Dissertação de Mestrado

Inatel
Instituto Nacional de Telecomunicações

MÉTODOS DE INTERPOLAÇÃO PARA RECONSTRUÇÃO DE DADOS FALTANTES EM SISTEMAS ROBUSTOS DE RECONHECIMENTO AUTOMÁTICO DE FALA

ELAINE PEREIRA LIMA SCARTEZZINI

JULHO/ 2015

MÉTODOS DE INTERPOLAÇÃO PARA RECONSTRUÇÃO DE DADOS FALTANTES EM SISTEMAS ROBUSTOS DE RECONHECIMENTO AUTOMÁTICO DE FALA

ELAINE PEREIRA LIMA SCARTEZZINI

Dissertação apresentada ao Instituto Nacional de Telecomunicações, como parte dos requisitos para obtenção do Título de Mestre em Telecomunicações.

ORIENTADOR: Prof. Dr. Carlos Alberto Ynoguti

Santa Rita do Sapucaí
2015

Scartezzini, Elaine Pereira Lima
S287m

Métodos de interpolação para reconstrução de dados faltantes em sistemas robustos de reconhecimento automático de fala. / Elaine Pereira Lima Scartezzini. – Santa Rita do Sapucaí, 2015.

57p.

Orientador: Prof. Dr. Carlos Alberto Ynoguti.

Dissertação de Mestrado – Engenharia de Telecomunicações – Instituto Nacional de Telecomunicações – INATEL.

Inclui bibliografia.

1. Reconhecimento Automático 2. Reconstrução 3. Dados Faltantes 4. Engenharia de Telecomunicações. I. Ynoguti, Carlos Alberto. II. Instituto Nacional de Telecomunicações – INATEL. III. Título.

CDU 621.39

FOLHA DE APROVAÇÃO

Dissertação defendida e aprovada em _____ /_____ /_____ ,
pela comissão julgadora:

_____

Prof. Dr. Carlos Alberto Ynoguti
INATEL

_____

Prof. Dr. Estevan Marcelo Lopes
INATEL

_____

Prof. Dr. Francisco José Fraga da Silva
UFABC

_____

Coordenador do Curso de Mestrado

Aos meus pais,
irmãs e marido pelo
apoio incondicional.

AGRADECIMENTOS

Primeiramente agradeço a Deus, por ter me concedido saúde e força para concluir este trabalho.

Aos meus queridos pais Mauri e Dirce pelo exemplo e por me ensinarem os verdadeiros valores da vida e as minhas irmãs Cássia e Ana Maria pelo apoio e companheirismo.

A minha amável sobrinha Laura que chegou a pouco e aumentou ainda mais a alegria e a união da nossa família.

Ao meu marido e melhor amigo Gerson, pelo amor e apoio incondicional.

Aos amigos tanto de Santa Rita quanto de Porto Alegre, que fizeram com que a mudança de lar e cultura não fosse tão dolorosa.

Ao Professor Dr. Carlos Alberto Ynoguti pela orientação, exemplo de profissional, críticas sempre construtivas e acima de tudo enorme paciência durante toda essa caminhada.

À Gisele pela atenção dispensada a todos os alunos e as inúmeras vezes que não poupou esforços para me ajudar durante esses anos.

Aos professores, coordenadores e funcionários do INATEL pela atenção e auxílio.

Aos amigos do laboratório de desenvolvimento de hardware do INATEL que fizeram parte do meu dia a dia por quase cinco anos. Aos gestores do ICC que me permitiram conciliar as aulas com o meu trabalho no laboratório no inicio desta jornada.

Por fim, gostaria de agradecer a todos que de alguma forma contribuíram para a realização deste trabalho.

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AD	*Analog to Digital* (Conversão analógico-digital)
ASR	*Automatic Speech Recognition* (Reconhecimento Automático de Fala)
CUED	*Cambridge University Engineering Department*
DC	*Direct Current* (Corrente contínua)
DCT	*Discrete Cosine Transform* (Transformada Discreta do Cosseno)
DFT	*Discrete Fourier Transform* (Transformada Discreta de Fourier)
FDP	Função Densidade de Probabilidade
FFT	*Fast Fourier Transform* (Transformada Rápida de Fourier)
GUI	*Graphical User Interface* (Interface gráfica de usuário)
HMM	*Hidden Markov Model* (Modelo Oculto de Markov)
HTK	*Hidden Markov Model Toolkit*
LPC	*Linear Predictive Coding*
MDT	*Missing data technique* (Técnica de dados faltantes)
MFCC	*Mel-frequency Cepstral Coefficients* (Coeficientes de Frequência Mel Cepstral)
MMSE	*Minimum Mean Square Error*
RAF	Reconhecimento Automático de Fala
SNR	*Signal Noise Ratio* (Relação Sinal Ruído)
WAV	*Waveform Audio File Format*

RESUMO

A técnica de dados faltantes foi desenvolvida para melhorar o desempenho dos sistemas de reconhecimento automático de fala em ambientes ruidosos. Esta abordagem identifica e utiliza no reconhecimento somente as partes de uma locução ruidosa que não foram drasticamente corrompidas pelo ruído, partes estas que são denominadas confiáveis. Existem dois métodos principais que podem ser utilizados para atingir este objetivo: a marginalização e a imputação. A marginalização utiliza somente a informação confiável enquanto a imputação tenta substituir os elementos não confiáveis (corrompidos pelo ruído e conhecidos como dados faltantes) por estimativas baseadas nos elementos confiáveis. Este trabalho tem como objetivo principal comparar três técnicas de reconstrução dos dados faltantes no reconhecimento automático de fala utilizando imputação: interpolação linear, polinomial e racional.

Palavras-chave: Reconhecimento automático de fala em ambientes ruidosos; dados faltantes; reconstrução de dados na técnica de imputação; reconstrução de espectrogramas; Engenharia de Telecomunicações.

ABSTRACT

The missing data approach was developed to perform automatic speech recognition in noisy environments. This technique identifies and uses in the recognition process only parts of a noisy utterance signal which were not heavily corrupted by the noise, these parts are called reliable. There are two main methods that can be used to achieve this goal: the marginalization and the imputation. The marginalization method uses only the utterance reliable information, whereas the imputation method tries to substitute the unreliable parts for estimates based on the reliable information. The purpose of this paper is to compare three imputation methods: the linear interpolation, the polynomial interpolation and the rational interpolation.

Keywords: Speech recognition under noise, missing data, imputation reconstruction, interpolation reconstruction; Telecommunications Engineering.

Capítulo 1

Introdução

A fala é o meio de comunicação mais utilizado e ainda é considerado o método mais eficaz para transmissão de mensagens entre seres humanos. A necessidade de prover este mesmo tipo de comunicação na interação do homem com máquinas fez os sistemas de reconhecimento automático, conhecidos mundialmente como *Automatic Speech Recognition* (ASR), ou em português, Reconhecimento Automático de Fala (RAF), se desenvolverem amplamente nos últimos anos [1].

Com o amadurecimento desta tecnologia, esta vem sendo cada vez mais utilizada em diversas aplicações, como discagem por voz em smartphones, controles de TV´s, navegadores de web, etc.

Embora o ouvido humano possua uma enorme capacidade de distinguir sons, mesmo que estes estejam imersos em ruído, esta característica ainda não conseguiu ser totalmente reproduzida pelos sistemas de reconhecimento automático de fala [2].

Grandes investimentos nesta tecnologia de reconhecimento estão sendo realizados, principalmente em dispositivos móveis (como telefones celulares, tablets e relógios inteligentes) e aparelhos (como televisores e rádios), o que acarretou uma grande evolução, mas seu desempenho ainda não é satisfatório em todos os tipos de ambientes [3].

Uma das maiores causas dos erros apresentados pelos sistemas RAF é a necessidade de reconhecimento da fala inserida em meios ruidosos, o que dificulta a separação dos componentes que pertencem à voz dos componentes dos sons que fazem parte do ambiente em que o dispositivo reconhecedor está envolto.

Técnicas para o reconhecimento automático de fala na presença de ruídos aditivos têm sido amplamente estudadas nos últimos anos. As principais técnicas são: subtração espectral [4], normalização da média cepstral [5] e teoria dos dados faltantes [6], sendo esta última o foco deste trabalho.

Na técnica de dados faltantes não é necessário ter conhecimento prévio sobre as características do ambiente onde o sistema reconhecedor está envolto, e o seu desempenho se mantém robusto mesmo em altos níveis de ruído. Esta técnica tem como característica detectar os pontos da locução nos quais a energia da fala predomina, separando-os dos pontos onde o ruído é predominante.

A técnica de dados faltantes pode ser dividida em dois principais métodos: marginalização e imputação. A marginalização trabalha somente com os dados não corrompidos pelo ruído descartando o restante, enquanto a imputação utiliza os pontos onde a energia da fala predomina para estimar os demais. Assim, o método de imputação apresenta algumas vantagens, pois permite que o reconhecimento seja feito com uma representação tempo-frequência completa.

Esta técnica possui diversas variações e o objetivo deste trabalho é apresentar uma comparação entre os métodos de reconstrução dos dados faltantes no reconhecimento automático de fala utilizando a imputação. Em outras palavras, alguns dados de uma composição vocal foram perdidos devido à inserção de ruído e o sistema criado tentará reconstruí-los utilizando os dados considerados confiáveis (sem predominância de ruídos).

Outra contribuição deste trabalho é a apresentação de uma plataforma que possibilita a escolha de como será feita a imputação dos dados, selecionando a maneira como os

dados serão divididos entre confiáveis (predomina a energia da fala) e não confiáveis (predomina o ruído), assim como a técnica utilizada na reconstrução da locução com a imputação. A principal característica deste desenvolvimento é que a plataforma criada possibilita a inclusão de novos métodos e técnicas de maneira fácil e simples, além de automatizar o processo de imputação e reconstrução.

O conteúdo deste trabalho se divide da seguinte forma: No capítulo 2 será apresentado o sistema reconhecedor da fala e suas variantes assim como o espectrograma mel e os modelos ocultos de Markov. No capítulo 3 será analisada a influência do ruído no sinal da fala. No capítulo 4 será apresentada a técnica de dados faltantes, as possibilidades de geração da máscara de dados, a imputação e a reconstrução do espectrograma. No capítulo 5 serão explicados os experimentos e os desenvolvimentos realizados. No capítulo 6 serão analisados os resultados e por fim, no capítulo 7 será apresentada a conclusão e possíveis sugestões para trabalhos futuros.

Capítulo 2

Reconhecimento da fala

2.1 Breve história

O reconhecimento de voz tem cada vez mais se infiltrado no dia a dia dos seres humanos através de celulares, tablets e seus aplicativos que facilitam a interatividade homem máquina e abrem caminho para a inserção dos sistemas de reconhecimento em diversos outros dispositivos.

O primeiro estudo relacionado ao reconhecimento automático de fala se iniciou no ano de 1952, nos laboratórios Bell, através da criação de um reconhecedor de dígitos isolados de um único locutor. Nos anos seguintes outras grandes instituições investiram neste ramo, como a NEC, IBM e AT&T o que acarretou em grandes evoluções [3] [7].

O projeto "*ARPA SUR 5-year*" de 1970 foi considerado um marco no reconhecimento automático de fala, pois modificou os conceitos utilizados até então e colocou em prática a utilização dos modelos escondidos de Markov nesta área. Nos anos seguintes o desenvolvimento dos reconhecedores obteve resultados mais expressivos se utilizando das abordagens estatísticas, se tornando mais modernos e mais próximos dos reconhecedores utilizados na atualidade [8].

2.2 Funcionamento do sistema reconhecedor

O RAF pode ser caracterizado de diversas maneiras de acordo com alguns parâmetros como a dependência do locutor, onde se define se o sistema reconhece a fala independente de quem esteja falando; o tamanho do vocabulário que é classificado de acordo com o número de palavras e o estilo da pronúncia que define se o reconhecimento será de palavras isoladas ou de fala contínua [9]. Podemos averiguar um resumo destes parâmetros na Figura 2.1 [10].

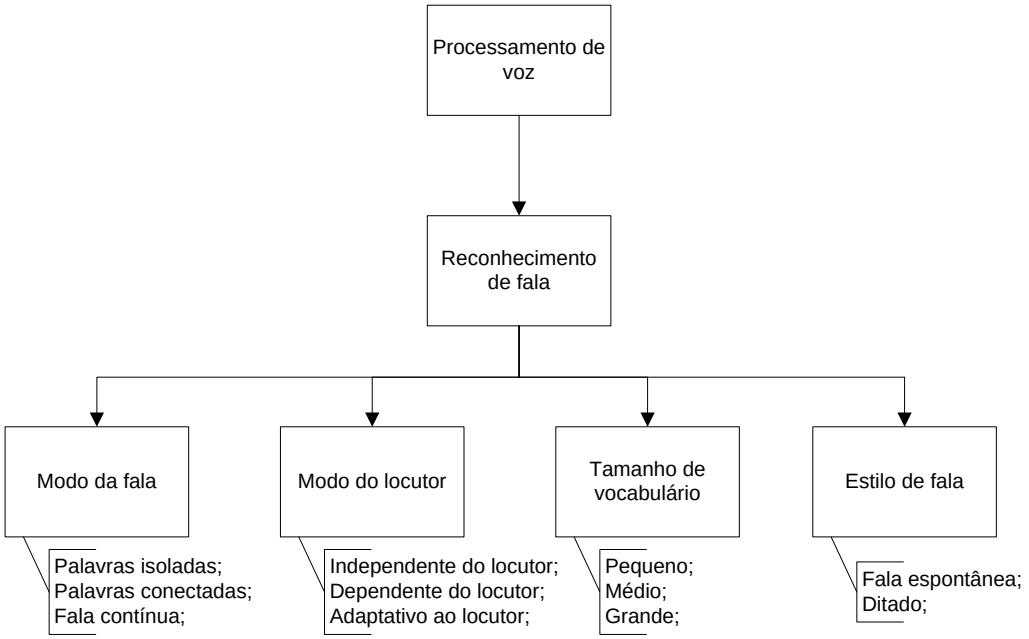

Figura 2.1 Características do sistema RAF.

Um reconhecedor tem como entrada um sinal de voz de onde são extraídos parâmetros que passam por um decodificador que utilizando modelos acústicos e de línguas previamente criados descobre a palavra falada. Estes modelos, também conhecidos como padrões são criados durante a etapa de treinamento. Assim, podemos dividir o RAF em algumas partes: aquisição da voz, normalização, janelamento e extração dos parâmetros, seguindo para a etapa final onde o reconhecimento é realizado, conforme ilustra o diagrama na Figura 2.2.

Figura 2.2 Principais etapas de um sistema RAF.

A primeira etapa de um reconhecedor é a aquisição da voz, onde as ondas acústicas são transformadas em ondas elétricas através de um transdutor, posteriormente passando por um filtro *anti-alising* e por uma conversão analógico-digital (AD). Ao final desta etapa o sinal passa por um processo de amostragem, onde a taxa de amostragem escolhida influencia diretamente na precisão do sistema.

No pré-processamento é realizada a normalização da amplitude das amostras e retirados níveis DC (D*irect Current*) que possam estar contidos no sinal. Em alguns sistemas, nesta etapa também são retirados os períodos de silêncio de uma locução que podem vir a atrapalhar o reconhecimento posterior.

As técnicas de extração dos parâmetros se mostram mais eficientes com sinais estacionários, ou seja, sinais cujas características são invariantes no tempo. A voz é um processo estocástico não estacionário, porém como a voz se altera lentamente durante uma fala contínua podemos considerá-la como estacionária tomando um curto intervalo de tempo. Assim, realiza-se a etapa de janelamento para dividir o sinal de voz em pequenos segmentos chamados de janelas ou *frames*, normalmente de 10ms a 45ms para que possamos admitir que o sinal seja estacionário nestes intervalos.

Para realizar o janelamento no domínio do tempo, multiplica-se o sinal pela função da janela utilizada. A técnica mais utilizada é a janela de Hamming definida pela Equação 2.1

$$h(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi m}{T-1}\right), 0 \leq m \leq T-1 \\ 0, caso\ contrário \end{cases}, \quad (2.1)$$

onde T representa o número total de amostras da janela e m o índice da amostra. Na prática, geralmente realiza-se o janelamento de 25ms a cada 10ms.

A extração dos parâmetros é essencial para o funcionamento satisfatório do sistema reconhecedor, visto que seria impossível analisar todas as informações contidas em um sinal de áudio, em um tempo pequeno e com grande precisão, dado que algumas dessas informações são redundantes e/ou não são significativas. Esta etapa deve utilizar a menor quantidade possível de parâmetros, de modo a prover para o reconhecedor as informações relevantes que caracterizam o sinal.

Os principais métodos de extração de parâmetros são: transformada rápida de Fourier FFT (*Fast Fourier Transform*), a análise cepstral, codificação por predição linear (LPC-*Linear Predictive Coding*) entre outros que o leitor interessado poderá obter maiores detalhes em [10] [11]. Dentre estes, o método mais utilizado é a análise cepstral, também conhecida como mel-cepstrum, de onde são extraídos os coeficientes mel-cepstrais (MFCC - *Mel Frequency Cepstral Coefficients*).

Seria intuitivo esperar que as amostras das locuções fossem igualmente espalhadas em frequência, porém verificou-se que para a voz, as frequências mais baixas são mais perceptíveis do que as frequências mais altas. Assim, criou-se um método de amostragem com espaçamento desigual, dando mais relevância às frequências mais baixas, conhecido como escala de frequência mel [12].

2.3 Escala mel

A análise da locução nas frequências Mel é baseada em experimentos na percepção humana, onde o ouvido humano funciona como um filtro que se concentra em alguns componentes de frequência. Estes filtros do ouvido humano não são uniformemente espaçados no eixo da frequência. Nas frequências mais baixas existem mais filtros do que nas frequências mais altas [13].

A escala Mel foi criada no ano de 1937 por Stevens, Volkman e Newman. Para cada valor em frequência tem-se um correspondente em mel que é a unidade desta escala. Nas frequências até 100Hz esta escala é praticamente linear, e acima deste valor é logarítmica. O valor da frequência mel é dado pela equação 2.2

$$mel(f) = 1127\ \ln(1 + \frac{f}{700})\ , \tag{2.2}$$

onde f é frequência em Hz.

Para o cálculo dos coeficientes MFCC, primeiro faz-se o quadrado do módulo da FFT das amostras da janela e depois realiza-se a filtragem destas amostras através de um banco de filtros triangulares. Normalmente são utilizados 24 filtros centrados nas frequências da escala mel, onde os 10 primeiros são distribuídos linearmente de 100 a 1000Hz, e acima de 1000Hz são distribuídos de forma logarítmica.

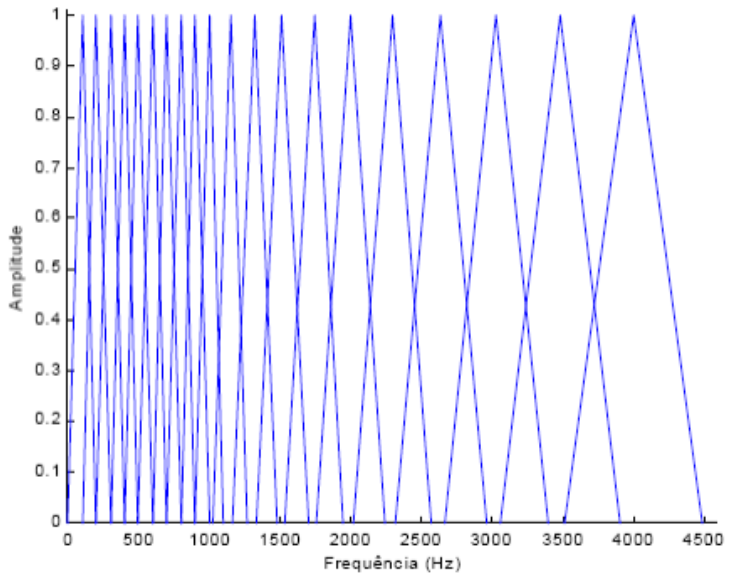

Figura 2.3 Banco de filtros da escala Mel [9].

Na Figura 2.3 temos um exemplo da distribuição de 20 filtros em um banco de filtros de escala mel.

Após a passagem pelo banco calcula-se o logaritmo da magnitude das saídas dos filtros onde, posteriormente, para se obter os coeficientes MFCC é aplicada a transformada discreta do cosseno (*Discrete Cosine Transform*, DCT) sobre os valores. No reconhecimento de fala geralmente são utilizados 12 coeficientes por janela [9].

Assim os MFCC´s são calculados através da equação 2.3

$$c_i = \sum_{k=1}^{M} 10 * \log_{10}(X_k) * \cos\left[\frac{i*(k-0.5)*\pi}{M}\right] \quad . \qquad (2.3)$$

Onde:
M é o número total de filtros;
X_k é a energia de saída do filtro;
k é o índice do filtro;
i é o índice do coeficiente MFCC.

Para o reconhecimento também são importantes os cálculos da energia e dos parâmetros diferenciais (delta-Mel-cepstrais e delta-delta-Mel-cepstrais).

Resumindo, para obter os coeficientes MFCC a partir de um sinal de voz temos que realizar as etapas ilustradas na Figura 2.4.

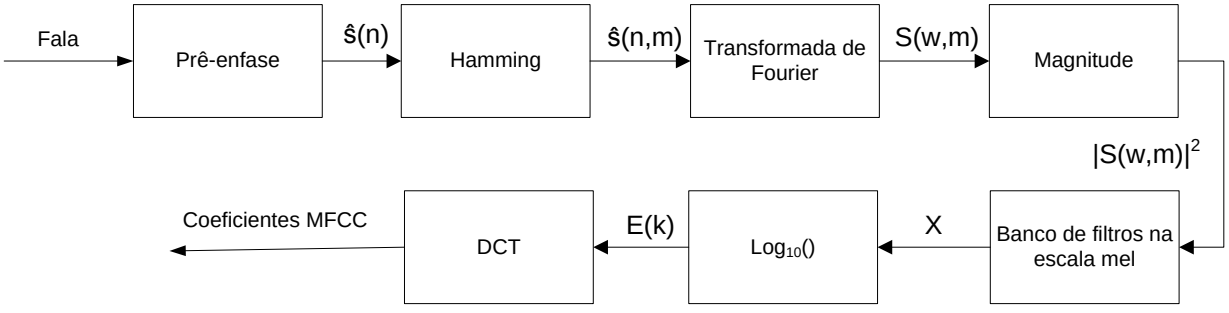

Figura 2.4 Extração dos coeficientes MFCC [14].

2.4 Modelos ocultos de Markov

Na etapa do reconhecimento da fala, a principal ferramenta utilizada são os modelos ocultos de Markov (HMM- *Hidden Markov Model*).

Os HMM foram inicialmente estudados por Leonard Baum e outros pesquisadores no final da década de 60, embora vieram a ser introduzidos na área de reconhecimento da fala somente no final dos anos 70.

Os modelos ocultos de Markov, também conhecidos como cadeias de Markov em tempo discreto, são considerados um processo duplamente estocástico, sendo um processo estocástico oculto ou escondido, que pode ser observado através de outro processo estocástico que possui uma sequência de símbolos visíveis ou observáveis [15] [16]. As saídas do processo observável são geradas de acordo com a função densidade de probabilidade (fdp) de cada estado.

Um modelo de Markov pode ser descrito como uma máquina de estados, ou seja, um conjunto finito de estados ligados entre si, que possuem probabilidades de transição e de permanência nos estados, além de probabilidades de emissão de símbolo, conforme ilustra a Figura 2.5 onde 1, 2 e 3 são os estados, a_{12},a_{13},a_{21},a_{23},a_{31} e a_{32} são as probabilidades de transição e a_{11},a_{22} e a_{33} são as probabilidades de permanência.

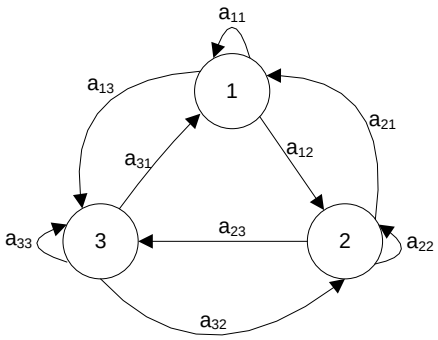

Figura 2.5 Cadeia de Markov de 3 estados.

Os principais elementos de uma cadeia de Markov são:

- Número total de estados do modelo (N)
- Estado no instante t (q_t)
- Probabilidade de iniciar a máquina no estado i (π_i)

$$\pi_i = P[q_{t1} = i], \ \ 1 \leq i \leq N \tag{2.4}$$

- Matriz de probabilidades das transições entre os estado A=(a_{ij})

$$a_{ij} = P[q_{t+1} = j | q_t = i], \ 1 \leq i, j \leq N \tag{2.5}$$

onde

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{2.6}$$

- Número de símbolos de saída distintos observáveis em cada estado(M)
- O conjunto discreto de símbolos de observação: V = (v_1, v_2,..., v_M).
- A sequência de observação O = (o_1,o_2,...,o_T)
- Distribuição de probabilidade dos símbolos observáveis em cada estado B=(b_1(k), b_2(k),...,b_j(k),...., b_N(k)), onde b_j(k) é a distribuição de probabilidade de símbolo k no estado j.

$$b_j(k) = P[o_t = v_k | q_t = j] \tag{2.7}$$

Pode-se observar que para uma definição completa do modelo de um HMM deve-se definir os parâmetros N e M, especificar os símbolos de saídas dos estados e as medidas de probabilidade A, B e π. Assim, a notação utilizada para um HMM é λ=(A,B, π).

Em uma analogia dos HMM´s com o reconhecimento de fala, consideramos a sequência no tempo do sinal da voz como sendo os estados internos de uma cadeia de Markov, e os símbolos de observação os fonemas.

Um HMM pode ser classificado dependendo da sua topologia, como modelos ergódigos, esquerda-direita paralelo ou esquerda-direita sequêncial. No modelo ergódigo todas as transições entre os estados são possíveis, quando no modelo esquerda-direita paralelo muitos caminhos são permitidos sendo possível saltar um ou mais estados. No

modelo esquerda-direita sequencial tem-se uma evolução em série podendo também haver saltos, mas sempre para frente ou direita, conforme ilustra a Figura 2.6.

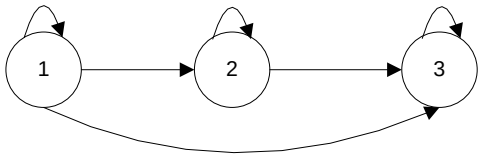

Figura 2.6 Cadeia de Markov esquerda-direita

No reconhecimento de fala, normalmente se utiliza o modelo esquerda-direita sequencial, pois o sinal da voz é contínuo e progressivo o que significa que as transições entre os estados se darão sempre da esquerda para a direita [17].

Outra característica do HMM que é utilizada no reconhecimento da fala é caracterizar os modelos de transição com funções de densidade de probabilidade (fdp) como misturas de Gaussianas, que são parametrizadas por vetores das médias (μ) e matriz de covariância (σ) [15].

Capítulo 3

Análise espectral da influência do ruído no sinal de fala

3.1 Representação da fala

Uma forma útil para representar o sinal da fala é o espectrograma, que mostra a evolução temporal da distribuição espectral da energia do sinal. O espectrograma apresenta uma maior correlação com a audição humana do que uma representação temporal [18].

O espectrograma é um gráfico que representa a intensidade do sinal com o escurecimento do traçado, com as faixas de frequência no eixo vertical e o tempo no eixo horizontal [19] [13]. A Figura 3.1 apresenta o espectrograma da palavra "*one*" da base de dados TIDIGITS [20] utilizada neste trabalho.

A voz é gerada por um conjunto de órgãos e músculos conhecidos como trato vocal. Sua geração somente é possível devido a um trabalho conjunto dos sistemas nervoso, respiratório e digestivo do ser humano.

A frequência e intensidade do som produzido dependem de diversos fatores, como o comprimento e a espessura das pregas vocais, além de sofrer influência de aspectos humanos como os hormônios e as emoções.

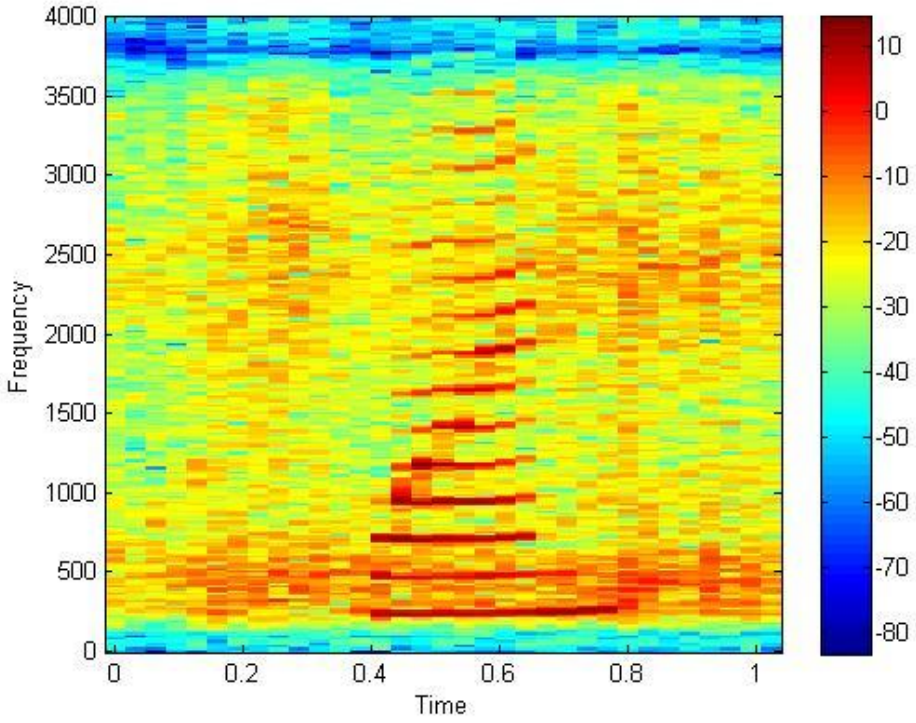

Figura 3.1 Espectrograma da palavra "*one*"

O timbre da voz varia de acordo com as cavidades que ressonam junto com as pregas vocais e seu conteúdo espectral está compreendido entre 20Hz e 20kHz, embora a maior parte da energia necessária para se entender uma locução está compreendida até 4kHz. As formações das cavidades do trato vocal geram as frequências de ressonância denominadas formantes.

A vibração das pregas conhecida como frequência fundamental (F_0), ocorre cerca de 220 vezes por segundos nas mulheres e 110 vezes por segundo nos homens. Ao mesmo tempo são geradas as frequências múltiplas inteiras de F_0 chamadas de harmônicos [21].

As estrias horizontais que podem ser vistas em um espectrograma de banda estreita da fala representam a evolução temporal dos harmônicos.

3.2 Efeitos do ruído no espectrograma mel

Usualmente o sinal da fala é afetado por ruído de fundo aditivo devido às outras fontes de áudio no ambiente em que o reconhecedor está envolvido [22]. Assim o sinal observado no domínio do tempo é definido por

$$y[m] = s[m] + n[m], \tag{3.1}$$

onde y[m] é a locução com ruído, s[m] é o sinal limpo e n[m] o ruído aditivo.

Calculando as DFT´s(*Discrete Fourier Transform*) destes sinais, no domínio da frequência o ruído ainda é aditivo

$$Y[k] = S[k] + N[k]. \tag{3.2}$$

Calculando a potência de saída de cada filtro do banco de filtros-mel temos:

$$\sum_k W_k^j |Y[k]|^2 = \sum_k W_k^j |S[k]|^2 + \sum_k W_k^j |N[k]|^2 + 2\sum_k W_k^j |S[k]|\,|N[k]| \cos(\theta[k]), \tag{3.3}$$

onde W_k^j é o peso para a DFT do índice k no filtro j do banco de filtros mel, θ[k] é o ângulo entre S[k] e N[k] e j=1, ... ,J.

A fim de simplificar a notação da equação (3.3) podemos definir:

$$Y_j^2 = \sum_k W_k^j |Y[k]|^2 \ , \tag{3.4}$$

$$S_j^2 = \sum_k W_k^j |S[k]|^2 \ , \tag{3.5}$$

$$N_j^2 = \sum_k W_k^j |N[k]|^2 \ , \tag{3.6}$$

Assim, reescrevendo a equação (3.3) têm-se:

$$Y_j^2 = S_j^2 + N_j^2 + 2\alpha_j S_j N_j \,, \tag{3.7}$$

onde

$$\alpha_j = \frac{\sum_k W_k^j |S[k]||N[k]| \cos(\theta[k])}{S_j N_j} \,. \tag{3.8}$$

No domínio log-espectral, teremos:

$$\boldsymbol{y} = \begin{bmatrix} \log Y_1^2 \\ \log Y_2^2 \\ \vdots \\ \log Y_J^2 \end{bmatrix}, \ \boldsymbol{s} = \begin{bmatrix} \log S_1^2 \\ \log S_2^2 \\ \vdots \\ \log S_J^2 \end{bmatrix}, \boldsymbol{n} = \begin{bmatrix} \log N_1^2 \\ \log N_2^2 \\ \vdots \\ \log N_J^2 \end{bmatrix} \tag{3.9}$$

e também

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_J \end{bmatrix} \quad . \tag{3.10}$$

Segundo [23] a relação entre a locução ruidosa, o sinal limpo e o ruído no domínio log-espectral pode ser definido pela equação 3.11 enquanto a relação entre eles no domínio cepstral pode ser definida pela equação 3.13, apresentadas respectivamente a seguir:

$$\boldsymbol{y} = \boldsymbol{s} + \log[1 + \exp(\boldsymbol{n} - \boldsymbol{s})] + e, \tag{3.11}$$

onde o termo *e* é definido como:

$$e = \log\left[1 + \frac{2\boldsymbol{\alpha} \circ exp\left(\frac{\boldsymbol{n}-\boldsymbol{s}}{2}\right)}{1+exp(\boldsymbol{n}-\boldsymbol{s})}\right] \,. \tag{3.12}$$

Considerando C a notação da DCT(*Discrete Cosine Transform*) da matriz, pode ser definido $y_C = Cy$, $s_C = Cs$ e $n_C = Cn$. Assim, temos:

$$\boldsymbol{y_C} = \boldsymbol{s_C} + C \log 1 + \exp\left(C^{-1}[\boldsymbol{n_C} - \boldsymbol{s_C}]\right) + Ce \ . \tag{3.13}$$

Para que a relação apresentada na equação anterior seja exata, C deve ser uma matriz DCT quadrada.

As relações apresentadas anteriormente são não lineares o que acarreta problemas quando é necessário calcular parâmetros estatísticos para a distribuição de uma locução ruidosa dada a distribuição da locução limpa e do ruído.

Capítulo 4

Técnica dos dados faltantes

4.1 Dados faltantes

A técnica de dados faltantes (MDT- *Missing Data Technique*) surgiu na Universidade de Sheffield, no Reino Unido no inicio da década de 90 [24]. Esta técnica é baseada em duas ideias principais: a) quando um sinal de voz é corrompido pelo ruído, alguns componentes espectro-temporais são mais corrompidos do que outros; e b) o sinal de voz possui redundâncias, possibilitando que o reconhecimento seja realizado baseado somente nos componentes de alta SNR (*Signal Noise Ratio*).

A eficiência dos sistemas RAF é diretamente afetada pelo ruído e outras variabilidades dos sinais de áudio. Existem diversas técnicas que lidam com ruídos aditivos em sistemas de reconhecimento automático, tentando diminuir ou até mesmo anular seus efeitos.

A técnica dos dados faltantes possibilita estimar, antes da decodificação, quais regiões espectro-temporais na representação acústica de uma locução ruidosa são confiáveis (dominados pela energia da fala) ou não confiáveis, também chamados de faltantes (dominados pelo ruído de fundo), através da análise do espectro do sinal da fala.

Uma matriz chamada matriz de dados faltantes é então criada para indicar quais partes do espectro podem ser consideradas confiáveis e quais não são. Em um processamento posterior, decide-se se os elementos não confiáveis serão ignorados ou substituídos por valores estimados no processo de decodificação.

Neste momento, definiremos algumas notações que serão seguidas no restante deste trabalho, conforme [25]. Cada quadro da representação acústica terá elementos confiáveis e elementos corrompidos pelo ruído (não confiáveis). Em um determinado quadro teremos o vetor espectral Y(t) que é composto de componentes confiáveis $Y_r(t)$ e de componentes não confiáveis $Y_u(t)$. As letras r e u utilizadas são derivadas das palavras na língua inglesa *reliable* e *unreliable*.

Assim, podemos definir

$$Y_r(t) = R(t)Y(t), \tag{4.1}$$

$$Y_u(t) = U(t)Y(t), \tag{4.2}$$

$$Y(t) = A(t)[Y_r(t)^T Y_u(t)^T]^T, \tag{4.3}$$

onde R(t) e U(t) são respectivamente as matrizes de permutação que selecionam os componentes que serão considerados confiáveis e não confiáveis, e A(t) é a matriz de permutação que reorganiza os componentes para obter Y(t) [26].

Considera-se X(t) como um vetor correspondente a Y(t), chamado de vetor verdadeiro, que representa o espectrograma que seria utilizado caso o sinal não estivesse corrompido pelo ruído [27]. Também faz-se relações entre $X_r(t)$ e $Y_r(t)$ e entre $X_u(t)$ e $Y_u(t)$ como pode ser visto nas expressões

$$X_r(t) \approx Y_r(t), \tag{4.4}$$

$$X_u(t) \leq Y_u(t). \tag{4.5}$$

Os valores de $X_r(t)$ e $X_u(t)$ não são conhecidos.

Os elementos considerados não confiáveis serão substituídos por elementos derivados dos confiáveis antes do reconhecimento, utilizando técnicas que serão descritas no item 4.3 deste capítulo.

A maior vantagem da MDT sobre outras técnicas é não ser necessário nenhum conhecimento prévio sobre o ruído que está afetando as locuções.

4.2 Máscara de dados faltantes

Na MDT é necessário criar uma matriz chamada máscara de dados faltantes, para indicar quais elementos podem ser considerados confiáveis e quais não são, ou seja, identificar os componentes espectrais onde o ruído predomina sobre a energia da fala [24].

Existem diversos métodos para estimação do ruído em espectrogramas que podem ser analisadas em [26]. As principais técnicas serão apresentadas a seguir:

- Identificação simples

  A maneira mais simples de estimar o ruído é utilizar os períodos da locução sem fala, ou seja, os períodos de silêncio. Usualmente calcula-se a média dos 10 primeiros quadros e considera- se este valor o espectro do ruído.

- Médias com pesos

  Método criado por Hirsch e Ehrlicher [28], realiza uma adaptação da estimativa das mudanças do ruído. Baseia-se na recursão de primeira ordem para estimar o ruído e utiliza um limiar adaptativo para finalizar a recursão. Para cada banda de frequência a magnitude do ruído no quadro i é obtida por

  $$se\ S(i) \leq \beta \widetilde{N}(i-1), então\ \widetilde{N}(i) = \alpha \widetilde{N}(i-1) + (1-\alpha)S(i) \qquad (4.6)$$
  $$senão\ \widetilde{N}(i) = \widetilde{N}(i-1),$$

onde S é a magnitude espectral, $\widetilde{N}$ a estimativa da magnitude do ruído, $\beta \approx 2$ e $\alpha \approx 0.98$. A inicialização é baseada na técnica de identificação simples descrita anteriormente.

- Método de segunda ordem

  Utiliza a equação 4.6 para uma superestimação do ruído em frequência onde a SNR é baixa. A recursão de segunda ordem apresentada a seguir é inserida para reduzir os erros.

$$E\left(\widetilde{N}^2\right)^i = \alpha E\left(\widetilde{N}^2\right)^{i-1} + (1-\alpha)S^2(i). \tag{4.7}$$

Após a estimativa de ruído é possível identificar os elementos confiáveis e não confiáveis da representação acústica. Para esta identificação também existem diversas técnicas, como as descritas por [25] [29]:

- Baseada na energia negativa

  O critério de energia negativa foi criado por Drygajlo e El-Maliki [30]. Considere a magnitude observável em um quadro como $|s+n|$, e a estimativa de ruído como $\hat{n}$. O elemento é considerado não confiável se

$$|s+n| - \hat{n} < 0\ . \tag{4.8}$$

- Critério SNR

  Este critério considera o elemento como não confiável quando a SNR é negativa

$$\log\left(\frac{\hat{s}^2}{\hat{n}^2}\right) < 0 \; ou \; \hat{s}^2 < \hat{n}^2 \quad , \tag{4.9}$$

  onde $\hat{s} = |s+n| - \hat{n}$.

Após a identificação dos elementos não confiáveis tem-se o que é chamada máscara dos dados faltantes, que pode ser representada por uma matriz, onde cada elemento

corresponde a um elemento do espectrograma da locução que está sendo analisada. Esta matriz é essencial para os próximos passos do sistema de reconhecimento automático.

4.3 Técnicas de reconhecimento com dados faltantes

Após a definição dos dados confiáveis através da máscara, como descrito no item 4.2 anteriormente, tem-se duas principais técnicas de reconhecimento com dados faltantes: a marginalização e a imputação [31].

Na técnica de marginalização, o reconhecimento é feito somente com os dados confiáveis, descartando os restantes. Essa abordagem requer modificação no mecanismo de reconhecimento, a fim de calcular a pontuação utilizando somente uma parte dos vetores de entrada.

A grande desvantagem do método de marginalização é não permitir a utilização do domínio cepstral no reconhecimento, pois este combina as informações de ambos componentes confiáveis e não confiáveis necessitando assim do vetor completo da locução para o reconhecimento. Outra desvantagem deste método é não permitir a utilização da média e normalização da variância para melhorar o reconhecimento, pois os espectrogramas possuirão dados incompletos.

Por outro lado, na técnica de imputação os valores faltantes são estimados a partir dos dados confiáveis permitindo que o reconhecimento seja realizado com uma representação espectro-temporal completa, portanto nenhuma modificação no mecanismo de reconhecimento será necessária. Esta será a técnica utilizada nos experimentos deste trabalho e será detalhada nas seções a seguir.

4.4 Imputação

A técnica de imputação dos dados faltantes permite a utilização do espectrograma completo no reconhecimento, pois os valores dos componentes não confiáveis são estimados baseados nos seus relacionamentos estatísticos com os componentes

confiáveis. Neste caso o reconhecedor não necessita ser modificado e os vetores cepstrais podem ser derivados, pois a representação tempo-frequência estará completa.

Como mencionado anteriormente, as probabilidade dos símbolos de saída dos estados são modeladas como mistura de Gaussianas. Para um vetor X(t), com componentes confiáveis desconhecidos $X_r(t)$ e componentes não confiáveis desconhecidos $X_u(t)$ a probabilidade de saída do estado s, pode ser escrita como

$$P(X(t)|s) = P(X_r(t), X_u(t)|s) = \sum_i c_{j,s} G(X_r(t), X_u(t): \mu_{j,s}, \theta_{j,s}) \ , \qquad (4.10)$$

onde $G(X_r(t), X_u(t): \mu_{j,s}, \theta_{j,s})$ é a Gaussiana com índice j da mistura de gaussianas da densidade para s, com média $\mu_{j,s}$ e matriz de covariância $\theta_{j,s}$, e $c_{j,s}$ representa o peso da mistura da Gaussiana índice j. Para calcular a probabilidade de saída do vetor real X(t) tem-se

$$P(X(t)|s) = P\left(Y_r(t), \hat{X}_u^s(t)\middle|s\right) = \sum_i c_{j,s} G(Y_r(t), \hat{X}_u^s(t); \mu_{j,s}, \theta_{j,s}) \ , \qquad (4.11)$$

onde $\hat{X}_u^s$ é o valor estimado da MMSE(*Minimum Mean Square Error*) de $X_u(t)$, que é calculada através da equação

$$\hat{X}_u^s(t) = \sum_j \gamma_{j,s}(Y(t)U(t)\mu_{j,s} \ , \qquad (4.12)$$

onde U(t) é a matriz de permutação que forma $Y_u(t)$ e conforme [32]

$$\gamma_{j,s}\big(Y(t)\big) = \frac{c_{j,s} \int_{-\infty}^{Y_u(t)} G\big(Y_r(t), X_u(t); \mu_{j,s}, \theta_{j,s}\big) dX_u(t)}{\sum_k c_{k,s} \int_{-\infty}^{Y_u(t)} G\big(Y_r(t), X_u(t); \mu_{k,s}, \theta_{k,s}\big) dX_u(t)} \ . \qquad (4.13)$$

Na técnica de imputação é necessário reconstruir todos os dados faltantes do espectrograma para utilizar os vetores cepstrais para a realização do reconhecimento. Como demonstrado por Bhiksha Raj Ramakrishnan em [33], o reconhecimento com cepstra tem uma maior exatidão do que o reconhecimento utilizando somente os vetores espectrais.

Tabela 4.1 *Comparação na precisão do reconhecimento* **[33]**.

Precisão no reconhecimento com vetores log espectral	Precisão no reconhecimento com vetores cepstrais
63%	82%

Assim, pode-se resumir no diagrama em blocos abaixo os passos do reconhecimento utilizando a imputação dos dados faltantes com reconstrução do espectrograma.

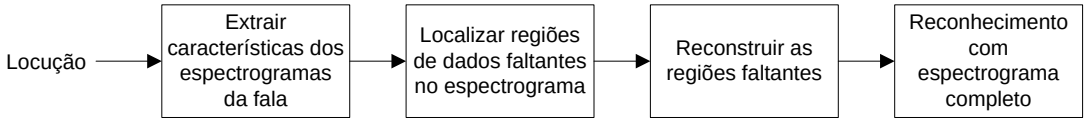

Figura 4.1 Diagrama em blocos da técnica de imputação.

Existem alguns métodos para a reconstrução dos dados faltantes na imputação, os mais utilizados são interpolação, correlação e cluster [34] [35]. O método de interpolação utiliza os componentes confiáveis mais próximos dos componentes não confiáveis para reconstrução dos dados faltantes. A correlação utiliza as dependências estatísticas entre os componentes do quadro atual e dos quadros vizinhos. No método de cluster os dados são modelados como misturas de Gaussianas e os dados faltantes são calculados através das relações estatísticas dentro de um quadro.

Neste trabalho o foco será no desempenho dos diferentes métodos da interpolação e seus resultados. Uma explanação sobre as diferentes abordagens será fornecida a seguir.

4.5 Reconstrução do espectrograma

Existem diversas maneiras de reconstruir um espectrograma após a separação dos elementos em confiáveis e não confiáveis. Uma das abordagens mais utilizadas devido à simplicidade e considerável exatidão nos reconhecimentos são os métodos de reconstrução geométrica, cujo principal técnica é a interpolação, que permite estimar

valores de uma função com base em algumas amostras previamente conhecidas, considerando que a função que modela estes dados é suficientemente suave. Existem três principais tipos de interpolações: Linear, Polinomial e Racional que são apresentadas em detalhes nas seções a seguir.

Como o espectrograma é bidimensional, os elementos podem ser adjacentes e ter continuidade tanto no eixo da frequência como no eixo do tempo, o que nos permitiria realizar a interpolação tanto na frequência quanto no tempo.

Conforme [33] a interpolação ao longo do tempo é geralmente mais efetiva do que a interpolação ao longo da frequência, devido ao espectrograma apresentar uma maior continuidade ao longo do tempo. Assim, para as comparações neste artigo consideraremos somente as interpolações no eixo do tempo. Essa interpolação é realizada depois do cálculo da FFT, no processo de extração dos coeficientes mel cepstrais. O processo de extração dos coeficientes mel está descrito no item 3.3 deste trabalho.

4.5.1 Interpolação Linear

A interpolação linear utiliza um polinômio de primeiro grau para representar uma função descontínua em um determinado intervalo.

Considere $P_1(x)$ o polinômio de grau um que passa pelos pontos $A=(x_i,f_i)$ e $B=(x_{i+1},f_{i+1})$. Assim, para cálculo da interpolação linear utilizamos a equação a seguir

$$z(\beta) \approx P_1(\beta) = z_i + (\beta - x_i)\frac{(z_{i+1}-z_i)}{(x_{i+1}-x_i)}, \tag{4.14}$$

onde:
β: é o ponto onde se deseja calcular o valor da função.
x_i: é um ponto no qual o valor da função é conhecido.
$x_{(i+1)}$: é outro ponto no qual o valor da função é conhecido.
$z_{(i+1)}$: é o valor da função no ponto $x_{(i+1)}$
z_i: é o valor da função no ponto x_i

4.5.2 Interpolação Polinomial – Fórmula de Lagrange

Sejam $x_0,x_1,...x_n$, (n+1) pontos distintos e $y_i=z(x_i)$, i=0,...n. Seja $p_n(x)$ o polinômio de grau menor igual a n que interpola z em $x_0,...x_n$.

De maneira compacta pode-se escrever a forma de Lagrange para o polinômio interpolador como

$$p_n(x) = \sum_{k=0}^{n} z(x_k)L_k(x)\,, \tag{4.15}$$

onde $L_k(x)$ são os fatores de Lagrange e são dados por

$$L_k(x) = \frac{\prod_{\substack{j=0 \\ j\neq k}}^{n}(x-x_j)}{\prod_{\substack{j=0 \\ j\neq k}}^{n}(x_k-x_j)}\,. \tag{4.16}$$

4.5.3 Interpolação Racional – Algoritmo de Bulirsch-Stoer

É um método de extrapolação que utiliza funções racionais para aproximar os pontos da solução de equação diferencial ordinária dentro de um determinado intervalo [36].

Considere a integração numérica de y´=F(x,y), apresentada na Figura 4.2 [37]:

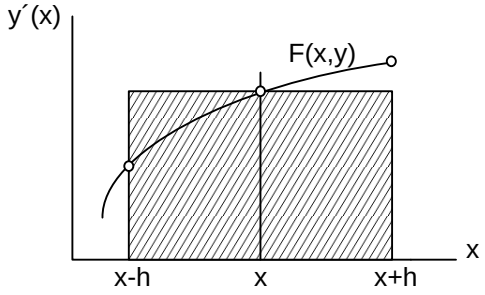

Figura 4.2 Integração Numérica.

A fórmula do ponto médio dessa integração pode ser descrita como

$$y(x+h) = y(x-h) + 2hF[x, y(x)]. \tag{4.17}$$

Dividindo o intervalo de integração(H) em n passos de tamanho h=H/n, pode-se considerar

$$\begin{aligned} y_1 &= y_0 + hF_0 \\ y_2 &= y_0 + 2hF_1 \\ y_3 &= y_1 + 2hF_2 \\ &\vdots \\ y_n &= y_{n-2} + 2hF_{n-1}, \end{aligned} \tag{4.18}$$

onde $y_i = y(x_i)$ e $F_i = F(x_i, y_i)$.

A primeira equação de (4.18) utiliza a fórmula de Euler para sustentar o método do ponto médio. As demais fórmulas são próprias do método do ponto médio.

O resultado final obtido pela média das equações (4.17) está apresentado na equação

$$y(x_0 + H) = \frac{1}{2}\left[\left(y_n + (y_{n-1} + hF_n)\right)\right]. \tag{4.19}$$

Para finalizar o processo é aplicada a extrapolação de Richardson [38] [39]. Mais detalhes sobre este método podem ser encontrados em [40] e [41].

Capítulo 5

Experimentos

5.1 Base de dados

Para os testes, a base de dados escolhida foi a TIDIGITS. Este banco de dados foi criado e coletado pela Texas Instruments em 1982, com o intuito de auxiliar na criação e avaliação de algoritmos de reconhecimento de fala independente do locutor [20].

A base é composta por locuções de dígitos, tanto isolados quanto conectados, criados por 326 locutores: 111 homens, 114 mulheres, 50 garotos e 51 garotas. As gravações estão na língua inglesa e os locutores são provenientes de 21 regiões diferentes dos Estados Unidos da América.

Para este trabalho os locutores foram divididos em dois grupos: treinamento e testes. Neste trabalho apenas as locuções com dígitos isolados serão utilizadas, nas quais cada locutor pronuncia 11 dígitos: *”zero”, ”oh”, ”one”, ”two”, ”three”, ”four”, ”five”, ”six”, ”seven”, ”eight”* e *”nine”* repetidas três vezes cada.

Os dados foram coletados em um ambiente de baixo ruído e digitalizados em 20kHz, com 16 bits de resolução. Apenas os locutores adultos foram utilizados nos experimentos, e o grupo de treinamento é composto por 57 mulheres e 55 homens enquanto o grupo de testes é composto por 57 mulheres e 56 homens. Os locutores dos testes são diferentes dos locutores do treinamento.

As locuções da base de dados são fornecidas em formato raw, ou seja, arquivos sem cabeçalhos e com dados crus. Para que as locuções possam ser utilizadas no HTK (*Hidden Markov Model Toolkit)* que é a ferramenta utilizada no processo de reconhecimento que será detalhado no item 5.3, será necessário primeiramente transformar a base de dados em arquivos wav e reamostrá-la em 8kHz.

Essa conversão foi realizada através de um comando da plataforma Linux "*Sound eXchange*" (sox) [42]. Para os experimentos realizados neste trabalho os arquivos raw foram transformados em wav e reamostrados para a frequência de 8kHz com um canal utilizando o comando

sox -x -c 1 -r 8000 -s -2 xxxxx.raw xxxxx.wav.

Como características acústicas, os coeficientes mel-cepstral, juntamente com a sua primeira e segunda derivada foram escolhidos. Estes coeficientes foram calculados a partir de janelas de 25ms, atualizadas a cada 10ms. Antes da parametrização, as locuções passam por um filtro de pré-ênfase de primeira ordem com a resposta do sistema de $1 - 0.97z^{-1}$.

Na Figura 5.1 é apresentada a sequência de passos resumida da transformação de uma locução em espectrograma mel.

Primeiramente os sinais de áudio passam pela transformada rápida de Fourier, seguido da utilização dos filtros Mel, onde tem-se o espectro mel cujos componentes serão utilizados no processo de reconhecimento através da análise espectral.

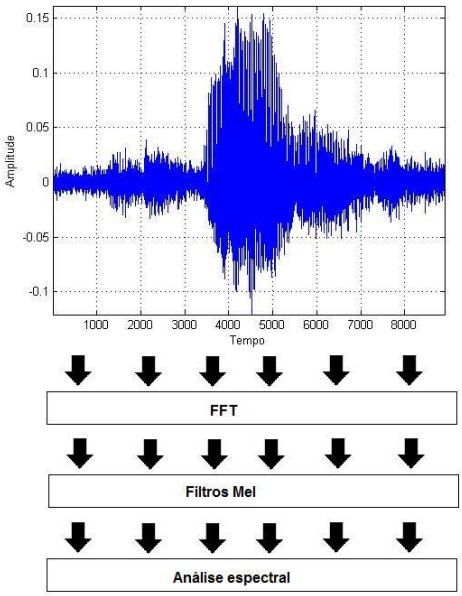

Figura 5.1 Transformando locuções em espectrogramas mel.

5.3 Mecanismo de reconhecimento

O HTK (Hidden Markov Model) toolkit [43] foi escolhido como o mecanismo de reconhecimento. Ele é uma ferramenta livre utilizada para criar e manipular modelos ocultos de Markov, originalmente desenvolvida pelo departamento engenharia da universidade de Cambridge (Cambridge University Engineering Department - CUED).

Esta ferramenta é amplamente utilizada no reconhecimento de voz, mas também é utilizada em inúmeras outras aplicações incluindo pesquisas em síntese de voz, reconhecimento de caracteres e sequenciamento de DNA [43].

O HTK possui dois principais processos: o treinamento onde se estima os parâmetros da HMM e a etapa de reconhecimento.

As ferramentas do HTK são utilizadas através de linhas de comando onde é possível atingir diversas configurações dependendo dos argumentos utilizados, assim como também existem arquivos de configuração para alguns comandos específicos.

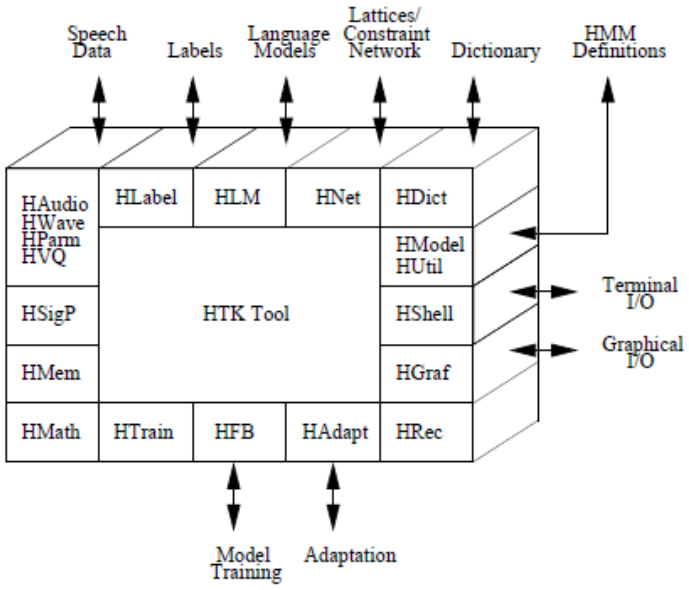

Figura 5.2 Arquitetura da ferramenta HTK [44].

Na Figura 5.2 pode-se verificar a arquitetura utilizada pelo HTK que é apresentada com detalhes no manual da ferramenta HTK Book [44].

A entrada e saída das locuções em nível de *wav* são realizadas através da função HWave e os parâmetros através da função HParm. A função HUtil provê uma grande quantidade de rotinas para manipulação das HMM´s enquanto o HTrain e HFB auxiliam na parte do treinamento do sistema. Por fim, HRec engloba as principais funções do reconhecimento.

Na Figura 5.3 é possível verificar as quatro principais fases do reconhecimento: preparação dos dados, treinamento, teste e análise, e as principais funções utilizadas em cada estágio.

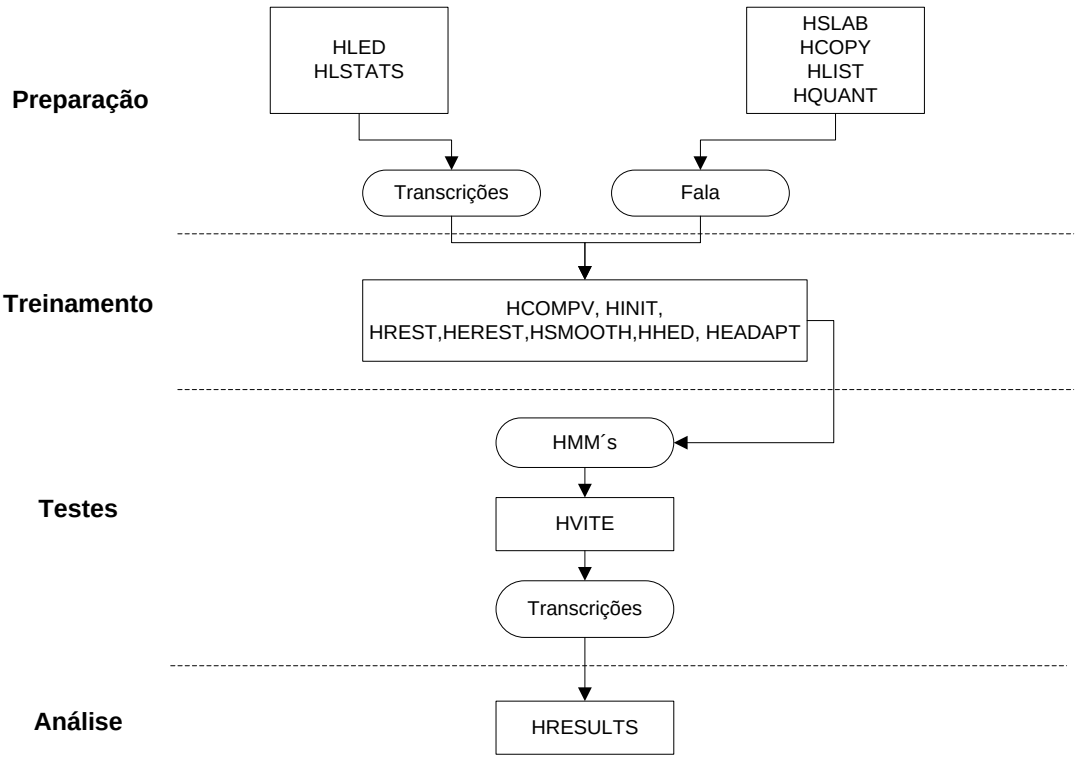

Figura 5.3 Estágios do processamento da ferramenta HTK.

Na ferramenta HTK é necessário especificar as principais características da topologia do HMM que será utilizado. O processo de treinamento da ferramenta está descrito na Figura 5.4.

Os arquivos das locuções parametrizadas utilizados tanto para treinamento quanto reconhecimento devem estar no formato específico requisitado pelo toolkit HTK. Formato este, que consiste em uma sequência de amostras precedidas de um cabeçalho.

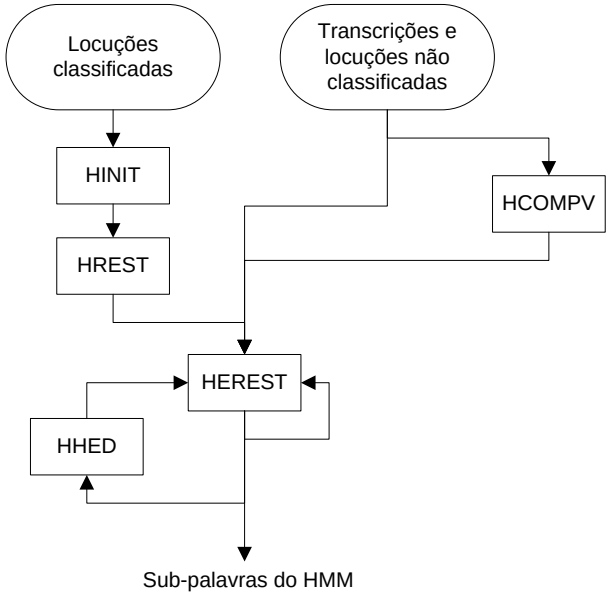

Figura 5.4 Sistema de treinamento da ferramenta HTK.

O cabeçalho do formato HTK possui 12 bytes e deve conter os seguintes itens:

- Número de amostras do arquivo (4 bytes)
- Período da amostra em unidade de 100ns (4 bytes)
- Número de bytes por amostra (2 bytes)
- Código indicativo de tipo de amostras (2 bytes)

Neste trabalho, com as locuções provenientes da base de dados TIDIGITS o cabeçalho indicará período da amostra de 100ms, tipo de dados coeficientes mel cepstrais (MFCC) incluindo o coeficiente 0.

5.4 Processamento dos dados

O Matlab [45] é uma ferramenta de linguagem alto nível e ambiente interativo para computação numérica, visualizações e programação. Neste trabalho, o Matlab será utilizado para o processamento dos dados dos espectrogramas, permitindo a imputação e

a reconstrução do mesmo das diversas maneiras apresentadas nos capítulos anteriores, além da extração dos MFCC e do espectrograma mel.

Para inserir as locuções no HTK é necessário utilizar formato esperado pela ferramenta, que deve conter em seus arquivos o cabeçalho de 12 bytes descrito no item 5.3.

Para transformar as locuções neste formato utiliza-se o VOICEBOX [46], que é uma ferramenta de processamento de voz para o Matlab. Esta ferramenta foi desenvolvida por Mike Brookes do departamento de engenharia elétrica e eletrônica do colégio imperial de Londres. Essa ferramenta é livre e segue os termos da licença livre do GNU.

Do VOICEBOX foram utilizadas as funções readhtk e writehtk que permitem a leitura e escrita dos arquivos de sinais de áudio e demais parâmetros utilizados pelo HTK.

Para transformar o sinal de áudio em espectrograma foi utilizada uma ferramenta para Matlab chamada RastaMat [47], desenvolvida pelo professor de engenharia elétrica da Universidade de Columbia, Dan Ellis.

Da ferramenta Rasta a principal função utilizada é a melfcc que nos permite ter acesso as matrizes que correspondem aos espectros de potência dos sinais e calcula os MFCC´s de um sinal wav.

Com a utilização das ferramentas citadas acima foi possível desenvolver diversas outras funções no Matlab que manipulam os dados, criando a máscara de dados faltantes, realizando a imputação e a reconstrução dos espectrogramas e transformando-os novamente no formato HTK para realização do reconhecimento.

5.5 Criando a máscara de dados faltantes

Para a criação da máscara de dados faltantes é necessário ter como entrada a matriz espectral do sinal de áudio, lida pela função readhtk e os espectros lineares dados pela FFT através da função melfcc.

Com a matriz cujos elementos representam o espectro de uma locução, a máscara de dados faltantes pode ser criada de uma das seguintes maneiras:

- A partir de um vetor ruído que pode ser adquirido através da média dos 10 primeiros quadros, como descrito no item 4.2 deste trabalho. Neste caso para detectar os elementos que serão descartados na imputação pode-se escolher entre os critérios de energia negativa ou SNR.
- Para fins de testes das técnicas de interpolação, pode-se criar a máscara de forma aleatória, escolhendo os elementos da locução que serão considerados corrompidos pelo ruído através de uma função Bernoulli, variando de 10% a 80% a quantidade de elementos que serão descartados na imputação.

No fluxograma ilustrado na Figura 5.5 pode-se acompanhar a sequência e as possibilidades de escolha na geração da máscara de dados na plataforma desenvolvida.

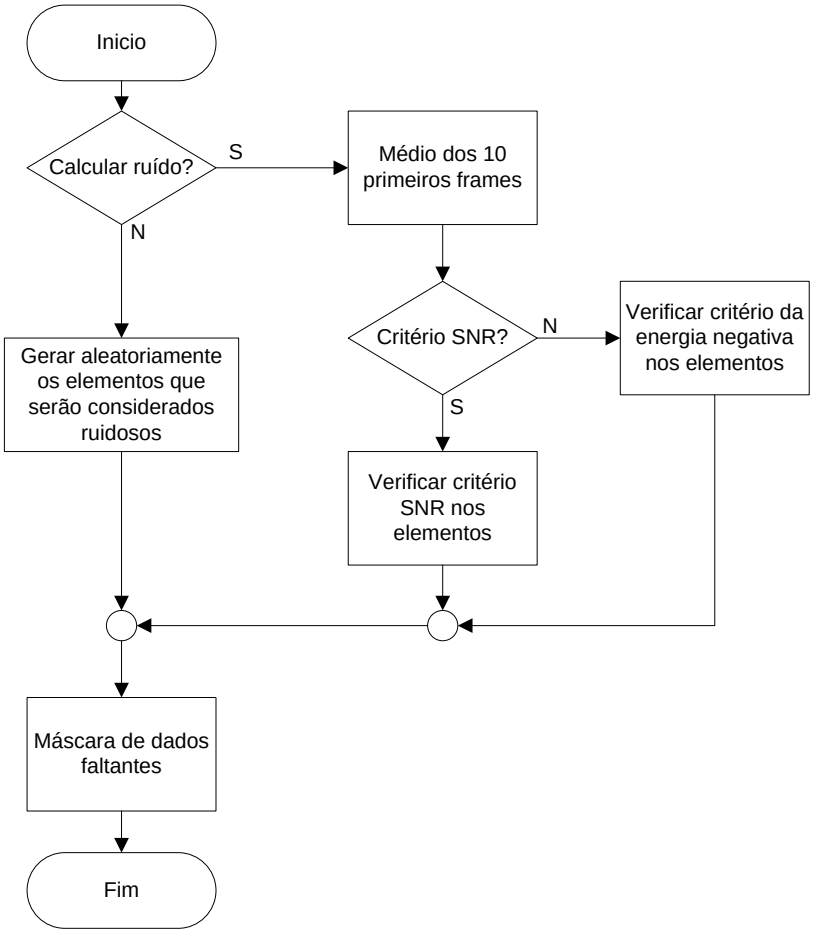

Figura 5.5 Fluxograma da criação da máscara de dados.

A função *createMask* foi desenvolvida para criação da máscara de dados.

5.6 Realizando a imputação

Após a realização da FFT nas locuções, com o espectro de potência e a máscara de dados faltantes, pode-se realizar a imputação escolhendo a técnica a ser utilizada:

- Interpolação simples no tempo
- Interpolação racional no tempo
- Interpolação polinomial no tempo
- Média simples no tempo

- Média simples na frequência

Um resumo das etapas de imputação e reconstrução está representada no fluxograma da Figura 5.6.

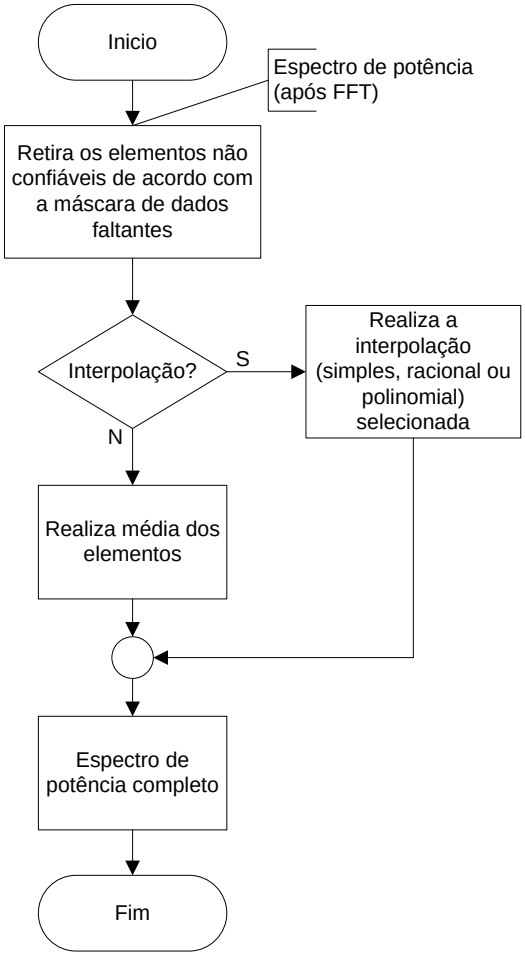

Figura 5.6 Fluxograma da interpolação na reconstrução dos dados.

A função desenvolvida no Matlab foi a *makeImputation*.

Após a imputação é realizada a extração dos coeficientes mel cepstrais baseados na representação completa do espectro de potência da locução.

5.7 Plataforma de imputação de dados faltantes

Para realizar o principal objetivo deste trabalho que é a comparação do desempenho dos métodos de interpolação na reconstrução do espectrograma foi criada uma ferramenta para a análise e criação de diversas maneiras de se realizar a imputação de dados faltantes. Assim, foi desenvolvida no Matlab uma plataforma onde estão desenvolvidas diferentes formas de criar a máscara de dados, realizar a imputação e retornar as locuções para o formato HTK possibilitando o reconhecimento das mesmas.

Na Figura 5.9 é apresentado o sistema de reconhecimento do qual a plataforma desenvolvida faz parte. Como dados de entrada da plataforma é necessário criar uma pasta contendo todos os arquivos wav que passarão pela imputação, e um arquivo txt que lista o caminho desses arquivos.

Na plataforma é necessário configurar o modo de geração da máscara e o método de reconstrução após a imputação. Como saída deste processamento tem-se a criação de um arquivo mfc para cada arquivo wav inicial, contendo em cada um destes arquivos a locução imputada e reconstruída pronta para o sistema de reconhecimento, ou seja, no formato requerido pelo HTK.

Por fim, na ferramenta HTK com o sistema já treinado deve-se organizar os arquivos de configuração e efetuar o reconhecimento e análise dos resultados.

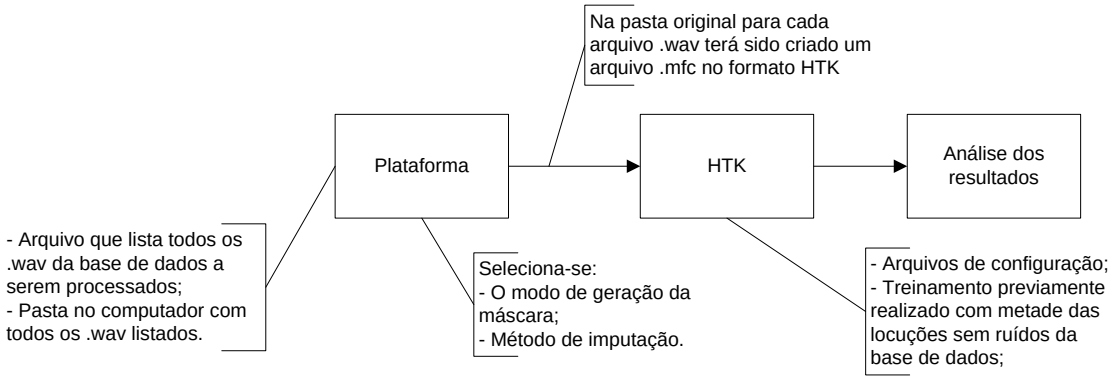

Figura 5.7 Diagrama em blocos do sistema.

A plataforma desenvolvida tem como blocos principais: o cálculo ou a geração aleatória do vetor ruído; a criação da máscara de dados; a imputação dos dados indicado pela máscara; a reconstrução do espectrograma; a extração dos coeficientes MFCC das locuções provenientes do banco de dados TIDIGITS; e a criação dos novos arquivos com os dados reconstruídos que serão utilizados no reconhecimento.

O resumo dos blocos principais está apresentado na Figura 5.10.

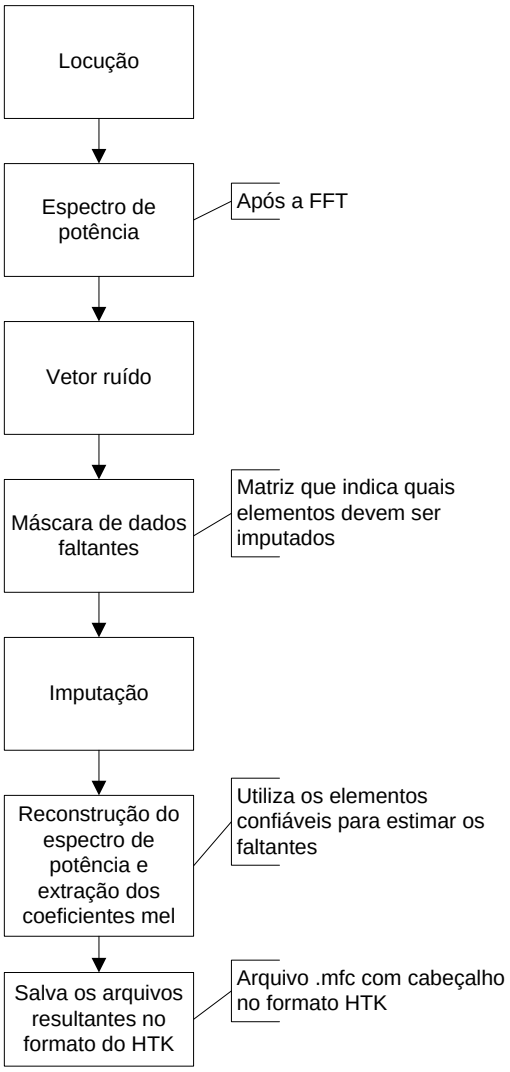

Figura 5.8 Diagrama em blocos da plataforma desenvolvida.

Na Figura 5.11 é apresentada a GUI (*Graphical User Interface*) da plataforma desenvolvida, onde é possível verificar as possibilidades de escolha na criação da máscara de dados, as opções de reconstrução dos dados assim como a seleção do arquivo de entrada com as listas das locuções wav que passarão pelo sistema.

Como saída deste processamento temos os arquivos mfc com as locuções reconstruídas com o cabeçalho adequado para o reconhecimento da ferramenta HTK.

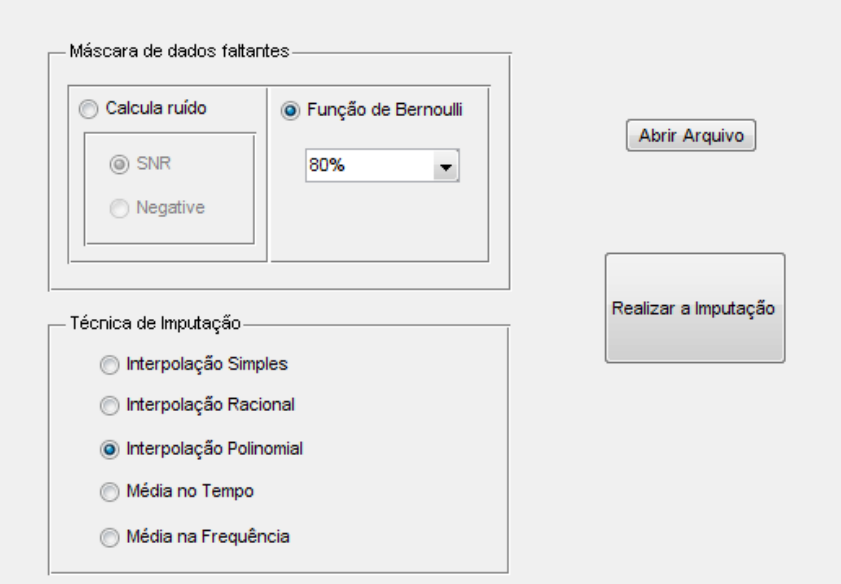

Figura 5.9 Interface da plataforma desenvolvida.

Capítulo 6

Resultados e discussões

6.1 Desempenho base

O primeiro experimento é o reconhecimento da base de dados de teste das locuções sem ruído que estabelece o limite teórico para a taxa de acertos para reconhecimento das locuções ruidosas.

Assim, foi atingida uma taxa de acertos de 99,39% no reconhecimento das locuções sem ruído que representa o limitante superior para este trabalho.

6.2 Experimento

Para realizar a comparação entre as técnicas de reconstrução de dados faltantes propostas neste trabalho foi utilizada a plataforma desenvolvida, onde as locuções da base de dados TIDIGITS separadas para o teste passaram pelo processo de imputação no Matlab antes de serem inseridas na ferramenta HTK para o reconhecimento.

Como o foco deste trabalho não é a criação da máscara de dados faltantes, mas é a reconstrução das locuções e sua precisão, vamos supor que os dados que serão imputados são conhecidos, ou seja, a máscara tem um desempenho ideal.

Interpolações foram feitas no eixo do tempo onde os experimentos mostram resultados mais significativos [25]. Como um exemplo, a Figura 6.1 representa um espectrograma de potência de uma locução sem ruído da base de dados TIDIGITS.

A Figura 6.2 mostra os pontos do espectrograma que foram imputados, e Figura 6.3 é a representação dessa mesma locução reconstruída com interpolação linear.

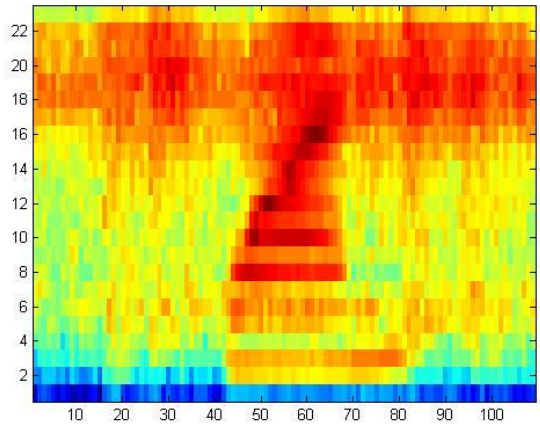

Figura 6.1 Espectrograma de uma locução sem ruído.

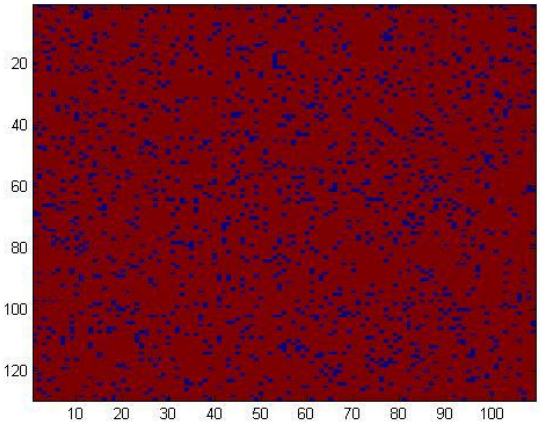

Figura 6.2 Máscara de dados faltantes.

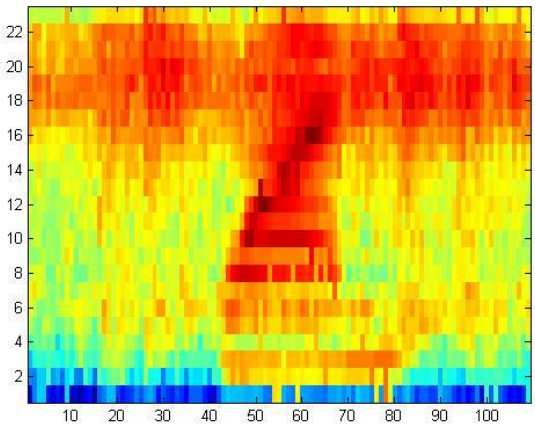

Figura 6.3 Espectrograma reconstruído.

A comparação entre a Figura 6.1 e Figura 6.3 mostra que é possível ter uma reconstrução razoável do espectrograma da fala mesmo com a maioria da informação original faltante.

6.4 Procedimentos e resultados

Nesta seção será apresentado o procedimento para a obtenção dos diferentes tipos de interpolação empregados na reconstrução das locuções, e os resultados obtidos com os mesmos.

Os elementos faltantes foram escolhidos randomicamente, seguindo uma distribuição de Bernoulli com probabilidade de ocorrência variando de 10% a 80% em passos de 10%.

Após a realização dos testes propostos com os três tipos diferentes de interpolações: simples, racional e polinomial na reconstrução do espectro da locução, foram obtidos os resultados apresentados na Figura 6.4.

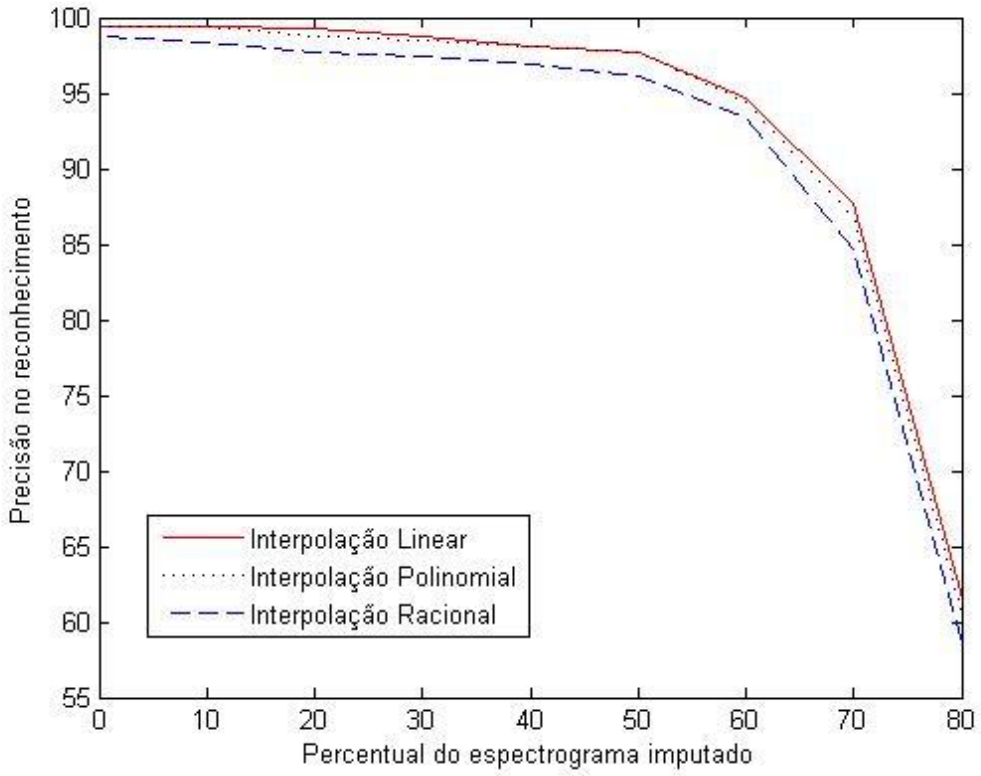

Figura 6.4 Resultados das interpolações propostas.

Analisando os resultados verificou-se que os três métodos apresentaram resultados similares, com uma pequena vantagem para a interpolação linear. Também é possível notar que os métodos de interpolação trabalham bem mesmo com pouca quantidade de informação original disponível: os resultados permaneceram estáveis até 50%, 60% de dados imputados, caindo drasticamente somente após a região de 70%.

Na Tabela 6.1 são apresentados os resultados acima de forma mais detalhada. A coluna nomeada Imp0% representa o reconhecimento das locuções sem nenhum ruído e as demais colunas apresentam o resultado do reconhecimento realizado após as imputações de 10% a 80%, reconstruindo o espectrograma com interpolação simples, polinomial e racional no tempo.

Tabela 6.1 ***Resultados das interpolações***

Taxa de acertos (%)									
Tipo de interpolação	Imp 0%	Imp 10%	Imp 20%	Imp 30%	Imp 40%	Imp 50%	Imp 60%	Imp 70%	Imp 80%
Linear	99,39	99,39	99,29	98,77	98,13	97,67	94,66	87,72	61,47
Polinomial	99,39	99,39	98,75	98,47	98,13	97,69	94,42	86,72	60,47
Racional	99,39	98,39	97,75	97,47	96,85	96,12	93,42	84,72	58,47

Capítulo 7

Conclusões e oportunidades para trabalhos futuros

O intuito deste trabalho foi comparar os três métodos de imputação para reconstrução de dados faltantes no reconhecimento automático de fala: interpolação linear, polinomial e racional para dígitos isolados independente do locutor e apresentar uma plataforma que facilite a realização da imputação dos dados faltantes.

Os métodos de interpolação apresentaram resultados similares com pequena vantagem para a interpolação linear, que é um ganho em dobro: melhor exatidão combinada com menor complexidade.

As taxas de reconhecimento permaneceram próximas à linha base de taxas de acerto de 99,39% mesmo com aproximadamente 60% dos dados originais imputados.

Para o futuro, o uso de máscara de dados faltantes não ideal, calculada a partir do ruído da locução deve ser testada a fim de verificar a real exatidão desta ideia.

Em suma, as principais contribuições deste trabalho foram:

- Plataforma para criar máscara de dados faltantes a partir do método selecionado: SNR, energia negativa ou aleatória (Bernoulli).

- Plataforma para realizar a imputação dos dados baseado na máscara e reconstrução do espectrograma utilizando os elementos consideráveis confiáveis.

- Comparação dos principais métodos de interpolação geométrica na reconstrução dos dados faltantes.

A plataforma desenvolvida possibilita a realização de diversas outras análises, mas como o objetivo principal deste trabalho foi comparar as técnicas de interpolação, não foram realizadas as diversas combinações disponíveis na plataforma.

Como propostas para trabalhos futuros sugerem-se:

- Análise de outras técnicas de reconstrução do espectrograma com dados faltantes, como reconstrução baseada em cluster e em covariâncias.

- Análise de outras técnicas mais complexas e precisas de criação da máscara de dados faltantes.

- Utilização de outros bancos de dados, que possibilite novos testes e consequentemente a validação dos resultados apresentados.

- Integração da plataforma com outras ferramentas de reconhecimento diferentes do HTK Toolkit.

- Criação de um sistema único que integra a plataforma com o sistema reconhecedor, sem a necessidade de intervenção do usuário.

REFERÊNCIAS BIBLIOGRÁFICAS:

[1] L. R. RABINER and B. H. JUANG, "Automatic Speech Recognition - A Brief History of the Technology," in *Elsevier Encyclopedia of Language and Linguistics*, 2004, pp. 1-24.

[2] R. P. LIPPMANN, "Speech recognition by machines and humans," *Speech Communication*, pp. 1-15, 1997.

[3] M. ANUSYA and S. KATI, "Speech Rognition by Machine: A Review," *International Journal of Computer Science and Information Security*, vol. 6, 2009.

[4] A. R. FUKANE and S. L. SAHARE, "Different Approaches of Spectral Subtraction method for Enhancing the Speech Signal in Noisy Environments," *International Journal of Scientific & Engineering Research*, vol. 2, 2011.

[5] F. LIU, R. M. STERN, X. HUANG and A. ACERO, "Efficient cepstral normalization for robust speech recognition," in *Proceedings of ARPA Speech and Natural Language Workshop*, 1993.

[6] B. RAJ and R. STERN, "Missing-feature approaches in speech recognition," *Signal Processing Magazine*, vol. 22, pp. 101-116, 2005.

[7] A. KLAUTAU, "Desenvolvimento de Aplicativos Usando Síntese e Reconhecimento de Voz," Universidade Federal do Pará, 2009.

[8] J. HIRSCHBERG, "Automatic Speech Recognition, Text-to-Speech and Natural Language Understanding Technologies," Columbia University, 1996.

[9] A. SILVA and T. REN, "Reconhecimento de voz para palavras isoladas," Universidade Federal de Pernambuco, 2009.

[10] R. DIAS, "Normalização de Locutor em Sistema de Reconhecimento de Fala," Universidade Estadual de Campinas, 2000.

[11] M. KESARKAR, "Feature Extraction For Speech Recognition," *M.Tech. Credit Seminar Report*, pp. 1-13, 2003.

[12] B. LOGAN, "Mel Frequency Cepstral Coefficients for Music Modeling," Cambridge Research Laboratory, 2000.

[13] K. PRAHALLAD, "Speech Technology: A Practical Introduction," Carnegie Mellon University &International Institute of Information Technology Hyderabad, 2005.

[14] C. CUADROS, "Reconhecimento de voz e de locutor em ambientes ruidosos: comparação das técnicas MFCC e ZCPA," Universidade Federal Fluminense, 2007.

[15] L. RABINER and B. JUANG, "An Introduction to Hidden Markov Models," Stanford, 1986.

[16] L. OLIVEIRA and M. MORITA, "Introdução aos Modelos Escondidos de Markov (HMM)," Pontifícia Universidade Católica do Paraná, 2010.

[17] F. BRUGNARA and R. D. MORI, "Survey of the State of the Art in Human Language Technology: HMM Methods in Speech Recognition," *Cambridge University Press*, pp. 20-57, 1997.

[18] J. DELLER, J. HANSEN and J. PROAKIS, "Discrete-Time Processing of Speech Signals," *IEEE Press*, p. 936, 2000.

[19] A. VALENTIM, M. CÔRTES and A. GAMA, "Análise espectrográfica da voz: efeito do treinamento visual na confiabilidade da avaliação," *Rev Soc Bras Fonoaudiol.*, vol. 15, pp. 335-342, 2010.

[20] J. PICONE, "TIDIGITS," [Online]. Available: http://www.isip.piconepress.com/projects/speech/software/tutorials/production/fundamentals/v1.0/section 02/s02 04 p01.html. [Accessed Março 2015].

[21] M. N. Vieira, "Uma introdução à acústica da voz cantada," Seminário Música Ciência Tecnologia: Acústica Musical, 2004.

[22] J. C. SEGURA, A. d. l. TORRE, M. C. BENITEZ and A. M. PEINADO, "Model-based compensation of the additive noise for continuous speech recognition.," *Eurospeech,* 2001.

[23] S. G. S. PETTERSEN, "Robust Speech Recognition in the Presence of Additive Noise," Department of Electronics and Telecommunications Norwegian University of Science and Technology, 2008.

[24] D. KOLOSSA and R. HAEB-UMBACH, Robust Speech Recognition of uncertain or missing data - Theory and applications, Springer, 2011.

[25] A. VIZINHO, P. GREEN, M. COOKE and L. JOSIFOVSKI, "Missing Data Theory, Spectral Subtraction And Signal-To-Noise Estimation For Robust Asr: An Integrated Study," Sixth European Conference on Speech Communication and Technology, Budapest, 1999.

[26] S. HSIANG, "Missing-Feature Approaches in Speech Recognition," SLP, 2006.

[27] B. R. RAMAKRISHNAN and M. L. S. R. M. SELTZER, "Reconstruction of missing features for robust speech recognition," *Speech Communication,* vol. 43, p. 275–296, 2004.

[28] H. G. HIRSCH and C. EHRLICHER, "Noise Estimation Techniques for Robust Speech Recognition," *ICASSP,* vol. 1, p. 153–156, 1995.

[29] C. CERISARA, S. DEMANGE and J. HATON, "On noise masking for automatic missing data speech recognition: a survey and discussion," in *Computer Speech and Language*, 2007, pp. 443-457.

[30] A. DRYGAJLO and M. EL-MALIKI, "Speaker Verification in Noisy Environment with Combined Spectral Subtraction and Missing Data Theory," *ICASSP,* vol. 1, pp. 121-124, 1998.

[31] B. RAJ and R. SINGH, in *Robust Speech Recognition of Uncertain or Missing Data - Theory and Applications*, 2011, pp. 127-156.

[32] L. JOSIFOVSKI, M. COOKE, P. GREEN and A. VIZINHO, "State based imputation of missing data for robust speech recognition and speech enhancement," EUROSPEECH, Budapest, Hungary, 1999.

[33] B. RAMAKRISHNAN, "Reconstruction of incomplete spectrograms for robust speech recognition," Department of Electrical and Computer Engineering Carnegie Mellon University, 2000.

[34] J. F. GEMMEKE and U. REMES, "Missing-Data Techniques: Feature Reconstruction in Techniques for Noise Robustness in Automatic Speech Recognition," in *Techniques for Noise Robustness in Automatic Speech Recognition*, 2012, p. Chapter 15.

[35] K. WAGSTAFF, "Clustering with Missing Values: No Imputation Required in Studies in Classification, Data Analysis, and Knowledge Organization," in *Classification, Clustering, and Data Mining Applications*, 2004, pp. 649-658.

[36] T. QUEIROZ and D. SANTEE, "Um aprimoramento no método de extrapolação de Gragg-Bulirsch-Stoer para obter alta precisão e baixo esforço computaional," in *4to Congreso Internacional, 2do Congreso Nacional de Métodos Numéricos en Ingeniería y Ciencias Aplicadas* , México, 2007.

[37] J. KIUSALAAS, "Bulirsch–Stoer Method," in *Numerical Methods in Engineering with Matlab*, Cambridge University Press, 2005, pp. 285-295.

[38] R. BULIRSCH and J. STOER, "Numerical Treatment of Ordinary Differential Equations by Extrapolation Methods," *Numerische Mathematik,* vol. 8, pp. 1-13, 1966.

[39] S. KIRPEKAR, "Implementation of Bulirsch Stöer Extrapolation Method," Department of Mechanical Engineering, UC, Berkeley/California, 2003.

[40] W. GRAGG, "On extrapolation algorithms for ordinary initial value problems," *SIAM J. Num. Anal. 3,* pp. 384-403, 1965.

[41] N. L. SCHYER, "An Extrapolation Step-Size Monitor for Solving Ordinary Differential Equations," in *Proceedings of the 1974 annual conference (ACM/CSC-ER)*, 1974.

[42] "SoX - Sound eXchange," [Online]. Available: http://sox.sourceforge.net. [Accessed Março 2015].

[43] "Hidden Markov Model Toolkit (HTK) Speech Recognition Toolkit.," [Online]. Available: http://htk.eng.cam.ac.uk/ . [Accessed Dezembro 2014].

[44] S. YOUNG, G. EVRMANN, M. GALES, T. HAIN, D. KERSHAW, X. LIU, G. MOORE, J. ODELL, D. OLLASON, D. POVEY, V. VALTCHEV and P. WOODLAND, The HTK Book.

[45] Matlab. [Online]. Available: http://www.mathworks.com/products/matlab/. [Accessed Abril 2015].

[46] "VOICEBOX: Speech Processing Toolbox for MATLAB," [Online]. Available: http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html. [Accessed Dezembro 2014].

[47] "RASTA-PLP," [Online]. Available: http://labrosa.ee.columbia.edu/matlab/rastamat/. [Accessed Abril 2015].

Apêndice

A seguir será apresentado o procedimento utilizado para obter os resultados apresentados neste trabalho.

O primeiro passo do experimento foi preparar a ferramenta de reconhecimento HTK utilizando o conjunto de locuções do banco de dados TIDIGITS destinados para o treinamento.

Para inicio da utilização da ferramenta HTK foi criado um arquivo chamado hcopy.conf que possui as configurações das locuções conforme:

```
#tempo é dado em 100 ns, ou seja, tem que multiplicar por 10^(-7)
USEILDET            =  FALSE
ENORMALISE          =  TRUE
NUMCEPS             =  12
CEPLIFTER           =  22
NUMCHANS            =  24
USEPOWER            =  FALSE
PREEMCOEF           =  0.970000
USEHAMMING          =  TRUE
WINDOWSIZE          = 200000.0 #window of 20 ms
SAVEWITHCRC         = FALSE
SAVECOMPRESSED      = FALSE
TARGETRATE          = 100000.0
TARGETKIND          = MFCC_E_D_A
#TARGETFORMAT       = HTK
ZMEANSOURCE         = TRUE
SOURCEFORMAT        = WAV
SOURCEKIND          = WAVEFORM
SOURCERATE          = 1250 #8 kHz
```

(a) *Arquivo de configuração hcopy.conf.*

Para realização do treinamento, deve-se criar um arquivo texto que liste o endereço da pasta onde locuções separadas para o treinamento se encontram.

Com a lista dos arquivos para o treinamento em formato wav é necessário utilizar o comando HCopy no HTK, para parametrizar os arquivos de acordo com a necessidade

da ferramenta e criar para cada arquivo wav um arquivo mfc. O comando deve ser executado da seguinte forma:

./HCopy -T 0001 -C /home/elaine/arq_tidigits/hcopy.conf –S home/elaine/arq_tidigits/train/hcopy_train.list

onde hcopy_train.list é o arquivo texto que lista as locuções wav do treinamento.

Também é necessário criar outro arquivo de configuração *config.fig* como mostrado a seguir.

```
#Coding parameters
#SOURCEFORMAT=WAV
TARGETKIND=MFCC_E_D_A
TARGETRATE=100000.0
#10ms
SAVECOMPRESSED=F
SAVEWITHCRC=F
WINDOWSIZE=200000.0
#25ms
USEHAMMING=T
PREEMCOEF=0.97
NUMCHANS=24
CEPLIFTER=22
NUMCEPS=12
ENORMALISE=T
#NATURALREADORDER=TRUE
#NATURALWRITEORDER=TRUE
SAVEBINARY=FALSE
```

(b) *Arquivo de configuração config.fig.*

Este arquivo de configuração, assim como os arquivos mfc gerados anteriormente serão utilizados no comando HCompV, que é responsável por calcular a média global e a covariância do conjunto de dados do treinamento.

Para concluir o treinamento também são utilizados os comandos HERest e HHEd. O comando HERest realiza a reestimação dos parâmetros da HMM gerada no treinamento e o comando HHEd manipula as definições da HMM.

Após a conclusão do treinamento, com a HMM concluída, podemos realizar os testes com as locuções que passaram pela plataforma de processamento do Matlab e já estão

no formato desejado pelo HTK, ou seja, iremos utilizar os arquivos mfc que foram as saídas da plataforma.

Ao processar as locuções é necessário utilizar o comando HVite, responsável por realizar o reconhecimento através de Viterbi. E para finalizar o teste utiliza-se o comando HResults, que realiza uma análise dos resultados obtidos pelo comando HVite.

O arquivo de saída do comando HResults apresenta as estatísticas conforme:

```
---------------------------- Overall Results -------------------
SENT:  %Correct=13.00 [H=13, S=87, N=100]
WORD:  %Corr=53.36, Acc=44.90 [H=460,D=49,S=353,I=73,N=862]
================================================================
```

(c) *Análise do resultado do reconhecimento*

A primeira linha apresenta o resultado do reconhecimento de uma frase ou sentença, enquanto a segunda linha apresenta os resultados do reconhecimento de palavras isoladas.

Na segunda linha H representa o número de palavras reconhecidas corretamente, D o número de palavras que não foram reconhecidas porque foram apagadas, S o número de substituições e N o número total de palavras.

O número de palavras reconhecidas corretamente é dado por:

$$\%Correta = \frac{H}{N} * 100\% \quad \text{(A.1)}$$

A precisão do sistema é calculada pela ferramenta através da equação 6.2.

$$Precisão = \frac{H-I}{N} * 100\% \quad \text{(A.2)}$$

A seguir pode-se averiguar um dos resultados obtidos através do comando HResults durante o desenvolvimento deste trabalho.

```
Subway   condition: clean1
====================== HTK Results Analysis =======================
  Date: Fri Dec 5 22:53:20 2000
  Ref : /pkg/recdata/aurora/recognizer/labels/N1.mlf
  Rec : /tmp/test_htk/result_clean.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=96.50 [H=966, S=35, N=1001]
WORD: %Corr=99.39, Acc=98.83 [H=3237, D=8, S=12, I=18, N=3257]
===================================================================
```

(d) *Resultado obtido em uma dos testes deste trabalho*