CARLOS DANIEL RIQUELME CUADROS

RECONHECIMENTO DE VOZ E DE LOCUTOR EM AMBIENTES RUIDOSOS: COMPARAÇÃO DAS TÉCNICAS MFCC E ZCPA

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Telecomunicações da Escola de Engenharia da Universidade Federal Fluminense como parte dos requisitos para obtenção do grau de Mestre em Ciências.

Professores Orientadores:

Edson Luiz Cataldo Ferreira, D. Sc. (GMA/UFF)
José Antonio Apolinário Junior, D. Sc. (SE/3/IME)

Niterói
2007

Resumo

Esta dissertação discute a comparação entre duas técnicas de extração de características da voz: a técnica MFCC, que utiliza coeficientes cepstrais de freqüência mel e a técnica ZCPA, que utiliza cruzamento por zero com amplitude de pico. Para tal comparação são utilizados modelos ocultos de Markov (HMM) e diferentes bases de vozes.

O maior destaque é dado à utilização da técnica ZCPA e o seu desempenho no caso de reconhecimento de locutor que é particularmente avaliado em ambientes ruidosos. Verifica-se que a técnica ZCPA é mais robusta que o MFCC quando é aplicado ruído aditivo; também os tipos de frases que ajudam no reconhecimento robusto de locutor são amplamente discutidos. Destaca-se, ainda, a aplicação da técnica ZCPA à base YOHO, amplamente conhecida. Os sinais desta base foram segmentados em dígitos isolados e ruído foi adicionado a cada dígito.

Vários cenários são abordados e implementados, tais como: dígitos isolados, dígitos concatenados e frases completas, com e sem ruído.

Palavras-chave: reconhecimento de voz, reconhecimento de locutor, MFCC, ZCPA.

Abstract

This work discusses the comparison between two features extraction techniques for speech signals: the Mel-Frequency Cepstral Coefficients (MFCC) and the Zero-Crossings with Peak Amplitudes (ZCPA). Hidden Markov Models (HMM) and different corpora are employed for this comparison.

The application of the ZCPA technique is highlighted and its speaker recognition performance is particularly evaluated in noisy environments. It is figured out that the ZCPA technique is more robust to additive noise than the MFCC; also, the types of sentences that help the task of speaker recognition are thoroughly discussed. Special attention is given to the application of ZCPA to the widely known YOHO corpus. The signals from this corpus were segmented in isolated digits and noise was added to each digit.

Many scenarios are addressed, including: isolated digits, concatenated digits, and complete sentence, with and without noise.

Key words: speech recognition, speaker recognition, MFCC, ZCPA.

Declaração de Originalidade

Esta dissertação foi produzida por mim e relaciona trabalho original de minha própria execução. A menos que de outra forma mencionado, os gráficos e tabelas exibidos foram produzidos a partir de dados obtidos durante a pesquisa. Sempre que materiais, idéias, ou algoritmos computacionais de outros pesquisadores tiveram sido usados ou adaptados, a fonte de informação foi claramente especificada. Esta dissertação não foi submetida para graduação ou qualificação profisional em nenhum outro lugar.

Carlos Daniel Riquelme Cuadros

Agradecimentos

À minha família, pela ajuda e por estarem sempre me apoiando, de tão longe, na conclusão desta dissertação.

Aos professores Edson Cataldo e José Apolinário pela orientação e pela amizade que ajudaram na execução desta dissertação.

Ao professor Dirceu Gonzaga, pelo apoio constante, pelas horas de trabalho e sua valiosa amizade.

À CAPES, por ter me condedido a Bolsa de Mestrado, sem a qual tudo teria sido muito difícil.

Ao curso de Pós-Graduação em Engenharia de Telecomunicações da Universidade Federal Fluminense que me concedeu esta grande oportunidade de aumentar meus conhecimentos.

Aos professores Alexandre de la Vega e Murilo Bresciani, pela disponibilidade do Laboratório de Processamento de Sinais.

Ao Instituto Militar de Engenharia (IME) pelo apoio em infraestrutura e equipamentos disponíveis no Laboratório de Voz.

Aos meus amigos e amigas que ajudaram a me fazer sentir em casa.

Ao povo Brasileiro por estar sempre de braços abertos para toda a América Latina.

Dedicatória

Dedico este trabalho a:

Carlos, Blanca e Daniel,

Jhonny, Briguitte, Paola,

Claudia, Daniel, Adriana, Melisa,

Sandra.

Conteúdo

Lista de Figuras **x**

Lista de Tabelas **xii**

1 Introdução **1**

1.1 Introdução 1

1.2 Objetivos da dissertação 4

1.3 Estado da arte 4

1.4 Contribuições desta dissertação 5

2 Fundamentos de produção da voz humana e seu pré-processamento 7

2.1 Geração do sinal de voz 7

2.2 Modelo de produção sonoro/surdo da voz 9

2.3 Pré-processamento da voz 10

2.3.1 Filtro de Pré-Ênfase 11

2.3.2 Janelamento 12

3 Técnicas de extração de características em sinais de voz **14**

3.1 Coeficientes Cepstrais de Freqüência Mel (MFCC) 14

3.1.1 Escala mel 15
3.1.2 Banda crítica 15
3.1.3 Banco de filtros triangulares 16
3.1.4 Cálculo dos MFCCs 17
3.2 Cruzamento por Zero com Amplitude Pico (ZCPA) 19
3.2.1 Banco de filtros 19
3.2.2 Cruzamento por zero 21
3.2.3 Criação dos histogramas 21
3.2.4 Princípio da freqüência dominante 24
3.3 Coeficientes Delta e Delta- Delta 25

4 Modelos ocultos de Markov **27**

4.1 Conceitos básicos 28
4.1.1 Probabilidade ou medida da probabilidade 28
4.1.2 Probabilidade condicional 28
4.1.3 Variável aleatória 29
4.1.4 Espaço amostral 29
4.1.5 Processo estocástico 30
4.1.6 Processo de Markov 30
4.1.7 Cadeia de Markov em tempo discreto 30
4.1.8 Cadeia de Markov em tempo contínuo 32
4.2 Modelos ocultos de Markov (HMMs) 32
4.2.1 Variáveis envolvidas nos HMMs 34
4.2.2 Os problemas básicos dos HMMs e suas soluções 36
4.2.3 Solução do Problema I ou de avaliação 38

4.2.4 Solução do Problema II ou de decodificação 41
4.2.5 Solução do Problema III ou de treinamento 44
4.2.6 Tipos de HMMs: Classificação por transições 47
4.3 Programa HTK (HMM Tool Kit) 48
4.3.1 Arquitetura do HTK 48

5 Reconhecimento automático de voz e locutor 51

5.1 Bases de áudio utilizadas 52
5.1.1 Base de dígitos 52
5.1.2 Base de frases 53
5.1.3 Base YOHO 53
5.1.4 Base de ruídos NOISEX 54
5.2 Reconhecimento de voz: dígitos conectados usando MFCC 54
5.2.1 Reconhecimento de dígitos isolados 56
5.2.2 Resultados do reconhecimento de dígitos isolados 68
5.2.3 Reconhecimento de dígitos conectados 69
5.3 Reconhecimento de locutor 73
5.3.1 Preparação dos arquivos para o treinamento 73
5.3.2 Reconhecimento da base de teste 76
5.4 Reconhecimento usando a base YOHO 90
5.4.1 Segmentação em dígitos isolados 90
5.4.2 Reconhecimento de locutor por dígitos isolados 94

6 Conclusões e trabalhos futuros 98

6.1 Conclusões 98
6.2 Trabalhos Futuros 100

Bibliografia **102**

Lista de Figuras

1.1 Descrição geral do processamento da voz. 2

2.1 a) Aparelho Fonador b) Cordas Vocais 8

2.2 Modelo discreto da produção da voz. 10

2.3 Exemplo de um sinal de voz (trecho da vocal /a/) 11

2.4 Divisão em quadros do sinal de voz. 12

3.1 Escala de freqüência Mel. 16

3.2 Banco de filtros na escala Mel. 17

3.3 Diagrama de fluxo para o cálculo dos MFCCs. 18

3.4 Modelo do ZCPA [15]. 20

3.5 Exemplo da extração dos coeficientes ZCPA. 24

4.1 Características de uma cadeia de Markov com tempo discreto. 32

4.2 Características de uma cadeia de Markov com tempo contínuo. 33

4.3 Exemplo de HMM usando urnas como estados. 36

4.4 Relação entre o HMM, a voz e os três problemas dos HMMs. 37

4.5 Seqüência de operações para o cálculo da variável (forward) $\alpha_{t+1}(j)$. . 40

4.6 Seqüência de operações para o cálculo da variável (backward) $\beta_t(i)$ . . 42

4.7 Tipos de HMMs classificação por transições: a) Ergódico, b) Esquerda-direita, c) Esquerda-direita paralelo 47
4.8 Arquitetura do HTK. 49

5.1 Forma de onda e espectro para 50ms de ruído: a) Branco; b) Fábrica e c) Babble. 55
5.2 Fragmento da lista de extração: Hwav2mfc.scp 58
5.3 Arquivo de configuração para a extração dos coeficientes MFCC: config01 59
5.4 Exemplo de um protótipo de HMM. 61
5.5 Exemplo da lista de treinamento para o dígito “oito”. 62
5.6 Diagrama de fluxo para o HInit. 63
5.7 Diagrama de fluxo para o HRest. 64
5.8 Exemplo da lista de teste. 65
5.9 Exemplo do dicionário empregado. 66
5.10 Exemplo da rede para dígitos isolados. 67
5.11 Exemplo da rede para dígitos conectados. 70
5.12 Exemplo do arquivo de resultados gerado pelo HTK. 72
5.13 Histogramas da técnica ZCPA de um trecho da palavra *fifth* para: a)sinal limpo, b) sinal com 10dB de SNR ruído branco, c) sinal com 5dB de SNR ruído branco. 79
5.14 Fragmento do arquivo “resultsTre”produto do reconhecimento da base de teinamento YOHO 93

Lista de Tabelas

4.1 Classificação dos processos de Markov. 31

5.1 Reconhecimento de dígitos isolados: Acertos=99.76% Total Acertos=1646 Erros=4 Total=1650 (sem c_0) 68

5.2 Reconhecimento de dígitos isolados: Acertos=99.88% Total Acertos=1648 Erros=2 Total=1650 com c_0 69

5.3 Matriz de confusão para dígitos conectados (com c_0). 72

5.4 Taxa de reconhecimento de locutor em % usando a frase E1 e os coeficientes estáticos. 77

5.5 Taxa de reconhecimento de locutor em % usando a frase E1 e os coeficientes estáticos e dinâmicos. 78

5.6 Taxa de reconhecimento de locutor em % com a frase E1 usando 15 coeficientes com $\Delta + \Delta\Delta$ para diferentes quadros. 78

5.7 Taxa de reconhecimento de locutor em % usando a frase E2 e os coeficientes estáticos. 80

5.8 Taxa de reconhecimento de locutor em % usando a frase E2 e os coeficientes estáticos e dinâmicos. 80

5.9 Taxa de reconhecimento de locutor em % usando a frase E2 usando 15 coeficientes com $\Delta + \Delta\Delta$ para diferentes quadros. 81
5.10 Taxa de reconhecimento de locutor em % usando a frase E1 e os coeficientes estáticos. 82
5.11 Taxa de reconhecimento de locutor em % usando a frase E1 e os coeficientes estáticos e dinâmicos. 82
5.12 Taxa de reconhecimento de locutor em % usando a frase E1 usando 15 coeficientes com $\Delta + \Delta\Delta$ para diferentes quadros. 83
5.13 Taxa de reconhecimento de locutor em % usando a frase E2 e os coeficientes estáticos. 84
5.14 Taxa de reconhecimento de locutor em % usando a frase E2 e os coeficientes estáticos e dinâmicos. 84
5.15 Taxa de reconhecimento de locutor em % usando a frase E2 usando 15 coeficientes com $\Delta + \Delta\Delta$ para diferentes quadros. 85
5.16 Taxa de reconhecimento de locutor em % usando a frase E1 e os coeficientes estáticos. 86
5.17 Taxa de reconhecimento de locutor em % usando a frase E1 e os coeficientes estáticos e dinâmicos. 86
5.18 Taxa de reconhecimento de locutor em % usando a frase E1 usando 15 coeficientes com $\Delta + \Delta\Delta$ para diferentes quadros. 87
5.19 Taxa de reconhecimento de locutor em % usando a frase E2 e os coeficientes estáticos. 88
5.20 Taxa de reconhecimento de locutor em % usando a frase E2 e os coeficientes estáticos e dinâmicos. 88

5.21 Taxa de reconhecimento de locutor em % usando a frase E2 usando 15 coeficientes com $\Delta + \Delta\Delta$ para diferentes quadros. 89
5.22 Nomes dos possíveis dígitos isolados da base YOHO e suas etiquetas para as unidades. 91
5.23 Nomes dos possíveis dígitos isolados da base YOHO e suas etiquetas para as dezenas. 91
5.24 Protótipos dos diferentes dígitos para o treinamento. 92
5.25 Taxa de reconhecimento de locutor em % usando os coeficientes MFCC na base YOHO (unidades). 96
5.26 Taxa de reconhecimento de locutor em % usando os coeficientes MFCC na base YOHO (dezenas). 96
5.27 Taxa de reconhecimento de locutor em % usando os coeficientes ZCPA na base YOHO (unidades). 96
5.28 Taxa de reconhecimento de locutor em % usando os coeficientes ZCPA na base YOHO (dezenas). 97

Capítulo 1

Introdução

1.1 Introdução

Os homens sempre buscaram meios de comunicação que facilitassem a interação homem máquina. Com os avanços tecnológicos da área de processamento digital de sinais, o meio de comunicação mais usado pelo homem, a voz, começou a ser mais explorado. A voz possui inúmeras vantagens, entre elas, uma cômoda adaptação do usuário com a máquina, aumentando a capacidade de transmitir informações de forma mais natural.

A voz é o meio mais natural de comunicação do homem. Quando dois indivíduos conversam, pode-se descubrir, por meio de sua voz, algumas características básicas como a idade, o sexo, o idioma, etc. A partir da voz, ainda há a possibilidade de identicar o grupo sócio-cultural, estado emocional, estado de saúde, a região onde mora (através do sotaque) e até mesmo sua identidade. Torna-se claro, portanto, que a partir do sinal de voz é possível obter caracterísitas importantes de cada pessoa. Consequentemente, o homem procura desenvolver tecnologias que permitam a sua comunicação com as máquinas, através da voz.

Os primeiros trabalhos descrevendo máquinas que podiam reconhecer, com certo sucesso, a pronúncia de determinadas palavras datam de 1952 [4]. Uma grande quantidade de trabalhos sobre o assunto surgiu nos anos 60, graças às descobertas de propriedades da voz através do uso de espectógrafos [17] e das novas facilidades que os computadores digitais vieram a oferecer.

Atualmente, a maioria dos sistemas práticos reconhecem somente palavras isoladas, com pequeno vocabulário e pouco robustos ao ruído ambiente. Sistemas que permitam uma comunicação mais natural entre homem e máquina ainda não estão completamente dominados. Por não requererem o uso das mãos e dos olhos do usuário, os sistemas baseados em voz podem ser utilizados nas mais diversas aplicações, como por exemplo: controle de tráfego aéreo, auxílio a deficientes físicos, transações bancárias por telefone e controle de acesso a ambientes restritos [6].

A comunicação vocal entre pessoas e máquinas engloba a síntese de texto para voz, reconhecimento automático de voz (conversão voz-texto), o reconhecimento de locutor e a codificação da voz. A Fig. 1.1 mostra uma descrição geral do processamento da voz, com ênfase em reconhecimento.

Figura 1.1: Descrição geral do processamento da voz.

O reconhecimento de voz pode ser subdividido em um grande número de sub-

áreas, dependendo de alguns fatores tais como tamanho do vocabulário, população de locutores, etc [24]. A tarefa básica no reconhecimento de voz é reconhecer uma determinada elocução de uma sentença ou "entender" um texto falado. Os problemas de reconhecimento de voz por máquinas estão relacionados à estrutura complexa da voz humana, que depende de fatores como: características vocais, entonação, velocidade da voz, estado emocional do usuário, etc.

O objetivo de um sistema de reconhecimento de locutor é reconhecer um locutor a partir da sua voz, sendo bastante útil em aplicacões de segurança como, por exemplo, o controle de acesso a ambientes restritos e o controle de acesso de dados em computadores. O processo de reconhecimento da identidade vocal de locutores consiste na extração de parâmetros da voz, de um dado locutor, de forma a definir um modelo que preserve as suas características vocais que o diferenciem de outros indivíduos.

Um dos grandes problemas comum aos dois sistemas (Reconhecimento de voz e de locutor) é que ainda não se sabe perfeitamente todas as etapas do processo que ocorre no aparelho auditivo e no cérebro humano durante o reconhecimento. Outro problema é a natureza não linear da audição, indicando que modelos matemáticos lineares e simples são inadequados para análise da voz; para isto, são utilizados modelos não lineares tais como Coeficientes Cepstrais de Frequência Mel (MFCC do inglês Mel-Frequency Cepstral Coefficents) e Cruzamento por Zero com Amplitude de Pico (ZCPA do inglês Zero Crossing with Peak Amplitude), que procuram se aproximar do funcionamento do ouvido humano.

1.2 Objetivos da dissertação

- Descrição da técnica ZCPA e sua aplicação em reconhecimento de locutor usando modelos ocultos de Markov;
- Comparar o desempenho das técnicas MFCC e ZCPA para o reconhecimento de locutor em ambientes ruídosos;
- Segmentação automática da base YOHO em dígitos isolados para seu uso em reconhecimento de locutor;
- Reconhecimento robusto de locutor, usando a base YOHO, usando dígitos isolados.

1.3 Estado da arte

Os reconhecimentos de voz e locutor, sem distorções, estão praticamente dominados [2]. Entretanto, com a inserção de ruído aditivo, a situação muda por completo. Devido a isso, vem ocorrendo uma intensificação dos estudos, visando aumentar a robustez do reconhecimento nas etapas seguintes:

- Pré-processamento,
- Processamento: extração das caracteristicas, classificador,
- Pós-processamento ou decisão.

Na fase de pré-processamento o objetivo é minimizar o ruído antes da extração das caracteristicas. No processamento, a extração de características é o alvo, melhorando a extração das características e os modelos estocásticos existentes. Ao final, no pós-processamento ou decisão, a intenção está concentrada no melhor aproveitamento

das informações obtidas na fase de processamento, utilizando sistemas inteligentes que analisam o significado do resultado obtido, em outras palavras, a coerência e o sentido.

Recentemente, na fase da extração de características, surgiram diversas técnicas mais robustas a ruídos aditivos, entre elas o Cruzamento por Zero com Amplitude de Pico (ZCPA) [16] que modificou o algoritmo de Conjunto de Histogramas de Intervalos (EIH - Ensemble Interval Histograms) [11], aumentando a robutez dessa técnica. Entretanto, devido ao alto custo computacional, é pouco utilizada em aplicações práticas.

Entre outras técnicas, cabe destacar a técnica de Histogramas de Centróides de Sub-Bandas do Espectro(SSCH) [10] [9] que é muito robusta frente ao ruído aditivo em aplicações de reconhecimento de voz isolada e contínua de vocabulario médio.

1.4 Contribuições desta dissertação

Uma das contribuições desta dissertação é a utilização da técnica ZCPA em reconhecimento robusto de locutor, até então utilizada em reconhecimento de voz apenas.

Outra contribuição é a comparação do desempenho das técnicas de MFCC e ZCPA para o reconhecimento de locutor em ambientes ruídosos.

A comprovação de que as palavras que apresentam o maior número de fonemas nasais frente àquelas que possuem fonemas orais são melhores para reconhecimento de locutor também pode ser considerada uma contribuição relevante para o desenvolvimento de sistemas de reconhecimento de locutor.

Finalmente, a segmentação da base YOHO em dígitos isolados, deve ser destacada, e tem a finalidade de auxiliar futuros pesquisadores nesta área. Além disso,

possibilita o reconhecimento robusto de locutor por dígitos isolados. Esta última é, talvez, uma boa contribuição do trabalho aqui desenvolvido.

Capítulo 2

Fundamentos de produção da voz humana e seu pré-processamento

2.1 Geração do sinal de voz

A voz humana é produzida por meio do aparelho fonador, formado pelos pulmões (como fonte de energia, na forma de um fluxo de ar), pela laringe (que contém as cordas vocais), pela faringe, pelas cavidades orais (ou bucais) e nasais e por uma série de elementos articulatórios: os lábios, os dentes, o alvéolo, o palato, o véu palatino e a língua. A Fig. 2.1-(a) mostra um esquema do aparelho fonador. As cordas vocais, principais elementos para a geração da voz, são duas membranas situadas na laringe Fig. 2.1-(b). Pela frente, as cordas vocais unem-se à cartilagem tiróide (o pomo de Adão) e, por trás, cada uma delas está presa a uma das cartilagens aritenóides, as quais podem se separar voluntariamente por meio de músculos. A abertura entre as cordas vocais se denomina glote.

Quando respiramos, as cordas vocais encontram-se separadas e a glote adota uma forma triangular. Nesse caso, o ar passa livremente e, praticamente, não é

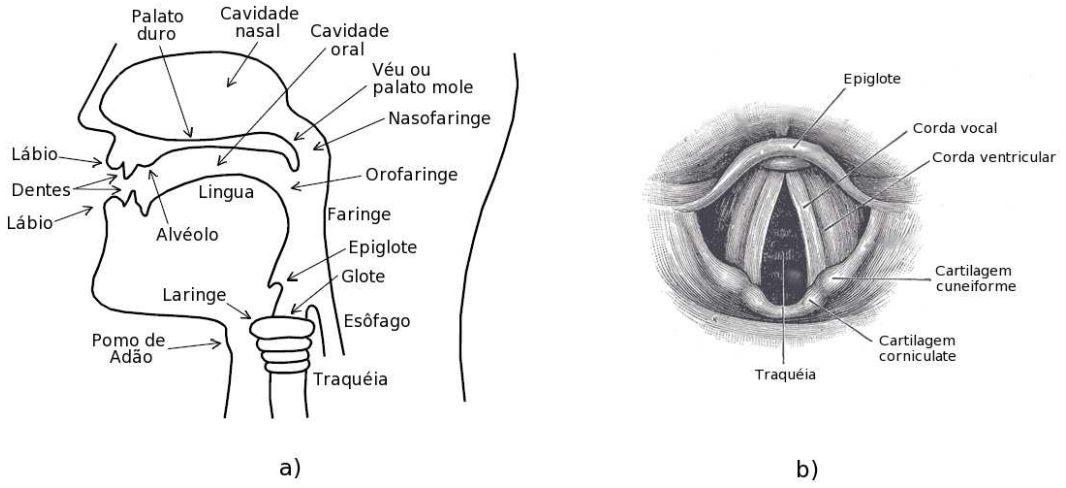

Figura 2.1: a) Aparelho Fonador b) Cordas Vocais

produzido som. Quando a glote começa a se fechar, o ar que a atravessa, proveniente dos pulmões, experimenta uma turbulência, ocasionando um ruído de origem aerodinâmica conhecido como aspiração (na verdade, acompanha uma aspiração). Isto sucede nos sons denominados aspirados. Ao se fecharem um pouco mais, as cordas vocais começam a vibrar, produzindo um sinal de pressão (quase) periódico, também chamado de *sinal glotal*. A freqüência fundamental deste sinal, também chamada de *pitch*, depende de vários fatores, como o tamanho e a massa das cordas vocais, a tensão aplicada nas cordas vocais e a pressão do ar proveniente dos pulmões. Finalmente, é possível fechar a glote completamente.

Quando as cordas vocais estão vibrando, os pulsos de ar do sinal glotal sofrem a influência do sistema de ressonância formado pelos órgãos do trato vocal e nasal (faringe e cavidades bucal e nasal) funcionando como um filtro. Há, ainda, a ação dos órgãos do sistema articulador (língua, palato mole, maxilar e lábios) que modificam as propriedades de filtragem dos órgãos do sistema de ressonância sobre o sinal glotal. Graças à filtragem do sistema articulador, é possível a geração dos diferentes

sons que emitimos quando falamos.

2.2 Modelo de produção sonoro/surdo da voz

Para um modelamento detalhado do processo de produção da voz, os seguintes efeitos devem ser considerados [24]:

1. Variação da configuração do trato vocal com o tempo;
2. Perdas próprias por condução de calor e fricção nas paredes do trato vocal;
3. A maciez das paredes do trato vocal;
4. Radiação do som pelos lábios;
5. Junção nasal;
6. Excitação do som no trato vocal, etc.

Um modelo detalhado para geração de sinais de voz, que leva em conta os efeitos da propagação e da radiação conjuntamente pode, em princípio, ser obtido através de valores adequados para excitação e parâmetros do trato vocal. A teoria acústica sugere uma técnica simplificada para modelar sinais de voz, a qual é bastante utilizada.

Essa técnica apresenta a excitação separada do trato vocal e da radiação. Os efeitos da radiação e o trato vocal são representados por um sistema linear variante com o tempo. O gerador de excitação gera um sinal similar a um trem de pulsos, ou sinal aleatório (ruído). Os parâmetros da fonte e sistema são escolhidos de forma a se obter na saída o sinal de voz desejado [24]. Colocando-se todos os componentes necessários, obtém-se o modelo da Fig. 2.2.

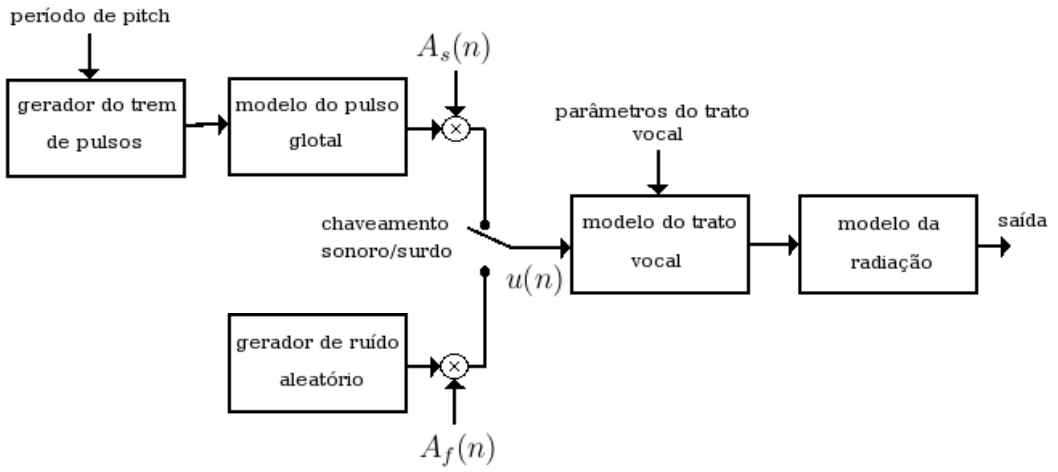

Figura 2.2: Modelo discreto da produção da voz.

Na figura acima, $u(n)$ é o sinal de excitação, $A_s(n)$ e $A_f(n)$ controlam a intensidade da excitação do sinal sonoro e do ruído, respectivamente, onde ocurre um chaveamento entre sonoro e surdo alterando o modo de excitação.

2.3 Pré-processamento da voz

A voz humana é um sinal de pressão acústica que varia com o tempo. Esse sinal, analógico, pode ser convertido em um sinal digital de modo a possibilitar seu processamento através de programas de computador. Portanto, o sinal de voz é usualmente captado por um microfone, e transformado em um sinal elétrico. O sinal obtido é amostrado com uma freqüência de amostragem maior que o dobro da freqüência máxima do sinal, segundo o Teorema da Amostragem [21]. Um exemplo do sinal de voz, produzido por uma vogal sustentada */a/*, é mostrado na Fig. 2.3.

Porém, algumas modificações devem ser realizadas no sinal de voz, antes de processá-lo. Elas são pré-ênfase e janelamento e serão discutidas a seguir.

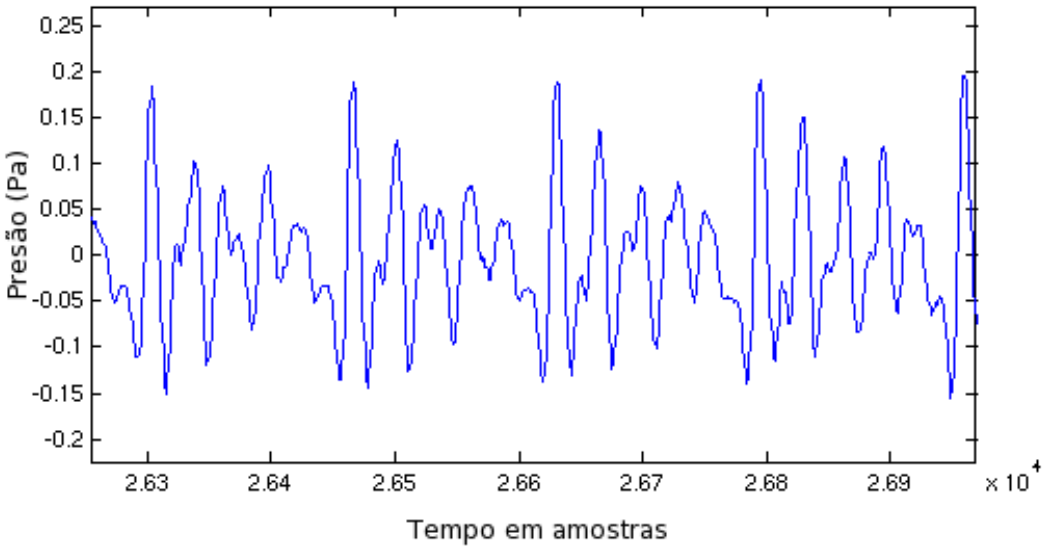

Figura 2.3: Exemplo de um sinal de voz (trecho da vocal */a/*)

2.3.1 Filtro de Pré-Ênfase

A filtragem de pré-ênfase serve para atenuar as componentes de baixa freqüência e incrementar as componentes de alta freqüência do sinal de voz, prevenindo contra instabilidade numérica e, também, minimizando o efeito dos lábios e da glote [5].

A função de transferência mais usada para um filtro de pré-ênfase é dada por [24]:

$$H(z) = 1 - az^{-1}, \qquad 0,9 \leq a \leq 1,0. \tag{2.1}$$

Neste caso, a saída do sistema de pré-ênfase $\tilde{s}(n)$ está relacionada à entrada $s(n)$ pela equação de diferenças:

$$\tilde{s}(n) = s(n) - as(n-1) \tag{2.2}$$

onde o valor de a usualmente usado é 0,95 [24].

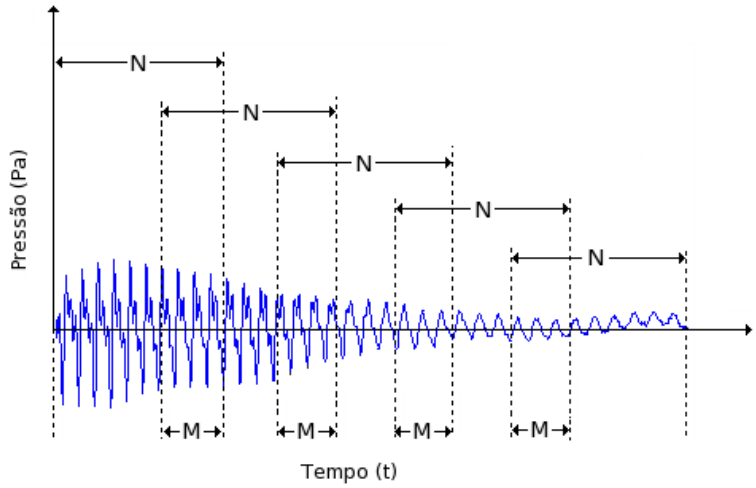

Figura 2.4: Divisão em quadros do sinal de voz.

2.3.2 Janelamento

Após a pré-ênfase, passa-se à etapa de "janelamento"do sinal de voz. Nesta etapa, são extraídos quadros, digamos, de N amostras a partir do sinal $\tilde{s}(n)$, tendo uma superposição de M amostras (ver Fig. 2.4). Tal divisão é extremamente importante devido ao fato de um sinal de fala ser variante no tempo. O janelamento de pequenos segmentos, que variam de 10 a 45 ms, possibilita minimizar as descontinuidades do sinal no começo e no final de cada janela (frame) e admitir que ele seja aproximadamente estacionário nesses intervalos [24], permitindo assim o uso de métodos tradicionais de análise espectral. Geralmente, para separar cada segmento do sinal de voz, usa-se uma janela de Hamming ([5][24]), definida por

$$h(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), & 0 \leq n \leq N-1 \\ 0, & c.c. \end{cases} \tag{2.3}$$

onde n é o índice da amostra e N é o número total de amostras da janela.

Há outros processos que podem ser introduzidos na fase de pré-processamento, tais como: filtro passa altas (pré-ênfase), filtros passa banda, filtragem adaptativa,

normalização de energia, etc [5].

Capítulo 3

Técnicas de extração de características em sinais de voz

Com o objetivo de aplicar ferramentas matemáticas, o sinal de voz pode ser representado por uma seqüência de vetores de características. Neste capítulo, serão apresentadas duas técnicas de extração de características, uma é chamada de Coeficientes Cepstrais de Freqüência Mel (MFCC) e a outra de Cruzamento por Zero com Amplitude Pico (ZCPA).

3.1 Coeficientes Cepstrais de Freqüência Mel (MFCC)

A técnica MFCC surgiu devido aos estudos na área de psicoacústica (a ciência que estuda a percepção auditiva humana). Esta ciência mostra que a percepção das freqüências de tons puros ou de sinais de voz não seguem uma escala linear, estimulando assim, a idéia de criar uma escala, chamada *mel*.

3.1.1 Escala mel

A escala "Mel" foi desenvolvida por Stevens e Volkman, em 1940. Como referência, definiu-se a freqüência de 1 kHz, com potência de 40 dB, como 1000 mels. Os outros valores subjetivos foram obtidos através de experimentos, onde pedia-se a ouvintes que ajustassem a freqüência física de um tom, até que a freqüência percebida fosse igual a duas vezes a freqüência de ; depois, 10 vezes a freqüência de referência e assim por diante. Essas freqüências teriam os valores de 2000 mels, 10000 mels e assim sucessivamente. O mesmo processo era efetuado na outra direção, ou seja, metade do tom de referência, um décimo do tom de referência, etc. Essas freqüências teriam valores de 500 mels, 100 mels, etc. Isto permitiu verificar que o mapeamento entre a escala de freqüência real, em Hz, e a escala de freqüências percebida, em mel, é aproximadamente linear abaixo de 1000 Hz e, logarítmica acima.

A Eq. 3.1 descreve a escala Mel e seu gráfico é mostrado na Fig. 3.1.

$$Mel(f) = 1127 \ln\left(1 + \frac{f}{700}\right). \tag{3.1}$$

3.1.2 Banda crítica

Alguns experimentos demostraram que a percepção humana de algumas freqüências de sons complexos não podem ser individualmente identificadas, dentro de certas bandas. Quando uma componente cai fora da banda, chamada de banda crítica, ela pode ser identificada. Uma explicação apresentada para esse fato é que a percepção de uma freqüência particular pelo sistema auditivo, por exemplo f, é influenciada pela energia de certa banda de freqüências em torno de f, o valor dessa banda varia nominalmente de 10% a 20% da freqüência central do som, começando em torno

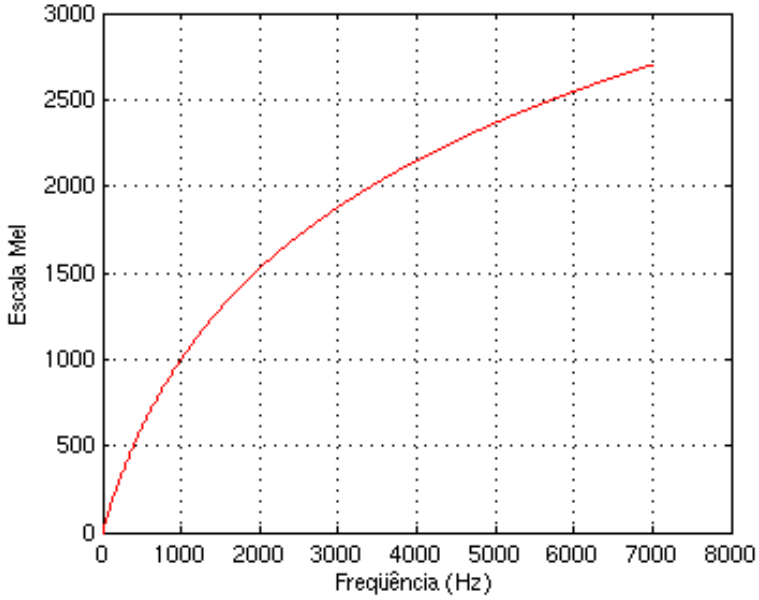

Figura 3.1: Escala de freqüência Mel.

de 100 Hz para freqüências abaixo de 1 KHz e aumentando em escala logarítmica acima [29].

Cabe destaque à representação *cepstral* associada à escala mel apresentando maior eficácia computacional, sendo chamada de Mel-Cepstral.

3.1.3 Banco de filtros triangulares

A Fig. 3.2 apresenta a configuração para o cálculo dos coeficientes Mel-Cepstrais. Para a faixa de freqüências de interesse (300 Hz - 3.4 KHz), utilizam-se 24 filtros (típico) centrados nas freqüências da escala mel. O espaçamento é de aproximadamente 150 mels e a largura de banda de cada filtro triangular é de 300 mels. Este banco de filtros simula a resposta em freqüência da *membrana basilar* [8].

Esses fenômenos (escala mel e banda crítica) sugeriram que seria mais interessante fazer algumas modificações na representação espectral do sinal. Tais modificações consistiram, primeiramente, em fazer uma ponderação da escala de

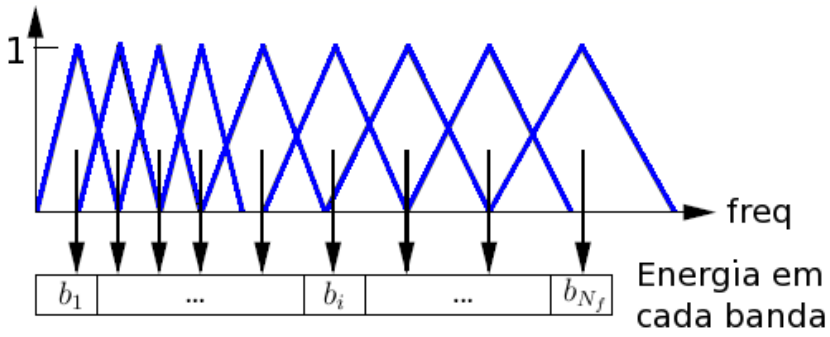

Figura 3.2: Banco de filtros na escala Mel.

freqüência para a escala mel, e depois, incorporar a noção de banda crítica na definição de distorção espectral. Ou seja, ao invés de usar simplesmente o logaritmo da magnitude das freqüências, passou-se a utilizar o logaritmo da energia total das bandas críticas em torno das freqüências mel. A aproximação mais utilizada para esse cálculo é a utilização de um banco de filtros triangulares [4], espaçados uniformemente em uma escala não linear (escala mel). A técnica de ponderação mel pode ser aplicada a vários tipos de representação espectral.

3.1.4 Cálculo dos MFCCs

Para o cálculo dos coeficientes cepstrais de freqüência mel, inicialmente, divide-se o sinal de voz $s(n)$ em janelas como já foi descrito no Capítulo 2. Para cada janela m estima-se o espectro $S(w, m)$, utilizando a FFT. O espectro modificado $P(i),\ i = 1, 2, \ldots, N_f$, consistirá na energia de saída de cada filtro, expresso por

$$P(i) = \sum_{k=0}^{N/2} |S(k, m)|^2 H_i \left(k \frac{2\pi}{N} \right) \tag{3.2}$$

onde N é o número de pontos da FFT, N_f é o número de filtros triangulares, $|S(k, m)|$ é o módulo da amplitude na freqüência do k-ésimo ponto da m-ésima janela e $H_i(w)$ é a função de transferência do i-ésimo filtro triangular.

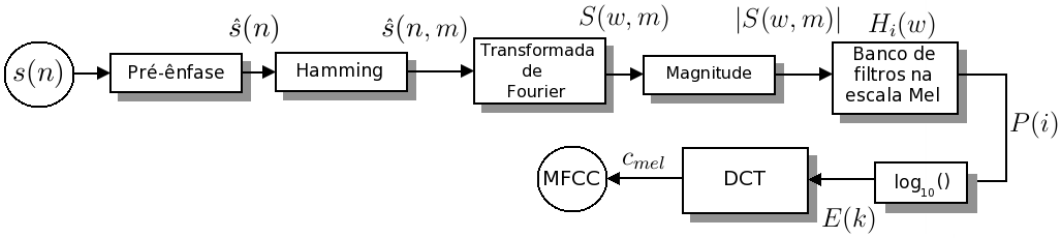

Figura 3.3: Diagrama de fluxo para o cálculo dos MFCCs.

Em seguida, define-se o conjunto de pontos $E(k)$ por

$$E(k) = \begin{cases} \log[P(i)] & k = k_i \\ 0 & \text{qq outro } k \in [0, N-1] \end{cases} \tag{3.3}$$

onde k_i é o ponto máximo do i-ésimo filtro.

Os coeficientes mel-cepstrais $c_{mel}(n)$ são então obtidos com o uso da Transformada Discreta de Coseno (DCT), dado por

$$c_{mel}(n) = \sum_{i=0}^{N_f} E(k_i) \cos\left(\frac{2\pi}{N} k_i n\right), \qquad n = 0, 1, 2, \ldots, N_c - 1 \tag{3.4}$$

onde N_c é o número de coeficientes mel-cepestrais desejado, N_f é o número de filtros e k_i é o ponto máximo do i-ésimo filtro. Se $N_c = 15$ então se terá um vetor como é mostrado a seguir:

$$c_{mel} = c_0, c_1, c_2, \ldots, c_{13}, c_{14}.$$

Nesse vetor, considera-se o primeiro coeficiente, denotado por c_0 que pode carregar muita informação do meio de transmissão [5]. Este coeficiente por vezes é considerado e por vezes não; isto vai depender do tipo de reconhecimento que se deseja. A Fig. 3.3 mostra uma representação do processo de obtenção dos coeficientes MFCC. Note-se que no diagrama de fluxo é usada diretamente a Transformada Discreta de Coseno, em lugar da IFFT, devido a suas propiedades homomórficas e de descorrelação dos dados [1].

3.2 Cruzamento por Zero com Amplitude Pico (ZCPA)

Motivado pela técnica Histograma do Conjunto de Intervalos (EIH do inglês Ensemble Interval Histogram) [11], Kim em [16] propôs uma modificação do EIH, mantendo um único nível para o detetor de cruzamento por zeros, enquanto a informação de intensidade foi preservada medindo-se a amplitude pico entre cruzamentos sucessivos por zero.

De uma maneira geral, conforme pode ser observado na Fig. 3.4, o ZCPA é obtido passando um sinal de voz através de um banco de filtros, onde o sinal é dividido em sub-bandas. Após essa etapa, em cada sub-banda é obtido o número de cruzamentos positivos de zeros. Para cada par de cruzamentos de zeros o inverso do intervalo entre eles é calculado. A partir dessas informações, um único histograma dos inversos dos intervalos é plotado para todas as sub-bandas do sinal. Nesse histograma, o incremento é dado pelo logaritmo da amplitude de pico detectada no intervalo. Por fim, é calculada a transformada discreta do cosseno (DCT) no histograma, calculando-se os coeficientes ZCPA cepstrum.

3.2.1 Banco de filtros

O banco de filtros usado pode ser implementado usando o modelo projetado por Lyon e Mead [11], o qual representa a propagação da onda ao longo da cóclea. Resultados experimentais em [15] mostram que o uso de filtros FIR (Finite Impulse Response) implementados por janelas de Hamming podem ser mais eficientes e aumentar a taxa de reconhecimento, apesar do tipo do ruído e da SNR (Signal-to-Noise-Rate).

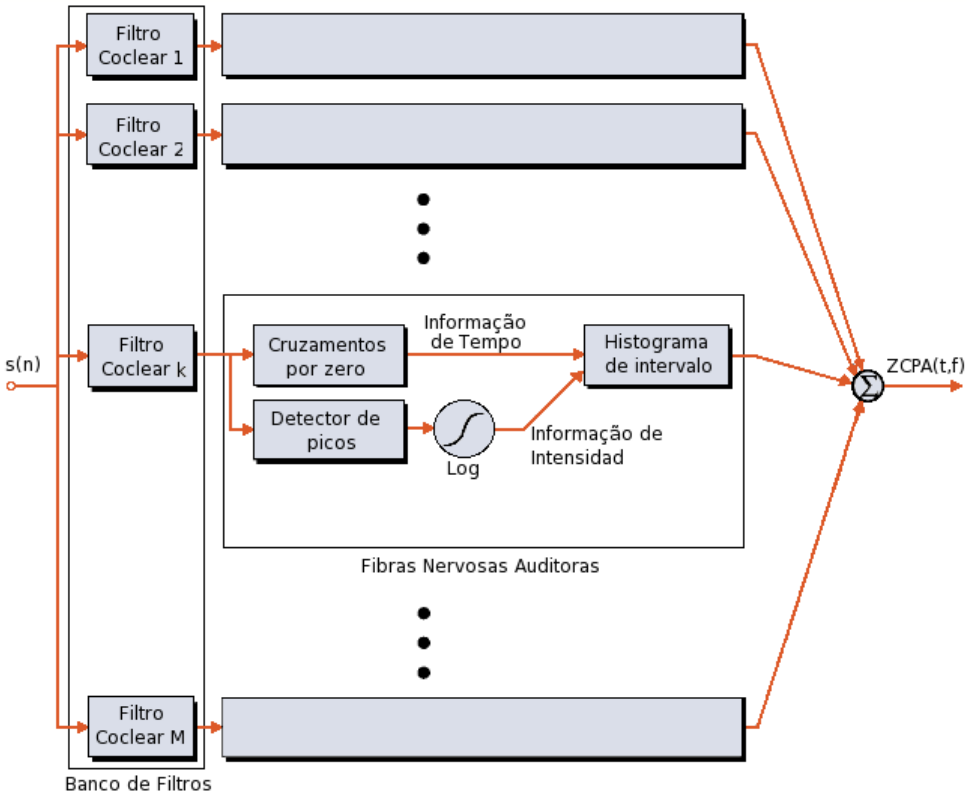

Figura 3.4: Modelo do ZCPA [15].

O número de canais K (ou bandas) indicados para uso no ZCPA em [15] é de $K = 16$ até $K = 23$ canais. As bandas são dispostas segundo a escala Bark dada pela Eq. 3.5,

$$f_{Bark} = 13atan\left(\frac{0.76f}{1000}\right) + 3.5atan\left(\frac{f}{7500}\right)^2, \tag{3.5}$$

onde f é a freqüência em Hertz e f_{Bark} é a freqüência perceptual em Bark correspondente.

A largura de banda de cada canal é fixo de acordo com resultados experimentais [8] em RAV (Reconhecimento Automático de Voz); onde cada um dos k canais tem de 2 ou 3 vezes a banda crítica [8] perceptual $BW_{critica}(fc_k)$ dada pela Eq. 3.6,

$$BW_{critica}(fc_k) = 25 + 75\left[1 + 1.4\left(\frac{f}{1000}\right)^2\right]^{0.69}, \tag{3.6}$$

onde f é dada em Hertz.

3.2.2 Cruzamento por zero

Após passar pelo banco de filtros que simulam a cóclea, a saída $s_k(n)$ de cada sub-banda é processado por um detector de cruzamento positivo por zero, com a finalidade de obter para cada dois cruzamentos sucessivos $z_k(i)$ e $z_k(i+1)$ o pico máximo (Eq. 3.7) $p_k(i)$ e o inverso do tamanho do intervalo entre eles (Eq. 3.8).

$$p_k(i) = \max_{z_k(i) \leq n < z_k(i+1)} \{s_k(n)\} \tag{3.7}$$

$$f_k(i) = \frac{1}{z_k(i+1) - z_k(i)} \tag{3.8}$$

3.2.3 Criação dos histogramas

Há dois fatores que afetam as propriedades do histograma. Uma delas é a alocação de raias (bins) na faixa de freqüência de interesse e a outra é a escolha do tamanho

da janela em que serão realizados a deteção dos cruzamentos por zero e os cálculos para construção do espectro.

A alocação das raias de freqüência é feita de acordo com a escala Bark, conforme a Eq. 3.5. A largura, R, de cada raia é dada pela Eq. 3.6. Ou seja, à medida que a freqüência aumenta, a largura de R também aumenta, levando o histograma a uma polarização nas altas freqüências.

Através de várias medições realizadas em [11], foi mostrado que o ouvido humano responde com uma alta resolução em freqüência e pobre resolução no tempo para baixas freqüências e vice-versa para as altas freqüências. Isso pode ser implementado com o uso de janelas temporais de tamanhos distintos. Além disso, a fim de que o sinal filtrado em cada canal $s_k(n)$ tenha o mesmo número de períodos, o comprimento da janela para o k-ésimo canal deve ser idealmente $L_k = N_p/fc_k$, onde N_p é o número de períodos desejado [15]. Isto leva em consideração que o sinal é senoidal com freqüência igual à freqüência central de cada canal. Desse modo, o comprimento da janela torna-se longo para as baixas freqüências e curto para as altas. Isso resulta numa alta resolução em freqüência e baixa resolução no tempo para as baixas freqüências e vice-versa para as altas freqüências, conforme já comentado. Considerando $N_p = 20$, note que $L_k = 20/fc_k$ leva a comprimentos de janelas muito diferentes das bandas mais altas para as mais baixas.

Para ilustrar esse problema, consideremos o caso de $K = 16$, $fc_1 = 150$ Hz e $fc_{17} = 3400$ Hz, de acordo com a freqüência central bark obtida por Eq. 3.5. Neste caso, teremos $L_1 = 133ms$ e $L_{17} = 6ms$. Obviamente, L_{17} é muito curto, uma vez que este valor é menor que um período de Pitch e muito menor que L_1. A fim de reduzir esse problema, [8] utilizou uma nova expressão para o cálculo do

comprimento das janelas para o RAV:

$$L_k = \frac{N_p}{\sqrt[2]{fc_k/1000}}\ milisegundos. \tag{3.9}$$

Para o mesmo exemplo, a Eq. 3.9 resulta em $L_{17} = 11ms$ e $L_1 = 52ms$, produzindo uma excursão muito mais aceitável. A utilização de janelas de comprimento elevado, parece ser uma característica intrínsica da estimação de freqüência no domínio do tempo, e isto está relacionado à necessidade de múltiplos pares de Cruzamentos por Zero positivos (ascendentes) para uma boa precisão na estimação (boa resolução em freqüência). A fim de ter uma boa precisão na estimação dos cruzamentos pelos zeros e em conseqüência uma boa estimação da freqüência, utilizam-se, normalmente, interpoladores nas saídas dos bancos de filtros.

Uma vez que são especificadas a alocação de raias (bins) e a janela de observação; procede-se com a construção do histograma de freqüências f_k para todos os sinais das sub-bandas $k = 1, \ldots, 16$.

O incremento no histograma é dado pelo logaritmo do $p_k(i)$ correspondente, assim:

$$inc(j) = \sum_k \sum_i \Psi_j \{f_k(i)\} \qquad \text{onde,} \tag{3.10}$$

$$\Psi_j \{f_k(i)\} = \begin{cases} \ln(p_k(i)) & ,\ f_k(i) \epsilon R_j\ ; \\ 0 & ,\ \text{caso contrario.} \end{cases} \tag{3.11}$$

No final, a Transformada Discreta do Coseno (DCT) é computada no histograma, visando uma descorrelação [1], obtendo-se os coeficientes cepstrais ZCPA; da mesma forma que os coeficientes MFCC, no caso do ZCPA pode-se ou não considerar o primeiro coeficiente.

A seguir na Fig. 3.5, é mostrado um exemplo prático de como obter os coeficientes ZCPA para um sinal de voz de $50ms$. Pode-se observar que o sinal é

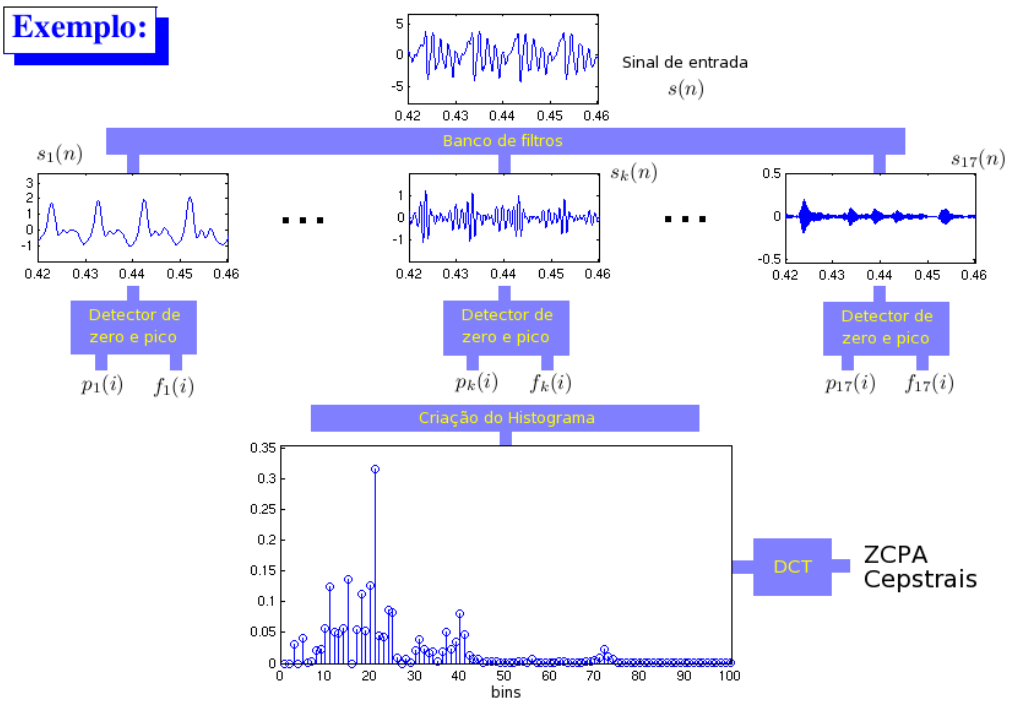

Figura 3.5: Exemplo da extração dos coeficientes ZCPA.

dividido em 17 sinais (sub-bandas), cada um representa uma saída do banco de filtros. Depois, para cada sinal é calculado o pico máximo e o inverso do tamanho do intervalo para cada cruzamento positivo por zero. Logo, é criado um histograma geral contendo todos os histogramas relativos de cada sub-banda. No final, é calculada a DCT do histograma e como resultado temos os coeficientes ZCPA.

3.2.4 Princípio da freqüência dominante

Do ponto de vista de processamento de sinais, o histograma do ZCPA pode ser visto como uma representação alternativa do espectro de voz. Isto está baseado no princípio da freqüência dominante [14] que estabelece que, se há uma freqüência significativamente dominante (maior potência) no sinal, então o inverso do intervalo de cruzamento por zero tende a tomar valores na vizinhança desta freqüência. Assim, o inverso do intervalo de cruzamentos por zero da k-ésima sub-banda pode ser visto

como uma estimativa da freqüencia dominante da sub-banda [27].

3.3 Coeficientes Delta e Delta- Delta

O desempenho dos sistemas de reconhecimento pode ser enormemente melhorado adicionando maior informação do sinal, por exemplo, a primeira e a segunda derivadas. Os coeficientes cepstrais, resultado do cálculo da DCT, são conhecidos também como coeficientes "estáticos"e os coeficientes obtidos a partir da primeira e segunda derivadas são chamados de coeficientes "dinâmicos", porque são utilizados para representar as mudanças dinâmicas no espectro da voz e, desse modo, detectar variações bruscas dentro do espectro. Uma equação muito usada é a seguinte [30]:

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta(c_{t+\theta} - c_{t-\theta})}{2\sum_{\theta=1}^{\Theta} \theta^2}, \tag{3.12}$$

onde d_t é o coeficiente delta (Δ) no tempo t calculado em termos dos correspondentes coeficientes estáticos $c_{t-\Theta}$ até $c_{t+\Theta}$. O valor de Θ é o número de amostras necessarias para o cálculo dos coeficientes dinâmicos e este valor é normalmente achado de forma empírica; segundo a literatura [27][8], os valores mais típicos são de 2, 4 ou 8, dependendo das mudanças do sinal. Os parâmetros de segunda ordem são obtidos reaplicando a derivada sobre os resultados obtidos na primeira derivação.

Assim, por exemplo, se queremos calcular 15 coeficientes MFCC com seus respectivos coeficientes dinâmicos, teríamos no final 15 coeficientes estáticos, 15 coeficientes obtidos da primeira derivada (Δ) e mais 15 coeficientes obtidos da segunda derivada ($\Delta\Delta$); isto é, um vetor de 45 coeficientes. Normalmente, o cálculo anterior é realizado sem considerar o primeiro coeficiente (c_0); portanto, se considerarmos o

c_0, então teríamos um vetor de 48 coeficientes.

Capítulo 4

Modelos ocultos de Markov

Dentre as ferramentas usadas em reconhecimento temos os modelos ocultos de Markov. Inicialmente introduzidos e estudados no fim dos anos 60 e princípios de 70, os modelos ocultos de Markov (HMMs - Hidden Markov Models) tornaram-se populares nos últimos anos. Há duas grandes razões para essa popularidade:

1) os HMMs têm uma forte estrutura matemática e, portanto, formam uma boa base teórica para uma ampla gama de aplicações;

2) os HMMs, quando usados apropriadamente, fornecem bons resultados nas tarefas de RAV e RAV.

3) os HMMs possuem uma representação direta quando usados no processamento da voz.

Neste capítulo, será apresentada a teoria relacionada aos modelos ocultos de Markov, assim como os algoritmos usados para sua implementação.

4.1 Conceitos básicos

O homem tem sempre tentado comprender e definir fenômenos que ocorrem de uma maneira inesperada ou que não podem ser previstos. Dentre esses fenômenos estão os chamados fenômenos aleatórios. Tais fenômenos estão presentes em quase todos os campos do conhecimento; como por exemplo, em Biotecnologia, Informática, Mecânica, Telecomunicações, etc. Consequentemente, sua modelagem é fundamental. Dentre as ferramentas para a modelagem de fenômenos aleatórios estão os modelos ocultos de Markov.

A seguir, apresentaremos um breve resumo teórico de Probabilidades e de Processos Estocásticos, ferramentas fundamentais ao estudo dos HMMs.

4.1.1 Probabilidade ou medida da probabilidade

Chamemos S o espaço amostral de um experimento, A é um evento. Chamemos P da medida da probabilidade ou, simplesmente, probabilidade, tal que:

$$P[A] \geq 0 \tag{4.1}$$

$$P[S] = 1 \tag{4.2}$$

$$\text{Se } A_1, A_2, \ldots \text{ são disjuntos } \Rightarrow \; P\left(\cup_{i=1}^{+\infty} A_i\right) = \sum_{i=1}^{+\infty} P(A_i) \tag{4.3}$$

4.1.2 Probabilidade condicional

Em alguns casos estamos interessados em determinar se dois eventos (subconjuntos do espaço amostral), digamos A e B, estão relacionados de maneira que a ocorrência de B afete ou não a probabilidade de A; ou seja, calcular a probabilidade condicional,

$P[A|B]$, do evento A dado que o evento B ocorreu. Esse cálculo é definido por:

$$P[A|B] = \frac{P[A \cap B]}{P[B]}, \qquad \text{para } P[B] > 0. \tag{4.4}$$

Em um caso particular, dois eventos A e B são ditos **independentes** se o conhecimento de um deles não altera a probabilidade de o outro; ou seja, $P[A|B] = P[A]$.

4.1.3 Variável aleatória

Uma variável aleatória é uma função X tal que $X(\zeta)$ é um número associado a cada resultado ζ de um experimento. Este número pode ser a voltagem de uma fonte aleatória, o custo de um componente aleatório ou qualquer outra quantidade que seja de interesse na realização de um experimento aleatório. Formalmente, uma variável aleatória X é uma função que associa a cada resultado ζ de um experimento um número $X(\zeta)$.

4.1.4 Espaço amostral

Quando um experimento aleatório é realizado, uma única saída ocorre. Diz-se, então, que as saídas são mutuamente exclusivas no sentido de que elas não podem ocorrer simultaneamente. Portanto, definimos o espaço amostral S de um experimento aleatório como o conjunto de todas as possíveis amostras.

Por exemplo, considere o experimento de medir a tensão correspondente de um sinal de voz de amplitude normalizada, em um determinado instante. Neste caso, o espaço amostral S é o intervalo $[1, -1]$.

4.1.5 Processo estocástico

Um processo estocástico é uma regra que associa a cada resultado ζ de um experimento aleatório uma função $X(t, \zeta)$. Assim, um processo estocástico é uma família de funções dependentes do tempo e do parâmetro ζ ou, equivalentemente, uma função de t e ζ. Usaremos $X(t)$ para representar um processo estocástico.

4.1.6 Processo de Markov

Um processo estocástico $X(t)$ é chamado de processo de Markov se a probabilidade condicional de qualquer estado futuro do processo depender somente de seu estado presente [18]. Isto é:

$$P[X(t_{k+1}) = x_{k+1} | X(t_k) = x_k, \ldots, X(t_1) = x_1] \quad , t_1 < t_2 < \ldots < t_k < t_{k+1}$$

$$= P[X(t_{k+1}) = x_{k+1} | X(t_k) = x_k] \text{ quando } X(t) \text{ é discreto, e} \tag{4.5}$$

$$P[a < X(t_{k+1}) \leq b | X(t_k) = x_k, \ldots, X(t_1) = x_1] \quad , t_1 < t_2 < \ldots < t_k < t_{k+1}$$

$$= P[a < X(t_{k+1}) \leq b | X(t_k) = x_k] \text{ quando } X(t) \text{ é contínuo.} \tag{4.6}$$

O conjunto (Q) dos valores possíveis do processo $X(t)$ é chamado de espaço de estados deste processo [28]. Segundo o espaço de estados e o espaço dos tempos, os processos de Markov podem ser classificados de acordo com a Tab. 4.1:

4.1.7 Cadeia de Markov em tempo discreto

Uma cadeia de Markov em tempo discreto é uma coleção de variáveis aleatórias de um processo de Markov que assumem valores dentro de um espaço de estados enumerável (finita ou infinitamente).

Tabela 4.1: Classificação dos processos de Markov.

Parâmetro t	**Espaço de estados**	
	Discreto	Contínuo
Discreto	Cadeia de Markov em tempo discreto	Processo de Markov em tempo discreto
Contínuo	Cadeia de Markov em tempo contínuo	Processo de Markov em tempo contínuo

Uma cadeia de Markov em tempo discreto se caracteriza por uma função distribuição de probabilidades (FDP) uni-dimensional de seus estados e por uma matriz de probabilidades de transição entre seus estados.

A Fig. 4.1 mostra uma representação das características de uma cadeia de Markov $X(t)$ sendo

$$\pi_i(m) = P[X(t_m) = x_i], \tag{4.7}$$

a distribuição de probabilidades dos estados de $X(t)$ e a matriz de transição

$$a_{ij}(m,n) = P[X(t_n) = x_j | X(t_m) = x_i], \qquad t_m < t_n. \tag{4.8}$$

Por exemplo, seja $X(t)$ um processo estocástico que representa o clima de uma região tal que a cada hora do dia o clima pode ser chuvoso, nublado ou ensolarado. Assim, associamos para cada instante t, uma variável aleatória $X(t)$ tal que o espaço amostral S_X dessa variável aleatória é: $S_X = \{chuvoso, nublado, ensolarado\}$. Escolhendo um conjunto de variaveis aleatórias de $X(t)$, caracterizamos uma cadeia de Markov em tempo discreto.

Uma cadeia de Markov é dita **homogênea** se a probabilidade de transição $a_{ij}(m,n)$ depende só de $n-m$; ou seja, a cadeia de Markov homogênea está com-

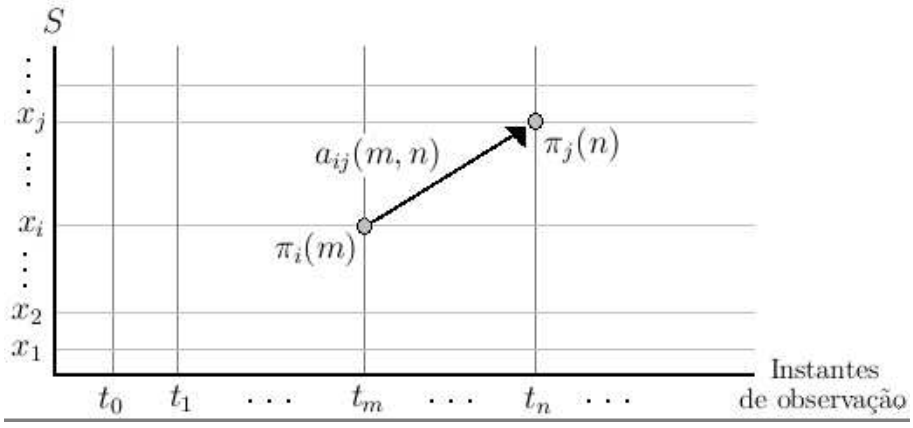

Figura 4.1: Características de uma cadeia de Markov com tempo discreto.

pletamente determinada pelo vetor das probabilidades iniciais dos estados, $\pi_i(0)$ e pela matriz de transição entre estados.

4.1.8 Cadeia de Markov em tempo contínuo

Um processo de Markov $X(t)$ é dito uma cadeia de Markov em tempo contínuo se o seu espaço de estados S é enumerável e o intervalo de tempo de observação é contínuo.

Neste tipo de cadeia, como representado na Fig. 4.2, dado que o tempo é contínuo, o comportamento da cadeia é uma função em escada com descontinuidades em pontos (aleatórios) correspondentes aos instantes de mudança de estado do processo.

4.2 Modelos ocultos de Markov (HMMs)

Os modelos ocultos de Markov são uma extensão das cadeias de Markov em tempo discreto, com a característica de serem duplamente estocásticos [23], pois compreendem:

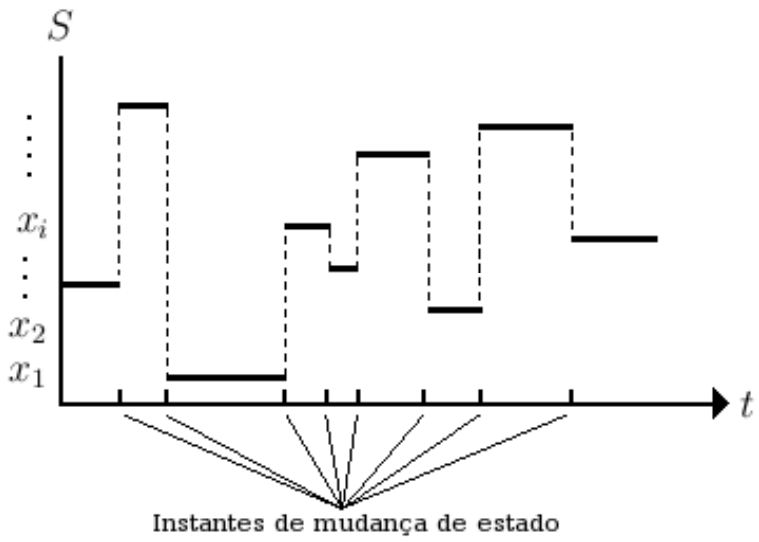

Figura 4.2: Características de uma cadeia de Markov com tempo contínuo.

- um processo estocástico *observável*, que consiste de um conjunto de saídas ou observações que são geradas, uma por cada estado, de acordo com uma função densidade de probabilidade (fdp);

- um processo estocástico *escondido* que consiste de uma cadeia de Markov.

Como exemplo, imaginemos o caso de um homem que recebe da esposa a tarefa de ir ao mercado e trazer várias frutas relacionadas em uma lista de compras. Após algum tempo, o homem retorna e apresenta à esposa as frutas compradas conforme a lista. Neste caso, as saídas ou observações são as frutas obtidas e é somente o que a dona de casa pode ver do processo. Não está explícito qual foi o caminho percorrido pelo homem através das finitas lojas e o que motivou esse caminho. Aspectos como distâncias, preços, quantidades e qualidades das frutas influenciaram no trajeto, e estes aspectos podem possuir características estocásticas. Esse é um exemplo de HMM.

Uma vez associado o modelo ao fenômeno, pode-se responder a perguntas tais como: qual é o melhor trajeto de visita pelas lojas? Qual trajeto possibilita a aquisição de toda a lista de compras de forma satisfatória? Estas perguntas constituem o que chamamos de problemas básicos dos HMMs, que serão discutidos mais à frente.

4.2.1 Variáveis envolvidas nos HMMs

As variáveis envolvidas em problemas modelados por HMMs são as seguintes:

- O número de estados do modelo: N.
- O estado no qual estamos no instante t: q_t.
- A probabilidade de começar o experimento no estado i: $\pi_i = P[q_{t_1} = i]$. Denotaremos o vetor das probabilidades iniciais de cada estado por $\pi = (\pi_1, \pi_2, \ldots, \pi_i, \ldots, \pi_N)$.
- A matriz de probabilidades das transições entre estados $A = (a_{ij}),$ onde $a_{ij} = P[q_{t+1} = j | q_t = i]$.
- O número dos diferentes símbolos de observação por estado: M.
- O conjunto discreto de possíveis símbolos de observação: $V = (v_1, v_2, \ldots, v_M)$.
- O símbolo observado no instante t: o_t.
- O tamanho da seqüência de observação: T.
- A seqüência de observação: $O = (o_1, o_2, \ldots, o_T)$.

- As distribuições de probabilidade dos símbolos de observação de cada estado $B = (b_1(k), b_2(k), \ldots, b_j(k), \ldots, b_N(k))$, onde $b_j(k)$ é a distribuição de probabilidade do símbolo k no estado j.

  - Se a distribuição é discreta: $b_j(k) = P[o_t = v_k | q_t = j]$;

  - Se a distribuição é contínua, geralmente será representada por um somatório ponderado de distribuições gaussianas:

$$b_j(o_t) = \sum_{l=1}^{L} c_{jl} N(o_t, \mu_{jl}, U_{jl}) \tag{4.9}$$

    onde $N(o_t, \mu_{jl}, U_{jl})$ é a mistura das distribuições de gaussianas, c_{jl} são os coeficientes de ponderação das L gaussianas, o_t é o vetor de observações no instante t, μ_{jl} é o vetor das médias e U_{jl} é a matriz de covariâncias.

A notação compacta para denotar um HMM é $\lambda = (A, B, \pi)$, sendo A a matriz de transições entre estados, B as distribuições de probabilidades de os símbolos de observação em cada estado e π o vetor de probabilidades iniciais de cada estado. Chamamos (A, B, π) de modelo λ.

Como um exemplo, aplicando a notação acima, podemos supor urnas escondidas atrás de uma cortina, contendo bolinhas de cores distintas. Alguém escolhe uma urna aleatoriamente e tira uma bolinha dizendo a cor dela. Outra pessoa, do outro lado da cortina, anota a cor. Esse exemplo está esquematizado na Fig. 4.3.

Onde:

- Urnas representam os estados do modelo: $N = 4$; sem ter em consideração os estados de Inicio (I) e Fim (F).

- π_i é a probabilidade de começar o experimento na urna i.

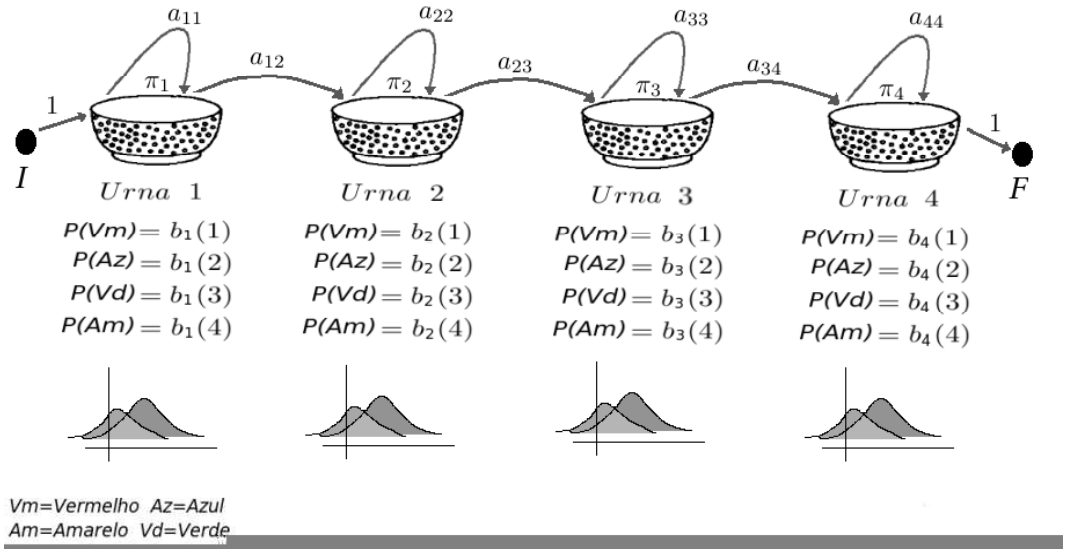

Figura 4.3: Exemplo de HMM usando urnas como estados.

- a_{ij} são as possíveis transições entre as urnas.
- v_k são os possíveis símbolos de observação, isto é, as cores das bolinhas.
- $b_j(v_k)$ são as probabilidades dos símbolos de observação.

No exemplo, consideramos uma fdp contínua para os símbolos de observação de cada estado, que com a finalidade de demostração, está representada pelo desenho de duas gaussianas debaixo de cada estado.

4.2.2 Os problemas básicos dos HMMs e suas soluções

Há três problemas chamados de problemas básicos dos HMMs. São eles:

- **Problema I ou de avaliação.** Dados um modelo e uma seqüência de observação, como se calcula a probabilidade de que a seqüência observada tenha sido produzida pelo modelo?

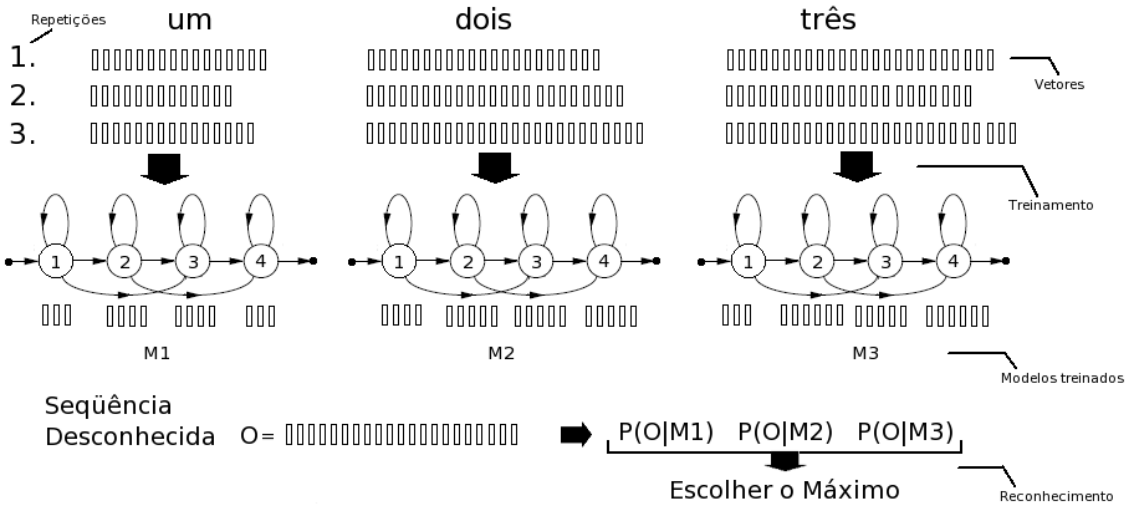

Figura 4.4: Relação entre o HMM, a voz e os três problemas dos HMMs.

- **Problema II ou de decodificação.** Qual a seqüência de estados "ótima" associada a uma seqüência observada dada?

- **Problema III ou de treinamento.** Este é o mais difícil dos três problemas. Trata de determinar um método para ajustar os parâmetros do modelo $\lambda = (A, B, \pi)$ para maximizar a probabilidade de a seqüência observada ter sido produzida por um dado modelo.

Na Fig. 4.4 podemos observar os três problemas dos HMMs em funcionamento com um exemplo prático onde vemos os vetores de caracteristicas da voz representados por rectangulinhos que sirvem de treinamento para os modelos, modificando a cada iteração a estatística do modelo. Também temos a fase de avaliação, na qual vai evaluar se uma seqüência de entrada desconhecida pertence ou não a um modelo dado.

4.2.3 Solução do Problema I ou de avaliação

Deseja-se, nesse caso, calcular a probabilidade de a seqüência observada ser $O = o_1, o_2, o_3, ..., o_T$, dado o modelo λ; ou seja, $P[O|\lambda]$.

A maneira mais simples para solucionar o problema consiste em considerar a seqüência de estados $Q = q_1, q_2, ...q_T$ onde q_1 é o estado inicial de onde gerou-se o símbolo de observação o_1. Então:

$$P[O|Q, \lambda] = \prod_{t=1}^{T} P[o_t|q_t, \lambda] \tag{4.10}$$

onde foi assumido que as observações são independentes estatisticamente. Temos:

$$P[O|Q, \lambda] = b_{q_1}(o_1).b_{q_2}(o_2)...b_{q_T}(o_T). \tag{4.11}$$

A probabilidade de a seqüência Q ter sido gerada pelo modelo λ pode ser escrita como

$$P[Q|\lambda] = \pi_{q_1}.a_{q_1q_2}.a_{q_2q_3}...a_{q_{T-1}q_T}. \tag{4.12}$$

Então, a probabilidade de O e Q serem gerados simultaneamente pelo modelo λ, é o produto das Eqs. 4.11 e 4.12. Temos,

$$P[O, Q|\lambda] = P[O|Q, \lambda].P[Q|\lambda]. \tag{4.13}$$

Portanto, a probabilidade de a seqüência O ser gerada dado o modelo λ é obtida pela Eq. 4.14:

$$\begin{aligned} P[O|\lambda] &= \sum_{\forall\, Q} P[O|Q, \lambda].P[Q|\lambda] \\ &= \sum_{q_1, q_2 ... q_T} \pi_{q_1} b_{q_1}(o_1) a_{q_1q_2} b_{q_2}(o_2)...a_{q_{T-1}q_T} b_{q_T}(o_T). \end{aligned} \tag{4.14}$$

Pode-se interpretar a equação acima da seguinte maneira:

Em $t = 1$, estamos no estado q_1, com probabilidade π_{q_1}, gerando o símbolo o_1, com probabilidade $b_1(o_1)$. Em $t = 2$, estamos no estado q_2 com probabilidade $a_{q_1q_2}$,

gerando o símbolo o_2, com probabilidade $b_{q_2}(o_2)$, e assim por diante até o último símbolo de observação.

A Eq. 4.14 requer $(2T-1)N^T$ multiplicações e N^T-1 somas (onde N é o número de estados e T o de observações). Este cálculo é computacionalmente inviável mesmo para pequenos valores de N e T ; havendo necessidade de um algoritmo mais eficiente. Felizmente, esse algoritmo existe e se chama **Método de Avanço e Retrocesso** (Forward-Backward).

Algoritmo Forward

Primeiro, é declarada uma variável chamada $\alpha_t(i)$ (Forward), definida por:

$$\alpha_t(i) = P[o_1 o_2 ... o_t, q_t = i|\lambda]. \tag{4.15}$$

Ou seja, $\alpha_t(i)$ é a probabilidade conjunta de a seqüência parcial de observação $o_1 o_2 ... o_t$ (até o instante t) ter sido gerada e o estado $q_t = i$ ocorrer, dado o modelo λ.

Pode-se resolver esse problema indutivamente como segue:

- Passo 1. Inicialização:

$$\alpha_1(i) = \pi_i b_i(o_1), \qquad 1 \leq i \leq N. \tag{4.16}$$

- Passo 2. Recursão:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1}), \qquad 1 \leq t \leq T-1. \tag{4.17}$$

- Passo 3. Terminação:

$$P[O|\lambda] = \sum_{i=1}^{N} \alpha_T(i). \tag{4.18}$$

O passo 1 inicializa as variáveis α como a probabilidade conjunta do estado i e da observação inicial o_1. O passo 2, que é o mais importante, mostra como o estado j pode ser alcançado no tempo $t+1$ desde os N possíveis estados, ver Fig. 4.5. Finalmente, o passo 3 fornece o valor desejado de $P[O|\lambda]$ como o somatório das variáveis $\alpha_T(i)$. Por definição,

$$\alpha_T(i) = P[o_1 o_2 ... o_T, q_T = i|\lambda], \tag{4.19}$$

onde $P[O|\lambda]$ é o somatório dos $\alpha_T(i)'s$.

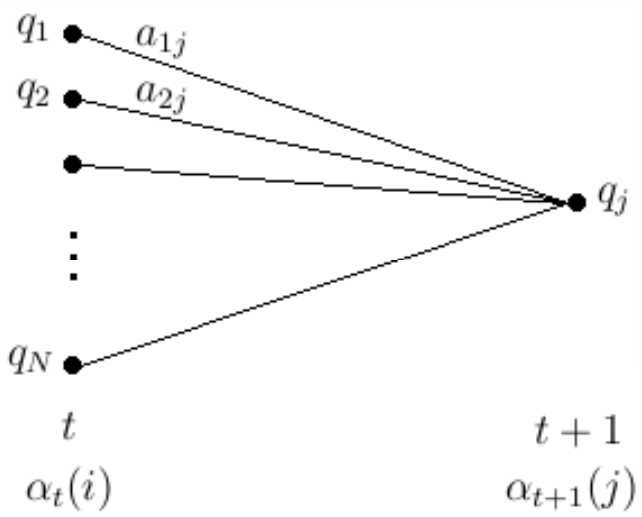

Figura 4.5: Seqüência de operações para o cálculo da variável (forward) $\alpha_{t+1}(j)$.

Se examinarmos a computação envolvida no cálculo de $\alpha_t(j)$, para $1 \leq t \leq T$ e $1 \leq j \leq N$, veremos que o mesmo requer exatamente $N(N+1)(T-1)+N$ multiplicações e $N(N-1)(T-1)$ somas.

Algoritmo Backward

Para resolver os problemas II e III, podemos considerar uma variável $\beta_t(i)$ ou Backward definida como:

$$\beta_t(i) = P[o_{t+1} o_{t+2} ... o_T|q_t = i, \lambda] \tag{4.20}$$

que é a probabilidade de a seqüência de observação parcial, desde $t+1$ até o final ter sido gerada, dado o estado $q_t = i$ e o modelo λ. Como foi feito para resolver a variável Forward (α_t), podemos resolver para $\beta_t(i)$, indutivamente como segue:

- Passo 1. Inicialização:

$$\beta_T(i) = 1, \qquad 1 \leq i \leq N. \tag{4.21}$$

- Passo 2. Recursão:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \tag{4.22}$$

$$\text{para} \qquad t = T-1, T-2, ..., 1 \quad ; \quad 1 \leq i \leq N.$$

- Passo 3. Terminação:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \tag{4.23}$$

$$\text{para} \quad ; \quad t = T-1, T-2, \ldots, 1 \;\; 1 \leq i \leq N.$$

No passo da inicialização, arbitrariamente define-se $\beta_T(i) = 1, \ \forall i$. O passo 2, que está ilustrado na Fig. 4.6, calcula o valor de $\beta_t(i)$ em função das N variáveis $\beta_{t+1}(j)$ do instante seguinte, em função das probabilidades de transição entre estados a_{ij} e em função das probabilidades de emissão do símbolo o_{t+1} desde os N estados j. Finalmente, o passo 3 calcula $P[O|\lambda]$ somando o valor das N variáveis $\beta_1(i)$ multiplicado pela probabilidade de que o estado i seja o estado inicial e pela probabilidade de que emita o primeiro símbolo de observação o_1.

4.2.4 Solução do Problema II ou de decodificação

A dificuldade da solução do problema II é a definição da seqüência ótima, devido ao fato que há muitos possíveis critérios de otimização. Por exemplo, um possível

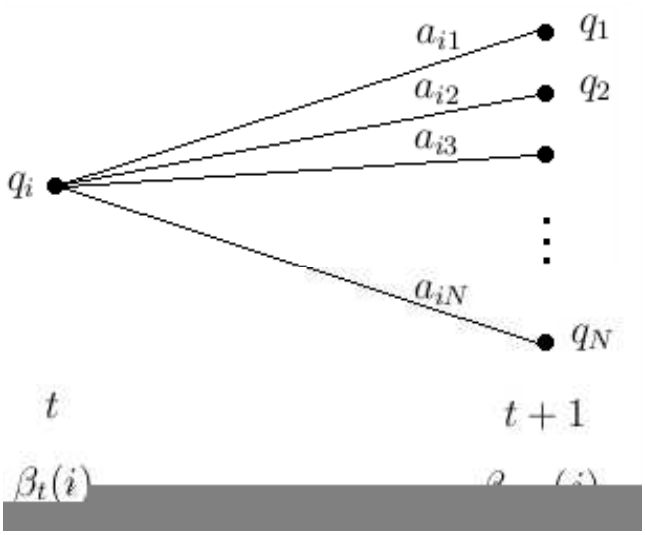

Figura 4.6: Seqüência de operações para o cálculo da variável (backward) $\beta_t(i)$

critério de otimização é escolher os estados q_t que são individualmente mais prováveis em cada instante de tempo. Este critério de otimização maximiza o número esperado dos estados individuais corretos.

Para implementar a solução do Problema II, definimos a variável:

$$\gamma_t(i) = P[q_t = i|O, \lambda] \tag{4.24}$$

Ou seja, a probabilidade de começar no estado i, no tempo t, dada a seqüência de observação O, e o modelo λ. A Eq. 4.24 pode ser expressa simplesmente em termos das variáveis α e β, por:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P[O|\lambda]} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \tag{4.25}$$

onde a variável $\alpha_t(i)$ armazena a probabilidade da seqüência parcial observada $o_1 o_2 ... o_t$, enquanto a variável $\beta_t(i)$ armazena a probabilidade de o resto da seqüência observada $o_{t+1} o_{t+2} ... o_T$ ser gerada, dado o estado $q_t = i$.

Usando $\gamma_t(i)$, podemos obter q_t^*, que é o estado individualmente mais provável

no instante t, como:

$$q_t^* = \arg\max_{1 \leq i \leq N}[\gamma_t(i)], \qquad 1 \leq t \leq T \tag{4.26}$$

onde o operador *arg* indica que se toma o índice i correspondente ao máximo da função $\gamma_t(i)$.

A Eq. 4.26 não funciona sempre. Por exemplo, quando o HMM tem transições entre estados iguais a zero ($a_{ij} = 0$), a seqüência ótima poderia não ser uma seqüência válida de estados. Isto deve-se ao fato da solução simplesmente determinar o estado mais parecido a cada instante sem considerar as probabilidades passadas.

O critério geralmente usado para encontrar a melhor seqüênca de estados é maximizar $P[Q|O, \lambda]$ ou, equivalentemente, maximizar $P[Q, O|\lambda]$.

Uma técnica formal usada é o algoritmo de Viterbi [7].

Algoritmo de Viterbi

Para encontrar a melhor seqüência de estados $Q = q_1 q_2 ... q_T$ para a seqüência de observação $O = o_1 o_2 ... o_T$ temos que definir a quantidade:

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P[q_1 q_2 ... q_t = i, o_1 o_2 ... o_t|\lambda] \tag{4.27}$$

onde $\delta_t(i)$ é o melhor *score* (a probabilidade mais alta) de se escolher o melhor caminho de estados, terminando no estado i, levando em conta as t primeiras observações. Então:

$$\delta_{t+1}(i) = \left[\max_{1 \leq i \leq N} \delta_t(i) a_{ij}\right] b_j(o_{t+1}) \tag{4.28}$$

Uma vez calculadas os $\delta_t(i)$ para todos os estados e para todos os instantes de tempo, a seqüência é construída para trás através de um caminho que memoriza o

argumento que maximizou a Eq. 4.28 para cada instante t e para cada estado j. Este caminho se armazena nas correspondentes variáveis $\psi_t(j)$, como segue:

- Passo 1. Inicialização:

$$\delta_1(i) = \pi_i b_i(o_1), \qquad 1 \leq i \leq N \tag{4.29}$$

$$\psi_1(i) = 0 \tag{4.30}$$

- Passo 2. Recursão:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(o_t), \qquad 2 \leq t \leq T \quad 1 \leq j \leq N \tag{4.31}$$

$$\psi_t(j) = \arg\max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \qquad 2 \leq t \leq T \quad 1 \leq j \leq N \tag{4.32}$$

- Passo 3. Terminação:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \tag{4.33}$$

$$q_T^* = arg \max_{1 \leq i \leq N} [\delta_T(i)] \tag{4.34}$$

- Passo 4. Construção para trás da seqüência de estados:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \qquad t = T-1, T-2, \ldots, 1 \tag{4.35}$$

Pode-se notar que o algoritmo de Viterbi é similar ao cálculo da variável Forward. A única diferença é que o somatório da Eq. 4.17 mudou com a maximização da Eq. 4.31 e a adição do passo final. A complexidade do algoritmo é da ordem de N^2T operações.

4.2.5 Solução do Problema III ou de treinamento

Não existe uma maneira conhecida de resolver analiticamente o conjunto de parâmetros do modelo que maximiza a probabilidade de a seqüência de observações ocorrer,

de uma maneira fechada. Entretanto, pode-se escolher $\lambda = (A, B, \pi)$ tal que sua probabilidade, $P[O|\lambda]$, é localmente maximizada usando um procedimento iterativo tal como o método de Baum-Welch também conhecido como o método EM (Expectation-Maximization), ou técnicas de gradiente [23]. O algoritmo de Baum-Welch, apresentado em termos das variáveis Forward e Backward, é utilizado na estimação das matrizes de transição e emissão [24].

Para uma única seqüência de observações $O = o_1, o_2, \ldots, o_T$, a re-estimação da probabilidade de transição do estado i para o estado j da matriz de transição de estados A é dada por

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i)}. \tag{4.36}$$

Também para uma única elocução, as funções de probabilidade dos símbolos de saída para os tipos de HMM discreto e contínuo são descritas a seguir.

- Para HMMs discretos

  No caso discreto, a quantidade de símbolos de saída é finita. A função de probabilidade re-estimada que um estado i emita um símbolo $o_t = v_k$ é obtida por:

  $$\bar{b}_i(k) = \frac{\sum_{\substack{t=1 \\ t.q.o_t=k}}^{T} \gamma_t(i)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{4.37}$$

  que possui as seguintes propriedades

  $$\bar{b}_i(k) \geq 0 \quad 1 \leq i \leq N, \quad 1 \leq k \leq K \tag{4.38}$$

  $$\sum_{k=1}^{K} \bar{b}_i(k) = 1 \tag{4.39}$$

- Para HMMs contínuos

  Neste tipo de HMM a função densidade de probabilidade (fdp) é contínua. Uma fdp muito utilizada é a mistura de Gaussianas. Em nosso caso, a fdp contínua apresenta-se na forma de mistura finita de Gaussianas, como definida na Eq. 4.9.

  No caso de HMMs contínuos, além da matriz A, os coeficientes de mistura c_{jl}, o vetor média μ_{jlx} e a matriz covariância U_{jl} também precisam ser re-estimados. Assim temos:

$$\bar{c}_{jl} = \frac{\sum_{t=1}^{T} \gamma_t(j,l)}{\sum_{t=1}^{T} \sum_{l=1}^{L} \gamma_t(j,l)} \tag{4.40}$$

$$\bar{u}_{jl} = \frac{\sum_{t=1}^{T} \gamma_t(j,l) o_t}{\sum_{t=1}^{T} \gamma_t(j,l)} \tag{4.41}$$

$$\bar{U}_{jl} = \frac{\sum_{t=1}^{T} \gamma_t(j,l)(o_t - \mu_{jl})(o_t - \mu_{jl})'}{\sum_{t=1}^{T} \gamma_t(j,l)}, \tag{4.42}$$

  para $1 \leq j \leq N$ e $1 \leq l \leq L$. O número total de observações da seqüência é especificado por T. A variável $\gamma_t(j,l)$ é a probabilidade de estar no estado j no instante t com o l-ésimo componente de mistura associado à observação o_t, ou seja,

$$\gamma_t(j,l) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \right] \left[\frac{c_{jl} N(o_t, \mu_{jl}, U_{jl})}{\sum_{r=1}^{L} c_{jr} N(o_t, \mu_{jr}, U_{jr})} \right]. \tag{4.43}$$

4.2.6 Tipos de HMMs: Classificação por transições

Os HMMs podem-se dividir de acordo com as transições entre estados da seguinte maneira:

- Modelo ergódico: todos os estados têm uma conexão entre eles; ou seja não há entradas nulas na matriz de transições: $a_i j \neq 0$. Fig. 4.7a)

- Modelo esquerda-direita: cada estado têm conexão com ele mesmo e com os seguintes. Fig. 4.7b)

  Obs: São os mais usados para reconhecimento de voz e locutor.

- Modelo esquerda-direita paralelo: igual ao anterior porém há possibilidade de usar outros caminhos. Fig. 4.7c)

  Obs: Usado para reconhecer voz contínua.

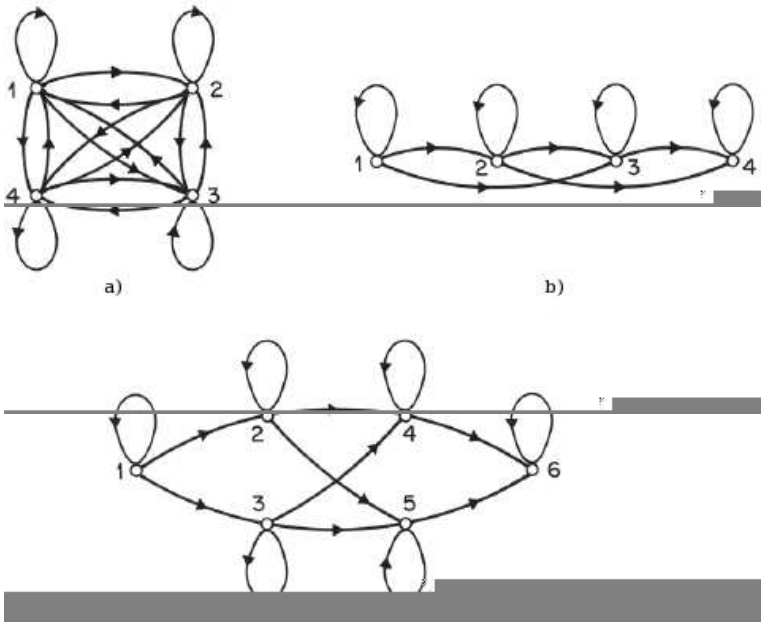

Figura 4.7: Tipos de HMMs classificação por transições: a) Ergódico, b) Esquerda-direita, c) Esquerda-direita paralelo

4.3 Programa HTK (HMM Tool Kit)

O HTK foi a primeira ferramenta computacional construída para reconhecimento de voz, baseada em modelos ocultos de Markov, disponível para a comunidade científica através do site do Departamento de Engenharia da Universidade de Cambrigde (Cambridge University Engineering Department - CUED) [31]. Em 1993, os direitos foram adquiridos pela Entropic Research Laboratory Inc. e, posteriormente, pela Microsoft Corporation que licenciou o HTK novamente para o CUED. O HTK foi basicamente projetado para a construção de HMMs baseados em ferramentas de processamento de fala, mais especificamente, no uso de sistemas de reconhecimento de fala, embora possa ser também aplicado no reconhecimento de caracteres, de locutor, no sequenciamento de DNA, etc. O HTK engloba dois principais estágios de processamento:

- As ferramentas de treinamento, usadas na estimação dos parâmetros do HMM por meio da base de treinamento e seus respectivos *scripts*.
- As ferramentas de reconhecimento que vão processar a base de sinais de teste e realizar o reconhecimento da mesma.

4.3.1 Arquitetura do HTK

A arquitetura do HTK correspondente à versão 3.3 pode ser resumida [30] conforme a Fig. 4.8, que ilustra todas as ferramentas utilizadas pelo HTK no processamento e reconhecimento do sinal de voz.

As ferramentas do HTK funcionam através de linhas de comando e cada ferramenta possui um certo número de argumentos principais e outros opcionais, que são

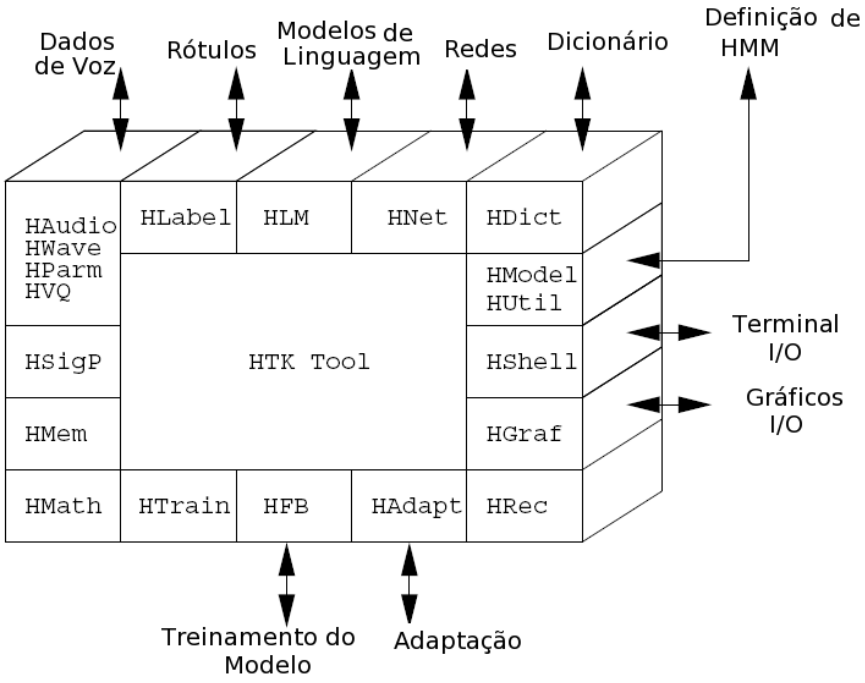

Figura 4.8: Arquitetura do HTK.

sempre precedidos pelo sinal menos (-). Os argumentos opcionais podem ser letras maiúsculas ou minúsculas e ser seguidas ou não de um número inteiro, um número real ou de uma cadeia de caracteres.

O exemplo a seguir mostra a linha de comando para executar a ferramenta fictícia HFoo [30].

HFoo -T 1 -f 34.3 -a -s meu arquivo arquivo1 arquivo2

Neste exemplo os argumentos principais são os arquivos chamados arquivo1 e arquivo2. Os argumentos opcionais são -T, -f, -a e -s. Com exceção do argumento opcional -a, todos são seguidos, respectivamente, por um número inteiro, um número real e por uma string (meu arquivo). Por não ser seguido por nenhum valor, o argumento opcional -a é usado como um flag para habilitar ou desabilitar alguma característica da ferramenta HFoo. Os argumentos opcionais que são letras

maiúsculas como -T, por exemplo, possuem a mesma função para todas as ferramentas do HTK. Arquivos contendo configurações específicas também podem ser usados para o controle das ferramentas do HTK. Para se obter um resumo da linha de comando e das opções utilizadas em qualquer ferramenta do HTK basta executar a ferramenta desejada, sem argumentos, no prompt.

Capítulo 5

Reconhecimento automático de voz e locutor

Atualmente, observa-se um amplo potencial de utilização tanto de reconhecimento de voz (o que se está falando) quanto de locutor (quem está falando). As principais aplicações são nas áreas de Informática, de Telecomunicações e na área comercial. Portanto, as diversas técnicas utilizadas tanto para o reconhecimento automático de voz (RAV) como para o reconhecimento automático de locutor (RAL) estão sendo estudadas e melhoradas cada vez mais; citemos como exemplo os avanços nas técnicas utilizadas para extração de características dos sinais de voz.

Neste capítulo, será feita uma comparação entre as técnicas ZCPA e MFCC, utilizando HMMs. A novidade está na comparação entre essas técnicas em ambientes ruídosos na aplicação de RAL [8][24][19][2][27].

5.1 Bases de áudio utilizadas

Duas bases de vozes, em português brasileiro, foram utilizadas: uma base de dígitos para RAV e outra de frases para RAL. Essas bases foram obtidas com o apoio do Instituto Militar de Engenharia (IME), produzidas em ambiente de laboratório e com a participação dos alunos de mestrado e de graduação do IME. Foi usada ainda uma outra base de dígitos chamada YOHO. Esta base é a primeira base de grande escala (científicamente controlada e coletada), de alta qualidade, feita para testes em RAL, com um alto grau de confiança. Esta base foi coletada pelo ITT e está disponível na Linguistic Data Consortium (University of Pensylvania).

5.1.1 Base de dígitos

A base de dígitos está estruturada para a realização de experimentos de RAV. Consta de 50 locutores femininos e 50 locutores masculinos, dos quais cada um deles repete três vezes as palavras: “zero”, “um”, “dois”, “três”, “quatro”, “cinco”, “seis”, “meia”, “sete”, “oito”, “nove”. Cada gravação tem uma taxa de amostragem de 11025 Hz e 16 bits de resolução com um só canal e gravados em ambiente de escritório. A formatação dos nomes dos arquivos é como segue:

D0R1LF01.wav

onde D representa a palavra dígito; 0 é o dígito gravado; R representa a palavra repetição; 1 é o número de repetições; L representa a palavra locutor; $F(M)$ representa a palavra feminino (masculino); 01 é o número do locutor; logo um arquivo de nome D2R3LM02.wav contém a terceira repetição do dígito 2 pronnciada pelo locutor masculino 02. A base está desenvolvida com o propósito de fazer um reconhecimento de voz dependente do texto.

5.1.2 Base de frases

Esta base tem estrutura adequada para a realização de experimentos de RAL. Contém 25 locutores (17 homens e 8 mulheres), onde cada locutor fala duas frases: E1 - "O prazo está terminando", a qual é predominantemente composta por fonemas orais e E2 - "Amanhã ligo de novo", onde a predominância é por fonemas nasais. Cada locutor repetiu 60 vezes cada uma das 2 frases. Cada gravação tem uma taxa de amostragem de 8000 Hz e 16 bits de resolução com um só canal e gravados em ambiente de escritório. A formatação dos nomes dos arquivos é como segue:

CR10E1LF1.wav

onde C representa a palavra cortada (com end-points); R representa a palavra repetição; 10 é o número de repetições; $E1$ representa a frase gravada; L representa a palavra locutor; $F(M)$ representa a palavra feminino (masculino); 1 é o número do locutor.

5.1.3 Base YOHO

É uma base de dígitos em inglês que tem a sintaxe de combinação de sequência de dígitos; isto é, uma frase de 3 cifras de 2 dígitos cada. Por exemplo, "twenty-six, eighty-one, fifty-seven". Cada frase consta de 6 dígitos aleatórios, sendo: as unidades podem ser os números 1 até o 9, sem considerar o "oito"nem o "zero"; as dezenas começam em 20 e vão até o 90.

A base YOHO tem 138 locutores dos quais 106 são homens e 32 mulheres, e foi colectada num período de 3 meses em ambiente real de escritório. A base de treinamento consta de 4 sessões por locutor e cada sessão tem 24 sequências. A base

de teste consta de 10 sessões por locutor e cada sessão tem 4 frases. No total, são 1380 arquivos válidos, amostrados com uma freqüência de 8000Hz com uma largura de banda analógica de 3.8kHz. No total, são 1.2 gigabytes de dados.

5.1.4 Base de ruídos NOISEX

Nesta dissertação foram utilizadas 3 tipos de ruídos da base NOISEX: ruído branco, ruído de fábrica e ruído de balbuceios ou "babble". Esta base foi produzida pelo "Institute for Perception-TNO from Netherlands"e "Speech Research Unit, RSRE from United Kingdom". Todos os ruídos foram adquiridos com uma taxa de amostragem de 19.98kHz e 16 bits de resolução [32].

O ruído branco foi adquirido amostrando um gerador de ruído analógico de alta qualidade e apresenta igual energia em todas as freqüências do espectro. O ruído de fábrica foi gravado perto de equipamentos que cortam prata e de soldadura elétrica. E por último, a fonte do ruído babble são 100 pessoas falando numa cantina onde o nível de som durante a gravação foi de 88dBA. Os espectros destes ruídos estão na Fig. 5.1.

5.2 Reconhecimento de voz: dígitos conectados usando MFCC

Esta seção está dividida em duas partes. A primeira apresenta um reconhecedor de dígitos isolados, que vai servir para testar e definir os parâmetros tanto para a extração como para os HMMs e, também, serve como base para a segunda parte. A segunda apresenta um reconhecedor de dígitos conectados.

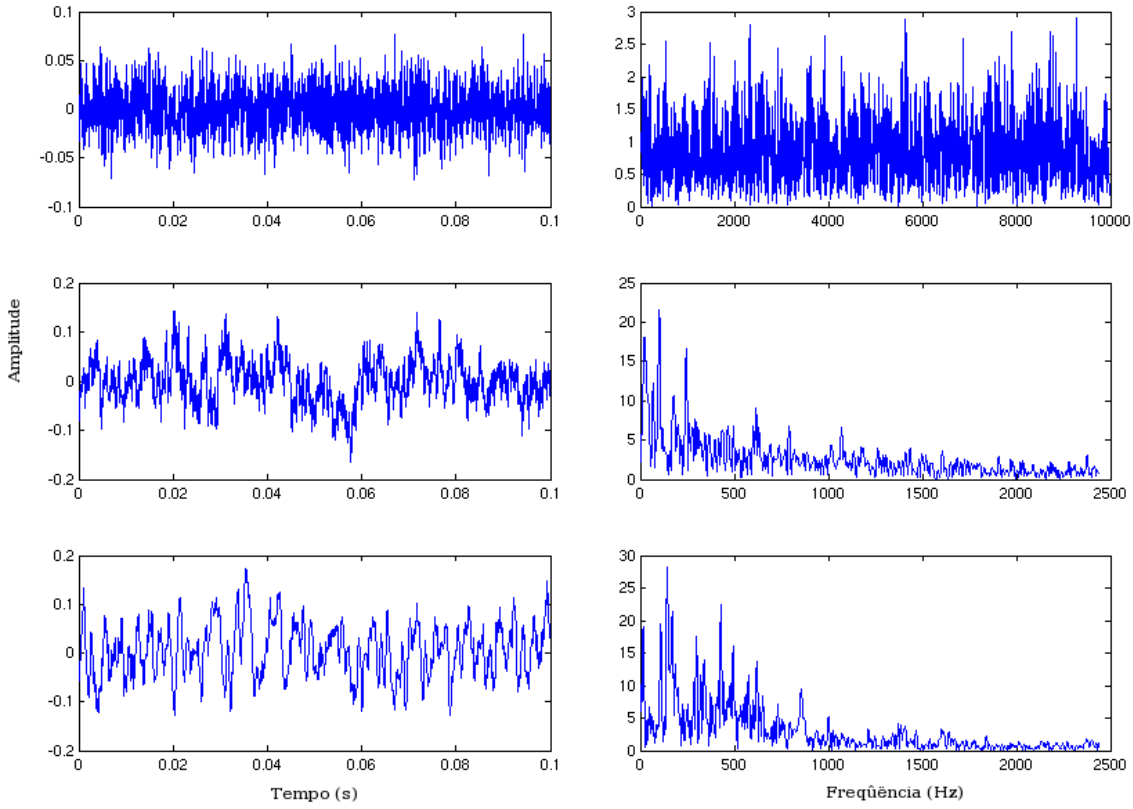

Figura 5.1: Forma de onda e espectro para 50ms de ruído: a) Branco; b) Fábrica e c) Babble.

5.2.1 Reconhecimento de dígitos isolados

No início, é necessário determinar ou definir a base com a qual se vai trabalhar. O processo de reconhecimento consta de duas fases, uma de treinamento dos HMMs e outra de teste.

Para a parte de treinamento, utilizou-se o software HTK para treinar os HMMs. Levando em conta que vamos reconhecer dígitos isoladamente, é necessário treinar um HMM por cada dígito dos locutores que serviram para o treinamento. Tais dígitos vão estar contidos em listas de treinamento (uma lista por dígito).

O processo de treinamento inicia-se pela extração das características MFCC da base inteira. Este processo é feito usando uma função específica do HTK que utiliza uma lista contendo todas as repetições dos dígitos dos locutores, tanto de treinamento como de teste, e um arquivo de configuração, o qual vai conter os parâmetros que servem para extrair os atributos MFCC. As características, já extraídas, ficam armazenadas em arquivos (um arquivo por dígito repetido) que são agrupados em uma pasta independente.

Depois de feita a extração, passa-se ao treinamento própriamente dito. Para isto, precisa-se de um arquivo chamado *protótipo*; que vai conter um protótipo de um HMM, consistindo na definição das variáveis envolvidas. Com um protótipo adequado e com as listas de treinamento, um HMM é inicializado usando funções específicas do HTK. Várias iterações são realizadas para modificar os valores das variáveis; uma vez que as mesmas não mudem mais de valor, ou seja, cheguem a uma convergência (que por *default* é de 0.0001 de diferência), o treinamento é finalizado!. Até aqui, temos todos os HMMs (um por dígito) treinados e prontos para serem usados.

Na parte de teste, o HTK requer uma lista com todas as repetições dos locutores que serviram de teste (lista de teste). Também requer um arquivo contendo um dicionário (lista dos dígitos), uma lista dos HMMs e um arquivo contendo a gramática empregada, a qual cada palavra e as transições entre elas são detalhadas. Esses arquivos vão servir como entrada para uma função do HTK e a saída gerada correspondente vai ser armazenada em outro arquivo chamado "results", o qual vai conter o HMM ganhador (por repetição) como também a verossimilhança.

Este arquivo resultado vai servir para analisar e criar novas formas de expressar o resultado obtido. Uma delas é a *matriz de confusão*, a qual vai conter quais dígitos foram confundidos com outros e quais foram acertados. Esta matriz é de muita ajuda para a análise do resultado. A seguir, todo o processo é detalhado.

Preparação da base

Para esta tarefa, utilizou-se a metade da base de dígitos para treinamento dos HMMs e a outra metade para teste. Assim, dispõe-se a base da seguinte maneira:

- Do locutor 1 até o 25 serviram para treinamento dos HMMs.
- Do locutor 26 até o 50 serviram para testes.
- Foram levados em conta os locutores femininos e masculinos conjuntamente.
- Para cada dígito há 150 repetições no total (25(loc.)x2(masc. e fem.)x3(rep.)).
- No total, há 1650 arquivos para treinamento e 1650 para teste (150(rep.)x11(díg.)).

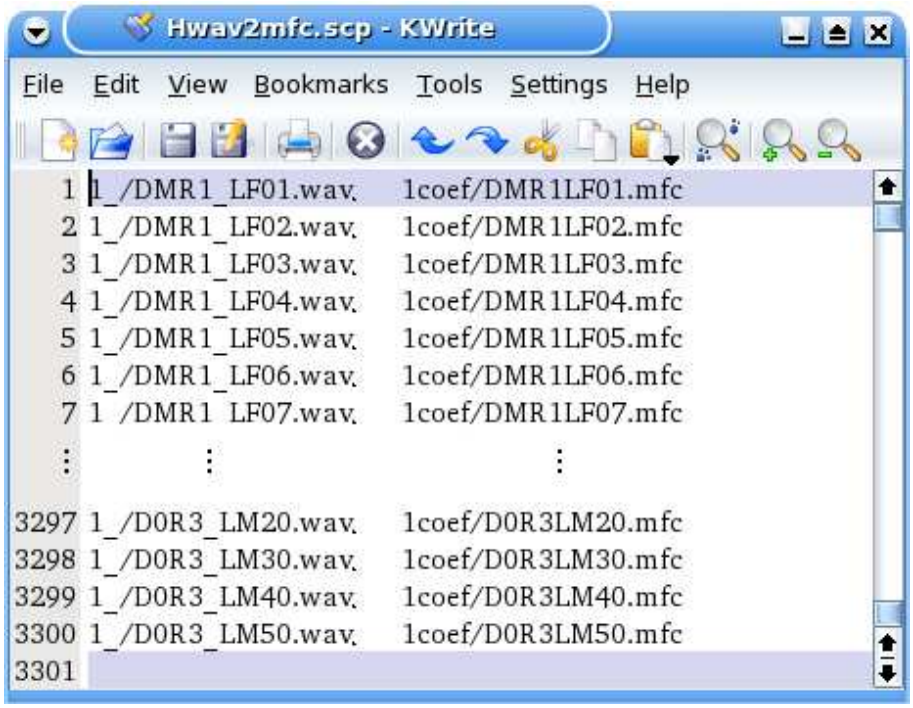

Figura 5.2: Fragmento da lista de extração: Hwav2mfc.scp

Criação da lista para extração das características

Nesta parte, uma lista será preparada contendo todos os arquivos da base inteira. Esta lista vai servir como entrada ao HTK, para a extração das características MFCC. A lista contém duas colunas, a primeira com os nomes dos arquivos em formato WAV da base inteira e a segunda com seus respectivos arquivos, já extraídos, em formato MFC. Para se ter uma idéia, pode-se ver na Fig. 5.2 como é feita a lista de extração. Para este propósito, usou-se o software MATLAB para criar a lista de uma maneira eficiente.

Criação do arquivo de configuração: config01

Este arquivo vai conter os parâmetros (um por linha) necessários para que o HTK possa extrair as características MFCC dos arquivos de áudio como é mostrado na Fig. 5.3. Para este experimento, usaram-se os coeficientes MFCC. Aplicou-se uma

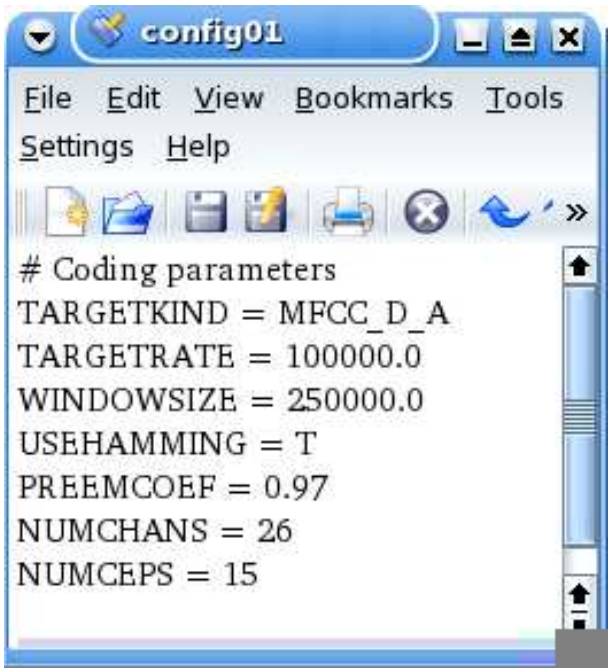

Figura 5.3: Arquivo de configuração para a extração dos coeficientes MFCC: config01

janela de Hamming de $25ms$ de duração e com $10ms$ de superposição. O pré-processamento foi feito com um filtro de pre-ênfase com um coeficiente igual a 0.97. Logo, escolheram-se 26 canais ou bandas para o banco de filtros na escala Mel, dos quais usaram-se 15 coeficientes cepstrais. Todos esses parâmetros foram obtidos da literatura encontrada, escolhendo-se os valores mais típicos.

Extração das características dos dígitos

Neste passo, as características dos arquivos de áudio dos dígitos são extraídos usando a base inteira. Para isso, a função HCopy do HTK é usada da seguinte maneira:

HCopy -T 1 -C config01 -S Hwav2mfc.scp

onde -T 1 representa uma opção geral do HTK que indica mostrar avanço da tarefa; $-C$ indica o arquivo de configuração usado; $-S$ indica o arquivo que contém a lista dos arquivos que formam a base inteira.

Depois de executar esta função, o HTK extrai as características de todos os arquivos de toda a base e coloca-os em uma pasta especificada na lista para a extração antes explicada.

Criação do HMM protótipo

Com o fim de treinar um HMM por dígito, é necessário ter um protótipo (ou molde), de onde se terá toda a informação para construir um HMM. Esse protótipo pode ser armazenado como um arquivo de texto simples e sua função é descrever a forma e a topologia de um HMM. É nesse protótipo que se tem a informação de quantos estados vai ter um HMM, quantas misturas de gaussianas, o tamanho do vetor de observações, a matriz de transições entre estados, etc. Um exemplo é mostrado na Fig. 5.4.

Preparação das listas de treinamento

As listas de treinamento, como explicado antes, consistem em arquivos contendo todos os arquivos de áudio, de um só dígito, pertencentes aos locutores que servem para o treinamento. Como exemplo, temos na Fig. 5.5 como é feita a lista para o dígito “oito”.

Treinamento dos HMMs

Para o treinamento dos HMMs, o HTK tem duas ferramentas: uma delas inicializa os HMMs e faz grande parte do treinamento e a outra faz um refinamento do que já foi treinado. Vamos ver agora o que cada uma dessas ferramentas faz com os HMMs e também quais arquivos e listas precisam.

```
~h "hmm1"
<BeginHMM>
    <VecSize> 4 <MFCC>
    <NumStates> 5
    <State> 2
        <Mean> 4
            0.2 0.1 0.1 0.9
        <Variance> 4
            1.0 1.0 1.0 1.0
    <State> 3
        <Mean> 4
            0.4 0.9 0.2 0.1
        <Variance> 4
            1.0 2.0 2.0 0.5
    <State> 4
        <Mean> 4
            1.2 3.1 0.5 0.9
        <Variance> 4
            5.0 5.0 5.0 5.0
    <TransP> 5
        0.0 0.5 0.5 0.0 0.0
        0.0 0.4 0.4 0.2 0.0
        0.0 0.0 0.6 0.4 0.0
        0.0 0.0 0.0 0.7 0.3
        0.0 0.0 0.0 0.0 0.0
<EndHMM>
```

Figura 5.4: Exemplo de um protótipo de HMM.

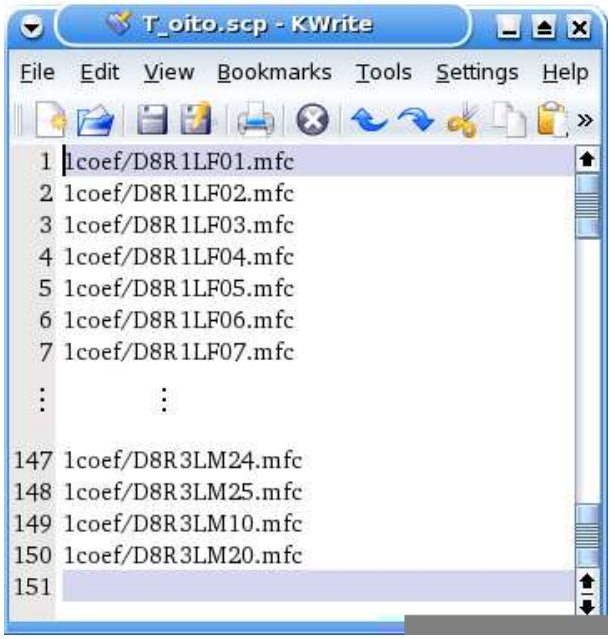

Figura 5.5: Exemplo da lista de treinamento para o dígito "oito".

- **HInit:** Esta ferramenta pode ser implementada como um processo iterativo, como mostrado na Fig. 5.6. Após as etapas de inicialização, o algoritmo de Viterbi é usado para encontrar a mais provável seqüência de estados correspondente a cada seqüência de treinamento; depois, os parâmetros do HMM são estimados. Como um efeito secundário, pode ser computada a verossimilhança. Portanto, o processo de estimação inteiro pode ser repetido até que a verossimilhança não seja mais incrementada.

  Este processo requer alguns valores iniciais dos parâmetros do HMM para começar. Para resolver este problema, HInit começa segmentando uniformemente a seqüência de treinamento e associa cada segmento sucessivo com sucessivos estados.

  Esta ferramenta foi utilizada da seguinte maneira:

  HInit -o oito -S T_oito.scp Proto10

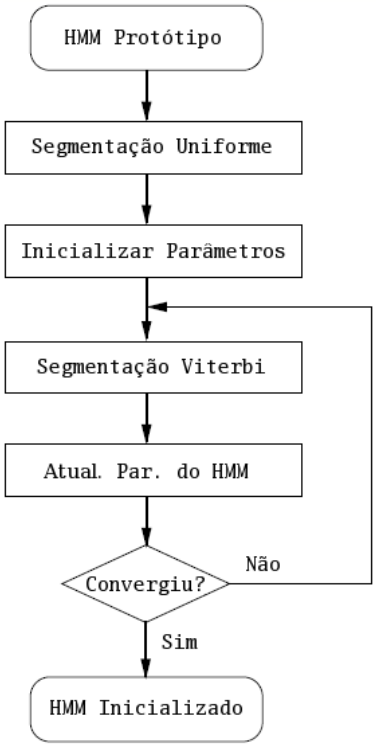

Figura 5.6: Diagrama de fluxo para o HInit.

onde, *-o oito* indica o arquivo de saída, *-S T_oito.scp* indica o nome da lista que vai servir para teinar o modelo e *Proto10* indica o arquivo que contém o protótipo do HMM.

- **HRest:** É usualmente aplicado diretamente aos modelos gerados por HInit. A operação desta ferramenta é similar ao HInit exceto que, como mostrado na Fig. 5.7, usa como entrada um HMM já inicializado e utiliza o algoritmo de Baum-Welch como re-estimação, em lugar do algoritmo de Viterbi. O algoritmo de Baum-Welch objetiva encontrar a probabilidade de estar em cada estado a cada instante de tempo, usando o algoritmo Forward-Backward. Assim, o treinamento com o algoritmo de Viterbi faz uma decisão grosseira de qual estado cada vetor de treinamento foi "gerado", enquanto o algoritmo de Baum-Welch toma uma desisão mais sofisticada.

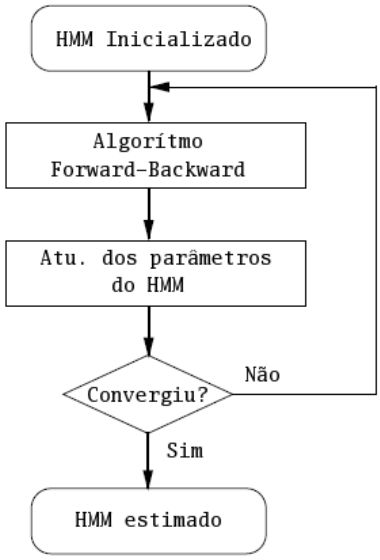

Figura 5.7: Diagrama de fluxo para o HRest.

A utilização desta ferramenta é usada como segue:

HRest -S T_oito.scp oito

onde, *-S T_oito.scp* indica a lista de treinamento e *oito* indica o HMM que está sendo re-estimado. Podemos ver que não há necessidade do protótipo.

Criação da lista de teste

A partir daqui, começa a fase de teste. Nesta parte, vai ser criado um arquivo contendo a lista dos arquivos que vão servir para teste. A lista compreende uma coluna, onde estão os nomes dos arquivos que contém as características MFCC dos arquivos de áudio como é mostrado na Fig. 5.8.

Criação do dicionário

O dicionário contém, em ordem alfabética, uma lista dos nomes das palavras a serem reconhecidas e ao seu lado a sua pronunciação. Como em nosso caso vamos

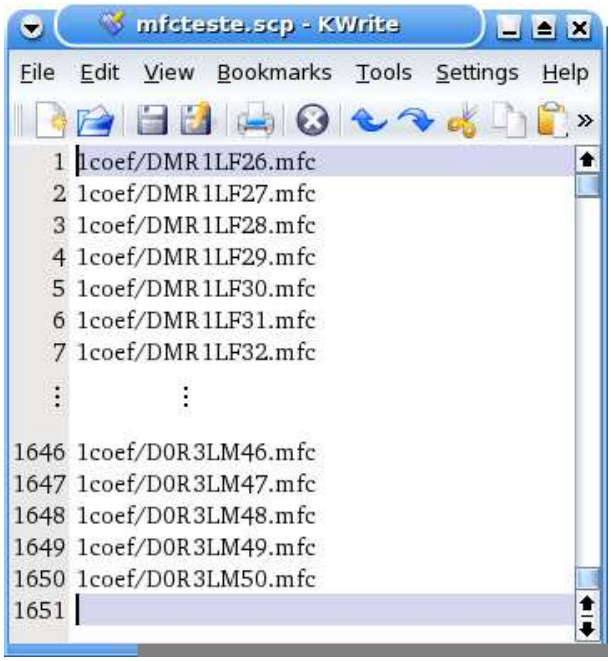

Figura 5.8: Exemplo da lista de teste.

reconhecer palavras inteiras e não fonemas, o nosso dicionário é muito simples, como é mostrado na Fig. 5.9.

Criação da lista dos HMMs

Trata-se de um arquivo que contém uma lista de todos os HMMs já treinados que vão participar dos testes. A lista tem que ser ordenada alfabeticamente e sua aparência é similar ao dicionário (Fig. 5.9) exceto que só tem uma coluna.

Criação da "rede"

Nesta parte, é definida uma "rede"de palavras que serão reconhecidas. A estrutura da rede vai depender da gramática e do que queremos reconhecer. Como em nosso caso queremos reconhecer só dígitos isolados, então a rede será muito simples, como mostrado na Fig. 5.10. Para esta tarefa, existe uma linguagem que simplifica a criação da rede. Esta linguagem consiste em uma série de símbolos alguns dos quais

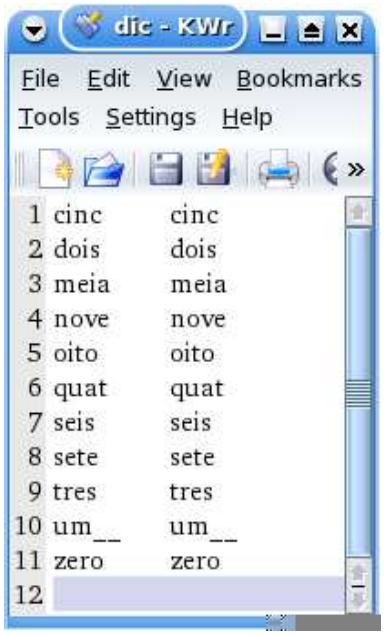

Figura 5.9: Exemplo do dicionário empregado.

detalham-se a continuação [30]:

- O símbolo "$" denota a utilização de uma variável.
- O símbolo "|" denota alternativas.
- Os símbolos "[ ]" denota ítems opcionais.
- Os símbolos "{ }" denota zero o mais repetições.
- Os símbolos "< >" denota uma o mais repetições.

Uma vez que o arquivo rede é criado, procede-se à geração da "net", que é simplesmente a conversão do arquivo rede para uma linguagem mais inteligível para o HTK. Essa conversão (compilação) é feita com a função HParse, como segue.

HParse rede net

onde, *rede* é o arquivo de entrada e *net* é o arquivo de saída gerado pelo HTK.

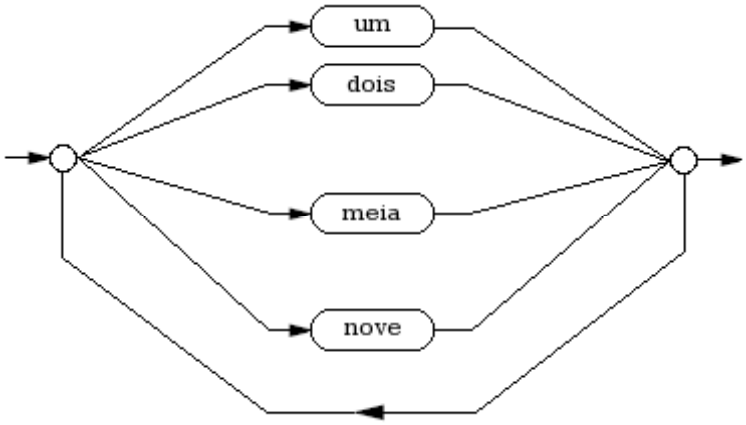

Figura 5.10: Exemplo da rede para dígitos isolados.

Reconhecimento dos arquivos de teste

Agora que estão criados os arquivos e listas necessários para o reconhecimento, passaremos a analisar a função HVite do HTK. Esta função é um reconhecedor de palavras de propósito geral, que, compara cada arquivo de voz (da lista de arquivos de teste) com todos os HMMs dando como resultado o início e o fim de cada palavra, a verossimilhança e o HMM ganhador por arquivo. A função HVite foi utilizada da seguinte maneira:

HVite -w net -S mfcteste.scp -i result dic lista

onde *-w net* indica o nome do arquivo onde esta representada a rede, *mfcteste.scp* denota o arquivo de teste que contém os arquivos de atributos a serem testados, -i result indica o arquivo que vai contêr os resultados e *dic lista* indicam o dicionário e a lista de HMMs empregados. Depois de obter o arquivo results, precisamos criar a matriz de confusão para ter uma maneira de analizar os resultados mais objetiva.

Para este fim, utilizaremos a função HResults do HTK. Esta função é muito útil pois ela usa os resultados e cria uma matriz onde estão representados os acertos e também, aqueles dígitos que foram confundidos com outros; assim como também

a taxa de acerto geral por cada dígito. A seguir os resultados obtidos para o reconheciento de dígitos isolados são apresentados.

5.2.2 Resultados do reconhecimento de dígitos isolados

Para o reconhecimento de dígitos isolados, escolheu-se um HMM do tipo esquerda-direita de 10 estados, com 3 misturas de gaussianas por estado, e com 2 saltos entre estados. Também usaram-se 15 coeficientes MFCC com seus respectivos coeficientes dinâmicos (Δ *e* $\Delta\Delta$), inicialmente sem considerar o primeiro coeficiente cepstral (c_0) e, depois considerando-o. Os resultados obtidos considerando ou não o c_0 são mostrados em matrizes de confusão [30]:

Tabela 5.1: Reconhecimento de dígitos isolados: Acertos=99.76% Total Acertos=1646 Erros=4 Total=1650 (sem c_0)

Fala. \ Dic.	zero	um	dois	três	quatro	cinco	seis	meia	sete	oito	nove
zero	150	0	0	0	0	0	0	0	0	0	0
um	0	149	0	1	0	0	0	0	0	0	0
dois	0	0	150	0	0	0	0	0	0	0	0
três	0	0	0	147	0	0	3	0	0	0	0
quatro	0	0	0	0	150	0	0	0	0	0	0
cinco	0	0	0	0	0	150	0	0	0	0	0
seis	0	0	0	0	0	0	150	0	0	0	0
meia	0	0	0	0	0	0	0	150	0	0	0
sete	0	0	0	0	0	0	0	0	150	0	0
oito	0	0	0	0	0	0	0	0	0	150	0
nove	0	0	0	0	0	0	0	0	0	0	150

Para a Tab. 5.1, o número de erros se deve principalmente à confusão entre o dígito "três"com o "seis", isto é de certa forma comum devido às semelhanças nos fonemas desses dígitos em português. Já na Tab. 5.2, o erro total diminui para 2

Tabela 5.2: Reconhecimento de dígitos isolados: Acertos=99.88% Total Acertos=1648 Erros=2 Total=1650 com c_0

Fala. \ Dic.	zero	um	dois	três	quatro	cinco	seis	meia	sete	oito	nove
zero	150	0	0	0	0	0	0	0	0	0	0
um	0	149	0	0	0	0	0	1	0	0	0
dois	0	0	150	0	0	0	0	0	0	0	0
três	0	0	0	149	0	0	1	0	0	0	0
quatro	0	0	0	0	150	0	0	0	0	0	0
cinco	0	0	0	0	0	150	0	0	0	0	0
seis	0	0	0	0	0	0	150	0	0	0	0
meia	0	0	0	0	0	0	0	150	0	0	0
sete	0	0	0	0	0	0	0	0	150	0	0
oito	0	0	0	0	0	0	0	0	0	150	0
nove	0	0	0	0	0	0	0	0	0	0	150

dígitos, graças à inclusão do primeiro coeficiente cepstral c_0. Esta tabela mostra a importância de considerar o c_0 para o reconhecimento de voz (dígitos), pois o c_0 contém informação da componente DC do sinal, e faz que o reconhecimento seja de melhor qualidade.

5.2.3 Reconhecimento de dígitos conectados

O reconhecimento foi feito utilizando a mesma base de voz do caso anterior. Os modelos já treinados do processo anterior são mantidos e agrega-se mais um modelo, relativo ao silêncio. A seguir, temos as principais mudanças com respeito ao processo anterior.

Figura 5.11: Exemplo da rede para dígitos conectados.

Criação do modelo do silêncio

Para a criação do modelo do silêncio, utilizou-se um protótipo diferente. Com um só estado e uma só gaussiana, o protótipo é treinado utilizando-se 5 repetições de silêncios ideais criados com ajuda do MATLAB. O treinamento foi feito da mesma maneira já explicada, resultando em um modelo já treinado chamado de "sil". Como foi adicionado um HMM, várias modificações foram feitas correspondentes a cada arquivo e lista antes descritos.

Concatenação

A concatenação dos dígitos isolados e o silêncio foi feita da seguinte maneira:

sil dig1 dig2 sil dig3 dig4 sil dig5 dig6 sil

onde cada dígito é escolhido aleatóriamente sem incluir o dígito "meia". Também é adicionado o silêncio ao início, a cada dois dígitos e ao final. Isto nos leva a modificar o arquivo "rede"antes visto, já que agora se tem uma nova estrutura; portanto, o arquivo rede fica como mostrado na Fig. 5.11.

Reconhecimento dos arquivos concatenados

Nesse caso, utilizou-se a função HVite do HTK. Com os arquivos necessários e adequados procede-se à execução de HVite, dando como resultado um arquivo chamado "results". Este arquivo vai conter os inícios e finais no tempo por cada arquivo que foi concatenado, assim como também suas respectivas verossimilhanças. Na Fig. 5.12, temos um exemplo de como está construído esse arquivo. Depois que for gerado o arquivo "results", ele é processado para obter-se a matriz de confusão, usando-se para isto o MATLAB. Na Tab. 5.3, apresenta-se a matriz de confusão para dígitos conectados, assim como a taxa de acerto usando a base clara de dígitos. Pode-se ver que a taxa de acerto diminuiu para 98% devido à mesma concatenação entre dígitos, já que se torna um pouco mais difícil a segmentação; assim como também o reconhecimento entre os dígitos que apresentam fonemas similares tais como o "seis"com o "três".

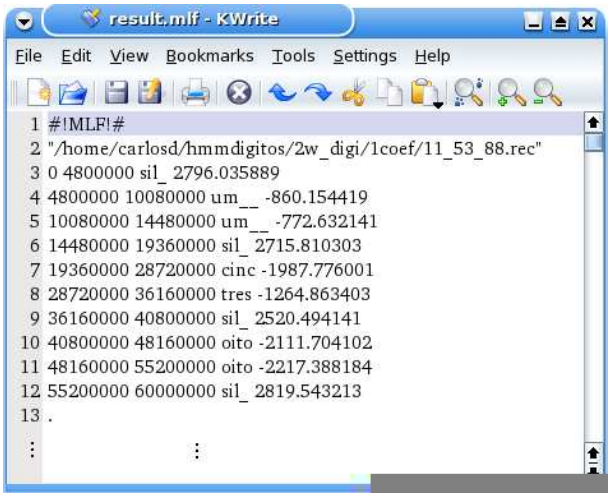

Figura 5.12: Exemplo do arquivo de resultados gerado pelo HTK.

Tabela 5.3: Matriz de confusão para dígitos conectados (com c_0).

Fala. \ Dic.	zero	um	dois	três	quatro	cinco	seis	sete	oito	nove
zero	22	0	0	0	0	0	0	0	0	0
um	0	32	0	1	0	0	0	0	0	1
dois	0	0	27	0	0	0	0	0	0	0
três	0	0	0	34	0	0	3	0	0	0
quatro	0	0	0	0	32	0	0	0	0	0
cinco	0	0	0	0	0	38	0	0	0	0
seis	0	0	0	0	0	0	29	0	0	0
sete	0	0	0	0	0	0	0	28	0	0
oito	0	0	0	0	0	0	0	0	33	0
nove	0	0	0	0	0	0	0	0	0	19

5.3 Reconhecimento de locutor

Na tarefa de reconhecimento de locutor, existem dois campos bem definidos, a identificação de locutor e a verificação de locutor. Na identificação, quer-se saber quem é a pessoa que está falando; enquanto que, na verificação, quer-se conhecer se a pessoa que está falando é quem diz ser.

Nesta seção, vamos tratar a identificação de locutor. Para isto, utilizaremos a base das frases. Esta base está projetada para o reconhecimento de locutor dependente do texto; isto significa que teremos um HMM por frase representando cada locutor. Depois usamos tanto as técnicas MFCC como ZCPA e faremos várias comparações entre elas, variando a relação sinal ruído (SNR) das locuções.

A maior parte do processo anterior (reconhecimento de voz) é reproduzida aqui (reconhecimento de locutor); isto é, os conceitos básicos para o treinamento e para o teste, salvo algumas exceções, que serão detalhadas a seguir.

5.3.1 Preparação dos arquivos para o treinamento

O treinamento para identificação de locutor estruturalmente é bem parecido com o treinamento de dígitos isolados. As principais diferenças estão na preparação da base de voz para o treinamento e nos parâmetros envolvidos tanto na extração de características como no protótipo usado.

Preparação da base

A base de frases é dividida em duas partes; uma com somente a frase "E1"e a outra com "E2". Ambas partes são preparadas da seguinte maneira:

- Da repetição 1 até a 30 de todos os locutores serviram para treinamento dos

HMMs.

- Da repetição 31 até a 60 para teste.
- Foram levados em conta os locutores femininos e masculinos conjuntamente.
- Para o treinamento (ou teste), por frase, tem-se: 30(rep.)x25(loc.)x1(frase)=750(arquivos).

Criação de listas e arquivos

Como já vimos no reconhecimento de dígitos, o treinamento inicia-se com a extração das características dos arquivos de áudio. No caso de usar a técnica MFCC, é necessário criar um arquivo de configuração chamado "config01". Este arquivo vai conter os parâmetros necessários para que o HTK, mediante a função HCopy, possa realizar a extração. Esses parâmetros vão ser mudados conforme cada experimento é realizado; tais mudanças vão estar descritas na seção de resultados.

Logo, para o treinamento é necessária a criação das listas de treinamento, neste caso para cada locutor. Estas listas vão depender da frase que se quer analisar como já vimos na preparação da base; elas foram criadas usando o MATLAB. Também é necessário o protótipo de HMM que vai servir como base para criar os HMMs de cada locutor; com este protótipo e com as listas pode-se dar início ao treinamento.

Configurações para a extração de características

As configurações usadas para a extração de características dos coeficientes MFCC são detalhadas a seguir:

- Coeficientes estáticos e estáticos mais dinâmicos, dependendo do experimento.
- Janelas de Hamming de 25ms com uma superposição de 10ms.

- Coeficiente de pré-ênfase de 0.97.
- Usando 22 filtros para o banco de filtros.
- Usando entre 12 e 25 coeficientes.

As configurações usadas para a extração de caracteristicas dos coeficientes ZCPA são detalhadas a seguir:

- O N_p é de 30ms com passos de 10ms.
- Com 17 filtros no banco de filtros do tipo FIR de Hamming e 2 Barks de largura de banda cada.
- O eixo "x"no histograma está na escala Bark.
- O número de bins é 100 no histograma.

Treinamento dos HMMs

O treinamento começa com a extração das características dos arquivos de treinamento. Os parâmetros para este fim, são escolhidos segundo o experimento que vai ser feito; porém, alguns desses valores são utilizados de forma geral, como 25ms de janelamento (Hamming), a superposição entre janelas de 10ms e o coeficiente de pré-ênfases de 0.97; todos esses valores são mantidos para todos os experimentos.

Outros tipos de parâmetros que devem ser fixados, pertencem ao protótipo de HMM que vai ser usado para o treinamento. Tais parâmetros, depois de várias tentativas, foram fixados da seguinte maneira:

- O protótipo de HMM é da forma esquerda-direita.
- Utilizaram-se 10 estados para o protótipo.

- Cada estado tem só uma gaussiana.

- São permitidos até 3 saltos entre estados.

Com esta informação definida, seguem os treinamentos dos HMMs, de acordo com o caso do experimento; como veremos a continuação.

5.3.2 Reconhecimento da base de teste

A base de teste vai sofrer várias modificações com respeito a sua relação com o ruído (SNR) começando desde os 5dB até chegar à base limpa (sem ruído). Por outro lado, o número de coeficientes MFCC ou ZCPA vai variar desde os 12 até os 25 coeficientes, sem usar o c_0 em todos os experimentos; assim como também o número de quadros para a obtenção da primeira e segunda derivada do sinal.

Na obtenção da base com ruído, utilizou-se a base de ruídos NOISEX [32], a qual, tem vários tipos de ruídos criados especialmente para pesquisa. Utilizou-se o ruído branco gaussiano, ruído babble ou balbuceo e o ruído de fábrica.

A seguir, os experimentos serão detalhados.

Experimento I

Este experimento foi realizado com os coeficientes MFCC e ZCPA sem usar o c_0 para a frase E1, utilizando a base de teste e o ruído branco; ele consta de três casos. O primeiro utiliza para a extração só os coeficientes estáticos para ambas técnicas. No segundo, são usados os coeficientes dinâmicos (Δ e $\Delta\Delta$), usando-se um tamanho de quadro de 2 amostras para seu cálculo e para o último caso, teremos os resultados do reconhecimento usando mais três diferentes tamanhos de quadros: 5, 8 e 11 quadros (ver Capítulo 3). Os resultados obtidos nos três casos estão agrupados nas Tabelas 5.4, 5.5 e 5.6, respectivamente. Como podemos observar na Tab. 5.4, utilizando somente os coeficientes estáticos, as taxas de reconhecimento para a técnica MFCC com sinal limpo, estão bem perto de 100% de acerto; já para 20dB de SNR tem uma queda brusca e para 5dB a taxa de acerto é muito pobre. A técnica ZCPA, com sinal limpo não supera a técnica MFCC, mas as taxas de acerto mantêm-se acima de 90% até 15dB de SNR. Porém, com um SNR de 5dB, a técnica ZCPA sofre uma queda significante. Já para a Tab. 5.5, os valores das taxas de acertos aumentam proporcionalmente para todos os SNRs de ambas as técnicas.

Tabela 5.4: Taxa de reconhecimento de locutor em % usando a frase E1 e os coeficientes estáticos.

	MFCC					**ZCPA**				
SNR (dB)	Limpo	20dB	15dB	10dB	5dB	Claro	20dB	15dB	10dB	5dB
12 coef.	**100**	38.4	20	7.87	0.93	97.87	96.13	91.6	72	15.87
15 coef.	99.87	53.07	33.6	13.87	4.27	**98**	**97.33**	**94.27**	79.07	20.53
18 coef.	99.47	56.9	39.33	22.4	6.8	97.07	96.13	93.2	81.07	26.4
20 coef.	99.47	**67.2**	**40.4**	**17.07**	**11.07**	96.27	95.87	93.2	**82.27**	**29.6**
25 coef.	99.33	44.4	30.67	14	5.85	96.8	96	93.33	78.67	27.47

Tabela 5.5: Taxa de reconhecimento de locutor em % usando a frase E1 e os coeficientes estáticos e dinâmicos.

	MFCC $+\Delta+\Delta\Delta$					**ZCPA** $+\Delta+\Delta\Delta$				
SNR (dB)	Claro	20dB	15dB	10dB	5dB	Claro	20dB	15dB	10dB	5dB
12 coef.	**100**	53.6	31.87	15.73	6.93	**98.53**	97.6	92.67	80.13	26.4
15 coef.	**100**	68.27	38.67	16.4	11.2	97.73	97.47	93.87	83.2	30.93
18 coef.	99.73	73.47	43.73	29.47	18.27	98	**97.73**	**94.8**	**87.2**	**35.33**
20 coef.	99.47	**77.73**	**49.33**	**33.87**	**18.93**	97.47	97.33	94.4	86.13	31.73
25 coef.	99.47	63.6 .	38.53	24.8	10.27	96.67	96.27	93.73	80.8	30.13

Tabela 5.6: Taxa de reconhecimento de locutor em % com a frase E1 usando 15 coeficientes com $\Delta+\Delta\Delta$ para diferentes quadros.

	MFCC $+\Delta+\Delta\Delta$					**ZCPA** $+\Delta+\Delta\Delta$				
SNR (dB) / Quadros	Claro	20dB	15dB	10dB	5dB	Claro	20dB	15dB	10dB	5dB
5	**100**	76.8	49.2	25.87	8.67	98.13	**98.13**	97.07	90.93	52.53
8	99.87	**79.07**	**51.33**	**26.93**	11.73	**98.27**	97.73	**97.47**	92.8	62.27
11	99.6	76.93	45.73	25.73	**18.4**	97.33	97.07	96.4	**92.93**	**67.87**

O fato que as taxas de acerto da técnica ZCPA não superam a técnica MFCC com o sinal limpo é devido à informação do sinal que não é considerado ou é suavizado no momento de criar os histogramas tendo em conta o proncipio da freqüência dominante. Este mesmo principio faz com que seja robusto ao ruído aditivo. A queda no passo de 10dB para 5dB é devido a que a energia do ruído no sinal seja maior que as amplitudes com informação no histograma. Como vemos na Fig. 5.13, no histograma do sinal para 10dB ainda temos informação importante do sinal, porém, quando passamos para 5dB, essa informação se perde nas diferentes energias

que aparecem por causa do ruído.

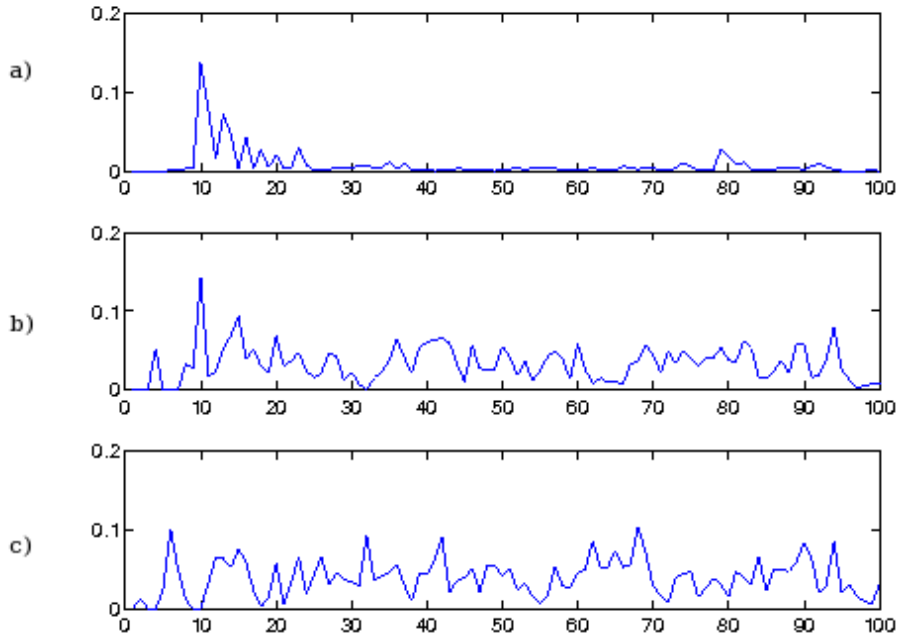

Figura 5.13: Histogramas da técnica ZCPA de um trecho da palavra *fifth* para: a)sinal limpo, b) sinal com 10dB de SNR ruído branco, c) sinal com 5dB de SNR ruído branco.

Na Tab. 5.6, existe uma melhora nas taxas de acerto para ambas técnicas, sobretudo para a técnica ZCPA com 5dB de SNR, que tem uma melhora de 35.33% (da Tab. 5.5) para 67.87%, no caso de usar quadros de 11 amostras.

Experimento II

Este experimento foi realizado também com os coeficientes MFCC e ZCPA e a mesma configuração que o experimento anterior, mudando da frase E1 para a frase E2.

Tabela 5.7: Taxa de reconhecimento de locutor em % usando a frase E2 e os coeficientes estáticos.

	MFCC					**ZCPA**				
SNR (dB)	Claro	20dB	15dB	10dB	5dB	Claro	20dB	15dB	10dB	5dB
12 coef	**100**	68.4	26.27	12.8	4.4	**99.07**	**98.53**	92.8	77.07	21.2
15 coef	99.87	73.47	41.2	16.93	**12.53**	98.93	98.13	94.93	82.8	30
18 coef	99.73	77.87	48.27	**22.8**	9.73	98.13	98.27	**95.87**	83.47	36.13
20 coef	99.47	**80**	47.6	21.73	9.73	98.27	97.73	95.07	83.6	38.53
25 coef	99.07	76.27	**52.13**	15.73	5.73	97.6	97.33	94.8	**84.53**	**43.07**

Tabela 5.8: Taxa de reconhecimento de locutor em % usando a frase E2 e os coeficientes estáticos e dinâmicos.

	MFCC $+\Delta+\Delta\Delta$					**ZCPA** $+\Delta+\Delta\Delta$				
SNR (dB)	Claro	20dB	15dB	10dB	5dB	Claro	20dB	15dB	10dB	5dB
12 coef	**100**	74	40.27	13.87	6.67	**99.47**	**98.93**	96.8	88.27	46.4
15 coef	99.87	78.13	45.73	14.67	**8.4**	99.07	98.4	**96.93**	**90.67**	55.73
18 coef	99.87	84.53	52.53	20.13	7.73	97.73	97.6	95.33	90	58.53
20 coef	99.73	**87.2**	**62.7**	**24.27**	7.07	98	97.6	96	89.87	58.4
25 coef	99.6	80.27	56	21.6	4.13	96.4	96.4	95.33	88.67	**60.13**

Tabela 5.9: Taxa de reconhecimento de locutor em % usando a frase E2 usando 15 coeficientes com $\Delta + \Delta\Delta$ para diferentes quadros.

	MFCC $+\Delta + \Delta\Delta$					**ZCPA** $+\Delta + \Delta\Delta$				
SNR (dB) / Quadros	Claro	20dB	15dB	10dB	5dB	Claro	20dB	15dB	10dB	5dB
5	**100**	84.13	55.2	14.93	7.2	**99.2**	**98.93**	97.87	92.4	67.6
8	**100**	82.93	54.93	21.33	**7.47**	98.93	98.8	**98**	**92.8**	**72.27**
11	**100**	**86.8**	**58.93**	**24.67**	7.33	98.27	98.53	97.87	91.6	71.73

Neste experimento vemos na Tab. 5.7 que as taxas de acerto aumentam comparando com a Tab. 5.4; este aumento deve-se, logicamente, ao fato de que a frase E2 tem predominantemente fonemas nasais que possuem uma melhor definição dos picos e dos vales no espectro gerado no trato vocal e nasal devido a anti-ressonâncias confirmando os resultados relatados em [12]. O mesmo acontece na Tab. 5.8 apresentando uma boa melhora para a técnica ZCPA onde, para 10dB de SNR, a taxa de acerto esta bem perto de 90%. A técnica MFCC há uma pequena melhora com relação à Tab. 5.7 mas, em alguns casos, a taxa de acerto está menor se comparada com a Tab. 5.4.

Para a Tab. 5.9, os resultados para ambas técnicas estão similares com relação à Tab. 5.6; contudo, a técnica ZCPA, para 5dB de SNR, apresenta uma melhora de 67.87% para 72.27% repectivmente.

Experimento III

Para este experimento, utilizou-se outro tipo de ruído da base NOISEX; o ruído de "Babble"ou "Balbuceio"[32]. Similarmente aos outros, este experimento consta dos mesmos casos descritos anteriormente.

Tabela 5.10: Taxa de reconhecimento de locutor em % usando a frase E1 e os coeficientes estáticos.

	MFCC					**ZCPA**				
SNR (dB)	Claro	20dB	15dB	10dB	5dB	Claro	20dB	15dB	10dB	5dB
12 coef	**100**	96	68	28.13	12.27	97.87	96.13	83.73	45.47	**16.4**
15 coef	99.87	96.53	68.53	27.33	**12.4**	**98**	**97.6**	86.4	46.13	15.47
18 coef	99.47	**97.2**	74	**29.47**	10.53	97.07	95.2	86.13	46.27	14.53
20 coef	99.47	**97.2**	**74.53**	26.67	9.2	96.27	94.93	**87.33**	46.4	14.27
25 coef	99.07	94.27	66.27	22.4	8.8	96.8	95.47	86.53	**46.53**	15.47

Tabela 5.11: Taxa de reconhecimento de locutor em % usando a frase E1 e os coeficientes estáticos e dinâmicos.

	MFCC $+\Delta+\Delta\Delta$					**ZCPA** $+\Delta+\Delta\Delta$				
SNR (dB)	Claro	20dB	15dB	10dB	5dB	Claro	20dB	15dB	10dB	5dB
12 coef	**100**	**99.47**	86.93	42	**15.87**	**98.53**	**96.27**	85.33	48.27	**16.4**
15 coef	**100**	98.27	86	40.4	12.67	97.73	96	88.13	50.53	15.33
18 coef	99.73	99.2	86.67	37.87	10.13	98	**96.27**	**89.6**	**55.87**	16.27
20 coef	99.47	99.07	**87.73**	37.6	9.07	97.47	96.13	88.53	51.47	14.8
25 coef	99.47	97.47	86.4	**44.4**	14.8	96.67	94.4	85.33	48	14.67

Tabela 5.12: Taxa de reconhecimento de locutor em % usando a frase E1 usando 15 coeficientes com $\Delta + \Delta\Delta$ para diferentes quadros.

	MFCC $+\Delta + \Delta\Delta$					**ZCPA** $+\Delta + \Delta\Delta$				
SNR (dB) / Quadros	Claro	20dB	15dB	10dB	5dB	Claro	20dB	15dB	10dB	5dB
5	**100**	**98.8**	**89.6**	43.47	**14.13**	98.13	**98**	94.53	65.73	26.4
8	99.87	98.27	87.6	**44.67**	12.53	**98.27**	**98**	**96**	72.4	29.73
11	99.6	97.47	83.87	42.93	10.53	97.33	96.4	95.07	**77.07**	**35.87**

Com este tipo de ruído, na Tab. 5.10, a técnica ZCPA supera à técnica MFCC consideravelmente até 10dB de SNR; a técnica MFCC apresenta uma melhoria se comparamos com a tabela similar do Experimento I; para 5dB de SNR, ambas técnicas da Tab. 5.10 apresentam similares taxas de acerto, ganhando por pequena margem os ZCPA. Já para a Tab. 5.11, as duas técnicas tem uma semelhança em relação a suas taxas de reconhecimento; em alguns casos, a técnica MFCC apresenta melhores taxas de acerto, em um caso para 15dB de SNR (12 coeficientes) e em todos os casos a partir de 20dB de SNR.

Este tipo de ruído apresenta diversas freqüências que interferem na faixa da voz, facendo com que a técnica ZCPA seja mais vulnerável e tenha taxas de acerto baixas se comparamos com tabelas similares do Experimento I. Isto é devido a que o tipo de ruído consta de muitas pessoas falando que interferem na freqüência dominante de interese para o reconhecimento usando a técnica ZCPA.

Utilizando os mesmos motivos, para a Tab. 5.12, os ZCPA tem uma melhoria significativa para 15, 10 e 5dB de SNR; porém, não tem uma vantagem ampla como no mesmo caso do Experimento I.

Experimento IV

Utilizou-se o ruído de "Babble"ou "Balbuceio"com a mesma estructura do Experimento III mas com a frase E2.

Tabela 5.13: Taxa de reconhecimento de locutor em % usando a frase E2 e os coeficientes estáticos.

	MFCC					**ZCPA**				
SNR (dB)	Claro	20dB	15dB	10dB	5dB	Claro	20dB	15dB	10dB	5dB
12 coef	**100**	98.4	85.47	38.27	11.47	**99.07**	**97.47**	90.67	61.2	**25.47**
15 coef	99.87	**99.2**	**89.87**	52.4	17.47	98.93	**97.47**	90.53	63.87	22.8
18 coef	99.73	97.87	85.87	48	15.2	98.13	**97.47**	**91.07**	64	22.93
20 coef	99.47	97.07	84.13	50.53	17.07	98.27	96.13	89.2	**65.6**	22.8
25 coef	99.07	97.07	85.33	**55.07**	**18.93**	97.6	95.07	88.4	64.67	21.2

Tabela 5.14: Taxa de reconhecimento de locutor em % usando a frase E2 e os coeficientes estáticos e dinâmicos.

	MFCC $+\Delta+\Delta\Delta$					**ZCPA** $+\Delta+\Delta\Delta$				
SNR (dB)	Claro	20dB	15dB	10dB	5dB	Claro	20dB	15dB	10dB	5dB
12 coef	**100**	**99.6**	93.87	60.8	21.6	**99.47**	**98.8**	**96.13**	78	37.87
15 coef	99.87	99.47	**94.4**	**70.13**	**26.53**	99.07	98.13	95.47	**80.8**	**39.07**
18 coef	99.87	98.27	90.93	66.67	23.33	97.73	96.93	94.53	80	35.33
20 coef	99.73	98.27	89.07	66.13	21.33	98	96.93	93.73	78.13	34.27
25 coef	99.6	98	87.87	61.73	19.2	96.4	96	91.6	76.4	27.6

Tabela 5.15: Taxa de reconhecimento de locutor em % usando a frase E2 usando 15 coeficientes com $\Delta + \Delta\Delta$ para diferentes quadros.

	MFCC $+\Delta + \Delta\Delta$					**ZCPA** $+\Delta + \Delta\Delta$				
SNR (dB) / Quadros	Claro	20dB	15dB	10dB	5dB	Claro	20dB	15dB	10dB	5dB
5	**100**	99.2	93.33	63.73	22.67	**99.2**	**98.8**	95.2	78.53	30.8
8	**100**	99.33	93.33	64.67	**25.73**	98.93	98.67	96.93	**85.47**	38
11	**100**	**99.47**	**95.6**	**68**	24.8	98.27	98	**97.2**	84.67	**41.2**

Agora com a frase E2, a Tab. 5.13 tem uma melhora significativa para 10dB de SNR comparando com a Tab. 5.10; porém, para 5dB, a melhora é pouca. Com os resultados dos coeficientes dinâmicos da Tab. 5.14, existe uma boa melhora para os casos de 10dB e 5dB de SNR para ambas técnicas e o mesmo acontece com a Tab. 5.15.

Os resultados deste experimento apresentam taxas de acerto superiores ao experimento anterior, devido que a frase E2 contém maior informação do locutor devido a que predominam fonemas nasais que ajudam no reconhecimento de locutor, já que o trato nasal não pode ser articulado.

Experimento V

Desta vez vamos analisar outro tipo de ruído, chamado "Factory"ou ruído de fábrica [32]. Começaremos com a frase E1.

Tabela 5.16: Taxa de reconhecimento de locutor em % usando a frase E1 e os coeficientes estáticos.

	MFCC					**ZCPA**				
SNR (dB)	Claro	20dB	15dB	10dB	5dB	Claro	20dB	15dB	10dB	5dB
12 coef	**100**	94.67	76.53	44.4	13.2	97.87	97.2	95.33	**85.87**	**48.4**
15 coef	99.87	94.8	80.4	**52.93**	**26.8**	**98**	**97.73**	**96.8**	84.4	44
18 coef	99.47	95.33	81.2	51.47	21.2	97.07	96.53	94.67	82	40.4
20 coef	99.47	95.73	78.8	47.07	15.47	96.27	96.4	94.27	81.33	38.67
25 coef	99.07	**96.67**	**82.93**	50.13	25.47	96.8	96.27	93.07	76.93	32

Tabela 5.17: Taxa de reconhecimento de locutor em % usando a frase E1 e os coeficientes estáticos e dinâmicos.

	MFCC $+\Delta+\Delta\Delta$					**ZCPA** $+\Delta+\Delta\Delta$				
SNR (dB)	Claro	20dB	15dB	10dB	5dB	Claro	20dB	15dB	10dB	5dB
12 coef	**100**	98.13	83.87	49.33	14.67	**98.53**	**97.87**	**96.13**	86.13	**53.07**
15 coef	**100**	98.4	85.47	57.6	21.07	97.73	97.33	**96.13**	86	47.6
18 coef	99.73	98.8	86.27	56.53	21.87	98	97.6	96	85.87	45.87
20 coef	99.47	**98.67**	87.6	57.2	19.2	97.47	97.07	95.6	**86.53**	43.47
25 coef	99.47	**98.67**	**88**	**60.27**	**27.33**	96.67	96.13	94.13	81.87	30.53

Tabela 5.18: Taxa de reconhecimento de locutor em % usando a frase E1 usando 15 coeficientes com $\Delta + \Delta\Delta$ para diferentes quadros.

	MFCC $+\Delta + \Delta\Delta$					**ZCPA** $+\Delta + \Delta\Delta$				
SNR (dB) / Quadros	Claro	20dB	15dB	10dB	5dB	Claro	20dB	15dB	10dB	5dB
5	**100**	**98.8**	88.4	65.73	23.47	98.13	**98.27**	97.6	93.2	62.27
8	99.87	98.67	**89.07**	**70.67**	28.53	**98.27**	97.6	**98**	**94.4**	65.73
11	99.6	98.67	88.13	69.07	**35.33**	97.33	96.67	97.07	93.47	**68.93**

Na Tab. 5.16, vemos que a técnica ZCPA apresenta melhores resultados que com o ruído babble e valores parecidos ao experimento com o ruído branco para os mesmos casos. Estes resultados melhoram ainda mais com a adição dos coeficientes dinâmicos como visto na Tab. 5.17 comparando com a tabela anterior. No caso da Tab. 5.18, os resultados para ambas técnicas melhoram consideravelmente.

Todo isto é devido a que no espectro do ruído de fábrica, aparecem energias nas freqüências altas do espectro; isto se deve ao fato de que o ruído contém golpes de martelos e, conhecendo que um golpe pode ser considerado como um impulso sua FFT apresenta componentes de energia em todas as freqüências. È por isso que este tipo de ruído têm menor interferencia na faixa da voz; portanto, a técnica ZCPA apresenta taxas de acerto iguais e até em alguns casos melhores que nos experimentos feitos com o ruído branco.

Outro motivo pela qual os resultados deste experimento são bons se comparamos com outros os outros experimentos é porque o ruído de fábrica tem a propiedade de ser não estacionário; ouseja, não se distribui uniformemente em todo o intervalo de tempo.

Experimento VI

Utilizando a mesma estructura e o tipo de ruído que o experimento anterior mas usando a frase E2.

Tabela 5.19: Taxa de reconhecimento de locutor em % usando a frase E2 e os coeficientes estáticos.

	MFCC					**ZCPA**				
SNR (dB)	Claro	20dB	15dB	10dB	5dB	Claro	20dB	15dB	10dB	5dB
12 coef	**100**	**98.4**	85.73	49.87	11.87	**99.07**	**98.53**	96.67	86.67	46.27
15 coef	99.87	**98.4**	86.93	51.33	11.73	98.93	98.27	**97.07**	86.4	46.93
18 coef	99.73	97.87	88.67	49.6	14.53	98.13	98.13	96.93	**86.93**	**47.6**
20 coef	99.47	97.47	89.33	49.2	12.8	98.27	97.07	94.8	83.73	44
25 coef	99.07	97.47	**91.07**	**63.47**	**20.67**	97.6	96.8	94.67	84	44.93

Tabela 5.20: Taxa de reconhecimento de locutor em % usando a frase E2 e os coeficientes estáticos e dinâmicos.

	MFCC $+\Delta+\Delta\Delta$					**ZCPA** $+\Delta+\Delta\Delta$				
SNR (dB)	Claro	20dB	15dB	10dB	5dB	Claro	20dB	15dB	10dB	5dB
12 coef	**100**	**99.2**	**93.73**	68.67	23.73	**99.47**	**99.07**	**97.6**	**92.8**	65.07
15 coef	99.87	98.53	92.53	68.8	24.53	99.07	98.93	97.07	**92.8**	**68**
18 coef	99.87	97.73	93.33	66.93	23.33	97.73	97.87	96.67	92.27	65.07
20 coef	99.73	98.13	93.47	67.73	21.73	98	97.33	95.87	91.47	62.93
25 coef	99.6	98	93.47	**72.13**	**24.67**	96.4	96.27	93.87	89.73	60.8

Tabela 5.21: Taxa de reconhecimento de locutor em % usando a frase E2 usando 15 coeficientes com $\Delta + \Delta\Delta$ para diferentes quadros.

	MFCC $+\Delta + \Delta\Delta$					**ZCPA** $+\Delta + \Delta\Delta$				
SNR (dB) / Quadros	Claro	20dB	15dB	10dB	5dB	Claro	20dB	15dB	10dB	5dB
5	**100**	99.07	94.67	70.13	29.87	**99.2**	**99.07**	97.87	93.07	72.4
8	**100**	99.47	94.67	71.07	**30.8**	98.93	98.93	**98.53**	**93.33**	**74.13**
11	**100**	**99.87**	**96.53**	**72.8**	30.13	98.27	98.53	97.87	92.8	72

Como já vimos, os resultados das taxas de reconhecimento para a frase E2, sempre tem uma melhora frente à frase E1. Na Tab. 5.19, os valores das taxas de acerto, não tem uma melhora muito significativa comparando com a Tab. 5.16; mas, quando são adicionados os coeficientes Δ e $\Delta\Delta$, as taxas de acerto melhoram de maneira geral para ambas as técnicas. Para a Tab. 5.21, as taxas são bem altas, chegando a passar os 30% para 5dB com os coeficientes MFCC e os 70% para os coeficientes ZCPA.

Isto é devido a que como já vimos em experimentos anteriores, os fonemas nasais presentes na frase E2, ajudam muito no reconhecimento de locutor, conseguindo assim, taxas de acerto muito boas.

5.4 Reconhecimento usando a base YOHO

Nesta seção, utilizaremos a base YOHO por ser uma base conhecida internacionalmente e ter um maior número de locutores. Ela foi projetada para ser utilizada em experimentos de reconhecimento de locutor dependente do texto.

O trabalho feito nesta seção consta de vários pontos. Primeiro, far-se-á a segmentação em dígitos isolados tanto para a base de treinamento como para a de teste. Segundo, tendo os dígitos já isolados, procede-se ao treinamento dos HMMs, um por dígito falado por cada locutor. E por último, com a base de teste, faz-se uma identificação de locutor por dígito isolado, usando os coeficientes MFCC e ZCPA.

5.4.1 Segmentação em dígitos isolados

Primeiramente, escolheram-se 50 locutores (43 homens e 7 mulheres) para o treinamento e teste que formarão a nossa nova base YOHO. Depois, foi feita uma segmentação manual em dígitos, dos mesmos locutores escolhidos antes, e extraídas 2 repetições por dígito de 20 locutores (15 homens e 5 mulheres). Isto foi feito com o fim de ter-se uma base de "treinamento" pequena de cada dígito, para assim poder fazer a segmentação da base inteira (treinamento e teste). A seguir, detalha-se o processo de segmentação.

Treinamento com a base pequena

Devido ao fato da base YOHO ser muito grande e termos a necessidade de fazer um reconhecimento de locutor por dígitos isolados. Precisamos então de um sistema de segmentação automático. Como o HTK, mediante sua ferramenta HVite, fornece como resultado os inícios e os finais de cada palavra que está sendo reconhecida, esta

tarefa de segmentação é traduzida para uma tarefa de reconhecimento para dígitos isolados.

Dado que agora é uma tarefa de reconhecimento, temos a necessidade de ter uma pequena base (derivada da base YOHO) de dígitos isolados; isto é, temos que segmentar manualmente algumas locuções faladas (seqüência de 6 dígitos) de alguns locutores representativos para servir de treinamento para esta tarefa inicial. A segmentação manual é trabalhosa e tem que ser realizada com muito cuidado.

Como já vimos antes, a base pequena está formada por 20 locutores, dos quais 15 são homens e 5 são mulheres, e foram escolhidos 2 repetições por dígito (ver Tab. 5.22 e Tab. 5.23) para cada locutor. Desta maneira vamos ter uma pequena base formada por 40 repetições para os 16 dígitos possíveis da base YOHO. Esta

Tabela 5.22: Nomes dos possíveis dígitos isolados da base YOHO e suas etiquetas para as unidades.

	Unidades							
Dígito	one	two	three	four	five	six	seven	nine
Etiqueta	one	two	thr	fou	fiv	six	sev	nin

Tabela 5.23: Nomes dos possíveis dígitos isolados da base YOHO e suas etiquetas para as dezenas.

	Dezenas							
Dígito	twenty	thirty	forty	fifty	sixty	seventy	eighty	ninety
Etiqueta	twt	tht	fot	fit	sit	set	eit	nit

base, vai nos servir para treinar HMMs para cada dígito e desta forma segmentar a base completa (treinamento e teste da tarefa principal). Vários experimentos

foram feitos com diferentes configurações tanto dos parâmetros para a extração das características como daqueles pertencentes aos protótipos de HMM; chegou-se assim à configuração que deu a maior taxa de reconhecimento para a base de treinamento (base grande). Tais configurações são as seguintes:

- 15 coeficientes MFCC incluindo c_0, Δ e $\Delta\Delta$.
- Janelas de Hamming com 10ms de largura e 5ms de superposição.
- Quadros de 4 amostras para o cálculo dos coeficientes dinâmicos.
- Os HMMs são do tipo esquerda-direita com até 3 saltos entre estados.
- Foram feitas variações do número de estados e das gaussianas para os dígitos como mostrados na Tab. 5.24. onde por exemplo 9e - 5g representa a um protótipo de 9 estados e 5 gaussianas.

Tabela 5.24: Protótipos dos diferentes dígitos para o treinamento.

Unidades	one	two	thr	fou	fiv	six	sev	nin
Protótipo	9e- 5g	9e- 8g	10e- 3g	10e- 3g	10e- 3g	10e- 3g	7e- 3g	10e- 3g
Dezenas	twt	tht	fot	fit	sit	set	eit	nit
Protótipo	10e- 3g	9e- 3g	10e- 3g	8e- 4g	9e- 5g	10e- 3g	6e- 3g	10e- 3g

- Para o modelo do silêncio "sil" usou-se um protótipo de 1 estado com 1 gaussiana.

Depois que os parâmetros foram definidos, procede-se ao treinamento dos HMMs e posteriormente ao reconhecimento da base grande de treinamento e teste da base YOHO. A parte da extração e treinamento é feito usando o programa HTK e suas diversas funções que já foram discutidas antes. Na parte de reconhecimento,

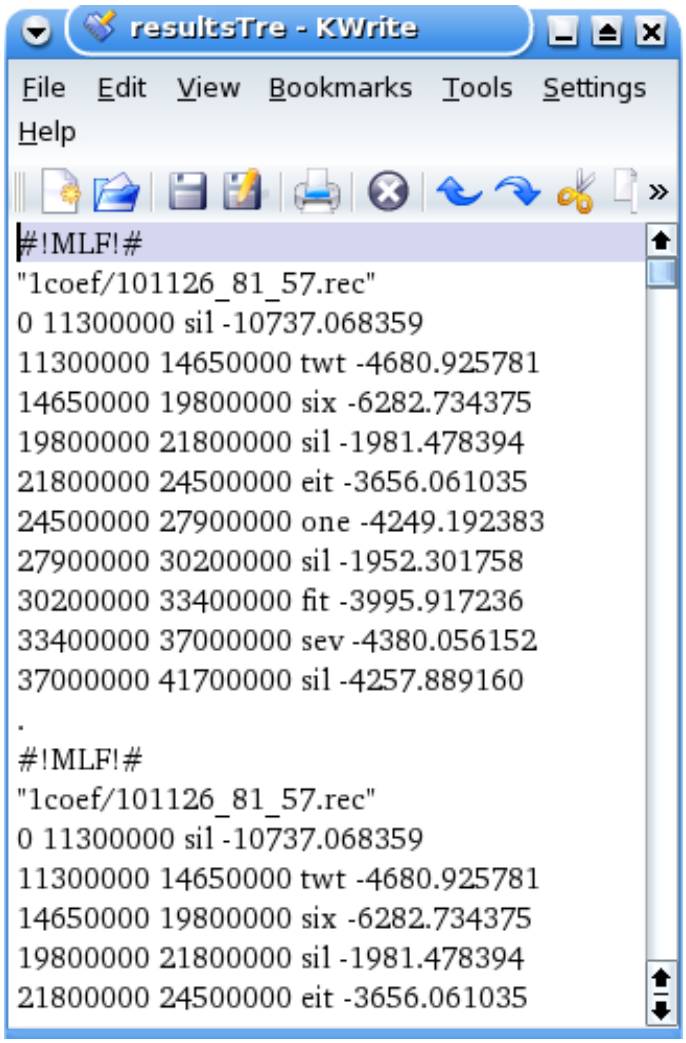

Figura 5.14: Fragmento do arquivo "resultsTre"produto do reconhecimento da base de teinamento YOHO

utilizou-se também a função HVite do HTK; ela dá como resultado um arquivo, como na Fig. 5.14, contendo todos os inícios e finais de cada frase reconhecida, assim como também, a verossimilhança e o modelo ganhador.

Processando os resultados

Nesta parte, depois de obter o arquivo de resultados tanto da base de treinamento como da de teste mediante o uso do programa MATLAB, foi feito um processamento dos arquivos de resultados do reconhecimento anterior, "resultsTre"e "re-

sultsTe"(resultados da base de treinamento e teste respectivamente), com o fim de extrair os inícios e finais de cada dígito reconhecido.

Logo, com o auxílio do MATLAB, todas as informações desses arquivos são extraídas e processadas com a finalidade de segmentar os arquivos de áudio correspondentes a cada locução. Tendo as bases de treinamento e de teste já segmentadas em dígitos isolados, temos que comprovar se a segmentação foi realizada corretamente. Para isto, vamos reconhecer os dígitos segmentados isoladamente com os mesmos HMMs treinados para a segmentação.

Este tipo de verificação deu como resposta 97.9286% de taxa de reconhecimento para a base de treinamento e 98.9813% para a base de teste. Este resultado, próximo mas não igual a 100%, é devido a alguns dígitos mal segmentados por causa, provavelmente, de imperfeições da própria base, tais como ruídos estranhos, reverberação, etc. Portanto, decidiu-se eliminar todos aqueles dígitos que estavam apresentando problemas no reconhecimento, ficando com a base limpa. Depois de tirar os "maus" elementos, a taxa de acerto obtida, produto do reconhecimento, foi de 100%.

5.4.2 Reconhecimento de locutor por dígitos isolados

Agora, já com a base de treinamento e de teste segmentados em dígitos isolados a partir da base YOHO, procede-se ao reconhecimento robusto de locutor por dígitos isolados. Nesta subseção, apresentaremos como é feito o treinamento e também os diversos testes com as técnicas MFCC e ZCPA usando a base YOHO segmentada e processada como descrita.

Treinamento com a base YOHO

Para o treinamento, várias pastas (uma por cada locutor) foram criadas, cada uma delas contendo as listas de treinamento (uma por dígito) de todos os possíveis dígitos (ver Tabs. 5.22 e 5.23) desse mesmo locutor. Logo, criou-se o protótipo que vai ser usado para todos os dígitos assim como também fixaram-se os parâmetros para a extração das características para cada ténica. O protótipo e esses parâmetros são:

- HMM protótipo com 5 estados e 2 gaussianas por estado.
- Permitiram-se 3 saltos entre estados.
- Para todos os testes, usaram-se 15 coeficientes estáticos mas desta vez sem c_0 e com seus Δ e $\Delta\Delta$.
- Para os MFCCs, foram usadas janelas de Hamming de 10ms com 5ms de superposição.
- Para os ZCPAs, foram usados bancos de 17 filtros, 100 bins de freqüência e 30ms de N_p.

Com todos esses dados, procede-se ao treinamento como já foi visto anteriormente. Com ajuda do MATLAB, todos os modelos para cada dígito de cada locutor foram treinados. Logo, esses modelos são guardados em pastas específicos com a finalidade de serem utilizados nos testes.

Testes com a base YOHO

A seguir, experimentos usando a base de teste da base YOHO foram conduzidos. Para realizar os experimentos só foi considerado o ruído branco, por ser este o mais

difícil de trabalhar. Os experimentos consistem em reconhecer o locutor usando dígitos isolados para 5dB, 10dB de SNR mais a base limpa. A seguir, são apresentados os resultados em % da taxa de reconhecimento para cada dígito:

Tabela 5.25: Taxa de reconhecimento de locutor em % usando os coeficientes MFCC na base YOHO (unidades).

	one	two	thr	fou	fiv	six	sev	nin
Limpa	95.83	85.83	96.34	90.08	93.69	92.59	99.38	97.28
10dB	2.78	8.31	4.13	3.41	17.87	11.27	4.38	30.9
5dB	2.23	1.95	2.07	4.34	4.2	3.38	2.34	7.64

Tabela 5.26: Taxa de reconhecimento de locutor em % usando os coeficientes MFCC na base YOHO (dezenas).

	twt	tht	fot	fit	sit	set	eit	nit
Limpa	95.39	97.85	96.19	92.15	95.27	98.7	81.16	97.22
10dB	9.82	5.52	4.27	20.26	7.99	6.23	15.94	46.41
5dB	2.98	1.53	2.74	1.76	2.66	3.04	4.83	12.15

Tabela 5.27: Taxa de reconhecimento de locutor em % usando os coeficientes ZCPA na base YOHO (unidades).

	one	two	thr	fou	fiv	six	sev	nin
Limpa	70.38	49.92	63.75	62.33	71.92	59.19	79.19	77.76
10dB	56.75	26.71	37.36	32.71	37.39	9.66	37.81	71.48
5dB	33.38	14.28	20.99	16.74	20.42	3.06	17.5	48.05

Tabela 5.28: Taxa de reconhecimento de locutor em % usando os coeficientes ZCPA na base YOHO (dezenas).

	twt	tht	fot	fit	sit	set	eit	nit
Limpa	73.21	65.34	64.94	52.4	59.62	73.77	51.85	77.6
10dB	54.76	48.16	39.94	16.83	16.72	32.32	45.25	67.2
5dB	19.05	30.67	28.2	9.46	7.25	18.12	26.89	42.75

Nas tabelas anteriores, podemos ver que as taxas de acerto para a base limpa utilizando a técnica MFCC (Tabs. 5.25 e 5.26) estão acima de 90% e com alguns dígitos passam de 95%. Isto indica que a técnica MFCC apresenta uma boa solução em reconhecimento de locutor usando dígitos isolados quando a SNR é muito elevada. Porém, se a SNR é muito baixo como 10dB, a técnica MFCC sofre uma queda muito grande, chegando a menos de 5% de taxa de acerto para alguns casos. Isto se deve ao fato que a técnica MFCC não é robusta ao ruído branco aditivo e também que a corta duração da seqüência é um fator importante para o reconhecimento de locutor.

Por outro lado, nas Tabs. 5.27 e 5.28, usando a técnica ZCPA, as taxas de acerto se bem melhoram significativamente para 10dB e 5dB, para a base limpa a técnica não apresenta boas taxas de acerto, provavelmente devido a que se teve uma polarização na segmentação utilizando a técnica MFCC, podemos ver que no máximo podem chegar até 77% para os dígitos "nin"e "nit"e 79% para o dígito "sev". Para estos casos, as taxas de acerto são as mais altas devido a que os dígitos apresentam maiormente fonemas nasais que facilitam o reconhecimento. Para o caso dos dígitos "fit"e "six", temos as piores taxas de acerto; isto é porque os fonemas destes dígitos são fricativos em sua maioria, aumentando o erro para ambas técnicas.

Capítulo 6

Conclusões e trabalhos futuros

6.1 Conclusões

1. Neste trabalho, foi proposta a comparação do desempenho entre as técnicas MFCC e ZCPA para a aplicação "identificação de locutor dependente do texto". A superioridade do desempenho da técnica ZCPA sobre a técnica MFCC, no caso de ambientes ruidosos, foi confirmada, através do uso de sinais de voz corrompidos por ruído branco gaussiano, ruído de fábrica e ruído "babble", para frases em português. Segundo as tabelas do Experimento I até o Experimento VI, a superioridade da técnica ZCPA em situações onde a base de teste tem uma baixa relação sinal-ruído (SNR) frente à técnica MFCC é evidente. Isto se deve principalmente ao fato que a construção dos histogramas do ZCPA consistem em atribuir a cada bin de freqüência, uma estimativa da potência da sub-banda correspondendo à freqüência dominante da sub-banda.

2. Considerando os resultados dos Experimentos I até o VI, pode-se claramente afirmar que as taxas de acerto da frase "E2" são melhores quando comparadas

com as taxas da frase“E1”; isto é devido ao número de fonemas nasais da frase “E2”ser predominante. Essa melhoria é uma vantagem das frases com um maior número de fonemas nasais frente às frases com predominância de fonemas orais. Tendo em vista que o trato nasal não pode ser articulado, os fonemas nasais apresentam uma informação mais precisa sobre a pessoa que está falando e com isto são úteis ao reconhecimento automático de locutor.

3. Podemos observar nos Experimentos I e II que as taxas de reconhecimento da técnica ZCPA são altas em comparação com a técnica MFCC; isto está associado ao fato do ruído branco ter uma distribuição da energia em todas as freqüências do espectro e a técnica ZCPA, graças à sua estimativa de potência em torno da freqüência dominante, fazendo com que esta técnica seja mais robusta que a técnica MFCC.

4. Nos resultados do Experimento III e IV, com o ruído “babble”, podemos ver que a técnica ZCPA tem uma queda nas taxas de acerto comparada com os casos similares com o ruído branco. Como o ruído “babble”está composto por várias pessoas falando; no seu espectro da Fig. 5.1c, encontramos diferentes freqüências, originadas pelos formantes, produzidas pelas pessoas que conformam o ruído. Estas freqüências distorcem a resposta da técnica ZCPA e, para casos mais críticos, como 10dB e 5dB de SNR, a técnica ZCPA apresenta taxas de reconhecimento inferiores à MFCC.

5. Considerando os Experimentos V e VI, com o ruído de fábrica os resultados melhoram se comparados aos dois experimentos anteriores. Como podemos ver na Fig. 5.1b, no espectro do ruído de fábrica, aparecem energias nas

freqüências altas do espectro; isto se deve ao fato de que é um tipo de ruído não estacionário; ouseja, que não esta distribuído uniformemente em todo o intervalo de tempo; além disso, contém golpes de martelos e, conhecendo que um golpe pode ser considerado como um impulso sua FFT apresenta componentes de energia em todas as freqüências tendo menos interferência na faixa de freqüências da voz. É por isso que a técnica ZCPA tem boas taxas de reconhecimento para 10dB e 5dB, neste caso, em comparação ao ruído "babble".

6. As Tabs. 5.25 e 5.26 apresentam taxas de reconhecimento boas para a base clara; para alguns dígitos tais como o "nin"e o "nit", as taxas de acerto são superiores a 30% para 10dB de SNR. Com a técnica ZCPA, como nas Tabs. 5.27 e 5.28, as taxas melhoram bastante para alguns dígitos mas são parecidas aos resultados da técnica MFCC para outros. Isto se deve a alguns dígitos, que apresentam fonemas nasais principalmente, terem uma boa taxa de reconhecimento com respeito aos que apresentam fonemas orais. Para o caso dos dígitos "fit"e "six", temos as piores taxas de acerto; isto é porque os fonemas destes dígitos são fricativos em sua maioria, aumentando o erro para ambas técnicas.

6.2 Trabalhos Futuros

Apresentamos, a seguir, algumas posibilidades de continuação da presente pesquisa:

- Pesquisar a extração de caracteristicas ZCPA em sincronia com o pitch.
- Implementar um sistema de segmentação automática robusta para a base

YOHO.

- Implementação de um sistema de verificação de locutor usando a base YOHO em ambientes com ruído aditivo.

- Implementar um sistema de verificação de locutor independente do texto (usando GMM) em ambientes ruidosos usando ZCPA.

- Pesquisar uma fusão entre as duas técnicas MFCC e ZCPA.

Bibliografia

[1] Ahmed N., Natarajan T., Rao K. R.,"Discrete Cosine Transform", IEEE Transactions on Computers, pp. 90-93, 1974.

[2] Campbell Jr J. P., "Speaker Recognition: A tutorial", Proceedings of the IEEE, vol. 85, no. 9, pp. 1437-1462, 1997.

[3] ChiWei C., Lin Q., Yuk D., "An HMM Approach to Text-Prompted Speaker Verification", CAIP Center, pp. 673-676, 1996.

[4] Davis K. H., "Automatic Recognition of Spoken Digits", Journal of the Acoustical Society of America, vol. 24, no. 6, pp. 637-642, 1952.

[5] Deller J. R., Hansen J. H., Proakis J. G., "Discrete-Time Processing of Speech Signals", IEEE Press, p. 936, 2000.

[6] Doddington G.R., "Speaker Recognition - Identifying People by their Voices", Proceedings of the IEEE, vol. 73, no. 11, pp. 1651-1664, 1985.

[7] Forney G. D., "The Viterbi algorithm", Proceedings of the IEEE, vol. 61, no. 3, pp. 268-278, 1973.

[8] Gajic B., "Auditory Based Methods for Robust Speech Feature Extraction", Telektronikk 2, pp. 45-58, 2003.

[9] Gajic B., Paliwal K. K., "Robust Feature Extraction using Subband Spectral Centroid Histograms", Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), pp. 85-88, 2001.

[10] Gajic B., Paliwal K. K., "Robust Speech Recognition in Noisy Environments Based on Subband Spectral Centroid Histograms", IEEE Transactions on Speech and Audio Processing, pp. 1-9, 2006.

[11] Ghitza O., "Auditory Models and Human Performance in Tasks Related to Speech Coding and Speech Recognition", IEEE Transactions on Speech and Audio Processing, vol. 2, no. 1, pp. 115-131, 1994.

[12] James W. G., Norbert K., "Speaker Identification Based on Nasal Phonation", The Journal of the Acoustic Society of America, vol. 43, no. 2, 1968.

[13] Juang B-H. and Rabiner L. R., "The Segmental K-Means Algorithm for Estimating Parameters of Hidden Markov Models", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, no. 9, pp. 1639-1641, 1990.

[14] Kedem B., "Spectral Analysis and Discrimination by Zero-Crossings", Proceedings of the IEEE, vol. 74, no. 11, pp. 1-5, 1986.

[15] Kim D. S., Lee S. Y., Kil R. M., "Auditory Processing of Speech Signals for Robust Speech Recognition in Real-World Noisy Environments", IEEE Transactions on Speech and Audio Processing, vol. 7, no. 1, pp. 55-69, 1999.

[16] Kim D. S., Jeong J. H., Kim J. H., Lee S. Y., "Feature Extraction Based on Zero-Crossings with Peak Amplitudes for Robust Speech Recognition in Noisy Environments", Korea Advanced institute of Science and Technology, pp. 61-64, 1996.

[17] Koenig W., "The Sound Spectrograph", Journal of the Acoustical Society of America, vol. 17, pp. 19-49, 1946.

[18] Leon-Garcia A., "Probability and Random Processes for Electrical Engineering", Second Edition, Addison-Wesley, Canadá, p. 596, 1994.

[19] Mohamed F. B., "Joint Speech and Speaker Recognition", Tese de Doutorado, Swiss Federal Institue of Technology Lausanne (EPFL), p. 123, 2005.

[20] Molla k. I., Keikichi H., "On the Effectiveness of MFCCs and their Statistical Distribution Properties in Speaker Identification", IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems, pp. 136-141, 2004.

[21] Oppenheim A. V., Schafer R. W., "Discrete-Time Signal Processing", Englewood Cliffs, NJ: Prentice Hall, p. 796, 1989.

[22] Picone J. W., "Signal Modeling Techniques in Speech Recognition", Proceedings of the IEEE, vol. 81, no. 9, pp. 1215-1247, 1993.

[23] Rabiner, L. R., "A Tutorial on Hidden Markov Models and selected Applications in Speech Recognition", Proceedings of the IEEE, vol. 85, no. 9, pp. 1437-1462, 1997.

[24] Rabiner L. R., Juang B., "Fundamentals of Speech Recognition", Prentice Hall, p. 493, 1993.

[25] Rabiner L. R., Wilpon J. G., Soong F. K., "High Performance Connected Digit Recognition Using Hidden Markov Models", IEEE Transactions on Acoustic, Speech, and Signal Processing, vol. 37, no. 8, pp. 1214-1225, 1989.

[26] Riquelme C. D., Cataldo E., Silva D., Alcaim A., Apolinário J. A., "Comparação entre as técnicas MFCC e ZCPA para reconhecimento robusto de locutor em ambientes ruidosos", XXX CNMAC - Congresso Nacional de Matemática Aplicada e Computacional, 2007.

[27] Silva D., Riquelme C. D., Alcaim A., "Reconhecimento Robusto de Locutor Baseado nos Atributos ZCPAC", XXV Simpósio Brasileiro de Telecomunicações - SBrT 2007, pp. 1-5, 2007.

[28] Solaiman B., "Processus stochastiques pour l'ingénieur", Première édition, PPUR-GET, France, p. 596, 2006.

[29] Stevens S. S., Volkman J., "The relation of pitch to frequency", American Journal of Psychology, vol. 53, p. 329, 1940.

[30] Young S., Evermann G., Gales M., "The HTK Book (for HTK Version 3.3)", Cambridge University Engineering Department, p. 354, 2005.

[31] http://htk.eng.cam.ac.uk/

[32] http://www.speech.cs.cmu.edu/comp.speech/Section1/Data/noisex.html