Curso de Ciência da Computação
Universidade Estadual de Mato Grosso do Sul

# ESTUDO E ANÁLISE DE MÉTODOS PARA RECONHECIMENTO DE PALAVRAS DITAS

Raiza Artemam de Oliveira
Willian Sousa Santos

Prof. MSc. André Chastel de Lima (Orientador)

DOURADOS-MS
2015

# Estudo e Análise de Métodos para Reconhecimento de Palavras Ditas

Raiza Artemam de Oliveira

Willian Sousa Santos

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso devidamente corrigida e defendida por Raiza Artemam de Oliveira e Willian Sousa Santos e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação.

Dourados, xx de novembro de 2015

Prof. MSc. André Chastel de Lima

Curso de Ciência da Computação
Universidade Estadual de Mato Grosso do Sul

# ESTUDO E ANÁLISE DE MÉTODOS PARA RECONHECIMENTO DE PALAVRAS DITAS

Raiza Artemam de Oliveira
Willian Sousa Santos

Outubro de 2016

BANCA EXAMINADORA:

Prof. MSc. André Chastel Lima (Orientador)
Área de Computação – UEMS

Profa. [Titulação] Nome do professor
Área de Computação – UEMS

Profa. [Titulação] Nome do professor
Área de Computação – UEMS

*Computer science is no more about computers than astronomy is about telescopes, biology is about microscopes or chemistry is about beakers and test tubes. Science is not about tools, it is about how we use them and what we find out when we do..*

*Edgar Dijkstra*

**Agradecimentos**

Gostariamos de agradecer ao professor André Chastel Lima pela dedicação e paciência durante o desenvolvimento deste trabalho. A professora Maria de Fátima pela ajuda no desenvolvimento do texto. Ao professor coordenador Nilton Cezar de Paula. A secretária dona Jandira pela atenção dedicada as nossas vidas acadêmicas. Por fim agradecemos a todos os professores que contribuiram nessa jornada.

Eu, Raiza, agradeço aos meus pais, Eneias e Sandra, por todo o apoio, generosidade, educação e valores que me ensinaram. Aos meus tios, Marcia e Juraci, por serem sempre prestativos e generosos comigo. Aos meus avós, Georgina e Otávio, e , Maria Lucilene (*in memorian*) e João. Por fim agradeço a professora Adriana Betania de Paula Molgora por sua orientação nos meus primeiros passos na vida acadêmica e por propiciar a oportunidade de desenvolver projetos de iniciação cientifica.

.

**RESUMO**

faca um resumo

**Palavras-chave:** Resumo. Palavras chaves . .

## SUMÁRIO

## Lista de siglas

ALSA - Advanced Linux Sound Architeture

API - Application Programming Interface

DCT - Discrete Cosine Transform

FFT - Fast Fourier Transform

HMM - Hiden Markov Model

MFCC - Mel Frequency Cepstral Coefficients

PCM - Pulse Code Modulation

RIFF - Resource Interchange File Format

WAVE - Waveform Audio File Format

## Lista de tabelas

## Lista de ilustrações

# 1 INTRODUÇÃO

Nos primeiros sistemas computacionais a comunicação entre pessoas e máquinas era realizada através de terminais por linha de comando. Apenas especialistas conseguiam utilizar estes sistemas. Depois, no início da década de 70, com a criação do mouse e a introdução da interface gráfica os sistemas tornaram-se mais amigáveis ao usuário, podendo ser utilizados por pessoas comuns sem necessidade de conhecimento técnico. Com o passar dos anos a interação entre pessoas e máquinas tornou-se mais intuitiva com as diversas interfaces entre o usuário e o sistema. No fim da década de 70 iniciaram-se as pesquisas de reconhecimento de fala. Interfaces por meio de fala são utilizadas em diversas áreas, tais como: sistemas embarcados, automação residencial, operações bancárias, conversão fala texto e dispositivos móveis.

O reconhecimento da fala é um campo de estudo amplo e necessário as diversas tecnologias que utilizam desta como um meio de comunicação entre o usuário e o sistema. Utilizar a fala como entrada de um sistema torna a comunicação entre o usuário e o sitema mais direta, intuitiva, rápida e precisa.

Como um campo de ampla aplicacação, o reconhecimento de fala tem diversos projetos em diferentes partes do mundo. Dentre os quais se destaca o projeto CMU Sphinx da universidade americana Carnegie Mellon. O projeto já tem cerca de 20 anos de pesquisas na área de reconhecimento de fala e de voz. Trata-se de um projeto open source voltado para linux, mas também conta com uma versão em java multiplataforma. O CMU Sphinx oferece suporte para várias linguagens, dentre elas o inglês, alemão, russo, francês e espanhol. O reconhecimento de fala pode ser classificado de acordo com o tamanho do vocábulario, de acordo com os algoritmos utilizados e de acordo com o tipo de fala a ser reconhecida (contínua ou discreta).

## 1.1 Justificativa

O reconhecimento de palavaras ditas é um campo de estudo de extrema importância para uma melhor comunicação entre usuários e sistema.

## 1.2 Objetivos

O objetivo deste trabalho é estudar os principais métodos de reconhecimento de fala. Analisar os algoritmos utilizados, suas vantagens e desvantagens. Apresentar os resultados para um pequeno vocábulario.

### 1.2.1 Objetivo geral

Estudar e analisar os algoritmos existentes para o reconhecimento de palavras ditas em um vocábulario pequeno e um ambiente não controlado.

### 1.2.2 Objetivo específico

Apontar a melhor solução para reconhecimento de palavras ditas em ambientes não controlados.

## 1.3 Metodologia

A metodologia adotada para a realização deste trabalho consiste nos seguintes passos:

- ❏ Pesquisa em livros, sites, artigos e notas de aula sobre o tema abordado e seus diversos aspectos;
- ❏ Estudo de algoritmos aplicados ao reconhecimento de fala;
- ❏ Implementação computacional de algoritmos aplicados ao reconhecimento de fala;
- ❏ Testes e validação dos algoritmos implementados;
- ❏ Análise e validação dos resultados obtidos com os métodos implementados;
- ❏ Documentação do trabalho.

# 2 FUNDAMENTAÇÃO TEÓRICA

De acordo com (RABINER L. R., 1993), os sistemas de reconhecimento de fala podem ser classificados em três grupos de acordo com a técnica utilizada. Estes grupos são :

- ❏ Reconhecedores por inteligência artificial;
- ❏ Reconhecedores por comparação de padrões;
- ❏ Reconhecedores baseados na análise acústico-fonética.

## 2.1 Sistemas de reconhecimento de fala

### 2.1.1 Reconhecedores por comparação de padrões

Estes reconhecedores usam o príncipio de que o sistema foi treinado para reconhecer os padrões. Os sistemas por reconhecimento de padrões possuem duas fases diferentes :

- ❏ Treinamento;
- ❏ Reconhecimento.

Durante a fase de treinamneto são criados padrões de referência para o sitema. Na fase de reconhecimento compara-se os padrões obtidos com os padrões de referência criados na fase anterior e calcula-se uma medida de similaridade entre os padões. O padrão mais similar ao desconheido é escolhido como reconhecido. Os sistemas que se baseiam nos Modelos Ocultos de Markov (HMM) se encaixam nesta categoria.

Dentre as diversas razões para usar a abordagem de comparação de padrões para reconheimento de fala podemos citar a simplicidade de uso, por ser um método de fácil entendimento que possui uma rica fundamentação matemática e é amplamente utilizado, e a robustez, trata-se de um método robusto e invariante para diferentes vocábularios, algoritmos de comparação de padrão e regras de decisão. Isto torna esta abordagem apropriada para uma vasta gama de unidades de fala, como fonemas, palavras isoladas ou frases (RABINER L. R., 1993).

### 2.1.2 Reconhecedores baseados na análise acústico-fonética

Os sistemas baseados na análise acústico-fonética decodificam o sinal de fala baseados nas características acústicas deste sinal e na relação entre elas (INCER, 1992). Os sistemas de análise desta classe devem considerar propriedades acústicas invariantes. Entre estas características estão a classificação entre sonoro e não sonoro, segmentção do sinal da fala, detecção das características que descrevem as unidades fonéticas e escolha do padrão que mais corresponde à sequencia de unidades fonéticas.

Os reconhecedores baseados na análise acústico-fonética trabalham em duas etapas. O primeiro passo na análise acústico fonética é chamado de fase de segmentação e rotulagem (RABINER L. R., 1993). Este passo envolve a segmentação do sinal da fala em regiões discretas, no tempo, onde as propriedades acústicas do sinal são representadas por um único fonema, ou estado. Em seguida uma ou mais etiqueta fonética é associada a cada região segmentada de acordo com as propriedades acústicas. O segundo passo para o reconhecimento tenta determinar uma palavra válida a partir da sequência de etiquetas fonéticas obtidas na fase anterior. As palavras são obtidas a partir de um determinado vocabulário, as palavras obtidas fazem sentido sintático e tem significado semântico.

### 2.1.3 Reconhecedores baseados em inteligência artificial

Os sistemas de reconhecimento de fala que utilizam a inteligência artificial usa propriedades tanto dos reconhecedores por comparação de padrões quanto dos reconhecedores baseados na análise acústico-fonética. Sistemas com redes neurais são encaixados nesta classe. As redes Multilayer Perceptron usam uma matriz de ponderação que representa as conexões entre os nós da rede, e cada saída esta associada a uma unidade a ser reconhecida (MORGAN; SCOFIELD, 1991).

A abordagem de inteligência artificial se baseia no processo humano natural de ouvir, analisar e tomar uma decisão sobre as características acústicas medidas para reconher a fala. Faz parte do processo de reconheciemento de fala pela abordagem de inteligência artificial o processo de segmentação e rotulagem usado na análise acústico-fonética (RABINER L. R., 1993). Esta abordagem aplica o conceito de que o conhecimento é dinâmico e os modelos devem adaptar-se frequentemente.

# 3 PRÉ-PROCESSAMENTO

A captura do sinal de áudio é uma parte fundamental para o desenvolvimento de um sistema reconhecedor de fala. O som se propaga no ambiente por meio de ondas de forma contínua no tempo e no espaço a uma velocidade média de *340 metros/segundo* fazendo ar vibrar. Esta onda sonora é capturada por meio de um microfone como uma onda analógia e é convertida para um sinal digital. A onda capturada é normalizada através de um filtro de passa-baixas. Circuitos que realizam esta conversão de onda são chamados de ADC ( *analog digital converter*). O tamanho das amostras, expressa em bits, é um dos fatores que determina a precisão com que o som é representado em forma digital. Outro fator importante que afeta a qualidade de som é a taxa de amostragem. O teorema de Nyquist afirma que a frequência mais elevada que pode ser representado com precisão é, no máximo, metade da taxa de amostragem (PROAKIS; MANOLAKIS, 1996).

## 3.1 Captura de Áudio

Para o processo de reconhecimento de fala de qualquer tipo, primeiro é necessário capturar o sinal de áudio. A fase de captura de áudio é essencial para o bom desempenho do projeto. Existem diversas bibliotecas open-source que oferecem funções que realizam a captura e gravação de áudio, entre elas a Allegro e OpenGL, entretanto a aplicação dessas bibliotecas implica em um maior custo computacional, uma vez que estas trazem milhares de linhas de código junto com outras funções além das necessárias para a implementação deste projeto. Com base nisso, buscou-se uma alternativa que integrasse eficiência e baixo custo computacional para aplicações em áudio.

### 3.1.1 ALSA

ALSA (*advanced linux sound architeture* ) consiste de um conjunto de drivers do kernel, uma biblioteca, uma API e programas utilitários para o suporte de som no linux. Jaroslav Kysela iniciou o projeto ALSA porque os drives de som do kernel Linux não estavam sendo devidamente mantidos e atualizados. Após a iniciativa mais desenvolvedores aderiram ao projeto e a estrutura da API foi refinada. ALSA foi incorporada ao kernel oficial do Linux 2.5. A biblioteca fornecida pelo ALSA, libasound, fornece uma nomeação lógica dos dispositivos de hardware. Os nomes podem ser de dispositivos de hardware reais ou plugins (TRANTER, 2004). Os dispositivos de hardware usam o formato $HW : i, j$, onde $i$ é o número

do cartão e $j$ do dispositivo do cartão. Uma placa de som tem um buffer de hardware que armazena amostras gravadas. Quando este buffer enche, ele gera uma interrupção. O driver de som do kernel, em seguida, utiliza o acesso direto à memória para transferir as amostras para um buffer de aplicativo na memória. O tamanho deste buffer pode ser programado por chamadas da biblioteca ALSA. Caso o buffer seja muito grande a tranferencia geraria uma latência excessiva. ALSA resolve isso dividindo o buffer em fragmentos e transfere os dados fragmentados. A Figura 1 ilustra a repartição do buffer em fragmentos, molduras e amostras.

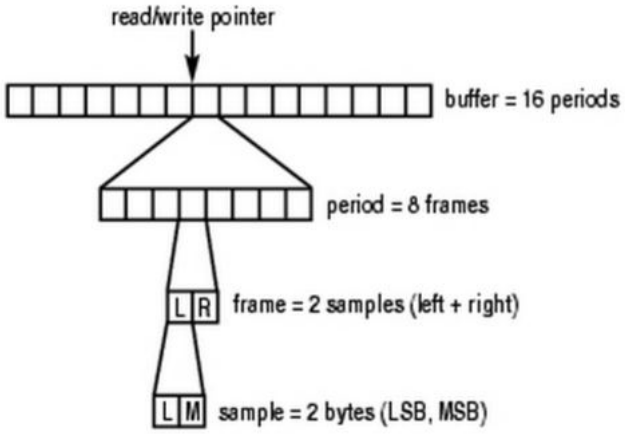


**Figura 1:** Buffer de aplicação. *fonte:(TRANTER, 2004)*

A API ALSA oferece seis principais interfaces. São elas a interface de controle, interface MIDI raw, interface de tempo, interface de sequência, interface mixer e interface de PCM. Esta última gerencia a captura e reprodução de áudio digital.

### 3.2 Arquivos WAVE

O formato de áudio adotado foi o WAVE. Neste tipo de formato o som é armazenado em sequências numéricas. O áudio é convertido em dados e armazenado bit a bit. O WAVE (.wav) foi criado pela IBM e pela Microsoft, nos anos oitenta e tem suporte a uma série de resoluções de bit, taxas de amostragens e canais de áudio. A taxa de amostragem em arquivo .wav refere-se ao número de amostras por segundo. O CD possui uma taxa de amsotragem de $44,100$, o que significa que cada segundo de áudio tem $44,100$ amostras. A quantidade de bits usada determina quanta informação pode ser armazenada no arquivo. A quantidade de bits também interfere na amplitude do sinal. Em uma gravação de 8 bits estará disponível 256 níveis de amplitude, variando de 0 à 255. Em uma gravação de 16 bits a quantidade de níveis

de amplitude disponíveis passa a $65,536$, variando entre $-32,768$ até 32767. A quantidade de 16 bits é suficiente para este projeto.

### 3.2.1 Cabeçalho WAVE

O cabeçalho de um arquivo .wav possui 44 bytes e é organizado como mostrado na Tabela 1.

**Tabela 1:** Formato de um cabeçalho de arquivo wave

| Posição | Valor | Descrição |
|---|---|---|
| 1 - 4 | RIFF | Define como um arquivo RIFF |
| 5 - 8 | Tamanho do arquivo (int) | Tamanho máximo do arquivos em bytes |
| 9 - 12 | "WAVE" | Arquivo tipo cabeçalho wave |
| 13 - 16 | "fmt" | Marca formato chunk |
| 17 - 20 | 16 | Tamanho do formato dos dados |
| 21 - 22 | 1 | Formato tipo PCM |
| 23 - 24 | 2 | Quantidade de canais |
| 25 - 28 | 44100 | Taxa de amostragem (sample rate) |
| 29 - 32 | 176400 | (sample rate * bitspersample * channels) / 8 |
| 33 - 34 | 4 | limites |
| 35 - 36 | 16 | Quantidade de bits por amostra |
| 37 - 40 | data | Marca o início da seção de dados |
| 41 - 44 | Tamanho do arquivo (dados) | Tamanho da seção de dados |

# 4 ATRIBUTOS MFCC

O primeiro passo para reconhecimento de fala é a extração de carcteristicas do sinal sonoro. Trata-se de algoritmos baseados na análise acustico fonética. Os atributos MFCC são extraídos do sinal sonoro de acordo com a escala *mel*, esta foi criada a partir de estudos com base na psicoacústica.

## 4.1 Psicoacústica

A psicoacústica estuda a relação entre estímulos sonoros e as sensações auditivas decorrentes destes estímulos. Pode ser dividida em psicoacústica externa ou interna. A primeira trata da quantização das sensações auditivas e estabelece relações matemáticas entre os estímulos acústicos e as sensações auditivas. Já a psicoacústica interna estuda os mecanismos fisiológicos responsáveis pela transformação do estímulo sonoro em sensações auditivas. A partir destes estudos é possível explicar processos como mascaramento no tempo e na frequência, discriminação de frequências, entre outros.

## 4.2 A escala *mel*

A frequência ouvida pelo sistema auditivo humano é subjetiva e varia de acordo com cada indivíduo. Esta impressão subjetiva de frequência é a sensação subjetiva da intensidade ou a amplitude de um som. O *pitch* é uma variável psicoacústica, e foi proposto por Stevens, Volkmann e Newmann em 1937. A escala *mel* é uma escala de pitches julgados pelos ouvintes como sendo igual em distância um do outro. O ponto de referência entre esta escala e a medição de freqüência normal é definida igualando um tom de 1000 Hz , 40 dB acima do limiar do ouvinte , com um pitch de 1000 *mels*. Abaixo de cerca de 500 Hz as escalas de *mel* e Hertz coincidem, acima disso intervalos cada vez maiores são julgados por ouvintes para produzir iteração igual aos pitches. A escala *mel* é baseada em um mapeamento entre a frequência real e o pitch aparentemente percebido do sistema auditivo humano. Para converter uma frequência em escala *mel* aplica-se a equação 1.

$$M(f) = 1125ln(1 + \frac{f}{700}) \tag{1}$$

## 4.3 Frequência *mel*

A análise de filtros mel, como já foi citado, baseia-se na audição humana, tal como o ouvido funciona como um filtro onde algumas frequências são análisadas e outras ignoradas. Os atributos da frequência *mel* são obtidos através da análise mel cepstral. O processo de análise cepstral consiste na conversão do sinal em cepstros. Um cepstro é o produto da aplicação da FFT sobre o sinal em escala logarítimica (TYAGI; WELLEKENS, 2005). A FFT possibilita diminuir o tempo de processamento em aplicações que requerem grande quantidade de cálculos. A extração de vetores de características MFCC é feita em etapas e esta ilustrada na Figura 2.

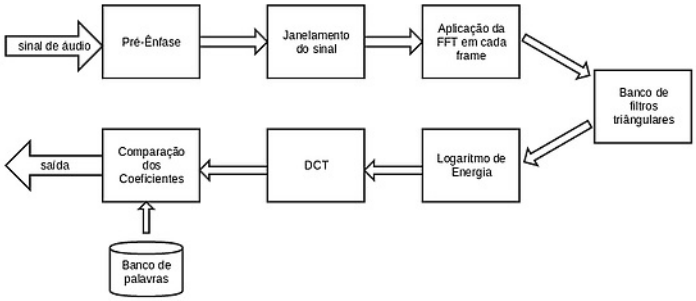


**Figura 2:** Etapas para extração de coeficientes MFCC. *fonte: Autoria própria*

O sinal de voz a ser parametrizado é passado através do filtro de pré-ênfase. Após o sinal ser filtrado, é necessário atenuar as descontinuidades causadas no início e no final de cada segmento, aplicando uma janela de Hamming de 25 ms de comprimento, com deslocamento de 10 ms, obtendo-se assim vetores MFCC a cada 10 ms. A terceira etapa consiste em aplicar a FFT para obter o espectro. Após a aplicação da FFT, aplica-se a equação 2 para obter a potência espectral.

$$S[k] = |X[k]|^2 = (real(X[k]))^2 + (imaginaria(X[k]))^2 \tag{2}$$

A próxima etapa consiste na aplicação do banco de filtros *mel* à potência espectral. Os filtros *mel* são definidos de acordo com a função 3.

$$H_m[k] = \begin{cases} 0 & k < k[m-1] \\ \dfrac{2(k-k[m-1])}{(k[m+1]-k[m-1])(k[m]-k[m-1])}, & k[m-1] \leq k \leq k[m] \\ \dfrac{2(k[m+1]-k)}{(k[m+1]-k[m-1])(k[m+1]-k[m])}, & k[m] \leq k \leq k[m+1] \\ 0 & k > k[m+1] \end{cases} \tag{3}$$

A Figura 3 mostra o banco de filtros usados na técnica MFCC. Cada filtro calcula a média do espectro em torno de um espectro central. Quanto maior a frequência, maior é a largura da banda.

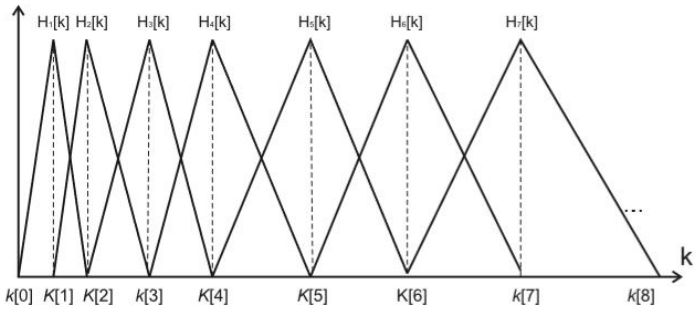


**Figura 3:** Banco de filtros triângulares MFCC. *fonte: (GORDILLO, 2013)*

Para determinar matematicamente os segmentos, parte- se da frequência extremas $f_l$ e $f_h$ que são as frequências de corte do banco de filtros em Hz. Esses valores são usados para dividir o intervalo em $B+1$ partes iguais. Para obter os valores em Hz, basta aplicar a função inversa 4.

$$k[m] = (\frac{N}{F_s})Mel^{-1}(Mel(f_l) + m\frac{Mel(f_h) - Mel(f_l)}{M+1}) \tag{4}$$

onde $F_s$ é a frequência de amostragem em Hz, M é o número de filtros e N o número de amostras da FFT. $k[m]$ são as frequências digitais e $Mel^{-1}$ determina a largura do banco de filtros e é dado por

$$Mel^{-1}(m) = 700(e^{\frac{m}{1125}} - 1) \tag{5}$$

Em seguida, obtém-se a log-energia da saída de cada um dos filtros *mel*. Por fim os coeficientes MFCC são obtidos aplicando a DCT ao logaritmo dos coeficientes de energia obtidos no passo anterior.

# 5 MODELOS OCULTOS DE MARKOV

Um modelo de Markov pode ser definido como um conjunto finito de estados ligados entre si por transições, formando uma máquina de estados. Estas transições estão ligadas por um processo estocástico . Há ainda um outro processo estocástico associado a um modelo de Markov, que envolve as observações de saída de cada estado. Se somente as observações de saída forem visíveis a um observador externo ao processo, diz-se então que os estados estão ocultos.

Um HMM é caracterizado por:

- ❏ Um conjunto de estados $S = \{S_1, S_2, \ldots, S_{n-1}, S_n\}$, onde $n$ é o número de estados;
- ❏ Função de probabilidade de estado inicial $\pi = \{\pi_i\}$ .

$$\pi_i = P[q_1 = S_i] \quad 1 \leq i \leq n \tag{6}$$

onde $q_1$ é o estado inicial $(t = 1)$.

- ❏ Função de probabilidade de transição A;
- ❏ Função de probabilidade de símbolos de saída B.

Considerando exclusivamente processos em que as probabilidades de transição não dependem do tempo e os HMMs são de primeira ordem, o conjunto de probabilidades de transição $A$ é definido por:

$$A = \{a_{ij}\} \tag{7}$$

$$a_{ij} = P[q_{t-1} = S_i][q_t = S_j] \quad 1 \leq i, j \leq n \tag{8}$$

onde $a_{ij}$ é a probabilidade de ocorrer uma transição do estado $S_i$ para o estado $S_j$.
Os coeficientes $a_{ij}$ devem obedecer às seguintes regras:

$$a_{ij} \geq 0 \quad 1 \leq i, j \leq n \tag{9}$$

$$\sum_{j=1}^{n} a_{ij} = 1 \quad 1 \leq i \leq n \tag{10}$$

A probabilidade de estar no estado $S_j$ no instante de tempo $t$ depende somente do instante de tempo $t_1$.

## 5.1 HMM e a função densidade de probabilidade

Um HMM também pode ser classificado de acordo com a função densidade de probabilidade.

### 5.1.1 Função densidade de probabilidade

Uma variável aleatória é uma função cujo valor é um número real determinado por cada elemento em um espaço amostral. Dada uma variável aleatória $X$, dizemos que $f(x)$ é uma função densidade de probabilidade de $X$, se e somente se $f(x)$ atender as seguintes condições:

$$f(x) \geq 0 \qquad a < x < b$$

$$\int_a^b f(x)dx = 1$$

### 5.1.2 HMM Discreto

O número de possíveis símbolos de saída é finito (RABINER L. R., 1993). A probabilidade de emitir o símbolo $V_k$ no estado $S_i$ é dada por $b_i(k)$. As propriedades da função de probabilidade $B$ são:

$$b_i(k) \geq 0 \qquad 1 \leq i \leq n \qquad 1 \leq k \leq K$$

$$\sum_{k=1}^{K} b_i(k) = 1 \qquad 1 \leq i \leq n$$

As observações são discretas por natureza ou discretizadas através de uma técnica de quantização vetorial, gerando assim codebooks.

### 5.1.3 HMM Contínuo

A função densidade de probabilidade é contínua. Geralmente uma função densidade elipticamente simétrica, tal como a função densidade de probabilidade Gaussiana (RABINER

L. R., 1993). As observações são contínuas e a FDP contínua é usualmente modelada como uma mistura finita de matrizes gaussianas multidimensionais.

********** DEFINIR AQUI A FDP A SER USADA (PROVAVELMENTE A GAUSSIANA CITADA EM (RABINER L. R., 1993))

### 5.1.4 HMM Semicontínuo

O modelo é um caso intermediário entre contínuo e o discreto. O conjunto função densidade probabilidade é o mesmo usado para todos os estados e todos os modelos. A probabilidade de emissão dos simbolos de saída é dada por :

$$b_j(O_t) = \sum_{V_k \in \eta(O_t)}^{-} c_j(k) f(O_t|V_k) \qquad 1 \leq j \leq n$$

onde:

$O_t$ é o vetor de entrada

$\eta(O_t)$ é o conjunto das funções densidade de probabilidade que apresentam os $M$ maiores valores de $f(O_t|V_k)$, $1 \leq M \leq K$

$K$ é o número de funções densidade de probabilidade, ou seja, os símbolos de saída

$V_k$ é o $k$-ésimo símbolo de saída

$c_j(k)$ é a probabilidade de emissão do símbolo $V_k$ no estado $S_j$

$f(O_t|V_k)$ é o valor da $k$-ésima função densidade de probabilidade.

## 5.2 Topologia

Uma maneira de classificar um HMM é de acordo com a estrutura de transição da matriz $A$ da cadeia de markov. Existem vários modelos de HMM, tal como o ergódico totalmente conectado onde qualquer estado pode ser alcançado com um único passo, o modelo de caminhos paralelos e o modelo "left-right", também chamado de modelo Bakis. Para o reconhecimento de fala este último é o mais usado (RABINER L. R., 1993).

********COLOCAR AQUI UM MODELO DE BAKIS FAZER A MATRIZ

# 6 Resultados

Os métodos de estudados de filtragem de sinal, codificação de áudio e reconhecimento de fala foram implementados em linguagem computacional C, complilada com GCC no sistema operacional Linux Ubuntu 12.04 LTS.

## 6.1 Estrutura do Código

A estrtura das funções implementadas é mostrada na Figura **??**.

### 6.1.1 Captura e filtragem do sinal

A API utilizada para a captura de áudio foi a ALSA por meio de funções. Uma estrutura de programa básico usando esta API esta sempre na forma:

```
Abre uma interface para captura ou reprodução
setar os parâmetros de hardware
Enquanto houver dados a serem processados:
ler dados PCM
ou escrever dados PCM
Fecha a interface
```

## REFERÊNCIAS BIBLIOGRÁFICAS

GORDILLO, C. D. A. **Reconhecimento de Voz Contínua Combinando os Atributos MFCC e PNCC com Métodos de Robustez SS, WD, MAP e FRN**. Dissertação (Mestrado) — PUC-RJ, 2013.

INCER, A. N. **Digital Speech Processing, Speech Coding, Syntesis and Recognition**. [S.l.]: Kluwer Academic Publishers, 1992.

MORGAN, D. P.; SCOFIELD, C. L. **Neural Networking and Speech Processing**. [S.l.]: Kluwer Academic Publishers, 1991.

PROAKIS, J. G.; MANOLAKIS, D. G. **Digital Signal Processing. Principles, Algorithms and Applications**. [S.l.]: Prentice-Hall: New Jersey, 1996.

RABINER L. R., J. B. H. **Fundamentals of Speech Recongnition**. [S.l.]: Prentice-Hall, 1993.

TRANTER, J. Introduction to sound programming with alsa. **Linux Journal**, 2004. Disponível em: <http://www.linuxjournal.com/article/6735>. Acesso em: 10.4.2015.

TYAGI, V.; WELLEKENS, C. On desensitizing the mel-cepstrum to spurious spectral components for robust speech recognition. **Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on, vol. 1**, p. 529– 532, 2005.