MINISTÉRIO DA EDUCAÇÃO

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

ESCOLA DE ENGENHARIA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA METALÚRGICA E DOS MATERIAIS - PPGEMM

# RECONHECIMENTO AUTOMÁTICO DE VOZ PARA PALAVRAS ISOLADAS E INDEPENDENTE DO LOCUTOR

por

Joel Augusto Luft

Engenheiro Eletricista (UFRGS, 1991)

Trabalho realizado no Departamento de Engenharia Elétrica da escola de Engenharia da UFRGS, dentro do Programa de Pós-Graduação em Engenharia Metalúrgica e dos Materiais - PPGEMM.

Porto Alegre

1994

Título da dissertação:

# RECONHECIMENTO AUTOMÁTICO DE VOZ PARA PALAVRAS ISOLADAS E INDEPENDENTE DO LOCUTOR

Apresentada ao Programa de Pós-Graduação em Engenharia Metalúrgica e dos Materiais - PPGEMM, como parte dos requisitos para a obtenção do título de

Mestre em Engenharia

Área de Concentração: Instrumentação Eletro-Eletrônica

por

Autor: Joel Augusto Luft  Titulação: Engenheiro Eletricista

Ano: 1994

Esta DISSERTAÇÃO foi julgada adequada para a obtenção do título de Mestre em Engenharia, Área de Concentração em Instrumentação Eletro-Eletrônica e aprovada na sua forma final pelo Orientador e pela Banca Examinadora do Curso de Pós-Graduação.

Orientador:

Altamiro Amadeu Suzim
Doutor em Informática
Professor do Programa de Pós-Graduação em Engenharia
Metalúrgica e dos Materiais da UFRGS

Co-orientador:

Thomas Weihmann
Mestre em Engenharia
Professor do Departamento de Engenharia Elétrica da UFRGS

Banca examinadora:

Philippe Olivier Alexandre Navaux
Doutor em Informática
Professor do Curso de Pós-Graduação em Ciência da Computação da UFRGS

Marco Túllio de Vilhena
Doutor em Fenômenos de Transporte
Professor do Departamento de Engenharia Nuclear da UFRGS

Altamiro Amadeu Suzim - Doutor em Informática
Professor do Programa de Pós-Graduação em Engenharia
Metalúrgica e dos Materiais da UFRGS

Thomas Weihmann - Mestre em Engenharia
Professor do Departamento de Engenharia Elétrica da UFRGS

Coordenador do PPGEMM
Professor Telmo Strohaecker

Agradecimentos:

Ao professor e orientador Dr. Altamiro Amadeu Suzim, pelo apoio e exemplo de dedicação e empenho na busca dos objetivos.

Ao professor e co-orientador Thomas Weihmann que apoiou de forma marcante, discutindo e participando no desenvolvimento deste trabalho.

Ao colega e amigo Marcelo Negreiros que esteve sempre presente compartilhando e colaborando com o trabalho realizado.

A toda minha família, cujo apoio e compreensão foram decisivos para a realização deste trabalho.

A todos colegas e amigos que me ajudaram no decorrer deste trabalho.

Ao CNPq - Conselho Nacional de Desenvolvimento Científico e tecnológico, pela bolsa de mestrado.

# SUMÁRIO

# LISTA DE FIGURAS

# LISTA DE TABELAS

# RESUMO

Neste trabalho são apresentadas diversas técnicas aplicadas no reconhecimento de voz para palavras isoladas e independente do locutor. Estas técnicas são estudadas abordando os aspectos referentes a sua aplicabilidade prática. É apresentada a implementação de um sistema de reconhecimento de voz em tempo real.

São estudadas as características do processo de produção da voz e da capacidade auditiva do homem. São abordadas as limitações relacionadas com o reconhecimento automática da voz e apresentada a estrutura de um reconhecedor de voz para palavras isoladas.

Diversas formas de representação do sinal de voz utilizando medidas de energia, cruzamento por zero, análise espectral e análise cepstral são apresentadas e estudadas de modo a serem utilizadas no processo de reconhecimento de voz. A técnica LPC de codificação do sinal de voz é analisada com a descrição dos algoritmos de extração dos parâmetros do sinal. Também são estudadas medidas de distorção entre parâmetros do sinal de voz para a avaliar as diferenças entre eles. É apresentado o processo de quantização vetorial que reduz o volume de dados utilizado no processo de reconhecimento.

Duas técnicas de reconhecimento de voz (DTW e HMM) são estudadas e detalhados os aspectos referentes à implementação prática de tais algoritmos. Também são apresentados algoritmos de detecção automática dos limites da palavra.

Os detalhes da implementação em tempo real com os resultados de diversos experimentos práticos são mostrados.

Conclusões gerais e a avaliação dos resultados obtidos são apresentados. Também são relacionados alguns aspectos para a melhoria e desenvolvimento do sistema de reconhecimento descrito neste trabalho.

# ABSTRACT

This work presents several techniques applied in speaker-independent isolated word speech recognition. These techniques are studied regarding its practical use. The implementation of a real time speech recognition system are presented.

The speech production mechanism and the human hearing characteristics are studied. The speech recognition constraints are analyzed and the structure of an isolated-word speech recognizer is presented.

Several representations of speech signal using energy measurement, zero crossing, spectral analysis and cepstral analysis are presented and studied related to the speech recognition process. The LPC coder is analyzed and the algorithms for parameters extraction are presented. The distortion measures are studied to evaluate the differences between speech parameters. The vector quantization process, which is important in data reduction, is presented.

Two speech recognition techniques (DTW and HMM) are studied and several aspects related to the practical implementation are detailed. Endpoint detection algorithms for isolated words are also presented.

The details of the real time implementation and the results of practical experiments are presented.

General conclusions and the evaluation of the results are presented. Some aspects to improve and to develop the recognition system described in this job are reported.

# 1 INTRODUÇÃO

Um dos meios básicos de comunicação entre os seres humanos é a fala. Através da voz, diversos tipos de informações podem ser transmitidas e interpretadas facilmente pelo homem, sendo as principais: a informação linguística, a informação de quem é o locutor e a informação do estado emocional do locutor. A primeira é a mais importante pois indica a mensagem que o locutor quer transmitir.

Como o uso de máquinas pelo homem é cada vez mais comum para diversas aplicações, é desejável prover tais equipamentos com a habilidade de comunicação através da voz. A utilização da voz para este fim apresenta uma série de vantagens, começando por tornar a tarefa fácil e natural, já que não necessita habilidade ou experiência. Além disso, é um meio rápido de entrada de informações (cerca de 3 a 4 vezes mais rápido que datilografar e 8 a 10 vezes mais rápido que escrever à mão) [5], permite a execução de outras tarefas simultaneamente, deixando livres as mãos e os olhos e, ainda, permite a locomoção.

O reconhecimento automático da voz é o processo de extração automática da informação linguística do sinal de voz. A execução deste processo não é uma tarefa simples. A informação lingüística contida no sinal de voz está codificada de modo que o elevado grau de variabilidade do sinal, causada pelo ambiente e pelo locutor, praticamente não interfere na percepção da informação pelo homem.

O reconhecimento automático de voz tem sido estudado intensamente nas últimas décadas. Vários sistemas foram implementados utilizando diversas abordagens para a solução do problema. No que se refere ao reconhecimento de palavras isoladas pode-se citar alguns trabalhos. Em 1975, Itakura [9] apresentou um sistema baseado no alinhamento temporal não linear (DTW) da voz, onde obteve uma taxa de acerto de 97.3% para um vocabulário de 200 palavras utilizando um único locutor. Rabiner [30], em 1983, utilizou as técnicas de quantização vetorial e HMM para um vocabulário de 10 palavras (dígitos de 0 a 9), obtendo taxas de acerto de 96.3 % para locutores que participaram do treinamento do sistema e de 92.8 % para locutores que não participaram do treinamento. Um sistema para palavras isoladas e independente do locutor com um pré-processamento baseado na quantização vetorial da palavra seguido de uma técnica convencional DTW foi proposto por Pan [24] em 1985, obtendo 97.9 % de acertos para um vocabulário de 10 palavras (dígitos de 0 a 9) e 87.4 % de acertos para um vocabulário de 129 palavras. Em 1986, Furui [6] propôs uma técnica para o

reconhecimento de palavras independente do locutor que utiliza características dinâmicas do espectro da voz combinada ao alinhamento temporal e obteve taxa de acerto de 97.6 % para um vocabulário de 100 palavras (nomes de cidades japonesas). Tohkura [33], em 1987, apresentou um sistema independente do locutor utilizando DTW e medida de distorção cepstral ponderada conseguindo taxa de acerto de 99 % para um vocabulário de 10 palavras (dígitos de 0 a 9). Em 1990, Hermansky [8] apresentou uma nova técnica de análise da voz (Perceptual Linear Predictive) atingindo um reconhecimento superior a 97 % para 11 palavras utilizando DTW com múltiplas referências.

Para colocar os diversos aspectos pertinentes ao problema de reconhecimento automático de voz serão apresentados neste capítulo as características do sinal de voz, com os fundamentos do mecanismo de produção da voz, um modelo simples para geração de voz e características do sistema auditivo. Serão abordados os problemas e restrições impostos aos sistemas de reconhecimento de voz e ainda serão apresentadas de maneira esquemática as partes componentes de um sistema de reconhecimento automático de palavras isoladas.

## 1.1 Características da voz

O processo da fala humana começa com a concepção da idéia que o locutor quer transmitir ao ouvinte. Esta idéia é convertida no cérebro para uma estrutura linguística composta por frases e palavras, de acordo com a linguagem utilizada, que é então transmitida aos diversos órgãos do aparelho fonador através de impulsos nervosos. O aparelho fonador produz variações de pressão no ar que se propagam até o ouvinte. Esta onda acústica, que corresponde ao sinal de voz, é um processo contínuo do ponto de vista fisico. Entretanto, após captada e processada pelo cérebro do ouvinte, resulta, em termos linguísticos, em uma seqüência de segmentos discretos encadeadas no tempo. O menor destes segmentos é o fonema.

A figura 1.1.a representa uma forma de onda típica de um sinal de voz. Esta ilustração permite a observação de algumas características do sinal de voz. A forma de onda e o espectro da voz comportam-se como um processo não estacionário; entretanto, estas características permanecem quase constantes nos diversos segmentos que a compõem. Basicamente dois tipos de forma de onda podem ser observados: um com sinais randômicos e de baixa amplitude (sons não-vozeados) e outro com sinais quase periódicos e de alta amplitude (sons vozeados). O pico de amplitude dos sons vozeados é muito maior que dos sinais não-vozeados.

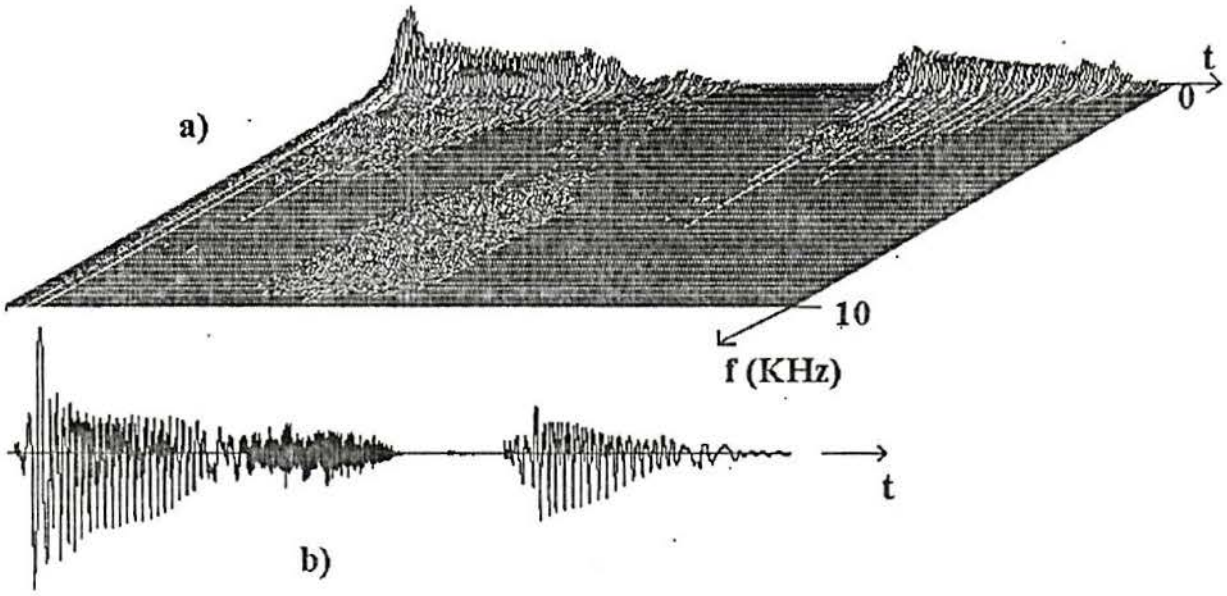


Figura 1.1 Espectrograma (a) e forma de onda (b) para a palavra /teste/.

Praticamente toda a informação contida no sinal está na faixa de freqüência até 10 kHz, estando a maior parte da energia abaixo de 5 kHz. A figura 1.1.b apresenta a variação do espectro de um sinal no tempo.

### 1.1.1 Processo de geração de voz

A voz é produzida pelo aparelho fonador, que é composto por diversos órgãos conforme esquematizado na figura 1.2. O processo de produção da fala envolve três subprocessos: geração do som, articulação e radiação.

A variação da pressão do ar nos pulmões produz um fluxo de ar através da glote, que é o orificio entre as cordas vocais. A glote está geralmente aberta durante a respiração permitindo o fluxo livre do ar. Quando as cordas vocais são tensionadas, a passagem de ar através da glote é periodicamente interrompida, modulando o ar em pulsos discretos. A forma destes pulsos pode ser aproximada por uma onda triangular assimétrica e é proporcional à variação temporal da abertura da glote. O envelope espectral desta onda tem uma declividade

que varia de 12 a 18 dB/oitava [5]. O período de oscilação das cordas vocais é chamado período fundamental ou *pitch*.

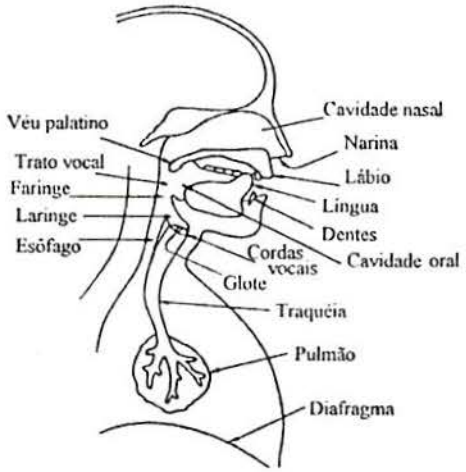


Figura 1.2 Diagrama esquemático do aparelho fonador.

A parte superior que começa na glote e termina nos lábios é chamada de trato vocal. O trato vocal é um tubo acústico não uniforme e variável, com cerca de 17 cm de comprimento e cuja área da secção transversal varia de 0 $cm^2$ até aproximadamente 20 $cm^2$ [29]. O ajuste da forma do trato vocal para produção dos diversos sons é chamado articulação. As partes do trato vocal que podem mover-se ativamente tais como a língua, os lábios, o maxilar e o véu palatino são chamadas articuladores. A cavidade nasal pode ser acoplada ao trato vocal pela ação do véu palatino que pode abrir ou fechar esta cavidade, controlando o som emitido pelas narinas. O trato vocal apresenta uma característica de transmissão extremamente dependente da posição dos articuladores, com freqüências de ressonância que são chamadas "formantes". Estas são nomeadas começando com a componente de mais baixa freqüência, a primeira formante ($F_1$), então a segunda formante ($F_2$) e assim sucessivamente. A onda acústica gerada é irradiada para fora do trato vocal pelos lábios e pelas narinas.

Os sons produzidos pelo aparelho fonador podem ser classificados em sons vozeados e não-vozeados. Os sons vozeados são produzidos quando a passagem de ar através do trato vocal se dá de forma contínua e sem turbulência. Os sons não-vozeados são produzidos quando o trato vocal impõe resistência à passagem do ar. Os sons não-vozeados ainda podem ser divididos em fricativos, plosivos e africados, de acordo com o tipo de constrição imposta ao fluxo de ar.

Nos sons vozeados, como é o caso das vogais, o trato vocal é excitado pelos pulsos de ar gerados pela vibração das cordas vocais; o som produzido é função das freqüências de ressonância do trato vocal naquele instante. Esta característica de transmissão varia lentamente com o tempo. Os sons vozeados, especialmente as vogais, geralmente apresentam três formantes [5]. É importante observar que, mesmo para um mesmo fonema, estas freqüências formantes variam em função do locutor. Além disso estas formantes dependem dos fonemas adjacentes durante a fala contínua. A ocorrência de um período de transição no meio de fonemas adjacentes, onde as características acústicas mudam continuamente, é referida como co-articulação. As vogais podem ser geralmente caracterizadas por $F_1$ e $F_2$, enquanto que a formante de mais alta ordem, $F_3$, geralmente não varia muito de uma vogal para outra. Esta formante geralmente está associada ao locutor ou mais precisamente ao comprimento do trato vocal do locutor [5].

Os sons não-vozeados estão associados a determinadas consoantes que são classificadas como fricativas, plosivas e africadas. Nas consoantes fricativas , como /s/ e /f/, ocorre a constrição do trato vocal em algum ponto de modo a produzir um fluxo de ar turbulento. As diferenças entre as consoantes fricativas são função da posição da constrição e da forma do trato. As consoantes plosivas, como /p/, /t/ e /q/, são sons impulsivos, causados pelo bloqueio completo da passagem de ar em algum ponto do trato, provocando um aumento de pressão atrás deste ponto seguido de uma rápida abertura do bloqueio. As africadas são produzidas pela sucessão de um som plosivo e um fricativo, ocorrendo a abertura lenta do bloqueio.

Estes tipos de consoante são independentes da vibração das cordas vocais. As características acústicas das consoantes são extremamente variáveis e sensíveis ao efeito de co-articulação com as vogais, pois as consoantes não apresentam um período de estabilidade como os sons vozeados. Este efeito é acentuado principalmente com o aumento da velocidade na fala, pois a articulação do próximo fonema começa antes de completada a pronúncia do fonema corrente.

Tanto os sons vozeados como os não-vozeados sofrem a influência das características ressonantes do sistema vocal. Como vários órgãos participam da produção da fala simultaneamente, e cada orgão apresenta sua própria constante de tempo, o fenômeno acústico resultante destes movimentos é extremamente complexo, tornando bastante difícil obter uma correspondência unívoca entre um determinado fonema e as características acústicas. O efeito de co-articulação pode até afetar fonemas além dos adjacentes.

### 1.1.2 Um modelo de geração de voz

O modelo aqui apresentado considera que as características do trato vocal não interferem na fonte de geração do som. A partir desta colocação podemos representá-los de maneira independente, separando a fonte g(t) da articulação h(t) e obtendo o som irradiado como sendo a convolução da excitação com a resposta ao impulso do trato vocal (figura 1.3). A influência do trato vocal é realmente pequena na freqüência de oscilação das cordas vocais mas apresenta considerável influência na forma de onda do pulso gerado na glote. Isto pode, entretanto, ser desprezado neste modelo incluindo este efeito nas características do trato [5].

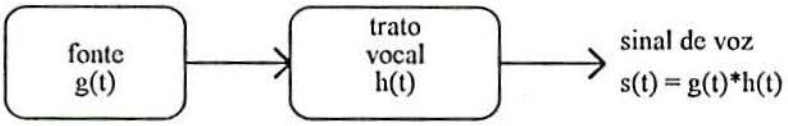


Figura 1.3 Modelo simples de geração de voz.

A fonte pode ser modelada de duas maneira: uma para sons vozeados e outra para sons não-vozeados. Para os sons vozeados consideramos a fonte como um gerador de pulsos cuja freqüência é a mesma da oscilação das cordas vocais. A amplitude destes pulsos controlam a amplitude do sinal de voz. Para os sons não-vozeados a fonte de som pode ser modelada por um gerador de ruído branco, cuja energia média corresponde à intensidade do sinal de saída. O trato vocal pode ser modelado por um filtro variável no tempo. Já que as características do trato (formantes) variam com o tempo, este filtro deve acompanhar estas variações continuamente.

Uma representação digital deste modelo é apresentada na figura 1.4. As fontes de excitação são um gerador de impulsos e um gerador de ruído branco. O gerador de impulsos é usado para produzir os sons vozeados e o intervalo entre os pulsos corresponde ao período fundamental (*pitch*). O gerador de ruído branco simula tanto o fluxo de ar turbulento como a onda criada pela liberação do ar contido por um bloqueio do trato. O espectro deste ruído deve ser plano. Estas fontes são aplicadas à entrada de um filtro digital cujos parâmetros variam no tempo de acordo com as características de transmissão do trato vocal. Estas características variam lentamente com o tempo, mas podem ser consideradas constantes durante intervalos de tempo da ordem de 10 ms [31]. O nível acústico do sinal de saída é ajustado por um controle de ganho colocado entre a fonte e a entrada do filtro.

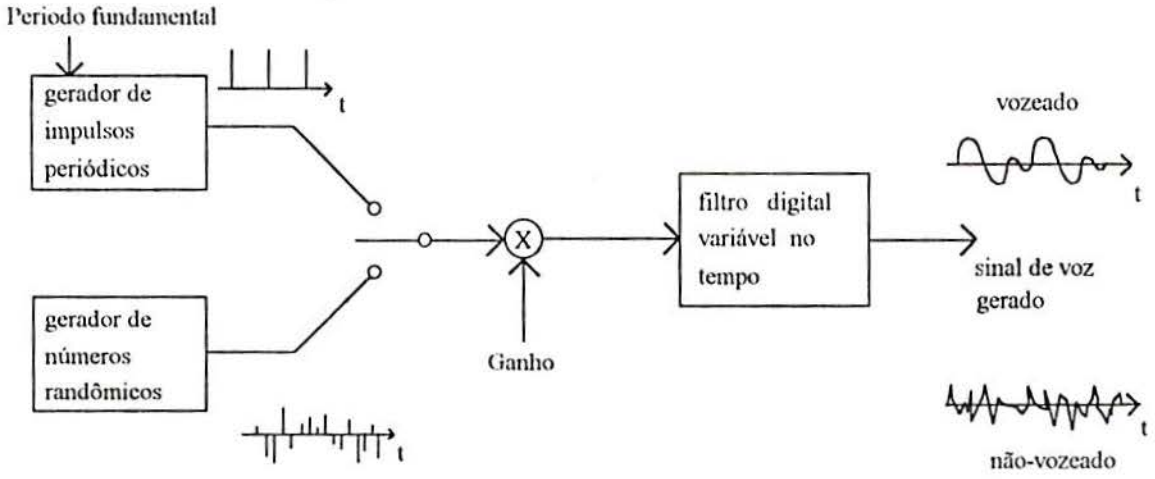


Figura 1.4 Modelo digital para geração de voz.

### 1.1.3 Características do sistema auditivo

Podemos dividir o mecanismo fisiológico da audição em três partes: ouvido externo, ouvido médio e ouvido interno. O ouvido externo é responsável pela captação do som, funcionando como uma corneta acústica para as freqüência entre 200 Hz e 5500 Hz. O ouvido médio transmite e amplifica mecanicamente as pressões sonoras captadas pelo ouvido externo até o interno. Este último, também chamado labirinto, tem uma estrutura bastante complexa e transforma as vibrações mecânicas em impulsos nervosos que vão para o cérebro. No ouvido interno existem cerca de 40 mil fibras nervosas, cada uma sintonizada em uma faixa de freqüência.

O ouvido humano normal pode perceber freqüências sonoras situadas entre 16 Hz e 17 kHz. Em acústica, considera-se como audível a faixa de 20 Hz a 20 kHz. Esta faixa está subdividida em freqüências baixas, médias (100 Hz a 3000 Hz) e agudas. Quanto ao nível de pressão sonora que o ouvido é capaz de perceber, os valores variam da ordem de $2 \times 10^{-4}$ microbares (limiar da percepção auditiva), até 200 microbares (acima deste valor pode causar danos ao sistema auditivo).

Em média é possível diferenciar cerca de 1400 freqüências diferentes, sendo estas escalonadas de uma maneira aproximadamente logarítmica na faixa audível. Com relação à freqüência, outra característica interessante é que se o ouvido perceber só as harmônicas de uma freqüência ele é capaz de, por si só, reconstruir a fundamental. Além disso, quando duas

freqüências são ouvidas simultaneamente, também são percebidas a soma e a diferença destas freqüências.

Para a avaliação da sensibilidade do ouvido deve-se introduzir o conceito de sonoridade, que corresponde à sensação subjetiva da intensidade do som. Pode-se dizer que a sonoridade é proporcional ao logaritmo da intensidade. A sonoridade não varia apenas em função da intensidade, varia também com outros fatores como freqüência, largura de banda, duração, etc. A sensibilidade é máxima na região entre 3000 Hz e 4000 Hz. Para se obter a mesma sonoridade em uma freqüência acima ou abaixo desta região é necessário um nível maior de intensidade. A mínima variação de intensidade necessária para que se perceba alguma diferença de sonoridade varia tanto com a freqüência como com o nível de pressão. Por exemplo, nas freqüência médias, a mínima variação perceptível costuma ser de 3 dB (este valor também varia em função da intensidade absoluta), nas freqüências baixas, como em 35 Hz, este valor é de 9 dB. O ouvido é capaz de diferenciar cerca de 280 níveis diferentes de intensidade (do limiar da audição até o limite da dor).

Um efeito interessante é o chamado mascaramento parcial: a sonoridade de sons próximos em freqüência não podem ser simplesmente somadas, eles se influenciam de tal modo que o mascaramento pode ser total. Por exemplo, um som com determinada sonoridade pode ser totalmente ocultado por outro se estiver próximo em freqüência. Entretanto, um som com a mesma sonoridade, porém com razoável separação no espectro, pode ser perfeitamente percebido.

Maiores detalhes da física e dos fatores subjetivos da audição são discutidos por Fanzeres [3],[4]. Várias das características apontadas acima são utilizadas em sistemas práticos de processamento de som e, mais especificamente, de voz. A redução da taxa de transmissão do sinal de voz pode ser conseguida através de técnicas de codificação que exploram a relação logarítmica entre a intensidade e a sonoridade [35]. O efeito de mascaramento é utilizado para extrair a informação desnecessária no som (não percebida pelo sistema auditivo humano) permitindo a codificação mais eficiente do sinal. A propriedade da escala de freqüência não linear (aproximadamente logarítmica) é empregada na representação do sinal de voz e usada no processo de reconhecimento conforme será visto mais adiante.

## 1.2 Restrições de um sistema automático de reconhecimento de voz

A concepção e implementação de um sistema que permita a livre conversação entre o homem e a máquina são a meta final da pesquisa em reconhecimento automático de voz.

Apesar dos estudos e das diversas técnicas e estratégias propostas e implementadas para alcançar esta meta, os sistemas práticos só conseguem atingir um bom desempenho quando são impostas certas restrições como, por exemplo, dependência ao locutor, palavras isoladas, vocabulário pequeno e gramática restrita. As principais dificuldades relacionadas ao reconhecimento de voz podem ser resumidas nos seguintes aspectos:

- o espectro de um fonema varia em função da palavra ou frase em que ele esta inserido. Um mesmo fonema pronunciado em palavras diferentes pode apresentar espectros distintos devido à articulação dos órgãos do aparelho fonador;

- dificuldades na segmentação da fala. Falta uma unidade consistente para a fala que seja insensível ao contexto. A menor unidade da fala é o fonema. Como o espectro do sinal de voz varia continuamente entre um fonema e outro devido à iteração mútua entre eles, é muito difícil determinar com precisão os limites de um fonema;

- variações nas características da fala. As propriedades acústicas variam inclusive quando uma mesma palavra é pronunciada duas vezes pela mesma pessoa. Existem diferenças não lineares no tempo (ritmo ), em freqüência (timbre) e em amplitude (intensidade);

- insuficiente uso do conhecimento linguístico. Um ouvinte geralmente é capaz de reconhecer uma palavra mesmo se esta contém uma informação fonética incompleta ou imprecisa.

Apesar das dificuldades encontradas, é possível a implementação prática de sistemas de reconhecimento automático de voz impondo certas limitações ao sistema. Tais restrições irão influenciar características como precisão, tipo de aplicação, custo, etc. Atualmente, ainda não é possível a existência de um sistema que funcione sem restrições. A seguir, tais restrições serão abordadas mais detalhadamente.

### 1.2.1 Dependência ao locutor

Um sistema de reconhecimento de voz pode ser classificado como dependente ou independente do locutor. O primeiro é capaz de reconhecer a fala de um único locutor para o qual foi treinado, ou seja, os padrões de referência usados devem ser modificados toda vez que mudar o locutor. O segundo é capaz de reconhecer a fala de qualquer novo locutor sem a necessidade de readaptar o sistema para o mesmo.

A independência ao locutor é o problema mais difícil de ser superado. Isto ocorre porque as representações da voz usadas são sensíveis às características de um locutor

específico. Dessa forma, o conjunto de padrões de referência adequado para um locutor não é adequado para outro.

Um dos meios pesquisados para superar esta limitação é a busca de parâmetros que possam ser extraídos do sinal de voz e que sejam relativamente invariantes entre locutores. A razão pela qual se justifica tal pesquisa é que experimentos com leitura de espectrogramas resultam em alto grau de precisão [37], o que leva a crer que parâmetros invariantes possam ser encontrados. Neste caso, o reconhecimento independente do locutor resultaria na mesma tarefa que o reconhecimento de um único locutor. Outra estratégia empregada é a utilização de múltiplas referências para representar as variações entre diferentes locutores. Nesta aproximação, referências são coletadas de diversos locutores. Depois, tais referências são divididas em vários grupos e geralmente é gerado um protótipo que representa cada grupo. Adaptar o reconhecedor para um novo locutor a partir de parâmetros já existentes é outra tentativa de tornar o sistema insensível ao locutor, mas um sistema deste tipo não é, na verdade, independente do locutor.

Devido às dificuldades encontradas e para que um sistema atinja uma precisão razoável, principalmente em sistemas de mais alta complexidade, acaba sendo necessário que o locutor treine o sistema. Esta fase de treinamento implica em uma série de problemas que certas aplicações não podem suportar. Entre estes problemas podem ser citados:

- a fase de treinamento é desagradável para o usuário;

- é necessária uma grande quantidade de processamento antes que o sistema possa ser usado: certas aplicações podem envolver muitos usuários, o que resultaria em uma considerável capacidade de armazenamento do sistema para os parâmetros dos diversos locutores;

- a voz de um mesmo locutor pode variar com o tempo devido ao cansaço, doença, posição, tipo do microfone, etc..

O problema da dependência ao locutor ainda precisa ser mais pesquisado devido à sua importância, possibilitando a ampliação dos possíveis usos para os sistemas de reconhecimento de voz. Como uma aproximação pode-se dizer que, com os métodos usualmente empregados, a taxa de erro de um sistema que opere independente do locutor é três a cinco vezes maior que um dependente do locutor [12].

### 1.2.2 Tipo de fala

Os sistemas de reconhecimento podem ser classificados em: reconhecimento de palavras isoladas e reconhecimento de fala contínua. No primeiro é exigido que exista um período mínimo de silêncio entre as palavras pronunciadas e no segundo esta restrição não é colocada.

O reconhecimento de fala contínua é significativamente mais difícil que o de palavras isoladas devido às características da fala contínua. Os limites das palavras, isto é, o início e fim das palavras, não são claros e, portanto, difíceis de serem encontrados pois as palavras se sobrepõem. Os efeitos de co-articulação são muito fortes na fala contínua. Estes efeitos podem se estender de uma palavra para outra tornando-os muito difíceis de serem previstos. Além disto, certas palavras são acentuadas em relação a outras que acabam sendo mascaradas e mal pronunciadas, tornando ainda mais difícil a segmentação das frases. Diversos fonemas acabam sendo distorcidos, encurtados e até deixam de ser pronunciados.

Duas grandes vantagens podem ser atribuídas ao reconhecimento de fala contínua em relação ao de palavras isoladas. O número de palavras pronunciadas continuamente é maior, permitindo um ganho significativo na velocidade de transmissão da informação. Outra grande vantagem é que a fala contínua é um meio natural de comunicação do homem. Forçar pausas entre as palavras torna a comunicação artificial e interrompe a seqüência do pensamento.

Como desvantagens, a taxa de erro aumenta substancialmente devido à complexidade do processo, e conseqüentemente, o tempo de processamento também aumenta.

### 1.2.3 Tamanho do vocabulário

O tamanho do vocabulário das palavras a serem reconhecidas também influencia na precisão do sistema de reconhecimento. Um vocabulário grande é mais propenso a conter palavras ambíguas. Palavras ambíguas são as que parecem semelhantes para o algoritmo de classificação. Este problema também pode ser encontrado em um vocabulário pequeno composto de palavras deste tipo.

Em um vocabulário pequeno, cada palavra pode ser modelada individualmente, não implicando em um custo exagerado de treinamento e armazenagem de informações. À medida que o vocabulário aumenta, o volume de treinamento e a quantidade de parâmetros a serem armazenados se tornam inviáveis se for utilizado um modelo individual para cada palavra.

Neste caso, são modeladas unidades menores que a palavra. Este procedimento acarreta um aumento na taxa de erro pois os efeitos de articulação não são tão bem modelados quanto no modelo da palavra completa.

O tempo de processamento necessário para pesquisar a base de dados referente a um grande vocabulário pode tornar o sistema inviável, sendo necessário utilizar técnicas de seleção que ignoram certos caminhos de pesquisa. Estas técnicas podem introduzir erros, já que desprezam referências que poderiam ser corretas.

### 1.2.4 Dificuldade da gramática

A gramática em um sistema de reconhecimento irá definir as seqüências de palavras permitidas. A dificuldade de uma gramática é medida pela perplexidade [12], que é uma medida da quantidade de restrições impostas pela gramática, ou melhor, o nível de incerteza em cada ponto de decisão. Sistemas com baixa perplexidade são potencialmente mais precisos que sistemas onde o usuário tem mais liberdade, porque tais sistemas limitam o vocabulário apenas às palavras que podem aparecer no contexto corrente.

O aumento da perplexidade geralmente está associado a uma substancial perda de precisão. A capacidade de aceitar gramáticas com alta perplexidade é uma importante meta a ser alcançada, pois somente tais sistemas podem ser versáteis o suficiente para aceitar tarefas complexas como, por exemplo, aceitar um texto ditado.

## 1.3 Reconhecedor de voz para palavras isoladas

A figura 1.5 apresenta um modelo genérico de reconhecimento de padrões usado na maioria dos sistemas de reconhecimento de voz para palavras isoladas. Existem três etapas básicas no modelo do reconhecedor:

1- extração de parâmetros

2- Comparação de padrões

3- Regras de decisão

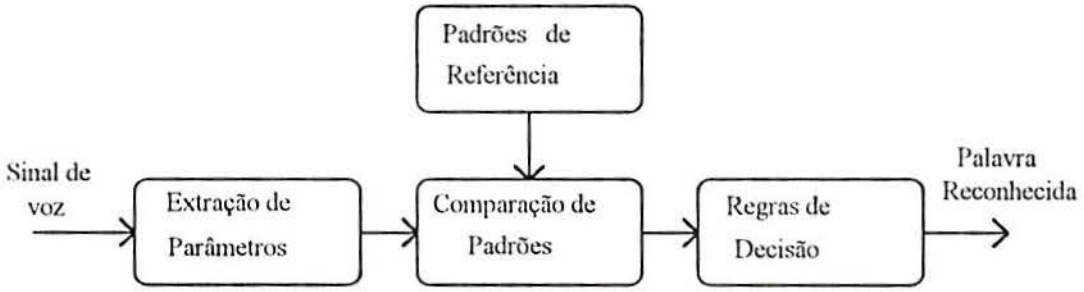


Figura 1.5 Modelo genérico de reconhecimento de padrões para reconhecimento de voz.

A entrada para o modelo é a forma de onda do sinal acústico da palavra pronunciada e a saída corresponde a melhor estimativa para esta palavra.

A extração de parâmetros é basicamente uma técnica de redução da quantidade de dados. Neste caso, amostras do sinal de voz adquirido a uma taxa apropriada, são transformadas em um pequeno conjunto de características que descrevem de maneira adequada as propriedades do sinal da voz. Taxas de redução de 10 a 100 vezes são comuns neste caso.

O segundo bloco do modelo é responsável pela comparação dos padrões, ou seja, uma vez extraídos os parâmetros do sinal em teste, esta etapa determina o grau de semelhança entre o padrão em teste e os padrões de referência.

O último passo é escolher entre os resultados obtidos na fase de comparação qual referência mais se aproxima do padrão em teste. A mais simples e usual aproximação para este procedimento é escolher aquele que, dependendo da técnica de comparação, apresenta a menor distância (ou distorção) entre o padrão de referência e o padrão em teste. A medida de distância está associada ao erro entre o padrão de referência e o sinal em teste.

A partir do modelo acima proposto serão apresentadas de maneira mais detalhada duas estruturas aplicáveis ao reconhecimento de sinais de voz para palavras isoladas. A diferença básica entre elas é a técnica empregada na fase de criação dos padrões de referência e de comparação com o padrão em teste.

A figura 1.6 apresenta os estágios iniciais comuns aos dois modelos. Esta fase compreende a digitalização do sinal de voz que inclui um filtro para limitar o sinal dentro da faixa da voz e evitar o *aliasing*. A etapa de pré-ênfase tem a função de tornar o espectro do sinal de voz mais plano. Quanto ao cálculo dos parâmetros, diversas alternativas têm sido propostas com os mais variados graus de complexidade. A escolha adequada destes

parâmetros depende das características do sistema. Vários meios de representação da voz serão apresentados e analisados neste trabalho. O último ponto a ser destacado é a detecção dos limites da palavra, ou seja, o início e o fim da palavra pronunciada, obtendo como resultado uma seqüência de vetores que representa a palavra. Esta seqüência será então comparada com os padrões de referência.

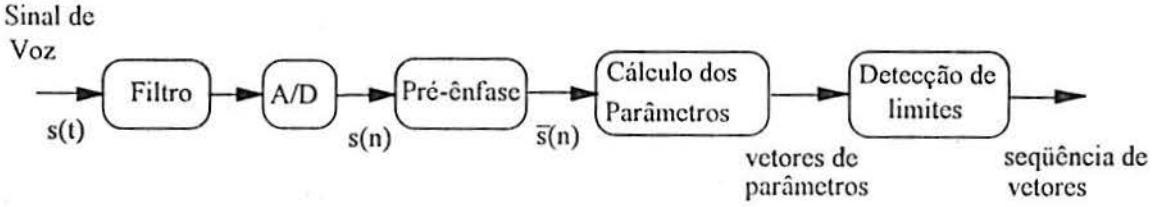


Figura 1.6 Componentes da fase de extração de parâmetros.

Outro item que estaria incluído nesta etapa é o tratamento do ruído. Neste trabalho não será analisada a influência de ruído no sinal de voz. Os experimentos realizados serão feitos em ambiente adequado de modo que a influência dos sinais interferentes possa ser desprezada.

Para as fases subseqüentes do sistema de reconhecimento automático, dois métodos para o reconhecimento de palavras serão abordados neste trabalho. As figura 1.7 e 1.8 mostram as estruturas em blocos dos dois métodos.

No primeiro método, figura 1.7, referido como "DTW" (*Dynamic Time Warping*), a seqüência de vetores com os parâmetros do sinal de voz, padrão em teste, é comparada com a seqüência de vetores de referência para uma determinada palavra, de modo a compensar as variações na velocidade que é pronunciada a palavra. A comparação é feita com todas as palavras do vocabulário (padrões de referência). Os padrões de referências são obtidos em uma fase de treinamento.

O resultado obtido com este método é a medida da distância entre a palavra em teste e cada uma das palavras do vocabulário. A medida de distância será vista em detalhes nos próximos capítulos.

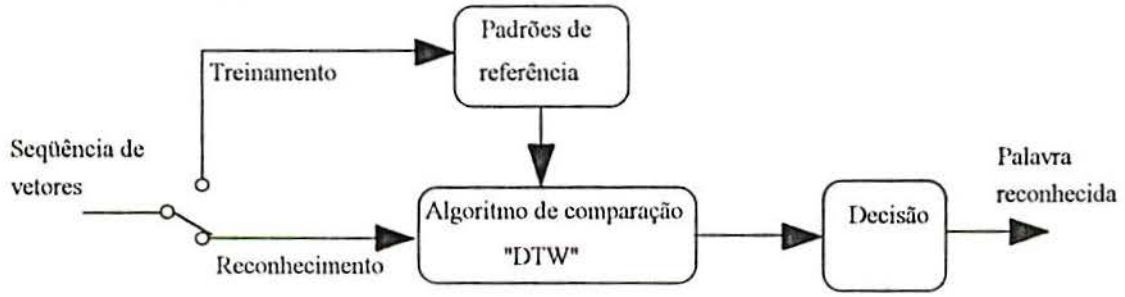


Figura 1.7 Diagrama em blocos de um reconhecedor de palavras utilizando DTW.

No segundo método, chamado "HMM" (*Hidden Markov Model*), existe um modelo estocástico para cada palavra. Na fase de reconhecimento é calculada a probabilidade de cada modelo estar associado à palavra em teste.

A palavra em teste neste caso é representada por uma seqüência de símbolos (números). Para a obtenção dos símbolos, cada vetor de entrada passa por um processo de classificação resultando em um símbolo associado a cada um destes vetores (quantização vetorial).

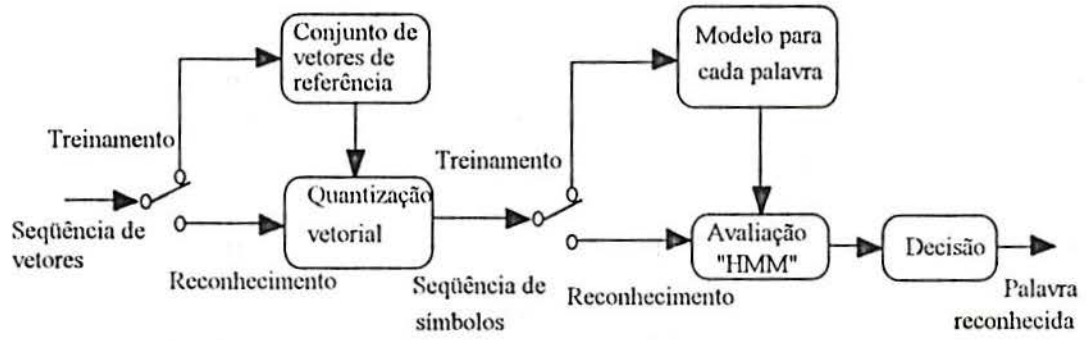


Figura 1.8 Diagrama em blocos de um reconhecedor de palavras baseado em HMM.

Os diversos aspectos relacionados aos métodos acima citados serão amplamente discutidos neste trabalho. Considerações a respeito da implementação prática com resultados experimentais serão apresentados.

## 2. REPRESENTAÇÃO DO SINAL DE VOZ

A complexidade computacional envolvida em um sistema de reconhecimento de voz envolve técnicas de processamento de sinais que não podem ser realizadas por métodos analógicos. Portanto, a primeira necessidade em um sistema de reconhecimento é a representação do sinal de voz no domínio digital.

A forma mais básica de representação digital do sinal de voz é a representação direta da forma de onda deste sinal. Este processo é baseado no teorema de amostragem de Shannon. De acordo com este teorema, qualquer sinal limitado em freqüência pode ser exatamente reconstruído a partir de amostras periódicas no tempo, desde que a taxa de amostragem seja igual ou superior ao dobro da máxima freqüência contida no sinal. O modo mais simples de representar a forma de onda de um sinal é codificar o sinal de forma linear usando um número fixo de bits por amostra. Este codificador de forma de onda é chamado PCM (Pulse Code Modulation).

Com relação à taxa de amostragem, sabe-se que o sinal de voz abrange a faixa de freqüências da ordem de 10 kHz. Portanto, para que toda a informação espectral seja representada, a mínima taxa de amostragem necessária é de 20 kHz. Por outro lado, nem toda esta informação é necessária para compreendermos a mensagem contida no sinal. Praticamente, as três primeiras formantes contêm esta informação, e estas estão tipicamente abaixo dos 3 kHz. Um exemplo disto são os sistemas telefônicos, cuja banda do sinal é limitada em cerca de 3 kHz. Na prática, taxas de amostragem entre 6 kHz e 20 kHz são usadas.

O número de bits por amostra necessários para codificar o sinal depende da relação sinal-ruído (*SNR*) desejada. Sendo $B$ a quantidade de bits usada para representar o sinal, a *SNR* resultante da quantização do sinal em $2^B$ níveis é [23]:

$$SNR_{dB} = 6.02B - \theta \qquad (2.1)$$

sendo $\theta$ função do degrau de quantização e da relação entre a amplitude máxima que pode ser codificada e o valor rms da amplitude do sinal. Por exemplo, para o sinal de voz são necessários 11 bits para atingir uma *SNR* de 60 db [31].

A digitalização da forma de onda da voz é apenas a primeira etapa no processo de reconhecimento, pois a forma de onda não é uma representação adequada do sinal de voz para ser usada diretamente como parâmetro de comparação. A variação temporal da amplitude do sinal de voz é fortemente afetada pelo ambiente e pelo locutor. Um exemplo desta variabilidade é apresentada na figura 2.1, onde as formas de onda correspondem ao fonema /a/ pronunciado isoladamente e amostrado a uma taxa de 10 kHz.

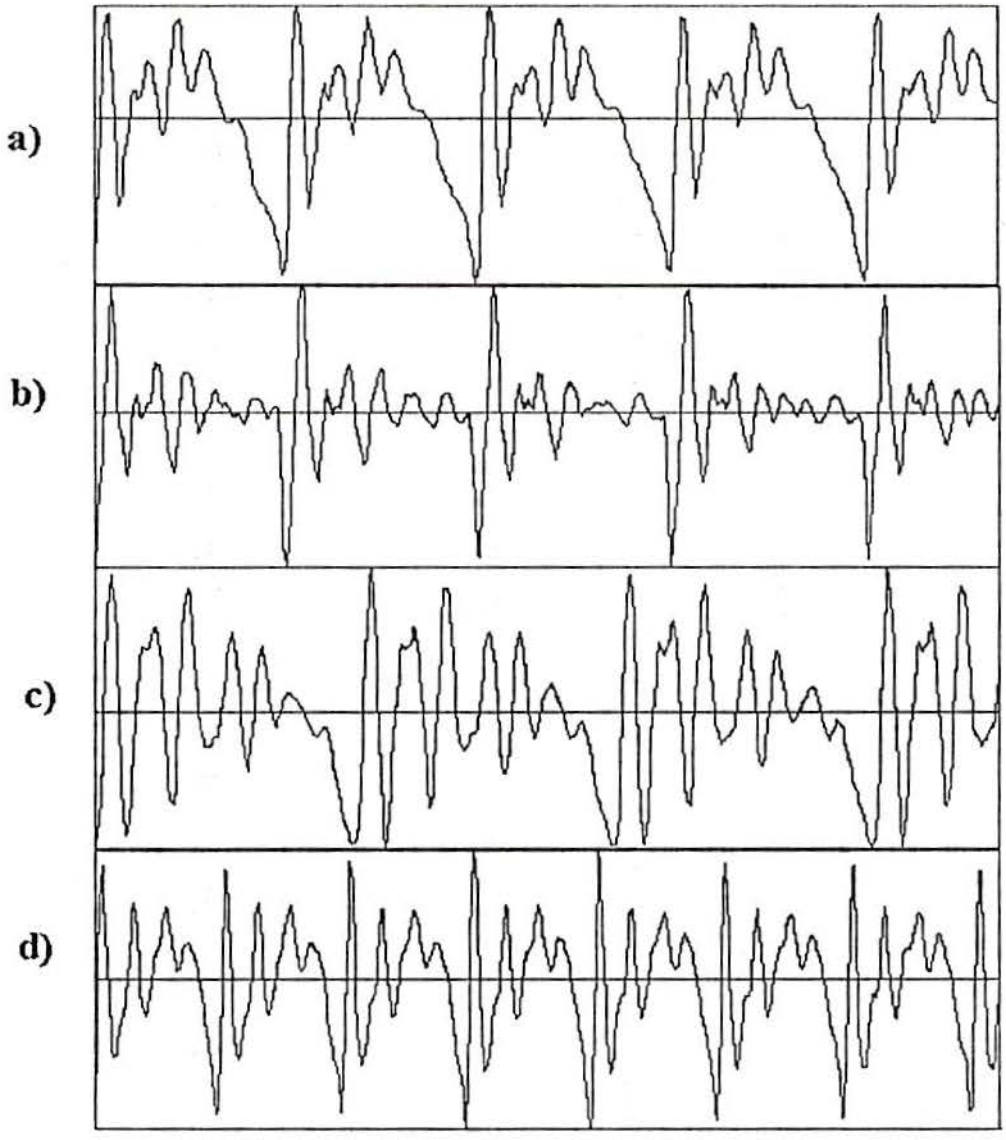


Figura 2.1 Exemplo da variabilidade da forma de onda do sinal de voz. Em a) e b) o fonema foi pronunciado pela mesma pessoa, porém, utilizando microfones diferentes; em c) e d) o fonema foi pronunciado por outras duas pessoas com o mesmo microfone do caso a); em a) e c) as pessoas são do sexo masculino e em d) do sexo feminino.

Vários fatores que não foram mostrados no exemplo acima, tais como ruído, efeitos da co-articulação e variação na amplitude do sinal, afetam sensivelmente a forma de onda. Esta variabilidade, além do grande volume de dados, inviabilizam o seu uso direto no processo de reconhecimento de voz.

A idéia básica das técnicas utilizadas para representação da voz é extrair da forma de onda do sinal um conjunto de características ou parâmetros que descrevam as propriedades acústicas da voz. Neste capítulo serão apresentados diversos conjuntos de características usadas na análise do sinal de voz. Tanto os métodos de extração destes parâmetros como a aplicabilidade dos mesmos no reconhecimento de voz serão abordados detalhadamente.

Dados dois vetores com parâmetros que caracterizam um determinado sinal, é necessário obter-se uma medida do grau de semelhança entre estes vetores. Na verdade, a avaliação feita neste caso é uma medida da diferença entre os parâmetros. Diversos métodos têm sido investigados para a obtenção de tais medidas. Tais métodos são genericamente conhecidos como medida de distância ou medida de distorção. Neste capítulo serão vistas as medidas de distorção utilizadas nos diversos tipos de representação do sinal de voz.

Também será apresentado o processo de quantização vetorial. Neste processo, vetores com características semelhantes são quantizados por um único vetor. Deste modo, o sinal de voz é caracterizado por um número finito de vetores resultando em uma grande redução na quantidade de dados.

## 2.1 Extração de parâmetros do sinal de voz

Muitos dos métodos de representação digital do sinal de voz têm por objetivo representar o sinal da maneira tão precisa quanto possível de modo que um sinal acústico possa ser reconstruído a partir da informação extraída do sinal original. No caso do reconhecimento automático de voz, não existe interesse na reconstrução do sinal; a única preocupação é representar o sinal de modo que as propriedades ou parâmetros obtidos possam ser usados para identificá-lo.

O sinal de voz é um sinal não estacionário. Entretanto, a propriedade básica na qual são baseados os métodos de representação da voz usados nos sistemas de reconhecimento, é que as propriedades da forma de onda podem ser consideradas invariantes durante um período de tempo na ordem de 10 ms a 30 ms [31]. A voz é geralmente representada como uma seqüência de parâmetros obtidos da análise de segmentos curtos do sinal de voz espaçados

uniformemente no tempo. Na prática, a periodicidade com a qual é feita esta análise do sinal varia de 5 ms a 20 ms.

Para analisar o sinal de voz em um período curto de tempo, este deve ser multiplicado por uma janela de tempo. A duração da janela, ou o número de amostras, deve ser escolhida de modo que o sinal possa ser considerado invariante neste intervalo. A forma da janela irá influenciar as características espectrais do sinal (ver seção 2.1.3).

### 2.1.1 Medida de energia

Uma das maneiras mais simples de representar um sinal é através de sua energia. Sendo $x[n]$ a n-ésima amostra do sinal $x$, a energia pode ser definida como:

$$E = \sum_{n=-\infty}^{\infty} x^2[n] \tag{2.1}$$

Para o caso do sinal de voz, o qual é não estacionário, a variação temporal da energia pode ser calculada da seguinte maneira:

$$E[n] = \sum_{m=0}^{N-1} \left[ w[m]x[n-m] \right]^2 \tag{2.2}$$

onde $w[m]$ é a janela aplicada ao sinal $x[n]$, e $N$ é o número de amostras na janela.

A medida de $E[n]$ é útil para mostrar as características da variação temporal da potência do sinal. A escolha de $N$ deve ser adequada pois, se $N$ for muito pequeno (menor que o período fundamental) $E[n]$ apresentará muitas flutuações. Se $N$ for grande demais, $E[n]$ terá uma variação muito pequena, e não irá refletir de maneira adequada a variação de amplitude do sinal de voz. Uma escolha adequada na prática para a janela é um período da ordem de 10 ms a 20 ms [31].

O cálculo da energia pela equação (2.2) tem a propriedade de dar grande ênfase aos sinais de maior amplitude. Este efeito pode ser minimizado usando os valores absolutos ao invés dos quadrados, temos então:

$$\hat{E}[n] = \sum_{m=0}^{N-1} |w[m]x[n-m]| \qquad (2.3)$$

Outra alternativa seria computar o logaritmo de $E[n]$. Esta representação facilita a observação dos sinais de baixa amplitude, como por exemplo, o ruído de fundo.

A figura 2.2 mostra as três representações de energia $E[n]$, $\hat{E}[N]$ e $\log E[n]$ para a palavra /teste/. Neste exemplo, o sinal foi amostrado com uma taxa de 20 kHz, sendo usada a janela retangular $w[m]=1$ com $N = 400$. O cálculo da energia foi executado a cada 10 ms, ou seja, a cada 200 amostras.

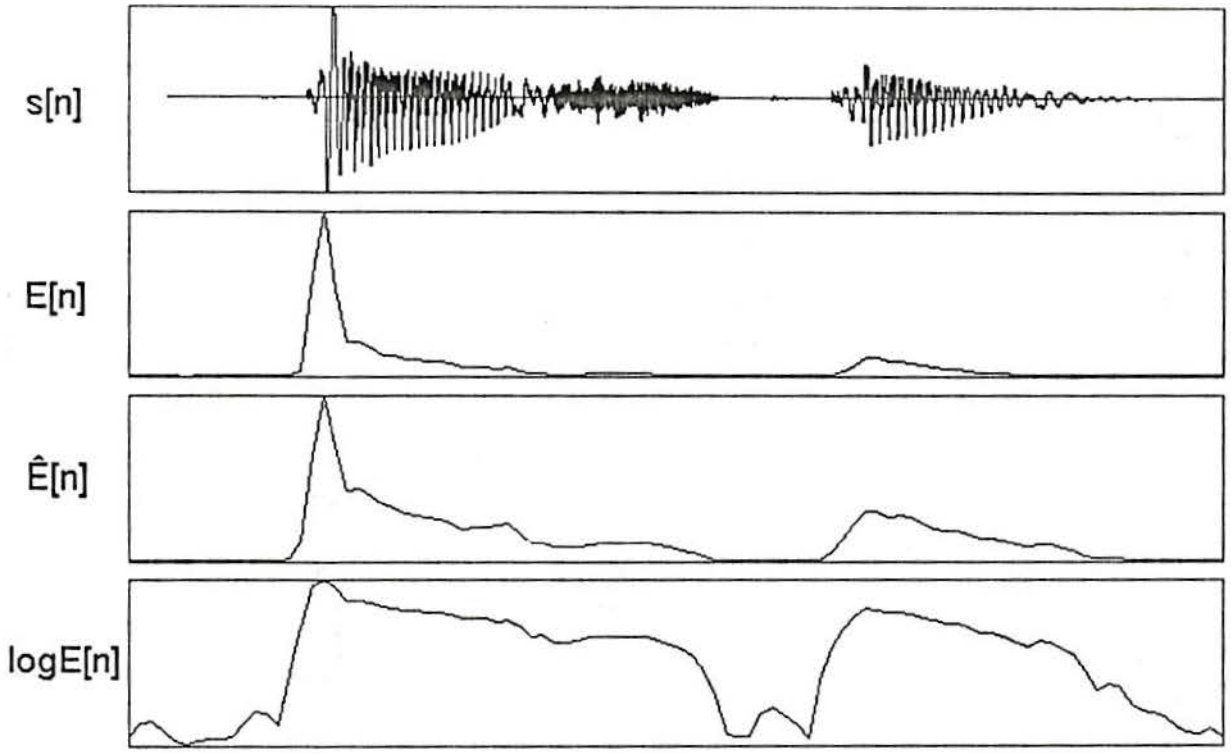


Figura 2.2 Energia para a palavra /teste/ (escala normalizada).

Em sistemas de reconhecimento de voz, a medida de energia pode ser usada para separar sons vozeados dos sons não-vozeados, já que a energia nos segmentos vozeados é maior que nos segmentos não-vozeados. A energia também pode ser usada na determinação dos limites da palavra. Neste caso, é escolhido um limiar de energia abaixo do qual o sinal é classificado como sendo silêncio.

O uso isolado da energia em um sistema automático de reconhecimento de voz não é, por si só, um parâmetro muito bom pois o conteúdo de informação é bastante pobre. Além disso, o valor absoluto da energia é bastante sensível ao locutor e ao ambiente. Uma alternativa para evitar o problema da variação da amplitude é normalizar a energia em função da estimativa do valor máximo. Em sistemas que operam em tempo real, o atraso necessário para estimar o máximo pode ser crítico, principalmente quando a fala é contínua. Neste caso, algum tipo de controle automático de ganho que minimize o problema pode ser usado.

A medida de energia pode ser útil quando utilizada em conjunto com outros parâmetros. Em sistemas simples, do tipo dependente do locutor, pequeno vocabulário e palavras isoladas, esta representação do sinal de voz pode ser usada como um dos principais parâmetros de comparação podendo atingir bons resultados [11]. Em sistemas mais complexos ela pode ser usada como um parâmetro adicional para melhorar o desempenho.

Ao invés de utilizar diretamente o valor absoluto da energia, a energia diferencial ou transitória pode ser usada. Esta medida fornece informação a respeito das variações relativas na amplitude do sinal. Esta medida mostrou-se mais significativa para o processo de reconhecimento que a energia absoluta [12]. A energia diferencial é calculada por:

$$DE[n] = E[n+\delta] - E[n-\delta] \tag{2.4}$$

onde $\delta$ é um valor inteiro escolhido arbitrariamente.

### 2.1.2 Cruzamentos por zero

A medida de cruzamentos por zero (*zero-crossing*) é outra forma simples de representar um sinal. Considerando o sinal amostrado $x[n]$, pode-se definir matematicamente a função $u[n]$ como sendo:

$$u[n] = \frac{x[n]}{|x[n]|} \tag{2.5}$$

onde $|x[n)|$ é o valor absoluto de $x[n]$. Desta forma $u[n]$ representa a polaridade ou o sinal de $x[n]$. Após definido $u[n]$, um cruzamento por zero ocorre entre os instantes de amostragem $n$ e $n$-1 se:

$$u[n] \neq u[n-1]. \tag{2.6}$$

A técnica mais simples de representação utilizando cruzamento por zero é a medida do número de vezes que ocorre um cruzamento por zero em um determinado intervalo de tempo (janela).

Este método é simples de ser implementado na prática, já que se resume na comparação dos sinais de duas amostras sucessivas. Entretanto, a medida de cruzamentos por zero é extremamente sensível à presença de ruído no sinal. Uma alternativa para a diminuir a influência do ruído é utilizar um detector de cruzamento por zero com histerése. Neste caso a função $u[n]$ tem um comportamento conforme a figura 2.3, onde $S$ é o valor de comparação e deve estar acima da amplitude do ruído do sistema.

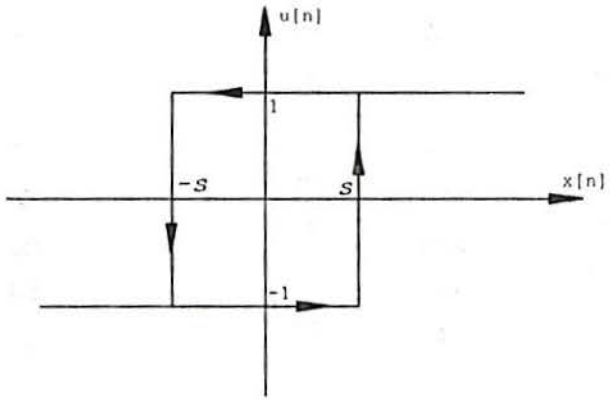


Figura 2.3 Função $u[n]$ para o detector de cruzamento por zero com histerése.

Outras técnicas de análise utilizando cruzamento por zero são apresentadas por Niederjohn [22].

A taxa de cruzamento por zero pode ser usada para estimar de maneira grosseira as propriedades espectrais do sinal. A medida de cruzamentos por zero é usada para avaliar se um determinado segmento de voz é vozeado ou não-vozeado. Sons vozeados apresentam a energia concentrada nas freqüência abaixo de 3 kHz, enquanto que, nos sons não-vozeados a

energia é mais concentrada acima de 3 kHz. Portanto, se a taxa de cruzamento por zero for baixa, o som pode ser classificado como vozeado; se a taxa de cruzamento por zero for alta, é mais provável que o segmento seja não-vozeado.

A figura 2.4 mostra a medida de cruzamentos por zero para a palavra /teste/.

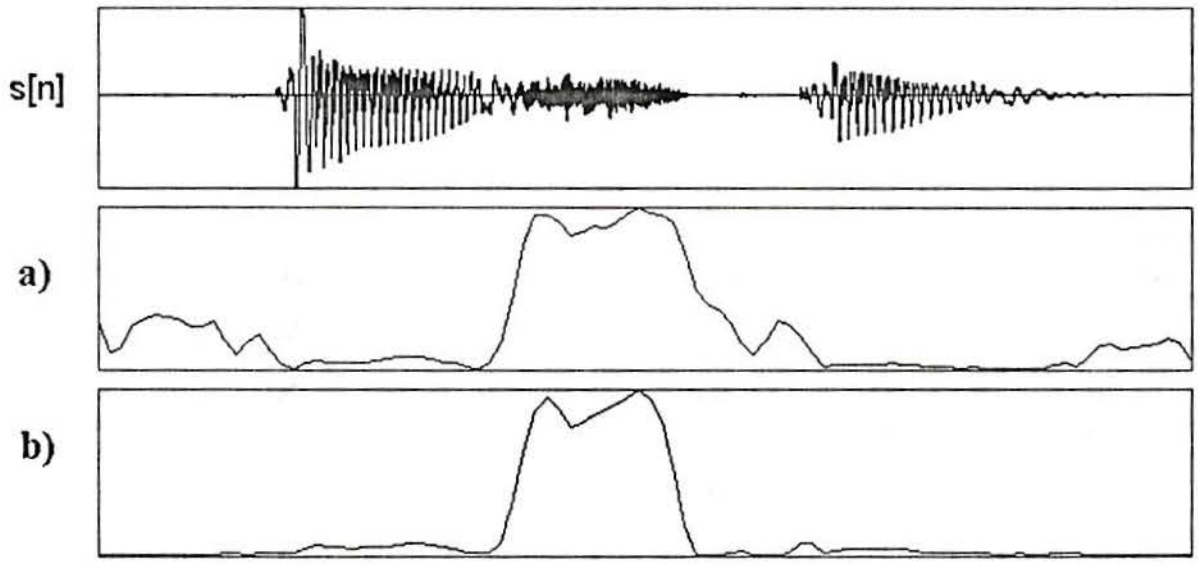


Figura 2.4 Exemplo de medida da taxa de cruzamento por zero para a palavra /teste/. a) medida direta e b) medida utilizando o comparador com histerése.

A taxa de cruzamento por zero também pode ser usada para estimar as freqüências formantes do sinal de voz [5]. A estimativa é feita passando o sinal por uma série de filtros passa-banda e medindo-se a taxa de cruzamento por zero dos sinais de saída dos filtros que apresentam os maiores níveis de energia.

A medida de cruzamentos por zero é também aplicada na detecção dos limites das palavras e pode ser utilizada como representação para o sinal de voz em sistemas simples de reconhecimento de voz [11].

2.1.3 Análise espectral

A medida das freqüências contidas no sinal de voz é uma das técnicas mais importantes na análise das características acústicas da fala. A base matemática da análise em freqüência é a transformada de Fourier. A Transformada de Fourier para sinais discretos é dada por:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \qquad (2.7)$$

onde $\omega$ é a freqüência.

Como colocado anteriormente, o sinal de voz é não estacionário, mas em um intervalo de tempo suficientemente curto, ele pode ser considerado invariante. Então, a representação espectral da voz pode ser obtida aplicando-se a Transformada de Fourier sobre um segmento do sinal de voz com N amostras. Este cálculo é feito pela Transformada de Fourier Discreta (DFT - Discrete Fourier Transform), dada pela expressão:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j(2\pi/N)kn} \quad , \ k=0,1,...,N-1. \qquad (2.8)$$

A DFT pode ser calculada de forma eficiente por algoritmos de Transformada Rápida de Fourier (FFT - *Fast Fourier Transform*) [23].

O sinal $x[n]$ é geralmente multiplicado por uma janela no tempo $w[n]$, com a duração de $N$ amostras. Na equação 2.8 foi usada a janela retangular, isto é, $w[n]=1$ para $0 \leq n$ · N. Outras janelas como a janela de Hamming e a janela de Hanning são geralmente usadas. Estas janelas diminuem as distorções causadas pela descontinuidade nos limites da janela.

Um dos usos do cálculo do espectro do sinal de voz é produzir um espectrograma, o qual mostra a energia do sinal em função da freqüência e do tempo, conforme apresentado na figura 1.1a. Esta representação pode ser usada para o reconhecimento de voz. Porém, em um sistema automático de reconhecimento que opere em tempo real, esta representação não é adequada pois o volume de dados a ser processado é muito grande.

As componentes espectrais obtidas através da FFT estão distribuídas linearmente em intervalos iguais de freqüência. Em sistemas de reconhecimento de voz, estes valores são geralmente redistribuídos em uma escala aproximadamente logarítmica de freqüência, levando

em consideração as características do sistema auditivo humano. Duas escala propostas para este fim são a escala de Bark e a escala de Mel. Elas são definidas respectivamente pela equações 2.9 e 2.10 [5].

$$B = 13\arctan(0.76f) + 3.5\arctan\left(\frac{f}{7.5}\right)^2 \tag{2.9}$$

$$Mel = 1000\log_2(1+f) \tag{2.10}$$

onde f é a freqüência em kilohertz.

Uma alternativa para a análise espectral em sistemas de reconhecimento de voz, é usar a saída de um banco de filtros. Nesta implementação, o sinal de voz é aplicado a um conjunto de $Q$ filtros passa-banda, e a estimativa da energia da saída de cada filtro é usada como representação para o sinal de voz.

O número de filtros usados, $Q$, pode variar de 5 até 32 [28], e o espaçamento entre os filtros segue geralmente uma escala aproximadamente logarítmica, como por exemplo as escalas de Bark e Mel. O cálculo da energia na saída de cada filtro pode ser realizada pelas equações (2.1) a (2.3), ou, pode ser utilizado um retificador de onda completa e um filtro passa baixa.

### 2.1.4 Análise Cepstral

O cepstro (coeficientes cepstrais) é definido como sendo a transformada inversa de Fourier do logaritmo da amplitude da transformada de Fourier, isto é,

$$c[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log\left|X\left(e^{j\omega}\right)\right| e^{j\omega n} d\omega \tag{2.11}$$

Técnicas baseadas no cepstro podem ser usadas em sistemas que obedecem ao princípio da superposição, como é o caso dos sistemas lineares.

Como o sinal de voz pode ser produzido por um sistema linear, podemos utilizar a análise cepstral para separar a excitação $g(t)$ da resposta ao impulso do trato vocal $h(t)$. Então, o sinal de voz $s(t)$ é dado pela convolução de $g(t)$ com $h(t)$ (ver figura 1.3):

$$s(t) = g(t) * h(t) \tag{2.12}$$

que é equivalente a

$$S(w) = G(w)H(w) \tag{2.13}$$

onde $S(w)$, $G(w)$ e $H(w)$ são as transformadas de Fourier de $s(t)$, $g(t)$ e $h(t)$, respectivamente.

Tomando o logaritmo dos módulos tem-se

$$\log|S(w)| = \log|G(w)| + \log|H(w)| \tag{2.14}$$

Os coeficientes cepstrais (cepstro) de $s(t)$ são então obtidos por

$$c(\tau) = F^{-1}\log|G(w)| + F^{-1}\log|H(w)| \tag{2.15}$$

onde $F^{-1}$ é a transformada inversa de Fourier.

Calculando os coeficientes cepstrais para um janela de $N$ amostras temos [5][23]:

$$c[n] = \frac{1}{N}\sum_{k=0}^{N-1}\log|X[k]|e^{j2\pi kn/N}, \tag{2.16}$$

para $0 \le n \le N$.

A figura 2.5 [23] exemplifica o processo de análise cepstral aplicada ao sinal de voz.

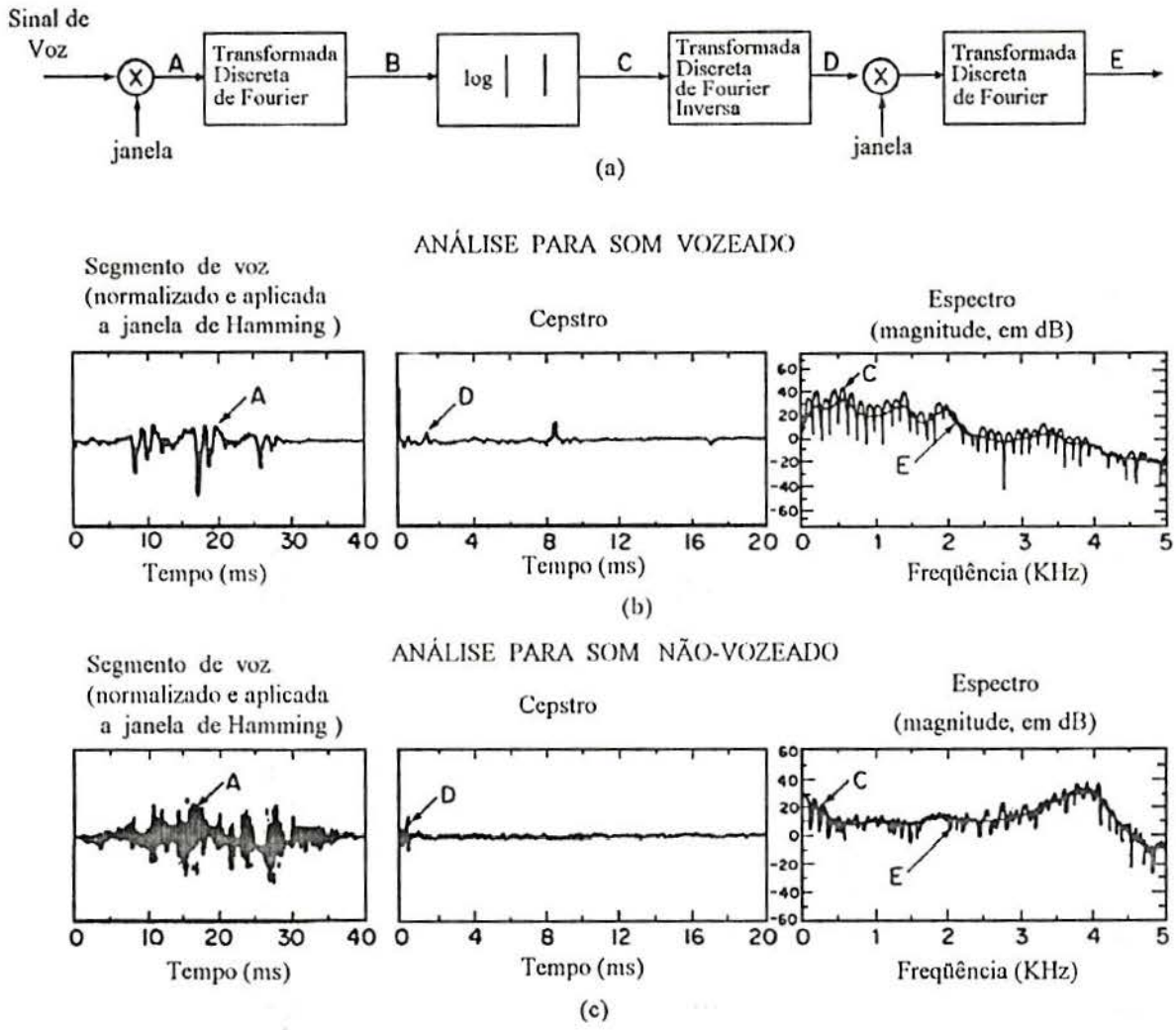


Figura 2.5 Análise cepstral de um segmento de um sinal de voz. (a) Operações básicas. (b) Análise para um sinal vozeado. (c) Análise para um sinal não-vozeado.

Os resultados das operações apresentadas na figura 2.5(a) podem ser vistos em (b) e (c), para sons vozeados e não-vozeados respectivamente. A curva C é o logaritmo do espectro do sinal de entrada A. Esta curva apresenta duas componentes, uma que contém a variação suave das componentes espectrais relacionada à função de transferência do trato vocal, e outra componente que varia rapidamente em função da freqüência causada pela excitação. A parcela que varia lentamente em A produz as componentes baixas na escala de tempo do cepstro D. No caso (b), onde o segmento de voz é vozeado; a componente periódica que aparece em C reflete-se no cepstro como um forte pico, neste caso, aproximadamente em 8 ms (pitch). Em (c) este pico não aparece devido à natureza randômica da excitação; deste modo, a componente de variação rápida no espectro não é periódica. Devido a estas características, o

cepstro serve para estimar o período fundamental da voz e para determinar quando um determinado segmento de fala é vozeado ou não-vozeado.

A função de transferência do trato vocal, também chamada de envelope espectral, pode ser obtida através do cepstro, multiplicando o mesmo por uma janela que somente passem as componentes de variação lenta no tempo, e calculando a DFT do cepstro resultante. O resultado deste processo é a curva E.

Como os primeiros coeficientes cepstrais estão relacionados com o envelope espectral, eles podem ser utilizados para representar o sinal de voz. Além de representar a função de transferência do trato vocal, estes coeficientes não contêm informação a respeito da fonte de excitação do trato g(t), o que é interessante para o processo de reconhecimento, pois a excitação é bastante variável com relação ao locutor. A representação do sinal de voz pelos coeficiente cepstrais tem apresentado bons resultados em sistemas de reconhecimento [6] [8] [12] [33].

Apesar da representação cepstral da voz ser usada em sistemas de reconhecimento, o método de cálculo apresentado na definição acima geralmente não é o empregado em sistemas práticos devido à complexidade computacional. Na prática, eles são derivados da representação do trato vocal obtido a partir da técnica de predição linear, que será apresenta na seção 2.1.5.1.

### 2.1.5 Codificação linear Preditiva (LPC)

Uma alternativa para a representação do sinal de voz é encontrar parâmetros que, ao invés de descreverem o sinal de voz, determinem as características do modelo de produção da voz. Como já mostrado, pode-se modelar a geração da fala com um filtro que é excitado por uma seqüência quase periódica de impulsos ou por um fonte de ruído randômico. Os parâmetros do filtro determinam as caraterísticas espectrais do som emitido. Portanto, considerando o modelo da figura 2.6, podemos representar a voz pelos parâmetros do filtro $H(z)$.

A análise por predição linear é um dos métodos mais utilizados para a representação da voz, apresentando bons resultados para diversas técnicas de reconhecimento de voz [2] [6] [8] [9] [14] [15] [30] [25] [27] [28] [33] [24].

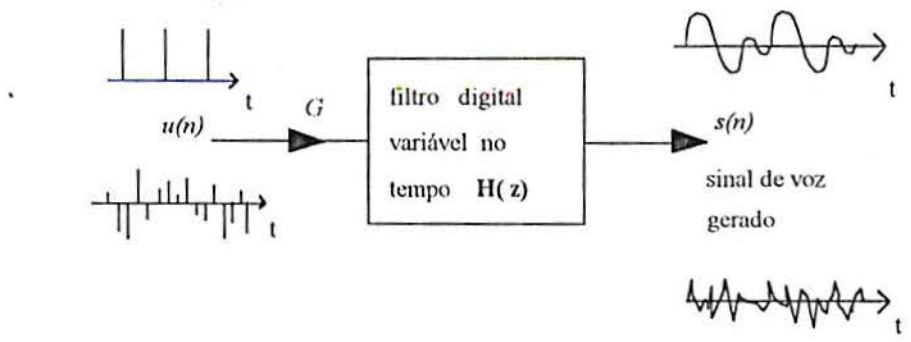


Figura 2.6 Modelo para produção de voz.

Considerando-se um sistema linear, podemos determinar a saída deste sistema $s[n]$ como uma combinação linear das entradas e saídas anteriores e da entrada atual. Deste modo o sistema pode ser descrito pela equação:

$$s[n] = -\sum_{k=1}^{p} a_k s[n-k] + G\sum_{l=0}^{q} b_l u[n-l] \quad ,b_0=1 \tag{2.17}$$

onde $a_k$, $b_l$ e $G$ são os parâmetros do sistema. O termo predição linear é usado pois o sinal $s[n]$ é estimado por uma combinação linear das amostras anteriores e da entrada atual do sistema.

No domínio freqüência a função de transferência $H(z)$ pode ser obtida pela transformada Z da equação (2.17). Obtemos então:

$$H(z) = G\frac{1+\sum_{l=1}^{q} b_l z^{-l}}{1+\sum_{k=1}^{p} a_k z^{-k}} \tag{2.18}$$

A $H(z)$, conforme a equação (2.18), é o modelo geral de pólos e zeros do sistema. Os pólos e zeros do modelo são, respectivamente, as raízes dos polinômios no denominador e no numerador.

Podemos então representar a função de transferência do trato vocal por um modelo de pólos e zeros. Entretanto, a determinação dos parâmetros deste modelo é bastante complexa.

A maior dificuldade do problema é que, para se determinar os valores de $a_k$ e $b_l$, o problema recai na solução de equações não-lineares. Para resolver estas equações são necessários métodos iterativos, cuja convergência para valores ótimos não é garantida [20].

Os zeros na função de transferência do trato vocal, geralmente presentes nos sons não-vozeados e nos sons nasais, se localizam dentro do círculo de raio unitário do plano Z. Portanto, cada zero da função de transferência pode ser aproximado por múltiplos pólos no denominador desta função [1]. Assim, o efeito dos zeros da função de transferência pode ser compensado de maneira satisfatória por um modelo que contenha apenas pólos. Neste modelo $b_l=0$, para qualquer $l \neq 0$ (saídas anteriores).

O modelo somente-pólos é amplamente utilizado na representação da voz por ser relativamente simples e pela disponibilidade de eficientes algoritmos para determinar seus parâmetros. A figura 2.7 mostra um modelo digital somente-pólos no domínio tempo onde o sinal $s[n]$ é a combinação linear das p saídas anteriores e da entrada atual $u[n]$.

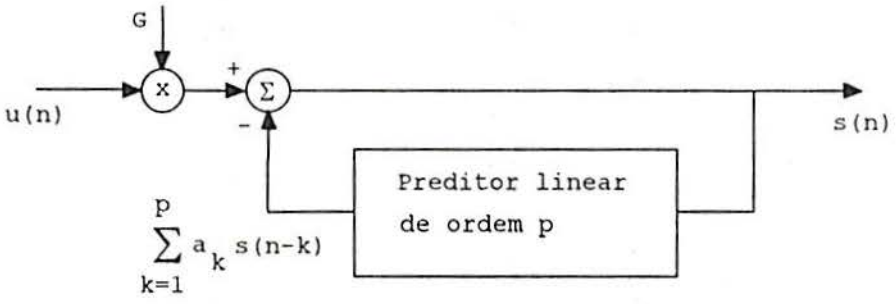


Figura 2.7 Modelo digital somente-pólos.

Neste modelo o sinal s[n] é dado por:

$$s[n] = -\sum_{k=1}^{p} a_k s[n-k] + Gu[n] \tag{2.19}$$

onde G é o ganho para a entrada atual. A função de transferência fica então:

$$H(z) = \frac{G}{1+\sum_{k=1}^{p} a_k z^{-k}} \tag{2.20}$$

O uso da predição linear é de grande importância na codificação da voz, pois permite que as características da voz sejam representadas de maneira eficiente e precisa utilizando um pequeno conjunto de dados. A seguir será apresentado a técnica para o cálculo dos parâmetros do preditor e de algumas representações alternativas para os mesmos.

### 2.1.5.1 Determinação dos parâmetros do preditor

Para determinar os parâmetros do modelo da figura 2.5 devemos considerar que o único sinal conhecido é o sinal de voz amostrado $s[n]$. Portanto, a função de transferência $H(z)$ deve ser estimada sem o conhecimento prévio da fonte de excitação do sistema. Deste modo supomos, como aproximação, que o sinal $s[n]$ pode ser estimado pela combinação linear das amostras anteriores do sinal. Temos então:

$$\tilde{s}[n] = -\sum_{k=1}^{p} \alpha_k s[n-k] \tag{2.21}$$

O erro entre o sinal amostrado $s[n]$ e o sinal estimado $\tilde{s}[n]$ é dado por:

$$e[n] = s[n] - \tilde{s}[n] = s[n] + \sum_{k=1}^{p} \alpha_k s[n-k] \tag{2.22}$$

$e[n]$ é denominado erro de predição ou resíduo. Segundo as equações (2.19) e (2.22), pode ser visto que o erro de predição será mínimo quando $\alpha_k = a_k$. Os parâmetros $\alpha_k$ podem ser obtidos pelo método dos mínimos quadrados, onde o erro é minimizado para cada um dos parâmetros.

O erro quadrático total $E$ é:

$$E = \sum_n e^2[n] = \sum_n \left( s[n] + \sum_{k=1}^{p} \alpha_k s[n-k] \right)^2 \tag{2.23}$$

Os valores de $\alpha_k$ que minimizam E são obtidos quando

$$\frac{\delta E}{\delta \alpha_i} = 0 \quad , \qquad \text{i=0,1,...,p.} \tag{2.24}$$

Substituindo a equação (2.23) em (2.24) obtém-se [20]

$$\sum_{k=1}^{p} \alpha_k \sum_n s[n-k]s[n-i] = -\sum_n s[n]s[n-i] \tag{2.25}$$

para $1 \le i \le p$. Os coeficientes $\alpha_k$ são calculados resolvendo-se este sistema de $p$ equações e $p$ incógnitas.

Para o cálculo do sistema (2.25) deve ser definida a faixa de valores $n$ sobre a qual é efetuada a soma. Para o caso do sinal de voz, a análise deve ser feita em intervalos curtos de tempo no qual as características do mesmo podem ser consideradas quase constantes. Um método adequado para a análise é multiplicar o sinal $s[n]$ por uma janela $w[n]$ obtendo $\hat{s}[n]$:

$$\hat{s}[n] = \begin{cases} s[n]w[n], & 0 \le n \le N-1 \\ 0, & \textit{para outros valores} \end{cases} \tag{2.26}$$

A função de autocorrelação do sinal $\hat{s}[n]$ é dada por:

$$R[i] = \sum_{n=0}^{N-1-i} s[n]s[n+i] \tag{2.27}$$

que é uma função par, isto é, $R[i]=R[-i]$. Neste caso (2.25) fica:

$$\sum_{k=1}^{p} \alpha_k R[i-k] = -R[i], \quad 1 \le i \le p \tag{2.28}$$

Os coeficientes $R[i\text{-}k]$ formam a matriz de autocorrelação, por isso o método baseado na equação (2.28) é conhecido como método da autocorrelação. A matriz de autocorrelação é um a matriz simétrica onde todos os elementos de cada diagonal são iguais (matriz Toeplitz). Expandindo a equação (2.28) na forma matricial temos:

$$\begin{bmatrix} R[0] & R[1] & R[2] & \cdots & R[p-1] \\ R[1] & R[0] & R[1] & \cdots & R[p-2] \\ R[2] & R[1] & R[0] & \cdots & R[p-3] \\ \vdots & \vdots & \vdots & & \vdots \\ R[p-1] & R[p-2] & R[p-3] & \cdots & R[0] \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_p \end{bmatrix} = - \begin{bmatrix} R[1] \\ R[2] \\ R[3] \\ \vdots \\ R[p] \end{bmatrix} \qquad (2.29)$$

Qualquer método de resolução de sistemas lineares pode ser utilizado para determinar os coeficientes $\alpha_k$. Entretanto, devido às características de simetria e igualdade dos elementos das diagonais, o sistema acima pode ser calculado de maneira mais eficiente pelo algoritmo recursivo de Durbin [20]. O método de Durbin segue o seguinte procedimento:

$$E_0 = R(0) \qquad (2.30a)$$

$$k_i = -\left[ R(i) + \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R(i-j) \right] \Big/ E_{i-1} \qquad (2.30b)$$

$$\alpha_i^{(i)} = k_i \qquad (2.30c)$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} + k_i \alpha_{i-j}^{(i-1)}, \quad 1 \le j \le i-1 \qquad (2.30d)$$

$$E_i = \left(1 - k_i^2\right) E_{i-1}. \qquad (2.30f)$$

onde $\alpha_j^{(i)}$ é o coeficiente de ordem $j$ na i-ésima iteração. As equações (2.30a) a (2.30f) são resolvidas recursivamente para i=1,2,...,$p$. A solução final é dada por:

$$\alpha_j = \alpha_j^{(p)}, \quad 1 \le j \le p. \qquad (2.30e)$$

A solução da equação (2.29) não é afetada se os coeficientes $R[i]$ forem normalizados. Os coeficientes de autocorrelação normalizados $r[i]$ são definidos como:

$$r[i] = \frac{R[i]}{R[0]}. \tag{2.31}$$

Esta normalização é útil em aplicações cujas operações utilizam aritmética de ponto fixo, pois $|r[i]| \leq 1$.

O valor $E_i$ calculado a cada iteração em (2.30) é o erro mínimo total, e deve diminuir ou permanecer o mesmo à medida que a ordem do preditor cresce [20]. Portanto:

$$0 \leq E_i \leq E_{i-1}, \quad E_0 = R[0]. \tag{2.32}$$

Quando os coeficientes de autocorrelação normalizados são usados, o erro mínimo total $E_i$ também é dividido por $R[0]$.

O ganho $G$ é dado por

$$G^2 = R[0] + \sum_{k=1}^{p} \alpha_k R[k] \tag{2.33}$$

onde $G2$ é a energia do sinal de entrada $Gu[n]$.

Outro método para determinar os coeficientes $\alpha_k$ a partir da equação (2.25) é o método da covariância. Neste método, o erro $E$ na equação (2.23) é minimizado para um intervalo finito. Neste caso, $0 \leq n \leq N\text{-}1$, a equação (2.25) resulta em:

$$\sum_{k=1}^{p} \alpha_k \varphi_{ki} = -\varphi_{0i}, \quad 1 \leq i \leq p \tag{2.34}$$

onde

$$\varphi_{ik} = \sum_{n=0}^{N-1} s[n-i]s[n-k] \tag{2.35}$$

é a covariância do sinal $s[n]$ para o intervalo de $N$ amostras. Os coeficientes $\varphi_{ki}$ formam a matriz de covariância. Esta matriz é simétrica ($\varphi_{ki}=\varphi_{ik}$), entretanto, os elementos de cada diagonal não são idênticos como no caso da matriz de autocorrelação.

Calculados os coeficientes do filtro $H(z)$, é necessário avaliar se o filtro resultante é estável. Para um filtro deste tipo ser estável é necessário que os pólos, que são as raízes do denominador, estejam dentro do círculo de raio unitário. Para o método da autocorrelação a estabilidade é garantida se $R_i$ é calculado a partir de um sinal não-nulo, o que não é válido para o método da covariância [20].

Mesmo no método da autocorrelação, a estabilidade de $H(z)$ deve ser verificada, pois erros de arredondamento podem levar a uma solução onde os pólos se localizem fora do círculo de raio unitário. Ao invés de calcular as raízes do denominador de $H(z)$, pode-se determinar a estabilidade através do algoritmo (2.30), onde a condição $E_i > 0$, $1 \le i \le p$, é necessária e suficiente para garantir a estabilidade *de* $H(z)$. Outra condição equivalente é $|k_i| < 1$, $1 \le i \le p$. Quando o método de resolução do sistema (2.25) não é o algoritmo (2.30), a estabilidade só pode ser garantida por outros métodos de teste. Um método é calcular os coeficientes $k_i$ pela equação (2.40) e então verificar a estabilidade.

Quanto à utilização de um ou de outro método, pelo método da covariância os coeficientes são calculados com mais precisão do que pelo método da autocorrelação, porém, existe o problema da estabilidade do filtro. Entretanto, se a janela, no caso da autocorrelação, for escolhida de modo que seja longa o suficiente para que a resposta ao impulso do filtro caia a valores pouco significativos, podem-se obter aproximações muito boas.

Na prática, o número de coeficientes $p$ usados varia entre 8 e 12. Segundo Atal e Hanauer [1], a memória do preditor linear deve ter duração igual a

$$\tau = 2\ell / c \qquad (2.36)$$

onde $\ell$ é o comprimento do trato e $c$ a velocidade do som. Um valor típico para $\tau$ é de 1 ms.

Os coeficientes do filtro preditor $\alpha_k$ podem ser usados diretamente como representação da voz em sistemas de reconhecimento. Além disso, os coeficientes podem ser usados de diversas maneiras para representar as propriedades do sinal de voz.

No modelo proposto, o filtro $H(z)$ representa a função de transferência do trato vocal. Portanto, a resposta em freqüência do trato é:

$$\left|H\left(e^{j\omega}\right)\right|^2 = \frac{G^2}{\left|1+\sum_{k=1}^{p}\alpha_k e^{-jk\omega}\right|^2} \qquad (2.37)$$

onde $\omega$ é a freqüência. A figura 2.8 mostra um exemplo onde o espectro foi obtido usando a equação (2.37) para diversas ordens do preditor. Neste exemplo, o sinal de voz (fonema /a/) foi amostrado a 10 kHz. O método empregado foi o da autocorrelação e o sinal foi multiplicado pela janela de Hanning com $N = 200$.

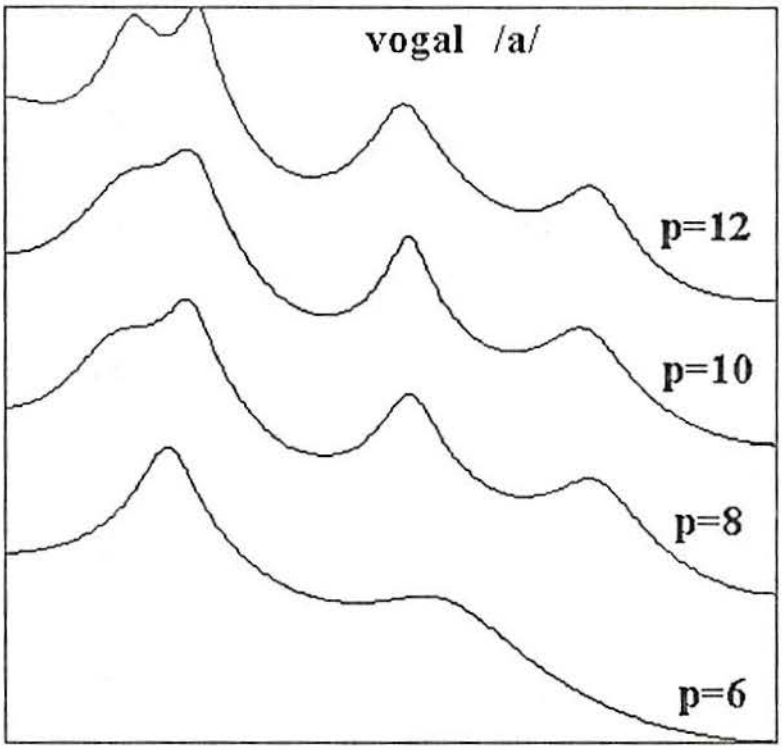


Figura 2.8 Espectro obtido por predição linear.

O espectro conforme mostrado na figura 2.8 pode ser usado para determinar as freqüências formantes. Isto pode ser feito automaticamente por um algoritmo de detecção de picos.

Uma forma simplificada da equação (2.37) é

$$\left|H\left(e^{j\omega}\right)\right|^2 = \frac{G^2}{\rho[0]+2\sum_{i=1}^{p}\rho[i]\cos(i\omega)}, \qquad (2.38)$$

onde

$$\rho[i] = \sum_{k=0}^{p-i} \alpha_k \alpha_{k+i} \qquad \alpha_0 = 1,\ 0 \le i \le p \tag{2.39}$$

são os coeficientes de autocorrelação de $\alpha_k$.

Outro parâmetro que pode ser usado para representar as características da voz são os coeficientes de reflexão, também chamados coeficientes de correlação parcial. Os coeficientes de reflexão $k_i$ são obtidos como um subproduto do cálculo dos coeficientes $\alpha_k$ pelo algoritmo de Durbin, equação (2.30b). Eles também podem ser obtidos a partir dos coeficientes do filtro preditor através do seguinte algoritmo:

$$k_i = \alpha_i^{(i)} \tag{2.40a}$$

$$\alpha_j^{(i-1)} = \frac{\alpha_j^{(i)} - \alpha_i^{(i)} \alpha_{i-j}^{(i)}}{1 - k_i^2}, \quad 1 \le j \le i-1 \tag{2.40b}$$

onde $i = p, p\text{-}1, \ldots, 2, 1$. A condição inicial é

$$\alpha_j^{(p)} = \alpha_j,\ 1 \le j \le p.$$

Considerando-se o trato vocal como uma seqüência de tubos com diferentes áreas, $k_i$ pode ser considerado como o coeficiente de reflexão entre duas seções. Sendo a relação entre as impedâncias acústicas de duas seções consecutivas dada por [20]

$$\frac{Z_{i+1}}{Z_i} = \frac{1+k_i}{1-k_i}, \quad 1 \le i \le p \tag{2.41}$$

Os coeficientes de reflexão também podem ser usados para estimar a área do trato vocal [5].

Como já apresentado em 2.1.4, outra representação da voz de grande interesse para o processo de reconhecimento são os coeficientes cepstrais. Os coeficientes cepstrais que estão

relacionados com a função de transferência do trato vocal são os primeiros coeficientes na escala de tempo (características espectrais de variação lenta). Tais parâmetros podem ser diretamente obtidos a partir dos coeficientes de predição linear $\alpha_k$ pela fórmula de recorrência [20]

$$c[n] = a_n - \sum_{m=1}^{n-1} \frac{m}{n} c[m] a_{n-m}, \quad 1 \le n \le p \tag{2.42}$$

## 2.2 Medida de distorção

A partir do conjunto de características ou parâmetros que representam um segmento de voz, é necessário determinar algum tipo de medida que avalie quantitativamente a semelhança ou a diferença entre dois conjuntos de parâmetros. Tais medidas são chamadas de medida de distorção ou medida de distância e são utilizadas para comparar dois vetores (conjuntos de parâmetros).

Uma medida de distorção d($\boldsymbol{x},\boldsymbol{y}$) é um número não-negativo que representa a distorção causada quando um vetor $\boldsymbol{x}$ é reproduzido por um vetor $\boldsymbol{y}$. Diversas medidas de distorção tem sido propostas e a escolha da medida adequada está relacionada ao tipo de representação da voz. Duas medidas de distorção podem ser consideradas equivalentes se, ao serem utilizadas para uma aplicação específica, não causam mudanças significativas no resultado. A seguir são apresentadas as principais medidas de distorção empregadas no processamento de voz.

A forma mais comum da medida de distorção é o erro quadrático médio (*mse - mean square error*)

$$d(x,y) = \frac{1}{N} \sum_{i=0}^{N-1} (x_i - y_i)^2 . \tag{2.43}$$

Onde N é a dimensão dos vetores. O mse é muito utilizado por causa de sua simplicidade. Também comuns são a norma euclidiana

$$d(x,y)=\left(\sum_{i=0}^{N-1}(x_i-y_i)^2\right)^{1/2}, \tag{2.44}$$

e o erro absoluto

$$d(x,y)=\sum_{i=0}^{N-1}|x_i-y_i|. \tag{2.45}$$

A medida (2.44) tem a característica de ser realmente uma medida de distância entre dois vetores, obedecendo a inequação

$$d(x,y)\le d(x,z)+d(z,y), \tag{2.46}$$

para qualquer $z$.

Outra medida de distorção é a média quadrática ponderada

$$d(x,y)=\frac{1}{N}\sum_{i=0}^{N-1}w_i(x_i-y_i)^2 \tag{2.47}$$

onde $w_i \ge 0$, i=0,1,2,...,N-1. Neste caso cada parâmetro pode receber pesos diferentes tornando a contribuição de certos parâmetros mais importante que a de outros. Uma forma mais geral de distorção quadrática é:

$$d(x,y)=(x-y)W(x-y)^T \tag{2.48}$$

$$=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}w_{i,j}(x_i-y_i)(x_j-y_j), \tag{2.49}$$

onde $W = \{w_{i,j}\}$ é uma matriz $N$ x $N$, geralmente uma matriz diagonal, e $N$ é o número de elementos de cada vetor.

Todas as medidas de distorção apresentadas até agora têm a propriedade de dependerem somente do vetor de erro $x$-$y$, o que as torna simples e fáceis de serem calculadas

em sistemas práticos. Entretanto, medidas de distorção mais complexas também são usadas para comparar vetores que caracterizam sinais de voz. Como a voz é geralmente representada e analisada através de suas características espectrais, são empregadas medidas de distorção que melhor avaliam as propriedades de tais características. Os coeficientes LPC são um exemplo onde medidas mais complexas de distorção são empregadas para a comparação da forma espectral da voz [7].

A medida de distorção utilizada quando o sinal de voz é representado pelo seu espectro pode ser feita por uma das formas apresentadas acima. Geralmente, o vetor diferença ***x-y*** é substituído pela diferença do logaritmo dos espectros. Por exemplo, considerando dois vetores $f$ e $\hat{f}$, os quais contém as componentes em freqüência de dois segmentos de voz, tem-se, para o erro quadrático médio:

$$d(f,\hat{f}) = \frac{1}{N}\sum_{i=0}^{N-1}\left(logf_i - log\hat{f}_i\right)^2 , \qquad (2.50)$$

o que é equivalente a

$$d(f,\hat{f}) = \frac{1}{N}\sum_{i=0}^{N-1}\left(log\frac{f_i}{\hat{f}_i}\right)^2 . \qquad (2.51)$$

Conforme apresentado no capítulo 1, a sensibilidade do sistema auditivo é aproximadamente logarítmica, portanto, esta medida de distorção inclui esta sensação subjetiva.

É importante considerar que a amplitude das componentes em freqüência é proporcional à intensidade com que cada palavra é pronunciada. Portanto, a energia do sinal de voz deve ser normalizada de alguma maneira para compensar estas variações.

Uma forma de compensar as variações de amplitude do sinal é normalizar a energia individualmente para cada vetor de modo que

$$\sum_{i=0}^{N-1}(\bar{f}_i)^2 = 1 , \qquad (2.52)$$

onde

$$\bar{f}_i = f_i \Big/ \left( \sum_{j=0}^{N-1} (f_j)^2 \right)^{1/2} \tag{2.53}$$

é a componente em freqüência normalizada em amplitude e $N$ é o número de parâmetros de cada vetor.

Outra alternativa para evitar o problema relativo às variações de energia dos espectros a serem comparados, é substituir o vetor com as componentes espectrais por um vetor que contenha a razão dos valores espectrais adjacentes [34], ou seja:

$$l_i = \frac{f_{i+1}}{f_i}, \quad 0 \le i \le N-2 \tag{2.54}$$

deste modo, a energia do sinal é desprezada e somente a relação entre as componentes espectrais é considerada.

Representando o vetor acima pelo logaritmo, tem-se:

$$L_i = \log f_{i+1} - \log f_i, \quad 0 \le i \le N-2 \tag{2.55}$$

portanto, a medida de distorção fica:

$$d(L, \hat{L}) = \frac{1}{N} \sum_{i=0}^{N-2} \left( L_i - \hat{L}_i \right)^2 \tag{2.56}$$

### 2.2.1 Medidas de distorção para vetores LPC

Quando os parâmetros a serem comparados são os coeficientes LPC a medida de distorção torna-se mais complexa. Esta análise pode ser interpretada como um método que minimiza a diferença entre o espectro do sinal de entrada e o espectro do modelo somente pólos. A medida de distorção usada neste caso é conhecida como distorção Itakura-Saito ($d_{IS}$), e tem a forma [24]

$$d_{IS}(R(\omega),T(\omega)) = \frac{1}{2\pi}\int_0^{2\pi}\left[\log\left(\frac{R(\omega)}{T(\omega)}\right)+\frac{T(\omega)}{R(\omega)}-1\right]d\omega \qquad (2.57)$$

onde $T$ é o espectro do sinal de entrada sob teste e $R$ é o espectro do modelo de referência somente pólos resultante dado por

$$R(\omega) = \frac{G^2}{\left|1+a_1e^{-j\omega}+a_2e^{-2j\omega}+\cdots+a_pe^{-pj\omega}\right|^2} \qquad (2.58)$$

Esta mesma medida de distorção pode ser usada para medir a diferença entre os espectros de dois modelos somente pólos. Dados dois conjuntos de parâmetros $a_R$ e $a_T$, a medida de distorção Itakura-Saito é calculada por

$$d_{IS}(a_R,a_T) = \frac{a_R R_T a_R^T}{G_T^{\,2}} + \log\left(\frac{G_T^{\,2}}{G_R^{\,2}}\right) - 1 \qquad (2.59)$$

onde $a_R=\{1,a_1,a_2,...,a_p\}^T$, $R_T$ é a matriz de autocorrelação obtida a partir da forma de onda do sinal de voz em teste, $G_R$ e $G_T$ são os ganhos dos modelos. Para o cálculo desta medida, G é obtido pela equação (2.33) e

$$a_R R_T a_R^T = R(0)\rho(0) + 2\sum_{i=1}^{p} R(i)\rho(i) \qquad (2.59a)$$

onde $R(i)$, conforme a equação (2.27), é função de autocorrelação do sinal e $\rho(i)$ são os coeficientes de autocorrelação dos $\alpha_k$ (ver equação (2.39).

A distância da forma da equação (2.59) apresenta o mesmo problema que no caso da medida através das componentes espectrais, pois ela é sensível à amplitude absoluta do sinal. Para distorções pequenas, esta medida se torna aproximadamente proporcional ao erro médio quadrático do logaritmo dos espectros [7]. Se for calculada a distorção entre um segmento de voz e o mesmo segmento multiplicado por uma constante a distorção computada será

$$d_{IS}(a_R, a_T) = \log\left(\frac{G_T^{\,2}}{G_R^{\,2}}\right) \neq 0 \qquad (2.60)$$

Para superar este problema são propostas duas versões para a medida de Itakura-Saito chamadas de distorção Itakura-Saito de ganho otimizado e distorção Itakura-Saito de ganho normalizado. A primeira, também conhecida como distorção de Itakura ou *log likelihood ratio* ($d_I$), é definida como

$$d_I(a_R, a_T) = \min_{\lambda > 0} d_{is}(a_R, \lambda a_T) \qquad (2.61)$$

e é calculada por

$$d_I(a_R, a_T) = \log\left[\frac{a_R R_T a_R^T}{a_T R_T a_R^T}\right], \qquad (2.62)$$

A segunda, (*likelihood ratio*) considera os ganhos $G_T$ e $G_R$ iguais e a medida de distorção é

$$d_{LR}(a_R, a_T) = \left[\frac{a_R V_T a_R^T}{a_T V_T a_T^T} - 1\right] \qquad (2.63)$$

Fica claro que nestas medidas a complexidade computacional é bem maior que nos casos baseados diretamente na diferença entre os vetores ($x_i$,$y_i$). Além de necessitar de mais operações ainda podem envolver operações logarítmicas.

Em contraste com os parâmetros do filtro preditor $a_i$, os coeficientes cepstrais derivados dos mesmos permitem a medida de distorção baseada no quadrado da diferença do respectivos coeficientes, pois esta medida, corresponde a distância entre o logaritmo dos espectros [5]. Uma variação para esta medida considera pesos diferenciados para os diversos coeficientes c[n]. A principal característica desta medida é que ela tende a equalizar as diferenças entre locutores diferentes. Segundo Tohkura [33], a escolha ótima para os pesos é o inverso da variância dos respectivos coeficientes. Também é observado que a variância é

pouco sensível ao conjunto de dados utilizado para extrair os parâmetros. A equação (2.64) mostra a simplicidade do cálculo da distância para os coeficientes cepstrais:

$$d(C_R, C_T) = \sum_{i=1}^{p} w_i \left( c_{R_i} - c_{T_i} \right)^2 \qquad (2.64)$$

onde $C_R=\{c_{R_i}\}$ são os coeficientes cepstrais da referência e $C_T=\{c_{T_i}\}$ os coeficientes cepstrais sob teste.

## 2.3 Quantização vetorial

Na técnica de reconhecimento HMM, descrita no capítulo 3, é necessário que a palavra pronunciada seja representada por um seqüência de parâmetros, os quais devem pertencer a um conjunto finito, ou seja, existe um número limitado de parâmetros possíveis. Para a obtenção e a utilização de tal conjunto é apresentado a seguir o processo denominado quantização vetorial.

A quantização vetorial é o processo pelo qual um vetor $\boldsymbol{x}=(x_1, x_2, ..., x_N)$ é mapeado para um novo vetor $\boldsymbol{y}$ de mesma dimensão. Diz-se que $\boldsymbol{x}$ é quantizado como $\boldsymbol{y}$, e $\boldsymbol{y}$ é o valor quantizado de $\boldsymbol{x}$ e escreve-se:

$$y = q(x) \qquad (2.65)$$

onde $q(\bullet)$ é o operador de quantização. Tipicamente $\boldsymbol{Y}$ assume um conjunto finito de valores $\boldsymbol{Y}=\{\mathbf{y_i},\ 1 \le i \le L\}$, onde $\mathbf{y_i}=(y_{i1}, y_{i2}, ..., y_{iN})$. O conjunto $\boldsymbol{Y}$ é referido como *codebook*, tabela de códigos, onde L é o número de vetores do *codebook*. No processo de quantização, o espaço N-dimensional do conjunto de vetores $\boldsymbol{x}$ usados para gerar o *codebook* é particionado em L regiões ou células $\{C_i,\ 1 \le i \le L\}$, cada célula $C_i$ é associada a um vetor $\mathbf{y_i}$. Tem-se então

$$q(x) = y_i, \quad se\ x \in C_i \qquad (2.66)$$

A figura 2.9 mostra um exemplo de partição de um espaço bidimensional. Qualquer vetor de entrada $x$ que estiver dentro da célula $C_i$ é quantizado como $y_i$.

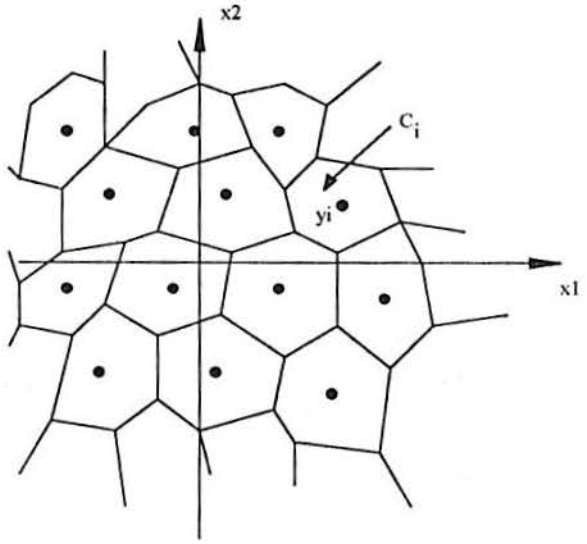


Figura 2.9 Células de um espaço bidimensional.

Quando um vetor $x$ é quantizado como $y$, ocorre um erro de quantização que pode ser avaliado pela medida de distorção d($x$,$y$).

O processo de quantização vetorial inicia com a obtenção de um *codebook* a partir de uma grande quantidade de vetores de treinamento $X$. O conjunto $\mathbf{Y}$ encontrado deve ser tal que minimize a distorção total de cada vetor de treinamento $x$ com seu respectivo vetor $y$.

Um quantizador é dito ótimo (ou globalmente ótimo) se ele minimizar a distorção média, ou seja, a distorção é minimizada sobre qualquer outro quantizador com o mesmo número de níveis L. Duas condições são necessárias mas não suficientes para a condição ótima. A primeira condição é a utilização da regra de seleção de mínima distorção (*nearest neighbor*):

$$q(x) = y_i, \quad se\ d(x,y_i) \le d\left(x,y_j\right),\ j \ne i,\ 1 \le j \le L. \tag{2.67}$$

A segunda condição necessária é que cada vetor $y_i$ seja escolhido de modo a minimizar a distorção média na célula $C_i$. O vetor $y_i$ é tal que minimiza

$$D_i = \xi[d(x,y)|x \in C_i] \tag{2.68}$$

onde $\xi[]$ representa a média. O vetor $y_i$ é chamado centróide da célula $C_i$

$$y_i = cent(C_i). \tag{2.69}$$

O cálculo do centróide depende da medida de distorção empregada. Para o caso do erro quadrático tem-se

$$D_i = \frac{1}{M_i} \sum_{x \in C_i} d(x, y_i), \tag{2.70}$$

onde $M_i$ é o número de vetores de treinamento associados à célula $C_i$ de acordo com o critério de mínima distorção. O centróide é obtido por [19]

$$y_i = \frac{1}{M_i} \sum_{x \in C_i} x, \tag{2.71}$$

ou seja, $y_i$ é simplesmente a média das amostras contidas na célula $C_i$. Para as medidas de distorção como a de Itakura-Saito o centróide é calculado por [17]

$$y_i = \left\{ \sum_{j:x_j \in C_i} R(x_j) \right\}^{-1} \sum_{j:x_j \in C_i} R(x_j) x_j^T. \tag{2.72}$$

onde $R(x_j)$ é a matriz de autocorrelação utilizada no cálculo dos coeficientes LPC que compõe o vetor $x_j$, equação (2.27).

A distorção média do quantizador para um dado conjunto de vetores de treinamento $X$, é dada por:

$$D(q) = \xi[d(X, q(X))] \tag{2.73}$$

Considerando o valor quantizado $y_i$ conforme (2.71) e um conjunto X de $M$ elementos tem-se

$$D(q) = \frac{1}{M}\sum_{j=1}^{M} d(x_j, q(x_j)) = \frac{1}{M}\sum_{j=1}^{M} d(x_j, y_i) \tag{2.74}$$

onde $y_i = q(x_j)$.

A condição ótima para um quantizador q de L níveis ocorre quando para qualquer outro quantizador q* de L centróides $D(q) \leq D(q^*)$.

### 2.3.1 Determinação do *codebook*

O método para o projeto do quantizador apresentado a seguir pode ser empregado genericamente para as diferentes medidas de distorção. O método é conhecido como algoritmo LBG [17] e atende às duas condições necessárias para atingir a situação de quantizador ótimo, porém a condição de ótimo global não é garantida. A solução geralmente não é única. A condição de ótimo global pode ser atingida inicializando os vetores $y_i$ com valores diferentes e repetindo o algoritmo com diversos conjuntos de inicialização. O quantizador que obtiver a menor distorção média é escolhido.

O algoritmo LBG divide o conjunto de M vetores de treinamento $\boldsymbol{X}$ em L grupos, cada grupo corresponde a uma célula $C_i$. O algoritmo é o seguinte:

Passo 1:

Inicialização: Dado um conjunto inicial de vetores $y_i$, ($1 \leq i \leq L$) (*codebook* inicial), um limiar de distorção $\varepsilon \geq 0$. Faça m = 0 e $D_{m-1} = \infty$.

Passo 2:

Classifique cada vetor de treinamento $\boldsymbol{x}$ na sua respectiva célula conforme o critério de mínima distorção (2.67): $\boldsymbol{x} \in C_i$ se $d(\boldsymbol{x}, y_i) \leq d(\boldsymbol{x}, y_j)$ para qualquer j. Calcule a distorção média para esta distribuição $D_m$.

Passo 3:

Se $(D_{m-1} - D_m)/D_m \leq \varepsilon$, o processo está completo e $Y_m$ é o conjunto de centróides do quantizador. Caso contrário continue.

Passo 4:

Calcule o centróide de cada conjunto encontrado no passo 2, $C_{im}$. Faça $Y_{m+1}=\{C_{im}\}$ e m $\leftarrow$m+1. Vá para o passo 2.

Existem diversas maneiras de inicializar $Y$. Um método (algoritmo K-means) utiliza os L primeiros vetores do conjunto $X$ como estimativa para os centróides. A cada nova iteração mais um vetor do conjunto de teste é acrescentado conforme o critério da mínima distorção (2.67) e é calculado o novo centróide. O processo termina com a inclusão do último vetor de $X$. Segundo Linde [17], esta escolha é adequada quando a seqüência de treinamento deve ser agrupada segundo o critério de baixa distorção, mas não apresenta bons resultados para quantizar vetores fora do conjunto de treino. Intuitivamente, é desejável que os vetores $y_i$ estejam "bem distribuídos", o que é pouco provável para amostras consecutivas.

Outra possibilidade para inicializar o quantizador é utilizar L vetores distribuídos no espaço de dimensão N, de modo que estejam dentro da região deste espaço que contenha todos ou a maioria dos vetores do conjunto de treinamento $X$. A distribuição pode ser feita, por exemplo, de maneira randômica. Diversas inicializações para o conjunto $Y$ podem ser tentadas até que o quantizador atinja o nível de distorção desejado.

Uma técnica alternativa é a da divisão binária (*split*). Neste método, o tamanho do codebook é aumentado até que se atinja o tamanho desejado ou até o nível de distorção média adequado. O algoritmo fica o seguinte:

Passo 1:

Faça K=1 (número de níveis do quantizador) e calcule o centróide de todo o conjunto de treinamento $X$. Este será o valor inicial do conjunto $Y_0(1)$.

Passo 2:

Dado o conjunto $Y$ contendo K vetores $\{y_i,i=1,2,...,K)$, substitua cada vetor por dois novos vetores $y_i + \varepsilon$ e $y_i - \varepsilon$, onde $\varepsilon$ é um vetor de perturbação. Y agora contém 2K vetores, portanto faça K=2K.

Passo 3:

Executar o algoritmo LBG.

Passo 4:

Se K=L o quantizador fica definido pelo conjunto Y(K) obtido no passo 3. Caso contrário vá para o passo 2.

No caso da distorção baseada no erro quadrático, Makhoul [19] propõe um método para inicializar a partição do espaço onde este é dividido em duas regiões por um hiperplano que passa pela média dos vetores de treinamento e que é ortogonal à dimensão que apresenta a maior variância para o conjunto de vetores. Os centróides dos dois conjuntos de vetores separados pelo hiperplano são usados como estimativa inicial. Um exemplo onde este método poderia ser empregado é o da codificação de voz pelos coeficientes cepstrais, cujo primeiro coeficiente é o que apresenta a maior variância [33].

Nesta técnica de divisão binária não é necessário que a divisão ocorra de maneira uniforme com a divisão de todos os ramos da árvore binária. Em qualquer ponto do processo a subdivisão de um centróide $Y_i$ pode ser interrompida. Com este procedimento qualquer tamanho de codebook pode ser conseguido. Além disso, durante o treinamento certos pontos (centróides) podem ter poucos vetores associados àquela célula e uma posterior divisão deste grupo pode não resultar na redução da distorção. Definido o tamanho final do codebook, um mínimo para a distorção pode ser obtido se a cada ponto no processo de divisão a distorção de cada célula for examinada. A célula que apresentar a maior distorção é subdividida e o processo é repetido até atingir o número de níveis desejado.

Para quantizar um vetor de entrada é necessária a pesquisa completa em todos os L vetores do conjunto $Y$, onde aquele que apresentar a menor medida de distorção com relação ao vetor de entrada é o resultado da quantização. Entretanto, o custo computacional pode ser reduzido com uma pequena diminuição no desempenho utilizando-se o método de pesquisa binário. Neste caso os centróides a cada subdivisão do espaço são armazenados e o vetor de entrada é quantizado conforme a árvore na figura 2.10 pelo caminho de menor distorção em cada nodo. O desempenho é reduzido pois podem ocorrer erros de quantização, ou seja, o vetor de entrada $x$ pode não ser quantizado pelo centróide ótimo.

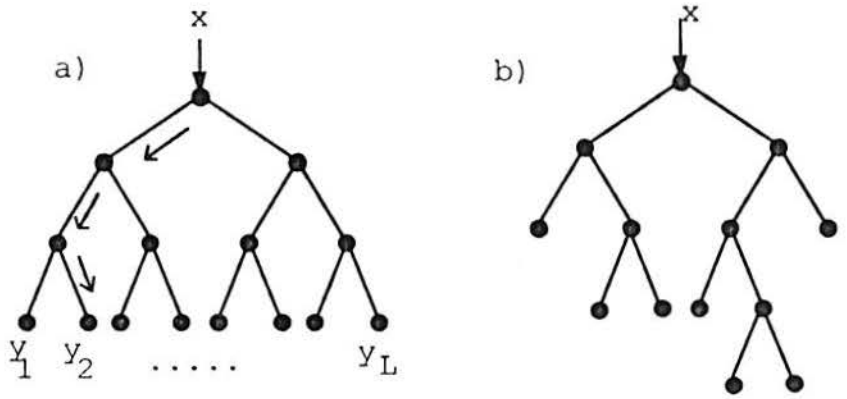


Figura 2.10 Árvore binária para um quantizador de pesquisa binária. a)uniforme, b) não uniforme.

O número total de cálculos de distorções se reduz a $2\log_2 L$ e o número de centróides armazenados cresce para 2(L-2). No caso da distorção do erro quadrático médio a redução do custo computacional pode ser ainda maior, pois o cálculo da distorção torna-se desnecessário. Ao invés de comparar o vetor de entrada com dois vetores, basta avaliar a posição do vetor em relação ao hiperplano que separa as duas regiões. Este teste envolve apenas o cálculo do produto escalar de dois vetores.

# 3. ALGORITMOS DE RECONHECIMENTO DE VOZ PARA PALAVRAS ISOLADAS

Após analisadas as ferramentas para extrair os parâmetros relevantes do sinal de voz, passa-se à etapa seguinte onde tais parâmetros devem ser comparados a padrões de referência para completar o processo de reconhecimento.

O primeiro ponto a ser apresentado é o método para determinação automática dos limites da palavra. Duas estratégias para o reconhecimento de palavras isoladas serão apresentadas neste capítulo. Além dos métodos de comparação, também serão apresentados os procedimentos para geração dos padrões de referência visando tornar o sistema independente do locutor.

Considerações de ordem prática com detalhes importantes para a implementação dos algoritmos serão abordadas.

## 3.1 Determinação automática dos limites da palavra

A localização do início e o final da palavra pronunciada, ou detecção dos limites, é um problema diretamente associado ao reconhecimento de palavras isoladas. Para que o processo de comparação tenha resultados eficientes é necessário que a detecção dos limites seja capaz de localizar os diversos eventos acústicos presentes na palavra.

A localização é simples se o sinal de voz for "limpo", ou seja, quando a amplitude dos sinais de voz de mais baixo nível for superior ao ruído de fundo. Entretanto, tal situação geralmente é conseguida somente em condições experimentais, tais como: local silencioso, microfones de qualidade e de captação diretiva, sistemas de cancelamento de ruído, etc..

Para uma aplicação em tempo real, um algoritmo de localização dos limites deve atender a requisitos como simplicidade e robustez, onde a detecção deve ocorrer de maneira síncrona com a pronúncia da palavra. O tempo de processamento necessário para esta tarefa deve ser pequeno se comparado ao processo de reconhecimento da palavra. O algoritmo também deve ser capaz de se adaptar às variações do ruído de fundo.

O principal parâmetro do sinal de voz utilizado na detecção dos limites é a energia do sinal. A energia pode ser calculada por um dos métodos apresentados no capítulo 2.1.1.

No caso em que o ruído de fundo é desprezível, a tarefa de detecção é trivial, bastando determinar um patamar de energia acima do nível de ruído e comparar a energia do sinal com este patamar. O sinal é então classificado como silêncio ou sinal de voz se estiver, respectivamente, abaixo ou acima deste nível.

O início da palavra é considerado no primeiro ponto onde a energia ultrapassa o patamar especificado. Para a detecção do fim da palavra deve ser considerado que durante a pronúncia de certas palavras ocorrem períodos de silêncio. Por exemplo, quando no meio de uma palavra existem consoantes do tipo plosivas, como /p/ e /t/, estas são precedidas de um período onde nenhum som é emitido. Portanto, é necessário especificar um período mínimo de silêncio entre duas palavras que seja maior que a duração do silêncio durante a pronúncia das mesmas. Na prática um intervalo de 150 ms é aceitável, porém, se a pronúncia for muito lenta, este valor deve ser aumentado.

Quando a medida de energia não é suficiente para separar sons fricativos, como /s/ e /f/, do ruído de fundo a medida de cruzamentos por zero apresenta bons resultados. Rabiner [25] apresenta um algoritmo que utiliza a energia e a taxa de cruzamento por zero. Neste algoritmo a primeira aproximação para os limites é feita através da energia. Dois patamares de energia são determinados e a detecção ocorre quando os dois níveis são ultrapassados, conforme ilustra a figura 3.1.

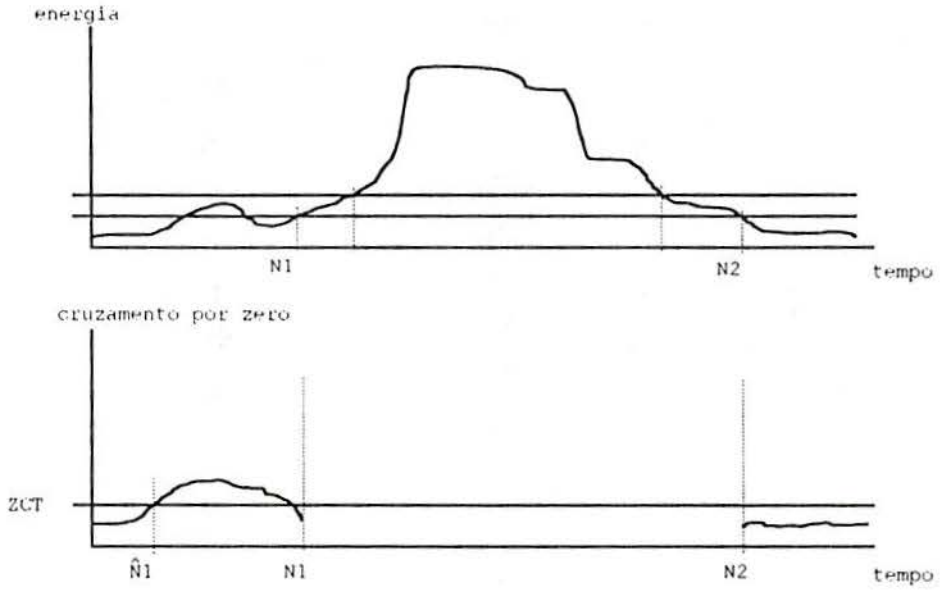


Figura 3.1 Exemplo de detecção dos limites da palavra pela energia e taxa de cruzamento por zero.

A partir dos pontos N1 e N2, detectados pela energia, é analisada a taxa de cruzamento por zero. Para o ponto inicial é feita uma pesquisa para trás (no tempo). Se a taxa de cruzamento por zero excede um limite pré-determinado (ZCT: *zero crossing threshold*) três ou mais vezes, o início da palavra é escolhido no instante em que a taxa de cruzamento por zero ultrapassa o limite ZCT, no instante $\hat{N}1$. Caso contrário, N1 é escolhido. Procedimento semelhante é executado para o fim da palavra. Na figura 3.1 foram escolhidos os limites $\hat{N}1$ e N2.

Os métodos acima descritos são classificados como métodos explícitos de detecção de limites, ou seja, a detecção precede e é independente das fases de comparação e decisão do reconhecedor. Num método implícito, a detecção dos limites é determinado somente pelas etapas de reconhecimento, isto é, não existe um estágio separado para a detecção dos limites. Neste caso, a não restrição dos pontos inicial e final permitem um nível de liberdade muito grande ao reconhecedor o que resulta em uma baixa taxa de reconhecimento [10]. Lamel [10] propõe um método híbrido no qual um conjunto de diferentes pares de estimativas para os pontos inicial e final é fornecido à etapa de reconhecimento.

Neste trabalho foi utilizado um método explícito conforme apresentado a seguir.

A detecção dos limites foi realizada a partir de uma estimativa para a energia. Esta estimativa foi obtida a partir da autocorrelação R(0) utilizada no cálculo dos coeficientes LPC. Este cálculo é equivalente à equação (2.2). Considerando a implementação em um processador de 16 bits, onde as operações matemáticas são executadas em ponto fixo, a estimativa de energia foi o fator de normalização aplicado à matriz de autocorrelação. Como este valor já foi obtido no cálculo dos parâmetros do sinal de voz, não implicou em nenhum gasto adicional de processamento. Além disso, a energia foi calculada após o filtro de pré-ênfase, o que aumenta a amplitude dos sons fricativos facilitando a sua detecção em relação ao ruído de fundo.

Para a classificação de cada segmento do sinal foi escolhido um patamar de energia de 8 vezes o ruído de fundo (PS: Patamar de Silêncio). O ruído de fundo é ajustado a cada 2 segundos como sendo a mínima estimativa de energia medida obtida neste período. Quanto à duração do sinal, foi considerado um intervalo mínimo de silêncio entre as palavras de 200 ms (Tint), sendo que a duração mínima de um sinal, para que este seja considerado como uma palavra (ou parte de uma palavra), é de 50 ms (Tmin).

Os algoritmos para a detecção do início e do fim da palavra pronunciada são mostrados respectivamente nas figuras 3.2 e 3.3.

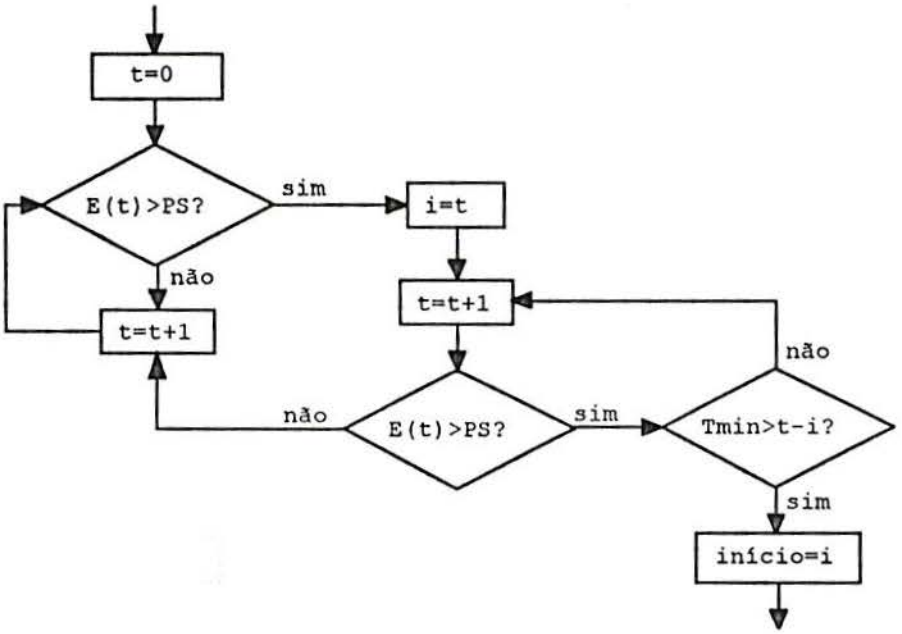


Figura 3.2 Fluxograma para determinação do início da palavra baseado na energia.

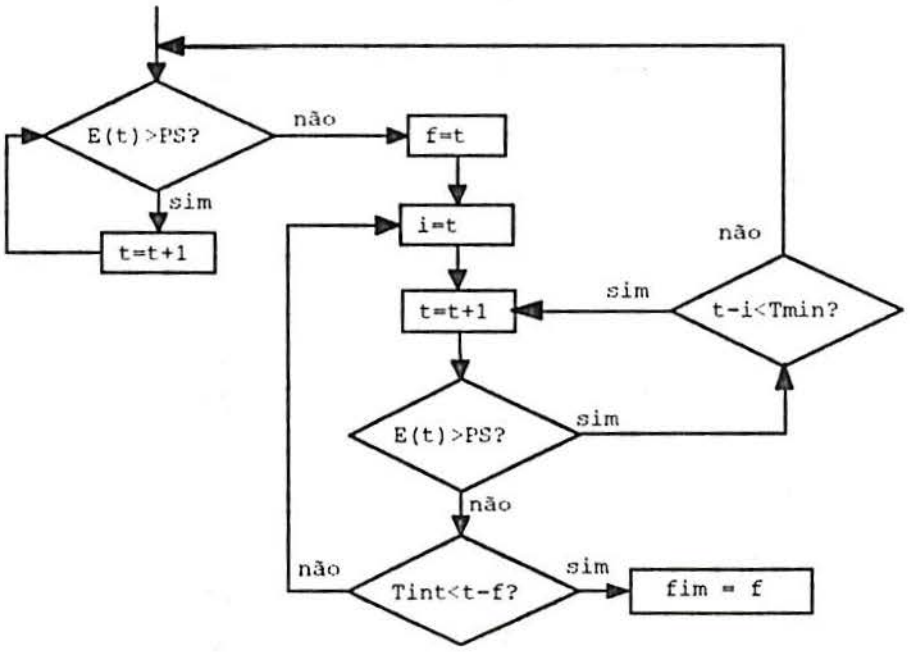


Figura 3.3 Fluxograma para determinação do fim da palavra baseado na energia.

Nos algoritmos acima E(t) é a estimativa de energia para o segmento t, PS é o patamar de energia abaixo do qual o segmento é considerado silêncio, Tmin é a duração mínima que um

trecho de fala deve ter, e Tint é o intervalo mínimo entre duas palavras. Nos algoritmos de detecção dos limites "i" é usada apenas como uma variável auxiliar para a verificação de Tmin.

## 3.2 Algoritmo DTW - *Dynamic Time Warping*

O principal objetivo do método DTW é possibilitar que a comparação entre o padrão de referência e a seqüência em teste seja insensível às variações na velocidade da pronúncia dos diversos segmentos que compõe uma palavra. Como já foi dito, a velocidade com que a palavra é pronunciada apresenta variações mesmo quando esta é pronunciada duas vezes pelo mesmo locutor. Resumidamente, o processo DTW provoca o alinhamento temporal dos dois padrões a serem comparados, de modo a minimizar a medida de distorção entre eles. Ele expande e contrai de modo não linear o eixo temporal para combinar a posição dos fonemas na palavra pronunciada e na referência. A figura 3.4 mostra um exemplo de alinhamento temporal de dois padrões unidimensionais.

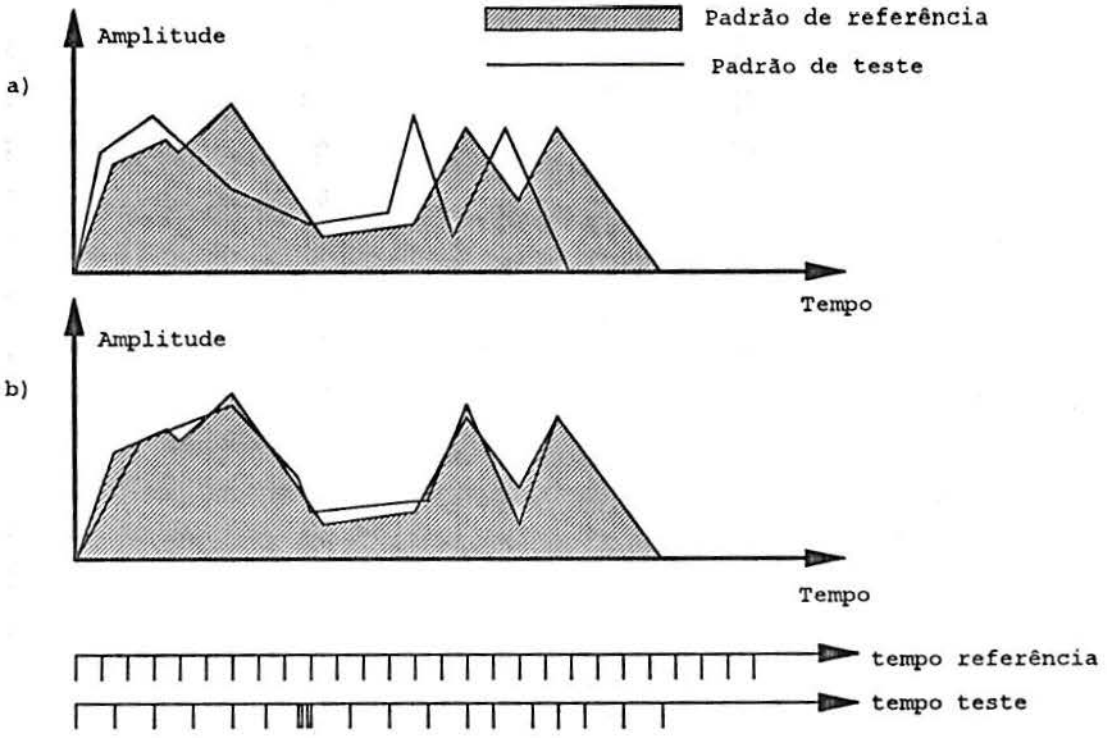


Figura 3.4 Exemplo de aplicação do algoritmo DTW. a)antes do DTW, b)após o DTW.

O algoritmo DTW fornece um mapeamento entre os índices de tempo $n$ e $m$ dos padrões de referência $R(n)$ e de teste $T(m)$ respectivamente. Tem-se então uma função de mapeamento $W$ entre $n$ e $m$ que é:

$$m = w(n). \tag{3.1}$$

O problema se resume em encontrar a função $w$ que minimiza a distorção total

$$D(R,T) = \sum_{n=1}^{N} d\left(R(n), T\left(w(n)\right)\right), \tag{3.2}$$

onde $d$ é a distância local entre o padrão de referência em $n$ e o de teste em $m=w(n)$, ou seja, $d$ é a distorção entre dois vetores, calculada por um dos métodos apresentados na seção 2.2.

Assume-se que os dois padrões $R$ e $T$ são duas seqüências de vetores que descrevem as características da voz para duas palavras, $R=\{r_1, ..., r_n, ..., r_N\}$ e $T=\{t_1, ..., t_m, ..., t_M\}$.

A função $w$ pode ser ilustrada em um plano $n$-$m$ (figura 3.5), onde o eixo vertical corresponde à escala de tempo do padrão de teste e o eixo horizontal à escala de tempo do padrão de referência.

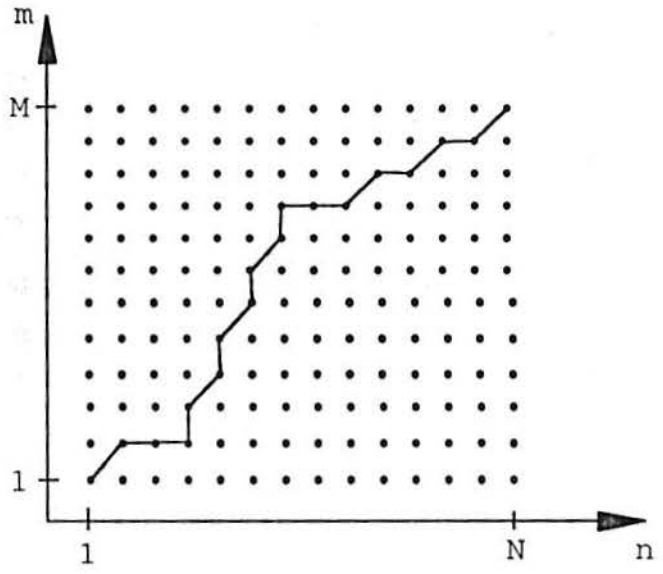


Figura 3.5 Exemplo para a função $w$.

Na figura 3.5, os pontos ligados correspondem ao caminho de mínima distorção. O processo de comparação DTW consiste na pesquisa ponto a ponto no plano (n,m) para encontrar o caminho de mínima distorção. O processo de pesquisa também é referido como programação dinâmica (DP - Dynamic Programming) e diversas restrições e limites são geralmente impostas em aplicações práticas, não sendo necessária a pesquisa completa em todo plano.

### 3.2.1 DTW para o reconhecimento de palavras isoladas

O processo de reconhecimento para palavras isoladas consiste em encontrar um padrão de referência que produza a mínima distorção com relação à palavra pronunciada em teste.

Para a aplicação prática da técnica DTW na comparação de padrões de voz, deve-se especificar certas restrições no caminho de pesquisa de modo a limitar as possibilidades de distorção da escala de tempo de acordo com as característica da fala na pronúncia de palavras isoladas.

Para implementar um algoritmo de DTW, diversas características devem ser especificadas. A primeira condição é que cada padrão seja sempre percorrido em um único sentido na escala de tempo

$$\begin{aligned} n(k-1) &\le n(k) \\ m(k-1) &\le m(k) \end{aligned} \qquad (3.3)$$

Outro ponto a ser observado são as condições de contorno referente aos limites das palavras. Tipicamente, na comparação de palavras isoladas é assumido que o início e o fim da palavra em teste esteja alinhado com o padrão de referência, ou seja:

$$\begin{aligned} w(1) &= 1 \\ w(N) &= M \end{aligned} \qquad (3.4)$$

Quando os limites da palavra não são determinados com precisão, esta restrição pode ser relaxada deixando a detecção implícita no processo[6] [28] [34].

O terceiro aspecto refere-se a restrição de continuidade, isto é, os possível tipos de caminho da função $w$ pelo plano (m,n). Tal restrição limita as possibilidades de expansão e contração impedindo que uma distorção elevada ocorra. Diversos tipos de limitações de continuidade tem sido propostas. Dentre estas pode-se citar:

1. Itakura assimétrica:

$$w(n+1)-w(n)=\begin{cases}0,1,2 & (w(n)\neq w(n-1))\\ 1,2 & (w(n)=w(n-1))\end{cases} \tag{3.5}$$

Esta condição resulta na inclinação limitada ente 1/2 e 2. A equação de pesquisa pode ser escrita da seguinte forma

$$g(n,m)=\min\left\{\begin{array}{c} g(n-1,m-1)\\ g(n-1,m-2)\\ \min\left\{\begin{array}{c} g(n-2,m-1)\\ g(n-2,m-2)\end{array}\right\}+d(n-1,m)\end{array}\right\}+d(n,m) \tag{3.6}$$

onde $g(n,m)$ é a distorção total no ponto $(n,m)$ para o caminho de menor distorção no plano $n$-$m$, e $d(n,m)$ é a distância local no ponto $(n,m)$.

Este algoritmo permite que pontos do padrão de teste sejam desprezados, ou seja, vetores do padrão de teste podem ser perdidos quando a escala de tempo deste padrão é comprimida. O alinhamento temporal é atingido pela compressão e expansão da escala de tempo do padrão de teste. Graficamente a equação pode ser observada na figura 3.6a.

2. Sakoe e Chiba simétrica

$$m(k)-m(k-1)\leq 1 \; e \; n(k)-n(k-1)\leq 1 \tag{3.7}$$

cuja equação é

$$g(n,m) = \min \begin{Bmatrix} g(n-1,m-2) + 2d(n,m-1) + d(n,m) \\ g(n-1,m-1) + 2d(n,m) \\ g(n-2,m-1) + 2d(n-1,m) + d(n,m) \end{Bmatrix} \qquad (3.8)$$

com as mesmas restrições de inclinação de 1. Os pesos neste algoritmo foram escolhidos para compensar a diferença entre diferentes caminhos. Caso contrário, o caminho diagonal seria favorecido pelo menor número de distorções computadas. Por este método, nenhum ponto é desprezado e o alinhamento é feito somente pela compressão do eixo temporal dos padrões de teste e referência (figura 3.6b).

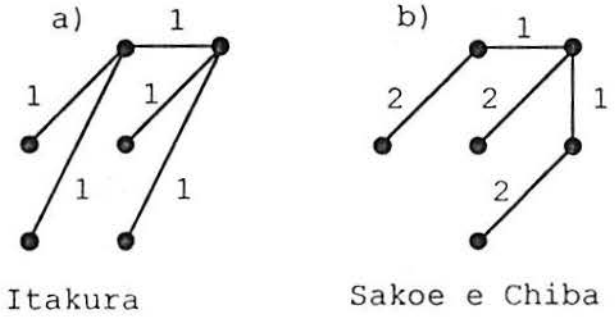


Figura 3.6 Exemplos de algoritmos de programação dinâmica empregados no método de comparação de padrões DTW.

Nos algoritmos acima a distância total para o melhor caminho é g(N,M). Existem diversas possibilidades de restrições locais que podem ser utilizadas além das apresentadas acima [2] [5] [21] [34].

Considerando a limitação de inclinação 1/2 e 2 e a condição dos limites coincidentes, a região possível no plano $(n,m)$ se reduz a da figura 3.7, onde a parte hachurada corresponde à região permitida.

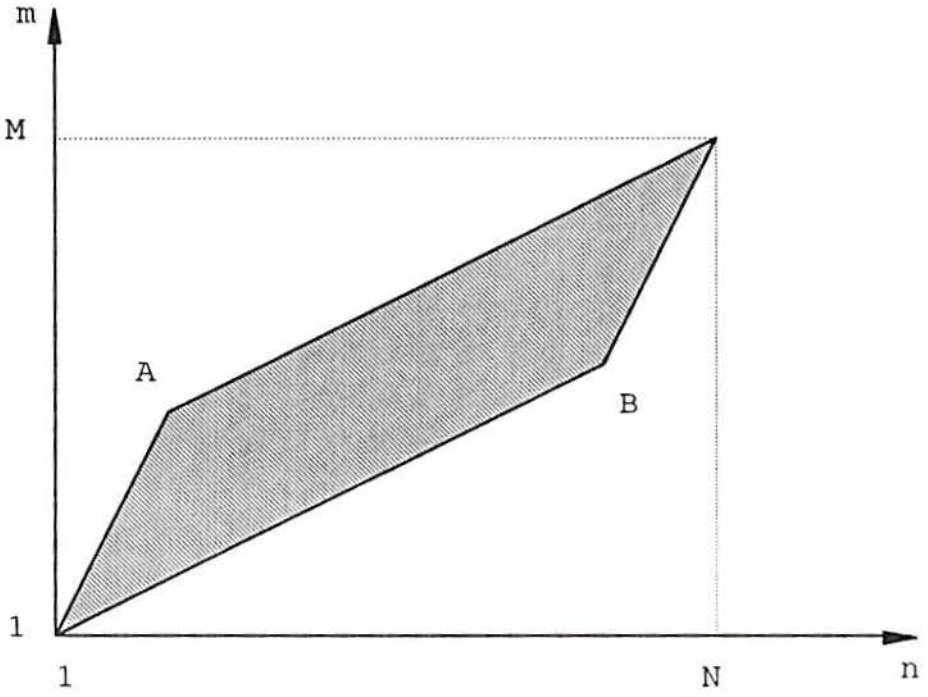


Figura 3.7 Região de pesquisa.

Os vértices do paralelogramo (pontos A e B da figura 3.5) são obtidos pela intersecção das seguintes retas

Ponto A: $$\left[m-1=2(n-1)\right]\cap\left[m-M=(n-N)/2\right] \tag{3.9a}$$

Ponto B: $$\left[m-1=\frac{1}{2}(n-1)\right]\cap\left[m-M=2(n-N)\right] \tag{3.9b}$$

### 3.2.2 Geração dos padrões de referência para DTW

Para implementar um reconhecedor de palavras isoladas independente do locutor, as variações de diversos locutores ao pronunciar a mesma palavra devem ser consideradas. Como neste tipo de algoritmo o padrão de referência corresponde a uma seqüência de parâmetros

que descreve a pronúncia de uma determinada palavra, não há como embutir em um padrão de referência as diversas possibilidades de variação de tais parâmetros. A solução para o problema é obter, não apenas um, mas múltiplos padrões para uma mesma palavra que caracterizem a variabilidade das características entre diferentes locutores. O procedimento neste caso é associar padrões da mesma palavra em grupos de tal modo que a distorção entre as palavras dentro de um mesmo grupo seja pequena. Cada grupo ou *cluster* pode então ser representado por um único padrão de referência. Neste caso, cada palavra necessita de um conjunto de padrões de referência para ser representada.

Três exemplos de técnicas para a escolha de padrões são [14]:

1. *Chainmap*

É uma técnica muito simples e consiste em reordenar os diversos padrões amostradas para uma determinada palavra. Considerando N padrões para uma determinada palavra, o processo começa com a escolha arbitrária de um dos N padrões. A partir deste padrão, os próximos são ordenados segundo o critério de mínima distorção. Ou seja, sendo $x_1$ o primeiro padrão, $x_2$ será o que apresentar a menor distorção com relação a $x_1$, $x_3$ será o de menor distorção com relação a $x_2$ e assim sucessivamente até que todos os padrões sejam ordenados formando a seqüência

$$x_1 \le x_2 \le x_3 \le \cdots \le x_N . \tag{3.10}$$

Associa-se a cada membro da seqüência a respectiva distorção pelo método DTW

$$D_k = D(x_k, x_{k+1}) \qquad 1 \le k \le N-1 . \tag{3.11}$$

onde $D_k$ é obtido pela equação (3.2).

Como resultado, os pontos onde a seqüência $D_k$ apresentar grandes picos correspondem aos limites dos grupos com características semelhantes. Como referência é escolhido um padrão de cada grupo. O procedimento pode ser sensível à escolha do ponto inicial. Neste caso, diversas tentativas podem ser feitas até encontrar um resultado satisfatório.

2. Método baseado no compartilhamento de padrões próximos

Neste método, os padrões de referência para uma determinada palavra são obtidos a partir de um conjunto de $N$ padrões de teste ($x_1$, $x_2$, ..., $x_N$) para esta palavra. Utilizando o método DTW, são encontrados, para cada um dos N padrões em teste, o conjunto dos k padrões de teste com a menor distorção. Obtém-se então a matriz:

$$L = \begin{bmatrix} x_1 & x_{1[1]} & x_{1[2]} & \cdots & x_{1[k]} \\ x_2 & x_{2[1]} & x_{2[2]} & & \\ x_3 & & & & \\ \vdots & & & & \\ x_N & x_{N[1]} & x_{N[1]} & \cdots & x_{N[k]} \end{bmatrix} \qquad (3.12)$$

onde cada linha $R_i$ contem uma lista de k+1 padrões de teste ordenada conforme o critério de mínima distorção. A ordenação de cada linha ocorre do mesmo modo que na técnica chainmap.

Supondo que $x_i \in R_j$ e $x_j \in R_i$ e ainda que

$$\left| R_i \cap R_j \right| \geq k_s \qquad (3.13)$$

para um $K_S$ fixo. Então $x_i$ e $x_j$ compartilham pelo menos $k_S$ vizinhos, sendo então classificados num mesmo grupo. Um padrão de cada grupo formado é escolhido como padrão de referência para a palavra.

Pode ocorrer que um mesmo padrão de teste seja classificado em dois grupos diferentes, tal situação indica superposição de grupos.

3. *K-means*

Este é um processo iterativo onde, dado um conjunto de N padrões de teste $X=(x_1, x_2, ..., x_N)$ para uma determinada palavra, são escolhidos M padrões de referência dentro do conjunto $X$.

O processo consiste basicamente de três etapas: classificação das palavras, cálculo dos centros de cada grupo de palavras, e teste de convergência.

Iniciando com o número de grupos desejados $M$, são escolhidos de maneira arbitrária $M$ padrões para serem o centro de cada grupo ou $C_i$.

A classificação é feita de acordo com o critério de mínima distorção

$$x_j \in C_i \ \ se \ \ D(x_j, x_i) \leq D(x_j, x_k) \ \ 1 \leq k \leq M \ \ e \ \ 1 \leq i \leq M \qquad (3.14)$$

onde a distorção $D$ é calculada pela equação (3.2), $x_i$ e $x_k$ são os centros dos grupos e $x_j$ é um padrão de teste.

Após todos os $N$ padrões de teste estarem associados aos respectivos grupos, os novos centros são calculados pelo critério do mínimo máximo. Ou seja:

$$x_i = x_j \ \ se \ max\{D(x_j, x_k)\}, \ x_j, x_k \in C_i \qquad (3.15)$$

for minimizado dentro da célula $C_i$, ou seja, calcula-se a distorção, pela equação (3.2), de cada padrão pertencente a um determinado grupo em relação a todos os outros padrões do mesmo grupo. O padrão que apresentar a menor distorção máxima em relação aos outros é escolhido como o novo centro do grupo.

O teste de convergência consiste em verificar se os centros de cada grupo são os mesmos da iteração anterior. Caso contrário, o processo reinicia com a classificação de todos os padrões de teste pela equação (3.14). A convergência deste método não é garantida, podendo ficar oscilando entre duas configurações. Esta oscilação indica que a escolha de $M$ não é adequada para este conjunto de dados.

Nos três procedimentos descritos um ponto importante é o cálculo da distância entre os padrões onde o uso do DTW, dependendo do tipo de equação empregada, resulta em distâncias que não são simétricas quando os dois padrões em teste são permutados. Por isso, para garantir a simetria, calcula-se a distorção entre dois padrões de teste por [15]

$$D_{ij} = D_{ji} = \frac{D(x_i, x_j) + D(x_j, x_i)}{2} \qquad (3.16)$$

Também pode acontecer que, devido à determinação da distância pelo DTW, a relação entre a duração de dois padrões esteja fora da faixa 2:1. Neste caso, através do histograma de duração das palavras envolvidas no cálculo, escolhe-se a faixa permitida que inclua o maior número de palavras (figura 3.8), as palavras fora desta faixa são excluídas do grupo.

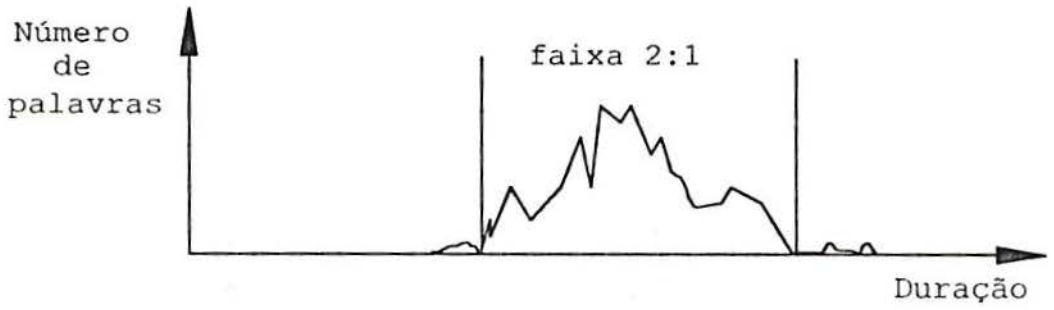


Figura 3.8 Exemplo de um histograma de duração de palavras.

## 3.3 HMM - *Hidden Markov Model*

No método de reconhecimento apresentado anteriormente (DTW), a referência para cada palavra do vocabulário era representado diretamente por uma ou mais amostras de referência que caracterizavam um padrão para a palavra. Nesta seção, será apresentada outra técnica onde a referência é representada por um modelo. Neste modelo estocástico, cada palavra é representada por um conjunto de probabilidades, ao invés de uma seqüência de parâmetros.

Considerando uma aproximação discreta, cada palavra corresponde a uma seqüência de símbolos caracterizada como uma seqüência de Markov simples ou seqüência de Markov de primeira ordem. Uma seqüência de Markov simples [36] é uma seqüência discreta cuja probabilidade do valor da seqüência no instante t depende somente do valor da seqüência no instante t-1.

Neste método de reconhecimento, cada palavra é representada por um modelo referido como "Hidden Markov Model", ou simplesmente, HMM. Um modelo HMM é um conjunto de estados conectados por transições. Dois tipos de probabilidade estão presentes no modelo, uma probabilidade associada a cada transição, e uma probabilidade associada à emissão de um

símbolo quando um estado é alcançado. O termo *hidden* (escondido) se deve ao fato do processo, ou a seqüência de transições, ser desconhecida para o observador, apenas a seqüência de símbolos é apresentada como resultado do processo.

Como o sinal de voz pode ser considerado como resultado de um processo estocástico este pode ser modelado por um outro processo estocástico, no caso, o modelo HMM. Um modelo HMM é definido pelos seguintes parâmetros:

- $\{q_i\}$ = conjunto de estados.

- A=$\{a_{ij}\}$ = conjunto de transições onde $a_{ij}$ é a probabilidade de ocorrer a transição do estado $i$ para o estado $j$

$$a_{ij} = P\left(q_j \; em \; t+1 \middle| q_i \; em \; t\right) \tag{3.17}$$

- B=$\{b_j(y)\}$ = conjunto de probabilidade de saída, ou seja, é a probabilidade de emitir o símbolo $y$ quando ocorre uma transição para o estado $j$.

$$b_j(y) = P\left(y \; em \; t \middle| q_j \; em \; t\right) \tag{3.18}$$

Os valores das probabilidades $a$ e $b$ devem satisfazer as seguintes propriedades:

$$a_{ij} \geq 0, \;\; b_j(y) \geq 0, \;\; \forall i, j, y \tag{3.19}$$

$$\sum_j a_{ij} = 1 \quad \forall i \tag{3.20}$$

$$\sum_y b_j(y) = 1 \quad \forall j \tag{3.21}$$

A figura 3.9 mostra um exemplo de um modelo HMM.

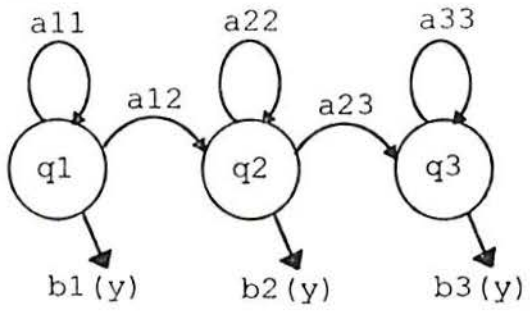


Figura 3.9 Diagrama de um modelo HMM.

O modelo de primeira ordem segue duas suposições. Primeiramente, a probabilidade da cadeia de Markov estar em um determinado estado $q_j$ no instante t+1 só depende do estado da cadeia no instante t, independente dos estados anteriores a t. Em segundo lugar, a probabilidade de um símbolo ser emitido no instante t depende somente do estado da cadeia no instante t.

Tais suposições limitam a memória do modelo reduzindo o número de parâmetros necessários para caracterizá-lo. A condição acima pode ser vista da seguinte maneira: como o estado q(t) depende do estado anterior q(t-1), então o estado q(t+1) não depende somente do estado q(t), mas também dos estados q(t-1), q(t-2), ... Entretanto, a influência de todos os estados anteriores podem ser acumuladas no estado q(t). Com esta condição, pode ser dito que o estado q(t+1) depende somente do estado q(t), que resume a propriedade do modelo de Markov de primeira ordem.

Para utilizar o HMM para o reconhecimento de voz é necessário resolver dois problemas específicos: estimar a probabilidade de uma seqüência observada, ou seja, dado um modelo e uma seqüência de símbolos observados (palavra pronunciada), estimar a probabilidade desta seqüência de símbolos ter sido gerada pelo modelo. Esta estimativa será usada para classificar a palavra pronunciada. Também é preciso estimar os parâmetros do modelo, ou seja, dado um modelo e um conjunto de observações, estimar os parâmetros do modelo para que ele forneça a máxima probabilidade de gerar as observações. Um modelo deve ser gerado para cada palavra do vocabulário.

Os procedimentos de treino e reconhecimento são realizados a partir de seqüências de símbolos. Numa seqüência $\boldsymbol{Y}$ de símbolos $y_1, y_2, \ldots, y_T$; cada $y_t$ para $1 \leq t \leq T$, pertence a um conjunto finito de símbolos obtidos pelo processo de quantização vetorial descrito no capítulo 2.3.

### 3.3.1 Avaliação do modelo

Na fase de avaliação ou classificação deseja-se determinar a qual modelo de um conjunto finito $\boldsymbol{M}=\{M_1, M_2, ..., M_N\}$ corresponde a palavra pronunciada. Neste conjunto $\boldsymbol{M}$, cada $M_i$ é o modelo para uma das palavras do vocabulário. Para a seqüência $\boldsymbol{Y}$ deve-se computar a probabilidade de cada um dos modelos $M_i$ do conjunto $\boldsymbol{M}$ gerar a seqüência. Considerando-se um vocabulário de $N$ palavras (modelos) calcula-se

$$P_i = P(Y|M_i) \quad para \ 1 \le i \le N \,. \tag{3.22}$$

A palavra então é classificada como $i$ se $P_i \geq P_j$ para $1 \leq j \leq N$.

O cálculo de $P$ em princípio envolve a soma da probabilidade de todos os caminhos possíveis na cadeia de Markov com comprimento $T$ (número de símbolos). Obviamente, o volume de cálculo é intratável, já que o número de caminhos possíveis aumenta exponencialmente em função de $T$. Felizmente, pela propriedade de Markov onde a probabilidade em um instante t depende somente da probabilidade no instante $t$-$1$, existe um método eficiente de cálculo onde a probabilidade é obtida de forma recursiva em $t$. O método referido como algoritmo de avanço (*forward algorithm*) tem a seguinte relação recursiva:

$$\alpha_j(t) = \begin{cases} 0 & t = 0 \wedge j \neq q_1 \\ 1 & t = 0 \wedge j = q_1 \\ \sum_i \alpha_i(t-1) a_{ij} b_j(y_t) & 1 \le t \le T \end{cases} \tag{3.23}$$

$\alpha_j(t)$ é a probabilidade do modelo estar no estado $j$ e ter gerado a seqüência de símbolos até o instante $t$, sendo $q_1$ o estado inicial. Deste modo, a probabilidade de estar no estado $j$, quando o símbolo $y_t$ é emitido no instante $t$, é a soma das probabilidades do modelo estar nos estados $q_i$ no instante $t$-$1$ multiplicadas pela probabilidade de ocorrer a transição de $q_i$ para o estado $q_j$ e pela probabilidade de $y_t$ na transição para o estado $q_j$.

Do mesmo modo é definido um algoritmo regressivo (*backward algorithm*) cuja formulação é a seguinte

$$\beta_i(t) = \begin{cases} 0 & t = T \wedge i \neq q_F \\ 1 & t = T \wedge i = q_F \\ \sum_j a_{ij} b_j(y_{t+1}) \beta_j(t+1) & 0 \leq t < T \end{cases} \tag{3.24}$$

$\beta_i(t)$ é a probabilidade do modelo estar no estado $i$ no instante $t$ e gerar a seqüência de símbolos observada entre $t+1$ e $T$, sendo $q_F$ o estado final.

As duas funções podem ser usadas para calcular $P$=prob($Y|M$), que é a probabilidade da seqüência observada $Y$ estar associada ao modelo $M$

$$P = \sum_i \alpha_i(t) \beta_i(t) \tag{3.25}$$

para qualquer $t$ tal que $0 \leq t \leq T$, ou seja, $P$ é o somatório das probabilidades do modelo atingir o estado $i$ tendo gerado a seqüência de símbolos até o instante $t$ multiplicado pela probabilidade de gerar a seqüência entre $t$ e $T$ partindo do estado $i$. Substituindo a equação (3.24) na equação (3.25) tem-se:

$$P = \sum_i \sum_j \alpha_i(t) a_{ij} b_j(y_{t+1}) \beta_j(t+1) \tag{3.26}$$

Assim, $P$ pode ser calculada somente pela probabilidade $\alpha$ ou pela probabilidade $\beta$, para $t$=$T$-1 e $t$=0, respectivamente.

$$P = \alpha_{q_F}(T) = \beta_{q_I}(0). \tag{3.27}$$

Calculando $P$ pela equação (3.26) ou (3.27), todas as seqüências de estados que possam gerar a seqüência $Y$ são incluídas. Pela própria definição do HMM, a seqüência de estados que gerou a seqüência $Y$ não é conhecida (hidden). Entretanto, uma alternativa para o cálculo de $P$ é encontrar a seqüência de estados que apresente a maior probabilidade de ter ocorrido para gerar a seqüência observada. Tal seqüência pode ser obtida pelo algoritmo de Viterbi:

$$v_j(t) = \begin{cases} 0 & t = 0 \wedge i \neq q_1 \\ 1 & t = 0 \wedge i = q_1 \\ \max_i v_i(t-1) a_{ij} b_j(y_t) & 1 \leq t \leq T \end{cases} \qquad (3.28)$$

$v_j(t)$ é a probabilidade do modelo estar no estado $j$ e ter gerado a seqüência de símbolos até o instante $t$, sendo considerada apenas a seqüência de estados com a maior probabilidade de ter ocorrido.

Na fase de reconhecimento tanto (3.26) ,(3.27) como (3.28) podem ser usados para classificar a palavra pronunciada dentro do vocabulário existente. Entretanto, a probabilidade obtida pelo algoritmo de Viterbi é uma aproximação para o algoritmo *forward*. Isto ocorre porque o procedimento de Viterbi encontra uma seqüência de estados ótima e na realidade um reconhecedor de palavras deve encontrar uma seqüência de símbolos ótima. A vantagem do algoritmo de Viterbi é que o volume de cálculo (somas e multiplicações) é menor que no algoritmo forward e o fato de obter uma seqüência de estados o torna útil no reconhecimento de fala contínua [12].

### 3.3.2 Treinamento do modelo

O problema de treinar o modelo, isto é, encontrar os parâmetros do modelo ($A$ e $B$) a partir de um conjunto de dados de treinamento, não apresenta uma solução tão simples quanto a simples avaliação da probabilidade associada a uma determinada seqüência. O problema é que não se conhece um método analítico para encontrar os parâmetros. O treinamento é feito por um procedimento iterativo conhecido por algoritmo *forward-backward*, também chamado de algoritmo de Baum-Welch.

A solução para o problema de estimação de parâmetros é obtida a partir das probabilidades $\alpha$ e $\beta$. A partir de uma distribuição inicial de probabilidades $a_{ij}$ e $b_j$ pode-se calcular o número esperado de transições, $\gamma_{ij}$, do estado $q_i$ para o estado $q_j$ condicionado à seqüência observada $Y$ por [16]

$$\gamma_{ij} = \frac{1}{P} \sum_{t=1}^{T} \alpha_i(t-1) a_{ij} b_j(Y_t) \beta_j(t) \qquad (3.29)$$

$\gamma_{ij}$ é a razão entre a probabilidade de ocorrer a transição do estado $q_i$ para o estado $q_j$ durante toda a seqüência $Y$ de símbolos (palavra pronunciada) e a probabilidade total $P$ do modelo gerar a palavra pronunciada.

Calcula-se então o número esperado de transições, $\gamma_i$, do estado $q_i$ para qualquer outro estado dada a seqüência $Y$

$$\gamma_i = \sum_{j=1}^{N} \gamma_{ij} = \frac{1}{P} \sum_{t=1}^{T} \alpha_i(t-1)\beta_i(t-1) \qquad (3.30)$$

onde $N$ é o número de estados do modelo.

A razão $\gamma_{ij}/\gamma_i$ é então uma estimativa da probabilidade do estado $q_j$, dado que o estado anterior foi $q_i$. Este valor é então a nova estimativa, $\bar{a}_{ij}$ , de $a_{ij}$. Tem-se então

$$\bar{a}_{ij} = \frac{\gamma_{ij}}{\gamma_j} = \frac{\sum_{t=1}^{T} \alpha_i(t-1)a_{ij}b_j(Y_t)\beta_j(t)}{\sum_{t=1}^{T} \alpha_i(t-1)\beta_i(t-1)} \qquad (3.31)$$

De maneira similar pode-se estimar $b_j(y)$ como a freqüência de ocorrência do símbolo $y$ no estado $j$ relativo a freqüência de ocorrência do estado $j$. Então [16]

$$\bar{b}_j(y) = \frac{\sum_{t:Y_t=y} \alpha_j(t)\beta_j(t)}{\sum_{t=1}^{T} \alpha_j(t)\beta_j(t)} \qquad (3.32)$$

Pela reestimativa, utilizando as equações (3.31) e (3.32), é garantido que $P$ cresce até que um ponto crítico seja alcançado, no qual uma nova estimativa permanecerá com o mesmo valor [16]. Entretanto, não é garantido que atinja um máximo global, ou seja, a probabilidade $P$ final depende da estimativa inicial dos conjuntos A e B. No processo de treinamento, as equações (3.31) e (3.32) são executadas até que a variação relativa entre duas estimativas consecutivas para os parâmetros atinja um determinado valor. Na prática, o treinamento termina quando a probabilidade $P$ do modelo estimado pára de crescer significativamente.

### 3.3.3 Considerações para implementação

O uso direto em aplicações práticas das equações apresentadas acima resulta em alguns problemas que devem ser considerados para que a implementação tenha sucesso.

O primeiro ponto a ser observado é que as soluções apresentadas tanto para o problema de classificação como de treinamento necessitam do cálculo de $\alpha_i(t)$ e $\beta_i(t)$ para $1 \leq t \leq T$. As fórmulas recursivas (3.21) e (3.22) resultam que, à medida que $T \to \infty$, $\alpha_i(t) \to 0$ e $\beta_i(t) \to 0$ de maneira exponencial. Na prática estes valores podem atingir valores abaixo da resolução mínima da máquina utilizada. Deste modo, algum método para escalonar os valores deve ser utilizado.

O escalonamento pode ser feito multiplicando $\alpha_i(t)$ e $\beta_i(t)$ por um coeficiente de escalonamento independente de $i$ de modo que mantenha estes valores dentro da faixa dinâmica da máquina. Calculando-se $\alpha_i(t)$ conforme a equação (3.23) o coeficiente de escalonamento fica [16]

$$c(t) = \left[ \sum_{i=1}^{N} \alpha_i(t) \right]^{-1} . \tag{3.33}$$

Desta forma

$$\sum_{i=1}^{N} c(t)\alpha_i(t) = 1 \quad para\ 1 \leq t \leq T \tag{3.34}$$

Calculando P pelas probabilidades *forward* e utilizando o fator de escala tem-se

$$P = C_T \sum_{i=1}^{N} \alpha_i(T) = 1 \tag{3.35}$$

onde $C_T$ é o produtório dos $c(t)$, onde conclui-se que

$$C_T = \prod_{t=1}^{T} c(t) = \frac{1}{P} \tag{3.36}$$

O produto dos fatores de escala não pode ser calculado por causa da resolução da máquina, porém é possível avaliar $P$ por

$$\log P = -\sum_{t=1}^{T} \log c(t) \tag{3.37}$$

Deste modo, a classificação da palavra é obtida pelo modelo que maximiza $\log P$.

No caso onde o escalonamento é necessário, o algoritmo de Viterbi se mostra uma alternativa bastante interessante para a classificação, pois neste caso, o cálculo pode ser feito sem a utilização do escalonamento. Como no algoritmo de Viterbi não é necessário executar o somatório, as probabilidades $a_{ij}$ e $b_j$ podem ser expressas pelos seus logaritmos. A equação (3.28) é modificada para

$$v_i(t) = \begin{cases} -\infty & t = 0 \wedge i \neq q_1 \\ 0 & t = 0 \wedge i = q_1 \\ \max\left[v_i(t-1) + \log a_{ij}\right] + \log b_j(Y_t) & 1 \leq t \leq T \end{cases} \tag{3.38}$$

No procedimento de treinamento, o escalonamento para as probabilidades $\beta_i(t)$ é feito utilizando o mesmo coeficiente de $\alpha_i(t)$, ou seja, $c(t)\beta_i(t)$. Desta forma, os coeficientes são cancelados quando as equações (3.31) e (3.32) são avaliadas. Assim tem-se:

$$\overline{a}_{ij} = \frac{\gamma_{ij}}{\gamma_j} = \frac{\sum_{t=1}^{T} C_{t-1}\alpha_i(t-1)a_{ij}b_j(Y_t)\beta_j(t)D_t}{\sum_{t=1}^{T} C_{t-1}\alpha_i(t-1)\beta_i(t-1)D_t} \tag{3.39}$$

onde

$$C_t = \prod_{\tau=1}^{t} c(\tau) \tag{3.40}$$

e

$$D_t = \prod_{\tau=t}^{T} c(\tau) \tag{3.41}$$

Tanto no numerador como no denominador de (3.39) aparece o termo $C_{t-1}D_t$ que equivale a $C_T$ conforme (3.40). Este termo pode ser retirado do somatório e cancelado, assim a equação (3.39) é equivalente a equação (3.31).

Para o caso da equação (3.32), $\beta_j(t)$ pode ser escalonado por $c(t+1)$, sendo $c(T+1)=1$, resultando em:

$$\bar{b}_j(y) = \frac{\sum_{t:Y_t=y} C_t \alpha_j(t)\beta_j(t)D_{t+1}}{\sum_{t=1}^{T} C_t \alpha_j(t)\beta_j(t)D_{t+1}} \tag{3.42}$$

O segundo aspecto referente à implementação prática é que o processo de treinamento de um modelo deve utilizar diversas amostras de uma mesma palavra para que as variações na pronúncia sejam consideradas no cálculo da probabilidades. Esta característica é importante para que o modelo adquira a característica de independência ao locutor.

A modificação no procedimento de treinamento é o seguinte: Sendo $O=\{O^1, O^2, ..., O^K\}$ o conjunto observado de seqüências, deve-se calcular a freqüência de ocorrência de cada evento (transições e ocorrência de símbolos) separadamente para cada seqüência e então somá-las. As novas fórmulas de reestimação ficam

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k} \alpha_i^k(t-1)a_{ij}b_j(O_t^k)\beta_j^k(t)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k} \alpha_i^k(t-1)\beta_i^k(t-1)} \tag{3.43}$$

e

$$\bar{b}_j(O) = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t:Y_t=y} \alpha_i^k(t)\beta_j^k(t)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T} \alpha_i^k(t)\beta_j^k(t)} \qquad (3.44)$$

onde $P_k$ é a probabilidade da seqüência $O^k$.

O terceiro problema em uma implementação prática é que o conjunto de observações é finito no processo de treinamento. Na fase de treinamento deve ser utilizada o maior número possível de observações para que o modelo cubra o máximo de variações de pronúncia. Entretanto, como este número é limitado, certamente resultará que alguns $b_j(Y_k)=0$. Supondo que posteriormente seja computada a probabilidade de uma seqüência a partir deste modelo, se somente apresentar uma ocorrência do símbolo $Y_k$, a probabilidade desta observação será zero. Este fenômeno é fatal no processo de classificação. A solução é bastante simples, e se resume a condicionar os parâmetros a um valor mínimo.

A modificação no algoritmo de Baum-Welch está especificada a seguir. Impõe-se a restrição $b_j(y_k)\geq\varepsilon$ para $1\leq j\leq N$ e $1\leq k\leq M$, onde $N$ é o número de estados do modelo e $M$ o número de símbolos existentes. Primeiro avalia-se $B$ conforme a fórmula de reestimação (3.32). Assumindo que $p$ elementos do conjunto $j$ de $B$ violaram a restrição, ou seja, $b_j(y_{k_i})<\varepsilon$ para $1\leq i\leq p$. Faça $b_j(y_{k_i})=\varepsilon$ para $1\leq i\leq p$ e reajuste os demais parâmetros de modo que a condição (3.21) permaneça válida, ou seja,

$$\bar{b}_j(y_k) = (1-p\varepsilon)\frac{b_j(y_k)}{\sum_{i=1}^{N-t} b_j(y_i)} \quad \forall k \notin \{k_i | 1 \leq i \leq p\} \qquad (3.45)$$

Valores prático para $\varepsilon$ podem variar de $10^{-3}$ a $10^{-10}$ e o valor exato não é de importância significativa [30].

Quanto às estruturas e ao número de estados de um modelo, diversas variações são possíveis. A estrutura pode variar desde um modelo sem restrições, onde transições são permitidas entre todos os estados, até modelos seriais como nos exemplos da figura 3.10, onde:

3.10a) $a_{ij}=0$ para $j<i$ e $j\geq i+2$ (transições simples),

3.10b) $a_{ij}=0$ para $j<i$ e $j\geq i+3$ (transições duplas).

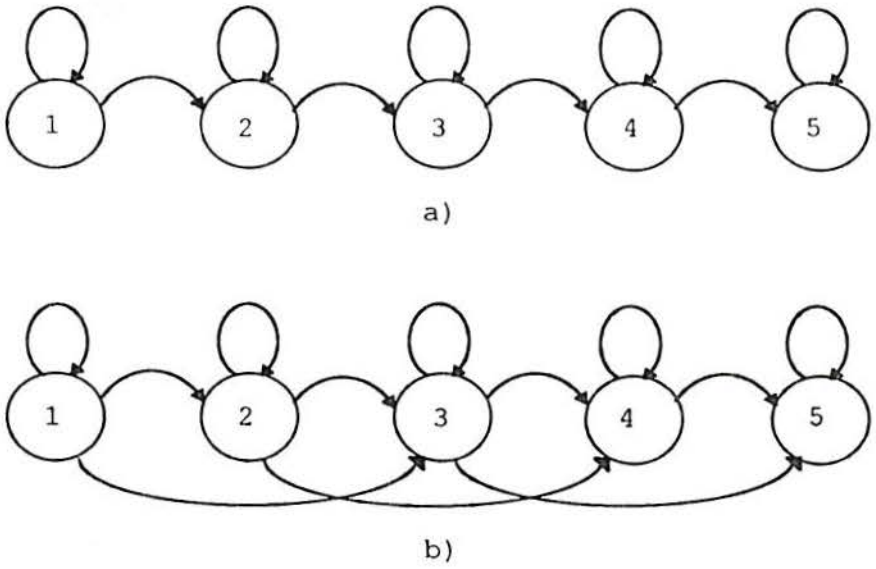


Figura 3.10 Modelos de Markov com restrição serial. a)transição simples, b)transições duplas.

Para o caso do reconhecimento de palavras isoladas, os modelos seriais são mais eficientes, pois a colocação de liberdades adicionas nas transições tendem a aumentar a probabilidade resultante para palavras incorretas.

Para o número de estados não existem regras para determinar qual o valor ideal para cada palavra do vocabulário. Experimentos realizados por Rabiner [30] com os dígitos de 0 a 9, utilizando o modelo 3.8b, mostram que a iteração entre o número de estados e a taxa de erro é bastante complexa, não demonstrando uma relação entre as características da palavra, por exemplo, o número de sílabas, e o tamanho do modelo. Também mostrou que para este caso não há vantagem em utilizar modelos com mais de 5 ou 6 estados.

# 4 IMPLEMENTAÇÃO E RESULTADOS

## 4.1 O sistema de reconhecimento em tempo real

Neste trabalho foi implementado um sistema de reconhecimento de voz em tempo real, independente do locutor, para palavras isoladas, utilizando um sistema de processamento digital de sinais baseado no processador de sinais TMS320C25, que está descrito no anexo A, e um microcomputador PC. As figuras 4.1 a 4.3 apresentam o diagrama em blocos do sistema implementado. Os detalhes da implementação de cada bloco são apresentadas nas seções seguintes. Todos os programas deste sistema, tanto os que rodam no microprocessador TMS320C25, quanto os que rodam no PC foram desenvolvidos no laboratório. O TMS320C25 executa os procedimentos relativos ao processamento digital do sinal de voz e foi programado em assembler. O PC realiza as funções de geração dos modelos que representam as palavras do vocabulário e execução dos algoritmos de reconhecimento em tempo real.

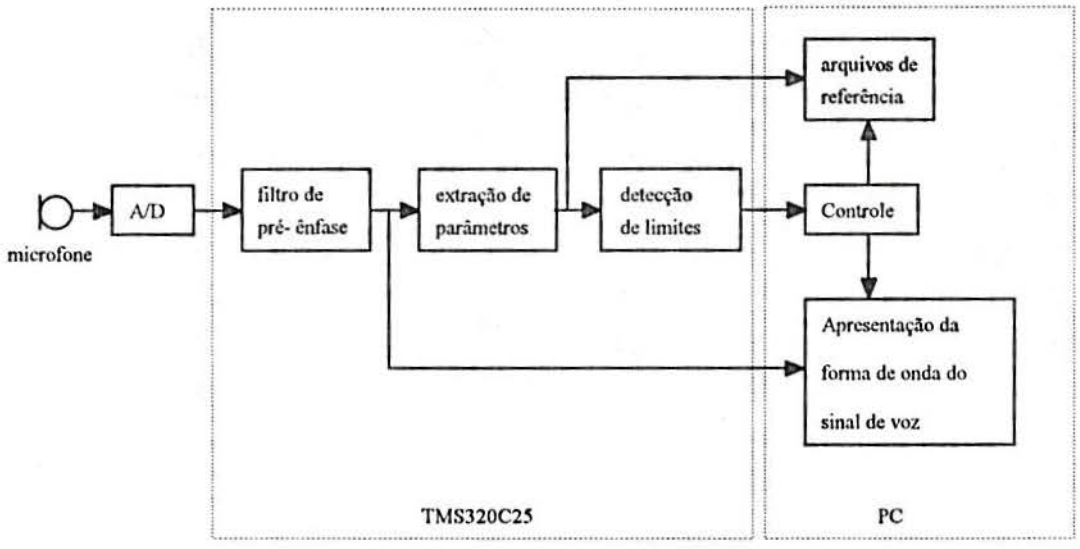


Figura 4.1 Diagrama em blocos da fase de aquisição de dados.

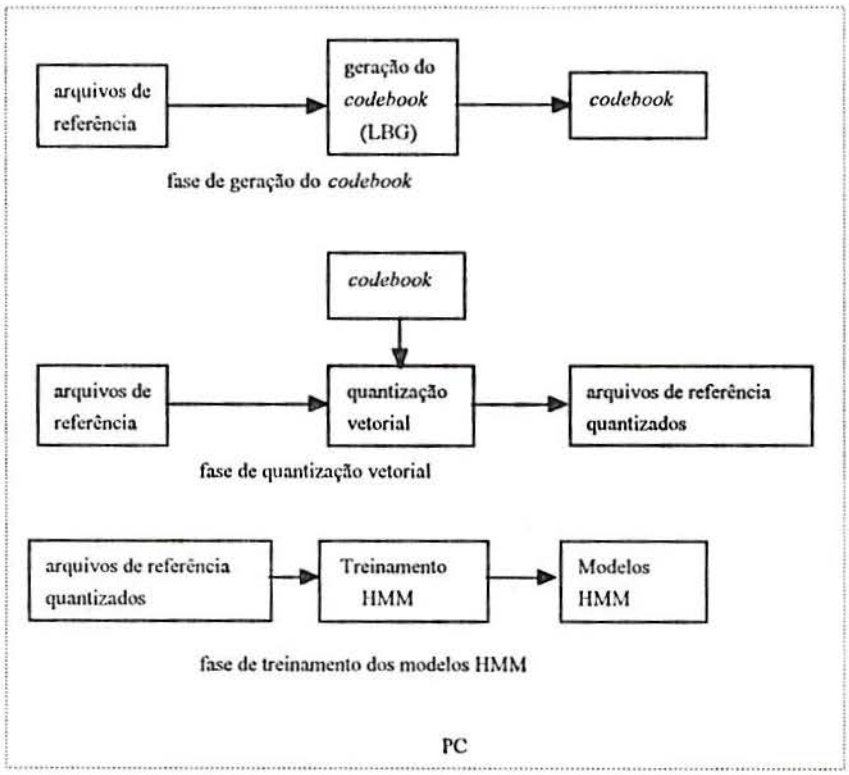


Figura 4.2 Diagrama em blocos das etapas de geração dos *codebooks* e treinamento HMM.

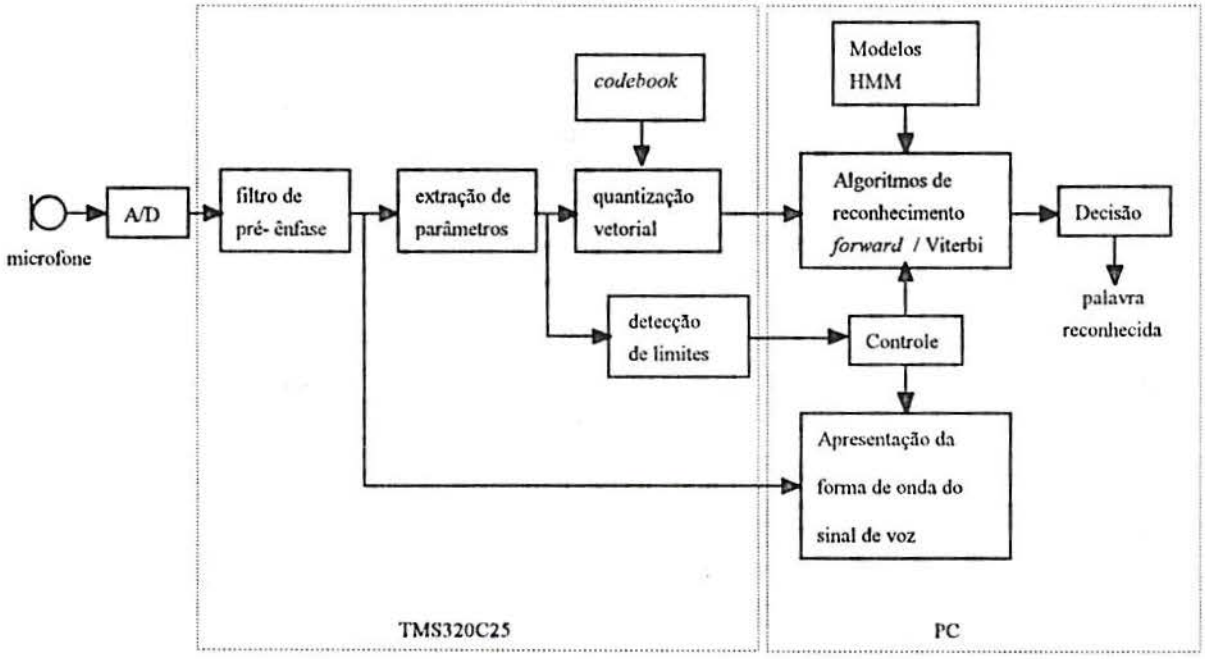


Figura 4.3 Diagrama em blocos da fase de reconhecimento da palavra em tempo real.

## 4.2 Pré-processamento do sinal

Nos experimentos práticos realizados, a voz foi captada por um microfone com resposta em frequência de 50 Hz a 15 kHz, sendo sinal gerado amplificado para se adequar aos níveis de tensão do conversor analógico digital utilizado. O sinal de voz foi limitado em freqüência de 280 Hz a 3300 Hz, semelhante à banda utilizada em sistemas telefônicos, e digitalizado com 14 bits a uma freqüência de amostragem de 8 kHz .

O sinal amostrado passa então por um filtro digital de pré-ênfase que aumenta a amplitude do sinal de voz nas freqüências mais altas. Este filtro foi implementado no TMS320C25 (ver figura 4.1) e tem a função de transferência do tipo $H(z)=1-az^{-1}$, com $a=0.95$, (figura 4.4) resultando na equação $\tilde{s}[n]=s[n]-as[n-1]$. Este filtro torna o espectro do sinal de voz mais plano diminuindo as diferenças de amplitude entre as baixas e altas freqüências. O filtro de pré-ênfase é de extrema importância para a estabilidade computacional quando é utilizada aritmética de precisão finita. No sistema de processamento de sinais utilizado, os parâmetros foram quantizados com 16 bits; neste caso, o filtro de pré-ênfase mostrou-se indispensável. Sem o filtro, o cálculo das autocorrelações pela equação (2.27) atinge valores elevados que não são adequadamente representados com o número de bits utilizados. Os erros se propagam nos cálculos subseqüentes apresentado resultados finais incorretos.

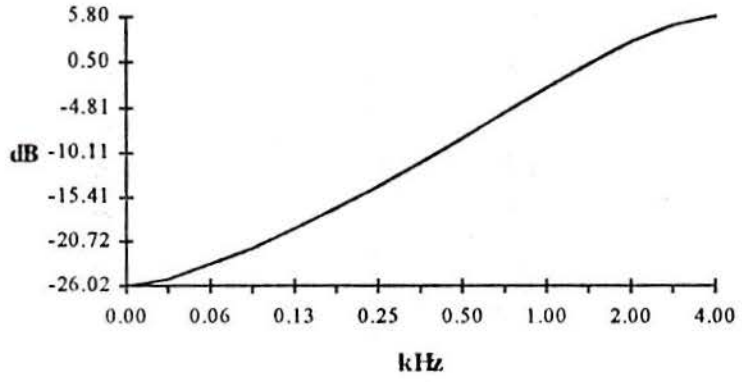


Figura 4.4 Resposta em freqüência do filtro de pré-ênfase.

Uma alternativa para o filtro de pré-ênfase, é a utilização de um filtro analógico com a mesma característica. Este procedimento não apresenta vantagem significativa com relação ao custo computacional, pois o tempo de processamento gasto com um filtro deste tipo é praticamente desprezível, quando comparado ao tempo necessário para a extração de

parâmetros. Entretanto, a utilização do filtro analógico aumenta a faixa dinâmica de utilização do conversor A/D, pois o pico de amplitude do sinal de voz é reduzido. Foi observado que ocorrendo a saturação do conversor o reconhecimento fica bastante comprometido.

## 4.3 Extração dos parâmetros e detecção dos limites

A análise do sinal foi feita em um intervalo de 20 ms, o que equivale a 160 amostras do sinal, sendo aplicada a janela de Hanning [23] para minimizar as distorções causadas pela descontinuidade nos extremos da janela. Os parâmetros do sinal de voz foram extraídos a cada 10 ms (80 amostras).

A análise LPC foi feita pelo método da autocorrelação utilizando o algoritmo de Durbin (2.30). Conforme apresentado na seção 2.1.5.2, este método garante que os parâmetros obtidos correspondem aos coeficientes de um filtro estável. Foram calculados 10 coeficientes LPC e a partir destes, foram derivados os coeficientes cepstrais conforme a equação (2.41). Nos períodos de silêncio, durante a pronúncia de uma palavra, o vetor de parâmetros com os coeficientes cepstrais foi substituído por um vetor nulo. Isto foi feito pois, neste caso, os parâmetros correspondem à análise do ruído de fundo ao invés do sinal de voz. Além disso, esse procedimento serve para enfatizar a existência do período de silêncio, o que facilita a distinção de palavras com silêncio e sem silêncio na fase de reconhecimento.

Os programas para a execução dos algoritmos de extração de parâmetros foram implementados para execução em tempo real no TMS320C25 (ver figura 4.1). Os coeficientes foram calculados utilizando aritmética de ponto fixo [31] e o erro apresentado, quando comparado ao cálculo efetuado em ponto flutuante, raramente ultrapassava 1%. Os cálculos foram feitos com resolução de 16 bits e os erros devidos a este arredondamento podem resultar em um filtro instável, entretanto, diversos testes foram realizados com sinais de voz e não foi observada tal situação (quando utilizado o filtro de pré-ênfase). Por este motivo, não foi verificada a convergência do filtro resultante para a implementação em tempo real. Além disso, a ocorrência de um eventual erro nos cálculos dos parâmetros.

O uso dos coeficientes cepstrais permite o cálculo da distância entre vetores através do erro quadrático médio, que é simples de ser implementado e envolve poucas operações. Esta facilidade é bastante útil para o processamento em tempo real. Nesta implementação, foi utilizado o erro quadrático semelhante a equação (2.44), onde o termo 1/N não foi considerado. Isto não afeta os resultados pois N é constante em todas as etapas de cálculo.

O primeiro passo para a implementação do sistema de reconhecimento foi a aquisição dos dados para o treinamento do sistema. Nesta etapa, amostras das palavras foram coletadas para diversos locutores.

O processo de aquisição foi realizado da seguinte maneira (figura 4.1): cada palavra do vocabulário ao ser pronunciada pelo locutor teve seus limites detectados automaticamente no TMS320C25 pelos algoritmos apresentados na seção 3.1. A forma de onda da palavra foi então apresentada na tela do microcomputador, permitindo a confirmação visual da detecção dos limites. As palavras cujos limites estavam incorretos foram desconsideradas. Os parâmetros (coeficientes cepstrais) foram calculados durante o processo de aquisição de cada palavra. Estes parâmetros foram então armazenados em arquivos no PC e utilizados posteriormente como padrões de referência no processo de criação do modelo referente à palavra pronunciada. Para cada palavra pronunciada foi criado um arquivo de referência (padrão de referência) contendo os vetores com os parâmetros calculados durante a pronúncia da palavra. Cada padrão de treinamento contém um conjunto de vetores, cada vetor com 10 elementos (10 coeficientes cepstrais). O número de vetores de cada padrão é função da duração da palavra .

Todo o processo de detecção dos limites e cálculo dos parâmetros foi realizado em tempo real no sistema de processamento de sinais (anexo A). Este processo de aquisição foi utilizado para garantir que os padrões usados para a criação dos modelos de referência de cada palavra tivessem as mesmas características que os padrões de teste na fase de reconhecimento.

## 4.4 Geração dos *codebooks* e treinamento HMM

Depois de obtidos os arquivos de referência com os parâmetros de cada palavra, foi executada a fase de treinamento. Esta etapa envolve a criação dos *codebooks* e dos modelos HMM de cada palavra. Os procedimentos de treinamento são esquematizados na figura 4.2 e foram executados no PC.

Os *codebooks* foram obtidos a partir dos arquivos de referência através do algoritmo LBG com a técnica de divisão binária uniforme. Para o conjunto de arquivos referentes ao vocabulário utilizado, todos os vetores pertencentes a este conjunto de treinamento foram utilizados para a geração dos *codebooks*. O vetor de perturbação utilizado no algoritmo foi um vetor constante de 0.00001 e a condição de convergência em cada iteração foi $(D_{m-1} - D_m)/D_m = 0.01$.

Na fase de treinamento, os arquivos de referência foram quantizados utilizando os *codebooks* gerados. Após a quantização vetorial, cada palavra passou a ser representada por uma seqüência de símbolos (números), onde cada símbolo corresponde a um dos vetores do *codebook* (centróides).

A partir dos arquivos de referência quantizados, foram treinados modelos HMM para cada palavra do vocabulário usado nos experimentos realizados (ver seção 4.5). O modelo HMM utilizado em todos os testes foi o de transições simples (ver figura 3.10a) com 6 estados. Para o treinamento, cada modelo foi inicializado com uma distribuição uniforme das probabilidades $A$ e $B$. O valor mínimo para os valores de $B$ foi de $10^{-5}$. Os parâmetros dos modelos foram estimados recursivamente pelas equações (3.43), (3.44) e (3.45), até que a média geométrica das probabilidades de cada padrão de treinamento convergisse para uma variação relativa de 0.01 entre duas iterações sucessivas.

## 4.5 Experimentos realizados

A fase de reconhecimento foi implementada conforme apresentado na figura 4.3. As diversas etapas desta fase (filtro de pré-ênfase, extração de parâmetros, detecção dos limites, quantização vetorial, algoritmo de reconhecimento e decisão) são executadas em tempo real. Apenas 200 ms após pronunciada a palavra (tempo mínimo necessário para separar duas palavras) o sistema fornece a palavra do vocabulário que foi reconhecida.

A avaliação da taxa de reconhecimento ($TR$) para os experimentos realizados foi obtida conforme descrito abaixo. Após a pronúncia de uma palavra, o sistema fornece a palavra do vocabulário cujo modelo apresentou a maior probabilidade e também a forma de onda da palavra em teste. A partir da forma de onda foi verificada a detecção dos limites. Para a medida da taxa de reconhecimento somente foram consideradas as palavras cujos limites foram detectados corretamente. A taxa de reconhecimento, expressa em percentual, foi calculada por:

$$TR = \frac{Número\ \ de\ acertos}{Número\ \ de\ total\ de\ palavras\ pronunciadas} \times 100 \qquad (4.1)$$

Todos os teste para a determinação da taxa de reconhecimento foram realizados em tempo real. Nestes testes, não foram utilizados os arquivos com as referências obtidos na fase de treinamento.

As etapas de extração de parâmetros e de quantização vetorial são realizadas pelo TMS320C25. Para a quantização vetorial é utilizado o *codebook* obtido na fase de treinamento. Na fase de reconhecimento, à medida que os vetores de parâmetros são calculados e quantizados, é transferido para o PC o símbolo que corresponde ao centróide obtido.

O método de reconhecimento utilizado na implementação em tempo real foi o HMM. Os algoritmos *forward* e Viterbi foram executados no microcomputador PC, utilizando aritmética de ponto flutuante com dupla precisão, cuja representação mínima para as variáveis do processo foi de $3.4 \times 10^{-4932}$, não sendo necessário o escalonamento.

Para a implementação em tempo real foi utilizado o método baseado nos modelos HMM pois este é bem mais eficiente em termos computacionais que o método DTW. Por exemplo, considerando uma palavra cuja pronúncia tenha a duração de 0.5 segundos com os parâmetros calculados conforme já descrito neste capítulo. Um modelo HMM de transições simples com 6 estados e codebook de 256 vetores necessita de aproximadamente 1550 operações de multiplicação, 550 operações de soma 256 comparações (determinação do mínimo entre dois números), enquanto que o método DTW (Itakura), considerando o padrão de referência também com duração de 0.5 segundos, necessita de aproximadamente 8500 operações de multiplicação e 10000 operações de soma e cerca de 2500 comparações.

As diversas medidas realizadas nos experimentos descritos nas seções seguintes são apresentadas em detalhes no anexo B.

### 4.5.1 Reconhecimento dependente do locutor (47 palavras)

Os primeiros experimentos realizados empregaram um vocabulário de 47 palavras. O vocabulário utilizado foi o seguinte:

| | | | | | |
|---|---|---|---|---|---|
| *um* | *dois* | *três* | *quatro* | *cinco* | *seis* |
| *sete* | *oito* | *nove* | *zero* | *abrir* | *ajuda* |
| *alterar* | *arquivo* | *cancelar* | *cascata* | *conteúdo* | *copiar* |
| *editar* | *excluir* | *executar* | *exibir* | *fechar* | *fim* |
| *grupo* | *imprimir* | *ir* | *item* | *janela* | *lado* |
| *localizar* | *marcar* | *maximizar* | *minimizar* | *mover* | *não* |
| *novo* | *ok* | *opções* | *organizar* | *para* | *procurar* |
| *propriedades* | *sair* | *salvar* | *sim* | *voltar* | |

Este vocabulário foi escolhido objetivando utilização como interface com um microcomputador.

No primeiro experimento, os modelos para as 47 palavras apresentadas acima foram gerados a partir de um conjunto de padrões obtidos de um único locutor. Para a obtenção do conjunto de padrões de treinamento, cada palavra foi pronunciada 8 vezes pelo mesmo locutor.

Todos os vetores pertencentes a este conjunto de treinamento foram utilizados para a geração dos *codebooks*. Neste caso o número de vetores foi de 18516. Três *codebooks* foram obtidos pelo algoritmo LBG com divisão binária uniforme, com tamanhos de 64, 128 e 256 vetores (centróides). Para cada palavra do vocabulário foram gerados três modelos HMM, um para cada tamanho de *codebook*. O número de vetores do *codebook* deve ser o menor possível pois a memória necessária para armazenar o modelo está diretamente associada ao tamanho do *codebook* (aproximadamente 6 kbytes para um modelo de 6 estados e *codebook* de 256 vetores), além disso o tempo de processamento para a execução da quantização vetorial também é proporcional ao tamanho do *codebook* ( para cada vetor do *codebook* é necessário um calculo de distorção, equação 2.44). Como cada vetor do *codebook* está relacionado com um som produzido durante a fala, por exemplo um fonema, o número de vetores do *codebook* foi escolhido de forma a abranger uma quantidade significativa de eventos acústicos. De acordo com os resultados obtidos pelo algoritmo forward (tabela 4.1), o codebook com 256

vetores apresentou o melhor resultado. Por isso, foi utilizado este tamanho de codebook em todos os testes subseqüentes.

Nas medidas realizadas, as 47 palavras do vocabulário foram pronunciadas em seqüência. A tabela 4.1 apresenta os resultados obtidos para o sistema treinado para um locutor.

| Algoritmo | *forward* | *forward* | *forward* | Viterbi |
|---|---|---|---|---|
| Tamanho do *codebook* | 64 | 128 | 256 | 256 |
| N° de palavras pronunciadas | 470 | 235 | 282 | 282 |
| N° de acertos | 441 | 229 | 278 | 277 |
| Taxa de reconhecimento | 93.8 | 97.4 | 98.6 | 98.2 |

Tabela 4.1 Taxa de reconhecimento para um vocabulário de 47 palavras para um locutor.

O algoritmo de reconhecimento DTW foi implementado e testado para um único locutor apenas para comparação com o método HMM. Neste caso foi utilizado o vocabulário de 47 palavras. Cada palavra foi pronunciada uma vez pelo locutor e a seqüência de parâmetros de cada palavra foi utilizada como padrão de referência. O resultado deste teste é mostrado na tabela 4.2.

| | DTW (Itakura) |
|---|---|
| N° de palavras pronunciadas | 188 |
| N° de acertos | 181 |
| *TR* | 96.3 |

Tabela 4.2 Taxa de reconhecimento para o vocabulário de 47 palavras utilizando algoritmo DTW com a equação de Itakura. Teste para um locutor.

### 4.5.2 Reconhecimento independente do locutor (47 palavras)

O segundo experimento envolveu múltiplos locutores. Na fase de treinamento, as 47 palavras do vocabulário foram pronunciadas por 7 locutores, cada palavra foi pronunciada 1 vez por cada um dos locutores. Um *codebook* de 256 níveis foi gerado a partir de todos os vetores deste conjunto de treinamento, um total de 18221 vetores. Um modelo HMM foi criado para cada palavra do vocabulário.

A taxa de reconhecimento foi avaliada para 8 locutores, sendo que 4 participaram da fase de treinamento e 4 não participaram. As tabelas 4.3 e 4.4 apresentam os resultados destes testes.

| | *forward* | Viterbi |
|---|---|---|
| N° de palavras pronunciadas | 564 | 564 |
| N° de acertos | 393 | 384 |
| *TR* | 69.7 | 68.1 |

Tabela 4.3 Taxa de reconhecimento para o vocabulário de 47 palavras utilizando 4 locutores que participaram do treinamento

| | *forward* | Viterbi |
|---|---|---|
| N° de palavras pronunciadas | 376 | 376 |
| N° de acertos | 198 | 184 |
| *TR* | 52.7 | 48.9 |

Tabela 4.4 Taxa de reconhecimento para o vocabulário de 47 palavras utilizando 4 locutores que não participaram do treinamento.

### 4.5.3 Reconhecimento independente do locutor (15 palavras)

Outro experimento foi realizado utilizando um vocabulário de 15 palavras, sendo estas:

| | | | | |
|---|---|---|---|---|
| ***um*** | ***dois*** | ***três*** | ***quatro*** | ***cinco*** |
| ***seis*** | ***sete*** | ***oito*** | ***nove*** | ***zero*** |
| ***abrir*** | ***cancelar*** | ***fechar*** | ***ok*** | ***para*** |

Para o treinamento dos modelo, cada palavra foi pronunciada 2 vezes por 20 locutores. O *codebook* com 256 níveis foi obtido a partir de todo o conjunto de treinamento, 30307 vetores. Os resultados são mostrados nas tabelas 4.5 e 4.6.

| | *forward* (treinamento c/ 20 locutores) | Viterbi (treinamento c/ 20 locutores) | *forward* (treinamento c/ 7 locutores) |
|---|---|---|---|
| N° de palavras pronunciadas | 450 | 450 | 450 |
| N° de acertos | 431 | 423 | 334 |
| *TR* | 95.8 | 94.0 | 74.2 |

Tabela 4.5 Taxa de reconhecimento para o vocabulário de 15 palavras utilizando 10 locutores que participaram do treinamento. No caso do treinamento con 7 locutores também foram utilizados os 10 locutores.

| | *forward* (treinamento c/ 20 locutores) | Viterbi (treinamento c/ 20 locutores) | *forward* (treinamento c/ 7 locutores) |
|---|---|---|---|
| N° de palavras pronunciadas | 180 | 180 | 180 |
| N° de acertos | 162 | 158 | 128 |
| *TR* | 90 | 87.8 | 71.1 |

Tabela 4.6 Taxa de reconhecimento para o vocabulário de 15 palavras utilizando 4 locutores que não participaram do treinamento.

Cada locutor pronunciou 3 vezes a seqüência de palavras para cada caso apresentado (*forward* e Viterbi para treinamento com 20 locutores e *forward* para treinamento com 7 locutores).

O sistema foi colocado em funcionamento em um experimento prático de controle de um dispositivo utilizando os modelos do experimento acima. Nesta implementação, foi definida uma seqüência pré-definida de comandos. Trata-se do controle de posição de uma válvula e, para isto, foi estabelecida uma sintaxe composta da seguinte seqüência de palavras: a palavra /abrir/, seguida de um ou dois dígitos, /zero/ a /nove/, e concluindo com a palavra /ok/. Caso a seqüência não for completada nesta ordem, o comando é ignorado devendo ser repetido do início novamente. Em qualquer instante o comando pode ser interrompido pela palavra /cancelar/. A colocação de uma seqüência rígida de comandos tornou o sistema bastante robusto, impedindo que a válvula fosse ligada com um comando indesejado, mesmo em um ambiente ruidoso e com diversas pessoas conversando no ambiente.

# 5 CONCLUSÃO

O objetivo deste trabalho foi estudar algumas técnicas de processamento de sinais voltadas para o reconhecimento de voz, mais especificamente, para o reconhecimento de palavras isoladas e independente do locutor. O estudo foi dirigido para a aplicação prática das técnicas abordadas, resultando na implementação em tempo real de um sistema de reconhecimento de voz para palavras isoladas.

Analisando primeiramente os resultados da implementação apresentada no capítulo 4, diversas considerações podem ser feitas.

No primeiro experimento, envolvendo apenas um locutor foi possível observar a influência da quantização dos parâmetros extraídos dos sinal de voz. O resultado obtido é bastante lógico: quanto maior a resolução do sistema melhor o resultado obtido. Por exemplo, um aumento de 128 para 256 vetores no tamanho do *codebook* resultou no aumento de 97.4% para 98.6% na taxa de reconhecimento. Deve ser considerado que o aumento no tamanho do *codebook* resulta no aumento no processamento e na memória, a mudança de 128 para 256 quase dobra a quantidade de memória necessária para armazenar o modelo HMM de cada palavra e também dobra o volume de processamento (na quantização vetorial). Este aumento de computacional se justifica, considerando-se que a taxa de erro caiu de 2.6% para 1.4%.

Foi observado que a fase de treinamento é muito importante em um sistema para que este melhore a taxa de reconhecimento. No experimento realizado com o vocabulário de 15 palavras a melhoria dos resultados foi significativa quando o número de palavras utilizadas para treinar cada modelo aumentou. Neste caso, ao aumentar de 7 para 40 o número de vezes que cada palavra foi pronunciada para determinar os parâmetros do modelo de cada palavra, a taxa de reconhecimento cresceu de 74.2% para 95.8%. A melhora foi também significativa para locutores que não participaram da fase de treinamento, de 71,1% para 90%. Portanto, para que um sistema atinja baixas taxas de erro é necessário que um grande número de amostras de cada palavra seja utilizado na fase de treinamento.

Nos testes realizados o desempenho do algoritmo de Viterbi foi pior que o algoritmo *forward*, porém, a diferença entre os dois foi sempre pequena, na faixa de 0.4% a 3.7%, o que demonstra que o uso do algoritmo de Viterbi se justifica, considerando o ganho computacional que este oferece.

O algoritmo DTW apresentou desempenho semelhante ao HMM com relação a taxa de reconhecimento para um locutor, porém, com um custo computacional bem maior. Para o caso de diversos locutores a situação seria bem pior, pois cada palavra teria diversos padrões de referência e a comparação da palavra em teste deve se feita com todas as referências. Uma alternativa para este problema seria a utilização da quantização vetorial. Neste caso, o cálculo da distorção entre dois vetores de parâmetros não precisaria ser realizado em tempo real e se resumiria a uma consulta em uma tabela. Também poderiam ser utilizadas técnicas de seleção onde, à medida que os cálculos fossem realizados, os padrões de referência cuja distorção fosse elevada seriam eliminados, diminuindo com isto a carga computacional.

O desempenho obtido com o sistema implementado não é o ideal devendo ser melhorado para que sua utilização tenha utilidade prática mais abrangente. Deve ser considerado que diversas limitações foram impostas para a implementação do sistema, como por exemplo a limitação da banda do sinal de voz em 3 kHz. Isto implica em uma real degradação do sinal dificultando o reconhecimento. Para esta faixa de freqüências, existe dificuldade até mesmo para o ouvido humano distinguir certos fonemas.

Com o sistema implementado foi possível observar, além do desempenho dos algoritmos, detalhes importantes que devem ser consideradas em uma implementação prática. O comportamento do locutor é um aspecto bastante importante a ser observado. Durante o processo de aquisição das amostras de voz para o treinamento dos modelos, o locutor geralmente pronuncia as palavras de maneira clara, ou seja, os fonemas são "bem" pronunciados. Esta não é a forma natural com que as pessoas costumam pronunciar as palavras. Foi possível perceber que na fase de reconhecimento os locutores falavam de maneira mais natural, sem a preocupação inicial de pronunciar as palavras com clareza. Entretanto, o locutor consegue com certa facilidade adaptar-se modificando a pronúncia de modo que o sistema consiga reconhecê-lo melhor. Foi observado que certos locutores mais cooperativos se adaptavam ao sistema, nestes casos as taxa de reconhecimento cresciam, chegando a atingir até 100%.

Um problema observado na detecção dos limites foi que diversos locutores liberavam o ar contido nos pulmões de maneira brusca ao terminar a pronúncia da palavra, isto provocava uma falha na detecção dos limites. Este problema pode estar associado ao fato de que a pronúncia de palavras isoladas não é uma forma de comunicação natural do ser humano. Para evitar este problema, quando necessário, o microfone era mantido distante da boca do locutor. Este procedimento resultava em outro problema: sons de baixa amplitude e de curta duração não eram captados sendo necessário que o locutor enfatizasse certos fonemas. Uma possível solução para este problema seria um método de detecção dos limites mais sofisticado. Neste caso, não só a energia deve ser utilizada como parâmetro de análise, mas também outras

características que permitam a identificação de um padrão que corresponda à respiração do locutor.

Uma outra abordagem para o problema que deve levar a um melhor desempenho é o reconhecimento de unidades menores da palavra. Foi observado que diversas palavras conseguiam ser identificadas apenas pela pronúncia de um trecho da palavra, por exemplo, uma sílaba. Uma sugestão para dar continuidade a este trabalho é a analise de segmentos da palavra, ou seja, tentar identificar os diversos fonemas que a compõe. Esta abordagem além de possibilitar uma melhoria pode ser estendida para aplicação em vocabulários bem maiores e também para a abordagem do reconhecimento de fala contínua. Um exemplo de como o reconhecimento de fonemas deve melhorar o desempenho é o seguinte: se ao analisar o início da palavra for possível identificar um determinado fonema com precisão, pode-se eliminar uma série de possibilidades, o que na abordagem atual não é considerado. Por exemplo, para o caso do vocabulário de 15 palavras utilizado neste trabalho, se fosse detectada a presença de um fonema /s/ no início da palavra as possibilidades se restringiriam às palavras /cinco/, /seis/ e /sete/; determinando outro fonema da palavra as possibilidades de erro começam a diminuir.

O estudo na área de reconhecimento de voz já teve um grande avanço. Ainda existem, porém, muitas restrições que impedem que a utilização da voz como interface entre o homem e a máquina faça parte do cotidiano. No futuro, sistemas capazes de reconhecer a livre conversação sem restrições, poderão ser uma realidade.

# ANEXO A - SISTEMA DE PROCESSAMENTO DE SINAIS BASEADO NO TMS320C25

## A.1 - Introdução

Nas diversas etapas do desenvolvimento deste trabalho foi utilizado um sistema de processamento digital de sinais desenvolvido no Departamento de Engenharia Elétrica da UFRGS [18], o qual foi utilizado para aquisição de amostras do sinal de voz e para execução em tempo real de diversas etapas no processo de reconhecimento de voz.

## A.2 - Descrição do Sistema

O sistema é composto por diversos módulos conforme mostrado na figura A.1. As características dos módulos que foram utilizados são apresentadas a seguir.

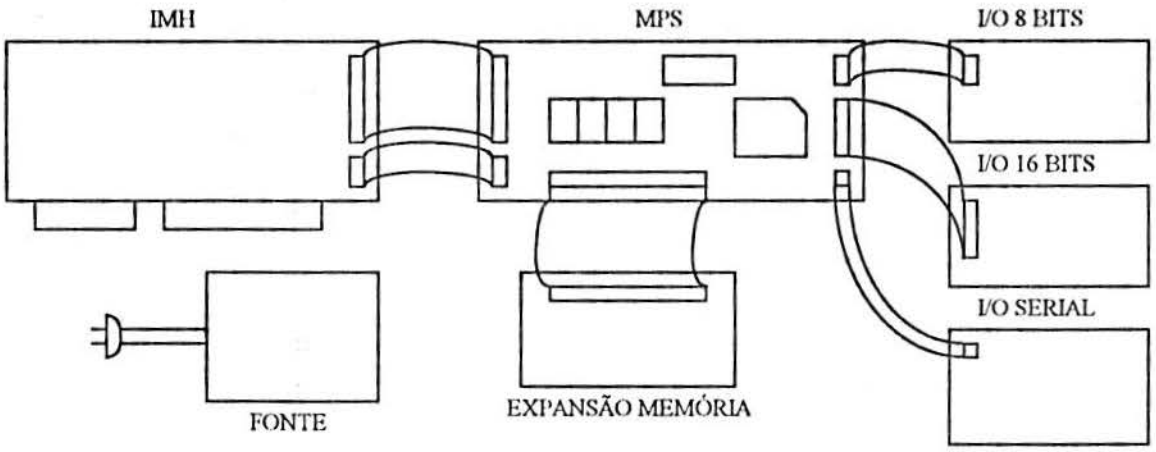


Figura A.1 Diagrama esquemático do sistema de processamento de sinais.

Módulo de Processamento de Sinais (MPS):

- Utiliza o processador TMS320C25, com clock de 40 MHz e ciclo de instrução de 100ns.
- memória local de 32 kW (permite o uso com RAM ou EPROM)
- barramento para expansão de memória
- interfaces de entrada e saída de 8 e 16 bits.
- interface serial.
- 3 temporizadores programáveis.
- conector para comunicação com placa de interface com micro hospedeiro.

Interface com Micro Hospedeiro (IMH):

- desenvolvida para microcomputadores compatíveis com o padrão PC AT.
- acesso em 16 bits.
- possibilidade de emulação dos sinais de controle do TMS320C250 pelo micro hospedeiro, ou seja, este pode colocar o TMS320C25 em estado de *hold* e controlar o barramento do módulo de processamento.
- registradores de 16 bits para comunicação entre o PC e placa.
- interrupções bidirecionais - PC gera interrupçao no módulo e vice versa.
- conector para comunicação com a placa de processamento de sinais.

Foram utilizadas duas interfaces de entrada e saída. A primeira trata-se de uma interface analógica que apresenta comunicação serial com MPS e contém 2 canais A/D e D/A que são multiplexados. Ela é composta por um conversor A/D cuja entrada pode ser multiplexada e dois conversores D/A.

Características:

- A/D de 14 bits , 80 ksamples/s (AD7871);
- 2 D/As de 8 bits (PM-7226);
- Comunicação serial.

A segunda placa de interface analógica também utiliza comunicação serial a apresenta as seguintes características:

- Conversores AD/DA com faixa dinâmica de 14 bits e linearidade de 10 bits (TLC32040);
- Filtros (capacitor chaveado) na entrada do conversor AD e na saída do conversor DA. O Filtro na entrada do AD pode ser programado por software;
- Ganhos das entradas selecionáveis (1,2 ou 4) por software;
- Taxa máxima de conversão AD ou DA de 19200 amostras/segundo;
- Possibilidade de dois canais de entrada multiplexados;
- Comunica-se com a placa de DSP pelo conector serial;
- Possui reguladores de tensão na placa para parte analógica, permitindo alimentação de 7.3 a 18V não regulada;
- Possui *buffers* de entrada e saída.

Maiores detalhes das características do processador TMS320C25 bem como do sistema utilizado podem ser obtidos nas referências [32][18].

## ANEXO B - MEDIDAS REALZADAS

Nas tabelas a seguir são apresentadas as medidas realizadas para a avaliação da taxa de reconhecimento do sistema apresentado neste trabalho.

A tabela B.1 mostra o total de erros obtidos para o reconhecimento de um conjunto de 47 palavras, dependente do locutor. Os modelos HMM foram treinados com 8 referências para cada palavra. A tabela apresenta o número de vezes que cada palavra foi pronunciada na fase de reconhecimento e o total de erros obtidos utilizando os algoritmos *forward*, com *codebooks* de 64 , 128 e 256 vetores, e Viterbi, com *codebook* de 256 vetores, e para o algoritmo DTW.

| | *forward* | *forward* | *forward* | Viterbi | DTW |
|---|---|---|---|---|---|
| Tamanho do *codebook* | 64 | 128 | 256 | 256 | ---- |
| Nº de repetições | 10 | 5 | 6 | 6 | 4 |
| palavras | *Nº de erros* | | | | |
| um | 1 | | | | |
| dois | | | | | |
| três | | | | | |
| quatro | | | | | |
| cinco | | | | | |
| seis | | | | | |
| sete | | | | | |
| oito | | | | | |
| nove | | | | | |
| zero | | | | | |
| abrir | | | | | 1 |
| ajuda | 2 | | | | |
| alterar | 2 | | | | |
| arquivo | | | | | |
| caseata | 8 | 1 | | | |
| cancelar | | | | | |
| conteúdo | | | | | |
| copiar | | | | | |
| editar | | | | | |
| excluir | | | | | |
| executar | 3 | | | | |
| exibir | 1 | | | | 2 |
| fechar | | | | 1 | |
| fim | 1 | | | 1 | 1 |
| grupo | 3 | | | | |
| imprimir | | | | 1 | |
| ir | | 2 | | 1 | 1 |
| item | | | | | |
| janela | | | | | |
| lado | | | | | |
| localizar | | | | | |
| marcar | | | 1 | | |
| maximizar | | | | | |
| minimizar | | | | | 1 |
| mover | | | | | |
| não | | | 2 | | |
| novo | | | 1 | | |
| ok | | | | | |
| opções | | | | | |
| organizar | 1 | | | | |
| para | 1 | 1 | | | |
| procurar | | | | | |
| propriedades | | | | | |
| sair | | | | | |
| salvar | 1 | | | | |
| sim | 1 | | | 1 | 1 |
| voltar | 3 | 1 | | | |
| total de erros | 29 | 6 | 4 | 5 | 7 |
| total de palavras pronunciadas | 470 | 235 | 282 | 282 | 188 |
| Taxa de erro (%) | 6.2 | 2.6 | 1.4 | 1.8 | 3.7 |

Tabela B.1 Medidas de erro para vocábulário de 47 palavras para um locutor (HMM e DTW).

A tabela B.2 mostra o total de erros obtidos com o modelo HMM para o reconhecimento de um conjunto de 47 palavras, independente do locutor. Os modelos HMM foram treinados com 7 locutores, com 7 referências para cada palavra, cada locutor forneceu uma referência por palavra. Os erros medidos estão tabelados em função dos algoritmos *forwrd* e Viterbi, ambos com *codebooks* de 256 vetores.

| Locutor | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Prticipou do treinamento | sim | | sim | | sim | | sim | | não | | não | | não | | não | |
| Algoritmo *forward*/Viterbi | *f* | V | *f* | V | *f* | V | *f* | V | *f* | V | *f* | V | *f* | V | *f* | V |
| N° de repetições | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| palavras | n° de erros | | | | | | | | | | | | | | | |
| zero | | | | 1 | | | 1 | 2 | 2 | 2 | 1 | 1 | | | | 1 |
| um | 1 | 2 | 1 | 2 | | 1 | 2 | 2 | 1 | 2 | 2 | 2 | | | 2 | 2 |
| dois | | | | 2 | | 1 | 1 | 2 | | 1 | 2 | 2 | | 2 | | 1 |
| três | | 1 | 3 | 1 | | 1 | 3 | 1 | | 1 | 2 | 2 | 2 | 1 | 2 | 1 |
| quatro | | | | | 1 | 3 | 1 | 2 | 2 | 2 | | 1 | 2 | 2 | | |
| cinco | | | | | | | 2 | 3 | | | 1 | 2 | | | 1 | 1 |
| seis | | | 1 | 1 | | | 1 | 2 | | | 2 | 2 | 2 | 2 | | |
| sete | | | 2 | 1 | | | 3 | | | 2 | 1 | 1 | | | | |
| oito | 1 | 1 | 2 | 1 | 2 | | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 1 | |
| nove | | | 3 | 3 | 3 | 2 | 3 | 3 | 1 | | 2 | 2 | 1 | 1 | 1 | 1 |
| abrir | | | 1 | 1 | 3 | 1 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | | 2 | 1 |
| ajuda | | | | | | | 2 | 1 | 1 | | | 2 | | 1 | 1 | |
| alterar | 1 | | | | | | 2 | 1 | 1 | 1 | 2 | 2 | 1 | | 1 | |
| arquivo | | | | | | | 1 | 1 | | | 1 | 2 | | 1 | | |
| cancelar | | | | 1 | 1 | | 1 | | 2 | | 2 | 2 | | 2 | 2 | 2 |
| cascata | 1 | | | 1 | | | | 1 | 1 | 2 | 1 | 2 | | | 1 | 1 |
| conteúdo | | | 1 | 1 | | | 3 | 2 | | | | 1 | | | | |
| copiar | 1 | 1 | 2 | 2 | | 1 | 1 | 1 | 2 | 2 | 2 | 1 | 2 | 1 | 1 | 2 |
| editar | 3 | 2 | 3 | 3 | 2 | 3 | 1 | | 2 | 2 | 2 | 2 | | 1 | 2 | |
| excluir | | | 1 | | | | | | 1 | 1 | 1 | 2 | | | | |
| executar | | | | 1 | 1 | | 1 | 2 | 2 | 2 | 2 | 2 | 2 | | 2 | 2 |
| exibir | | | | 2 | 2 | 2 | 1 | | 2 | 1 | 2 | 2 | | 1 | | 2 |
| fechar | 1 | 1 | 1 | 2 | 1 | | 1 | 2 | 2 | 2 | | 1 | 2 | 2 | | |
| fim | | 1 | 1 | | 2 | | 2 | 2 | 2 | 2 | 1 | 1 | | 2 | 1 | 2 |
| grupo | | | | 1 | | 1 | | | 2 | 2 | 1 | 2 | 1 | | 1 | |
| imprimir | 1 | 1 | | 1 | | | 3 | 3 | 2 | 1 | | 1 | | | 1 | 2 |
| ir | 1 | 1 | 3 | 3 | 2 | 2 | 2 | 3 | 2 | 2 | 2 | 1 | 2 | 1 | 1 | |
| item | | | 3 | 3 | 1 | | 3 | 3 | 2 | 1 | | 1 | 1 | | 2 | |
| janela | | | | | | | | 1 | | 1 | | | | 2 | | 1 |
| lado | | | | | | 1 | 2 | 3 | | 1 | 1 | 2 | 1 | | 1 | 1 |
| localizar | | | 1 | 2 | | | 1 | | 1 | | 2 | 1 | | 1 | | |
| marcar | | | 2 | 3 | 1 | 1 | 2 | | | | 1 | 2 | | | | |
| maximizar | | | | | | | 2 | 2 | 1 | 2 | 2 | | | | | |
| minimizar | | | 1 | 1 | 1 | | 1 | 1 | 2 | 2 | 2 | 2 | | 1 | 2 | 2 |
| mover | 2 | 2 | 2 | 3 | | 3 | 3 | 3 | 2 | 2 | 1 | 1 | | 1 | 2 | 2 |
| não | | 1 | 3 | 1 | | 1 | 3 | 3 | 2 | 2 | 2 | 2 | | 1 | 1 | |
| novo | | | 3 | 3 | | | 1 | 2 | 1 | | 1 | 2 | 1 | 1 | 2 | 2 |
| ok | 1 | 3 | 3 | 3 | | | 3 | 3 | | | 2 | 1 | 2 | 2 | 1 | 1 |
| opções | | | 2 | 1 | | | | | | | | | | | | |
| organizar | 1 | 1 | 1 | 2 | 1 | 1 | 3 | 2 | 1 | 2 | | 1 | | | 2 | 2 |
| para | | 1 | 2 | 1 | | | | 1 | 2 | 2 | | | | 2 | 2 | 1 |
| procurar | | | 1 | | | 1 | | 2 | 2 | 2 | | 1 | 1 | | 1 | 1 |
| propriedades | | | | | | | | 2 | | | | | | | | |
| sair | | 1 | 2 | 1 | | | 3 | 3 | | 1 | 2 | 2 | 1 | 1 | 2 | 2 |
| salvar | | | | | | | 2 | 3 | 1 | 2 | | 1 | | | | |
| sim | 1 | | 1 | 1 | | | 1 | 2 | 1 | | 1 | | 1 | 2 | 2 | 2 |
| voltar | | | 3 | 2 | 1 | | 3 | 2 | 2 | 1 | 1 | 1 | | | 1 | |
| total de erros | 16 | 20 | 56 | 58 | 25 | 26 | 74 | 76 | 54 | 55 | 53 | 64 | 27 | 35 | 44 | 38 |
| Taxa de erro (%) | 11.3 | 14.2 | 39.7 | 41.1 | 17.7 | 18.4 | 52.5 | 53.9 | 57.4 | 58.5 | 56.4 | 68.1 | 28.7 | 37.2 | 46.8 | 40.4 |

Tabela B.2 Medidas de erro para o vocábulário de 47 palavras independente do locutor.

A tabela B.3 mostra o total de erros obtidos com o modelo HMM para o reconhecimento de um conjunto de 15 palavras, independente do locutor. Os erros medidos estão tabelados em função do número de palavras usadas no treinamento de cada modelo, para os algoritmos *forwrd (40 e 7 palavras)* e Viterbi(40 palavras), todos com *codebooks* de 256 vetores.

| locutor | 1 | | | 2 | | | 4 | | | 3 | | | 12 | | | 5 | | | 6 | | | 9 | | | 10 | | | 11 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| N° de palavras no treino | 40 | 40 | 7 | 40 | 40 | 7 | 40 | 40 | 7 | 40 | 40 | 7 | 40 | 40 | 7 | 40 | 40 | 7 | 40 | 40 | 7 | 40 | 40 | 7 | 40 | 40 | 7 | 40 | 40 | 7 |
| Algoritmo | f | V | f | f | V | f | f | V | f | f | V | f | f | V | f | f | V | f | f | V | f | f | V | f | f | V | f | f | V | f |
| N° de repetições | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| palavras | n° de erros | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| zero | | | | | | | 1 | 1 | 1 | | | 1 | | | 3 | 1 | | 2 | 1 | | 3 | | | 2 | | | 2 | | 1 | |
| um | | | 1 | | | 3 | | | | | | 3 | | | 3 | 1 | | | | 1 | 1 | | | 3 | | | 1 | | 2 | 1 |
| dois | | | | | | 1 | | 1 | | | | 2 | | | 1 | | | | | | 3 | | | | | | | | | 1 |
| três | 1 | | 1 | | | 2 | | | 1 | | | | | | 1 | | | 1 | | 2 | 3 | | 2 | 2 | | | | | | 1 |
| quatro | | | | | | 1 | | | 2 | | | | 2 | | 3 | 1 | | 3 | | | | | | | | | | | 1 | |
| cinco | | | | | | | | 1 | | | | | | | | | | | 1 | | | 1 | 1 | 1 | 2 | | | | | |
| seis | | | | | | | | | | | | | | | | | | | | | 2 | | | | | | | | | |
| sete | | | | | | 1 | | | 2 | | | | | | 1 | | | | | | | | | | | | | | | |
| oito | | | | | | | | | 2 | | | 1 | | | | | | | 2 | 3 | 2 | | | | | | | | | |
| nove | | | | | | 2 | 2 | 1 | 1 | | | | | | 1 | | | | | | 2 | 1 | | 3 | | | 1 | | 2 | 2 |
| abrir | | | | | | 1 | | | 1 | | | | | | 2 | | | 1 | | | 3 | | | | | | | | | 2 |
| cancelar | | | | | | | | | 1 | | | | | | 1 | | 1 | | | | | | | 1 | | | | | | 1 |
| fechar | | | | | | | | | | | | | | | | 1 | | | | | 2 | | | | | | | | | |
| ok | | | | | | | | | 3 | | | | | | 1 | | 1 | | | | | | | | | | | | | 1 |
| para | | | | 1 | 3 | 2 | | | 2 | | 1 | 1 | 1 | 1 | | | | 3 | | | | | | | 1 | 1 | 2 | | | |
| total de erros | 1 | 0 | 2 | 1 | 3 | 13 | 3 | 4 | 16 | 0 | 1 | 8 | 3 | 1 | 17 | 4 | 2 | 10 | 4 | 6 | 22 | 2 | 3 | 13 | 1 | 1 | 6 | 0 | 6 | 9 |
| % de erro | 2.2 | 0 | 4.4 | 2.2 | 6.6 | 28.8 | 6.6 | 8.8 | 35.5 | 0 | 2.2 | 17.7 | 6.6 | 2.2 | 37.7 | 8.8 | 4.4 | 22.2 | 8.8 | 13.3 | 48.8 | 2.2 | 6.6 | 28.8 | 2.2 | 2.2 | 13.3 | 0 | 13.3 | 20 |

Tabela B.3 Medidas de erro para o vocabulário de 15 palavras com locutores que participaram da fase de treinamento.

A tabela B.4 mostra os mesmos resultados da tabela B.3, mas para locutores que não participaram do treinamento.

| Locutor | 12 | | | 13 | | | 14 | | | 15 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Nº de palavras no treino | 40 | 40 | 7 | 40 | 40 | 7 | 40 | 40 | 7 | 40 | 40 | 7 |
| Algoritmo | f | V | f | f | V | f | f | V | f | f | V | f |
| Nº de repetições | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| palavras | nº de erros | | | | | | | | | | | |
| zero | | | 3 | 1 | 2 | 2 | | 2 | 3 | | | 1 |
| um | | | 2 | 1 | | 1 | 3 | 1 | 3 | 1 | 1 | 3 |
| dois | | | 2 | 1 | | | | | 1 | | | |
| três | 1 | | 1 | 1 | 2 | 1 | | 1 | 1 | | | |
| quatro | | | | | 1 | | | | | | | |
| cinco | | | 1 | | | 1 | | | 1 | | | |
| seis | | | 1 | 1 | 2 | 3 | | | | 1 | | |
| sete | | | | | | | | | | | | |
| oito | | | | 1 | 2 | | | | | | | |
| nove | | | 1 | | | 2 | 3 | 3 | 3 | 3 | 3 | 1 |
| abrir | | | | | | | | | | | 1 | 1 |
| cancelar | | | 2 | | | | | | | | | |
| fechar | | | | | | | | | | | | 2 |
| ok | | | | | 1 | 2 | | | | | | |
| para | | | | 2 | | 2 | | | 3 | 1 | | 1 |
| total de erros | 1 | 0 | 13 | 8 | 10 | 14 | 5 | 7 | 15 | 4 | 5 | 10 |
| % de erro | 2.2 | 0 | 28.8 | 17.7 | 22.2 | 31.1 | 11.1 | 15.6 | 33.3 | 8.8 | 11.1 | 22.2 |

Tabela B.4 Medidas de erro para o vocabulário de 15 palavras com locutores que não participaram da fase de treinamento.

# REFERÊNCIAS BIBLIOGRÁFICAS


1. ATAL, B.S., HANAUER, S.L. Speech analysis and synthesis by linear prediction of the speech wave. **The Journal of Acoustical Society of America**, v.50, p.637-655, Aug. 1971.
2. BROWN, M.K., RABINER, L.R. An adaptive, ordered, graph search technique for dynamic time warping for isolated word recognition. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v.30, n.11, Aug. 1982.
3. FANZERES, A. Fatores subjetivos da audição humana. **Nova Eletrônica**, n.108, p.8-16, fev. 1986.
4. FANZERES, A. Física e psicologia do nosso ouvido. **Nova Eletrônica**, n.107, p.10-14, jan. 1986.
5. FURUI, S. **Digital speech processing, synthesis, and recognition**. New York, Marcel Dekker, INC., 1989. 390p.
6. FURUI, S. Speaker-independent isolated word recognition using dynamic features of speech spectrum. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v.34, n.1, Feb. 1986.
7. GRAY, R.M., BUZO, A., GRAY, A.H., MATSUYAMA, Y. Distortion mesures for speech processing. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v.28, n.4, p.367-376, Aug. 1980.
8. HERMANSKY, H. Perceptual linear predictive (PLP) analysis of speech. **The Journal of Acoustical Society of America**, v.87, p.1738-1752, Apr,1990.
9. ITAKURA, F. Minimum prediction residual principle applied to speech recognition. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v.23, n.1, p.67-72, Feb. 1975.
10. LAMEL, L.F., RABINER, L.R., ROSENBERG, A.E., WILPON, J.G. An improved endpoint detector for isolated word recognition. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v.29, n.4, p.777-785, Aug. 1981.

11. LAU, Y., CHAN, C. Speech recognition based on zero crossing rate and energy. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v.33, n.1, p.320-323, Feb. 1985.

12. LEE, K. **Automatic speech recognition: the development of the SPHINX system**. Kluwer Academic Publishers, 1989. 207p.

13. LEVINSON, S.E., Structural methods in automatic speech recognition. **Proceedings of the IEEE**, v.73, n.11, p.1625-1650, Nov. 1985.

14. LEVINSON, S.E., RABINER, L.R., ROSENBERG, A.E., WILPON, J.G. Interactive clustering techniques for selecting speaker-independent reference templates for isolated word recognition. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v.27, n.2, p.134-141, Apr. 1979.

15. LEVINSON, S.E., RABINER, L.R., ROSENBERG, A.E., WILPON, J.G. Speaker-independent recognition of isolated word using clustering techniques. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v.27, n.4, p.336-349, Aug. 1979.

16. LEVINSON, S.E., RABINER, L.R., SONDHI, M.M. An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. **The Bell System Technical journal,** v.62, n.4, p.1035-1074, Apr. 1983.

17. LINDE, Y., BUZO, A., GRAY, R.M. An algorithm for vector quantizer design. **IEEE Transactions on Communications**, v.28, n.1, p.84-95, Jan. 1980.

18. LUFT, J. A., NEGREIROS, M., WEIHMANN, T. Sistema de processamento de sinais usando TMS320C25. **Documentação interna. Laboratório de processamanto de sinais, Dep. de Eng. Elétrica, UFRGS**. Outubro, 1993.

19. MAKHOUL, J., ROUCOS, S., GISH, H. Vector quantization in speech coding. **Proceedings of the IEEE**, v.73, n.11, p.1551-1588, Nov. 1985.

20. MAKHOUL, J. Linear prediction: a tutorial review. **Proceedings of the IEEE**, v.63, p.561-580, Apr. 1975.

21. MYERS, C., RABINER, L.R., ROSENBERG, A.E. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v.28, n.6, p.623-635, Dec. 1980.

22. NIEDERJOHN, R.J. A mathematical formulation and comparison of zero-crossing analysis techniques which have been applied to automatic speech recognition. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v.23, n.4, p.373-380, Aug. 1985.

23. OPPENHEIM, A.V., SCHAFER, R.W. **Discrete time signal processing**. Englewood Cliffs, N.J.: Prentice Hall, 1989. 879p.

24. PAN, K., SOONG, F.K., RABINER, L.R. A vector-quantization based preprocessor for speaker-independent isolated word recognition. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v.33, n.3, June 1985.

25. RABINER, L.R. On creating reference templates for speaker independent recognition of isolated words. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v.26, n.1, p.34-42, Feb. 1978.

26. RABINER, L.R., SAMBUR, M.R. An algorithm for determining the endpoints of isolated utterances. **The Bell System Technical journal**, v.54, n.2, p.297-315, Feb. 1975.

27. RABINER, L.R., ROSENBERG, A.E., LEVINSON, S.E. Considerations in dynamic time warping algorithms for discrete word recognition. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v.26, n.6, p.575-582, Dec. 1978.

28. RABINER, L.R., LEVINSON, S.E. Isolated and connected word recognition-theory and selected applications. **IEEE Transactions on communications**, v.29, n.5, p.621-659, May 1981.

29. RABINER, L.R., GOLD, B. **Theory and application of digital signal processing**. Englewood Cliffs, N.J.: Prentice Hall,1975. 762p.

30. RABINER, L.R., LEVINSON, S.E., SONDHI, M.M. On the application of vector quantization and hidden Markov models to speaker-independent, isolated word recognition. **The Bell System Technical journal**, v.62, n.4, p.1075-1105, Apr. 1983.

31. SCHAFER, R.W., RABINER, L.R. Digital representation of speech signals. **Proceedings of the IEEE**, v.63, n.4, p.662-677, Apr. 1975.

32. TEXAS INSTRUMENTS. **TMS320C25 user's guide**. 1986.

33. TOHKURA, Y. A weighted cepstral distance measure for speech recognition. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v.35, n.10, p.1414-1422, Oct. 1987.

34. WEIBEL, A., YEGNANARAYANA, B. Comparative study of nonlinear time warping techniques in isolated word speech recognition sistems. **Technical Report CMU-CS-81-125,** Carnegie-Mellon University, Computer Science Deparatment, June 1981.

35. WEIHMANN, T. **Processamento digital de sinais aplicado à transmissão de voz**. Dissertação de mestrado, PPGEMM, Universidade Federal do Rio Grande do Sul, 1992.

36. ZAYEZDNY, A., TABAK, D., WULICC, D. **Engineering Aplications of Stochastic Processes: theory, problems and solutions**. England: Research Studies Press Ltd., 1989.

37. ZUE, V.W. The use of speech knowledge in automatic speech recognition. **Proceedings of the IEEE**, v.23, n.11, Nov. 1985.