. S. Hunnicutt e D. Klatt do MIT desenvolveram um conversor texto-fala bem conhecido chamado *MITalk* [4]. Em 1981, Klatt introduziu um sucessor, o sistema *Klattalk* [32].

Os primeiros produtos comerciais populares de conversão texto-fala surgiram nos Estados Unidos, na década de 1980. Uma versão comercial do MITalk foi comercializada pela Telesensory Systems (TSI). As tecnologias *MITalk* e *Klattalk* formam a base de muitos sistemas de síntese de fala atuais, como *Prose 2000* (1982) e *DECtalk* (1983).

Capítulo 9

Síntese de Fala

A produção artificial da voz humana é um sonho antigo, e os primeiros sucessos significativos nessa direção datam do século 18 [34]. A produção por meios elétricos foi conseguida na década de 1920, e os primeiros sistemas digitais foram construídos na década de 1960.

A tecnologia tornou-se popular apenas na década de 1980, quando apareceram os primeiros circuitos integrados para realizar essa função, como o Votrax SC-01 (1980) e o Texas Instruments TMS–5110 (1980). Este último possibilitou o brinquedo educativo *Speak-n-Spell* da Texas Instruments.

Entretanto, resultados satisfatórios nessa área somente foram obtidos a partir da década de 1990, quando a velocidade e capacidade dos computadores possibilitaram o processamento de som em tempo real. As principais técnicas atualmente usadas na produção da fala artificial são: *síntese concatenativa* (descrita na seção 9.2), *síntese por filtragem* (seção 9.3), e *síntese articulatória* (seção 9.4).

Há vários artigos e documentos WWW com informações sobre a história da tecnologia de síntese de fala [50, 63, 73, 61, 56, 34].

9.1 Aplicações

Sintetizadores de fala são comumente usados como componentes de "baixo nível" de sistemas de conversão texto-fala, cujas aplicações foram mencionadas no capítulo 8. Entretanto, eles têm algumas outras aplicações próprias, em que não existe um "texto de entrada" no sentido comum do termo. Algumas dessas aplicações são mencionadas a seguir [74].

Telecomunicações: Uma das principais aplicações de sintetizadores de fala (e também uma das mais antigas) é a compressão de sinais de voz para maximizar a capacidade efetiva de linhas telefônicas e outros canais de comunicação. Em tais contextos, o sinal de fala de um usuário é resumido numa seqüência de parâmetros, que controlam um sintetizador de fala na outra ponta da linha.

Deficientes vocais e auditivos: Sintetizadores de fala podem ser de grande utilidade para as pessoas que possuem problemas na fala ou impossibilitadas de usar a voz, como os deficientes auditivos, por exemplo, na interação dinâmica com pessoas comuns que não conhecem a linguagem de sinais, ou para conversar utilizando o telefone.

Um exemplo importante é *cabeça falante* (*talking head*), que consiste de animação facial sincronizada com um sistema de síntese de fala, onde a animação facial concentra-se nos movimentos dos lábios e suas vizinhanças [11, 60]. Esta técnica se revelou muito útil para ajudar pessoas com problemas auditivos ou vocais a aprender a falar, melhorar a impostação da voz ou corrigir a pronúncia.

Serviços por telefone: Sintetizadores de fala têm sido utilizados há décadas em vários sistemas de consulta e comércio via linha telefônica.

Aplicações automotivas: Uma área de aplicação que tem crescido muito nos últimos tempos é o uso de fala sintética em carros e outros veículos, para transmitir informações

variadas ao condutor sem desviar sua atenção da estrada. Essas informações incluem avisos de segurança, estado do veículo, aconselhamento de rotas, boletins de trânsito, etc.

9.2 Síntese Concatenativa

A síntese concatenativa utiliza uma base de *unidades de fala*— segmentos pré-gravados de fala natural. O sinal de fala artificial é gerado concatenando unidades escolhidas dessa base. Veja o esquema dessa técnica na figura 9.1.

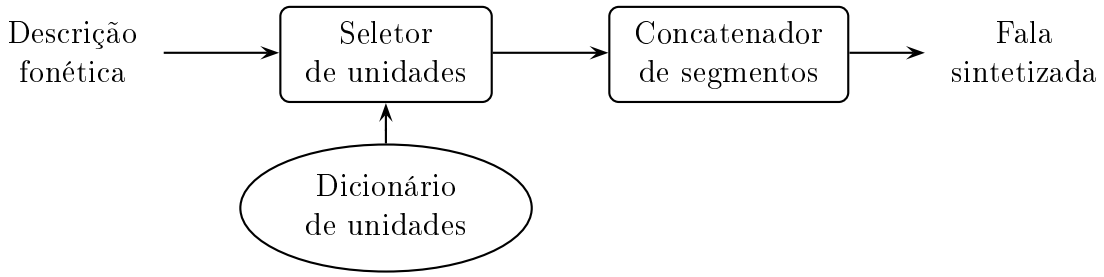


Figura 9.1: Esquema do método de síntese concatenativa.

A síntese concatenativa é uma técnica bastante simples e fácil de implementar. A parte mais trabalhosa e demorada é a construção do inventário de unidades. Em geral, escolhe-se um locutor com boa dicção, com timbre e altura de voz agradáveis. Esse locutor passa horas gravando textos e frases. Essas gravações são em seguida recortadas em segmentos, os quais são indexados e armazenados.

Os segmentos podem ter comprimentos variados. Na versão mais trivial, o dicionário contém sentenças completas, pré-gravadas pelo locutor humano. Esta solução é muito fácil de implementar e produz fala de ótima qualidade, mas está limitada a um conjunto fixo de sentenças. Para um texto de entrada de tamanho arbitrário e com vocabulário irrestrito, é necessário trabalhar com unidades menores, tais como sílabas, fones, pares de fones, etc.

Teoricamente seria possível usar os fones da língua como segmentos; porém, na prática é difícil obter uma transição suave entre dois fones, especialmente em ditongos. Por esta razão,

uma escolha comum é o *difone*, uma unidade que consiste do final de um fone e do início de outro. Por exemplo, a palavra *casa* seria formada concatenando os difones /#k/, /ka/, /az/, /zA/ e /A#/, onde /#/ indica uma pausa entre palavras. Neste tipo de sistema, o dicionário é de tamanho médio (por volta de 1.000 a 2.000 unidades), pois é guardada apenas uma cópia de cada par de fones que ocorre na língua. Alguns sons podem exigir *trifones* (triplas de fones) como em /tra/, /pra/, etc.

A síntese por difones e trifones pode produzir som de qualidade aceitável, mas os resultados podem ser melhorados com o uso de unidades ainda maiores (polifones). A desvantagem é o tamanho do dicionário, que pode ter mais de 5.000 unidades.

Alguns sistemas usam, em vez de um dicionário, uma extensa base de fala natural (com várias horas de duração), que é previamente segmentada e etiquetada com sua transcrição fonética. Para sintetizar um texto, um sistema desse tipo procura segmentos da base de fala (fones, polifones, palavras ou frases) que coincidem com trechos do texto dado, nos mesmos contextos, e concatena esses segmentos. Este método pode produzir fala sintetizada indistingüível da fala natural [55]; porém, a localização do segmento adequado, levando em conta o contexto, exige algoritmos sofisticados para ser efetuada com eficiência aceitável.

De qualquer forma, por maior que seja a base de unidades de fala, a síntese concatenativa não permite reproduzir todas peculiaridades de uma língua, como emoções, ênfases, palavras estrangeiras etc. Outra limitação intrínseca do método é que ele consegue produzir apenas uma voz — a do locutor que gerou o banco de unidades.

Um dos primeiros produtos comerciais a usar síntese concatenativa foi o *Echo*, um sintetizador de baixo custo lançado em 1982 pela Street Electronics. Outro sistema empregando esta tecnologia foi criado em 1985 por Olive e Liberman da AT&T, usando um amplo dicionário de morfemas de Coker e regras de conversão ortográfica-fonética de Church [32].

9.2.1 Concatenação suave

As unidades escolhidas para compor a fala não podem ser simplesmente justapostas, pois isso geralmente resulta num ruído audível (estalido) no momento da junção. O estalido decorre da diferença entre a última amostra do primeiro segmento e a primeira amostra do segmento seguinte, que é geralmente muito grande. Veja a figura 9.2.

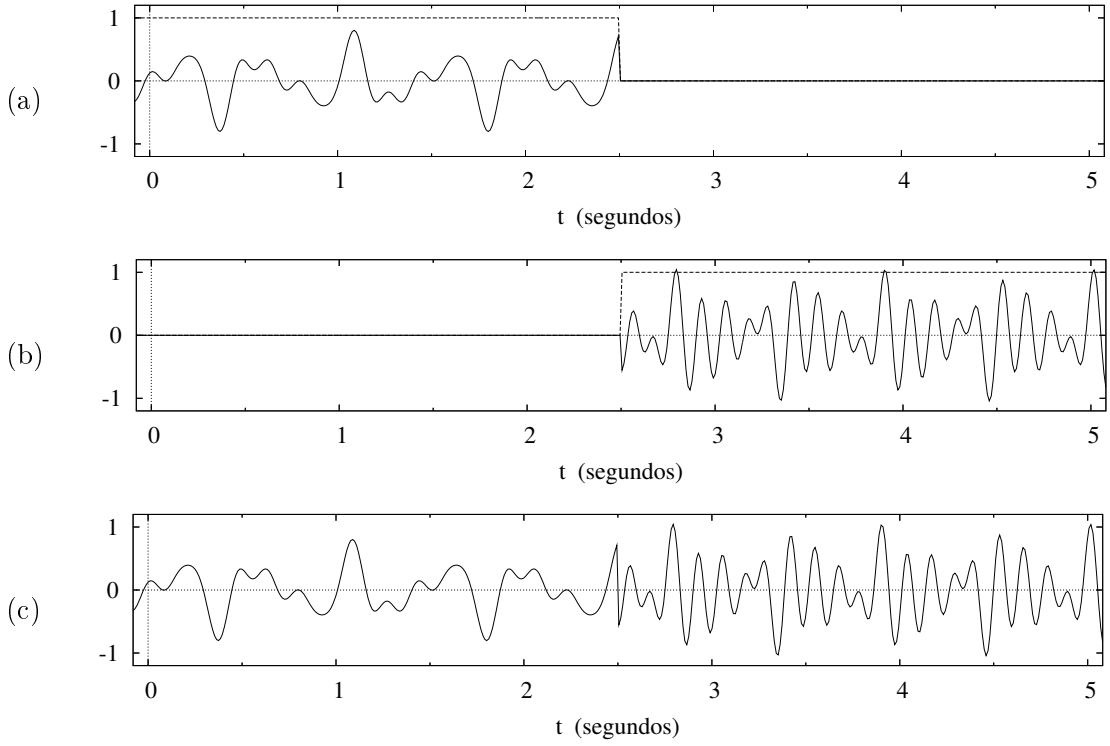


Figura 9.2: Concatenação de duas unidades de fala por simples justaposição. (a,b) As unidades separadas. (c) Resultado da concatenação. Note o salto em $t = 2,5$ s.

Para evitar este defeito, é necessário garantir uma transição gradual e suave entre os dois segmentos. A solução mais simples é calcular uma combinação linear dos dois sinais, alinhados de modo que eles se sobreponham por um certo número m de amostras. Nessa combinação, o peso do primeiro segmento cai de 1 para 0 no decorrer dessas m amostras, ao mesmo tempo que o peso do segundo aumenta de 0 para 1.

Para minimizar o ruído na junção, os pesos devem variar de maneira suave. Uma boa escolha são as duas metades da função de janelamento de Hann (descrita na seção 3.7). Veja a figura 9.3.

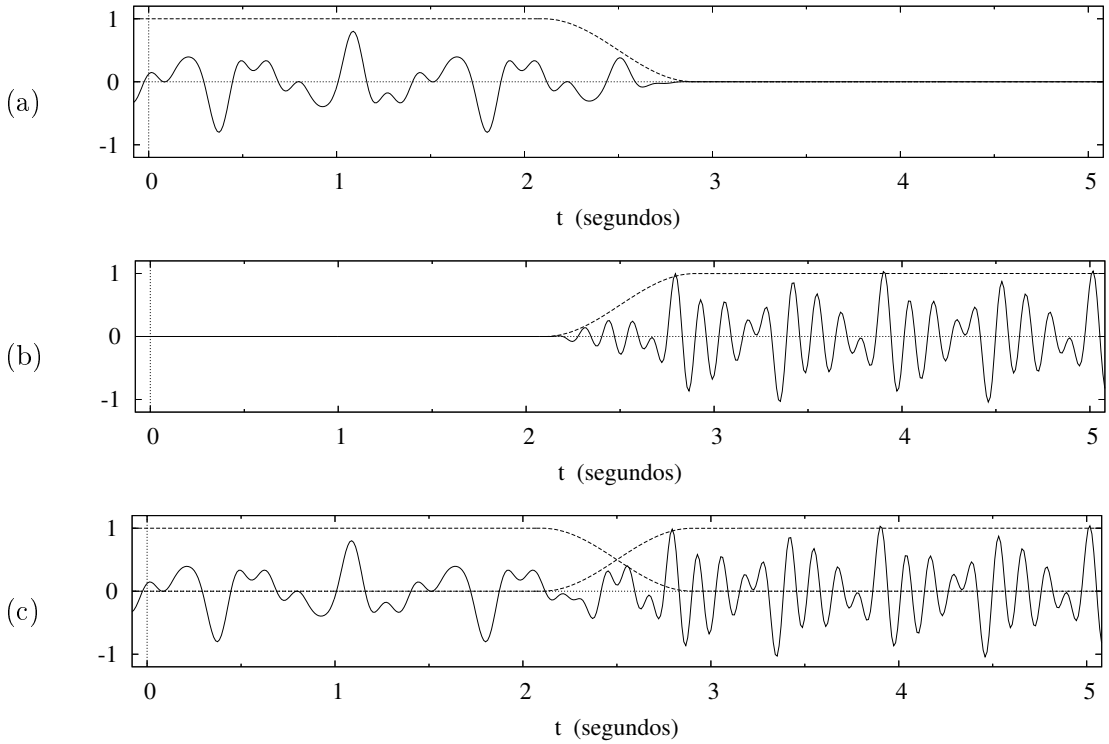


Figura 9.3: Concatenação suave de duas unidades de fala. (a,b) Unidades recortadas com janelamento de Hann. (c) Resultado da combinação linear com sobreposição.

9.2.2 Ajuste de duração

Para produzir o ritmo correto da fala, há necessidade de um algoritmo para aumentar ou diminuir a duração de uma determinada unidade da base u_i, de modo a obter uma duração especificada d_i.

Isso pode ser feito eliminando-se ou duplicando-se um trecho escolhido no meio do sinal. Neste caso, as emendas entre as várias partes do sinal devem ser feitas por combinação linear suave, como descrito na seção 9.2.1.

9.2.3 O método PSOLA

Na concatenação suave de dois sinais, é necessário escolher o número de amostras m da sobreposição. A solução mais popular é o método PSOLA (*Pitch Synchronous Overlap and Add*) originalmente desenvolvido pelo Centro Nacional de Estudos das Telecomunicações (CNET) da France Telecom [56].

O método PSOLA é simples de implementar, e é capaz de gerar um sinal de fala sintetizado de boa qualidade, inclusive com ritmo correto, a um baixo custo computacional. O método ainda permite alterar a altura da fala, o que pode ser importante para dar a entonação desejada.

Existem várias versões do algoritmo PSOLA, mas na sua essência todas funcionam da mesma maneira. Pela sua eficiência computacional, o mais utilizado é TD-PSOLA (*Time-Domain Pitch-synchronous Overlap and Add*) [43].

Basicamente, o algoritmo consiste de três passos. No primeiro passo, o sinal de fala original é decomposto em uma seqüência de sinais menores, parcialmente sobrepostos — denominados *sinais elementares*, ou *elementos* — cuja soma resulta no sinal original. No caso de sons sonoros (periódicos ou quase-periódicos), a duração dos sinais elementares equivale a um período fundamental; ou seja, $1/f$ segundos, onde f é a freqüência de vibração das pregas vocais. A função de janelamento de Hann é usada para recortar cada sinal elementar. Veja a figura 9.4.

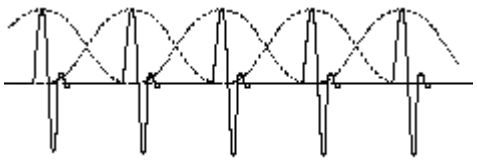


Figura 9.4: Decomposição de um sinal de voz em sinais elementares, pelo método TD-PSOLA.

O passo seguinte do algoritmo TD-PSOLA depende da alteração que se deseja fazer. Para di-

minuir a duração, deve-se omitir alguns de seus elementos. Para aumentá-la, deve-se duplicar alguns elementos. As figuras 9.5 e 9.6 ilustram cada caso.

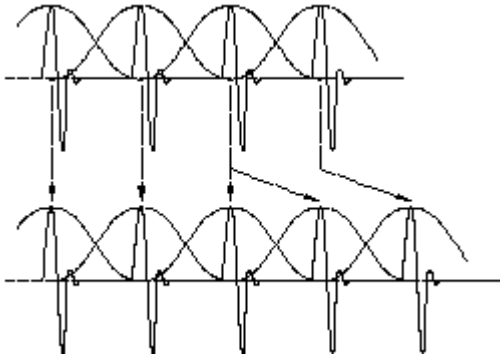


Figura 9.5: Aumento da duração de um sinal de voz por duplicação de sinais elementares.

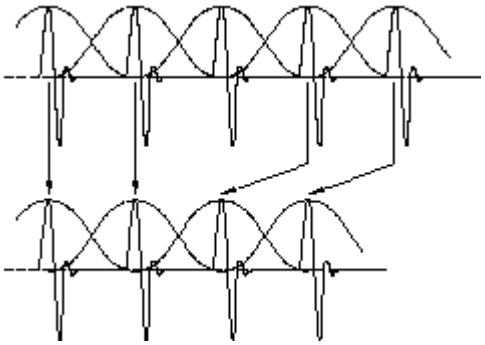


Figura 9.6: Redução da duração de um sinal de voz por omissão de sinais elementares.

O último passo do algoritmo consiste em somar os sinais elementares para obter o sinal sintetizado. Neste passo, o último elemento de cada unidade escolhida deve ser sobreposto ao primeiro elemento da unidade seguinte.

O método PSOLA não produz bons resultados no caso de sons aperiódicos. Uma vez que eles não possuem freqüência fundamental, o espaçamento dos sinais elementares deve ser escolhido arbitrariamente; e a duplicação de elementos torna o sinal sintetizado quase-periódico, produzindo sons "metálicos" [43].

9.2.4 Ajuste de altura

O método PSOLA também pode ser usado para ajustar a altura (freqüência fundamental) de um sinal de voz, modificando-se o intervalo de tempo entre sinais elementares consecutivos. Aumentando o intervalo de tempo diminuímos a freqüência, e vice-versa. As figuras 9.7 e 9.8 ilustram os processos.

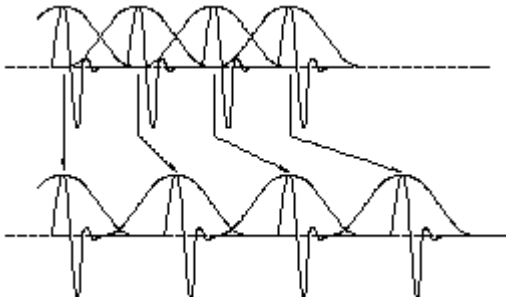


Figura 9.7: Redução da freqüência fundamental de um sinal.

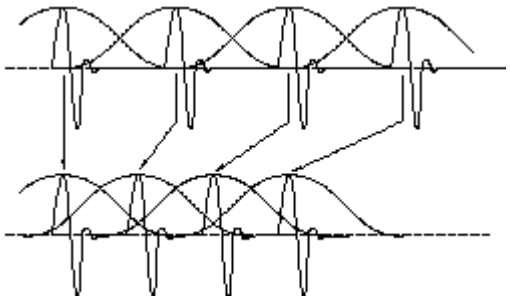


Figura 9.8: Aumento da freqüência fundamental de um sinal.

Este processo altera a duração do sinal, e portanto geralmente exige um ajuste apropriado da duração. Este método de ajuste de altura é relativamente fácil de implementar, mas não garante que os formantes do sinal alterado terão a freqüência e intensidade corretas (isto é, correspondentes à *mesma pessoa* enunciando as mesmas palavras com outra altura de voz).

9.3 Síntese por filtragem

Esta técnica se baseia no modelo fonte-filtro da teoria acústica da produção de fala. A idéia é utilizar fontes simples de sons "primordiais" para produzir um *sinal de excitação* similar aos sons produzidos pelas pregas vocais e/ou pelo fluxo turbulento do ar. Este sinal de excitação é modificado por um filtro cuja função de transferência se assemelha à do trato vocal.

O sinal de excitação u pode ser um trem de impulsos, com freqüência fundamental $f_\circ$ correspondente à freqüência de vibração das pregas vocais; ou um ruído aleatório (chiado) de espectro largo, simulando a turbulência produzida pelo fluxo do ar por uma abertura estreita. A primeira fonte é usada nos sons sonoros como vogais,/m/, /n/, /l/; a segunda nos fricativos como em /s/, /f/, /ch/, etc. Uma excitação mista pode ser usada para as consoantes fricativas sonoras como /z/, /v/, /j/, e alguns sons de aspiração. Em qualquer caso,o sinal primordial passa em seguida por um filtro, cuja função de transferência pode ser controlada, e por um amplificador com ganho variável. Veja a figura 9.9.

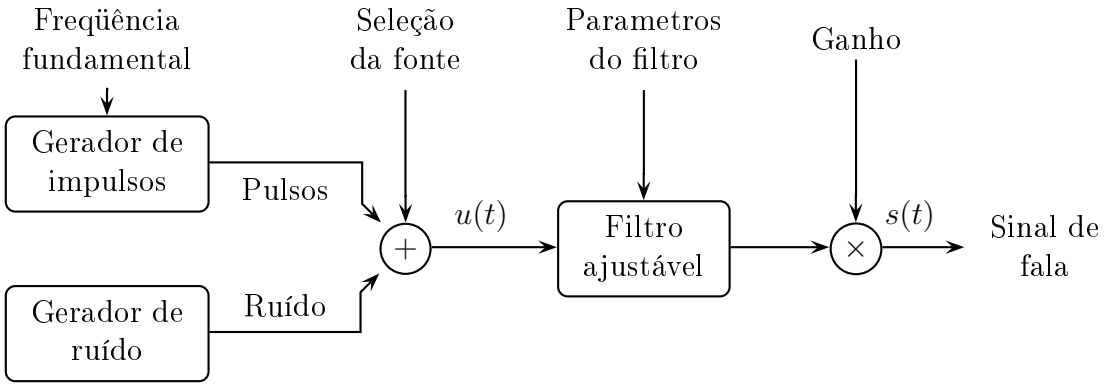


Figura 9.9: Modelo simplificado de síntese da fala por filtragem.

Os *parâmetros externos* que controlam o funcionamento deste sintetizador são a freqüência fundamental dos impulsos, a seleção entre as duas fontes (impulsos ou ruído), a função de transferência do filtro, e o ganho do amplificador (volume). Estes parâmetros devem ser atualizados pelo menos a cada 20 ms, período durante o qual o trato vocal pode ser suposto estacionário.

9.3.1 Sistemas mecânicos

Na verdade, a abordagem fonte-filtro é mais antiga que a síntese concatenativa. Ela tem sido utilizada desde o século 18, quando os cientistas começaram a compreender a natureza física do som e as características espectrais da fala humana.

Por exemplo, em 1779 o cientista dinamarquês Christian Kratzenstein apresentou em São Petersburgo (Rússia) um aparelho acústico-mecânico para demonstrar as diferenças fisiológicas entre as vogais. Na máquina de Kratzenstein, o sinal de excitação era produzido por palhetas (como as do oboé e outros instrumentos de sopro), e as diferentes vogais eram produzidas por ressonadores acústicos específicos [24, 34]. Veja a figura 9.10 abaixo.

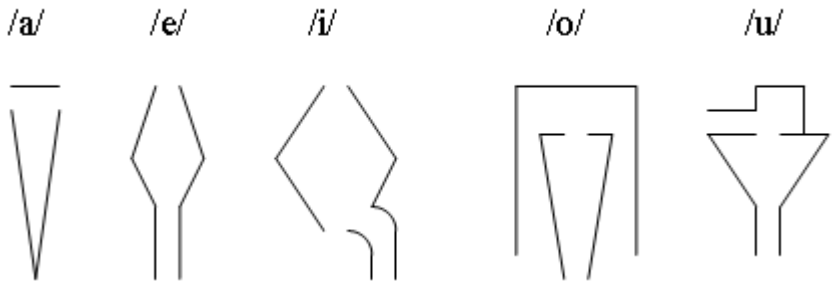


Figura 9.10: Os ressonadores de Kratzenstein (1779).

Outro sintetizador acústico-mecânico foi apresentado em 1791 por Wolfgang von Kempelen em Viena [24, 34]. Esta "máquina falante" podia produzir, além de vogais isoladas, também combinações de sons. As partes essenciais eram uma câmara de ar (simulando os pulmões), um par de palhetas (as pregas vocais) e um tubo flexível de couro (o trato vocal). Para produzir diferentes vogais, o operador manipulava o formato do tubo. As consoantes eram produzidas utilizando-se peças no interior do tubo. Tempos depois, o autor apresentou uma "máquina falante" melhorada, que produzia também a maioria das consoantes, inclusive as nasais. O projeto de von Kempelen inspirou muitos outros, como a "máquina falante" de Charles Wheatstone (1837), a *Euphonia* de M. Faber (1857) e o sintetizador de Paget (1923) [24, 71].

9.3.2 Sistemas elétricos

O primeiro sintetizador fonte-filtro inteiramente elétrico foi construído em 1922 por J. Stewart [58]. Esse aparelho era constituído de uma fonte de sinal de excitação e dois circuitos ressonantes, cujas freqüências poderiam ser independentemente ajustadas a fim de modelar os dois principais formantes do trato vocal. A máquina podia gerar sons isolados das vogais com dois formantes, mas não conseguia produzir as consoantes ou seqüência de fones [34]. Outro modelo semelhante, com quatro ressonadores ajustáveis, foi demonstrado por Wagner [24].

Um marco notável foi o sintetizador *Voice Operating Demonstrator* (*Voder*), apresentado por Homer Dudley dos Laboratórios Bell em 1939, na feira mundial de Nova York [24, 19, 44, 23]. Veja a figura 9.11.

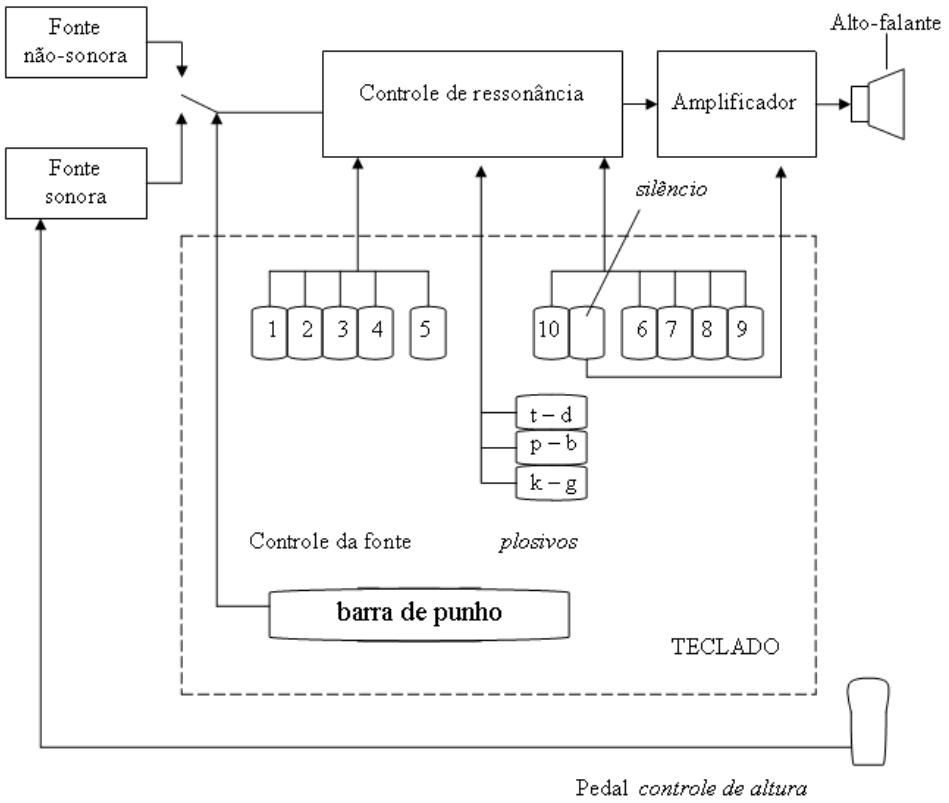


Figura 9.11: O sintetizador de fala *Voder* de Dudley (1939).

Este sistema foi inspirado pelo *Voice Coder* (*Vocoder*), um sistema de compressão de fala desenvolvido sigilosamente para fins militares. O filtro do *Voder* era formado por um banco de dez ressonadores separados, ligados em paralelo, com freqüências de ressonância fixas. Os controles externos consistiam de um teclado e um pedal. Uma tecla larga, acionada com a mão, selecionava a fonte de excitação, e o pedal controlava a freqüência fundamental da fonte sonora. Os ressonadores eram ativados por teclas acionadas com os dedos das duas mãos. Uma tecla adicional permitia reduzir o ganho do amplificador, para produzir silêncio, e três teclas separadas produziam os plosivos /t/-/d/, /p/-/b/ e /k/-/g/.

O funcionamento do *Voder* dependia muito da habilidade do operador; eram necessários meses de treinamento para produzir frases inteligíveis. Apesar da baixa qualidade da fala produzida, o *Voder* demonstrou que a produção da fala artificial era eminente, e atraiu a atenção de muitos cientistas para esse problema.

Assim, por exemplo, em 1951 Franklin Cooper e seus associados desenvolveram o sintetizador automático *Pattern Playback* nos Laboratórios Haskins (Inglaterra). Nesse sistema, o filtro era controlado por uma fita transparente, na qual era pintado o espectrograma de baixa resolução do som desejado, mostrando os formantes como faixas escuras [24].

Em 1953, Walter Laurence construiu o *Parametrical Artificial Talker* (PAT), outro sintetizador automático cujo filtro consistia de três ressonadores ajustáveis, ligados em paralelo. Os parâmetros do sistema eram as três freqüências dos ressonadores, a amplitude da fonte de ruído, e a amplitude e freqüência fundamental da fonte de impulsos. Estes parâmetros eram controlados por marcas numa placa de vidro móvel [34].

Em 1970, Richard Gagnon, um pesquisador solitário, desenvolveu um sistema original de síntese de fala baseado em formantes, controlado por tabelas de parâmetros. Este sistema deu origem ao chip Votrax SC-01 (1980), que foi usado em vários produtos comerciais de conversão texto-fala de baixo custo, como o Votrax *Type-n-Talk*.

De modo geral, na síntese por formantes, cada som exige o acionamento simultâneo de

vários ressonadores. Com ressonadores de freqüência fixa, o caráter da voz (homem, mulher, etc.) é fixo, e o repertório de fones é limitado. A fala produzida geralmente soa "robótica", e é difícil modelar corretamente as transições entre os diferentes sons.

9.3.3 Síntese por predição linear

Sintetizadores fonte-filtro modernos geralmente utilizam filtros digitais de predição linear, ou filtros LPC, descritos na seção 5.9.1. Veja a figura 9.12.

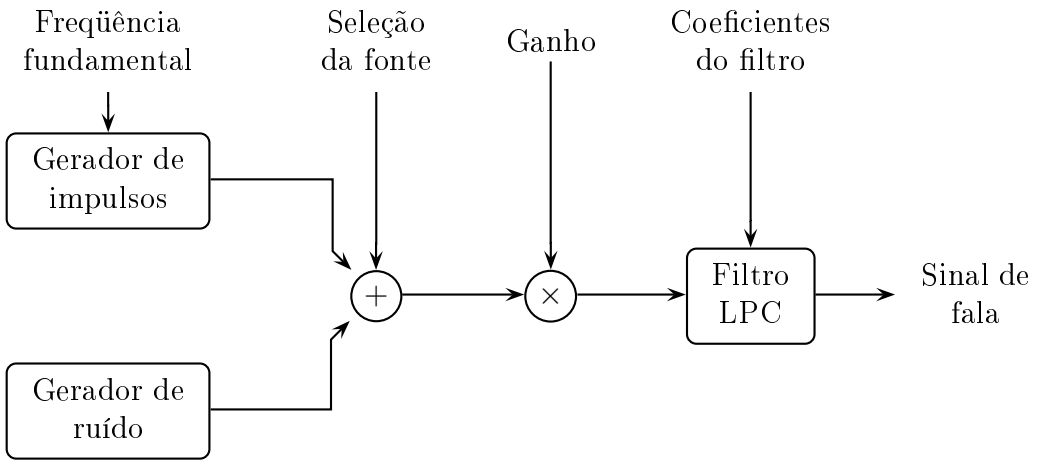


Figura 9.12: Esquema de um sistema de síntese utilizando filtro de predição linear.

Recordamos que, em um filtro LPC, cada amostra sinal de entrada (*excitação*) é somada a uma combinação linear das k últimas amostras do sinal de saída (*predição*). No caso de sinais de voz, verifica-se que a excitação pode ser reduzida a um trem de impulsos com certa freqüência, ou a um ruído aleatório; e que os coeficientes da predição são uma representação compacta mas muito flexível da função de transferência do trato vocal [52, 23].

9.3.4 Determinação dos parâmetros

Os parâmetros do sintetizador (sinal de excitação, ganho do amplificador, e coeficientes de predição linear do filtro) podem ser determinados pela análise de um ou mais exemplos do

segmento desejado (fone, palavra ou frase), gravados por um determinado locutor. O filtro de predição linear, definido pela fórmula (5.19), pode ser entendido como uma tentativa de prever o sinal desejado s_i por uma fórmula $\tilde{s}_i$

$$\tilde{s}_i = \sum_{k=1}^{p} a_k s_{i-k} \tag{9.1}$$

O objetivo da análise é escolher os parâmetros $a_1, \ldots, a_p$ de modo a minimizar o *erro* ou *resíduo*, que é a diferença e entre o sinal s e a predição linear $\tilde{s}$:

$$e_i \;=\; s_i - \tilde{s}_i \;=\; s_i - \sum_{k=1}^{p} a_k s_{i-k} \tag{9.2}$$

Mais precisamente, os coeficientes a_k para um determinado trecho do sinal, com n amostras, são escolhidos de forma a minimizar a soma Q dos erros sobre todas as n predições realizadas nesse trecho

$$Q = \sum_{m=0}^{n-1} e_{i-m}^2 = \sum_{m=0}^{n-1} \left| s_{i-m} - \sum_{k=1}^{p} a_k s_{i-m-k} \right|^2 \tag{9.3}$$

Os coeficientes a_k que minimizam a soma Q podem ser obtidos resolvendo-se o sistema de equações

$$\frac{\partial Q}{\partial a_k} = 0 \tag{9.4}$$

para $i = 1, 2, \ldots, p$. Este é um sistema de p equações lineares com p incógnitas $a_1, a_2, \ldots, a_p$.

Quando os parâmetros a_k são calculados desta forma para um sinal de voz, observa-se que o resíduo e se aproxima do sinal de excitação — um trem de impulsos para sons sonoros e um ruído aleatório para sons fricativos. Portanto, uma análise simples do resíduo e fornece os demais parâmetros do modelo fonte-filtro: a seleção da fonte, a freqüência fundamental (no caso de impulsos) e o ganho do amplificador.

Quando esta tecnologia é aplicada à transmissão de voz, o transmissor determina os coeficientes ótimos para um trecho do sinal de tamanho pré-fixado, e transmite apenas esses coeficientes, o tipo de excitação, o período de pitch e o ganho. O receptor reproduz o sinal de excitação, e soma a ele a predição linear calculada a partir das p amostras anteriormente geradas.

Os parâmetros do filtro devem ser atualizados a cada 10ms ou menos. Quando eles são obtidos a partir de análise de segmentos de fala natural, eles podem ser calculados, por exemplo, em janelas de cerca de 20 ms, com superposição de 10 ms.

A síntese baseada no modelo LPC produz som de boa qualidade com baixo custo computacional. Ela reproduz automaticamente as características da voz do falante, sem que seja necessário determinar a freqüência e intensidade dos formantes.

9.3.5 O dicionário falado *Speak-n-Spell*

Um dos primeiros produtos comerciais a usar esta tecnologia foi o dicionário falado *Speak-n-Spell*, lançado pela Texas Instruments em 1980. Este brinquedo educativo, projetado para ajudar crianças a aprender a ler e soletrar, utilizava um sintetizador fonte-filtro controlado por parâmetros LPC (*chip* TMS–5110), previamente gravados em memórias ROM [49]. Esta tecnologia também foi empregada pelo *chip* TMS-5220 da Texas Instruments, usado no conversor texto-fala *Echo* [32].

9.4 Síntese articulatória

Em princípio, a solução mais natural para produção da fala seria a *síntese articulatória*, que procura modelar diretamente a física do aparelho fonador humano (pregas vocais, língua, lábios, véu palatino, etc.).

Podemos enquadrar nesta abordagem o *Orator Verbis Eletris* (*OVE 1*) construído por Gunnar Fant em 1953, que usava ressonadores ajustáveis em cascata para modelar a modificação do som da glote pelos diferentes trechos do trato vocal [32]. Uma versão melhorada desse sistema, o *OVE 2*, foi apresentada por Fant e Martyony em 1962 [32]. Este sistema possuía partes separadas para modelar as funções de transferência do trato vocal para as vogais orais, vogais nasais e consoantes plosivas. As fontes excitatórias eram um trem de

pulsos periódicos para sons sonoros, uma fonte de ruído para aspiração, e outra fonte de ruído para os fricativos. Outras versões deste sistema *OVE 3* e *GLOVE* foram produzidas na Escola Superior Técnica Real (*Kungliga Tekniska Högskolan*, KTH) da Suécia. A companhia Infovox foi criada em 1983 para comercializar produtos que combinavam esta tecnologia com concatenação de difones, como o *Infovox AS-101*.

Outra linha de pesquisa em síntese articulatória começou em 1958 com o trabalho de George Rosen no Instituto de Tecnologia de Massachusetts (MIT) [32]. Seu sintetizador, o *Dynamic Analog of Vocal tract* (DAVO), foi melhorado por Hecker em 1962 [32], pela adição de um modelo acústico da cavidade nasal. Outro exemplo recente dessa abordagem é o projeto ASY de P. Rubin e L. Goldstein, do Laboratório Haskins [54].

Deve-se notar também que o primeiro conversor texto-fala completo para a língua inglesa, desenvolvido por Noriko Ueda e outros em 1968, utilizava um modelo articulatório para síntese de baixo nível. Em 1973, James Flanagan e Kenzo Ishizaka, pesquisadores dos Laboratórios Bell da AT&T, usaram um modelo de síntese articulatório para gerar sentenças, usando dados de controle derivados do sistema TTS Coker et al (1973).

Devido à complexidade do aparelho fonador humano, o modelo de síntese articulatória é considerado o mais difícil de implementar. Apesar de ter sido um modelo bastante popular entre pesquisadores na década de 1990, ele ainda não é usado em sistemas comerciais.

9.5 Síntese baseada em cadeias de Markov

Uma dificuldade presente em todos os modelos de síntese de fala é reproduzir corretamente as variações de duração, volume, e outros parâmetros de cada fone que dependem do contexto. Estas variações são essenciais para uma pronúncia "humana" em vez de "robótica". As *cadeias de Markov* são um modelo matemático muito usado para modelar essas variações [37, 8].

9.5.1 Cadeias de Markov gerais

Uma cadeia de Markov é essencialmente um autômato finito probabilístico. Formalmente, ela consiste de um conjunto Q de *estados*, e um conjunto de *símbolos de saída*. Para cada estado há uma distribuição de probabilidades sobre os símbolos, e outra distribuição de probabilidade sobre os estados [76, 38, 39, 66, 67]. Veja a figura 9.13..

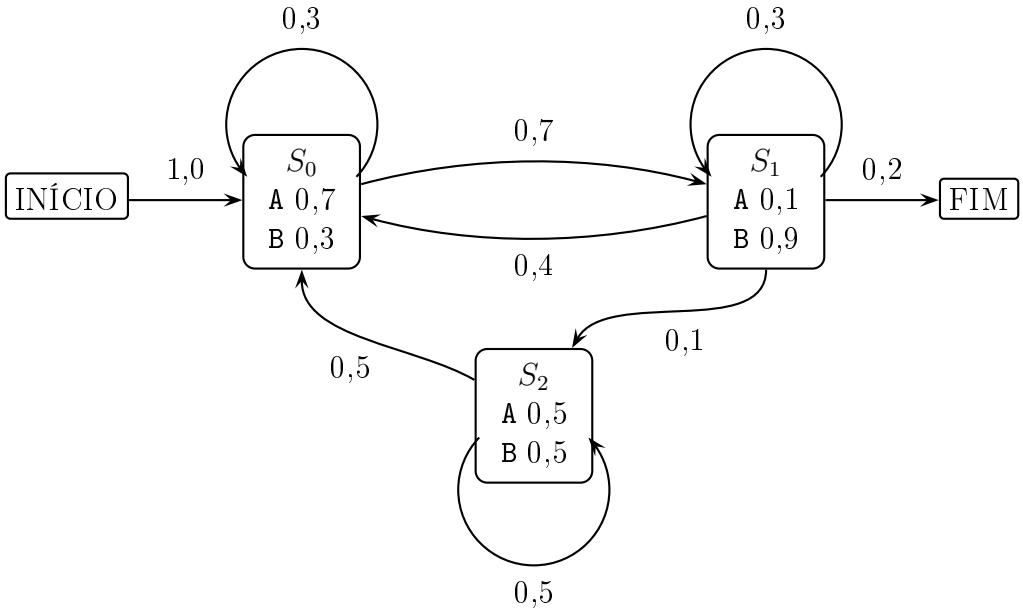


Figura 9.13: Exemplo de uma cadeia de Markov.

Uma cadeia de Markov é geralmente entendida como um gerador aleatório de seqüências de símbolos. A cada passo na geração de uma seqüência, ocorre a *emissão* de um símbolo, e ocorre uma *transição* para outro estado ou para o mesmo estado, dependendo do estado corrente.

Mais precisamente, a geração começa com a escolha aleatória de um estado s, de acordo com as probabilidades das transições que saem de um estado específico, o *estado inicial*. A cada passo, as probabilidades de emissão associadas ao estado s são usadas para escolher um símbolo de saída x, que é concatenado à seqüência de saída. Em seguida, as probabilidades de transição associadas a s são usadas para escolher um novo estado s', que substitui o estado s. Este processo é repetido até que seja atingido um estado designado como *estado final*.

A probabilidade de que uma cadeia de Markov percorra uma determinada seqüência de estados é o produto das probabilidades de todas as transições ao longo do mesmo. A probabilidade que ela gere uma determinada seqüência de símbolos ao longo desse caminho é o produto das probabilidades desses símbolos nos estados correspondentes. A probabilidade total da cadeia produzir uma seqüência de símbolos é a soma dessas probabilidades para todos os caminhos possíveis.

Assim, por exemplo, a cadeia de Markov da figura 9.13 pode seguir o caminho $C = s_0 s_1 s_2 s_0 s_1 s_1$ com probabilidade $\Pr(C) = 1,0 \times 0,7 \times 0,1 \times 0,5 \times 0,7 \times 0,3 \times 0,2 = 0,00147$. Por esse caminho, ela pode gerar a seqüência $s =$ `AABBBA` com probabilidade $\Pr(C, S) = \Pr(C) \times 0,7 \times 0,1 \times 0,5 \times 0,3 \times 0,9 \times 0,1 = 0,00147 \times 0,000945 = 0,00000138915$. E assim por diante.

9.5.2 Modelos de Markov para palavras isoladas

Para síntese de fala de palavras isoladas, cada palavra do vocabulário é modelada por uma cadeia de Markov com uma estrutura específica ilustrada na figura 9.14 [29].

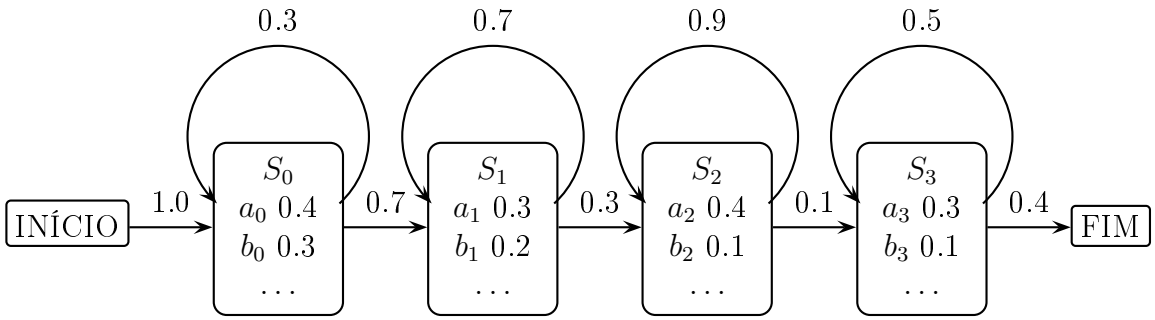


Figura 9.14: Exemplo de uma cadeia de Markov usada para modelar uma palavra falada. Os símbolos de saída são a_i, b_i, etc.

Cada estado desta cadeia tem duas transições, uma para si próprio e uma para o estado seguinte. Cada símbolo de saída a_i, b_i, etc. é um vetor de parâmetros do sintetizador fonte-filtro,

referente a uma janela temporal do sintetizador. Por exemplo, no caso de um sintetizador baseado em LPC, cada símbolo de saída especificaria a seleção da fonte, o ganho do amplificador e os coeficientes a_k do filtro, válidos para uma janela de 5 ms.

As probabilidades das transições podem ser ajustadas para produzir a duração desejada de cada fone. Em particular, se a transição de um estado para si mesmo tem probabilidade p, a cadeia permanecerá nesse estado por k quadros com probabilidade $p^{k-1}(1-p)$.

Na modelagem de fala contínua, cada fone é modelado por uma cadeia de Markov, e essas cadeias são concatenadas para formar uma única cadeia para uma palavra ou frase.

9.6 Conclusões

A tecnologia de síntese da fala continua progredindo e encontrando novas aplicações. Hoje em dia, o modelo concatenativo de síntese ainda é o mais usado nas aplicações onde a naturalidade é mais importante do que a flexibilidade de locutor — como, por exemplo, em sistemas de conversão texto-fala. O modelo fonte-filtro ainda é importante em telecomunicações.

Apesar de ainda serem poucos os produtos que utilizam métodos estocásticos de síntese, como cadeias de Markov, o interesse nessas técnicas cresceu na década de 1990, impulsionada pelo aumento na velocidade e capacidade de memória dos computadores.

Uma preocupação geral em todas as aplicações é aperfeiçoar a naturalidade da fala produzida pelo sintetizador, que é uma tarefa difícil, porque envolve a complexidade da prosódia.

Capítulo 10

Reconhecimento de fala

Reconhecimento de fala é o processamento de sinais de voz para identificar as palavras que foram ditas. O resultado do reconhecimento pode ser apenas a transcrição (fônica ou ortográfica) do que foi dito, ou o acionamento de uma aplicação ou dispositivo mecânico.

A *identificação do locutor* pelas características de sua voz é uma necessidade presente em muitas aplicações; por exemplo, em controle de acesso, transações bancárias por telefone, ou na autenticação de gravações telefônicas para fins judiciais [64]. No entanto, este é um problema bem distinto do reconhecimento de fala, que não abordaremos neste trabalho.

10.1 Aplicações

Sistemas de reconhecimento de fala comerciais estão disponíveis desde os anos 1990. Apesar do aparente sucesso da tecnologia, atualmente ainda são poucas as pessoas que a usam para interagir com seus computadores. A maioria dos usuários ainda prefere editar e criar documentos por meio de periféricos tradicionais, como teclado e *mouse*, ou por reconhecimento de escrita manual (especialmente em *palmtops*). A bem da verdade, a tecnologia de reconhecimento de fala ainda está muito longe do ideal.

Ditado: Sistemas para produção de documentos por ditado são a aplicação mais comum de reconhecimento de fala. Tais sistemas são utilizados para produzir receitas e relatórios médicos [15, 59], apólices de seguro, relatórios jurídicos, artigos jornalísticos, etc. Estas aplicações exigem vocabulários extensos e sistemas adaptáveis ao locutor. Em geral, são operados em ambientes sem muito ruído, como escritórios isolados, utilizando microfones de cabeça para minimizar ruídos.

Telefonia: Hoje já existem inúmeros serviços comerciais e sociais por telefone, que dependem do reconhecimento de fala. Podemos citar, por exemplo, consultas de listas telefônicas, serviços bancários, reserva de passagens ou entradas, envio de fax ditado, etc. Certas operadoras oferecem a possibilidade de discar telefones por fala.

Processamento de documentos falados: Sistemas de reconhecimento de fala são essenciais para o gerenciamento e processamento de arquivos de gravações, por exemplo de telefonemas grampeados por ordem judicial. A conversão das gravações em texto permite estender o uso de sistemas de busca e recuperação textual (como o *Google*) a tais arquivos.

Comando e Controle: Sistemas de reconhecimento de fala podem ser usados para controlar vários dispositivos através de comando de voz. Por exemplo, uma pessoa pode gerenciar sua caixa postal por telefone, comandando por voz a reprodução ou eliminação de mensagens. A mesma técnica poderia ser usada para ligar ou desligar lâmpadas, fornos e panelas elétricas, sistemas de segurança, etc. Em particular, comandos de voz podem ser usados para abrir ou fechar portas mediante senhas — como no conto clássico de Alí Babá.

Educação: A tecnologia de reconhecimento de fala pode auxiliar o aprendizado de língua estrangeira, incluindo correção de defeitos de pronúncia. Ela pode ser útil também na tradução de aulas e palestras, e possibilitar o diálogo entre professores e alunos que falam línguas diferentes (sistemas de tradução/diálogo).

Apoio a deficientes físicos: Pessoas que têm dificuldade em usar o teclado, devido a limitações físicas, podem utilizar o computador ativando-o via voz [22]. Deficientes auditivos podem utilizar sistemas de reconhecimento de fala para atender chamadas telefônicas. Num futuro próximo, tais sistemas podem permitir que alunos com deficiência auditiva aproveitem aulas tão bem quanto alunos com audição normal.

10.2 Tipos de Reconhecedores

Reconhecedores de fala podem ser classificados por vários critérios: restrições sobre a forma de elocução, flexibilidade quanto ao locutor, tamanho do vocabulário, assunto, etc.

10.2.1 Tamanho do vocabulário

Todo sistema de reconhecimento de fala tem um *vocabulário* ou *dicionário*, uma lista de palavras que o sistema deve reconhecer. Geralmente, quanto menor o vocabulário, mais fácil e mais preciso é o reconhecimento. Vocabulários com até vinte palavras são considerados de porte pequeno, até cem palavras são de tamanho médio, até mil são grandes e com mais do que mil palavras são muito grandes. Vale notar que o ser humano adulto é geralmente capaz de reconhecer 50.000 ou mais palavras distintas de sua língua nativa [1].

10.2.2 Precisão

A principal qualidade de um reconhecedor de fala é sua habilidade de reconhecer corretamente palavras e frases que estão em seu dicionário, e rejeitar aquelas que não estão. A percentagem de identificações corretas, para sinais típicos da aplicação, é chamada de *precisão*.

A precisão depende muito da qualidade do sinal captado e do nível de ruído presente no ambiente. Por essa razão, é recomendável usar microfone fixado na cabeça.

10.2.3 Natureza da elocução

Alguns sistemas são capazes de reconhecer apenas *palavras isoladas*, obrigatoriamente separadas por pausas. Outros sistemas são capazes de reconhecer seqüências de *palavras concatenadas* formando blocos de estrutura simples, desde que haja pausas entre os blocos. Por exemplo, um sistema deste tipo pode ser capaz de reconhecer número de telefone como 3287-3232 (*pausa-trêsdoisoitosetetrêsdoistrêsdois-pausa*). Os sistemas mais avançados são capazes de reconhecer *fala contínua* tal como enunciada normalmente por pessoas em conversa natural, onde nem sempre há pausas entre as palavras.

10.2.4 Dependência de locutor

Sistemas de reconhecimentos de fala podem ser classificados também pela sua flexibilidade quanto ao locutor. *Sistemas dependentes de locutor* são projetados para uma determinada pessoa; eles reconhecem corretamente a fala desse usuário, mas não necessariamente a de outra pessoa. Os sistemas *adaptáveis ao locutor* permitem a mudança de usuário, através de uma sessão de treinamento que consiste na repetição pelo novo locutor de frases padronizadas, conhecidas pelo sistema. Os sistemas *independentes de locutor* são projetados para reconhecer a fala de qualquer pessoa (homem, mulher, criança, etc.), sem necessidade de re-treinamento.

Atualmente, há reconhecedores adaptáveis ao locutor que, depois de treinamento adequado, conseguem reconhecer a fala contínua, com um vocabulário grande e num ritmo normal, com cerca de 98% de precisão.

10.2.5 Assunto

Muitos reconhecedores são projetados para reconhecer palavras, frases ou textos específicos de uma determinada área ou aplicação; por exemplo, nomes, dígitos, números de telefone, comandos de computador, termos médicos ou farmacêuticos, termos jurídicos, etc.

10.3 Técnicas para reconhecimento da Fala

Na maioria das técnicas de reconhecimentode fala, a identificação das palavras é feita por comparação de certos parâmetros do sinal de fala com parâmetros correspondentes das palavras do vocabulário [77]. Dependendo da aplicação, pode ser necessária também uma análise sintática das palavras reconhecidas, para juntá-las em sentenças e extrair seu sentido. Veja a figura 10.1.

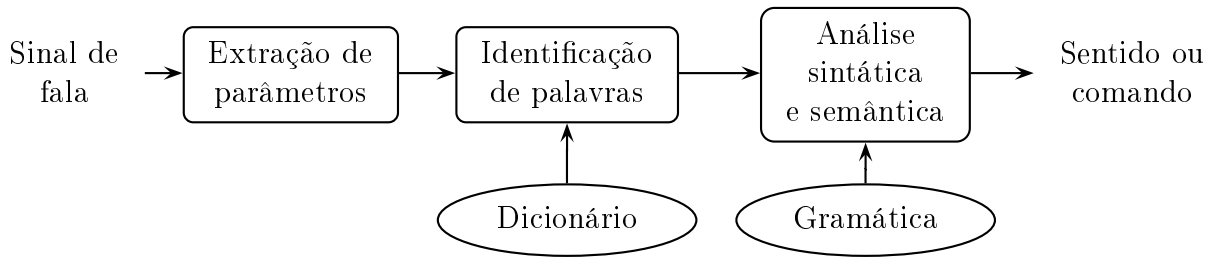


Figura 10.1: Estrutura típica simplificada de um sistema de reconhecimento de fala [51, 53, 76].

Há muitos fatores que dificultam o reconhececimento da fala humana, mesmo para um locutor fixo. Por exemplo, a mesma palavra pode ter pronúncias diferentes (timbre, ritmo, intensidade, etc.) dependendo do contexto em que é utilizada, ou do estado físico e emocional do locutor.

Na fala contínua a dificuldade é ainda maior, pois as palavras normalmente são emendadas, sem pausas. Neste caso, o sistema precisa também efetuar, em algum momento, a *segmentação* do sinal da voz em palavras, sílabas ou fonemas. Outro problema é a complexidade e flexibilidade da língua. A fala natural geralmente contém erros de gramática, repetições e correções, bem como elocuções que não fazem parte da frase, por exemplo: *hum,aha*, *ehe*, ou cacoetes de linguagem tais como *né*, *pois é*, etc.

As tecnologias mais usadas para identificação das palavras são: *redes neurais*, *modelos ocultos de Markov* e *modelos híbridos* [38].

10.3.1 Redes neurais naturais

Para comparação com as tecnologias artificiais, é interessante examinar a estrutura do cérebro humano, que é responsável pelo reconhecimento da fala a partir dos sinais enviados pela cóclea (seção 6.3.1). O cérebro humano é constituído de aproximadamente 100 bilhões de células especializadas (*neurônios*), conectadas entre si por junções eletro-químicas chamadas *sinápses*, formando uma enorme rede (*rede neural*).

O neurônio consiste de um *corpo celular*, várias ramificações denominadas *dendrites* e uma fibra nervosa denominada *axônio*. Veja a figura 10.2.

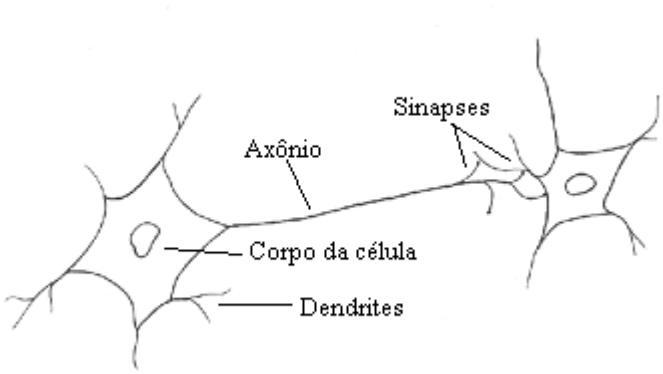


Figura 10.2: Estrutura simplificada de neurônios e suas conexões.

O cérebro processa a informação na forma de impulsos elétricos. Um impulso gerado por um neurônio percorre o axônio até as sinápses na sua extremidade. Em cada sinápse, o impulso elétrico gera uma descarga de gotículas de uma substância química denominada *neuro-transmissor*. Essa substância atravessa a sinápse atingindo a membrana de outro neurônio, geralmente numa dendrite. Quando essa nova célula recebe uma quantidade suficiente de neuro-transmissor, ela inicia outro impulso, repetindo o processo [70].

Apesar de intensamente investigadas há mais de 100 anos, a estrutura dos "circuitos" do cérebro (as ligações entre neurônios) e suas funções ainda são praticamente desconhecidas, especialmente quanto ao processamento da fala.

10.3.2 Redes neurais artificiais

As *redes neurais artificiais* (RNA; em inglês, *artificial neural networks*, ANN) são modelos matemáticos para reconhecimento de padrões em geral, vagamente inspirados na estrutura do cérebro humano [64, 38].

Uma rede neural artificial consiste de uma coleção de elementos simples de processamento, os *neurônios artificiais*, dispostos em uma ou mais camadas. Veja a figura 10.3.

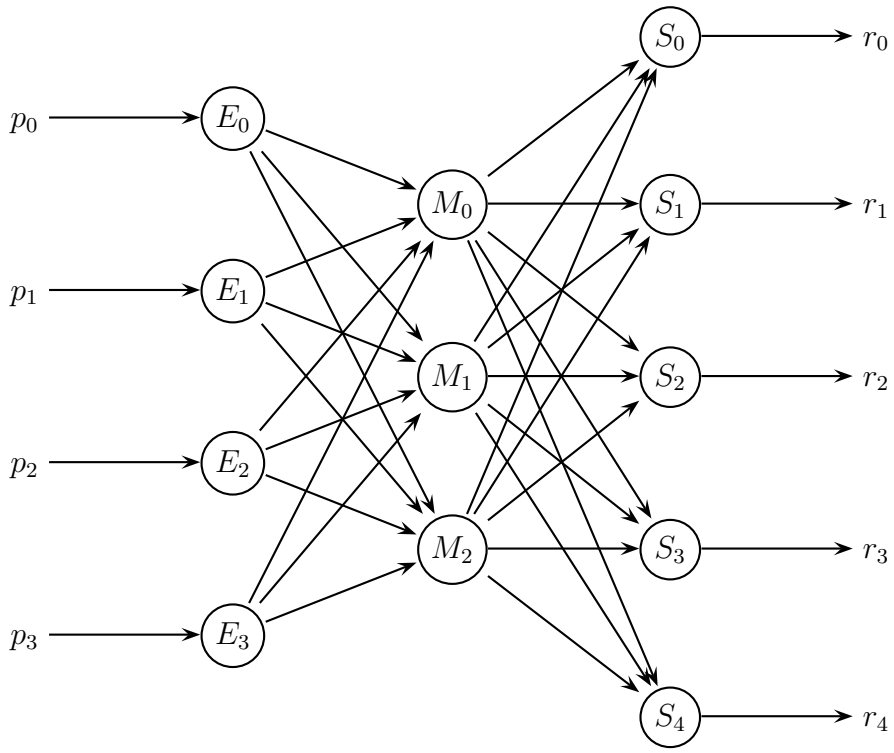


Figura 10.3: Ilustração de uma rede neural artificial com três camadas.

Tipicamente, as entradas e saídas são números reais entre -1 e $+1$. Cada neurônio artificial tem uma ou mais entradas, e uma única saída (que pode ser ligada às entradas de vários outros elementos da camada seguinte). Por definição, cada neurônio calcula uma combinação linear L de suas entradas, cada qual com um peso próprio, e produz na saída uma função não-linear (*função de ativação*) de L, variando entre -1 e $+1$.

Normalmente, cada entrada p_i da rede corresponde a um parâmetro extraído do objeto

a classificar, e cada saída r_j a uma categoria da classificação. Para usar uma RNA, os parâmetros são alimentados aos elementos da camada de entrada da rede, e as saídas de todos neurônios são calculadas, camada por camada. A saída r_j com maior valor determina a categoria atribuída ao objeto. Ajustando-se os pesos das entradas, é possível fazer com que cada saída r_j seja uma função bastante complexa das entradas $p_0, p_1, \ldots$.

Antes de ser usada para reconhecer padrões, uma RNA deve passar por uma fase de *treinamento* ou *aprendizado*. Nessa fase, a RNA é alimentada com dados cuja classificação é conhecida. Os pesos de cada elemento são ajustados até que as saídas correspondam à classificação correta.

Num sistema de reconhecimento de fala baseado em RNA, as entradas $p_0, p_1, \ldots$ são parâmetros extraídos do sinal de áudio, que são considerados relevantes para o reconhecimento — por exemplo, uma versão grosseira do espectrograma da palavra; e cada saída r_j corresponde a uma palavra do vocabulário. Na fase de treinamento, o sistema mostra cada palavra do vocabulário ao usuário, que deve pronunciá-la várias vezes.

Segundo a literatura, redes neurais artificiais são capazes de reconhecer apenas palavras (ou frases curtas) isoladas e vocabulários pequenos.

10.3.3 Modelos Ocultos de Markov

A teoria dos *modelos ocultos de Markov* (*hidden Markov models*, HMM) para reconhecimento de fala foi desenvolvida por L. Baum, da Universidade de Princeton, no final da década de 1960, e implementada pela primeira vez por J. Baker da Universidade Carnegie-Mellon e F. Jelinek da IBM, no início da década de 1970. Apresentamos aqui uma versão extremamente simplificada da abordagem, pois uma descrição completa seria extremamente complexa e fugiria ao escopo deste trabalho. Mais detalhes podem ser encontrados nos livros de Rabiner e Juang [51] e Jelinek [29].

Nesta abordagem, cada palavra do vocabulário a reconhecer é modelada por uma cadeia de Markov, como descrito na seção 9.5, sendo que as probabilidades dos símbolos de saída e das transições refletem as variações de pronúncia esperadas entre o(s) usuário(s) do sistema. Veja a figura 10.4. Diz-se que uma cadeia de Markov tem *estados ocultos* quando não é possível deduzir o estado em que a mesma se encontra a partir do símbolo emitido no mesmo; ou seja, quando vários estados diferentes podem emitir o mesmo símbolo. As cadeias usadas para modelar a fala humana geralmente possuem essa característica; daí o nome comum de *modelo oculto de Markov* (*hidden Markov model*, HMM) para essa técnica.

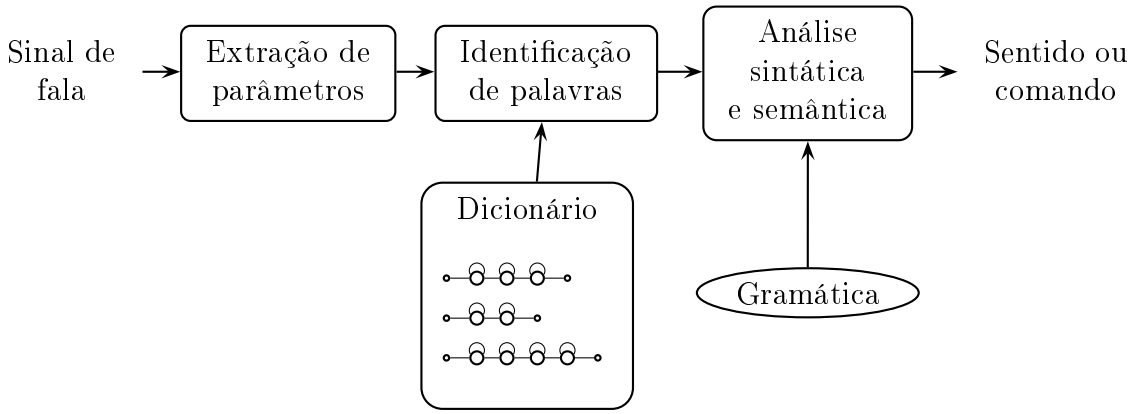


Figura 10.4: Ilustração de um sistema de reconhecimento de fala baseado em cadeias de Markov.

Antes de ser usado, o reconhecedor precisa passar por uma fase de treinamento, na qual as probabilidades dos símbolos e transições das cadeias de Markov são determinadas através da análise de vários exemplos gravados de fala natural. Para se construir um reconhecedor independente de locutor, o treinamento do sistema é feito por diversos locutores com vozes, timbres e sotaques variados.

Durante a operação do sistema, o sinal da fala é dividido em quadros, por exemplo, de 10ms em 10ms, e cada quadro é analisado e reduzido a um vetor de parâmetros (volume, coeficientes do filtro LPC, etc.). Ou seja, o sinal é transformado numa seqüência $x = x_0x_1x_2...x_m$ símbolos de saída da cadeia de Markov. O passo seguinte é determinar, para cada elemento v do dicionário, a probabilidade que a cadeia de Markov correspondente a v produza a seqüência

x. Por meio do Teorema de Bayes, calcula-se então a probabilidade do usuário ter enunciado v. A palavra v com maior probabilidade é então escolhida pelo sistema.

A tecnologia HMM ainda é a favorita de um grande número de pesquisadores e empresas, inclusive IBM, Philips, AT&T, Dragon Systems e outros. Ela funciona bem para o reconhecimento de palavras isoladas, pela sua tolerância em relação a variações naturais na duração e características dos fones. Ela é indispensável para o reconhecimento de fala contínua, principalmente pela sua capacidade de detectar os limites de cada palavra mesmo na ausência de pausas.

10.3.4 Sistemas Híbridos

Há sistemas híbridos, que combinam as duas tecnologias — redes neurais e modelos ocultos de Markov. Por exemplo, pode-se usar modelos ocultos de Markov para os fones, em que a distribuição de probabilidade para a emissão de parâmetros é implementada por redes neurais [38].

10.4 Histórico

Um dos primeiros dispositivos de reconhecimento de fala que se tem notícia é o *Radio Rex*, um brinquedo fabricado em 1922. Segundo David e Selfridge [21], citado por Roweis [53], tratava-se de um cachorrinho de celulóide, preso por um eletro-íma dentro de uma casinha fixada a uma base metálica. Quando o dono pronunciava a palavra *Rex*, as componentes com freqüência próxima a 500 Hz do fone /e/ faziam vibrar uma barra metálica dentro da base, que desligava o eletro-íma. Uma mola então fazia o cachorrinho saltar para fora de sua casinha.

As primeiras tentativas em reconhecimento de fala iniciaram-se ainda na década de 1950, com dispositivos eletrônicos para reconhecimento de fonemas e dígitos isolados [53]. Na

década de 1960, o reconhecimento de fala passou a ser um dos tópicos de pesquisa em inteligência artificial. Porém, a dificuldade do problema logo ficou evidente. Em 1969, John Pierce dos Laboratórios Bell [48] publicou uma carta bastante crítica da pesquisa do seu tempo:

> *... General purpose speech recognition seems far away. Social-purpose speech recognition is severely limited. It would seem appropriate for people to ask themselves why they are working in the field and what they can expect to accomplish ... We are safe in asserting that speech recognition is attractive to money. The attraction is perhaps similar to the attraction of schemes for turning water into gasoline, extracting gold from the sea, curing cancer, or going to the moon. ... To sell suckers, one uses deceit and offers glamour ... Most recognizers behave, not like scientists, but like mad inventors or untrustworthy engineers. ...*

O maior projeto de pesquisa em reconhecimento de fala de todos os tempos parece ter sido o *Speech Understanding Research* (SUR) lançado por Laurence Roberts na Agência de Projetos Avançados de Pesquisa da Defesa (DARPA) em 1971. Este projeto, cujo objetivo era desenvolver um sistema computadorizado capaz de entender a fala contínua, consumiu cerca de três milhões de dólares por ano durante cinco anos, e envolveu a Universidade Carnegie-Mellon (CMU), os laboratórios do SRI, o Laboratório Lincoln do MIT, a Systems Development Corporation (SDC) e a Bolt, Beranek e Newman (BBN).

Em 1975, pesquisadores da Carnegie-Mellon University (CMU) introduziram dois reconhecedores dependentes de locutor: *Dragon*, com vocabulário de 194 palavras e, logo após, Hearsay, com um vocabulário de 1.011 palavras.

Dentre os primeiros produtos comerciais para reconhecimento de fala, adaptáveis ao locutor, citamos o *Prima* da Ericson Business Systems (1982) e o *Tangora* da IBM (1985). Ambos estavam limitados a palavras isoladas, e o Tangora tinha um vocabulário de 5000 palavras. Um dos primeiros sistemas capazes de reconhecer fala contínua foi o *Byblos* da Bolt, Beranek

and Newman (BBN), com vocabulário de 997 palavras. Sistemas com vocabulário grande o bastante para uso em ditado, mas ainda com palavras isoladas, foram lançados por várias empresas (incluindo Dragon, IBM, Kurzweil e Philips) a partir de 1995. O primeiro sistema para ditado contínuo foi o *Naturally Speaking* da Dragon Systems, lançado em 1997 [15]. Um produto para ditado muito difundido nos Estados Unidos é o *Via Voice* da IBM.

Dentre os sistemas independentes de locutor, vale mencionar um reconhecedor de palavras isoladas criado pelos Laboratórios Bell em 1982, com vocabulário de 129 termos. Nos anos seguintes, a Covox produziu reconhecedores de fala recreativos (*The Voice Master* e *The Speech Thing*) para o Commodere 64, o Atari 400/800 e o IBM PC. Os Laboratórios Bell lançaram um reconhecedor de dígitos isolados em 1988. Outros sistemas com aplicações comerciais foram o *Teleton* da Deutsche Bundespost Telekom (1988), o *MarieVox* da France Telecom (1990), o *Teledialogue* da dinamarquesa Jydsk Telefon (1992) e o *Audiotex* da espanhola Telefônica, todos com vocabulário pequeno. Já o *WorldWindow* da Global Communication Ltd. (1992) tinha um vocabulário grande.

Um exemplo notável dessa tecnologia foi o sistema *VoiceBroker* da corretora de valores Charles Schwab (1996), desenvolvido em parceria com a Nuance. O sistema atendia até 360 clientes simultaneamente, com 95% de precisão. Outras empresas, tais como Sears, Roebuck and Co., UPS e E*Trade Securities seguiram o exemplo.

Mais detalhes sobre a história do reconhecimento da fala podem ser encontrados em vários artigos e documentos WWW [63, 16, 73, 59].

Parte IV

Projeto prático

Capítulo 11

O Projeto `karacat`

11.1 Introdução

Com o objetivo de consolidar os conhecimentos adquiridos durante a elaboração desta monografia, implementamos um sintetizador para canções populares japonesas, que chamamos `karacat`.

A entrada do sintetizador é um arquivo contendo a partitura da música e a transcrição fônica da letra da canção, sincronizadas. A saída é um arquivo de áudio digital, contendo uma voz humana que interpreta a canção (sem acompanhamento).

O método usado é síntese concatenativa, baseada num dicionário de sílabas pré-gravadas em todas as alturas necessárias. Para cada sílaba da canção, o programa extrai do dicionário o trecho correspondente, na altura correta, e ajusta sua duração de acordo com a duração da nota indicada na partitura.

O programa pode ter alguma utilidade para a comunidade dos amantes do *karaoke*. Por exemplo, se uma pessoa deseja aprender uma música mas tem apenas a partitura e a letra da música utilizando este sistema, ela poderá ouvir a música cantada sinteticamente.

11.2 Estrutura do programa

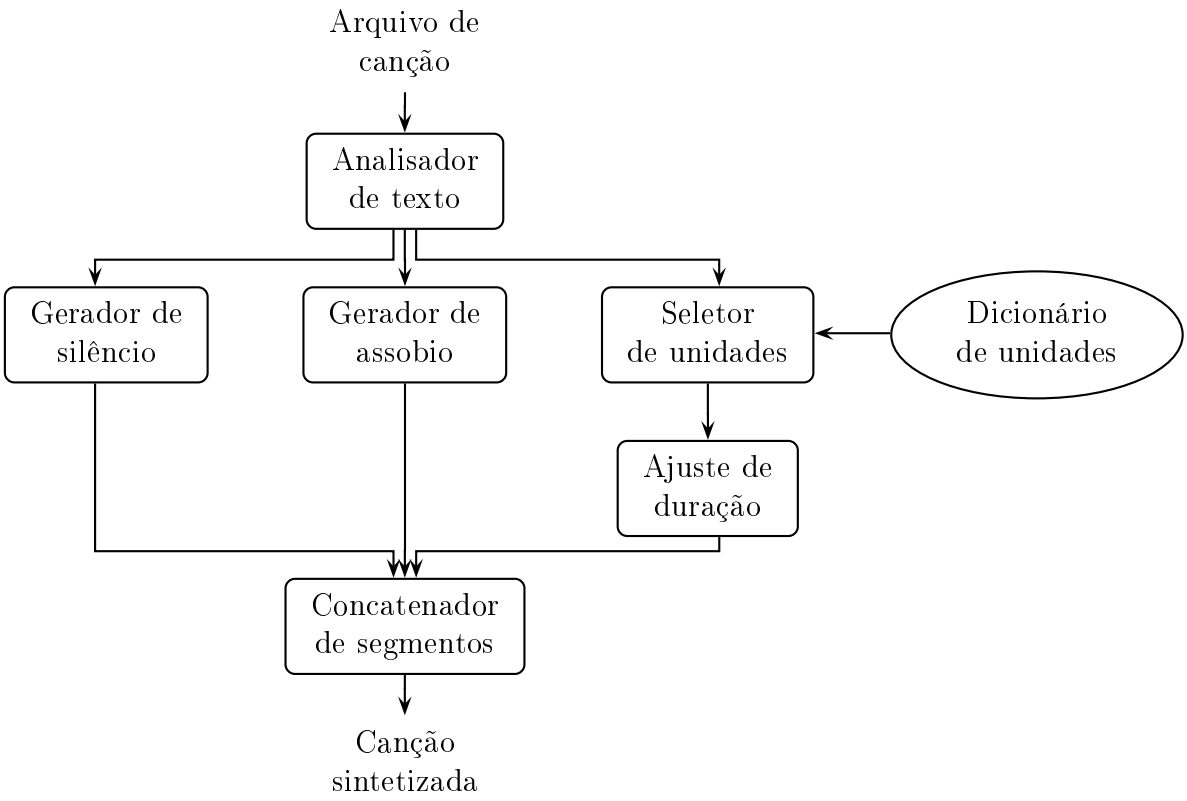


Figura 11.1: Esquema de blocos do programa `karacat`.

A figura 11.1 mostra a estrutura do `karacat`. O módulo *analisador de texto* lê o arquivo de entrada e produz uma seqüência de triplas (a_i, d_i, f_i), uma para cada nota, onde a_i identifica a sílaba japonesa a ser cantada, d_i é a duração e f_i sua altura.

Para cada tripla, o módulo *seletor* procura no dicionário um segmento pré-gravado e_i, correspondente à sílaba a_i cantada na altura f_i. Caso o segmento e_i seja encontrado no dicionário, ele passa para o módulo *ajuste de duração*, que estica ou encolhe o sinal pré-gravado, de modo a obter a duração desejada d_i. Caso o par (a_i, f_i) não seja encontrado no dicionário, o módulo *gerador de assobio* cria um elemento artificial, consistindo de um assobio de freqüência f_i e duração d_i. No caso de uma pausa, o *módulo de silêncio* gera uma série de amostras nulas com duração d_i.

Os segmentos produzidos por estes módulos são emendados pelo *concatenador de segmentos* e gravados em um arquivo no formato *Sun Audio* (extensão "`.au`"), que pode ser tocado pela maioria dos *players* de áudio.

11.3 Resumo da fonética do idioma japonês

11.3.1 Fones da língua japonesa

A língua japonesa foi escolhida porque tem uma fonética relativamente simples, comparada com as do português ou inglês [69, 31]. A língua japonesa moderna tem apenas 14 consoantes simples, três consoantes duplas, 5 vogais, e quatro ditongos, relacionados na tabela 11.1.

Consoantes simples					
/k/	***k**amikaze*	***c**asa*	/g/	***g**akkō*	***g**ato*
/s/	***s**akura*	***s**apo*	/z/	***z**en*	***z**ero*
/ch/	*su**sh**i*	***ch**ato*	/t/	***t**aiko*	***t**udo*
/d/	***d**enwa*	***d**ado*	/n/	***n**inja*	***n**ada*
/h/	***h**ai*	***h**ere* (ing.)	/f/	***f**uji*	***f**oto*
/p/	***p**inku*	***p**ato*	/b/	***b**onsai*	***b**ola*
/m/	***m**oyashi*	***m**uda*	/r/	***r**aku*	*ca**r**o*
Consoantes duplas					
/ts/	***ts**unami*	*pa**ts**y* (ing.)	/tch/	*ba**ch**an*	***ch**urch* (ing.)
/dj/	***j**ūdō*	***j**ob* (ing.)			
Vogais					
/a/	***a**rigatō*	***a**ro*	/i/	***i**chi*	***i**sto*
/u/	***u**don*	***u**ma*	/e/	***e**bi*	***e**ste*
/o/	***o**rigami*	***o**vo*			
Ditongos					
/ya/	***ya**ma*	*p**ia**no* (ita.)	/yu/	*fu**yu***	***you*** (ing.)
/yo/	***yo**shi*	***io**gurte*	/wa/	***wa**tashi*	*q**ua**se*

Tabela 11.1: Os fones da língua japonesa. A notação dos fones usada na primeira coluna é baseada na ortografia portuguesa. Os exemplos de palavras do japonês na segunda coluna usam a grafia Hepburn, baseada na grafia inglesa mas muito usada no Brasil. A terceira coluna mostra sons semelhantes do português, inglês ou italiano.

As vogais /e/ e /o/ do japonês são sempre fechadas, como no português *selo* e *toda*. O som /f/ é bilabial, e não lábio-dental como em português; e o som /r/ nunca é vibrado.

11.3.2 Sílabas da língua japonesa

Além de ter um número relativamente pequeno de fones, a língua japonesa têm restrições significativas quanto à maneira que esses fones podem ser combinados.

Cada sílaba pode conter apenas uma vogal ou ditongo, precedida opcionalmente de uma consoante simples ou dupla, e seguida opcionalmente do som /n/. Na verdade, dentre todas as combinações com esta estrutura, apenas uma centena são sílabas válidas do idioma. Veja a tabela 11.2.

–	/a/	/i/	/u/	/e/	/o/	/ya/	/yu/	/yo/
k–	/ka/	/ki/	/ku/	/ke/	/ko/	/kya/	/kyu/	/kyo/
g–	/ga/	/gui/	/gu/	/gue/	/go/	/gya/	/gyu/	/gyo/
s–	/sa/	*	/su/	/se/	/so/	*	*	*
z–	/za/	*	/zu/	/ze/	/zo/	*	*	*
ch–	*	/chi/	*	*	*	/cha/	/chu/	/cho/
t–	/ta/	*	*	/te/	/to/	*	*	*
d–	/da/	*	*	/de/	/do/	*	*	*
tch–	*	/tchi/	*	*	*	/tcha/	/tchu/	/tcho/
dj–	*	/dji/	*	*	*	/dja/	/dju/	/djo/
ts–	*	*	/tsu/	*	*	*	*	*
n–	/na/	/ni/	/nu/	/ne/	/no/	/nya/	/nyu/	/nyo/
h–	/ha/	/hi/	*	/he/	/ho/	/hya/	/hyu/	/hyo/
f–	*	*	/fu/	*	*	*	*	*
b–	/ba/	/bi/	/bu/	/be/	/bo/	/bya/	/byu/	/byo/
p–	/pa/	/pi/	/pu/	/pe/	/po/	/pya/	/pyu/	/pyo/
m–	/ma/	/mi/	/mu/	/me/	/mo/	/mya/	/myu/	/myo/
r–	/ra/	/ri/	/ru/	/re/	/ro/	/rya/	/ryu/	/ryo/
w–	/wa/	*	*	*	*	*	*	*

Tabela 11.2: Os sons silábicos da língua japonesa falada que terminam em vogal. A notação é a mesma da tabela 11.1. As sílabas /gue/ e /gui/ são pronunciadas como em português (com "*u*" mudo). As sílabas marcadas com "*" não ocorrem na língua japonesa, exceto em palavras estrangeiras não assimiladas.

Cada sílaba da tabela 11.2 pode ocorrer com vogal curta (por exemplo, /no/) ou longa (/nō/). Todas as sílabas da tabela 11.2 podem ocorrer seguidas de um som /n/. Além disso, na língua japonesa (como na língua italiana), a consoante de certas sílabas pode ser "dobrada", afetando o sentido da palavra. Por exemplo, *neko* significa *gato*, enquanto que *nekko* significa *raiz*. Na pronúncia, esse detalhe corresponde a uma pequena pausa antes da consoante.

11.3.3 Ortografia japonesa

Nas grafias silábicas japonesa *hiragana* e *katakana*, os sons da tabela 11.2 são indicados de maneira mais complexa. Por exemplo, em *hiragana*, a sílaba /djō/ da tabela é escrita com o caracter da sílaba /shi/, modificado por um acento ″ para indicar glotalização e pelo caracter da sílaba /yo/ (subscrito) para indicar o ditongo, seguido pelo sinal da sílaba /u/ para indicar o prolongamento da vogal /o/.

A grafia japonesa *hiragana* distingue a sílaba /ji/ de /dji/, e /zu/ de /dzu/. Porém, muitos autores e dicionários consideram esses pares fonicamente equivalentes [31, 69, 41].

Há vários sistemas para transcrição de palavras japonesas para o português ou inglês, que usam notações diferentes para os fones da tabela 11.2. Por exemplo, no sistema Hepburn para transcrição do japonês com letras latinas (originalmente desenvolvido para falantes do inglês, mas muito usado em português), a sílaba /djō/ da tabela seria escrita *joo* ou *jō* [69, 41]. No sistema Shin-Kurei-Shiki (usado em alguns livros-texto ocidentais para ensino de japonês), essa sílaba é escrita *zyō* [31].

11.3.4 Fônica das canções populares japonesas

Nas canções japonesas, as sílabas longas da língua geralmente são cantadas como notas de duração maior, respeitando-se o ritmo da música. Veja a figura 11.2.

(a) *. . . bashō wa nasake ni . . .*

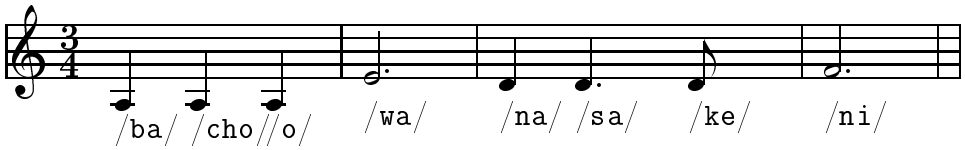
(b) 

Figura 11.2: Um verso da canção popular *Bashōfu*. (a) O verso na transcrição Hepburn. (b) Transcrição fônica alinhada com a partitura.

Foneticamente, quando o fone /n/ ocorre no final de uma sílaba, ele é parte da mesma. Porém, nas escritas japonesas *hiragana* e *katakana*, ele é escrito como caracter separado do silabário; e, em canções, ele é geralmente cantado como uma nota separada. Veja a figura 11.3. O programa karacat leva em conta este fato, e trata /n/ como uma sílaba separada.

(a) *. . . asa ji kun ji no . . .*

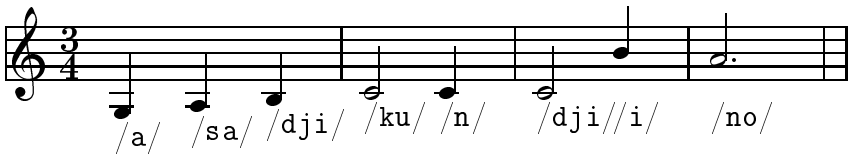
(b) 

Figura 11.3: Outro verso da canção *Bashōfu*. (a) Na transcrição Hepburn. (b) transcrição fônica alinhada com a partitura.

O programa `karacat` ainda não tem tratamento especial para as consoantes dobradas; mas esse detalhe seria relativamente simples de implementar.

11.3.5 Criação do dicionário de sons

Por conveniência, o dicionário foi produzido na forma de um arquivo separado para cada sílaba, contendo essa sílaba cantada em 10 a 13 alturas diferentes na escala Dó (C). Para gravação foi utilizado o programa `audacity` [45]. A voz da cantora foi digitalizada com freqüência de amostragem $f_* = 44.100$ Hz, monaural. Após a gravação, ajustamos a amplitude de cada instância da sílaba com o `audacity`, de modo que todas as instâncias tivessem

aproximadamente o mesmo volume (1 dB abaixo da amplitude máxima). Nas sílabas iniciadas por consoantes sustentáveis, como /s/ ou /m/, tomamos cuidado para que a consoante fosse a mais curta possível, a fim de evitar problemas no ajuste de duração. Veja a figura 11.4.

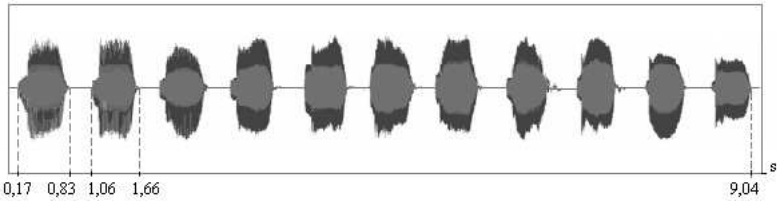


Figura 11.4: Gráfico da pressão para a sílaba *ma*, cantada em 11 alturas distintas (de G3 a C5).

O sinal digital de cada gravação foi armazenado num arquivo no formato Ogg-Vorbis com qualidade 9 (máxima) chamado ⟨*sílaba*⟩`.ogg`, onde ⟨*sílaba*⟩ é a sílaba na notação da tabela 11.2. Estes arquivos foram posteriormente convertidos para o formato Sun AU com codificação de 16 bits PCM linear pelo comando Linux

`sox -V` ⟨*sílaba*⟩`.ogg -s -w` ⟨*sílaba*⟩`.au` (11.1)

11.3.6 Leitura e segmentação do dicionário

A primeira etapa do programa lê estes arquivos AU, e constrói uma representação do dicionário na memória. Para isso, o arquivo ⟨*sílaba*⟩`.au` deve ser dividido nos trechos correspondentes a cada nota. A segmentação é descrita em um arquivo texto chamado ⟨*sílaba*⟩`.pic`, preparado manualmente.

Cada linha do arquivo ⟨*sílaba*⟩`.pic` consiste de quatro campos. Veja a figura 11.5.

```
0174 0684 0220 C#3
0913 1453 0220 D3
1663 2260 0220 F3
2438 3009 0220 G3
3166 3696 0220 A3
3895 4419 0220 C4
4628 5158 0220 D4
5360 5849 0220 E4
6069 6539 0220 F4
6768 7291 0220 G4
7533 7985 0220 A4
8220 8699 0220 A#4
8963 9392 0220 A5
```

Figura 11.5: Exemplo de arquivo de segmentação ⟨*sílaba*⟩`.pic`.

Os dois primeiros campos de cada linha são o início e o fim do trecho do som a ser segmentado, em milissegundos, contados a partir do início do arquivo. O terceiro campo é o início da vogal da sílaba, em milissegundos contados a partir do início do segmento. O quarto campo é a altura do som do trecho considerado.

O campo de início da vogal é usado pelo módulo de ajuste de duração, como explicado na seção 11.4. A altura é indicada por uma letra na escala `C`, `D`, `E`, `F`, `G`, `A`, `B` (correspondente às notas musicais dó–si), por um sinal de sustenido `#` opcional, e por um algarismo que indica a oitava. Nessa notação, `C0` significa a nota dó da oitava 0, uma nota com freqüência fundamental $f_\circ = 16,351$ Hz (um pouco abaixo do limite inferior de audibilidade, e bem abaixo do limite inferior da voz humana), enquanto que a fundamental de `B#9` é o "si sustenido" da oitava 9 (ou seja, dó da oitava 10), com $f_\circ = 16.267$ Hz (ainda audível, mas bem acima do limite de produção da voz humana).

Na memória do computador, cada elemento do dicionário é armazenado na forma de uma quádrupla (a, f, h, v), onde a é uma cadeia de caracteres identificando a sílaba, f é a freqüência fundamental correspondente à nota cantada, h é a duração da "cabeça" da sílaba (vide seção 11.4) e v é a seqüência de amostras correspondente.

11.3.7 Formato do arquivo da canção

Na segunda fase, o programa lê a letra e música de uma canção de um *arquivo de canção*, chamado ⟨*título*⟩`.kar`, e escreve o arquivo de áudio sintetizado, chamado ⟨*título*⟩`.au`.

Cada linha do arquivo de canção tem três campos (figura 11.6). O primeiro campo é a sílaba a ser cantada, na notação da tabela 11.2. O segundo campo é a duração da nota musical. A unidade básica de duração é a semi-colcheia, que o programa por enquanto supõe que tem a duração fixa de 0,25 segundos. O terceiro campo é a altura da nota musical, na mesma notação usada no arquivo de segmentação ⟨*sílaba*⟩`.pic`.

```
u     4 G4
mi    4 G4
no    3 G4
-     1 -
a     8 G4
o     2 G4
o     2 A4
sa   10 G4
a     2 F4
ni   10 E4
-     2 -
so    8 C4
ra    2 C4
a     2 D4
no    8 E4
a     2 E4
a     2 F4
o    10 D4
-     2 -
```

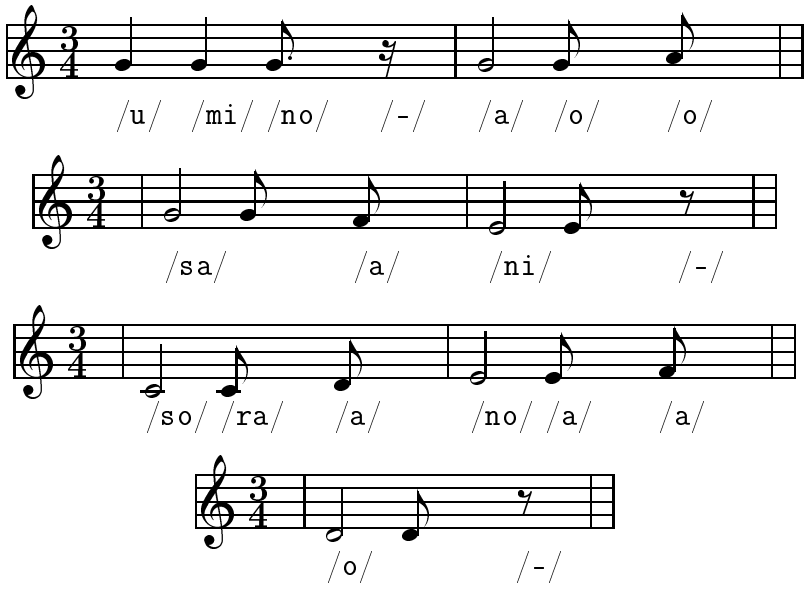


Figura 11.6: Exemplo de arquivo de canção ⟨*título*⟩`.kar`.

11.4 Ajuste de duração

O módulo de ajuste de duração precisa esticar ou encolher um elemento do dicionário, de duração d'_i, de modo a obter um segmento com duração especificada d_i. Para esse fim, o

programa escolhe um trecho adequado do elemento (o *miolo*), definido por dois instantes a e b, que é retirado ou repetido conforme necessário.

11.4.1 Escolha do miolo

O miolo é sempre escolhido dentro vogal final da sílaba, pois essa é a parte cuja duração é normalmente ajustada por um cantor humano. O início do miolo é indicado por um campo do dicionário (originalmente lido do arquivo ⟨*sílaba*⟩`.pic`). Este parâmetro deve ser suficiente para pular as consoantes iniciais e a semi-vogal inicial dos ditongos, mesmo nas sílabas como /`tcho`/ e /`ryo`/.

Para encolher o elemento, o instante final b do miolo é escolhido de modo que sua duração $b - a$ seja maior ou igual à diferença $\Delta = d'_i - d_i$. Veja a figura 11.7.

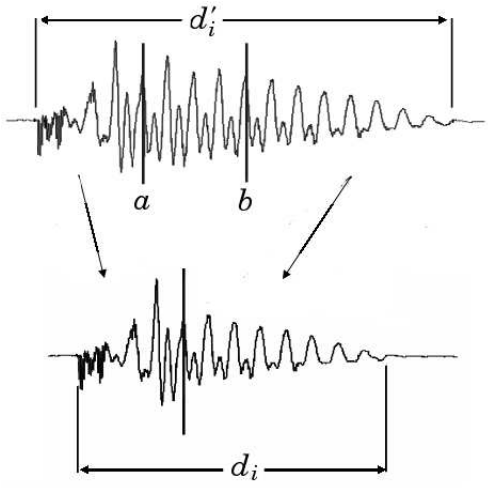


Figura 11.7: Encolhendo uma sílaba. Sílaba original (no alto) dividida em cabeça, miolo e cauda, e a sílaba encolhida (em baixo).

Para esticar o elemento, o instante final b do miolo é escolhido de modo que $b - a$ seja igual ao período fundamental $T_\circ = 1/f_\circ$. Veja a figura 11.8.

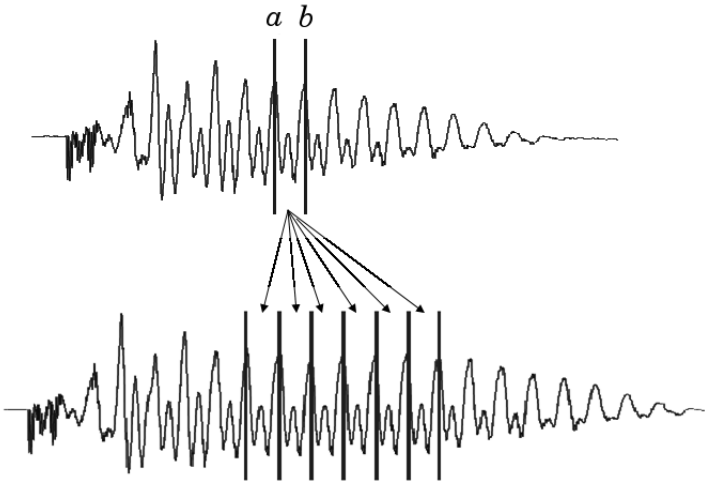


Figura 11.8: Alongando uma sílaba. Sílaba original (no alto) dividida em cabeça, miolo e cauda, e a sílaba alongada (em baixo).

Nos dois casos, o valor exato de b é escolhido de modo a sincronizar o corte com a freqüência fundamental $f_\circ$ da vogal, como no método TD-PSOLA, conforme detalhado na seção 11.4.2.

Este método simples de ajuste seria bastante inadequado se os elementos fossem fones ou polifones arbitrários. Com certeza, ele produziria resultados muito inferiores aos do método TD-PSOLA — que, em vez de um único miolo, duplica ou remove vários sinais elementares distribuídos ao longo de todo o segmento. No nosso caso, entretanto, sabemos que a vogal final da sílaba é um sinal quase-periódico, com freqüência fundamental bem conhecida e cuja forma de onda é praticamente a mesma em cada ciclo (exceto por variações graduais de volume).

11.4.2 Sincronização dos cortes

Para sincronizar as posições dos cortes a e b, usamos o fato de que a freqüência fundamental $f_\circ$ da vogal é aproximadamente igual a f_i, a freqüência nominal com que a sílaba foi cantada. Portanto, calculamos uma estimativa superior $T = 1,2/f_\circ$ para o período fundamental $T_\circ$, e escolhemos o valor de b de tal forma que o trecho do sinal no intervalo $[a - T/2, a + T/2]$

seja o mais parecido possível com o trecho em $[b - T/2, b + T/2]$, a menos de um ajuste de volume.

Mais precisamente, seja $n = \lfloor T f_*/2 \rfloor$ o número de amostras contidos em um trecho de duração $T/2$ do sinal, onde f_* é a freqüência de amostragem. Sejam i e j os índices de duas amostras. A similaridade $V(i,j)$ do sinal na vizinhança desses dois instantes é definida pela fórmula

$$V(i,j) = \frac{\displaystyle\sum_{k=-n}^{n} r_{i+k} r_{j+k}}{\sqrt{\left(\displaystyle\sum_{k=-n}^{n} r_{i+k}^2\right)\left(\displaystyle\sum_{k=-n}^{n} r_{j+k}^2\right)}} \tag{11.2}$$

Esta fórmula é o *coeficiente de correlação* entre as amostras dos dois trechos do sinal de duração T centrados em r_i e r_j. Ela pode ser interpretada como o cosseno do ângulo entre os dois trechos, considerados como vetores de $\mathbf{R}^{2n+1}$. Portanto, o valor de $V(i,j)$ é 1 quando os dois trechos diferem apenas por um fator de escala (ganho), e menor que 1 em todos os outros casos.

No programa karacat, o índice i_a da amostra inicial do miolo é fixado em $\lfloor a f_* \rfloor$, onde a é o tempo do início do elemento ao início do miolo, especificado no dicionário. O programa calcula $V(i_\mathrm{a}, j)$ variando o índice j entre $j_\mathrm{min} = \lfloor b_\mathrm{min} f_* \rfloor$ e $j_\mathrm{max} = j_\mathrm{min} + n$, onde b_min é o valor mínimo para o fim do miolo (que depende do objetivo, esticar ou encolher). O valor de j que fornece o maior valor de $V(i_\mathrm{a}, j)$ define o índice i_b da amostra final do miolo, e portanto o instante correspondente $b = i_\mathrm{b}/f_*$.

11.4.3 Concatenação com ajuste de volume

No ajuste da duração de um elemento, cada trecho selecionado do mesmo é concatenado ao sinal de saída pelo procedimento

$$\texttt{concatena\_suave}(s, m, r,\ i_\mathrm{a}, \gamma_\mathrm{a}, w_\mathrm{a},\ i_\mathrm{b}, \gamma_\mathrm{b}, w_\mathrm{b})$$

O parâmetro s é a seqüência de amostras do sinal que está sendo sintetizado, e m é o índice nominal da última amostra no mesmo. O parâmetro r é a seqüência de amostras do elemento do dicionário. Os parâmetros i_a e i_b são índices de amostras em r. O procedimento recorta um trecho do som r que vai da amostra $r[i_\mathrm{a}]$ (inclusive) até $r[i_\mathrm{b}]$ (exclusive), e soma esse trecho ao som s, alinhando a amostra $r[i_\mathrm{a}]$ com a amostra $s[m]$. Os parâmetros γ_a, w_a, γ_b e w_b especificam detalhes do recorte nesses dois pontos, como explicado mais adiante.

O recorte é feito usando uma função de janelamento que começa com uma meia-janela de Hann crescente, com largura w_a, tem valor 1 entre $r[i_\mathrm{a}]$ e $r[i_\mathrm{b}]$, e termina com uma meia-janela de Hann descrescente de largura w_b. Portanto, o trecho recortado na verdade começa na amostra $r[i_\mathrm{a} - w_\mathrm{a}]$ e termina na amostra $r[i_\mathrm{b} + w_\mathrm{b} - 1]$, sendo que as primeiras $w_\mathrm{a} + 1$ e as últimas $w_\mathrm{b} + 1$ amostras são multiplicadas pela função de janelamento.

O sinal recortado e ajustado é somado ao sinal s, alinhado de tal forma que a amostra $r[i_\mathrm{a}]$ é somada a $s[m]$. Portanto, o sinal s é alterado a partir da amostra $s[m - w_\mathrm{a}]$. A função também soma ao parâmetro m o comprimento nominal $i_\mathrm{b} - i_\mathrm{a}$ do trecho copiado. Note que este comprimento não inclui as "abas" do trecho recortado criadas pela função de janelamento. Portanto, ao fim do procedimento, o sinal s na verdade se estende até a amostra $s[m + w_\mathrm{b}]$.

11.4.4 Ajuste do volume na concatenação

A função `concatena_suave` também ajusta o ganho do trecho copiado do som r, de modo a evitar mudança brusca de volume na junção. Isto é necessário principalmente quando o elemento é encolhido, pois nesse caso o miolo removido geralmente se estende por muitos períodos fundamentais, e pode haver diferença substancial de volume.

A amostra $r[i_\mathrm{a}]$ é multiplicada por γ_a, a amostra $r[i_\mathrm{b}]$ é multiplicada por γ_b, e amostras intermediárias são multiplicadas por valores intermediários entre γ_a e γ_b, em progressão geométrica.

11.5 Resultados

A figura 11.9 ilustra o resultado do programa `karacat`.

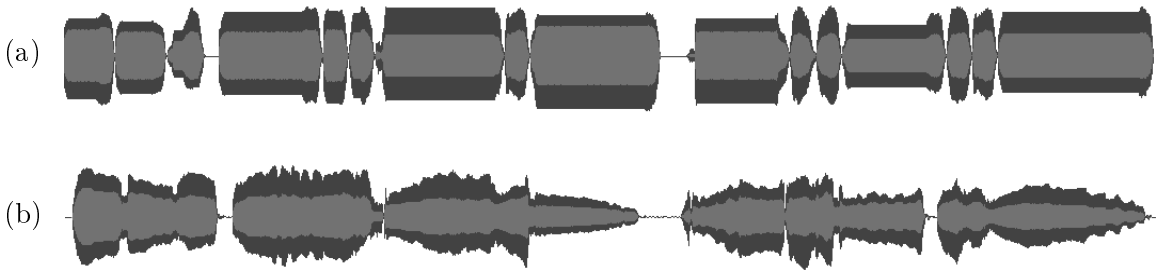


Figura 11.9: Exemplo de saída do programa `karacat`. (a) Gráfico dos primeiros compassos da canção *Bashōfu*, sintetizada pelo programa a partir do arquivo de canção da figura 11.6. (b) Gráfico do mesmo trecho, cantado pela aluna.

11.6 Conclusões e trabalhos futuros

No seu estágio atual, o programa karacat já demonstra a viabilidade do método. A decisão de usar notas pré-gravadas (em vez de algum método de ajuste da freqüência fundamental por software) revelou-se acertada, pois, apesar de exigir mais trabalho na criação do dicionário, preserva a qualidade da voz do cantor em toda a extensão da escala.

Há vários detalhes do programa que poderiam ser melhorados com relativa facilidade. Para introduzir mais "emoção" no canto sintetizado, seria interessante acrescentar mais um parâmetro a cada nota da partitura, indicando seu volume relativo. Outra melhoria simples seria permitir a duplicação da consoante da sílaba, como em *satte*, que indica uma pequena pausa no início da sílaba.

Outra melhoria desejável, mas muito mais trabalhosa de implementar, seria a introdução (ou preservação) do *vibrato* na canção sintetizada.

A preparação do dicionário também poderia ser facilitada pelo desenvolvimento de ferramentas específicas. A necessidade mais óbvia é uma rotina para segmentação automática dos

arquivos ⟨*sílaba*⟩.`au` em notas individuais e para a identificação automática do trecho estável (a vogal principal da sílaba). Esta rotina permitiria dispensar os arquivos ⟨*sílaba*⟩.`pic`, cuja criação atualmente toma mais tempo que a gravação do arquivo ⟨*sílaba*⟩.`au`. Outra necessidade é uma ferramenta que verificasse se a freqüência fundamental de cada unidade do dicionário é a da nota correspondente.

Apêndice A

Produtos de síntese de fala

Este apêndice relaciona alguns produtos para síntese de fala, incluindo conversão texto-fala, disponíveis atualmente. Uma vez que seria impossível apresentar uma relação completa, relacionamos apenas os produtos mais conhecidos. Listas mais completas podem ser encontradas na tese de S. Lemmetty [34] e no site do projeto DISC [50]. Infelizmente, ambas são relativamente antigas (1999).

Sistemas de conversão texto-fala geralmente incluem as duas partes (alto nível e baixo nível). Porém, em muitos casos estas partes são desenvolvidas ou disponibilizadas separadamente. Alguns fornecedores se dedicam apenas ao sistema de baixo-nível, outros apenas ao de alto-nível.

A.0.1 Produtos Livres

MBROLA: é um sistema de síntese de baixo nível, criado em 1995 pela Faculdade Politécnica de Mons, na Bélgica. O objetivo principal do projeto é desenvolver um sistema não comercial que atenda o maior número possível de idiomas. A tecnologia é baseada na concatenação de difones, e disponível em 34 idiomas, inclusive em português do Brasil [62].

Cybertalk: desenvolvido pela Panasonic, usa um modelo híbrido: um sintetizador baseado em formantes para vogais e consoantes sonoros, e segmentos de ruídos pré-gravados para sons plosivos e fricativos. Números e algumas seqüências alfanuméricas são produzidos separadamente pelo método de síntese concatenativa. Pode ser utilizado em ambiente Windows, disponível no idioma inglês [50].

Festival: desenvolvido por A. Black e P. Taylor do Centro de Pesquisas em Tecnologia da Fala (CSTR) da Universidade de Edimburgo, em colaboração com o *Generic Speech Synthesis System* (CHATR), dos Laboratórios ATR de Pesquisa em Interpretação de Telecomunicações, Japão. Utiliza a tecnologia concatenativa de difones. Disponível para vários idiomas [12].

Flite (Festival-lite): desenvolvido na Universidade Carnegie-Mellon, é uma versão alternativa mais rápida e menor do Festival para ambiente Linux e Windows [9].

Epos: foi projetado inicialmente como ferramenta para pesquisa. A tecnologia empregada é síntese fonte-filtro do tipo LPC, com informações prosódicas geradas por redes neurais e regras. Disponível nos idiomas checo e eslovaco [28].

Gnuspeech: é um pacote TTS expansível baseado no modelo articulatório por regras. Funciona em tempo real, em ambientes Linux [27].

Free TTS: conversor texto-fala escrito inteiramente em Java, baseado no sistema *Flite* [40].

HMM-Based Speech Synthesis System (HTS): sistema baseado no modelo HMM, desenvolvido pelo Departamento de Ciência da Computação do Instituto de Tecnologia de Nagoya. É uma versão modificada do *Hidden Markov Model Tool Kit* (HTK) combinado com o *Speech Signal Processing Tool Kit* (SPTK). Disponível nos idiomas japonês e inglês, para ambientes Windows e Linux [65].

Klatt-style System: disponível para ambiente Unix [50].

A.0.2 Produtos Comerciais

Natural Voices: desenvolvido pelos Laboratórios Bell da AT&T (hoje Lucent Technologies). Baseado no modelo concatenativo de polifones. Disponível em vários idiomas, incluindo inglês, espanhol, italiano, alemão, russo, romeno, chinês e japonês. Disponível para ambientes Windows, Linux e Solaris.

Elan Sayso: Sistema de síntese de voz de alta qualidade, desenvolvido em 2002 pela Elan Speech.

DecTalk: desenvolvido pela Digital Equipment Corporation (DEC). Este sistema é originado de MITalk e Klattalk. Disponível nos idiomas: inglês americano, espanhol e alemão. O DecTalk possui, provavelmente, o melhor pré-processador e controles de pronúncia. A sua tecnologia é baseada na síntese de formantes.

Aculab Prosody TTS: desenvolvido pela Aculab, é um sistema baseado na tecnologia concatenativa para ambientes Windows 2000, Linux e Solaris. Disponível em vários idiomas, incluindo português do Brasil.

Tem parceria com a Loquendo, cuja matriz está em Turin, Itália e está disponível em dezessete idiomas, inclusive português do Brasil. A Loquendo desenvolve ASR também.

Laureate: desenvolvido pelo Laboratório da British Telecom (BT), com estrutura modular.

CNET PSOLA: desenvolvido na década de 1980, pelo Centro Nacional de Estudos em Telecomunicações (CNET) da France Telecom. É um sistema de baixo nível baseado na concatenação de difones. Os produtos comerciais são distribuídos pela Élan Informatique como sistema *TTS ProVerbe*.

Real Speak: desenvolvido pela Nuance Communications para os idiomas francês, inglês, alemão, grego, espanhol, e outros, inclusive português do Brasil, perfazendo um total de vinte idiomas. Ambiente Windows, Solaris e Linux.

VoiceTex: desenvolvido pela NeoSpeech de Fremont, Califórnia, é um sistema baseado na tecnologia concatenativa em ambiente Windows, Unix, Linux. A sua origem é a Voiceware Co. Ltd., da Coréia.

FlexVoice: desenvolvido pela MindMaker, utiliza os modelos concatenativo e híbrido, para ambientes Linux e Windows. Pode ser usado também na área da música para efeitos especiais de voz.

SoftVoice: sistema extensível para animação, desenvolvido pela SoftVoice. Usado em vários computadores populares da década de 1980, como Commodore C64 (*Software Automatic Mouth*, SAM) Amiga (*Narrator*), Apple Macintosh (*Mactalk*) e Atari. Baseado no modelo dos formantes. Disponível para ambiente Windows, em inglês e espanhol, com vários tipos de vozes, tais como: diferentes timbres masculinos e femininos, criança, robô, alienígenos, etc.

ORATOR: desenvolvido pela Bell Communication Research (Bellcore) e comercializado pela Telcordia Technologies, baseado na tecnologia concatenativa de semi-sílabas. Para ambientes Windows, Solaris, AIX, OSF. Disponível também o reconhecedor de fala.

FAAST: sistema TTS da Fonix, com vocabulário ilimitado, tipo concatenativo em ambientes Windows, Linux e Solaris

Fonix DecTalk: sistema baseado em formantes, para ambientes Windows, Solaris e Linux.

Lernout&Hauspie: possuem vários sistemas TTS, baseados no modelo concatenativo de polifones. Disponíveis nos idiomas: inglês, alemão, holandês, espanhol, italiano, coreano,

japonês, árabe e chinês, para ambientes Windows.

HADIFIX (HAlbsilben, DIphone, sufFIXe): Um sistema desenvolvido pela Universidade de Born, da Alemanha, é baseado na concatenação de semi-sílabas, difones e sufixos, de acordo com Portele et al (1992). Possui suporte para voz cantada.

SPRUCE (Speech Response from UnConstrained English): é um sistema de **alto-nível** desenvolvido pelas Universidades Bristol e Essex [35]. Foi projetado, teoricamente, para ser acoplado a qualquer sistema de **baixo-nível**. Serve para ambientes: SUN Solaris, Unix e PC (MS-DOS) compatíveis.

WHISTLER: (Whisper Highly Intelligent Stochastic Talker): desenvolvido pela Microsoft, é baseado no método concatenativo, utiliza HMM para o procedimento de treinamento para o sistema de reconhecimento de fala (esclarecido no próximo capítulo). Este software utiliza o reconhecimento de fala para rotular os segmentos de fala, e o módulo para a análise de texto é derivado do sistema TTS da Lernout&Houspie.

ViaVoice: sistema de reconhecimento de fala desenvolvido pela IBM, disponível para vários idiomas, incluindo português do Brasil, principalmente para sistemas Windows.

rVoice: desenvolvido pela Universidade de Edimburgo e comercializado pela Rhetorical. É baseado no modelo concatenativo e para ambientes Windows, Linux e Solaris. Disponível em vários idiomas, incluindo inglês, alemão e grego.

Bestspeech: sistema desenvolvido pela Berkeley Speech Technologies (BST), com versões disponíveis para vários idiomas. Roda nos ambientes: SGI, IBM, Macintosh e SUN Solaris.

Vocaloid: desenvolvido pela Yamaha é dedicado à música. Recebe como entrada a letra da música com a partitura tem como saída a música cantada artificialmente. Este sistema

roda no ambiente Windows.

Acapela: uma gama de sistemas de síntese de voz, incluindo Telecom (para serviços de telefonia), Multimedia (para aplicações de multimeios e negócios), Onboard (para a indústria automobilística e outras aplicações "embarcadas"), e Mobility (para dispositivos portáteis) [26].

Apêndice B

Produtos de reconhecimento de fala

Relacionamos neste apêndice alguns softwares disponíveis para reconhecimento de fala, gratuitos e comerciais. Listas mais completas podem ser encontradas na tese de S. Lemmetty [34] e no *site* do projeto DISC [50]. Infelizmente, ambas são relativamente antigas (1999).

B.0.3 Produtos Livres

XVoice: é um reconhecedor de fala contínua, tipo ditado, que necessita para seu funcionamento da instalação da versão livre do *ViaVoice*.

cVoice Control/kVoice Control: é um reconhecedor básico que permite ao usuário executar comandos Linux utilizando comandos de voz.

gVoice: é um *software* que utiliza *ViaVoice* para controlar aplicativos na plataforma Linux, com interface Gtk/Gnome. Este inclui arquivos para inicialização, processador de reconhecimento, manipulação de vocabulário e controle de painel.

Kit ISIP: um sistema adaptável ao locutor, desenvolvido pelo Institute for Signal and Information Processing (ISIP) da Universidade do Mississippi. O *kit* inclui a entrada, o decodificador, e o módulo de treinamento.

Sphinx: um sistema desenvolvido inicialmente na Universidade Carnegie Mellon. Atualmente, o *Sphinx* não é um produto final, portanto para usá-lo é necessário um bom conhecedor da área. Este produto inclui treinos, reconhecedores, modelos acústicos, e modelos de linguagem e roda em ambientes GNU/Linux [25].

NICO ANN toolkit: trata-se de uma rede neural otimizada para aplicações em reconhecimento de fala.

Myers' Hidden Markov Model Software: um reconhecedor desenvolvido por Richard Myers, baseado no modelo HMM e implementado em C++. É uma ilustração de um exemplo e uma ferramenta de aprendizagem para HMM descrito no livro de Rabiner e Juang [51].

Hidden Markov Tool Kit (HTK): um reconhecedor desenvolvido em 1989 por Steve Young do grupo de Fala, Visão e Robótica do Departamento de Engenharia da Universidade de Cambridge (CUED). Disponível no site da CUED [75] desde o ano 2000.

B.0.4 Produtos Comercializados

ViaVoice: é um sistema de ditado da IBM, adaptável ao locutor, de vocabulário muito grande. Tem um ótimo desempenho, mas faz exigências consideráveis sobre o sistema, comparado com a maioria dos reconhecedores básicos. O pacote inclui: treinador, sistema de ditado e comando e controle *ViaVoice*. Este produto domina o mercado dos Estados Unidos nas aplicações de ditado, comando e controle.

Vocalis Speechware: é um reconhecedor, independente do locutor, usando o método híbrido HMM e ANN.

SpeechWorks: é um interpretador de código de fonte aberto para para VoiceXML, que roda em plataformas da família Unix. O *SpeechWorks* domina o mercado de reconhecimento de fala para servidores de telefonia e aplicações PC.

Dragon Naturally Speaking: pertence a Nuance Communications (antiga Scansoft), e roda nas plataformas Unix/Linux e Windows [15]. O sistema de ditado, teoricamente, foi construído para 99% de precisão; mas este nível de desempenho dificilmente é atingido, pois pessoas geralmente usam vocabulários maiores do que 300.000 palavras, muito maior que o vocabulário do *Naturally Speaking*.

SpeechMagic: é um reconhecedor de voz e ditado da Philips. Ele utiliza o algoritmo *Intelligent Speech Interpretation* (ISC), que consegue interpretar o que o usuário pretende dizer, e não se limita apenas em reconhecer as palavras isoladas. Ele é utilizado na área de saúde e jurídica para preenchimento de receitas e relatórios médicos, documentos jurídicos, etc. O *SpeechMagic* suporta atualmente 23 idiomas e pode ser instalado no servidor *Citrix*, viabilizando ao usuário utilizá-lo via telefone, gravador digital, celular ou alguma outra forma de captar a voz, por exemplo um microfone de mesa.

Referências Bibliográficas

[1] J. Aitchinson and A. Gilchrist. *Thesaurus Construction.* Aslib, London, 1987. Citado por [30].

[2] E. C. Albano and A. A. Moreira. Archisegment-based letter-to-phone conversion for concatenative speech synthesis in Portuguese. In *Proceedings of International Conference on Spoken Language Processing - ICSLP'96*, volume 3, pages 1708–1711, 1996.

[3] E. C. Albano, A. A. Moreira, A. H. P. Silva, P. A. Aquino, and R. K. Kakinohana. Um conversor ortográfico-fônico e uma notação prosódica mínima para síntese de fala em língua portuguesa. In E. M. Scarpa, editor, *Estudos de Prosódia no Brasil*, pages 85–105. Editora da UNICAMP, 1 edition, 1999.

[4] Jonathan Allen, M. Sharon Hunnicutt, and Dennis Klatt. *From Text to Speech: The MITalk System.* Cambridge University Press, 1987.

[5] Plínio A. Barbosa, Fábio Violaro, Eleonora C. Albano, Flávio Simões, Patrícia Aquino, Sandra Madureira, and Edson Françozo. Aiuruetê: A high-quality concatenative text-to-speech system for Brazilian Portuguese with demisyllabic analysis-based units and a hierarchical model of rhythm production. In *Proceedings of Eurospeech 99*, volume 5, pages 2059–2062, September 1999.

[6] Carl Barks. Gyro Gearloose: Glad! Sad! Mad! *Uncle Scrooge*, (22), July-August 1958. Story code W-US-22-04.

[7] Carl Barks. Prof. Pardal: O homem e a máquina. *Zé Carioca*, XVIII(799), 1968.

[8] M. J. Barros, R. S. Maia, K. Tokuda, F. G. V. Resende Jr., and D. Freitas. HMM-based European Portuguese TTS system. In *Proc. Eurospeech 2005*, pages 2581–2584, September 2005.

[9] Alan W. Black and Kevin A. Lenzo. Flite: A small, fast run time synthesis engine. Documento eletrônico em `http://www.speech.cs.cmu.edu/flite/`; último acesso em 05/out/2006, November 2005.

[10] Ronald N. Bracewell. *The Fourier Transform and its Applications*. McGraw-Hill, 2nd edition, 1978.

[11] N. M. Brooke and S. D. Scott. Two- and three-dimensional audio-visual speech synthesis. In *International Symposium on Speech, Image Processing, and Neural Networks (ISSIPNN)*, pages 73–76, 1994.

[12] Rob Clark and Alan Black. The Festival speech synthesis system. Documento eletrônico em `http://www.cstr.ed.ac.uk/projects/festival/`; último acesso em 05/out/2006, February 2005.

[13] Arthur C. Clarke. *2001: A Space Odyssey*. Hutchinson, 1968.

[14] Arthur C. Clarke and Stanley Kubrick. 2001: A space odissey. Filme, 1968.

[15] Nuance Communications. Dragon Naturally Speaking 9.0 - Medical transcription and voice recognition software by Nuance - ScanSoft. Documento eletrônico em `http://www.dragon-medical-transcription.com/index.html`; último acesso em 12/ago/2006, 2004.

[16] Nuance Communications. A timeline & history of voice recognition software. Documento eletrônico em `http://www.dragon-medical-transcription.com/history_speech_recognition_timeline.html`; último acesso em 12/ago/2006, 2004.

[17] The Wikipedia Community. A-law algorithm. Artigo da Wikipedia, disponível em `http://en.wikipedia.org/wiki/A-law_algorithm`; último acesso em 06/out/2006.

[18] The Wikipedia Community. Mu-law algorithm. Artigo da Wikipedia, disponível em `http://en.wikipedia.org/wiki/Mu-law_algorithm`; último acesso em 06/out/2006.

[19] Simon Crab. 120 years of electronic music: Homer Dudley's speech synthesisers, "The Vocoder" (1940) & "voder" (1939). Documento eletrônico em `http://www.obsolete.com/120_years/machines/vocoder/`; último acesso em 12/ago/2006, 2004.

[20] David Crystal. *The Cambridge Encyclopedia of Language*. Cambridge University Press, 2nd edition, 1997.

[21] E. David and O. Selfridge. Eyes and ears for computers. *Proceedings of the IRE*, pages 1093–1101, May 1962.

[22] Inc Education Development Center. Understanding the use of continuous speech recognition software for writing. Documento eletrônico em `http://www.edc.org/spk2wrt/Resources/ucsr.html`; último acesso em 12/ago/2006, 1999.

[23] Francisco Egashira. Síntese de voz a partir de texto para a língua portuguesa. Master's thesis, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, June 1992.

[24] J. L. Flanagan. *Speech Analysis Synthesis and Perception*. Springer, 2nd edition, 1972.

[25] E. Gouvêa. The CMU Sphinx Group open source speech recognition engines. Documento eletrônico em `http://cmusphinx.sourceforge.net/html/cmusphinx.php`; último acesso em 05/out/2006, October 2006.

[26] Acapela Group. Speech technologies. Documento eletrônico em `http://www.acapela-group.com/products/products.asp`; último acesso em 12/ago/2006, 2006.

[27] David Hill. Gnuspeech. Documento eletrônico em `http://www.gnu.org/software/gnuspeech/`; último acesso em 05/out/2006, March 2005.

[28] Petr Horak and Jirka Hanika. The Epos speech synthesis system - A open text-to-speech synthesis platform. Documento eletrônico em `http://epos.ure.cas.cz/`; último acesso em 05/out/2006, August 2005.

[29] Frederick Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1998.

[30] Elizabeth Johnston. Investigating minds - Lecture 12: The mental lexicon. Documento eletrônico em `http://pages.slc.edu/~ebj/IM_97/Lecture12/L12.html`; último acesso em 05/nov/2006, 1997.

[31] Eleanor Harz Jorden and Hamako Ito Chaplin. *Beginning Japanese*. Yale University Press, 1963. (2 volumes).

[32] Dennis H. Klatt. Review of text-to-speech conversion for English. *Journal of the Acoustical Society of America*, 82(3):737–793, September 1987. Versão eletrônica (HTML) disponível em `http://www.mindspring.com/~ssshp/ssshp_cd/dk_737a.htm`; último acesso em 04/out/2006.

[33] R. Kurzweil. The Kurzweil reading machine: A technical overview. In *Science, Technology, and the Handicapped*, volume 76-R-11, pages 3–11. American Association for the Advancement of Science, 1976.

[34] Sami Lemmetty. Review of speech synthesis technology. Master's thesis, Laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology, March 1999.

[35] Eric Lewis and Mark Tatham. SPRUCE - High specification text-to-speech synthesis. Documento eletrônico em `http://www.cs.bris.ac.uk/~eric/research/spruce97.html`; último acesso em 12/ago/2006, March 1997.

[36] Cepstral LLC. Text-to-speech - applications. Documento eletrônico em `http://www.cepstral.com/applications/`; último acesso em 12/ago/2006, 2006.

[37] R. S. Maia, H. Zen, K. Tokuda, T. Kitamura, and F. G. V. Resende Jr. Towards the development of Brazilian Portuguese text-to-speech system based on HMM. In *Proc. Eurospeech 2003*, pages 2465–2468, September 2003.

[38] José Antônio Martins. *Avaliação de Diferentes Técnicas para Reconhecimento de Fala.* PhD thesis, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, 1997.

[39] Takashi Masuko, Keiichi Tokuda, and Takao Kobayashi. A very low bit rate speech coder using HMM with speaker adaptation. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 609–612, May 1998.

[40] Sun Microsystems. Freetts 1.2 - a speech synthesizer written entirely in the Java programming language. Documento eletrônico em `http://freetts.sourceforge.net/docs/index.php`; último acesso em 05/out/2006, February 2005.

[41] Keiko Miyamoto. *Novo Dicionário Romanizado Japonês-Português.* Editora Animanga, 2004.

[42] Xiaolong Mou. *Towards a Unified Framework for Sub-lexical and Supra-lexical Linguistic Modeling.* PhD thesis, Department of Electrical Engineering and Computer Sciences, Massachusetts Institute of Technology, May 2002.

[43] E. Moulines. *Algoritmes de Codage et de Modification das Paramètres Prosodiques pour la Synthèse de la Parole à Partir du Texte.* École National Supérieure des Télécommunications, 1990.

[44] IEEE Virtual Museum. Vocoders and Voders. Documento eletrônico em `http://www.ieee-virtual-museum.org/`; último acesso em 12/ago/2006, 2006.

[45] Tony Oetzmann. Audacity - The free, cross-platform sound editor. Documento eletrônico em `http://audacity.sourceforge.net/`; último acesso em 05/out/2006, October 2006.

[46] Harry F. Olson. *Music, Physics and Engineering*. Dover, 1967.

[47] A. Oppenheim, Ronald W. Schafer, and John R. Buck. *Digital Signal Processing*. Prentice Hall, 2nd. edition, 1999.

[48] J. Pierce. Whither speech recognition? *Journal of the Acoustical Society of America*, 46:1049–1051, 1969.

[49] Rich Polivka. The Texas Instruments Speak & Spell. Documento eletrônico em `http://www.99er.net/spkspell.html`; último acesso em 12/ago/2006, January 2005.

[50] DISC Project. A survey of exiting methods and tools for developing and evaluation of speech synthesis and of commercial speech synthesis systems. Documento eletrônico em `http://www.disc2.dk/tools/SGsurvey.html`; último acesso em 05/out/2006, December 1999.

[51] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.

[52] Lawrence R. Rabiner and Ronald W. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall, 1979.

[53] Sam Roweis. Speech processing background. Documento eletrônico em `http://www.cs.toronto.edu/~roweis/notes/spblet.ps.gz`; último acesso em 07/out/2006.

[54] Philip Rubin and Louis Goldstein. Articulatory synthesis. Documento eletrônico em `http://www.haskins.yale.edu/facilities/asy-demo.html`; último acesso em 12/ago/2006, August 1995.

[55] Masaharu Sakamoto and Takashi Saito. Speaker recognizability evaluation of a voicefont-based text-tospeech system. In *Proc. 7th International Conference on Spoken Language Processing (ICSLP 2002)*, pages 2529–2532, September 2002.

[56] Flávio Olmos Simões. Implementação de um sistema de convesão texto-fala para o português do Brasil. Master's thesis, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, May 1999.

[57] Eric Smalley. Quantum math models speech. Boletim eletrônico em `http://www.trnmag.com/Stories/2004/100604/Quantum_math_models_speech_100604.html`; último acesso em 12/ago/2006, October 2004.

[58] J. Q. Stewart. An electrical analogue of the vocal organs. *Nature*, 110:311–312, 1922.

[59] Frost & Sullivan. Summary of the european healthcare voice recognition systems market. Documento eletrônico acessível através `http://www.g2speech.com/`; último acesso em 12/ago/2006, November 2005.

[60] Masatsume Tamura, Takashi Masuko, Takao Kobayashi, and Keiichi Tokuda. Visual speech synthesis based on parameter generation from HMM: Speech-driven and text-and-speech-driven approaches. In *Proc. 1998 International Conference of Auditory-Visual Speech Processing*, pages 219–224, 1998.

[61] Michael Tanenblatt. Bell Labs - Text-to-speech synthesis. Documento eletrônico em `http://www.bell-labs.com/project/tts/tts-overview.html`; último acesso em 12/ago/2006, September 1999.

[62] Mbrola Team. The mbrola project. Documento eletrônico em `http://tcts.fpms.ac.be/synthesis/mbrola/mbrola_entrypage.html`; último acesso em 12/ago/2006, January 2005.

[63] TCTS Team. The history behind TCTS Lab. Documento eletrônico em `http://tcts.fpms.ac.be/history.php`; último acesso em 12/ago/2006, December 2005.

[64] Antônio Pedro Timoszczuk. *Reconhecimento Automático do Locutor com Redes Neurais Pulsadas*. PhD thesis, Escola Politécnica, Universidade de São Paulo, March 2004.

[65] Keiichi Tokuda. HMM-based speech synthesis system (HTS). Documento eletrônico em `http://hts.ics.nitech.ac.jp/`; último acesso em 05/out/2006, March 2006.

[66] Keiichi Tokuda, Takayoshi Yoshimura, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Speech parameter generation algorithms for HMM-based speech synthesis. In *Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1315–1318, June 2000.

[67] Keiichi Tokuda, Heiga Zen, and Alan W. Black. An HMM-based speech synthesis system applied to English. In *Proc. IEEE TTS Workshop 2002, Santa Monica, CA*, pages 227–230, 2002.

[68] Fábio Violaro, Plínio A. Barbosa, Eleonora C. Albano, and Edson Françozo. Um conversor texto-fala para o português brasileiro com processamento lingüístico de alta qualidade. In *Anais do VII Simpósio Brasileiro de Microondas e Optoeletrônica e do XIV Simpósio Brasileiro de Telecomunicações (TELEMO '96)*, pages 361–366, CEFET, Curitiba, PR, Brazil, July 1996.

[69] Katsunori Wakizaka, editor. *Michaelis Dicionário Prático Japonês-Português*. Editora Melhoramentos e Aliança Cultural Brasil-Japão, 2005.

[70] Comunidade Wikipedia. Neuron. Artigo da Wikipedia, disponível em `http://en.wikipedia.org/wiki/Neuron`; último acesso em 07/out/2006.

[71] Comunidade Wikipedia. Speech synthesis. Artigo da Wikipedia, disponível em `http://en.wikipedia.org/wiki/Speech_synthesis`; último acesso em 26/ago/2006.

[72] Comunidade Wikipedia. Window function. Artigo da Wikipedia, disponível em `http://en.wikipedia.org/wiki/Window_function`; último acesso em 12/ago/2006.

[73] Susan Wilson. Voice and speech synthesis recognition. Documento eletrônico em `http://www.sinc.sunysb.edu/stu/smwilson/est585/fact.html`; último acesso em 12/ago/2006, April 2000. (Trabalho de curso).

[74] I. H. Witten. *Principles of Computer Speech*. Academic Press, 1982.

[75] Phil Woodland and Gunnar Evermann. Htk speech recognition toolkit. Documento eletrônico em `http://htk.eng.cam.ac.uk/`; último acesso em 12/ago/2006, 2002.

[76] Carlos Alberto Ynoguti. *Reconhecimento de Fala Contínua Usando Modelos Ocultos de Markov*. PhD thesis, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, 1999.

[77] Carlos Alberto Ynoguti and Fábio Violaro. Desenvolvimento de um conjunto de ferramentas para pesquisas em reconhecimento de fala. *Telecomunicações*, 4(2):36–43, 2001.