Wellington Lopes de Freitas

Desenvolvimento de Sistema de Reconhecimento de Fala em Plataforma Embarcada

Alegrete, RS

9 de julho de 2019

Wellington Lopes de Freitas

Desenvolvimento de Sistema de Reconhecimento de Fala em Plataforma Embarcada

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Telecomunicações, Área de Concentração em Sinais e Sistemas, da Universidade Federal do Pampa (Unipampa, RS), como requisito parcial para a obtenção do grau de **Bacharel de Engenharia de Telecomunicações**.

Universidade Federal do Pampa – Unipampa

Curso de Engenharia de Telecomunicações

Orientador: Prof. Me. Fabiano Tondello Castoldi

Alegrete, RS

9 de julho de 2019

Ficha catalográfica elaborada automaticamente com os dados fornecidos
pelo(a) autor(a) através do Módulo de Biblioteca do
Sistema GURI (Gestão Unificada de Recursos Institucionais) .

F866d Freitas, Wellington Lopes de
Desenvolvimento de sistema de reconhecimento de fala em plataforma embarcada. / Wellington Lopes de Freitas.
81 p.

Trabalho de Conclusão de Curso(Graduação)-- Universidade Federal do Pampa, ENGENHARIA DE TELECOMUNICAÇÕES, 2019.
"Orientação: Fabiano Tondello Castoldi".

1. Processamento digital de sinais. 2. Reconhecimento Automático de fala. 3. Mel Frequency Cepstral Coefficients. 4. Dynamic Time Warping. I. Título.

WELLINGTON LOPES DE FREITAS

DESENVOLVIMENTO DE SISTEMA DE RECONHECIMENTO DE FALA EM PLATAFORMA EMBARCADA

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Telecomunicações da Universidade Federal do Pampa, como requisito parcial para obtenção do título de Bacharel em Engenharia de Telecomunicações.

Área de Concentração: Sinais e Sistemas

Trabalho de Conclusão de Curso defendido e aprovado em: 26 de Junho de 2019.

Banca examinadora:

Fabiano Tondello Castoldi

Prof. Me. Fabiano Tondello Castoldi
Orientador

Crístian Müller

Prof. Dr. Crístian Müller
UNIPAMPA

Eliezer S. Flores

Prof. Me. Eliezer Soares Flores
UNIPAMPA

Dedico este trabalho à minha família, por todo amor, carinho,
apoio e sacrifícios a mim dedicados, principalmente durante a minha graduação.

Agradecimentos

Inicialmente, agradeço a minha família. Aos meus pais, Silvani e Vilma, por todo amor e empenho dedicados durante a minha criação, e por todos os sacrifícios realizados para poder me dar a oportunidade de estudar. Agradeço também a minha irmã, Jéssica, pelo companheirismo e as palavras de apoio durante os momentos difíceis. Agradeço aos meus sobrinhos, Vinícius e Caio, a quem tenho como meus filhos, por todo aprendizado e momentos de descontração. Agradeço também ao Jair, a quem considero um membro da família. Agradeço ainda aos meus avós, tios e primos, que mesmo longe tentaram se fazer presentes.

A minha namorada, Tiane do Nascimento Vargas, pelo amor, companheirismo, apoio, por todos os conselhos e por me dar um ombro para desabafar durante esses anos. Agradeço também a sua família, em especial ao meu sogro e minha sogra, Jorge e Janete, por terem me acolhido como um filho.

A Bruno Felice, Matheus Cortez, Mateus Oliveira e Marcelle Alves, pela amizade criada nesses anos de graduação. Agradeço a Daniel Baú, Gabriel Cocco, Cleiton Lucatel e aos demais colegas de de laboratório do GPSEL, mais conhecida como eterna 115, pelos momentos de aprendizado e pela descontração durante o mate e o café. Agradeço ainda ao pessoal do LEMA, principalmente ao Juner Menezes e ao Diego Fumagalli, por sempre estarem por perto para ajudar.

Agradeço aos meus demais colegas de graduação que de qualquer forma participaram deste período, principalmente a Geovana Araújo, Arielly Rodrigues e ao Fabrício Trindade. Agradeço ainda ao Frederico Goulart e ao Bruno Muswieck, por me dar uma oportunidade de estágio, onde muito aprendi.

Ao professor Fabiano Castoldi por ter me ajudado durante a produção deste trabalho e pelos conselhos dados. Agradeço ainda a todos os outros professores que de uma maneira ou outra participaram da minha formação.

Aos meus amigos de longa data: Evandro Lima, Evandro Santana e Thiago Lima. Agradeço a galera do PT: Victor e Fernanda Mendes Pereira, Rômulo Silva; Marco, Rafael e Daniel Teixeira, Ricardo Balista, Carlos Nascimento e Luana Russo, que apesar da distância, sempre estiveram por perto. Por fim, agradeço a todas as pessoas que de uma forma ou outra participaram positivamente deste período da minha vida. Eu queria que soubessem que cheguei até aqui me apoiando em vocês. Portanto a TODOS, meu muito obrigado!

"It matters not how strait is the gate,
How charged with punishments the scroll,
I am the master of my fate,
I am the captain of my soul."

Invictus - William Ernest Henley

Resumo

Este trabalho apresenta o projeto de um sistema de reconhecimento de fala para palavras isoladas e dependente de locutor, implementado em um sistema computacional de baixo custo utilizando linguagem de programação interpretada de propósito geral Python®. No desenvolvimento desse sistema, o bloco de processamento de sinais é implementado utilizando a técnica *Endpoint detection* para a detecção do início e fim de uma elocução, e na fase de extração de características do sinal de fala foi abordada a técnica de extração dos coeficientes cepstrais em escala de frequências Mel (Mel Frequency Cepstral Coefficients). O reconhecimento das palavras foi realizado através da abordagem por comparação de padrões, utilizando a técnica de distorção dinâmica do tempo (Dynamic Time Warping). Para o projeto do sistema, primeiro foram realizados testes com o objetivo de encontrar quais as configurações das etapas de processamento de sinais e reconhecimento produziam os melhores resultados em relação a taxa de acertos de palavras e tempo de execução médio por palavra. Nestes testes, a melhor taxa de acertos obtida para um vocabulário de 25 palavras pré-gravadas, foi de 96,8% com tempo médio de execução por palavra inferior a 2 segundos. Após a obtenção da configuração ótima do sistema, foram realizados novos testes, simulando o seu uso em situação real, onde a taxa de reconhecimento no melhor caso foi de 88,16%, com um tempo de reconhecimento por palavra inferior a 2 segundos.

Palavras-chave: Processamento Digital de Sinais. Reconhecimento Automático de Palavras Isoladas. *Endpoint Detection. Mel Frequency Cepstral Coefficients. Dynamic Time Warping.*

Abstract

This work presents the design of an isoleted words and speak dependent automatic speech recognition system, implmented in a low-cost computer system using the general purpouse interpreted programming language Python®. In the development of this system, the signal processing stage was implemented using the end point detection technique, aiming detect the beginning and end of an utterance; the technique used in the speech signal feature extraction was Mel frequency cepstral coefficients technique. The training and recognition stages were performed through the pattern comparison approach using Dynamic Time Warping -based techniques. For the system design, firstly, tests were performed in order to find out which configurations of the signal processing and recognition stages produced the best results in relation to the accuracy ratio and average runtime per word. In these tests, the best accuracy rate obtained for a vocabulary of 25 pre-recorded words was 96.8% with an average execution time per word of less than 2 seconds. After obtaining the optimum configuration of the system, new tests were performed, simulating its use in real situation, where the accuracy rate in the best case was 88.16%, with a decoding time per word of less than 2 seconds.

Keywords: Digital Signal Processing. Isoleted Words Automatic Speech Recognition. End Point Detection. Mel Frequency Cepstral Coefficients. Dynamic Time Warping.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ANN	*Artificial Neural Networks*
API	*Aplication Programming Interface*
ARM	*Advanced RISC Machine*
ASR	*Automatic Speech Recognition*
CPU	*Central Processing Unit*
DCT	*Discret Cossine Transform*
DFT	*Discrete Fourier Transform*
DTW	*Dynamic Time Warping*
EI	Erro provocado por palavras identificadas incorretamente
ENI	Erro provocado por palavras não identificadas
GUI	*Graphical User Interface*
HMM	*Hidden Markov Models*
HTK	*Hidden Markov Model Tool Kit*
IoT	*Internet of Things*
LPC	*Linear Prediction Coding*
LPCC	*Linear Prediction Cepstral Coeficients*
MFCC	*Mel Frequency Cepstral Coefficients*
NT	Número de *templates*
PCM	*Pulse Coding Modulation*
SLIT	Sistema Linear Invariante no Tempo
STE	*Short Term Energy*
STFT	*Short Term Fourier Transform*
TDTW	Tempo médio de execução da técnica DTW por palavra
TTS	*Text-To-Speech*

TVAD	Tempo médio de execução da técnica VAD por palavra
VAD	*Voice Activity Detection*
ZCR	*Zero Crossing Rate*

Sumário

1 Introdução

Da pré-história da humanidade passando pelos dias atuais e indo em direção ao futuro, a comunicação por fala tem sido, e provavelmente, continuará sendo a maneira dominante com a qual os humanos interagem socialmente e trocam informações. Esse tipo de comunicação hoje estende-se por meio da tecnologia com o emprego de recursos como o telefone, rádio, cinema, televisão e internet (HUANG; ACERO; HON, 2001).

O uso de sistemas de reconhecimento de fala automáticos (*Automatic Speech recognition* - ASR) que auxiliam na execução de tarefas cotidianas é a muito tempo especulado, sendo retratados a décadas, por exemplo, em filmes de ficção científica (RABINER; JUANG, 1993). Esse anseio mostra a preferência do ser humano pela comunicação em fala nas interações entre humanos e máquinas. Atualmente, a grande maioria dos sistemas computadorizados ainda faz uso de interfaces gráficas de usuário (*Graphical User Interface* - GUI), baseadas na representação gráfica de objetos e funções mostradas em uma tela ao usuário. Entretanto, o avanço tecnológico tem proporcionado o início da mudança nesse cenário.

Por muito tempo, o requisito computacional exigido para a implementação prática destes dispositivos foi um fator que inviabilizou o uso deste tipo de tecnologia. Entretanto, o aumento da capacidade computacional tem possibilitado o emprego de sofisticados métodos matemáticos para reconhecimento de fala em computadores e dispositivos embarcados os quais possuem por objetivo mudar a maneira como humanos interagem com as máquinas e seu ambiente. Nos dias de hoje já existe a possibilidade de se realizar tarefas em um smartphone ou computador por comandos de fala utilizando aplicações como: Google® Now, Apple® Siri, Microsoft® Cortana.

Com a explosão do paradigma da internet das coisas (*Internet-of-Things* - IoT) outras aplicações que também tiram vantagem de sistemas de reconhecimento de fala são encontradas em alguns modelos de automóveis, proporcionando a estes a possibilidade de realizar algumas tarefas utilizando comandos de fala como, por exemplo, atender uma ligação. Existe também a presença de dispositivos reconhecedores de fala em sistemas de automação residencial como o Google® Home e o Amazon® Echo.

Em todos os cenários de aplicação de dispositivos de automação de tarefas por comando de fala, desafios significantes existem, incluindo: robustez, flexibilidade, facilidade de integração e eficiência. O objetivo de construir sistemas de reconhecimento de fala viáveis tem atraído por muito tempo a atenção de engenheiros e cientistas por todo mundo (JURAFSKY; MARTIN, 2007).

Dentre os diversos cenários possíveis para a aplicação desses dispositivos, hoje em

dia, pode-se destacar o investimento de grandes companhias como a Google®, a Amazon® e a Apple® estão investindo em automação residencial, visando fornecer ferramentas que facilitem a rotina diária das pessoas. Contudo, os sistemas de automação residencial comerciais que utilizam tecnologias de reconhecimento de fala ainda não são vendidos oficialmente no Brasil. Quando adquiridos por importação, estes dispositivos possuem preços proibitivos para uma grande parcela da população.

Há ainda algumas alternativas que podem ser utilizadas para a construção de sistemas de ASR e que serão apresentadas na Seção 2.5, contudo, o uso dessas alternativas esbarra em impedimentos como: cobrança de taxas quando utilizados em aplicações comerciais, inexistência de suporte ao português brasileiro, ou ainda, emprego de métodos complexos que demandam muitos recursos de hardware, podendo tornar o seu emprego inviável em sistemas de baixo custo. Em vista destas restrições, este trabalho propõe a implementação de um sistema de ASR para palavras isoladas e dependente de locutor utilizando métodos de baixa complexidade, visando analisar o desempenho destas técnicas em microcomputador de baixo custo.

1.1 Objetivos do Trabalho

1.1.1 Objetivos Gerais

Os objetivos gerais deste trabalho são:

- Projetar, implementar e avaliar o desempenho de um sistema de ASR para palavras isoladas e dependente de locutor em sistema computacional embarcado. Para isso, primeiro será analisado o desempenho de algumas técnicas, as quais são amplamente utilizadas para a criação de sistemas de ASR de baixa complexidade. Por fim, será implementado o sistema de ASR a empregando as melhores configurações analisadas e o seu desempenho será avaliado.

1.1.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Implementar o bloco de processamento de sinais de um sistema de ASR utilizando uma técnica de baixa complexidade computacional chamada de detecção de extremos (*endpoint detection*) (RABINER; SAMBUR, 1975) para a remoção de trechos de silêncio existentes na palavra pronunciada e extrair os características de classificação linguística do sinal de fala de maneira a obter coeficientes cepestrais na escala de frequências Mel (*Mel Frequency Cepstral Coefficients* - MFCC) (HUANG; ACERO; HON, 2001);

- Implementar o bloco de reconhecimento de padrões empregando o método de distorção dinâmica do tempo (*Dynamic Time Warping - DTW*);
- Avaliar o desempenho das diferentes configurações das técnicas implementadas nos itens anteriores utilizando o microcomputador *Raspberry Pi 3 Model B+*;
- Implementar um sistema de ASR para palavras isoladas e dependente de locutor utilizando as configurações que apresentaram os melhores resultados no item anterior e avaliar o desempenho do sistema como um todo.

1.2 Organização do Documento

A partir do próximo capítulo, o presente trabalho de conclusão de curso está estruturado da seguinte forma: no Capítulo 2 é realizada uma revisão bibliográfica sobre os tipos de sistemas de processamento de fala, apresentando suas classificações de acordo com o emprego, bem como uma breve análise do processo de formação em compreensão de um sinal de fala, a arquitetura do sistema de ASR proposto, os principais produtos comerciais que empregam tecnologias ASR e outras ferramentas que possibilitam a construção destes sistemas; o Capítulo 3 apresenta uma análise mais detalhada do sinal de fala e das técnicas utilizadas para a implementação do bloco de processamento de sinais; no Capitulo 4 é desenvolvida uma breve revisão bibliográfica sobre os métodos de reconhecimento de palavras com uma análise mais detalhada da técnica DTW; no Capítulo 5 é mostrada a arquitetura típica dos sistemas de reconhecimento de fala implementados e, será feita uma breve revisão sobre a plataforma de testes, o microcomputador *Raspberry Pi 3 Model B+*; no Capítulo 6 são apresentados os resultados obtidos nos testes realizados. Neste mesmo capítulo também é realizada ainda a discussão a cerca destes resultados; por fim, no Capítulo 7 são apresentadas as considerações finais do trabalho e dadas algumas sugestões para trabalhos futuros.

2 Sistemas de Processamento de Fala

2.1 Tipos de Sistemas de Processamento de Fala

O processamento de fala é um ramo que estuda os fenômenos de produção e compreensão da fala com o objetivo de criar tecnologias relacionadas a esses fenômenos. Um sistema de processamento de fala geralmente pode ser formado por ao menos um dos três sistemas apresentados a seguir: sistema de reconhecimento automático de fala (*Automatic Speech Recognition* - ASR), sistema de texto-fala (*Text-To-Speech* - TTS) ou um sistema de compreensão de linguagem (HUANG; ACERO; HON, 2001).

Um sistema de ASR é geralmente criado com a função de reconhecer os padrões de um sinal de entrada de fala por meio de técnicas de reconhecimento de fala, transformando-o em palavras para que outra aplicação faça uso do resultado obtido. Os sistemas TTS podem ser vistos como o oposto a um sistema de ASR, ou seja, são sistemas geradores de fala a partir de uma entrada de texto.

Esses sistemas também podem ser chamados de sintetizadores de fala e são muito utilizados comercialmente, como, por exemplo, nas centrais de atendimento automático por telefone. Por fim, sistemas de compreensão de linguagem por sua vez são aparatos mais complexos, projetados para planejarem e executarem ações de acordo com os sinais de fala ou linguagem escrita utilizados como entrada. Esses sistemas devem ser robustos e implementados de tal maneira que possam interpretar a linguagem falada ou escrita. Os sistemas de compreensão de linguagem geralmente utilizam sistemas de ASR e TTS como subsistemas (HUANG; ACERO; HON, 2001).

Para aplicações de reconhecimento de fala, os sistemas de ASR são, atualmente, mais utilizados que sistemas de compreensão de linguagem. O principal motivo para essa preferência é o fato de não fazerem a interpretação do sinal de fala, visto que, essa operação é onerosa em termos de complexidade e de dados para o treinamento de um sistema de reconhecimento necessários para que um sistema interprete corretamente o comando.

Os sistemas de ASR são categorizados com base na dependência do locutor e/ou no tipo de fala. Para a primeira premissa pode-se ter sistemas dependentes ou independentes de locutor. Em um sistema dependente de locutor o reconhecimento é centralizado nos padrões de fala de um locutor específico. Já em sistemas independentes de locutor o sistema é apto a reconhecer a fala de qualquer locutor.

Em relação a segunda premissa um sistema pode ser projetado para reconhecimento de comandos ou palavras isoladas, ou pode ser desenvolvido para reconhecer fala contínua. Sistemas de reconhecimento de palavras isoladas são mais simples e tendem a ter resultados

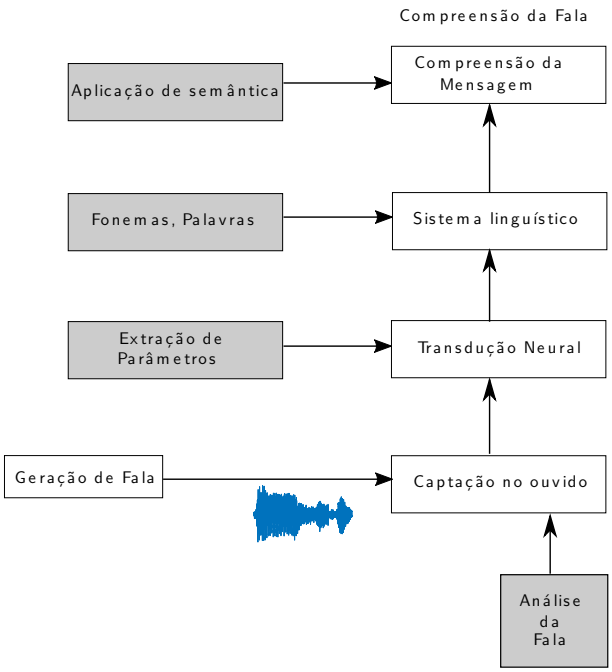
superiores aos de fala contínua, pois o sistema utiliza os intervalos existentes entre as palavras para ter referência do início ou término de uma palavra. Já sistemas de reconhecimento de fala contínua são mais complexos, visto que ocorrem poucas pausas durante a fala espontânea, além de outras características da linguagem falada poderem dificultar o reconhecimento das palavras (SILVA, 2010). A seguir será discutida a arquitetura de sistemas de ASR.

2.2 Arquitetura de Sistemas de ASR

2.2.1 Análise da Produção e Compreensão da Fala

A análise da percepção e construção da fala é um passo essencial para a implementação de um sistema de ASR, visto que, por meio dela é possível produzir caracterizações importantes na elaboração desses sistemas. De maneira simplificada, um sistema de ASR tenta replicar artificialmente os processos que ocorrem naturalmente nos seres humanos (RABINER; JUANG, 1993). Uma abordagem ao mecanismo de compreensão da fala é mostrada no diagrama de blocos exibido pela Figura 1, onde os blocos em cinza representam os métodos computacionais correspondentes a um sistema de processamento de fala natural.

Figura 1 – Diagrama de blocos das etapas envolvidas na compreensão de fala natural e seu equivalente computacional.



Fonte: Adaptado de (HUANG; ACERO; HON, 2001).

A compreensão da fala é realizada em estágios sequenciais: o primeiro estágio do processo natural de compreensão da fala inicia-se na captação do sinal nos ouvidos do

ouvinte, onde estruturas internas do ouvido atuam como bancos de filtros e realizam a análise em frequência e a filtragem do sinal recebido. No próximo estágio o sinal é captado por nervos auditivos através de um processo de transdução neural que extraem as componentes desejadas do sinal para o processo de entendimento da fala.

As características extraídas são então mapeadas pelo sistema linguístico e produzem mensagens a partir de palavras, fonemas e outras sub partículas do sistema de linguagem do ouvinte. Por fim, há a aplicação de semântica e outros recursos de interpretação da mensagem que concluem a tarefa e o sinal de fala recebido é compreendido pelo ouvinte.

Já o mecanismo de construção da fala é composto basicamente por duas componentes: a componente dependente de locutor e a componente que é independente de locutor. A componente dependente de locutor é criada no processo de formação da fala, por meio do ar expelido pelos pulmões. Se nesse processo as cordas vocais estão relaxadas, a passagem do ar pelo aparelho vocal não irá gerar sons com vibração e têm-se então a produção de sons não vozeados, que geralmente são tratados como ruído.

Entretanto, se as cordas vocais estão tensas, ao receber a passagem de ar elas vibrarão, gerando sons com oscilações periódicas, resultando nos sons vozeados. A frequência de vibração das cordas vocais varia de acordo com o orador e depende de fatores biológicos como idade, sexo e constituição biológica do trato vocal de cada indivíduo (BENESTY; SONDHI; HUANG, 2008).

A outra componente que compõe um sinal de fala é denominada formante. Os formantes são formados após o sinal de fala sair da laringe e passar pelas cavidades nasais e trato bucal (SILVA, 2003). Esse conjunto de aparelhos exerce o papel de dar formato aos sons vozeados oriundos da laringe, atuando como filtros acústicos, dando início ao processo de formação das vogais por meio da modulação da envoltória espectral de um som vozeado, caracterizando assim o timbre de um som de fala (SANTOS, 2013).

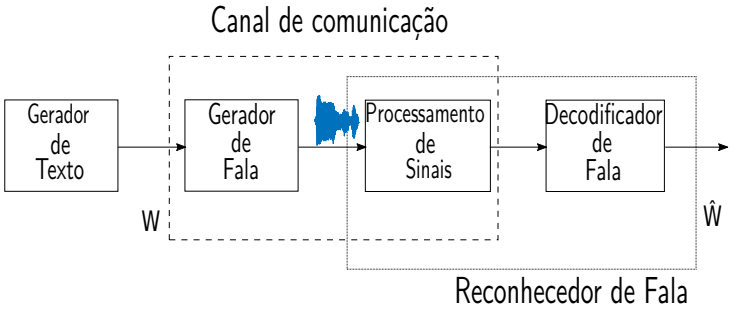
Dependendo das configurações do filtro acústico, como a posição da língua ou movimento dos lábios, pode-se ter formação de diversos padrões de formantes, modificando as características do som inicialmente produzido nas cordas vocais. Os formantes são características independentes de locutor, pois são dependentes da constituição do sistema linguístico de um idioma, visto que são ligados a pronúncia de uma vogal.

2.3 Arquitetura do Sistema de ASR Implementado

Para a sintetização artificial de um sistema de reconhecimento de fala que empregue os conceitos de compreensão e construção da fala é comum a adoção do modelo fonte-canal exibido na Figura 2 para elaboração dos problemas relacionados ao reconhecimento de fala (HUANG; ACERO; HON, 2001). Nesse modelo, a mente do locutor (fonte) geralmente é

modelada como o gerador de texto o qual decide a sequência de palavras (W) que será enviada para seu aparato gerador de fala. A fonte então envia os sinais de fala através do aparato vocal, que por sua vez é modelado como um canal de comunicação ruidoso, tendo por finalidade transformar a sequência de texto gerada, na fonte, nas formas de onda acústicas que serão transmitidas aos ouvintes. Esses sinais ao serem recebidos passarão por um sistema de processamento de sinais, no aparato auditivo do ouvinte, que irá reconhecer o sinal acústico. Por fim, o sinal chega ao sistema nervoso do ouvinte que irá reconhece-lo em uma sequência de palavras (), com a estimativa da mensagem transmitida.

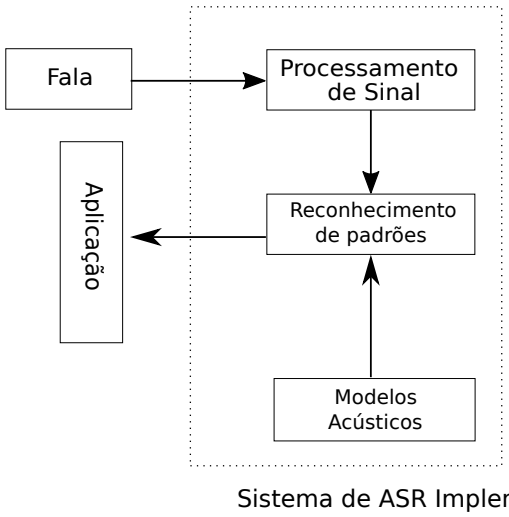
Figura 2 – Modelo de Fonte-Canal para um sistema de ASR.



Fonte: Adaptado de (HUANG; ACERO; HON, 2001).

O sistema de ASR implementado neste trabalho pode ser descrito como demonstrado na área pontilhada do diagrama de blocos exibido na Figura 3. Ao receber um comando de fala o sistema irá enviá-lo para o bloco de processamento de sinais. Esse bloco é responsável por separar a sentença de fala dos momentos de silêncio, detectando os pontos onde o sinal de fala se inicia e termina, extrair as características relevantes de um sinal de fala e enviá-lo na forma de um vetor acústico para o bloco de reconhecimento de padrões.

Figura 3 – Arquitetura geral dos sistemas de ASR implementados.



Fonte: Adaptado de (HUANG; ACERO; HON, 2001).

O reconhecedor de padrões é o bloco central do sistema de ASR. Esse bloco irá receber as informações enviadas por todos os outros blocos que compõem o sistema, ou seja, os blocos de processamento de sinais e de modelos acústicos, para decidir qual o grupo de palavras que um sinal de fala fornecido pelo bloco de processamento do sinal pertence.

No bloco de modelos acústicos encontram-se modelos com os sinais de referência do sistema previamente treinados, ou seja, os trechos de fala que serão utilizados como comandos com suas características de classificação linguística já extraídas que serão utilizadas para o reconhecimento da palavra de entrada pelo bloco de reconhecimento de padrões. Neste bloco encontra-se ainda o dicionário com os comandos utilizados para implementação do sistema.

A descrição detalhada da implementação de cada módulo do sistema será apresentada nos Capítulos 3 e 4. A seguir serão apresentados alguns sistemas de automatização residenciais comercializados e outras ferramentas que podem ser utilizadas para a construção de sistemas de ASR.

2.4 Sistemas de Reconhecimento de Fala Comerciais

Atualmente, existem diversas plataformas de sistemas de ASR comerciais para automação das mais variadas tarefas. Geralmente esses dispositivos são desenvolvidos para serem os principais componentes de ecossistemas compostos dos mais diversos dispositivos: smartphones, *smartwatches*, eletrodomésticos em geral, controladores de iluminação, entre outras. Dentre essas plataformas destacam-se: Google *Home*®, Amazon Echo® e Apple *HomePod*®.

2.4.1 Google *Home*®

O Google *Home*® é uma série de auto-falantes inteligentes (*smart speakers*) desenvolvida pela Google e lançada em 2017, com o objetivo de fornecer a automação de diversas atividades por meio de comandos de fala. Baseado no serviço de assistente pessoal Google *Assistant*, o Google *Home*® é um dispositivo eletrônico ASR e TTS, custando a partir de $ 49,90 dólares americanos (USD) para a versão mini (GOOGLE INC., 2019b).

Com capacidade para reconhecer a fala de até 6 usuários distintos, este dispositivo é acionado com o comando de fala: "*Hey Google*" e é configurado utilizando um smartphone que possua os sistemas operacionais Android ou iOS e precisa de acesso contínuo a internet (GOOGLE INC., 2019b). Este dispositivo pode ainda operar como unidade central de um ecossistema de aplicativos e dispositivos de automação residencial construídos pela Google e por empresas terceiras como: TP-Link, Samsung, entre outras.

Algumas funcionalidades disponibilizadas por esse sistema são: acesso às agendas, ligações telefônicas, leitura de *e-mails*, controle de centrais multimídia, entre outras. Para automação residencial existem dispositivos eletrônicos atuadores como: Google ChromeCast®, termostatos, trancas inteligentes para portas, controladores de iluminação, eletrodomésticos em geral, controladores de tomadas entre outros. Por fim, até o término deste trabalho este sistema não possui suporte oficial ao Português Brasileiro (TECNOBLOG, 2019).

2.4.2 Amazon Echo®

O Amazon *Echo*,® tal qual o Google *Home*® é uma marca de *smart speakers* desenvolvida pela Amazon encontrando-se atualmente em sua terceira geração (AMAZON INC., 2019). Utilizando o sistema de ASR de assistência pessoal Alexa, o *Echo*® é um aparato eletrônico que desempenha atividades semelhantes ao dispositivo da Google apresentado anteriormente.

Custando a partir de $ 49,99 USD e ativado pelo comando "*Alexa*", o dispositivo pode reconhecer a fala de múltiplos usuários, entretanto, exige que seja realizada a troca de perfil de usuário manualmente por meio de um aplicativo instalado em um smartphone com os sistemas operacionais Android ou iOS ou por um computador com acesso à internet (AMAZON INC., 2018). Este sistema também requer o uso de um smartphone ou computador para sua configuração (AMAZON INC., 2018). Tal qual o dispositivo da Google, esta tecnologia não é comercializada oficialmente no Brasil (AMAZON INC., 2019).

2.4.3 Apple HomePod®

O Apple *HomePod*® é uma marca de *smart speakers* desenvolvida pela Apple para concorrer com os sistemas apresentados anteriormente. Lançado em 2018 (APPLE INC., 2018), este dispositivo utiliza o sistema de ASR de assistência pessoal Siri, atua nos mesmos cenários de seus concorrentes e é ativado por meio do comando: "*Hey Siri*".

Para a tarefa de automação residencial, este produto utiliza diversos dispositivos que compõe o ecossistema Apple *HomeKit* ®, que consiste em um grupo de diversos dispositivos construídos para serem compatíveis com smartphones *Iphone*®. O *HomePod*® custa a partir de $ 349,00 USD. Para seu funcionamento o *HomePod*® ainda exige a sincronia com um smartphone *Iphone*®, além de suportar o reconhecimento de fala de apenas um usuário por dispositivo (THE AMBIENT, 2018). O *HomePod*® não é comercializado oficialmente no Brasil (APPLE INC., 2018).

2.4.4 IBM *Embedded* ViaVoice®

O IBM *Embedded* ViaVoice® foi um sistema de ASR e TTS para dispositivos embarcados, desenvolvido pela IBM para atuar, principalmente, em sistemas de automação veicular (IBM, 2007). Este sistema não precisava de conexão com a internet e possuía a capacidade de distinguir até 500 palavras em idiomas como: inglês, espanhol, francês, alemão, português, entre outros.

Outra característica interessante deste sistema era a possibilidade de treiná-lo a partir de frases ditas pelo usuário (IBM, 2007). Este procedimento era realizado com dois objetivos: melhorar o desempenho do sistema, treinando-o para reconhecer as características e padrões de fala de um usuário e para possibilitar a personalização de comandos, treinando o sistema com palavras específicas de acordo com a tarefa a ser executada. Dentre os produtos que fizeram uso deste sistema, destaca-se a sua aplicação em veículos da montadora Honda® (EXTREME TECH, 2019).

2.5 Outras Plataformas ASR

Além dos aparelhos eletrônicos já apresentados, existem diversas soluções que dispõem de *engines* com ferramentas para implementação de sistemas de ASR. Uma *engine* é formada por um conjunto de bibliotecas que auxiliam no desenvolvimento de uma aplicação e são acessadas por interfaces de programação de aplicações (*Aplication Programming Interface -API*). Para sistemas de ASR estas bibliotecas podem ser compostas por reconhecedores de padrão, dicionários do idioma entre outras (SILVA, 2010).

Dentre as soluções comerciais pode-se citar produtos proprietários fornecidos por empresas como: Microsoft com o serviço *Azure*®, Nuance e o software *Dragon Naturally Speaking*®, e o *SpeechTexter*®. Normalmente, as soluções fornecidas por essas empresas são pagas, não são multi-plataformas ou não suportam o português brasileiro (MICROSOFT INC., 2019), (NUANCE, 2019), (SPEECHTEXTER, 2019).

Para as soluções gratuitas destacam-se as *engines*: *Hidden Markov Model Toolkit* - HTK, Julius e *CMU Sphinx*. O HTK é uma ferramenta amplamente utilizada para fins de pesquisa em processamento de fala e foi desenvolvida pelo laboratório de inteligência artificial da universidade de Cambridge (HTK, 2019). Possuindo diversas ferramentas que permitem o desenvolvimento de aplicações ASR, o HTK possui ainda vasta documentação e uma comunidade ativa. Entretanto, atualmente a Microsoft detém os direitos sobre o código fonte do sistema (HTK, 2019), restringindo a distribuição das aplicações criadas com essa ferramenta.

O Julius é um reconhecedor de padrões para aplicações de ASR *Open Source*, gratuito, atualmente desenvolvido e mantido pela Universidade de Kyoto (JULIUS TEAM,

2019). Este software desempenha apenas a função de reconhecimento de padrões, necessitando de ferramentas fornecidas por aplicações terceiras para implementar um sistema de ASR completo. Por fim, o *CMU Sphinx* é uma *engine* desenvolvida pelo grupo *CMU Sphinx* da Universidade de Carnegie Mellon (CMU SPHINX, 2019) e oferece diversos recursos para implementação de sistemas de ASR como reconhecedores de padrões, classificadores de palavras, processadores de sinais, entre outros.

Todas essas *engines open source* utilizam em seus reconhecedores de padrões sofisticados métodos estatísticos especializados em fala contínua baseados em cadeias ocultas de Markov (*Hidden Markov Models* - HMM) (HTK, 2019), (JULIUS TEAM, 2019), (CMU SPHINX, 2019), que demandam um grande número de amostras de fala para seu treinamento (BENESTY; SONDHI; HUANG, 2008) (HUANG; ACERO; HON, 2001). Visto que nenhuma dessas ferramentas possuem suporte para o português brasileiro, o seu emprego está condicionado a criação de um banco de dados de arquivos de fala em português brasileiro e adaptação de seus modelos acústicos e linguísticos para este idioma.

Existem ainda diversas API's ligadas a *engines* de fala direcionadas a criação de soluções utilizando ASR como: Google Web*Speech API*®, *Houndify*® e *IBM Speech to text*®. Apesar de todas estas API's serem gratuitas, elas possuem problemas para os requisitos do projeto descrito neste trabalho como: cobranças de taxas para uso comercial, algumas delas não oferecem suporte ao português brasileiro, as *engines* são proprietárias não permitindo a customização do sistema (REAL PYTHON, 2019).

3 Sistema de Processamento de Sinais

3.1 O Sinal de Fala

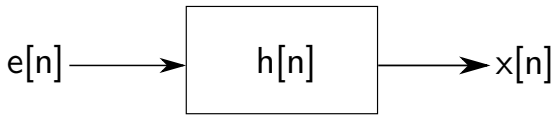
Para uma boa implementação do módulo de processamento de sinais, é necessário que se compreenda as características acústicas da formação de sons que compõem um sinal de fala e como estes sons interagem com os mecanismos de pronuncia de uma palavra. Para tal, na literatura, é comum realizar-se a decomposição de um sinal de fala no modelo fonte-filtro (HUANG; ACERO; HON, 2001),(BENESTY; SONDHI; HUANG, 2008),(RABINER; JUANG, 1993).

Neste modelo, como ilustra a Figura 4, um sinal de fala (x[n]) pode ser representado como a convolução entre um sinal de variação rápida, produzido pela excitação do trato vocal no processo de formação da fala (e[n]), e um filtro de variação lenta no tempo (h[n]), correspondente ao processo de pronúncia de uma sentença (HUANG; ACERO; HON, 2001).

Tradicionalmente, os sistemas de reconhecimento de fala baseiam-se na estimação das características do filtro que modela a envoltória do sinal acústico gerado pela pronúncia de uma palavra, ignorando a fonte formadora de sons e obtendo os formantes por meio da extração das características deste filtro. Matematicamente, segundo (HUANG; ACERO; HON, 2001) o modelo fonte-filtro pode ser definido por

$$x[n] = e[n] \circledast h[n]. \tag{3.1}$$

Figura 4 – Modelo básico de fonte-filtro para sinais de fala.

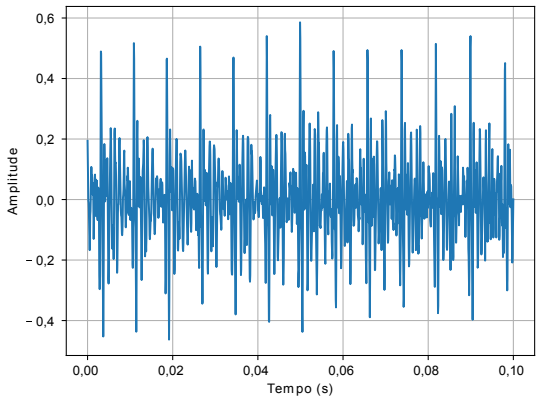


Fonte: Adaptado de (HUANG; ACERO; HON, 2001).

Para estimação do filtro, deve-se primeiro compreender as propriedades físicas dos sons vozeados e não vozeados que constituem a fala. O sons vozeados consistem em tons de frequência periódicas, com uma maior quantidade de energia nas frequências mais baixas e que são produzidos quando uma vogal é pronunciada em um fonema, dando origem aos formantes (SANTOS, 2013). A sua produção dá-se pela vibração das cordas vocais quando o ar expelido pelos pulmões passa pelo trato vocal, com a frequência fundamental dependendo da constituição vocal de cada locutor. A Figura 5 ilustra um segmento de 100 mili segundos (ms) da forma de onda no domínio temporal para a pronuncia da vogal “a”,

mostrando a característica de periodicidade da pronuncia de uma vogal em um fonema. Os sons não vozeados não possuem estrutura periódica, possuem uma maior concentração

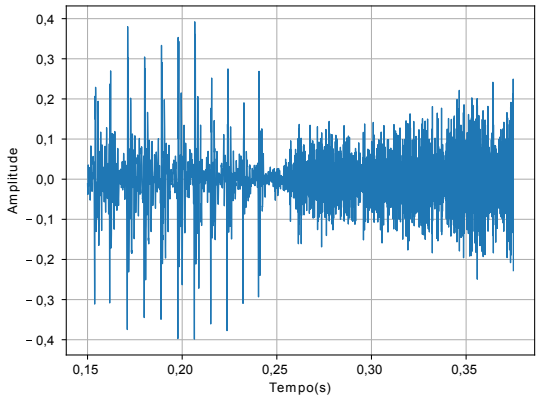
Figura 5 – Forma de onda para a pronuncia da vogal “a” segmentada em 100ms.



Fonte: Autoria Própria.

de energia nas frequências mais altas e são caracterizados como ruído, pois durante a sua pronúncia não há excitação das cordas vocais quando o ar passa pelo trato vocal, gerando uma estrutura de sinal aleatória (RABINER; JUANG, 1993). A Figura 6 ilustra um segmento de 250 ms da forma de onda no domínio temporal para a pronuncia do trecho que é formado pelas letras “a” e “c” na palavra “acender”. Pode-se observar nesta figura a característica de ruído da pronuncia de uma consoante em relação a pronúncia de uma vogal.

Figura 6 – Forma de onda da transição entre a vogal “a” e a consoante “c” na palavra “acender”.

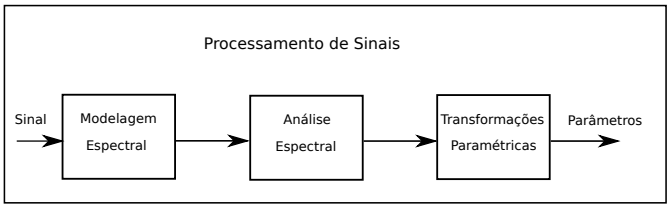


Fonte: Autoria Própria.

Aproveitando-se dessa característica, pode-se implementar no módulo de processamento de sinais técnicas que estimem os formantes de uma sentença para sua classificação

linguística. O bloco de processamento de sinais em um sistema de ASR é implementado em três etapas: a modelagem espectral, a análise espectral e, por fim, a extração de características (PICONE, 1993). A estrutura de um módulo de processamento de sinais é descrita com o diagrama de bloco da Figura 7.

Figura 7 – Diagrama de blocos do módulo de processamento de sinais.



Fonte: Adaptado de (PICONE, 1993).

3.2 Modelagem Espectral

A etapa de modelagem espectral é a primeira etapa do processo de processamento de sinais e envolve basicamente duas operações: a conversão analógico-digital (A/D) do sinal de entrada, onde as ondas de pressão analógicas do som serão transformadas em formas de onda digitais, e a detecção dos trechos de fala, visando realizar a remoção dos segmentos em momentos onde só há silêncio em um sinal de fala para a melhora no desempenho do sistema. Nesta etapa, ainda são realizados procedimentos que serão utilizados durante a fase de extração de características do sinal como a segmentação e o janelamento.

Primeiramente, o sinal analógico de fala é amostrado e quantizado, resultando em um sinal em tempo discreto com níveis de amplitude específicos. Este processo é feito de acordo com o teorema da amostragem de Nyquist-Shanon, onde um sinal limitado em banda com uma certa frequência finita f_{max} é amostrado com uma frequência de amostragem mínima de $2f_{max}$, para que o sinal possa ser corretamente representado e a reconstrução do sinal a partir da sua forma digital seja possível (RABINER; SCHAFER, 2007).

Como a fala humana se concentra principalmente entre as frequências de 100 Hz a 8 kHz, deve-se utilizar no mínimo 16 kHz de frequência de amostragem. Na prática, um sinal digital com frequência de amostragem em 16 kHz, com 16 bits de quantização e apenas um canal de áudio (*mono*) é uma configuração suficiente para uma aplicação de reconhecimento de fala com bom desempenho (HUANG; ACERO; HON, 2001).

Com esta etapa finalizada, inicia-se o processo de preparação do sinal para a implementação das etapas seguintes: detecção de início e fim das sentenças, também conhecida como detecção de atividade de voz (*Voice Activity Detection - VAD*) (RABINER;

JUANG, 1993), (BENESTY; SONDHI; HUANG, 2008), e extração de características de classificação do sinal. Para isso, é realizada a segmentação do sinal de fala em quadros de menor duração de tempo.

Esta operação é necessária pois o processo de geração do sinal de fala possuí características não estacionárias, ou seja, as suas componentes de frequência, ou componentes espectrais, variam em todo intervalo de tempo. Esta condição ocorre devido a natureza de variação lenta da pronuncia de uma palavra em relação aos sinais produzidos pela excitação do trato vocal. A ressonância criada no trato bucal e nasal durante o evento de pronúncia de uma palavra também é um fator que causa este tipo de comportamento. Esta condição impossibilita a aplicação direta de técnicas VAD e uma adequada extração de características de classificação do sinal de fala (HUANG; ACERO; HON, 2001).

Este problema pode ser contornado através da segmentação do sinal de fala original em quadros de menor duração de tempo. Neste processo, o tempo de duração de um quadro será menor que o tempo necessário para o sinal de fala variar, tornando-o estacionário, ou seja, constante no intervalo de tempo de duração do quadro (RABINER; SCHAFER, 2007). Pode-se obter a segmentação do sinal de fala original em quadros de menor duração de tempo por meio da multiplicação do sinal original por uma função janela.

Existem diversas funções janela que oferecem condicionamentos específicos de sinais como a janela Gaussiana, de Hamming, de Kaiser, entre outras. Todavia, para processamento de fala a janela de Hamming é empregada quase exclusivamente, com a finalidade de evitar os efeitos de vazamento espectral (BENESTY; SONDHI; HUANG, 2008). Matematicamente, têm-se

$$x_i[n] = x[n]w[i-n], \quad 0 \leq n \leq N-1 \tag{3.2}$$

onde $x_i[n]$ representa o sinal de fala $x[n]$ no instante de tempo representado pelo quadro i e $w[n]$ é a função janela de Hamming deslocada no tempo, a qual é definida por

$$w[n] = 0,54 - 0,46cos\left(\frac{2\pi n}{N-1}\right) \tag{3.3}$$

com $0 \leq n \leq N-1$ sendo o tamanho da janela.

A duração do período de segmentação do sinal, em quadros, nesta etapa tem relação com o idioma do locutor e é uma relação entre desempenho e tempo de resposta. Sistemas que empregam quadros com longos períodos de duração (entre 40 ms e 100 ms) são processados mais rapidamente, contudo, podem perder informações importantes da variação do sinal. Já o uso de quadros com tempo de duração inferiores a 10 ms, aumentam o tempo necessário para processamento do sinal e, muitas vezes, não trazem benefícios (HUANG; ACERO; HON, 2001).

Na literatura, para esse bloco é comum segmentar o sinal em quadros com duração de 25 ms com o deslocamento do quadro de análise a cada 10 ms buscando uma sobreposição

de quadros de ao menos 50 % entre os quadros, para a maioria dos idiomas, sendo exceção poucos tipos de idioma, como alguns dialetos chineses (BENESTY; SONDHI; HUANG, 2008).

Finalizado o processo de segmentação do sinal, inicia-se a etapa onde separam-se os trechos de sinais de fala dos segmentos de silêncio, onde há apenas ruído de fundo, aplicando-se as técnicas VAD. Atualmente, existem na literatura diversas técnicas para extração de características de um sinal de fala, criadas com a finalidade de projetar sistemas VAD. Usualmente, estas técnicas utilizam o modelo fonte-filtro para efetuar a análise das características do sinal de fala como: classificação de sons vozeados e não vozeados, período de frequência fundamental da fala, entre outros.

Em (RABINER; SAMBUR, 1975) é apresentado um método denominado de detecção de extremos (*endpoint detection*), que utiliza características temporais do sinal de fala para se obter informações sobre os sons vozeados e os sons não vozeados. Nessa técnica realiza-se o cálculo da energia em tempo curto (*Short Term Energy* - STE), e da taxa de cruzamento por zeros (*Zero Crossing Rate* - ZCR) de uma sentença de fala segmentada em quadros de menor duração para determinar os períodos de início e fim de uma palavra. Por ser um técnica bem difundida na literatura, que obtém bons resultados e exige baixo custo computacional, neste trabalho optou-se pelo uso desta técnica.

3.2.1 Algoritmo *Endpoint Detection*

O algoritmo apresentado por (RABINER; SAMBUR, 1975) aplica as análises STE, para localizar os pontos onde existem trechos de sons vozeados e a ZCR que determina a presença ou ausência de sons não vozeados, para implementar um método de estimação de início e fim de uma sentença baseado em 3 premissas:

- Ser simples e de processamento eficiente;
- Informar de forma confiável a localização de eventos acústicos significantes;
- Ter a capacidade de ser aplicado nos mais diversos cenários.

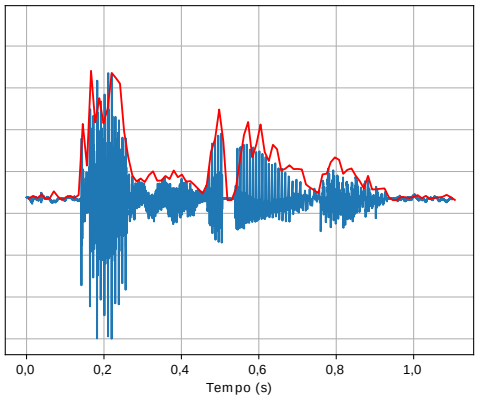
Segundo (RABINER; SAMBUR, 1975), para o seu correto funcionamento este algoritmo deve segmentar o sinal em quadros de ao menos 10ms de duração para o cálculo da STE e ZCR do sinal de fala, e restringir os 10 primeiros quadros a existência apenas do silêncio, para que se possa obter estatísticas como: a média e o desvio padrão (σ) da ZCR do ruído e a sua STE média. Para a implementação desta técnica, primeiramente calcula-se a energia existente em cada quadro de um sinal de fala após a sua segmentação usando a STE. Em (RABINER; SCHAFER, 2007) define-se a STE de um sinal de fala

discreto por

$$E_i[n] = \sum_{n=0}^{N-1} x_i[n]^2 \tag{3.4}$$

Nesta técnica, a duração do período de tempo do quadro deve ser ajustada para que se possa ter uma boa representação das variações de energia da forma de onda do sinal de fala. A literatura comumente adota períodos de tempo com duração entre 10ms e 100ms (RABINER; SAMBUR, 1975), (RABINER; SCHAFER, 2007). A Figura 8 exibe um exemplo da extração da STE, sobreposta em vermelho, de um sinal de fala utilizando a segmentação em quadros com duração de 10ms. Pode-se observar que com o uso da STE têm-se uma boa estimativa da envoltória do sinal de fala, o que pode ajudar a identificar o início de uma sentença.

Figura 8 – Sinal de fala com a STE sobreposta.



Fonte: Autoria Própria.

A STE é uma técnica amplamente utilizada em aplicações voltadas ao reconhecimento fala, extraindo informações importantes do sinal de fala e demandando um baixo custo computacional. Especificamente, essa técnica possibilita uma boa distinção de sons vozeados em um sinal de fala e em situações de baixo ruído de fundo pode indicar o início ou o fim de uma sentença (RABINER; SCHAFER, 2007).

Apesar de ser uma técnica eficiente na distinção de sons vozeados no sinal de fala, muitas vezes pode-se ter sons não vozeados no início ou no fim de uma sentença, e o uso da STE por si só pode não ser capaz de identificar estes sons, pois podem ser confundidos como o ruído de fundo (RABINER; SCHAFER, 2007). Como sons não vozeados possuem uma maior concentração de energia em frequências mais altas, estes apresentam um valor de ZCR mais alto que o ruído de fundo, uma vez que existem mais cruzamentos por zero na forma de onda nestes períodos.

Assim, o emprego da ZCR pode oferecer uma boa métrica da presença ou ausência de sons não vozeados no início ou fim das sentenças. De maneira similar a STE, a ZCR é

um tipo de análise em tempo curto que calcula o número de vezes em que uma forma de onda cruza o eixo zero em determinado quadro do sinal.

Segundo (RABINER; SCHAFER, 2007), a ZCR é uma técnica de baixo custo computacional, é calculada utilizando as mesmas configurações de segmentação do sinal original de fala e pode ser obtida por

$$Z_i[n] = \sum_{n=0}^{N-1} \frac{1}{2} |sgn(x[n]) - sgn(x[n-1])| w[i-n] \tag{3.5}$$

onde

$$sgn(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases} \tag{3.6}$$

após o cálculo da STE e ZCR do sinal de fala, o próximo passo para a implementação da técnica de VAD proposta é calcular os limiares de decisão para as métricas de energia por meio de:

$$I_1 = 0,03(I_{MX} - I_{MN}) + I_{MN} \tag{3.7}$$

$$I_2 = 4I_{MN} \tag{3.8}$$

$$I_{TL} = min(I_1, I_2) \tag{3.9}$$

$$I_{TU} = 5I_{TL} \tag{3.10}$$

onde I_{MX} é o pico de energia do quadro, I_{MN} é a média de energia do ruído, I_1 representa 3% do nível de energia de pico ajustada pela energia do ruído, I_2 representa quatro vezes o nível de energia do ruído, I_{TL} é o limiar mínimo de energia e I_{TU} é o limiar máximo.

Para o cálculo do limiar de ZCR (I_{ZCT}) utiliza-se

$$I_{ZCT} = min(I_F, I_{ZC} + 2\sigma_{i_{zc}}) \tag{3.11}$$

onde o índice I_F determina um limiar fixo de ZCR. Em (RABINER; SAMBUR, 1975) este limite é fixado em 25 para uma taxa de amostragem de 10 kHz. O autor adaptou este limiar por meio de

$$I_F = \frac{25fs}{10000} \tag{3.12}$$

onde fs é a frequência de amostragem de 16 kHz utilizada. O parâmetro I_{ZC} é média da ZCR presente nos quadros iniciais, onde existe apenas ruuído de fundo, determinado por:

$$I_{ZC} = \sum_{n=1}^{M} \frac{ZCR_{Ruido}[n]}{M} \tag{3.13}$$

onde M é o número total de quadros durante o trecho de silêncio e $\sigma_{I_{ZC}}$ é o desvio padrão do parâmetro I_{ZC} definido por

$$\sigma_{I_{ZC}} = \sqrt{\frac{1}{M-1} \sum_{n=1}^{M} \left| ZCR_{Ruido}[n] - \frac{\sum_{n=1}^{M} ZCR_{Ruido}[n]}{M} \right|}. \tag{3.14}$$

A Figura 9 exibe o diagrama de blocos de (RABINER; SAMBUR, 1975) para implementação do algoritmo, e o processo de busca de início da sentença deste algoritmo, que segue os seguintes passos:

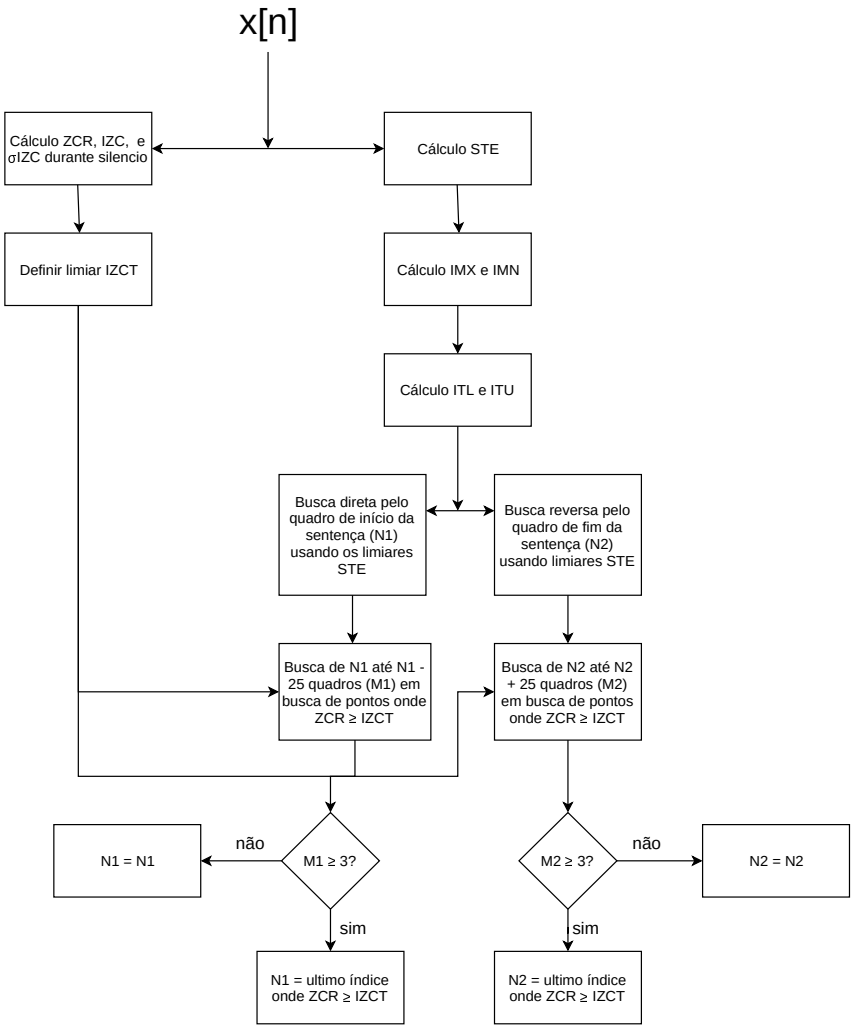
1. Iniciar buscando o quadro onde STE[n] $\geq I_{TL}$. Quando este evento ocorrer o quadro deve ser marcado como referência para o início da sentença (STE[r]);
2. Iniciar uma nova busca usando o quadro de referência como ponto de início até encontrar um quadro onde STE[r] $\geq I_{TU}$ e marcar este quadro como início da sentença (N1), se nesta busca encontrar quadro onde STE[r] $\leq I_{TL}$ parar o laço e reiniciar o primeiro item com n = n+1;
3. Se r = n então mantém N1, se não N1 = N1 - 1;
4. Iniciar busca reversa a partir de N1 e voltar 25 quadros calculando ZCR;
5. Se ZCR $\geq I_{ZCT}$ então N1 é igual ao quadro de referência, se não, retorna ao início;
6. Termina algoritmo.

Para o algoritmo de busca de fim da sentença o método é análogo, iniciando o algoritmo a partir do último quadro do sinal de fala e finalizando no primeiro. Após o término do algoritmo de *endpoint detection* inicia-se a próxima etapa do sistema de processamento de sinais, o bloco de análise espectral.

A Figura 10 exibe o resultado obtido após a implementação desta técnica utilizando a segmentação do sinal de fala em quadros com duração de 10ms. É possível observar que o algoritmo implementado atingiu o objetivo de extrair os trechos de silêncio no inicio e no fim da sentença de fala proferida. Para o exemplo ilustrado nesta figura a redução atingida foi de aproximadamente 60% do arquivo de áudio.

Um aspecto importante a se analisar na técnica *endpoint detection* é a sua sensibilidade a certos trejeitos da fala. Os trejeitos que ocorrem mais comumente durante a

Figura 9 – Diagrama de blocos para implementação do algoritmo usado.



Fonte: Adaptado de (RABINER; SAMBUR, 1975).

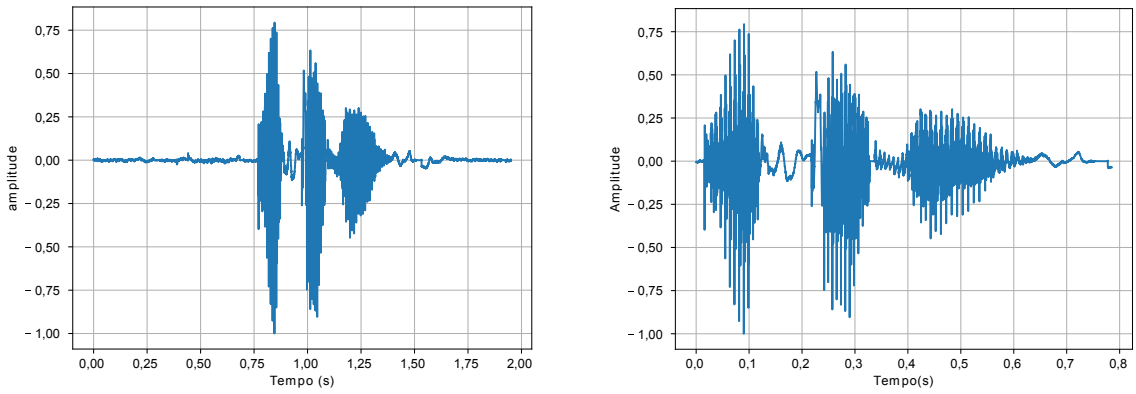
pronuncia de uma palavra são o *click* e o sopro (BENESTY; SONDHI; HUANG, 2008). O *click* pode ocorrer devido ao estalar da língua ou dos lábios, antes ou depois da pronúncia da palavra. Já o sopro ocorre de maneira mais comum devido a respiração após a pronúncia de uma palavra ou devido a inspiração antes da pronúncia.

A Figura 11 ilustra os resultados obtidos após o uso da técnica *endpoint detection* em um sinal de fala que possuí em sua composição os dois fenômenos, deliberadamente inseridos para ilustrar o problema. Nota-se nesta figura que esta técnica não é capaz de identificar corretamente estes fenômenos, resultando em recortes ruins do sinal de fala, o que pode gerar erros de reconhecimento.

A Figura 12 apresenta a análise da STE no sinal de fala com a presença do *click* e do sopro ilustrado pela Figura 11a. Observa-se que o nível de energia para esses dois fenômenos é alto, em relação ao ruído. Na técnica *endpoint detection* primeiro é realizado a análise da STE do sinal de fala, para determinar as regiões de início e fim da elocução.

Como os fenômenos de *click* e sopro possuem um nível de energia alto em relação ao ruído, o que pode ser visto na Figura 12, estes se enquadram como regiões de início e fim da sentença pelo algoritmo, resultando em um recorte ruim do sinal de fala.

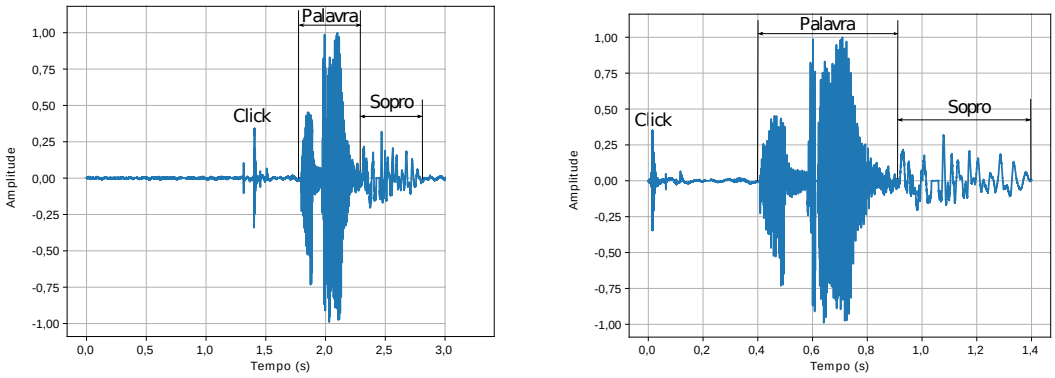
Figura 10 – Formas de onda de sinal de fala antes (a) e depois (b) da aplicação do *endpoint detection*.



(a) Sinal de fala original. (b) Sinal de fala obtido.

Fonte: Autoria Propria.

Figura 11 – Analise um sinal de fala antes (a) e depois (b) da aplicação do *endpoint detection* com os efeitos de *click* e sopro.



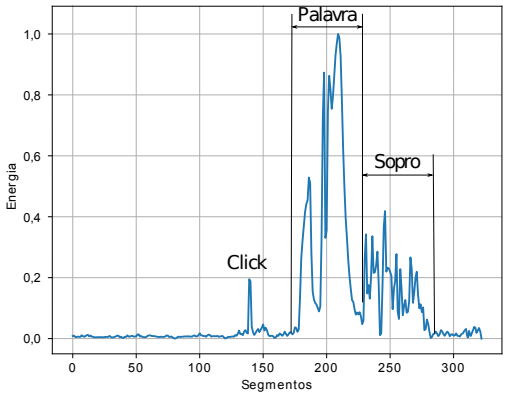
(a) Sinal de fala original. (b) Sinal de fala obtido.

Fonte: Autoria Propria.

3.3 Análise Espectral

O bloco de análise espectral tem por objetivo calcular o espectro do sinal de fala, realizar a medida de sua potência e transformá-la em escala logarítmica, visto que, a audição humana tem percepção de sons dessa maneira (BENESTY; SONDHI; HUANG,

Figura 12 – Análise da STE após a aplicação do *endpoint detection* com os efeitos de *click* e sopro.



Fonte: Autoria Propria.

2008). O cálculo do espectro de um sinal digital pode ser feito por meio do uso da transformada discreta de Fourier (*Discrete Fourier Transform* - DFT). Entretanto, dadas as características não estacionárias da fala, não é possível o uso direto da DFT.

Conforme apresentado na Seção 3.2, este problema pode ser contornado através da segmentação do sinal de fala original em quadros de menor duração de tempo. Após a segmentação do sinal de fala pode-se calcular suas componentes espectrais utilizando a transformada de Fourier de tempo curto (*Short-Time Fourier Transform* - STFT) (BENESTY; SONDHI; HUANG, 2008). Esta técnica trata-se simplesmente da aplicação da DFT, quadro a quadro, para efetuar a análise espectral de cada quadro do sinal por meio de

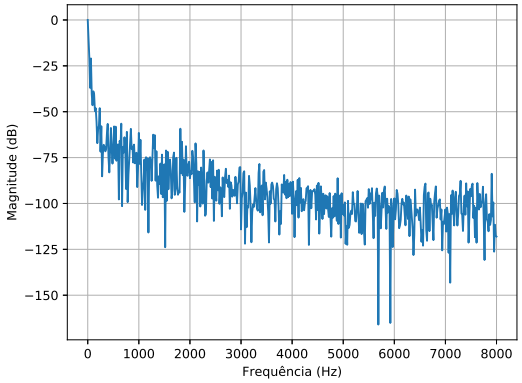
$$X_i[k] = \sum_{n=0}^{N-1} x_i[n] e^{\frac{-j2\pi nk}{N}}, \tag{3.15}$$

onde k é o índice de frequência normalizada e N é o tamanho da DFT.

As Figuras 13 e 14 ilustram resultado do cálculo do espectro de um sinal de fala. Na Figura 13 têm-se a representação espectral sem a segmentação do sinal. A Figura 14 exibe o mesmo sinal de fala após o uso da segmentação. Comparando as duas figuras observa-se que durante o período de segmentação há a presença de estruturas periódicas características do sinal de fala, o que facilita a sua análise.

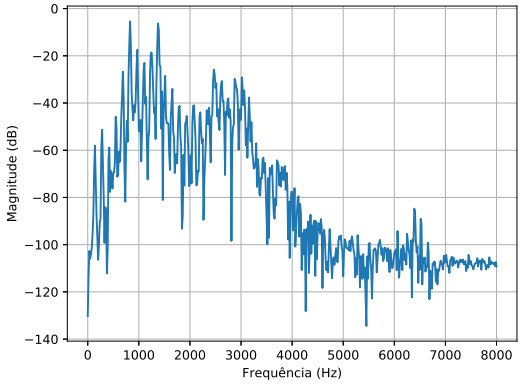
Já a Figura 15 exibe a análise desta sentença por meio de um espectrograma. Como pontuado por (HUANG; ACERO; HON, 2001), espectrogramas são ferramentas de análise amplamente empregadas em sistemas de processamento de fala e áudio. A ideia por trás de um espectrograma é calcular a DFT de um sinal em pequenos períodos de tempo, exibindo o sinal em uma representação em duas dimensões com o domínio da frequência em seu eixo vertical e o domínio do tempo em seu eixo horizontal. Quando aplicado a um sinal de fala o espectrograma permite visualizar tanto o tempo quanto as frequências onde

Figura 13 – Espectro da sentença de fala “acender” sem a utilização da operação de janelamento.



Fonte: Autoria Própria.

Figura 14 – Espectro da sentença de fala “acender” utilizando da operação de janelamento.



Fonte: Autoria Própria.

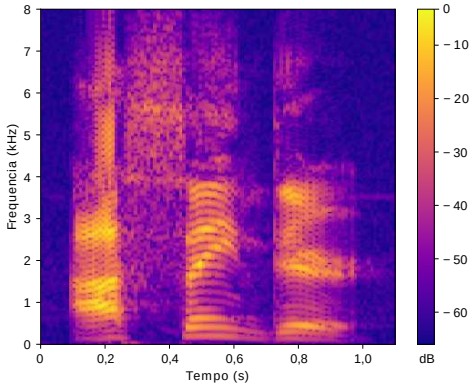
encontram-se os formantes, denotados pelas estruturas de características periódicas no espectrograma da Figura 15.

3.4 Extração de Características

Após processo de análise espectral, aplica-se o procedimento para a extração das características para a classificação linguística de uma mensagem. Um sinal de fala contém muito mais informações além das necessárias para sua classificação que, se forem consideradas, poderão acarretar em erros de precisão na detecção de fala (BENESTY; SONDHI; HUANG, 2008).

Atualmente, existem diversas técnicas propostas para a extração das características de um sinal de fala, tais como a análise por predição linear (*Linear Prediction Coding* - LPC) (HUANG; ACERO; HON, 2001), a análise por predição linear utilizando coeficientes

Figura 15 – Espectrograma da sentença “acender”.



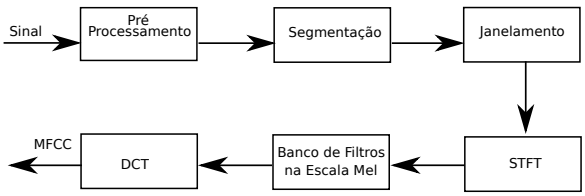
Fonte: Autoria Própria.

cepstrais (*Linear Prediction Cepstral Coefficients* - LPCC) (HUANG; ACERO; HON, 2001), coeficientes cepstrais em escala de frequências Mel (*Mel Frequency Cepstral Coefficients* - *MFCC*) (HUANG; ACERO; HON, 2001), (BENESTY; SONDHI; HUANG, 2008), a extração de parâmetros baseada em *Wavelets* (RABINER; JUANG, 1993) e a extração de parâmetros utilizando redes neurais (BENESTY; SONDHI; HUANG, 2008).

3.4.1 MFCC

Dentre as técnicas acima apresentadas, o MFCC é bastante empregado em aplicações ASR (BENESTY; SONDHI; HUANG, 2008). Baseado em experimentos que visavam compreender o mecanismo humano de entendimento de palavras, essa técnica imita partes desse mecanismo, como a percepção humana de intensidade e a frequência fundamental (*pitch*) de um som, tentando separar as componentes dependentes de locutor das componentes independentes de locutor (MERMELSTEIN, 1976). Baseado nos procedimentos já adotados e nos que serão adotados adiante, a Figura 16 ilustra o diagrama de blocos implementação desta técnica.

Figura 16 – Diagrama de blocos para implementação da técnica MFCC.



Fonte: Autoria Própria.

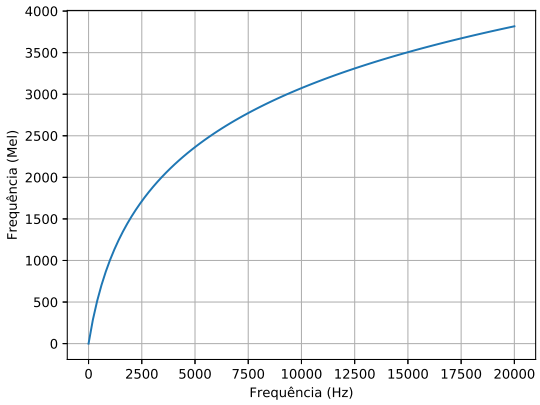
A técnica do MFCC aproveita-se da existência de formantes em um sinal de fala para a extração de parâmetros que permitem o seu reconhecimento. Para tal, o primeiro procedimento do método emprega um banco de filtros digitais do tipo passa-faixas, que

tem por objetivo reproduzir, de forma artificial, os estágios iniciais de transdução no sistema auditivo humano. Por possuir essa característica de mímica do aparelho de audição humana, o banco de filtros deve ser projetado de modo que possua resposta em frequência de acordo com a escala de percepção humana.

Estudos mostram que a percepção do ouvido humano responde a variação do *pitch* de forma linear para frequências até 1000 hz e que, acima disso, a escala de percepção torna-se logarítmica (MERMELSTEIN, 1976). Para mapear as frequências acústicas para uma escala perceptual, é comum em sistemas de ASR utilizar a escala de frequências Mel (PICONE, 1993), que foi elaborada ao medir-se a percepção humana ao incremento de frequência do *pitch*. A Figura 17 ilustra a escala Mel em relação a escala de frequência. A sua obtenção, segundo (HUANG; ACERO; HON, 2001), pode ser feita por meio de

$$f_{Mel} = 2595 log_{10}\left(1 + \frac{f}{700}\right) \tag{3.16}$$

Figura 17 – Escala de percepção Mel x escala em frequência.



Fonte: Autoria Própria.

O estudo apresentado por (MERMELSTEIN, 1976) mostra ainda que o aparelho auditivo não é capaz de distinguir individualmente entre dois tons que possuam frequências próximas uma certa frequência nominal arbitrária, ou seja, se esses tons estiverem dentro da largura de banda da frequência central. Contudo, se um dos tons estiver fora dessa largura de banda, ele será distinguido. A literatura refere-se a essa largura de banda como largura de banda crítica (RABINER; JUANG, 1993), a qual pode ser calculada como

$$BW_{crítico} = 25 + 75\left(1 + 1,4\left[\frac{f}{1000}\right]^2\right)^{0,69}. \tag{3.17}$$

Combinando as duas teorias apresentadas, pode-se implementar uma técnica de análise conhecida como banco de filtros de banda crítica (PICONE, 1993). Esses filtros são simplesmente um banco de filtros FIR passa-faixa, de fase linear, que são arranjados

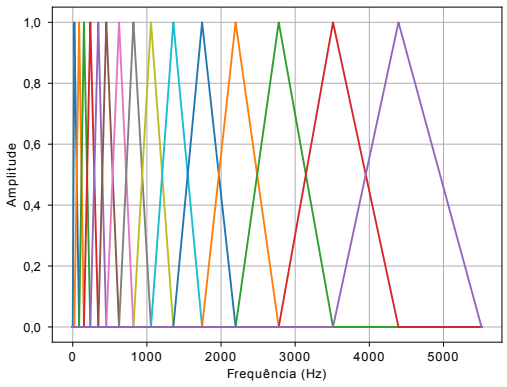
linearmente sobre a escala de frequências Mel. A largura de banda desses filtros é escolhida de forma que sejam iguais a largura de banda crítica de uma frequência central correspondente. A literatura apresenta diversos formatos de filtros que podem ser implementados, tais como filtros retangulares e triangulares, entretanto, a forma mais popular é o filtro triangular (HUANG; ACERO; HON, 2001). Um banco de filtros triangulares pode ser implementado utilizando

$$H_m[k] = \begin{cases} 0 & k < f[m-1]] \\ \frac{(k-f[m-1])}{(f[m]-f[m-1])} & f[m-1] \leq k \leq f[m] \\ \frac{(f[m+1]-k)}{(f[m+1]-f[m])} & f[m] \leq k \leq f[m+1] \\ 0 & > f[m+1]] \end{cases} \tag{3.18}$$

onde M é o número de filtros tal que $m = 1, 2, \ldots, M$. Para uma melhor ilustração, um banco de filtros com 15 componentes e frequência máxima de 5 kHz é mostrado na Figura 18. Trabalhos na literatura recomendam utilizar um número de coeficientes entre 26 e 40 filtros, onde a escolha do número de coeficientes é um compromisso entre precisão e desempenho do sistema (HUANG; ACERO; HON, 2001). O sinal resultante pode ser expressado por:

$$S_i[m] = \sum_{k=0}^{N-1} X_i[k] H_m[k]. \tag{3.19}$$

Figura 18 – Banco de filtros na escala Mel com 15 coeficientes e frequência máxima de 5 kHz.



Fonte: Autoria Própria.

O último procedimento a ser executado é a extração dos formantes através do uso dos coeficientes cepstrais. Para isso, primeiro calcula-se o logaritmo do sinal resultante do banco de filtros e, após este passo, é feito o cálculo da transformada inversa de Fourier neste sinal. Como retratado na Seção 3.1, um sinal de fala pode ser modelado pelo modelo fonte-filtro definido pela Equação 3.1. O objetivo deste procedimento é extrair os formantes

da sentença estimando as caraterísticas do filtro (BENESTY; SONDHI; HUANG, 2008). Matematicamente, o cálculo dos coeficientes cepstrais transforma a Equação 3.1 em:

$$\hat{x}[n] = \hat{e}[n] + \hat{h}[n] \tag{3.20}$$

onde $\hat{x}[n]$, $\hat{e}[n]$ e $\hat{h}[n]$ são versões estimadas de $x[n]$, $e[n]$ e $h[n]$.

Este procedimento é um artifício matemático para transformar a resposta do sistema que modela o sinal de fala na soma do sinal que modela o filtro com o sinal que modela a fonte. Observe que esta operação transforma a convolução do modelo fonte-filtro na combinação linear entre os dois sinais. Pode-se então estimar as características do filtro que produz os formantes, partindo do princípio da superposição em sistemas lineares invariantes no tempo (SLIT). Neste princípio, a resposta de um sistema SLIT formado pela combinação linear de diferentes sinais pode ser representada pela resposta obtida pelo sistema a cada um dos sinais separados (LATHI, 2005).

Para realizar esta transformação explora-se a relação de dualidade entre a convolução e multiplicação de sinais nos domínios temporal e espectral. Tirando proveito do fato da convolução no domínio do tempo equivaler a uma multiplicação no domínio da frequência o modelo fonte-filtro no domínio da espectral pode ser definido (BENESTY; SONDHI; HUANG, 2008) por

$$X[k] = E[k]H[k], \tag{3.21}$$

a sua magnitude é

$$|X[k]| = |E[k]||H[k]| \tag{3.22}$$

e aplicando-se o logaritmo

$$log_{10}(|X[k]|) = log_{10}(|E[k]|) + log_{10}(|H[k]|). \tag{3.23}$$

Por fim, convertendo o sinal de volta ao domínio temporal têm-se a Equação 3.20.

Para a obtenção dos coeficientes MFCC após a conversão do sinal ao domínio cepstral, a literatura recomenda o uso da Transformada Discreta do Cosseno (*Discrete Cossine Transform* - DCT) (HUANG; ACERO; HON, 2001), (BENESTY; SONDHI; HUANG, 2008). O uso da DCT tem o objetivo de simplificar a separação dos formantes do sinal de fala, pois, esta operação caracteristicamente armazena os sinais de variação mais lenta em seus coeficientes de menor valor e os sinais de variação mais rápida em seus coeficientes de maior valor.

Como apresentado na Seção 3.1, os formantes são representados por sinais de variação mais lenta, logo, serão armazenados nos coeficientes de menor valor do vetor resultante após o cálculo da DCT, bastando realizar o descarte dos coeficientes de ordem

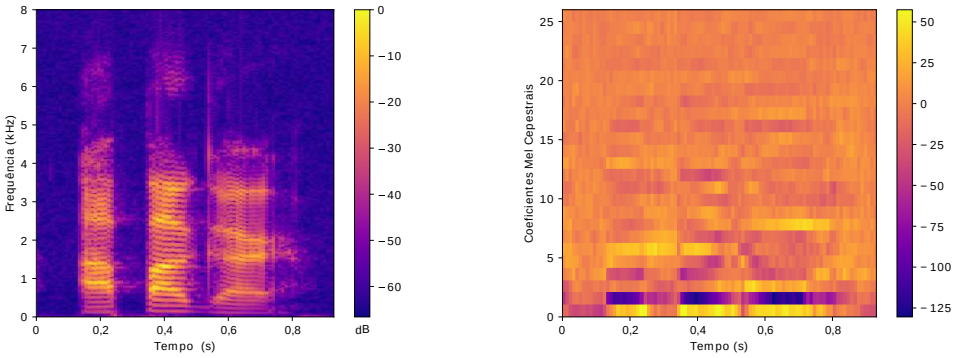
mais elevados para obter-se a separação dos formantes das características dependentes de locutor de um sinal de fala. Para calcular a DCT utiliza-se

$$c_i[n] = \sum_{m=0}^{M-1} 20log_{10}(S_i[m])cos\left(\pi n\left(\frac{m+\frac{1}{2}}{M}\right)\right), \tag{3.24}$$

onde M é o número de coeficientes cepstrais desejados. Na literatura recomenda-se a utilização do número de coeficientes entre 12 e 20, dependendo do número de elementos utilizados no banco de filtros (BENESTY; SONDHI; HUANG, 2008) (HUANG; ACERO; HON, 2001).

A Figura 19 ilustra o espectrograma do sinal de fala original (19a) e os coeficientes Mel-cepstrais (19b), respectivamente, obtidos após a extração de características do sinal de fala utilizando a técnica MFCC. Na Figura 19a, observa-se a existência do primeiro formante representado pelas estruturas periódicas de maior energia entre os instantes de tempo de 0,14 e 0,24 segundos, do segundo formante entre 0,33 e 0,48 segundos e o terceiro formante entre 0,54 e 0,72 segundos aproximadamente.

Figura 19 – Espectrograma e coeficientes Mel-Cepstrais da palavra “apagar”.



(a) Espectrograma. (b) Coeficientes Mel-Cepstrais.

Fonte: Autoria Propria.

Na Figura 19b, verifica-se a separação dos formantes no sinal de fala, visto que os elementos de maior energia estão alocados nos coeficientes de menor valor no espectrograma exibido. Desse modo, pode-se descartar os coeficientes de maior valor, de maneira a obter as informações necessárias para o reconhecimento da sentença pelo bloco de reconhecimento de padrões.

4 Reconhecimento de Padrões

Os sistemas de ASR são projetados com a função de reconhecer sinais de fala de entrada e convertê-los em texto, aplicando técnicas de reconhecimento de maneira eficiente e precisa (RABINER; JUANG, 1993). O bloco de reconhecimento de padrões é o bloco de um sistema de ASR que utiliza as informações fornecidas por todos os outros blocos do sistema para realizar esta tarefa.

De maneira geral, os reconhecedores de padrões de fala são divididos em grupos distintos de acordo com a técnica utilizada para o reconhecimento (MARTINS, 1997): reconhecedores por comparação de padrões e reconhecedores que usam inteligencia artificial empregando Redes Neurais Artificiais (*Artificial Neural Networks* - ANN). Os reconhecedores baseados em comparações de padrões, tradicionalmente obtinham melhores resultados quando comparados com reconhecedores que utilizam ANN (HUANG; ACERO; HON, 2001) (BENESTY; SONDHI; HUANG, 2008).

Entretanto, com a evolução e criação de novas técnicas de ANN e do aumento do poder computacional disponível nos dispositivos eletrônicos, nos dias de hoje, bons resultados vêm sendo obtidos com o emprego deste tipo de reconhecedor (DAHL et al., 2012). Atualmente, empresas como a Google realizam diversas pesquisas com sistemas que fazem uso desta abordagem, desenvolvendo sofisticados reconhecedores híbridos, que utilizam técnicas ANN e por comparações de padrões trabalhando em conjuto (GOOGLE INC., 2019a).

Quando empregado reconhecedores que utilizam comparação de padrões, o sistema é desenvolvido de maneira que possa ser treinado para reconhecer determinados padrões, a partir de exemplos das palavras a serem reconhecidas previamente armazenadas (RABINER; JUANG, 1993). Este bloco é implementado em duas etapas: treinamento e comparação de padrões para o reconhecimento.

Durante a fase de treinamento, é construído um modelo, a partir dos padrões apresentados pelos sinais de fala utilizados como referência, o qual será utilizado durante a etapa de comparação. Na etapa subsequente, compara-se os padrões do sinal de fala que deseja-se reconhecer com o modelo de referência com o intuito de classificar este sinal de modo adequado.

Na classe de reconhecedores por reconhecimento de padrões, podem-se ter dois tipos de distintos de abordagem: a primeira utiliza os modelos de referência previamente obtidos na fase de treinamento diretamente durante a etapa de reconhecimento, calculando a similaridade entre o sinal de fala original e os modelos de referência de maneira determinística. Estes reconhecedores de padrões são geralmente chamados de reconhecedores de

modelo (*templates*), uma vez que os modelos de referência utilizados são amostras de cada palavra que será reconhecida (RABINER; JUANG, 1993).

A segunda abordagem são reconhecedores de padrões que utilizam métodos estocásticos baseados em HMM para obter de forma probabilística os modelos de referência. O uso de HMM permite a criação de modelos baseados em distribuições probabilísticas discretas, semi contínuas e contínuas, durante a fase de treinamento. A escolha do tipo de modelo é condicionada ao tipo de sistema a ser implementado (MARTINS, 1997).

A escolha do método para o bloco de reconhecimento de padrões depende principalmente do tipo de sistema que pretende-se desenvolver. Os principais fatores que influenciam nesta decisão são: o tipo de fala, ou seja, se o sistema será desenvolvido para reconhecer palavras isoladas ou fala contínua, a dependência ou independência de locutor e o tamanho do vocabulário.

Para aplicações em dispositivos embarcados, geralmente são construídos sistemas que utilizam técnicas voltadas a detecção de palavras isoladas, especializados em reconhecer apenas comandos, devido as limitações de hardware geralmente encontradas nestes dispositivos. Neste tipo de situação, as técnicas que utilizam a abordagem por *templates*, que foram dominantes nas décadas de 70, 80 e início dos anos 90 (RABINER; JUANG, 1993) ainda são amplamente difundidas (ZAHARIA et al., 2010).

O principal motivo para o uso deste tipo de abordagem está relacionado a eficiência das técnicas utilizadas, que possuem uma alta taxa de acertos (RABINER; JUANG, 1993), (ABDULLA; CHOW; SIN, 2003), e são de menor complexidade, facilitando a implementação em hardware (ABDULLA; CHOW; SIN, 2003). Outro motivo para a utilização da abordagem por *templates*, é a maior facilidade para construção de modelos de referência durante a fase de treinamento pois utilizam-se os dados obtidos diretamente da fase de extração de características do sinal, agilizando o processo.

Entretanto, o uso deste tipo de reconhecedor limita os sistemas a um vocabulário reduzido de palavras pois, estes tem como característica a comparação direta entre o sinal de fala desconhecido e os *templates* de referência. Com o aumento do vocabulário, aumenta-se o número de comparações e palavras necessárias para o reconhecimento de maneira correta, além da quantidade de armazenamento exigida do hardware, que precisará armazenar mais dados, o que pode levar a problemas de desempenho. Das técnicas para a construção de reconhecedores de padrões baseados em *templates* na literatura se destaca o uso do DTW (RABINER; JUANG, 1993), (BENESTY; SONDHI; HUANG, 2008), (HUANG; ACERO; HON, 2001), que será apresentado a seguir.

4.1 DTW

O DTW é uma técnica que obteve considerável sucesso para o reconhecimento de fala, principalmente em sistemas especializados em reconhecer palavras isoladas (RABINER; JUANG, 1993). Introduzida por (BELLMAN; KALABA, 1959) no fim da década de 50, esta técnica foi criada para realizar o alinhamento temporal, através de uma transformação não linear, e medir a similaridade entre duas séries temporais. No trabalho de (SAKOE; CHIBA, 1978), esta técnica foi utilizada pela primeira vez para medir a similaridade entre dois sinais de fala, constituindo o bloco de reconhecimento de padrões de um sistema de ASR de palavras isoladas. Após a aplicação em sistemas de ASR, esta técnica ganhou notoriedade e passou a ser usada em diversos outros problemas tais como: reconhecimento digital de assinaturas, reconhecimento de gestos utilizados em língua de sinais, mineração de dados, processamento de áudio, sequenciamento e alinhamento de proteínas (SILVA, 2017).

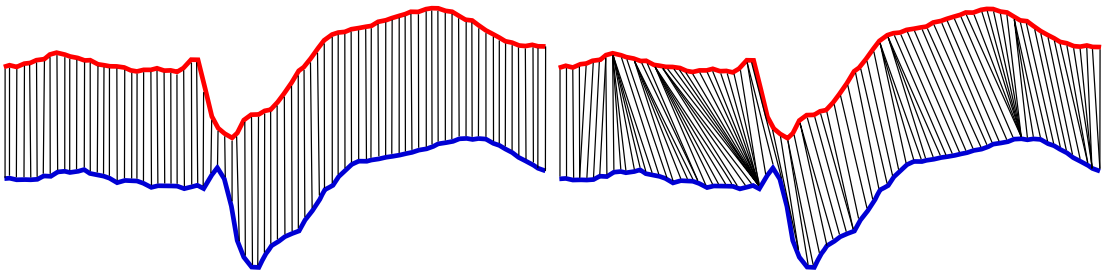
O DTW compara, por meio de uma medida de similaridade, o padrão de fala que deseja-se reconhecer e os *templates* utilizados como referência, minimizando a interferência das flutuações temporais causadas pelas variações da pronúncia de uma palavra por um indivíduo (SAKOE; CHIBA, 1978). Assim, o algoritmo DTW reconhece o sinal em questão de acordo com o sinal de referência cujo padrão resulta na menor distância. O resultado obtido pode ser interpretado como: quanto menor for esta medida, mais similar são os padrões de fala.

Antes da adoção do DTW, o uso técnicas de tradicionais para medir a similaridade entre dois padrões de sinais de fala, como a distância euclidiana, provaram-se pouco efetivas pois, são sensíveis a variação temporal existente entre os padrões (SAKOE; CHIBA, 1978). A Figura 20 ilustra o processo de medição de similaridade entre dois padrões utilizando a distância euclidiana (Figura 20a) e usando DTW (Figura 20b).

Observa-se na Figura 20 a semelhança entre os padrões e um ligeiro deslocamento temporal entre ambas. Na Figura 20a, observa-se que o uso da distância euclidiana não consegue distinguir o deslocamento temporal entre os padrões em análise, que resulta em comparações de padrões pouco precisas.

A Figura 20b ilustra o processo de comparação de padrões feita pelo DTW. É possível observar, nos pontos onde há deslocamentos temporais entre os padrões, que há o mapeamento de diversos pontos de um dos padrões em um único ponto no outro padrão. Este mapeamento é realizado pelo DTW para eliminar as distorções temporais entre o dois sinais em análise, permitindo o alinhamento temporal e resultando em comparações mais precisas.

Figura 20 – Comparação entre os alinhamentos realizados pela distância euclidiana e pelo DTW.



(a) Distancia euclidiana. (b) DTW.

Fonte: Adaptado de (SILVA, 2017).

4.1.1 Formulação do DTW

Após a etapa de extração das características realizada pelo bloco de processamento de sinais, um sinal de fala pode ser caracterizado por uma sequência de vetores de características de dimensão igual ao número coeficientes Mel-cepstrais, organizados na forma

$$\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \ldots \ \mathbf{a}_i \ \ldots \ \mathbf{a}_I] \tag{4.1}$$

onde $\mathbf{A}$ é matriz que armazena o conjunto de vetores $\mathbf{a}_i$, formando o padrão de fala, I é o número total de vetores que compõem o sinal de fala e i é o índice de um determinado vetor do conjunto.

Considerando o problema de comparar dois padrões de fala que podem ou não ter o mesmo número de vetores, pode-se reescrever 4.1 como:

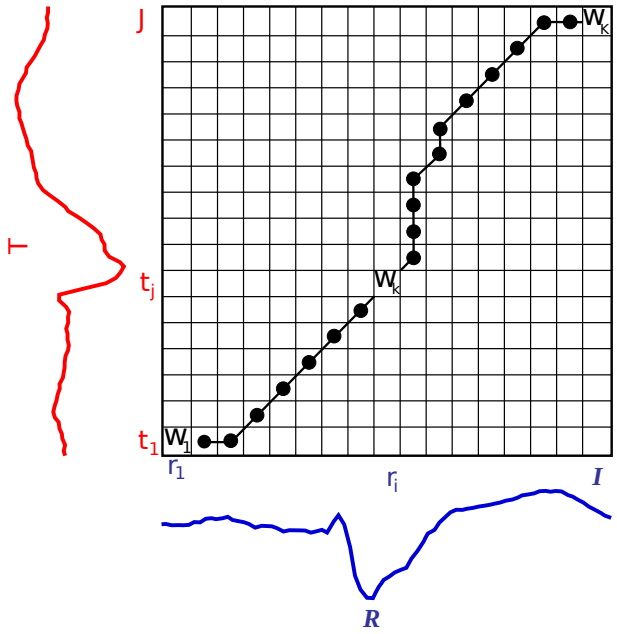
$$\begin{aligned} \mathbf{R} &= [\mathbf{r}_1 \ \mathbf{r}_2 \ \ldots \ \mathbf{r}_i \ \ldots \ \mathbf{r}_I] \\ \mathbf{T} &= [\mathbf{t}_1 \ \mathbf{t}_2 \ \ldots \ \mathbf{t}_j \ \ldots \ \mathbf{t}_J] \end{aligned} \tag{4.2}$$

onde $\mathbf{R}$ é a matriz que representa o padrão de referência e $\mathbf{T}$ a matriz que representa o padrão de teste (isto é, o padrão que deseja-se reconhecer).

Para o desenvolvimento do método, considere que $\mathbf{R}$ e $\mathbf{T}$ são séries temporais unidimensionais R e T compostas apenas por pontos escalares r_i e t_j. Dessa forma, podemos obter uma matriz de custo $\mathbf{D}$, mostrada na Figura 21, de dimensões $I \times J$ onde os elementos dessa matriz são as distâncias locais $d(r_i, t_j)$ entre os pontos r_i e t_j pertencentes as sequências R e T, respectivamente. Para a obtenção de $d(r_i, t_j)$, a literatura (MYERS; RABINER, 1980) (SAKOE; CHIBA, 1978) comumente utiliza a distância euclidiana, dada por

$$d(r_i, t_j) = |r_i - t_j| \quad \text{para} \quad i = 1 : I, \quad \text{para} \quad j = 1 : J. \tag{4.3}$$

Figura 21 – Um exemplo da função de distorção.



Fonte: Autoria Própria.

Para alinhar temporalmente as duas séries, pode-se traçar um caminho em **D**, iniciando em $d(1,1)$ e terminando em $d(I,J)$, mapeando todos os elementos de **D** que pertencem a este caminho. A Figura 21 ilustra esse processo, onde cada elemento pertencente ao caminho traçado é representado pela sequência de pontos $w_1, \ldots, w_k, \ldots, w_K$. Na literatura, esse caminho é chamado função de distorção (SAKOE; CHIBA, 1978) ou função de alinhamento (MYERS; RABINER, 1980) e é definida como

$$\mathbf{W} = w_1, w_2, \ldots, w_k, \ldots, w_K \tag{4.4}$$

onde **W** é a função de distorção, w_k é o k-ésimo elemento pertencente a **W** que possui coordenadas

$$w_k = (i,j)_k \tag{4.5}$$

e w_K representa o último elemento de **W**.

Para encontrar a medida de similaridade produzida pelo DTW, deve-se encontrar na matriz **D** o melhor alinhamento temporal entre R e T, ou seja, deve-se realizar o mapeamento da matriz **D**, encontrando todas as funções de distorção, até obter a função de distorção ótima, onde a soma da distância local de cada um dos elementos pertencentes a esta função seja a menor possível, produzindo a medida de similaridade onde as séries estão melhor alinhadas temporalmente. Para encontrar as funções de distorção em **D**, algumas restrições ao processo de busca devem ser impostas de maneira a garantir que o processo de alinhamento ocorra corretamente (SAKOE; CHIBA, 1978). Estas funções devem obedecer as seguintes restrições:

- **Restrição de Fronteira:** $w_1 = d(1,1)$ e $w_K = d(I,J)$, ou seja, a função de distorção deve iniciar no primeiro elemento de $\mathbf{D}$ e terminar no último. Esta restrição garante que as sequências serão mapeadas completamente, evitando que uma delas seja mapeada parcialmente;

- **Restrição de continuidade:** dado $w_k = d(i,j)$ e $w_{k-1} = d(i',j')$ então, $i - i' \leq 1$ e $j - j' \leq 1$. Esta restrição permite que sejam mapeados apenas elementos adjacentes entre si em $\mathbf{D}$, evitando que existam saltos no tempo;

- **Restrição de monotonicidade:** dado $w_k = d(i,j)$ e $w_{k-1} = d(i',j')$ então, $i - i' \geq 0$ e $j - j' \geq 0$. Esta restrição garante que a ordem de busca dos elementos em $\mathbf{D}$ será preservada, evitando que as funções de distorção retrocedam no tempo.

A medida de similaridade DTW pode então ser encontrada somando todos os elementos que compõe a função de distorção que atendam estas restrições e possua o menor valor, ou seja, a função de distorção ótima, dada por

$$DTW(R,T) = min\left[\frac{\sum_{k=1}^{K} w_k}{C}\right] \tag{4.6}$$

onde C é o coeficiente de normalização dado por $I + J$.

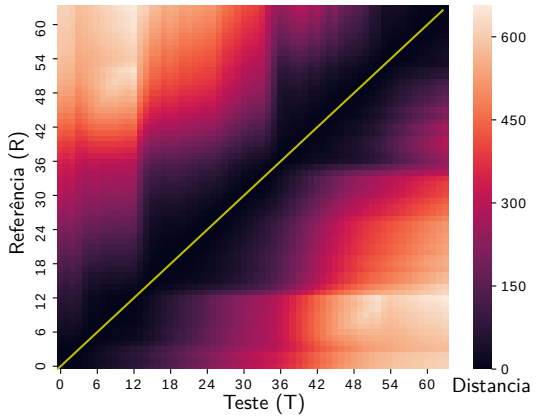
Uma análise interessante do funcionamento do DTW pode ser obtida observando a função de distorção com o melhor alinhamento temporal. Quando analisa-se duas séries iguais, ou seja, quando não há diferenças temporais entre as séries comparadas pelo método, o caminho feito pela função de distorção ótima vai coincidir com a diagonal da matriz $\mathbf{D}$. Isto é facilmente observável a partir da Equação 4.3, onde, se as duas sequências em análise forem iguais, teremos a distância euclidiana nula (zero), nos pontos onde $r_i = t_j$.

Por outro lado, a medida que as séries se diferenciam o caminho percorrido se afasta da diagonal. Intuitivamente, quando as séries são semelhantes, mas com deslocamentos temporais, o caminho percorrido pela função de distorção ótima tende a se manter próximo a diagonal de $\mathbf{D}$. A Figura 22 ilustra o comportamento da função de distorção temporal ótima para as situações onde se analisa o mesmo padrão (Figura 22a), dois padrões pertencentes a mesma classe (Figura 22b) e padrões diferentes (Figura 22c) calculados a partir do algoritmo do DTW, que será explicado na próxima seção.

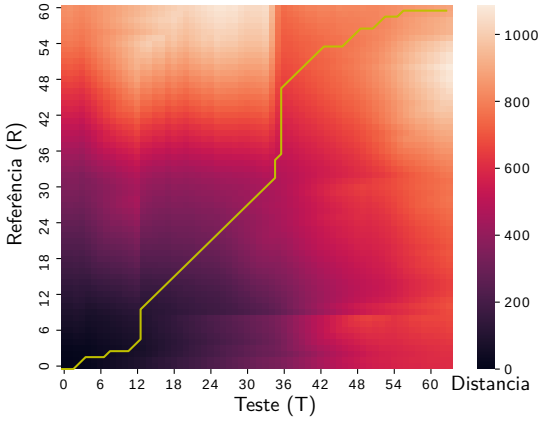
4.1.2 Algoritmo do DTW

O DTW em essência é uma técnica de alta complexidade computacional pois, existe um valor exponencial de caminhos dentro da matriz $\mathbf{D}$ que atendem aos critérios de restrição (RABINER; JUANG, 1993). Para encontrar a função de distorção ótima, todos esses caminhos devem ser checados até se encontrar o caminho de menor custo, o que, dependendo do tamanho das sequências, pode-se tornar impraticável. Contudo, é possível

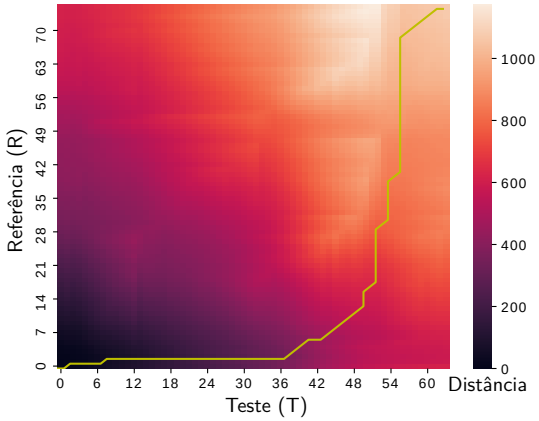
Figura 22 – Exemplo ilustrando o comportamento da função de distorção.



(a) Mesmo padrão.



(b) padrões de mesma classe.



(c) Padrões diferentes.

Fonte: Autoria Própria.

implementar um algoritmo via programação dinâmica de maneira computacionalmente eficiente. Para isso, utiliza-se a equação de recorrência (SAKOE; CHIBA, 1978), (MYERS; RABINER, 1980), (SILVA, 2017):

$$g(i,j) = d(r_i, t_j) + min[g(i-1,j), g(i,j-1), g(i-1,j-1)]. \tag{4.7}$$

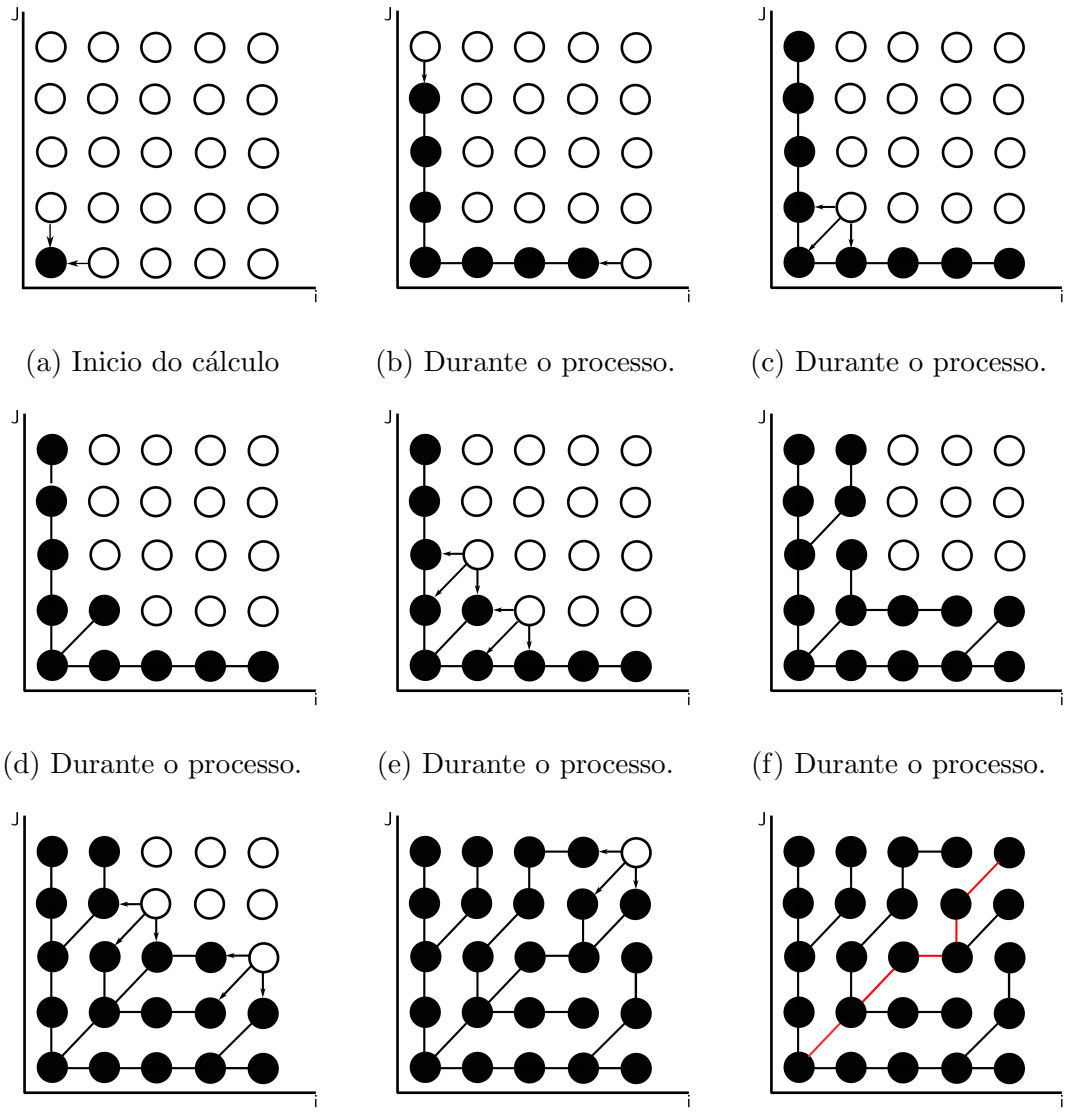
Na prática, ao se implementar a Equação 4.7, cria-se uma matriz de custo acumulado $\mathbf{G}$ de dimensão $I \times J$ que irá calcular para todos os elementos da matriz $\mathbf{G}$, de maneira iterativa, o menor custo para se sair do primeiro elemento da matriz $\mathbf{G}$ e ir até o elemento em análise. Esse procedimento será repetido progressivamente para cada elemento em $\mathbf{G}$, até que se obtenha o menor custo acumulado para todos os elementos de $\mathbf{G}$. Ao término de execução do algoritmo, o último elemento da matriz $\mathbf{G}$ armazenará a distância DTW.

É importante observar que, como é necessário calcular o custo acumulado para

todos os elementos da matriz $\mathbf{G}$, esta implementação do algoritmo DTW tem complexidade computacional igual a $\mathcal{O}(IJ)$. Para o caso particular onde $\mathbf{G}$ é uma matriz quadrada, isto é, quando $I = J$, este método terá complexidade computacional $\mathcal{O}(I^2)$.

Observe que nesta implementação do DTW, durante o cálculo da distância acumulada entre as amostras de R e T não se utiliza a função de distorção ótima diretamente. Contudo, uma vez que $\mathbf{G}$ está pronta, pode-se encontrar a função de distorção usando um algoritmo de refinamento em $\mathbf{G}$. Chamado de *backtracking* (SILVA, 2017), este algoritmo é usado para mapear todos os elementos de $\mathbf{G}$ que fazem parte do menor caminho entre $g(I, J)$ e $g(1, 1)$. A Figura 23 ilustra o processo iterativo realizado durante a implementação do DTW.

Figura 23 – Exemplo do processo iterativo realizado no cálculo do DTW.



(a) Inicio do cálculo (b) Durante o processo. (c) Durante o processo.

(d) Durante o processo. (e) Durante o processo. (f) Durante o processo.

(g) Durante o processo. (h) Durante o processo. (i) Cálculo finalizado e W.

Fonte: Autoria Própria.

4.1.3 Modificações do DTW

Na literatura existem diversas técnicas criadas com a finalidade de melhorar o desempenho do algoritmo do DTW (SAKOE; CHIBA, 1978),(MYERS; RABINER, 1980) e (ITAKURA, 1975). Nestes trabalhos, são propostas técnicas que visam reduzir do tempo de execução do DTW e aumentar a sua precisão. Nesta seção, serão tratadas as modificações implementadas neste trabalho que cumpram os dois objetivos.

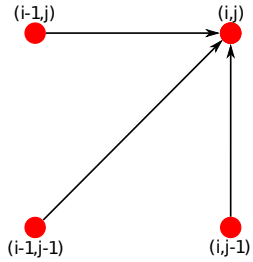
4.1.3.1 Restrições Locais

Como mencionado na Seção 4.1.2, o método DTW cria uma matriz $\mathbf{G}$ com a distância acumulada, obtida a partir da Equação 4.7. Nesta equação, o termo $min[g(i-1,j), g(i,j-1), g(i-1,j-1)]$ determina em quais células adjacentes ao elemento que está sendo calculado será feita a busca pelo menor valor de custo acumulado. Este padrão de busca é resultante da equação de recorrência mais simples que pode-se obter respeitando as restrições de monotonicidade e continuidade (MYERS; RABINER, 1980). Contudo, este padrão não é único e, a partir das restrições de monotonicidade e continuidade, pode-se derivar diversos outros padrões de busca. Na literatura este padrão de busca é chamado de restrição local (MYERS; RABINER, 1980), ou restrição de inclinação (SAKOE; CHIBA, 1978).

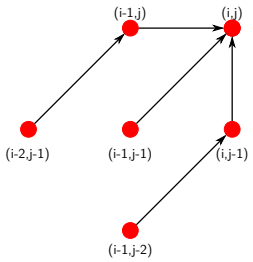
As restrições locais são geralmente utilizadas para melhorar a precisão do DTW (SAKOE; CHIBA, 1978). O principal objetivo do emprego deste tipo de restrição é encontrar o padrão de buscas que melhor modele as flutuações temporais entre as sequencias em análise. Em algumas situações, o uso de um certo tipo de restrição local não consegue modelar corretamente as flutuações temporais entre as duas sequências em análise, e pode ocorrer o mapeamento de segmentos muito curtos de uma sequência em segmentos muito longos de outra. No uso do DTW para o reconhecimento de fala, este fenômeno pode ocorrer na transição de consoantes muito curtas para vogais durante a pronuncia de uma palavra(SAKOE; CHIBA, 1978).

Na literatura, existem alguns trabalhos como os apresentados por (SAKOE; CHIBA, 1978) e (MYERS; RABINER, 1980), que propõem diversas modificações para as restrições locais. Neste trabalho, além da formulação clássica de restrição local, serão também analisadas as restrições locais *Type I*, *Type II* apresentadas em (MYERS; RABINER, 1980), e o tipo $\frac{P}{2}$ apresentado em (SAKOE; CHIBA, 1978). As restrições locais *Type I* e *Type II* foram escolhidas por apresentarem os melhores desempenhos nos testes realizados em (MYERS; RABINER, 1980). Já a restrição do tipo $\frac{P}{2}$ foi escolhida por ser um tipo de restrição que contém em seu padrão de deslocamento todas as outras restrições escolhidas. A Figura 24 ilustra os padrões usados. As equações de recorrência para as restrições locais *Type I*, *Type II* e $\frac{P}{2}$ são implementadas pelas Equações 4.8, 4.9 e 4.10 respectivamente.

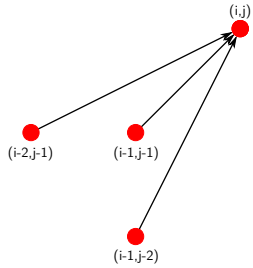
Figura 24 – Restrições Locais usadas.



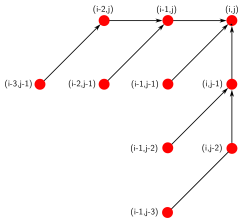
(a) Restrição clássica.



(b) Restrição *Type I*.



(c) Restrição *TypeII*.



(d) Restrição $\frac{P}{2}$.

Fonte: Autoria Própria.

$$g(i,j) = d(r_i, t_j) + min \begin{bmatrix} g(i-2, j-1) + d(r_i - 1, t_j) \\ g(i-1, j-1) \\ g(i-1, j-2) + d(r_i, t_j - 1) \end{bmatrix} \tag{4.8}$$

$$g(i,j) = d(r_i, t_j) + min[g(i-1, j-2), g(i-2, j-1), g(i-1, j-1)] \tag{4.9}$$

$$g(i,j) = d(r_i, t_j) + min \begin{bmatrix} g(i-3, j-1) + d(r_i - 1, t_j) + d(r_i - 2, t_j) \\ g(i-2, j-1) + d(r_i - 1, t_j) \\ g(i-1, j-1) \\ g(i-1, j-2) + d(r_i, t_j - 1) \\ g(i-1, j-3) + d(r_i, t_j - 1) + d(r_i, t_j - 2) \end{bmatrix} \tag{4.10}$$

4.1.3.2 Restrições Globais

As restrições globais, também conhecidas na literatura como bandas (SAKOE; CHIBA, 1978), ou funções janela (MYERS; RABINER, 1980) (SILVA, 2017), pertencem a classe de restrições que definem o espaço de busca dentro da matriz **G**. Esta classe tem por objetivo aumentar a velocidade de cálculo do DTW, limitando o número de elementos de **G** em que é realizado o processo de cálculo da distância acumulada.

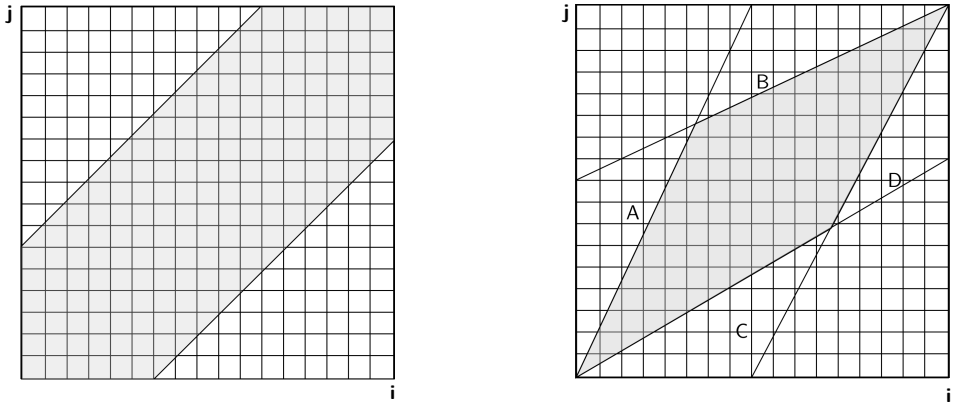
Na literatura existem alguns tipos de restrições globais propostos, contudo, a banda de Sakoe-Chiba (SAKOE; CHIBA, 1978) e o paralelogramo de Itakura (ITAKURA, 1975), ilustrados nas Figuras 25a e 25b, respectivamente, são as mais utilizadas (SILVA, 2017). A matriz **G** que utilize as restrições globais do tipo banda de Sakoe-Chiba ou paralelogramo de Itakura, pode ser obtida por meio das Equações 4.11 e 4.12, respectivamente.

$$j = \begin{cases} i + r & \text{para limite superior} \\ i - r & \text{para limite inferior} \end{cases} \tag{4.11}$$

onde r é o tamanho da janela e

$$\begin{cases} \text{Reta A:} & j = 2(i-1) + 1 \\ \text{Reta B:} & j = \left(\frac{i-I}{2}\right) + J \\ \text{Reta C:} & j = \left(\frac{i-1}{2}\right) + 1 \\ \text{Reta D:} & j = 2(i-I) + J \end{cases} \tag{4.12}$$

Figura 25 – Restrições globais.



(a) Banda de Sakoe-Chiba. (b) Paralelogramo de Itakura.

O uso de restrições globais ajuda a acelerar o tempo de execução do DTW, limitando a área de buscas permitida em **G**. Contudo, o uso da banda de Sakoe-Chiba como restrição global está condicionado a um estudo cauteloso das séries a serem analisadas de modo determinar o tamanho da janela de restrição. Se esta for configurada de maneira muito rígida, a janela de restrição ficará próxima a linha diagonal de **G**, e séries semelhantes podem não ser analisados corretamente. Para o paralelogramo de Itakura este problema não é encontrado pois, como mostrado pela Equação 4.12, a determinação das dimensões do paralelogramo é diretamente ligado ao tamanho das séries em análise.

5 Implementação

Nos capítulos anteriores, foram realizadas a revisão bibliográfica sobre sistemas de reconhecimento de fala e das técnicas empregadas em seu desenvolvimento, enfatizando as técnicas utilizadas para criação do sistema de ASR proposto neste trabalho. Entretanto, ainda não foram abordados alguns pontos relacionados implementação deste sistema. Neste capítulo, será apresentado os recursos computacionais utilizados para a implementação deste trabalho, em seguida feita uma breve revisão sobre a plataforma utilizada para os experimentos, o micro computador *Raspberry Pi 3 Model B+*.

Na sequência, serão apresentados os diagramas de bloco que demonstram como foi construído o reconhecedor de padrões utilizado para avaliar o desempenho da DTW e do bloco de processamento de sinais. Logo após, será apresentado o diagrama de blocos do sistema construído para realização da etapa de treinamento empregando o método de seleção de melhores *templates*. Por fim, será apresentada a metodologia proposta pelo autor para a determinação de um limiar de identificação de palavra, implementado com o objetivo de solucionar alguns problemas decorrentes do uso do DTW em reconhecedores de padrões.

5.1 Recursos Computacionais

Para a implementação da parte prática deste trabalho, foi utilizada a linguagem de programação interpretada de propósito geral Python®. As técnicas e sistemas implementados foram primeiramente desenvolvidas e testadas em um *notebook* pessoal com processador Intel *Core* I5, com 8GB de memória RAM, sistema operacional Linux Mint 19 e utilização da distribuição Anaconda® Python® para implementação e validação dos métodos empregados. Para a aquisição dos sinais de fala utilizou-se um microfone direcional do tipo *headset* e uma interface genérica de áudio USB.

Esta interface foi empregada com o objetivo de usar o mesmo hardware que será empregado na plataforma embarcada, visto que, normalmente, microcomputadores de baixo custo não possuem interface de áudio integrada. Para validação do sistema implementado os arquivos de áudio foram pré-gravados com duração de 2 segundos por palavra usada no sistema, possuem a extensão .wav, frequência de amostragem de 16 kHz, com 16 bits de quantização utilizando codificação por modulação de pulso (*Pulse Coding Modulation* - PCM) e um canal de áudio.

Após o teste e validação dos métodos, foi feita a migração para a plataforma de testes, o microcomputador *Raspberry Pi 3 Model B+* que será melhor descrito na Seção

5.2. O sistema operacional utilizado foi o Linux Raspian e a distribuição Berryconda Python®. A substituição de distribuição foi necessária pois, a distribuição Anaconda ® Python® não está disponível para sistemas que utilizam arquiteturas ARM. O Berryconda Python® é uma das alternativas disponíveis para fazer a substituição desta distribuição em dispositivos que possuem este tipo de arquitetura.

5.2 *Raspberry Pi 3 Model B+*

O *Raspberry Pi* é um microcomputador de custo reduzido, com todas as funções de um computador comum, que pode ser utilizado como dispositivo embarcado, e possuí tamanho similar a um cartão de crédito produzido pela fundação inglesa sem fins lucrativos *Raspberry Pi fundation* (RASPBERRY PI FOUNDATION., 2019b). O *Raspberry Pi* ainda é capaz de interagir com diversos dispositivos e sensores, e é utilizado em inúmeros projetos eletrônicos como: criação de estações meteorológicas, fliperamas, máquinas musicais, entre outras (RASPBERRY PI FOUNDATION., 2019b).

Disponibilizado em diferentes modelos, o *Raspberry Pi* apresenta unidades centrais de processamento (*Central Processing Unit* - CPU) com arquitetura RISC fornecida pela *Advanced RISC Machine* - ARM, conexões USB, Ethernet, HDMI, RCA e em alguns modelos Wi-Fi e *Bluetooth*. Para seu funcionamento, o *Raspberry Pi* necessita de um cartão de memória do tipo *Micro SD* para instalação do sistema operacional, onde a capacidade de armazenamento do cartão depende do sistema operacional a ser instalado.

O *Raspberry Pi* utilizado neste trabalho foi configurado usando o Sistema Operacional *Raspian*, uma versão customizada e otimizada da distribuição Linux Debian. Este sistema disponibiliza uma série de ferramentas e pacotes para configuração e desenvolvimento de aplicativos, em linguagens como C, C++ e Python®, facilitando a migração dos sistemas desenvolvidos no PC para o embarcado. Neste trabalho, foi utilizado o modelo *Raspberry Pi 3 Model B+*, que tem suas características compiladas na Tabela 1 (RASPBERRY PI FOUNDATION., 2019a). A Figura 26 ilustra o modelo utilizado com alguns de seus componentes identificados, já a Figura 27 exibe o exemplar utilizado neste trabalho.

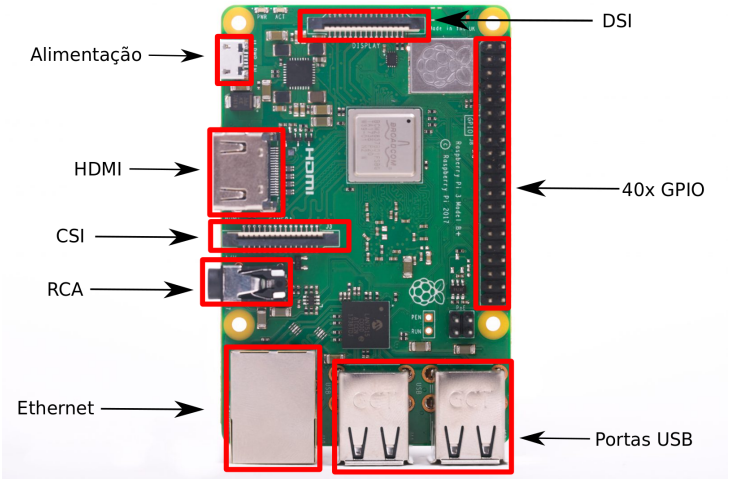
5.3 Arquiteturas de Reconhecimento de Padrões e Treinamento

O sistema implementado para avaliação de desempenho dos reconhecedores de padrões usando a técnica DTW tem sua arquitetura demonstrada pela área pontilhada na Figura 3. Para realização dos experimentos, gravou-se previamente diversas elocuções das palavras que serão utilizadas nestes testes. A implementação realizada neste trabalho seguirá a seguinte ordem:

Tabela 1 – Características do *Raspberry Pi 3 Model B+*

CPU	*Quad Core* Broadcom BCM2837B0, Cortex-A53 @ 1,4 GHz
Memória RAM	1 GB LPDDR2
Rede *Ethernet*	10/100 Mbps *Ethernet* com suporte a POE
Rede *Wi-Fi*	2,4 GHz e 5 GHz IEEE 802.11.b/g/n/ac
Rede *Bluetooth*	4.2 *Bluetooth* e BLE
Conexão HDMI	Conector padrão
Conexão GPIO	40 pinos
USB	4 USB 2.0 com suporte a Rede Gigabit *Ethernet*
Alimentação	Micro USB 5V/2,5A
Conexão SD	Micro SD
Outras Conexões	RCA, DSI e CSI para conexão de *displays* e câmeras

Figura 26 – *Raspberry Pi 3 Model B+*.



Fonte: Adaptado de (RASPBERRY PI FOUNDATION., 2019a).

Figura 27 – Exemplar do *Raspberry Pi 3 Model B+* utilizado.

Fonte: Autoria Própria.

1. Segmentação e janelamento;
2. Remoção dos trechos de silêncio utilizando a técnica VAD *endpoint detection*;
3. Divisão dos sinais de fala previamente gravados em dois grupos mutualmente excludentes. A proporção utilizada para dividir os grupos foi: dois terços para o grupo que de sinais de fala que serão utilizados durante a etapa de treino para construir os *templates* de referência e um terço para o grupo de sinais de fala de testes;
4. extração de características do sinais de de fala pertencentes ao grupo de treino utilizando a técnica MFCC;
5. Treinamento do sistema;
6. Reconhecimento utilizando cada um dos sinais pertencentes ao grupo de testes;

a seguir, será descrito como foi elaborado o sistema de reconhecimento de padrões.

5.3.1 Reconhecedor de Padrões Utilizando o DTW

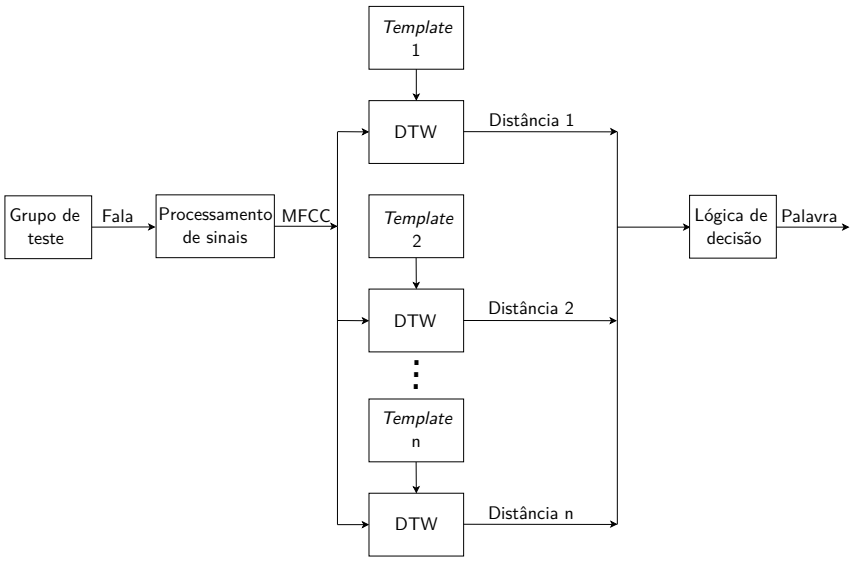
A fase de reconhecimento de padrões aplicando o DTW é ilustrada pela Figura 28. Neste reconhecedor, é escolhido um sinal no conjunto de sinais de teste de maneira aleatória, para dar início ao processo de reconhecimento da palavra, enviando-o ao bloco de processamento de sinais. Com os procedimentos neste bloco finalizados, realiza-se a comparação de cada um dos sinais de teste com todos os *templates* existentes no bloco de modelos acústicos, utilizando a técnica DTW. As distâncias obtidas em cada uma das comparações são armazenadas no bloco de lógica de decisão. Por fim, o sistema estima como a palavra a ser reconhecida o *template* que obtiver o menor valor de distância em relação ao sinal de entrada.

5.3.2 Seleção dos Melhores *Templates*

A principal desvantagem de reconhecedores por comparações de padrões, classe em que o DTW se encontra, é número de sinais de fala utilizados como referência necessários para cada palavra a ser reconhecida. Neste tipo de reconhecedor, a taxa de reconhecimento é diretamente relacionada ao número de sinais de referência utilizados. Esta relação existe devido as variações existentes no sinal de fala, uma vez que é praticamente impossível para um indivíduo pronunciar uma mesma palavra diversas vezes da mesma maneira. Outros fatores como estado emocional e físico também influenciam no jeito que uma palavra é pronunciada, dificultando o processo de reconhecimento.

Para contornar este problema, os reconhecedores de padrões utilizam várias amostras da pronúncia de cada palavra que o sistema pode reconhecer, com o objetivo possuir em

Figura 28 – Sistema de reconhecimento de padrões.



Fonte: Autoria Própria.

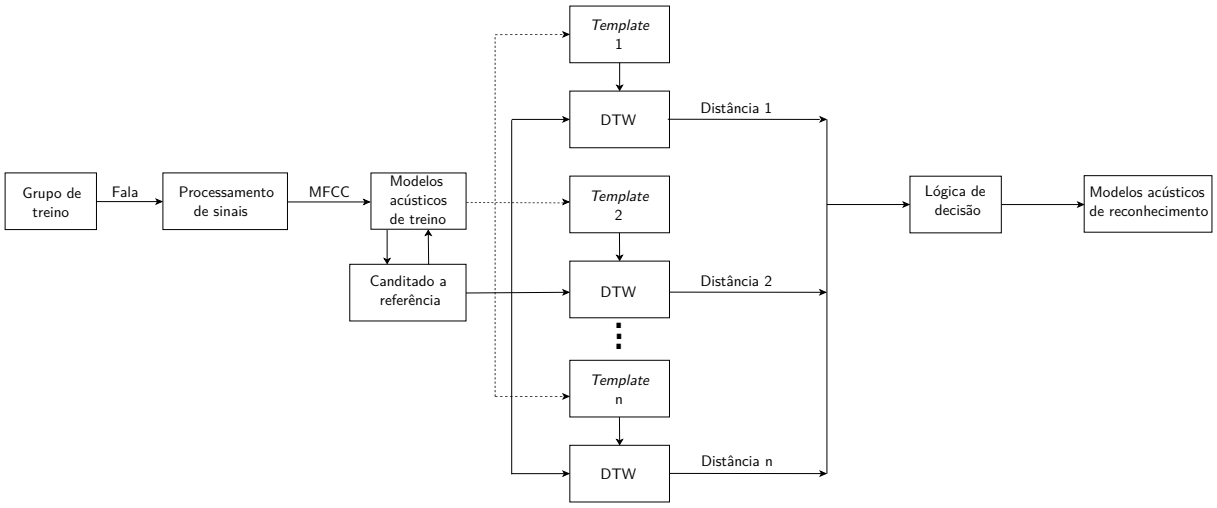
seu banco de modelos acústicos o máximo de variações possíveis da pronuncia de cada uma dessas palavra. Contudo, dependendo do número de palavras que serão utilizadas e das características do hardware onde este sistema será implementado, o reconhecimento utilizando o DTW pode se tornar inviável.

Em (RABINER; JUANG, 1993) é sugerido um método que visa diminuir o número de *templates* usados no reconhecedor por classe de palavras. Nesta técnica, é implementada durante a etapa de treinamento a seleção dos *templates* que melhor representam cada uma das palavras a serem reconhecidas. O objetivo desta técnica é reduzir o número de *templates* de referência, encontrando quais são os *templates* utilizados para treinamento que possuem as melhores características em relação a variação da pronúncia de um sinal de fala. A Figura 29 ilustra o processo para implementação da etapa de treinamento para o sistema de ASR que utiliza o método de seleção de melhores *templates*.

Primeiro, os sinais de fala previamente gravados são divididos de maneira análoga a apresentada anteriormente. Em seguida, separam-se os sinais de fala no grupo de treinamento em classes menores, onde cada classe possuí apenas sinais de fala da mesma palavra. Finalizada esta etapa, os sinais da mesma classe são encaminhados para o sistema de treinamento, onde são processados no bloco de processamento de sinais e armazenados no bloco de modelos acústicos. Em seguida, escolhe-se de maneira aleatória um dos sinais neste bloco para ser o sinal candidato a referência. O banco de modelos acústicos é então atualizado, removendo o sinal escolhido como candidato a referência.

Logo após, realiza-se a comparação utilizando o DTW entre os *templates* armazenados no banco de modelos acústicos e o sinal escolhido como candidato a referência. As distâncias obtidas em cada uma das comparações são armazenadas, e o processo é

Figura 29 – Sistema de treinamento utilizando a seleção de melhores *templates*.



Fonte: Autoria Própria.

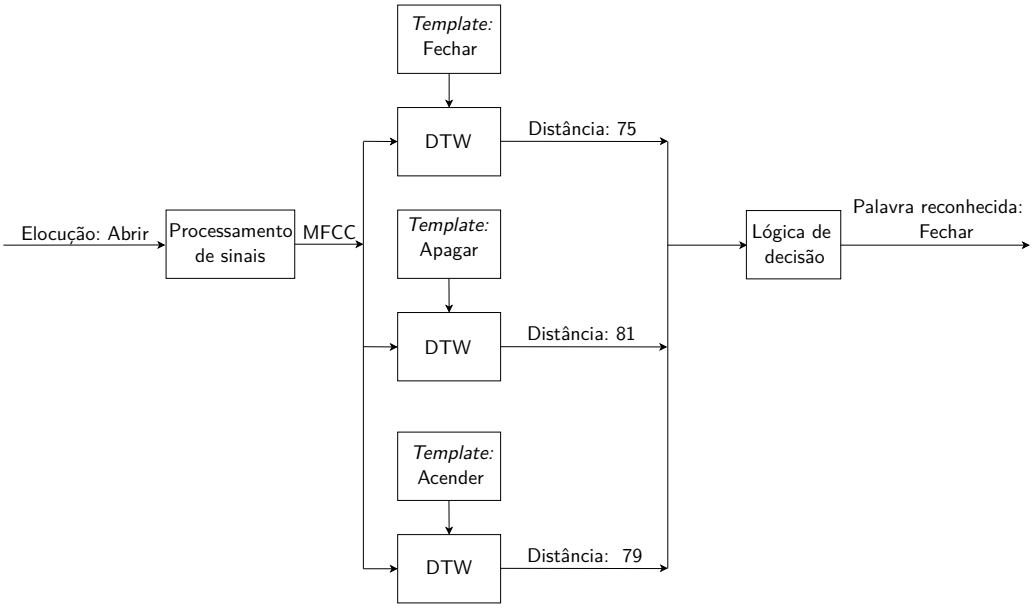
repetido até que todos os *templates* pertencentes a classe de palavra em treinamento sejam testados como candidatos a referência. Na sequência, o bloco de decisão seleciona os *templates* que possuírem a menor distância em relação aos outros modelos de seu grupo, para compor o bloco de modelos acústicos que serão utilizados como referência na etapa de reconhecimento. Este processo é repetido para todas as classes de palavras existentes no reconhecedor de padrões. Por fim, é realizada a etapa de reconhecimento, de maneira análoga a discutida anteriormente.

5.3.3 Limiar de Identificação de Palavra Proposto

Um dos principais problemas encontrados na implementação de reconhecedores de padrões que utilizam a menor distância obtida pelo DTW para determinar a palavra reconhecida, é qualificar se a palavra reconhecida está realmente correta. De modo geral, este tipo de reconhecedor de padrões sempre irá se decidir como palavra a ser reconhecida o *template* que possuir a menor distância calculada pelo DTW, mesmo quando a palavra pronunciada pelo usuário não pertencer ao conjunto de palavras existentes no sistema de ASR, gerando um erro de reconhecimento. Este problema pode ser observado na Figura 30.

Para contornar este problema, é necessário que seja definido um valor que estabeleça qual é a maior distância mínima obtida após o uso do DTW, para qual garanta-se o sucesso da etapa de reconhecimento. Entretanto, o problema para definição deste valor é a falta de conhecimento à priori dos valores de distância obtidas para cada palavra reconhecida corretamente, o que impossibilita a determinação deste valor de maneira que não seja empírica. Neste trabalho, para o desígnio deste valor, chamada de limiar de identificação de palavra, foi realizada a análise das estatísticas das distâncias mínimas obtidas pelo

Figura 30 – Exemplo de erro de reconhecimento utilizando somente a distância mínima obtida pelo DTW.



Fonte: Autoria Própria.

DTW após a realização de um teste onde repetiu-se diversas vezes todas as palavras que o sistema pode reconhecer. A partir destas estatísticas determinou-se então qual é o valor de distância mínima que será utilizado no limiar.

6 Resultados e Discussões

Este capítulo irá apresentar os resultados e as discussões referentes a implementação técnicas utilizadas nos reconhecedores de padrões apresentadas. No primeiro experimento serão realizadas diversas avaliações do reconhecedor utilizando a técnica DTW, com o objetivo de encontrar as melhores configurações para a implementação desta técnica.

No segundo experimento serão realizados testes com o objetivo de se obter as melhores configurações para o bloco de processamento de sinais. Tradicionalmente, a literatura recomenda o uso de valores padrões para estas configurações. Contudo, estes números são apenas uma estimativas, e fatores como a base de dados a ser reconhecida, tipo de método usado no reconhecimento de padrões e o idioma do a ser reconhecido, influenciam no valor destas configurações.

No último experimento será avaliado o desempenho de dois sistemas de ASR construídos com as técnicas discutidas neste trabalho. Todos estes serão implementados com as melhores configurações obtidas nos experimentos anteriores. No primeiro sistema não será utilizado nenhum limiar de identificação de palavra e, no segundo irá se utilizar o limiar proposto pelo autor.

Em todos os experimentos propostos serão analisadas as seguintes métricas de desempenho: a porcentagem de acertos e tempo médio para reconhecimento por palavra. A porcentagem de acertos foi calculada da seguinte maneira:

$$\%\text{ de acertos} = \frac{N}{M} \times 100\%, \tag{6.1}$$

onde N é o número total palavras do conjunto de testes que foram reconhecidas corretamente e M é o número total de palavras do conjunto de testes.

6.1 Base de Dados

A Tabela 2 exibe as palavras utilizadas para gravação da base de dados que será utilizada neste trabalho. Estas palavras foram escolhidas pois podem ser usadas em um sistema de automação residencial por comandos de fala. Para os primeiros experimentos deste trabalho, a base de dados foi previamente grava pelo autor, possuí 25 palavras com 15 amostras por palavra, resultando em 375 elocuções. Estas amostras foram divididas aleatoriamente da seguinte maneira: cada classe de palavras para o grupo de teste contém 5 amostras de cada palavra, totalizando 125 elocuções. O grupo de sinais de referência que serão utilizados para treinamento possuem 10 amostras para cada palavra, totalizando 250 elocuções.

Para o último experimento, foi utilizada a mesma base de dados e cada palavra do grupo de testes foi repetida 25, totalizando 625 elocuções. Este aumento no número de palavras foi feito com o objetivo de se obter estatísticas mais precisas do desempenho do sistema em uma simulação de uso real. Por fim, é importante salientar que em todos os testes a captação dos áudios foram feitas em ambiente residencial, possuindo ruído de fundo característico deste tipo de ambiente. Esta informação é importante, pois o desempenho de um sistema de ASR tem relação direta com o ambiente onde será utilizado, devido ao ruído de fundo presente neste ambiente (HUANG; ACERO; HON, 2001).

Tabela 2 – Palavras usadas para construção da base de dados.

Música	Acender	Lâmpada	Apagar	Luz
Tocar	Ventilador	Tomada	Abaixar	Porta
Emergência	*Notebook*	Abrir	Aumentar	Televisão
Cafeteira	Rádio	Fechar	*Internet*	Computador
Parar	Panela	Ligar	Janela	Volume

6.2 Primeiro Experimento

O objetivo deste experimento é avaliar o desempenho do reconhecedor usando a técnica DTW, ilustrado na Figura 28. Para isso, a primeira avaliação foi feita com o objetivo de determinar qual é o tamanho de r mais adequado para o uso da restrição global do tipo banda de Sakoe-Chiba. O objetivo da segunda avaliação é obter quais combinações de restrições globais e locais produzem os melhores resultados. Por fim, a terceira avaliação tem como meta determinar qual é o número de *templates* utilizados no sistema que produza a melhor combinação de resultados.

Como este experimento objetiva analisar apenas o reconhecedor de padrões, o número de coeficientes MFCC extraídos pelo bloco de processamento de sinais, o número de filtros utilizados no banco de filtros e os intervalos de tempo de duração de cada quadro no sistema VAD e para a extração dos coeficientes MFCC seguem as recomendações da literatura dadas em (BENESTY; SONDHI; HUANG, 2008), (HUANG; ACERO; HON, 2001), (RABINER; SAMBUR, 1975), (RABINER; SCHAFER, 2007), e foram fixadas em: 13 coeficientes MFCC, 26 filtros, 10ms de duração para cada quadro e 25ms de duração para cada quadro com sobreposição de 10ms entre os quadros respectivamente.

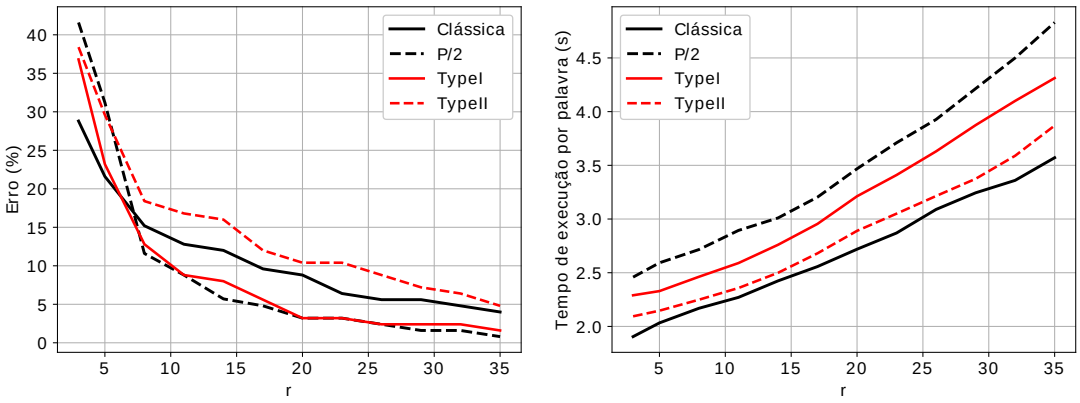
6.2.1 Primeira Avaliação

A primeira avaliação deste experimento foi realizada para determinar o tamanho de r quando se usa a restrição global do tipo banda de Sakoe-Chiba. Conforme explicado na Seção 4.1.3.2, para o uso da banda de Sakoe-Chiba como restrição global primeiro é necessário que se determine adequadamente o tamanho da janela de análise. Portanto,

neste teste, foi avaliado o desempenho do DTW variando o tamanho de r da banda de Sakoe-Chiba para todas as restrições locais mostradas na Seção 4.1.3.1.

O tamanho de r inicial utilizado foi 3 pois, valores menores ficaram muito próximos a linha diagonal de $\mathbf{G}$ produzindo muitos erros. O último valor de r utilizado foi 35, visto que, acima deste valor a área de análise tornava-se muito ampla, não justificando o uso da restrição. A Figura 31 demonstra os resultados obtidos para a avaliação de taxa de acertos (Figura 31a) e tempo médio de execução do DTW (TDTW) para reconhecimento por palavra (Figura 31b).

Figura 31 – Resultado obtido para o teste de tamanho r da banda de Sakoe-Chiba.



(a) Taxa de erro em função de r. (b) TDTW em função de r.

Fonte: Autoria Propria.

Observe que a escolha do tamanho de r é um compromisso entre precisão e tempo de reconhecimento por palavra. Aumentando o tamanho de r têm-se uma área de buscas em $\mathbf{G}$ menos restritiva, permitindo o correto casamento de padrões de fala pertencentes a mesma classe de palavras com maior desalinhamento temporal. Entretanto, com o aumento de r, aumenta-se o número de elementos em $\mathbf{G}$ que serão analisados, aumentado o TDTW por palavra.

Estas figuras mostram ainda que esta relação custo-benefício também é observada quando analisado o tipo de restrição local utilizada. Observe que quanto mais restrito for r, maior é a taxa de erros apresentada pelas restrições locais que cobrem uma maior área durante o processo de busca em $\mathbf{G}$. Contudo, aumentando o valor de r, mais tempo é necessário para se reconhecer uma palavra usando as restrições locais que cobrem uma maior área.

Observa-se nos resultados que as restrições locais do tipo $\frac{P}{2}$ e *TypeI* alcançaram as maiores taxas de acertos, e o aumento de precisão para r maior que 20 é pequeno em relação ao aumento de tempo gasto para se reconhecer uma palavra, não justificando a

escolha de r maior que 20 para essas restrições. Observa-se ainda que, para as restrições que alcançaram as menores taxas de acerto, mesmo com r igual a 35, a taxa de acertos são inferiores as obtidas pelas restrições do tipo $\frac{P}{2}$ e *TypeI* com r igual a 20 e o TDTW por palavra é maior, não justificando a escolha deste tamanho de r. Considerando esses fatos, foi escolhido tamanho de 20 para r.

6.2.2 Segunda Avaliação

Este ensaio analisou qual das combinações entre restrições globais e locais obteriam o melhor desempenho quando analisados a taxa de acertos obtida e o TDTW por palavra. Para os métodos que utilizaram a restrição global do tipo banda de Sakoe-Chiba foi utilizado o valor de r de 20. A Tabela 3 compila os resultados obtidos.

Tabela 3 – Avaliação de desempenho das restrições locais e globais.

Restrição local	**Restrição global**	**Acertos (%)**	**TDTW (s)**
Clássica	Sakoe-Chiba	91,20	2,75
Clássica	Itakura	90,40	2,66
$\frac{P}{2}$	Sakoe-Chiba	96,80	3,47
$\frac{P}{2}$	Itakura	96,80	3.41
TypeI	Sakoe-Chiba	96,80	3,29
TypeI	Itakura	96,80	3,19
TypeII	Sakoe-Chiba	89,60	2,89
TypeII	Itakura	88,80	2,83

A partir dos resultados da Tabela 3 pode-se concluir que a combinação que utilizada a restrição local do tipo *TypeI* e restrição global do tipo paralelogramo de Itakura é a mais eficiente em modelar as variações temporais entre os sinais de fala existentes nesta base de dados, sendo a melhor escolha para continuidade do trabalho. Esta conclusão é baseada no fato desta combinação apresentar a maior taxa de acertos com o menor TDTW, em relação as outras combinações que também apresentaram a maior taxa de acertos.

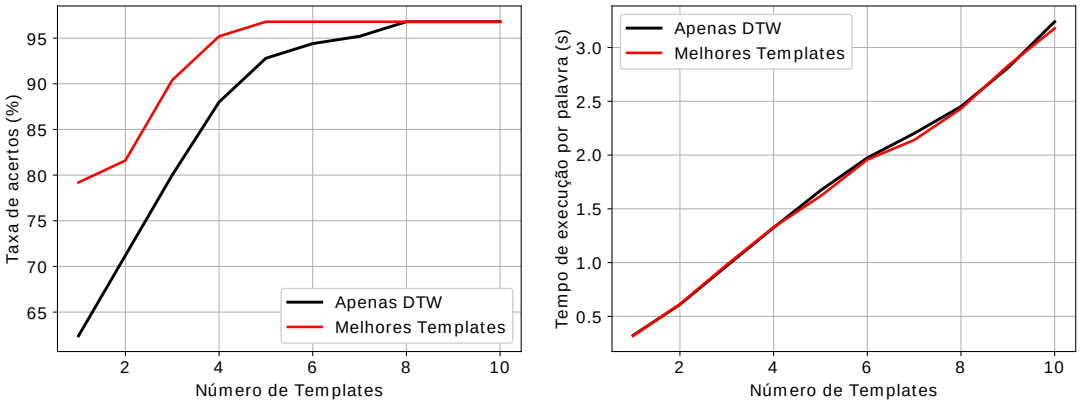
6.2.3 Terceira Avaliação

A terceira avaliação teve por objetivo determinar qual o número de *templates* necessários para o sistema de ASR proposto produzir a melhor relação entre taxa de acertos e o TDTW por palavra. Para isso, nesta avaliação será testado os dois métodos de treinamento: o primeiro será o método tradicional, onde na etapa de treinamento é construído o banco de modelos acústicos diretamente após etapa de extração de características do sinal fala; no segundo método, será utilizando a seleção de melhores *templates*.

A Figura 32 ilustra os resultados para taxa de acertos (32a), e o TDTW por palavra (32b) obtidos para o reconhecedor que utiliza o método de treinamento tradicional e para

o método que faz uso da seleção de melhores *templates* durante etapa de treinamento, respectivamente.

Figura 32 – Avaliação do desempenho dos reconhecedores usando o método de treino tradicional e o método de treino de seleção de melhores *templates* em função do número de *templates* utilizados.



(a) Taxa de acertos. (b) TDTW por palavra.

Fonte: Autoria Propria.

Observe que há uma melhora gradativa na taxa de acertos conforme aumenta-se o número de *templates* de referência usados para ambos os métodos avaliados. Contudo, o preço a se pagar por esta melhora é o aumento do TDTW por palavra, mostrando a existência de uma relação de custo-benefício na seleção do número de *templates* de referência.

Também nota-se que o uso do método de seleção de melhores *templates* obtém uma taxa de acertos maior utilizando menos modelos de referência, resultando em um menor TDTW por palavra, tornando a sua adoção benéfica. Baseando-se nestes resultados, para a continuidade deste trabalho, nos próximos reconhecedores testados será utilizado na etapa de treinamento a seleção de melhores *templates*, utilizando 5 *templates* por sinal de fala de referência, pois esta é a configuração que alcançou a melhor taxa de reconhecimento com o menor TDTW por palavra nesta avaliação.

6.3 Segundo Experimento

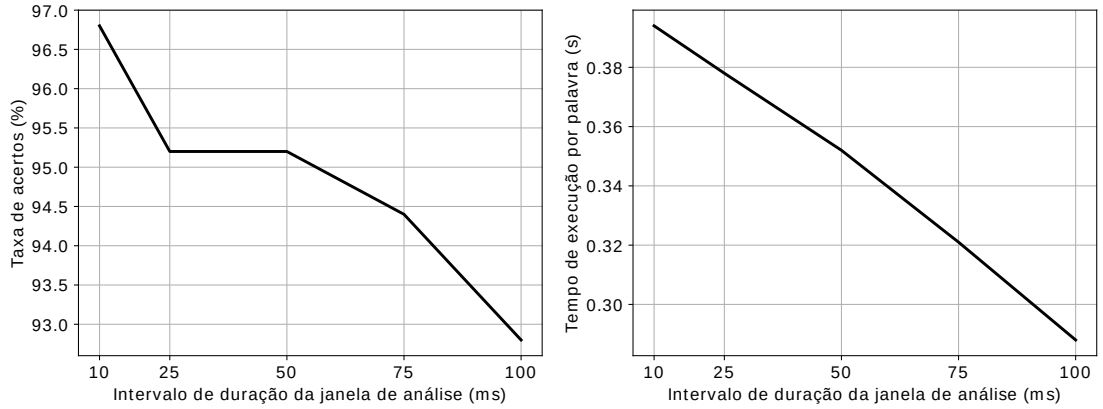
Neste experimento será realizado um estudo dos efeitos da variação das configurações existentes no bloco de processamento de sinais. Para tal, primeiro será avaliado qual é a duração de tempo de cada quadro para a técnica VAD empregada que apresenta os melhores resultados em relação a taxa de acertos e tempo de execução por palavra, mantendo as outras configurações constantes com valores de acordo com o recomendado pela literatura.

Em seguida, esta avaliação será repetida variando o intervalo de duração de tempo de cada quadro para extração dos coeficientes MFCC. Na sequência, este teste será repetido variando o número de filtros utilizados no banco de filtros. Por fim, será feito um ensaio variando do número de coeficientes MFCC utilizados no bloco de processamento de sinais.

6.3.1 Primeira Avaliação

Nesta avaliação, foi realizado o estudo dos efeitos da variação do intervalo de tempo utilizado para segmentar em quadros o sinal de fala para a utilização da técnica VAD em intervalos de 10, 25, 75 e 100ms, medindo a taxa de acertos e o tempo médio de execução desta técnica (TVAD). A Figura 33 apresenta os resultados obtidos. Nota-se que o melhor resultado foi obtido quando o sinal de fala foi segmentado em quadros com duração de 10 ms.

Figura 33 – Resultados para o sistema de ASR implementado em função da variação do intervalo de tempo utilizado para segmentar em quadros o sinal de fala para o sistema VAD.



(a) Taxa de acertos. (b) TVAD.

Fonte: Autoria Propria.

Este resultado já era esperado pois, quanto menor o período de duração de cada quadro, mais detalhadas são as informações disponíveis sobre o sinal de fala, permitindo que se identifique com mais clareza o ponto onde a sentença de fala incia e termina. Contudo, quanto maior o número de segmentos por sinal, maior o custo computacional para processar o sinal de fala, aumentando o tempo gasto na execução. Todavia, os resultados mostram que o aumento do tempo de processamento para o sinal segmentado em quadros com menor período de duração é baixo, e não impacta severamente no desempenho do sistema.

De forma geral, o uso da quadros com menor período de duração é considerado vantajoso pois, a mesma apresenta a maior taxa de reconhecimento sem um aumento significativo do tempo de execução. Para a próxima avaliação será utilizado o valor de

10ms para o intervalo de duração de cada quadro quando o sinal de fala é segmentado para implementação do sistema VAD.

6.3.2 Segunda avaliação

Nesta avaliação, foi realizada a análise dos efeitos da variação do intervalo de tempo utilizado para segmentar em quadros o sinal de fala para a extração dos coeficientes MFCC. Para este teste foi variado também a proporção de sobreposição entre os segmentos do sinal de fala. Em (BENESTY; SONDHI; HUANG, 2008) recomenda-se uma sobreposição de ao menos 50% entre os quadros, entretanto, neste mesmo trabalho para um sinal de fala utilizando segmentado em quadros com duração de 25ms cita-se como valor tradicional o deslocamento dos quadros a cada 10ms, resultando em uma proporção de sobreposição de 60%. Portanto, neste teste foram avaliadas as proporções de 50% e 60%. Neste teste avaliou-se apenas a taxa de acertos pois, durante os testes realizados, notou-se o tempo de execução afeta de maneira significante somente a etapa de treinamento, que não está sendo avaliada. A Tabela 4 mostra os resultados obtidos.

Tabela 4 – Resultados para o sistema de ASR implementado em função da variação do intervalo de tempo utilizado para segmentar em quadros o sinal de fala para extração dos coeficientes MFCC.

Duração do quadro (ms)	**proporção (%)**	**Acertos (%)**
10	50	96,80
10	60	96,80
25	50	96,80
25	60	96,80
50	50	95,20
50	60	95,20
75	50	93,60
75	60	94,30
100	50	92,80
100	60	93,60

Os resultados mostram que não há diferenças em relação a taxa de acertos utilizando segmentação em quadros com período de duração de 10ms e 25ms. Para valores de intervalo de tempo maiores o resultado obtido era esperado visto que, aumentado o período de tempo do quadro perde-se informações sobre a variação do sinal de fala, diminuindo a taxa de reconhecimento obtida. Baseado nos resultados obtidos, a próxima avaliação será feita utilizando o valor de 25ms com sobreposição entre quadros de 50% pois este valor alcança a máxima taxa de acertos e requer um menor processamento, uma vez que o sinal de fala é segmentado em um menor número de quadros durante a análise.

6.3.3 Terceira Avaliação

Nesta avaliação foi realizada a análise da variação do número de filtros usados para a extração dos coeficientes MFCC. Neste teste foi avaliado apenas a taxa de acertos devido aos mesmos motivos apresentados na avaliação anterior. A Tabela 4 exibe os resultados obtidos.

Tabela 5 – Resultados para o sistema de ASR implementado em função da variação do número de filtros utilizados.

Número de filtros	**Acertos (%)**
20	96,80
25	96,80
30	96,80
35	96,80
40	96,80

A partir dos resultados apresentados pela Tabela 4, pode-se concluir que para o uso do DTW como reconhecedor de padrões para palavras isoladas, o número de filtros utilizados para obtenção dos coeficientes MFCC não é um fator determinante no desempenho da taxa de acertos. Os resultados mostram que a taxa de acertos foi igual para todos os valores de coeficientes de filtros avaliados. De acordo com (HUANG; ACERO; HON, 2001), o uso de um maior número de filtros pode aumentar o desempenho do sistema em até 10%, contudo, não é especificado que tipo de reconhecedor foi utilizado, e nem que tipo de sistema de ASR foi avaliado para obtenção deste dado.

Acredita-se que a variação desta configuração tenha efeito na taxa de precisão do sistema quando o reconhecedor de padrões é implementado utilizando HMM para o reconhecimento de fala contínua pois, neste tipo de reconhecedor, são utilizadas partículas menores da fala, como fonemas para o reconhecimento. Com o aumento do número de filtros, aumenta-se a seletividade em frequência de cada filtro, resultando em uma melhor distinção das frequências onde encontram-se os formantes nestes fonemas, o que pode contribuir para este tipo de sistema conseguir diferenciar fonemas parecidos.

Baseado nestas considerações, para o prosseguimento do trabalho foi adotado um valor de 20 filtros, considerando que o uso de um menor número de filtros no banco de filtros diminui o tempo de execução de cada avaliação, e agiliza o processo de treinamento nos reconhecedores implementados para as próximas avaliações.

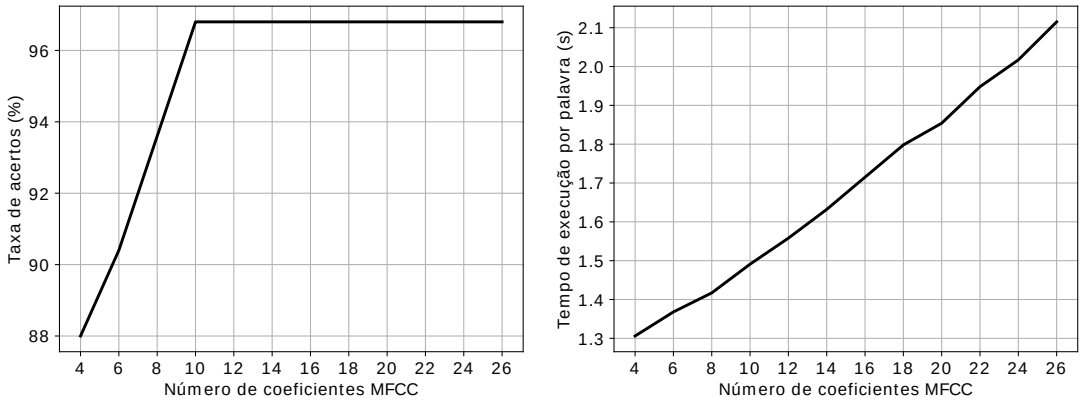
6.3.4 Quarta Avaliação

A Figura 34 apresenta os resultados obtidos para a análise realizada no quarto teste, onde foi avaliado o desempenho do sistema de ASR em função do número de coeficientes

MFCC utilizados. Nesta avaliação foram analisadas a taxa de acertos e o TDTW por palavra.

Os resultados obtidos mostram que há um aumento progressivo do percentual de acertos, conforme o número de coeficientes MFCC aproxima-se do valor recomendado pela literatura, alcançando o valor máximo a partir de 10 coeficientes. Analisando este resultado, pode-se concluir que abaixo de 10 coeficientes MFCC para este método de reconhecimento de padrões, não há informação suficiente sobre os formantes em algumas palavras que compõem o sistema, causando erros de reconhecimento. Para os valores acima do recomendado pela literatura, os resultados mostram que apesar do uso de coeficientes MFCC de ordem mais elevada adicionarem mais informações sobre o sinal de fala, não há melhora na taxa de acertos, indicando que a presença dos formantes nos coeficientes de menor ordem é o suficiente para o correto reconhecimento das palavras quando utilizado o DTW.

Figura 34 – Desempenho do sistema de ASR implementado em função do número coeficientes MFCC utilizados.



(a) Taxa de acertos.

(b) TDTW por palavra.

Fonte: Autoria Propria.

Em relação ao TDTW, os resultados mostram que quanto maior o número de coeficientes utilizados, maior é o tempo de reconhecimento. Este resultado era esperado pois, aumentando o número de coeficientes MFCC, aumenta-se a dimensão de cada vetor acústico que representa o sinal de fala após o emprego da técnica MFCC. Para aplicação do DTW é feito o cálculo da distância euclidiana entre o sinal a ser reconhecido e os *templates* de referência, conforme discutido no Capítulo 4. Aumentando-se as dimensões dos vetores acústicos que representam os sinais de fala, aumenta-se o número de cálculos a serem feitos durante o processo de cálculo da distância euclidiana no DTW, consequentemente, aumentando o seu tempo de execução. Baseado nos resultados obtidos nesta avaliação, para a continuidade deste trabalho foi selecionado o uso de 10 coeficientes MFCC após a

extração das características do sinal de fala.

6.4 Terceiro Experimento

Este experimento foi realizado com o objetivo de simular o comportamento real do sistema de ASR proposto, implementado utilizando a mesma base de dados para treinamento, e as melhores configurações obtidas nos experimentos anteriores, ilustradas na Tabela 6.

No primeiro teste, será feita a avaliação de desempenho do sistema implementado sem a utilização de nenhum limiar de identificação de palavra, medindo a taxa de acertos, TDTW e o tempo de execução total por palavra reconhecida. Nesta avaliação não será analisado o tipo de erro produzido pelo sistema pois, conforme explicado na Seção 5.3.3, quando não se utiliza limiares de identificação de palavra, há a troca de palavras em todos os erros produzidos pelo sistema. Nesta avaliação ainda será observado as estatísticas sobre as distâncias produzidos pelo DTW, para a sua utilização determinação do limiar de identificação de palavra.

No segundo ensaio, será avaliado o desempenho do sistema utilizando o limiar de identificação de palavra proposto pelo autor. Para tal, serão medidos a taxa de acertos, a taxa de erros produzidos por palavras identificadas incorretamente (EI), a taxa de erros gerados por palavras não identificadas (ENI), isto é, as palavras que foram rejeitadas pelo sistema por possuírem distância mínima maior que o limiar de identificação de palavra, o TDTW e o tempo total de execução por palavra.

Tabela 6 – Configurações do Sistema de ASR implementado.

Restrições local e global do DTW	*TypeI* e Paralelogramo de Itakura
Treinamento usado no reconhecedor de padrões	Seleção dos melhores *templates*
Número de *templates* usados	5
Tempo de quadro na técnica VAD	10ms
Tempo de quadro para o MFCC	25ms
Filtros utilizados no banco de filtros	20
Coeficientes MFCC utilizados	10

6.4.1 Primeira Avaliação

A Tabela 7 exibe os resultados obtidos em relação a taxa de acertos, o TDTW e o tempo total de execução por palavra para esta avaliação. Nota-se que há redução na taxa de acertos em relação as taxas de acerto obtidas nos experimentos anteriores. Esta variação tem três motivos principais: variações da pronúncia de uma mesma palavra, influência de recortes ruins dos sinais de fala feitos pelo sistema VAD e o ruído de fundo do ambiente de gravação.

Tabela 7 – Resultados obtidos para o sistema de ASR proposto.

Acertos (%)	**TDTW (s)**	**Tempo total (s)**
90,24	1,48	1,90

Para a primeira fonte de erros, conforme ponderado na Seção 5.3.2, uma das principais desvantagens do DTW é a sua sensibilidade a variação da pronúncia de uma palavra. Nos experimentos anteriores foram realizados diversos testes para configuração do sistema, de modo a aumentar a sua robustez a essas variações. Contudo, o sistema continua sensível a pronúncias de palavras que variem muito em relação aos *templates* de referência. Como neste ensaio foram realizadas 5 vezes mais elocuções da mesma palavra em relação aos testes anteriores, pode-se concluir que, em algum momento, algumas palavras foram pronunciadas de maneira muito diferente em relação aos *templates* de referência existentes no sistema, gerando erros de reconhecimento.

Em relação ao segundo principal motivo, como discutido na Seção 3.2.1, a técnica de VAD utilizada neste trabalho não é robusta aos trejeitos existentes na fala. Como neste teste foram feitas diversas elocuções de uma mesma palavra, é razoável concluir que durante a pronúncia de algumas palavras foram produzidos recortes ruins pelo sistema VAD por causa destes fenômenos.

Justifica-se o terceiro motivo devido as variações sonoras existentes no ambiente de testes. Como neste teste foram feitas diversas repetições da mesma palavra, o tempo necessário para sua conclusão foi relativamente alto e, durante a realização da avaliação, eventos como veículos passando na rua causaram grandes variações sonoras no ambiente, gerando erros de reconhecimento.

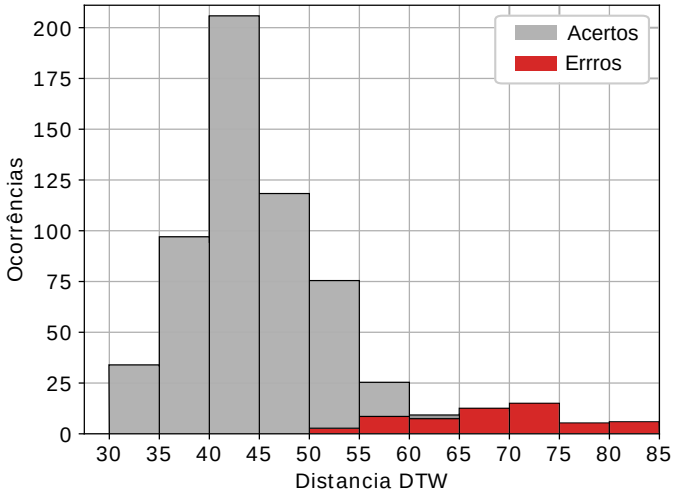
Por fim, observando os resultados obtidos em relação ao TDTW e ao tempo total de execução por palavra observa-se que, o tempo total é aproximadamente igual a soma do TDTW e o TVAD obtido no experimentos anteriores. O TDTW obtido mostra que o sistema levou aproximadamente 0,012 segundo para comparar cada *template*, uma vez que, existem 25 palavras no sistema e cada uma possuí 5 *templates* de referência.

Durante a realização deste teste foram ainda observados os valores de distância produzidos pelo DTW para cada palavra que foi reconhecida, para utiliza-los na determinação do limiar de identificação de palavra. Em posse destes dados, elaborou-se o histograma exibido pela Figura 35. O resultado obtido a partir deste histograma permite chegar a algumas conclusões.

A primeira delas é sobre a distribuição dos acertos obtidos. A figura mostra que em algumas palavras que foram corretamente identificadas, a distância DTW foi elevada em relação a média de acertos, fazendo intersecção com a região onde existem erros de reconhecimento. Indicando que para determinar um valor de limiar de identificação

de palavra que minimize os erros obtidos e não afete severamente a taxa de acertos, inevitavelmente, algumas palavras que seriam reconhecidas corretamente serão consideradas erros de identificação.

Figura 35 – Histograma da distância DTW obtida para número de acertos e de erros.



Fonte: O Autoria Própria.

Analisando a distribuição dos erros obtidos, observa-se que, em algumas palavras que foram reconhecidas de maneira errada, o valor de distância calculado pelo DTW foi menor que valor mais comum obtido pelos erros, e em uma faixa que existem um número considerável de palavras identificadas corretamente. Esse resultado nos mostra que, para determinar um limiar de identificação de palavra que permita ao sistema ter uma boa taxa de acertos, alguns erros de reconhecimento terão que ser tolerados, uma vez que selecionar um valor de limiar que elimine esses erros, irá afetar severamente a taxa de acertos do sistema.

De maneira geral, conclui-se que para determinar o limiar de identificação de palavra, é necessário escolher o valor que ofereça a maior taxa de acertos possível, tolerando o menor número de erros. Após a análise destes resultados, optou-se pelo uso de 63 para o limiar de identificação de palavra. Valores menores reduzirão os erros durante o reconhecimento, porém, também irão reduzir muito a taxa de acertos. Por outro lado, valores maiores aumentarão a taxa de acertos, mas também irão aumentar de maneira considerável os erros no sistema.

6.4.2 Segunda Avaliação

Neste ensaio repetiu-se o teste realizado na avaliação anterior, adicionando o limiar de identificação de palavra. Para avaliar o desempenho do sistema de ASR implementado neste teste foram utilizadas as seguintes métricas: taxa de acertos, taxa de erros provocados por palavras identificadas incorretamente (EI), a taxa de erros obtida por palavras não

identificadas (ENI), isto é, as palavras que foram rejeitadas pelo sistema por possuírem distância mínima maior que o limiar de identificação de palava, o TDTW e o tempo total de execução por palavra. A Tabela 8 exibe os resultados obtidos.

Tabela 8 – Resultados obtidos para o sistema de ASR implementado com o limiar de identificação de palavra.

Acertos (%)	**EI (%)**	**ENI (%)**	**TDTW (s)**	**Tempo total (s)**
88,16	1,60	10,24	1,47	1,88

Os resultados mostram que houve uma leve redução na taxa de acertos do sistema após a implementação do limiar de identificação de palavra proposto pelo autor. Este resultado era esperado pois, como analisado na avaliação anterior, com a adoção deste limiar, algumas palavras reconhecidas corretamente possuíam distâncias mínimas maiores que o valor do limiar escolhido e, inevitavelmente, serão consideradas erradas pelo sistema.

Os resultados mostram ainda que a taxa de erros por palavras identificadas incorretamente foi baixo, mostrando que o valor escolhido para o limiar de identificação de palavra foi bem balanceado. Analisando especificamente a origem deste tipo de erro, foi notado que eles ocorrem principalmente quando as palavras pronunciadas foram confundidas com palavras similares. Esta confusão pode ocorrer devido a variações na pronúncia da palavra ou por recortes mal feitos pelo sistema VAD.

Em síntese, apesar do sistema implementado com o uso do limiar de identificação de palavra apresentar uma leve redução na taxa de acertos, o número de palavras identificadas incorretamente foi baixo, mostrando que a adoção desta estratégia é benéfica para o desempenho do sistema. Analisando o ponto de vista de uso real de um sistema de ASR, é mais vantajoso que uma palavra não seja identificada e o usuário tenha que repetir a elocução da palavra, uma vez que será necessário apenas a repetição desta palavra, e não haverá transtornos para o usuário causados por ações indesejadas executadas devido ao erro de identificação. Em um cenário onde o sistema implementado apresenta uma alta taxa de palavras identificadas incorretamente, o usuário terá que falar um comando para cancelar a ação que foi executada de maneira errada e, somente então, repetir o comando desejado.

7 Considerações Finais e Trabalhos Futuros

7.1 Considerações Finais

Este trabalho apresentou o projeto de um sistema de reconhecimento de fala para palavras isoladas e dependente de locutor, implementado em um sistema computacional embarcado. Inicialmente, foram apresentados alguns cenários onde a automatização de tarefas utilizando comandos de fala podem ser utilizada. Estes cenários incluem aplicações como automatização de agendas pessoais, atendimento de chamadas telefônicas de maneira remota em um veículo e automação residencial. Além disso, foram apresentados alguns produtos que fazem uso de sistemas de reconhecimento de fala e são comercializados por empresas como a Google, Amazon e Apple. Algumas ferramentas *open source* que possibilitam a criação destes sistemas foram apresentadas e brevemente discutidas.

Em seguida, a arquitetura do sistema de ASR projetado e implementado foi apresentada. Esta arquitetura é composta por três blocos principais: o bloco de processamento de sinais, o bloco de modelos acústicos e o bloco de reconhecimento de padrões, de acordo com a arquitetura proposta por (HUANG; ACERO; HON, 2001). O bloco de processamento de sinais é responsável por realizar todas as etapas entre a aquisição do sinal de fala analógico até a extração de suas características, transformando-o em um sinal digital composto por vetores acústicos. No bloco de modelos acústicos são feitos os procedimentos necessários para o treinamento do sistema, e o bloco de reconhecimento de padrões emprega as técnicas utilizadas para o reconhecimento de uma palavra.

O primeiro bloco a ser estudado e desenvolvido foi o de processamento de sinais, destacando as técnicas *endpoint detection* para remoção de trechos de silêncio no sinal de fala, e a extração de características deste sinal, por meio da obtenção dos coeficientes MFCC, que foram utilizadas neste trabalho. A respeito da técnica *endpoint detection*, observou-se que, apesar desta técnica ser amplamente utilizada na literatura, a mesma possuí algumas limitações que impedem seu uso em uma situação real. No decorrer das análises realizadas neste trabalho, observou-se que esta técnica não é robusta aos trejeitos presentes na fala, o que acaba gerando erros de reconhecimento. As razões para essa limitação foram apresentadas, contudo, a solução deste problema continua em aberto. Em relação a técnica MFCC, observou-se que esta é eficiente em extrair características do sinal de fala e suas configurações precisam ser ajustados para obter-se os melhores resultados, contudo, neste trabalho não foi realizado um estudo de seu comportamento em ambientes muito ruidosos, o que pode prejudicar o seu desempenho (BENESTY; SONDHI; HUANG, 2008).

O bloco de reconhecimento de padrões foi construído baseado no algoritmo DTW e suas modificações. O desempenho deste bloco foi analisado e constatou-se que, para o sistema proposto neste trabalho, o algoritmo DTW com a restrição global do tipo paralelogramo de Itakura e restrição local do tipo *TypeI* obteve os melhores resultados, alcançando 96,8% de precisão. Nesta análise, também foi avaliado o desempenho do sistema quando seu bloco de modelos acústicos é construído utilizando a etapa de treinamento convencional e a baseada na seleção de melhores *templates*. Observou-se um melhor desempenho quando utilizado a seleção de melhores *templates*, uma vez que foi necessário o uso de menos modelos de referência para obter-se a taxa máxima de acertos.

Testes para se obter as melhores configurações dos diversos subsistemas que compõem o bloco de processamento de sinais foram realizados. Constatou-se que a melhor configuração para o sistema proposto neste trabalho possui: intervalo de duração de 10ms para cada quadro na técnica *endpoint detection*, 25ms de duração de cada quadro com sobreposição a cada 10ms, 20 filtros para o processo de extração dos coeficientes MFCC, dos quais apenas 10 coeficientes MFCC foram utilizados. Com estas configurações, o sistema atingiu 96,8% de precisão nos testes e um tempo de execução médio por palavra inferior a 2 segundos.

Implementou-se o sistema de ASR com os melhores resultados obtidos nos testes realizados no decorrer do trabalho, com o objetivo de simular o seu desempenho em uma situação real. Nestes testes, a taxa de reconhecimento obtida para o melhor caso foi de 88,16%, com um tempo de reconhecimento por palavra inferior a 2 segundos.

Em síntese, os resultados obtidos neste trabalho são considerados satisfatórios, apesar do sistema proposto possuir limitações, como ter sido desenvolvido para reconhecer palavras isoladas. O seu projeto e implementação podem ser vistos como a primeira etapa para o desenvolvimento de um sistema mais sofisticado, eficiente e robusto.

No decorrer das análises realizadas verificou-se que, apesar da literatura recomendar diversas configurações padrões para as técnicas utilizadas na implementação de um sistema de ASR, estas configurações são dependentes de diversos fatores, e precisam ser cuidadosamente ajustadas para que o sistema apresente o melhor desempenho. Além disso, durante as avaliações, notou-se que alguns aspectos da fala como a variações de pronúncia e os trejeitos existentes na fala possuem grande influência no desempenho de um sistema de ASR e, geralmente não recebem grande destaque da literatura. Por fim, observou-se que o microcomputador avaliado possui uma boa capacidade computacional, indicando que o mesmo possa ser adequado para a construção de um sistema de automação residencial por comandos de fala que utilize em sua implementação reconhecedores de padrões mais complexos, como os baseados em HMM.

7.2 Trabalhos Futuros

Para a continuidade deste trabalho existem diversas técnicas que propõe melhorias em todos os blocos que compõem um sistema de ASR que podem ser estudas. Além disso, pode-se propor ainda novas possibilidades que agreguem novas funções para este sistema. A seguir, serão elencadas algumas destas melhorias e possibilidades que se sugere para trabalhos futuros.

- Propor soluções para lidar com as limitações da técnica *endpoint Detection* em relação aos trejeitos da fala encontrados;
- implementar outras técnicas VAD compará-las com a técnica *endpoint Detection*;
- analisar comparativamente a técnica MFCC com outras técnicas que são mais robustas a presença de ruído como a *Power Normalized Cepstral Coefficients* - PNCC (KIM; STERN, 2016) e a *Relative Spectral - Perceptual Linear Predictive* - RASTA-PLP (MERMANSKY et al., 1992), com intuito de melhorar a representação do sinal de fala em situações que possuem a presença marcante de ruído;
- implementar e analisar o desempenho reconhecedores utilizando outras técnicas, como as baseadas em HMM e redes neurais na plataforma computacional utilizada neste trabalho;
- propor a implementação de novas abordagens para determinação do limiar de identificação de palavra;
- no âmbito da automação residencial, pode-se implementar no sistema de ASR um bloco que execute o comando de fala após a etapa de reconhecimento, controlando um dispositivo eletrônico remotamente. Para isso, pode-se realizar a integração entre o sistema de ASR executando no *Raspberry Pi 3 Model B+* e diversos dispositivos eletrônicos microcontrolados por meio do protocolo de troca de mensagens otimizado para sensores e dispositivos embarcados chamado *Message Queuing Telemetry Transport* - MQTT (MQTT, 2016).

Referências

ABDULLA, W. H.; CHOW, D.; SIN, G. Cross-words reference template for DTW-based speech recognition systems. In: *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*. [S.l.]: IEEE, 2003. Citado na página 45.

AMAZON INC. *Help & Costumer Service: Use Household Profiles on Alexa Devices*. 2018. Disponível em: <https://www.amazon.com/gp/help/customer/display.html?nodeId=201628040>. Acesso em: 20 mai. 2019. Citado na página 24.

AMAZON INC. *Amazon Store*. 2019. Disponível em: <https://www.amazon.com/gp/product/B0792K2BK6>. Acesso em: 20 mai. 2019. Citado na página 24.

APPLE INC. *HomePod arrives February 9, available to order this Friday*. 2018. Disponível em: <https://www.apple.com/newsroom/2018/01/homepod-arrives-february-9-available-to-order-this-friday/>. Acesso em: 20 mai. 2019. Citado na página 24.

BELLMAN, R.; KALABA, R. On adaptive control processes. *IRE Transactions on Automatic Control*, v. 4, n. 2, p. 1–9, nov. 1959. Citado na página 46.

BENESTY, J.; SONDHI, M. M.; HUANG, Y. *Springer Handbook of Speech Processing*. First. Berlin: Springer, 2008. Citado 16 vezes nas páginas 21, 26, 27, 30, 31, 35, 37, 38, 39, 42, 43, 44, 45, 63, 68 e 75.

CMU SPHINX. *About CMU Sphinx*. 2019. Disponível em: <https://cmusphinx.github.io/wiki/about/>. Acesso em: 20 mai. 2019. Citado na página 26.

DAHL, G. E. et al. Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. *IEEE Transactions on Audio, Speech and Language Processing*, v. 20, n. 1, p. 30–42, jan. 2012. Citado na página 44.

EXTREME TECH. *IBM, Honda Accord Brings Speech To Cars*. 2019. Disponível em: <https://www.extremetech.com/extreme/51642-ibm-honda-accord-brings-speech-to-cars>. Acesso em: 09 jul. 2019. Citado na página 25.

GOOGLE INC. *Google AI - Publication databease*. 2019. Disponível em: <https://ai.google/research/pubs/?area=SpeechProcessing>. Acesso em: 20 abr. 2019. Citado na página 44.

GOOGLE INC. *Google Store*. 2019. Disponível em: <https://store.google.com/us/category/home_entertainment>. Acesso em: 20 mai. 2019. Citado na página 23.

HTK. *What is HTK?* 2019. Disponível em: <http://htk.eng.cam.ac.uk/>. Acesso em: 20 mai. 2019. Citado 2 vezes nas páginas 25 e 26.

HUANG, X.; ACERO, A.; HON, H.-W. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Primeira edição. New Jersey: Prentice-Hall, 2001. Citado 22 vezes nas páginas 16, 17, 19, 20, 21, 22, 26, 27, 29, 30, 37, 38, 39, 40, 41, 42, 43, 44, 45, 63, 69 e 75.

IBM. *IBM Embedded ViaVoice*. Primeira edição. Florida: IBM Corporation, 2007. Citado na página 25.

ITAKURA, F. Minimum Prediction Residual Principle Applied to Speech Recognition. *IEEE Transactions Acoustics, Speech and Signal processing*, v. 23, n. 1, p. 67–72, fev. 1975. Citado 2 vezes nas páginas 52 e 54.

JULIUS TEAM. *About Julius*. 2019. Disponível em: <https://github.com/julius-speech/julius>. Acesso em: 20 mai. 2019. Citado na página 26.

JURAFSKY, D.; MARTIN, J. J. *Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Segunda edição. New Jersey: Prentice-Hall, 2007. Citado na página 16.

KIM, C.; STERN, R. M. Power-Normalized Cepstral Coefficients (PNCC) for Robust Speech Recognition. *IEEE Transactions on Audio, Speech and Language Processing*, v. 24, n. 7, p. 1315–1329, mar. 2016. Citado na página 77.

LATHI, B. P. *Linear Systems and Signals*. Segunda edição. New York: Oxford University Press, 2005. Citado na página 42.

MARTINS, J. A. *Avaliação de diferentes técnicas para reconhecimento de fala*. Tese (Doutorado) — Universidade Estadual de Campinas - UNICAMP, 1997. Citado 2 vezes nas páginas 44 e 45.

MERMANSKY, H. et al. Rasta-PLP Speech Analysis Technique. In: *ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*. [S.l.]: IEEE, 1992. Citado na página 77.

MERMELSTEIN, P. Distance Measures for Speech Recognition - Psychological and Instrumental. *Pattern Recognition and Artificial Intelligence*, p. 374 – 388, 1976. Citado 2 vezes nas páginas 39 e 40.

MICROSOFT INC. *Speech to text*. 2019. Disponível em: <https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/>. Acesso em: 20 mai. 2019. Citado na página 25.

MQTT. *What is MQTT?* 2016. Disponível em: <https://mqtt.org/faq>. Acesso em: 11 jun. 2019. Citado na página 77.

MYERS, C.; RABINER, L. Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition. *IEEE Transactions on Acoutstics, Speech, and Signal Processing*, ASSP-28, n. 6, p. 623 – 635, dez. 1980. Citado 5 vezes nas páginas 47, 48, 50, 52 e 53.

NUANCE. *Dragon Speech Recognition Software*. 2019. Disponível em: <https://www.nuance.com/dragon.html>. Acesso em: 20 mai. 2019. Citado na página 25.

PICONE, J. W. Signal Modeling Techniques in Speech Recognition. *Proceedings of the IEEE*, v. 81, n. 9, p. 1215–1247, Set. 1993. Citado 2 vezes nas páginas 29 e 40.

RABINER, L.; JUANG, B.-H. *Fundamentals of Speech Recognition*. Primeira edição. New Jersey: Prentice-Hall, 1993. Citado 12 vezes nas páginas 16, 20, 27, 28, 30, 39, 40, 44, 45, 46, 49 e 59.

RABINER, L.; SAMBUR, M. An Algorithm for Determining the Endpoints of Isolated Utterances. *The Bell System Technical Journal*, p. 297 – 315, 1975. Citado 7 vezes nas páginas 17, 31, 32, 33, 34, 35 e 63.

RABINER, L. R.; SCHAFER, R. W. *Introduction to Digital Speech Processing*. 1. ed. The Netherlands: Now Publishers Inc., 2007. Citado 6 vezes nas páginas 29, 30, 31, 32, 33 e 63.

RASPBERRY PI FOUNDATION. *Raspberry Pi 3 Model B+*. 2019. Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>. Acesso em: 20 abr. 2019. Citado 2 vezes nas páginas 56 e 57.

RASPBERRY PI FOUNDATION. *What is a Raspberry Pi?* 2019. Disponível em: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>. Acesso em: 20 abr. 2019. Citado na página 56.

REAL PYTHON. *The Ultimate Guide To Speech Recognition With Python*. 2019. Disponível em: <https://realpython.com/python-speech-recognition/>. Acesso em: 20 mai. 2019. Citado na página 26.

SAKOE, H.; CHIBA, S. Dynamic Programming Algorithm Optimization For Spoken Word Recognition. *IEEE Transactions Acoustics, Speech and Signal processing*, v. 26, n. 1, p. 43–49, fev. 1978. Citado 7 vezes nas páginas 46, 47, 48, 50, 52, 53 e 54.

SANTOS, G. B. dos. *Análise Fonético-Acústica das Vogais Orais e Nasais do Português: Brasil e Portugal*. Tese (Doutorado) — Universidade Federal de Goiás, 2013. Citado 2 vezes nas páginas 21 e 27.

SILVA, C. P. A. da. *Um Software de Reconhecimento de Voz para o Português Brasileiro*. Dissertação (Mestrado) — Universidade Federal do Pará, 2010. Citado 2 vezes nas páginas 20 e 25.

SILVA, D. F. *Large-scale similarity-based time series mining*. Tese (Doutorado) — Universidade de São Paulo - USP, 2017. Citado 6 vezes nas páginas 46, 47, 50, 51, 53 e 54.

SILVA, T. C. *Fonética e Fonologia do português: Roteiro de Estudos e Guia de Exercícios*. Nona edição. São Paulo: Editora Contexto, 2003. Citado na página 21.

SPEECHTEXTER. *About*. 2019. Disponível em: <https://www.speechtexter.com/about>. Acesso em: 20 mai. 2019. Citado na página 25.

TECNOBLOG. *Google Home expande suporte a comandos e voz em português*. 2019. Disponível em: <https://tecnoblog.net/281590/google-home-expande-portugues/>. Acesso em: 20 mai. 2019. Citado na página 24.

THE AMBIENT. *Apple HomePod guide: Your missing manual to the Siri smart speaker*. 2018. Disponível em: <https://www.the-ambient.com/guides/apple-homepod-release-date-price-features-182>. Acesso em: 08 out. 2018. Citado na página 24.

ZAHARIA, T. et al. Quantized Dynamic Time Warping (DTW) algorithm. In: *2010 8th International Conference on Communications*. [S.l.]: IEEE, 2010. Citado na página 45.