ENGENHARIA DE COMPUTAÇÃO

Carlos Fabiel Bublitz

SISTEMA DE RECONHECIMENTO DE LOCUTOR INTEGRADO A COMUNICAÇÃO E CONTROLE DOS MOVIMENTOS DE UM ROBÔ HUMANOIDE

Santa Cruz do Sul
2015

ENGENHARIA DE COMPUTAÇÃO

Carlos Fabiel Bublitz

SISTEMA DE RECONHECIMENTO DE LOCUTOR INTEGRADO A COMUNICAÇÃO E CONTROLE DOS MOVIMENTOS DE UM ROBÔ HUMANOIDE

Prof. Dr. Leonel Pablo Tedesco
Orientador

Santa Cruz do Sul
2015

ENGENHARIA DE COMPUTAÇÃO

Carlos Fabiel Bublitz

SISTEMA DE RECONHECIMENTO DE LOCUTOR INTEGRADO A COMUNICAÇÃO E CONTROLE DOS MOVIMENTOS DE UM ROBÔ HUMANOIDE

Prof. Dr. Rolf Molz

Avaliador

Santa Cruz do Sul

2015

ENGENHARIA DE COMPUTAÇÃO

Carlos Fabiel Bublitz

SISTEMA DE RECONHECIMENTO DE LOCUTOR INTEGRADO A COMUNICAÇÃO E CONTROLE DOS MOVIMENTOS DE UM ROBÔ HUMANOIDE

Prof. MSc. Charles Neu

Avaliador

Santa Cruz do Sul

2015

Carlos Fabiel Bublitz

SISTEMA DE RECONHECIMENTO DE LOCUTOR INTEGRADO A COMUNICAÇÃO E CONTROLE DOS MOVIMENTOS DE UM ROBÔ HUMANOIDE

Trabalho de Conclusão II apresentado ao Curso de Engenharia da Computação da Universidade de Santa Cruz do Sul, como requisito parcial para obtenção do título de Bacharel em Engenharia da Computação.

Orientador: Prof. Dr. Leonel Pablo Tedesco

Santa Cruz do Sul

2015

“Se você encontrar um caminho sem obstáculos, ele provavelmente não leva a lugar nenhum.”
- Frank Clark

RESUMO

A fala é uma das formas de comunicação mais efetivas e utilizadas por seres humanos. Quando adotada em sistemas computacionais como forma de entrada, torna qualquer interação homem-máquina em princípio mais natural. De acordo com essa tendência, sistemas de Reconhecimento Automático de Locutor (RAL) foram criados com o intuito de identificar o indivíduo emissor de um determinado sinal de voz, através da análise das características desse sinal. Uma das aplicações na utilização desses sistemas permite que se adicionem níveis de segurança à comunicação, através da autenticação de usuários e suas permissões. Em vista disso, o objetivo deste trabalho é embarcar um sistema de reconhecimento automático de locutor em um robô humanoide, no qual se permita a comunicação apenas por pessoas autorizadas. Esta comunicação poderá ser realizada de duas formas, sendo uma delas através de comandos para controlar a movimentação do robô, e também pela interpretação de simples perguntas, nas quais o robô deve ser capaz de processar uma resposta. Ao final, obteve-se uma aplicação com níveis aceitáveis de reconhecimento de locutor, sendo capaz de controlar, através da fala, a movimentação do robô e estabelecer uma comunicação simplificada na forma de perguntas e respostas. Para as etapas de extração de características do sinal de voz e identificação do locutor, são utilizados os métodos de Coeficientes Mel-Cepstrais e Gaussian Mixture Models, considerados o estado da arte pela literatura.

Palavras-chave: Reconhecimento de Locutor, Gaussian Mixture Models, Coeficientes Mel-Cepstrais, Humanoide.

ABSTRACT

Speech is one of the most effective and used ways of communication by humans. When adopted in computer systems as a form of input, it makes any human-machine interaction, in principle, more natural. According to this trend, Automatic Speaker Recognition systems were created in order to identify the sender of a particular individual voice signal, by analyzing the characteristics of that signal. One of the applications of these systems, allows us to add differents security levels to communications by authenticating users and their permissions. In view of this, the objective of this work is to develop an automatic speaker recognition into a humanoid robot, which should allow communication only by authorized people. This communication may be done in two ways, one being through commands to control the movement of the robot, and also for the interpretation of a simple question, which the robot must be able to process an answer. In the end, there was obtained an application with acceptable levels of speaker recognition and also it was able to control, through speech, the humanoid movements and providing a simplified communication in form of question and answers. For the extracting features of the speech signal and identifying speaker steps, the methods of Mel-Cepstral Coefficients and Gaussian Mixture Models are applied, which are considered state of art in the literature.

Keywords: Speaker Recognition, Gaussian Mixture Models, Mel-Cepstral Coefficients, Humanoid.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS

DCT	*Discrete Cosine transform*
DFT	*Discrete Fourier Transform*
FFT	*Fast Fourier Transform*
GMM	*Gaussian Mixture Models*
HMM	*Hidden Markov Models*
LPC	*Linear Prediction Coding*
LPCC	*Linear Prediction Cepstrum Coeffcient*
MFCC	*Mel-frequency Cepstral Coefficients*
RAL	*Reconhecimento automático de locutor*
SVM	*Support Vectors Machines*
VQ	*Vector Quantizantion*

SUMÁRIO

1 INTRODUÇÃO

Existe uma crescente demanda por uma interação homem-máquina mais natural, permitindo realizar, através dela, uma comunicação de forma simples e efetiva. Para tal, pensar nas características do usuário e suas possíveis limitações são a melhor forma de criar uma usabilidade satisfatória, pois essas características de interação fazem uma máquina ser mais ou menos aceitável entre os seres humanos (AWAIS, 2010). De acordo com esta tendência, em um futuro próximo será possível observar o crescimento de aplicações móveis e embarcadas capazes de reconhecer a fala e o locutor (DHINESH, 2011).

Os sistemas de reconhecimento automático de locutor têm como objetivo determinar automaticamente o indivíduo emissor de um determinado sinal de voz, através da análise das características extraídas desse sinal e também de locuções anteriores. Este sistema pode ser dividido em duas aplicações: verificação de locutor e identificação de locutor. Na verificação de locutor, o propósito é realizar uma decisão binária para aceitar ou rejeitar a identidade de uma pessoa com base nas suas ondas de voz, enquanto que na identificação de locutor é diferenciar o locutor utilizando um conjunto existente de pontos de voz (FAN, 2010).

A complexidade envolvida em aplicações de RAL está relacionada com a modalidade de texto. Essa modalidade pode ser classificada em dependentes de texto (LAXMAN, 2013) e independentes de texto (SEDDIK, 2004). Em sistemas dependentes de texto para que ocorra o reconhecimento, o locutor deve pronunciar um texto pré-determinado, no qual o sistema já deve ter sido treinado para recebê-lo como entrada. Por outro lado, em sistemas independentes de texto, o locutor deve ser reconhecido independente do texto que está sendo falado.

São inúmeras as aplicações dos sistemas de RAL, dentre elas, a autenticação de transações comerciais, controle de acesso e monitoramento de pessoas. Para cada aplicação pode-se utilizar um método computacional diferente para extrair as características do sinal de voz. Métodos como, extração de coeficientes de frequência Mel, do Inglês, *Mel-frequency Cepstral Coefficients* (MFCCs) da locução e modelos estatísticos de *Hidden Markov Models* (HMMs) e *Gaussian Mixture Models* (GMMs) são os mais populares, devido a qualidade dos resultados obtidos. A escolha do método depende principalmente da modalidade de texto envolvida e da taxa de reconhecimento que se deseja alcançar.

O uso de um método de RAL, permite a passagem de comandos de forma segura à uma determinada máquina. Após a autenticação do locutor, o sinal de voz pode ser convertido em texto e através de comandos pré-definidos a máquina pode realizar as funções desejadas. No caso de um robô humanoide, esses comandos podem resultar na sua movimentação ou na comunicação com o locutor. A facilidade da comunicação por voz faz com que a interação com o robô seja mais natural proporcionando um maior conforto ao usuário. Porém, a complexidade envolvida aumenta exponencialmente, sendo necessário desenvolver métodos capazes de reconhecer as solicitações do usuário e de processar uma saída adequada.

O objetivo principal deste trabalho é desenvolver um sistema de reconhecimento automático de locutor para o controle dos movimentos e uma comunicação simplificada com um robô humanoide. A aplicação deve ser capaz de identificar o locutor previamente cadastrado e depois de realizada a autenticação, interpretar os comandos recebidos pelo mesmo. Esses comandos podem ser, por exemplo, de acionamento de módulos de movimentação do humanoide ou simples perguntas, para as quais se espera uma resposta do robô.

Este trabalho está organizado da seguinte forma: O Capítulo 2 introduz os conceitos teóricos necessários para compreender o trabalho proposto. O Capítulo 3 apresenta trabalhos relacionados com este. O Capítulo 4 descreve a ferramenta desenvolvida e seus módulos. O Capítulo 5 apresenta os testes realizados e resultados alcançados. Por fim, o Capítulo 6, conclui o trabalho desenvolvido.

2 REFERENCIAL TEÓRICO

Este capítulo faz uma introdução aos métodos e tecnologias utilizadas para o desenvolvimento deste trabalho. Inicia com uma breve revisão a respeito dos sistemas de reconhecimento de locutor e após uma revisão mais detalhada do método a ser utilizado neste trabalho. As tecnologias de síntese de voz e a plataforma a ser embarcada na aplicação também serão discutidas neste capítulo.

2.1 Reconhecimento de Locutor

O sinal de voz contém não apenas a mensagem que está sendo dita, mas também informações a respeito da identidade do usuário. A partir dessas informações é possível aplicar diferentes métodos de reconhecimento de locutor. O método mais eficaz para o reconhecimento depende principalmente da modalidade de texto associada ao problema.

As modalidades de texto podem ser divididas em duas categorias:

- **Dependentes de texto:** O locutor deve pronunciar um texto pré-determinado pelo sistema. Este texto deve ser fixo e o sistema já deve ter sido treinado para reconhecê-lo.
- **Independentes de texto:** O locutor deve ser reconhecido independentemente do texto que seja falado.

O sucesso destas tarefas depende da extração e modelagem das características do sinal de voz, as quais podem efetivamente distinguir um locutor de outro. Segundo (MAFRA, 2002), os Hidden Markov Models (HMMs) demonstram os melhores resultados em aplicações dependentes de texto por serem modelos estatísticos, com grande capacidade de modelagem das dependências temporais associadas aos sinais de voz. Ainda segundo ele, outro modelo bastante utilizado é a Rede Neural Artificial, pois é um modelo com grande capacidade de reconhecimento e classificação de padrões estáticos.

Este método representa uma robusta solução para o problema podendo alcançar taxas superiores a 95% de identificação utilizando capturas curtas de 5 segundos de sinal de áudio (Reynolds, 1995). Esse resultado varia de acordo com o nível de ruído presente no sinal, porem por ser extremamente robusto para

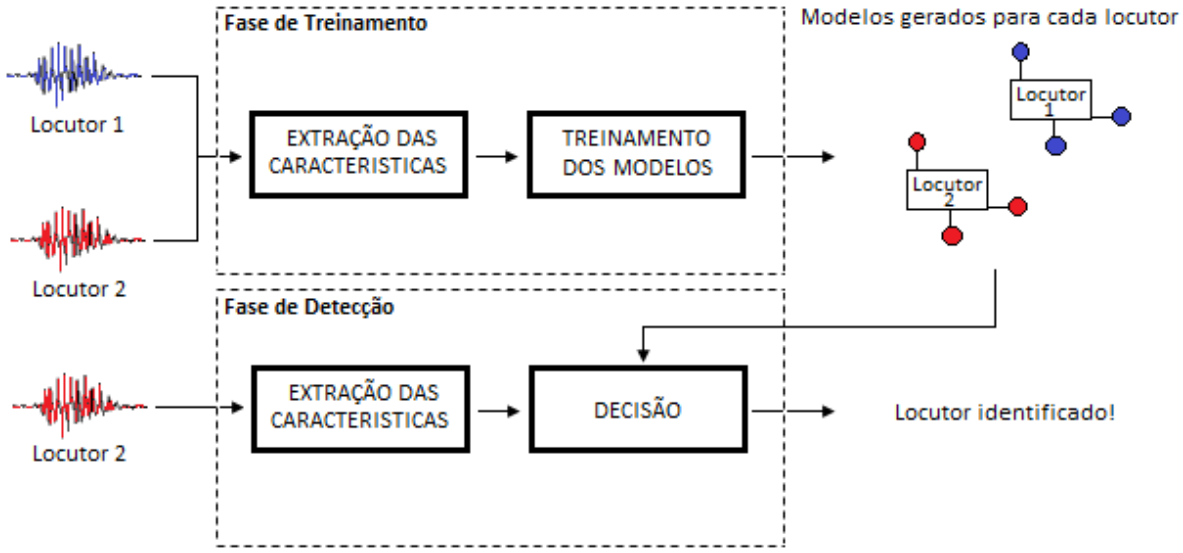
aplicações independentes de texto e com durações curtas de sinais de voz, este foi o método escolhido para ser utilizado neste trabalho.

Segundo Reynolds *et al.* (2000) o reconhecimento de locutor pode ser dividido em duas etapas:

- **Fase de Treinamento:** Nesta fase é gerado um modelo que representa cada locutor. Para isso, é realizada a extração de características do sinal de voz e com base nelas são gerados os Gaussian Mixture Models para cada locutor.
- **Fase de Detecção:** Após extrair as características do sinal de voz, esse conjunto de informações é aplicado nos modelos de cada locutor pré-cadastrado, para que seja realizada a identificação.

A Figura 1 apresenta as fases de treinamento e detecção do locutor.

Figura 1 - Fases do reconhecimento de locutor



Fonte: Adaptado de (REYNOLDS; HECK, 2000)

Para embarcar este módulo de reconhecimento de locutor no robô humanoide é necessário realizar uma sequência de etapas até alcançar o resultado final. Essas etapas são descritas a seguir.

2.1.1 Captura do Sinal de Voz

O sinal de voz é capturado através do microfone presente na estrutura do robô humanoide. Para realizar esta captura, foi desenvolvido um algoritmo para a detecção de início e fim da fala baseado na técnica de *Voice Activity Detection* (MOATTAR, HOMAYOUNPOUR, 2009). O sistema reconhece o início da fala do locutor de forma automática, sem precisar que seja sinalizado de alguma forma este início e também encerra a gravação do áudio ao detectar o fim da fala. Dessa forma, robô permanecesse em um estado de *listening* sendo capaz de identificar automaticamente quando o locutor falar com ele.

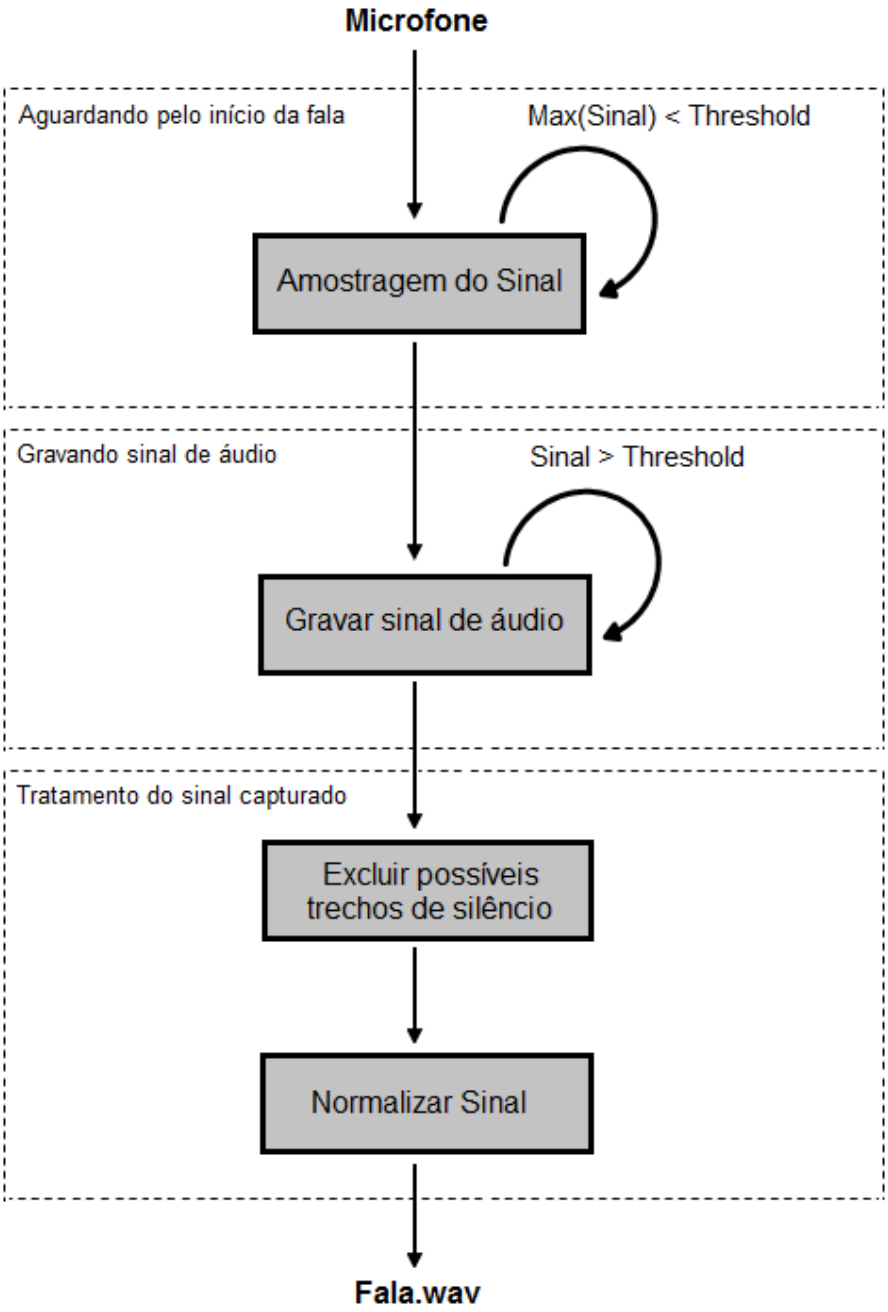
Neste algoritmo toda amostra de sinal de áudio capturada pelo microfone é analisada para que seja possível identificar o início e o término da fala. Para cada amostra capturada em tempo real, é necessário identificar se trata-se de silêncio ou fala. Para tal, foi definido, via experimentação uma constante de *threshold*. Essa constante foi definida em 50 db, sendo que todo sinal que tiver amplitude maior que esta constante será considerado fala. Este valor foi escolhido, pois trouxe uma maior fidelidade ao ambientes em que foi testado.

Quando identificado um sinal acima do *threshold* definido, a gravação do áudio é inicializada. Esta gravação é encerrada após o sinal capturado permanecer por alguns segundos com a amplitude abaixo do *threshold* definido. Dessa forma, é possível capturar a fala do locutor e salvá-la em um arquivo de áudio *.wav*.

Para melhorar a qualidade do sinal capturado são aplicados dois diferentes tratamentos: a eliminação de possíveis trechos de silêncio e a normalização do sinal. O primeiro tratamento é responsável por eliminar possíveis trechos de áudio considerados silêncio no fim da fala do locutor. Isto é necessário por conta do tempo que o sistema leva para identificar que o locutor parou de falar. Para isto é aplicado um filtro que elimina todos os sinais que estão abaixo do *threshold*. Após o sinal é normalizado, ou seja, é identificada a amplitude média do sinal e aplicada esta amplitude a todo o sinal, resultando um sinal uniforme.

Na Figura 2, observa-se o fluxo de captura do sinal de áudio com todas as etapas de obtenção e tratamento do sinal.

Figura 2 - Captura do sinal de áudio



Fonte: O Autor

2.1.2 Extração das características

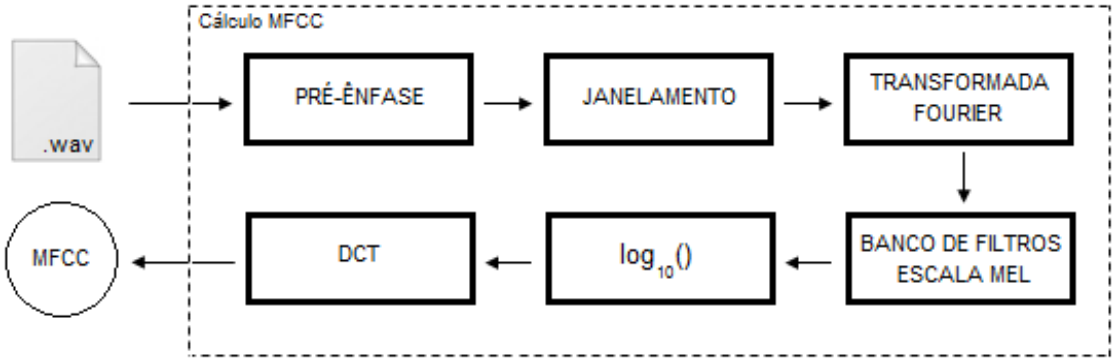
Para que ocorra o reconhecimento, é necessário obter uma representação paramétrica dos sinais, que possua todos os atributos necessários para a identificação. Segundo (REYNOLDS; HECK, 2000) um bom conjunto de atributos deve ser prático, robusto e seguro, o que significa que essas características devem ser facilmente extraídas do sinal de voz e serem pouco suscetíveis aos ruídos.

Existem algumas formas de distinguir os locutores baseadas em características da voz, porém a análise do espectro de voz têm sido um dos métodos mais efetivos para a identificação. Isto acontece porque o espectro reflete informações da estrutura do trato vocal, o que é a principal característica para distinguir duas pessoas. Para representar esse espectro de voz, um dos métodos que apresenta uma maior resistência à ruídos é a utilização de Coeficientes Cepstrais de Frequência Mel (MFCCs).

Coeficientes Cepstrais de Frequência Mel

Em sistemas de reconhecimento de locutor é comum a extração de coeficientes mel-cepstrais, de modo a mapear as características do sinal. O cálculo dos coeficientes segue o fluxo apresentado na Figura 3.

Figura 3 - Cálculos dos coeficientes Mel-Cepstrais



Fonte: Adaptado de (REYNOLDS; HECK, 2000)

O filtro de pré-ênfase é utilizado para atenuar as componentes de baixa frequência e incrementar as componentes de alta frequência do sinal de voz, pois a

energia carregada pelas altas frequências é pequena quando comparada com as baixas frequências. Isto se faz necessário para obter amplitudes mais homogêneas conservando assim as informações importantes que estão presentes nas altas frequências. A Equação 2.1 apresenta a função usada para o filtro de pré-ênfase.

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

O valor de a normalmente é definido em 0,95 (RABINER, 1993).

Após a pré-ênfase, são extraídos quadros de N amostras do sinal original, possibilitando assim trabalhar com menores porções do sinal original. Para separar cada um dos seguimentos, utiliza-se uma janela de Hamming, definida na Equação 2.2, onde n é o índice da amostra e a é o número total de amostras da janela.

$$W[n] = 0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right), n = 0, 1, 2, \dots, N-1 \tag{2.2}$$

Com um conjunto de janelas de amostras formado, pode-se aplicar a análise de *Fourier* em cada uma delas. Para isso aplica-se a DFT (Discrete Fourier Transform) em cada segmento, através do algoritmo de FFT (Fast Fourier Transform), estimando-se assim o espectro $S\,(w, m)$.

O banco de filtros da escala *mel* é formado com base em uma escala psi-coacústica de sensibilidade do ouvido para diversas frequências do espectro audível, chamada de escala *Mel* (STEVENS; VOLKMAN, 1940). Um *mel* é uma unidade de medida de frequência percebida para uma determinada frequência de entrada. A Equação 2.3 define a escala *Mel*.

$$Mel(f) = 112\; ln\left(1 + \frac{f}{700}\right) \tag{2.3}$$

A aplicação dos filtros da escala mel é dada pela Equação 2.4, onde N é o número de pontos da FFT, $|\mathrm{S(k,w)}|$ é módulo da amplitude na freqüência k-ésimo ponto da m-ésima janela e $\mathrm{H_i(w)}$ é a função de transferência do i-ésimo filtro triangular.

$$P(i) = \sum_{k=0}^{N/2} |S(k,m)|^2 Hi\left(k\frac{2\pi}{N}\right) \tag{2.4}$$

Em seguida, aplica-se a função log em todos os pontos do conjunto $P(i)$dada pela Equação 2.5.

$$L(k) = \log[P(i)] \tag{2.5}$$

Por fim, é calculada a Transformada Discreta de Cosseno (DCT) pela Equação 2.6, obtendo assim os coeficientes $c_{mel}(n)$.

$$c_{mel}(n) = \sum_{i=0}^{N_f} E(k_i)\, cos\left(\frac{2\pi}{N} k_i n\right), \qquad n = 0,1,2 \ldots, N_c - 1 \tag{2.6}$$

2.1.3 Gaussian Mixture Models

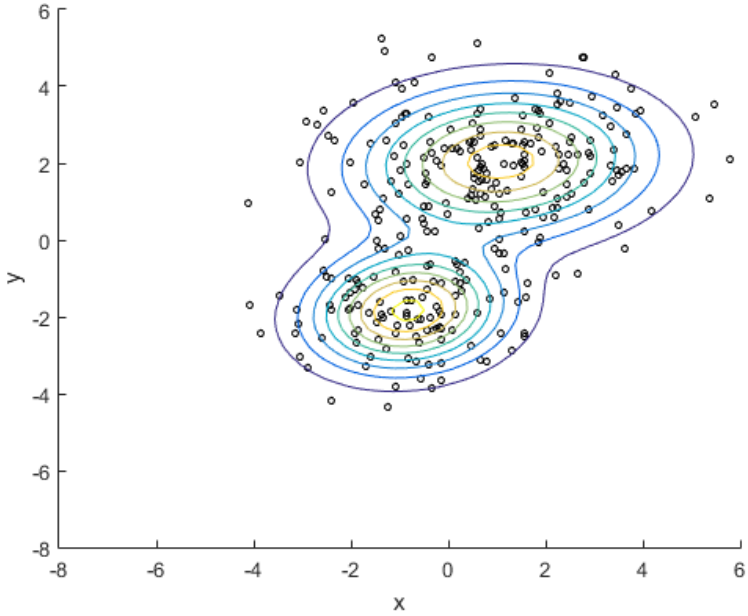
Para gerar um modelo que representa as características extraídas do sinal de áudio, é necessário utilizar alguma técnica de clusterização de dados. A clusterização é uma técnica utilizada para agrupar um conjunto de dados de acordo com a sua similaridade. Para realizar este agrupamento e gerar um modelo que representa as características da voz, utilizou-se a técnica de clusterização por Gaussian Mixture Models (GMM).

Os GMM´s são comumente usados como um modelo paramétrico da distribuição de probabilidade de medições ou funções contínuas em sistemas biométricos, tais como características espectrais do trato vocal para o uso em um sistema de reconhecimento de locutor (REYNOLDS; HECK, 2000). Este método de clusterização agrupa os dados através de funções gaussianas. Pode ser definido por uma função de densidade de probabilidade representada como soma ponderada das densidades de M componentes Gaussianos. Esta função é dada pela Equação 2.7.

$$P(\vec{x}|\gamma_s) = \sum_{i=1}^{M} p_i b_i(\vec{x}) \tag{2.7}$$

Este processo de clusterização é realizado através do algoritmo de *Expectation Maximization* (MOON, 2002). Este algoritmo tem seu funcionamento dividido em três etapas. Inicialmente, deve-se aplicar randomicamente *N* componentes Gaussianos ao conjunto de dados. Este conjunto de dados que pode ser visto como pontos distribuídos em um sistema cartesiano. Na segunda etapa, deve-se computar a probabilidade de cada um destes pontos pertencerem a alguma destas componentes do modelo. O processo da aplicação das componentes sobre o conjunto de dados pode ser visto na Figura 4.

Figura 4 - Aplicação das Gaussianas sobre um conjunto de dados

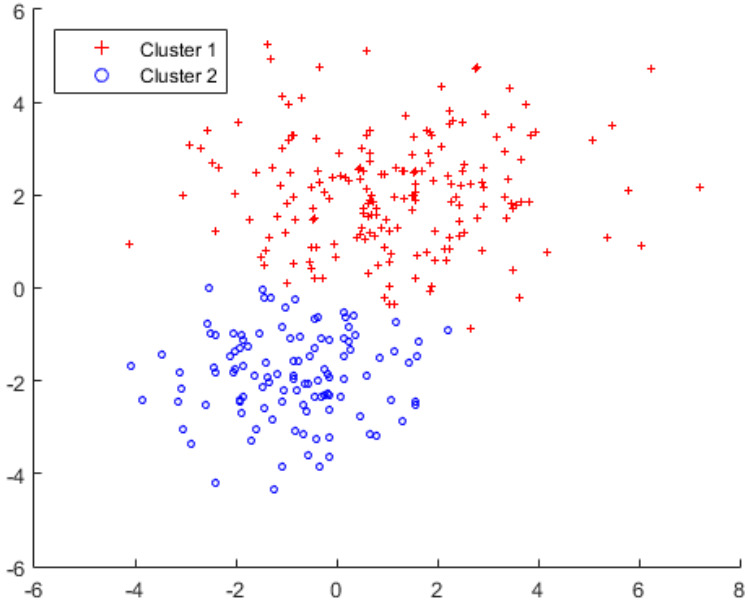


Fonte: Adaptado de (MATHWORKS, 2015)

A última etapa é responsável por atualizar a centroide de cada componente, baseada na posição média dos dados que a compõem. O objetivo é reposicionar a componente de forma a agrupar melhor o conjunto de dados.

A segunda e terceira etapa são repetidas até que a posição da centroide de cada componente não se altere mais, convergindo para a melhor posição. Na Figura 5 observa-se uma representação do conjunto de dados, na qual evidencia-se dois exemplos de clusters gerados pelo processo de clusterização.

Figura 5 - Processo de clusterização dos dados



Fonte: Adaptado de (MATHWORKS, 2015)

Para cada grupo S de locutores cadastrados S = {1,2,...,S}, existe uma representação GMM's dita por $y1, y2, \ldots, yS$. Para que ocorra a identificação de um locutor, deve-se encontrar o modelo com maior probabilidade seguindo a Equação 2.8.

$$S = \arg \max_{1 \leq k \leq S} p(x|\gamma_k) \tag{2.8}$$

As características extraídas da voz do locutor são aplicadas em cada modelo GMM's. Seguindo a equação 2.8, o modelo com maior probabilidade será considerado o locutor.

2.2 Reconhecimento e Síntese de Fala

Além de tratar o sinal de áudio capturado para reconhecer o locutor, deseja-se também, extrair o texto falado. Esta informação é crucial para módulos responsáveis pela movimentação e comunicação com o robô, pois é através do texto que são interpretados os comandos e perguntas. Para realizar esta tarefa, se faz necessário o uso de métodos de reconhecimento e síntese de fala.

As técnicas de reconhecimento de fala são responsáveis por transformar o sinal de áudio capturado em texto. Para que isso seja possível, o sinal de voz, que é analógico, deve ser digitalizado e amostrado. Dessa forma, pode-se dividí-lo em pequenos segmentos, que são comparados com os fonemas conhecidos de cada linguagem (RABINER; JUANG, 1993). Esse conjunto de fonemas é analisado utilizando funções probabilísticas para determinar a melhor saída (RABINER; JUANG, 1993).

Síntese de fala é uma técnica utilizada para a produção artificial da fala humana. Em uma visão alto nível desta técnica, uma possível implementação seria através da junção de diferentes gravações de palavras separadas, formando assim, a frase desejada.

Estas duas técnicas serão empregadas neste trabalho em duas diferentes situações. Primeiramente, deseja-se transformar o sinal de áudio proveniente da voz do locutor em texto, para que seja interpretado pelo módulo responsável por diferenciar comandos de perguntas. Após, se faz necessário transformar a resposta criada para uma determinada pergunta em áudio, fazendo com que o robô interaja com o usuário pela fala. O uso destas técnicas será feito através da API *Google Translate* (GOOGLE, 2015), a qual oferece os métodos necessários para as conversões de *Speech-to-Text* e *Text-to-Speech*. Esta API reproduz um tom de voz muito parecido com o humano, tornando assim, a fala mais amigável. Além disso, oferece uma grande facilidade em integrar essa tecnologia ao projeto.

2.3 DARwIn-OP

O robô humanoide DARwIn-OP (Dynamic Anthropomorphic Robot with Intelligence) de plataforma aberta, foi desenvolvido para a pesquisa nas áreas de robótica, inteligência artificial e educação. Projetado no núcleo de pesquisa das universidades de Tokyo e Virginia, em parceria com a empresa Robotics (ROBOTICS, 2015), no qual teve o seu desenvolvimento iniciado em 2004. É categorizado como robô humanoide por ter sua aparência global baseada na aparência do ser humano com braços, pernas e cabeça, imitando assim muitos dos movimentos dos seres humanos. Na Figura 4, pode-se visualizar a estrutura física do robô e suas características.

Figura 6 - Robô humanoide DARwIn-OP

Fonte: Adaptado de (ROBOTICS, 2015)

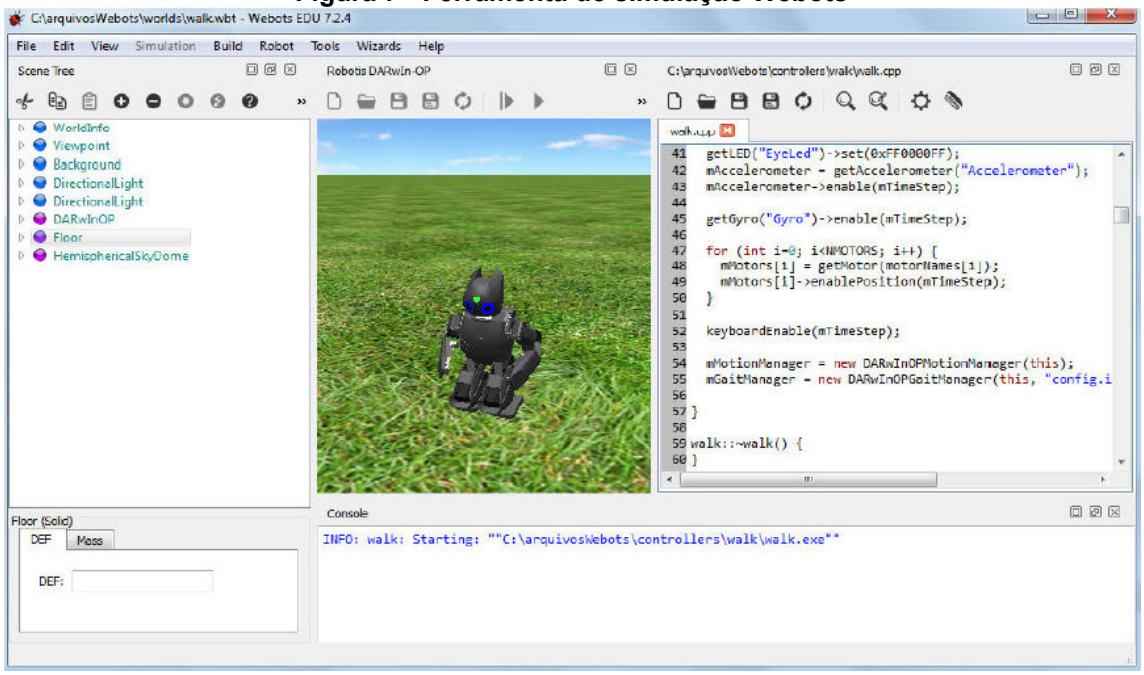
Baseado em uma estrutura modular, permite que os desenvolvedores façam alterações tanto de software como de hardware. Além dos diferentes dispositivos de entrada e saída, o humanoide possui um computador embarcado de plataforma *Linux*, o que oferece todo o poder computacional necessário para executar todos os módulos deste trabalho.

2.4 WEBOTS

Para programar e simular os movimentos do robô é necessário utilizar alguma ferramenta de simulação para facilitar este desenvolvimento. Testar o código desenvolvido diretamente no robô não é uma boa prática, pois pode danificá-lo já que algo pode acontecer errado durante execução. A ferramenta a ser utilizada para as simulações foi o Webots (WEBOTS, 2014).

O Webots é um ambiente de desenvolvimento usado para modelar, programar e simular robôs móveis, desenvolvido pelo Instituto Federal Suíço de Tecnologia. A partir dele é possível criar modelar um robô próprio e ambiente ou utilizar os vários modelos disponíveis, sendo possível alterar as propriedades de cada objeto, tal como forma, cor, textura, massa, fricção entre outras. Na Figura 7 é possível visualizar a interface do software.

Figura 7 - Ferramenta de simulação Webots



Fonte: O Autor

Este capítulo abordou detalhadamente cada uma das técnicas utilizadas nesse trabalho, apontando as tecnologias utilizadas como base para o desenvolvimento de cada um dos módulos. No próximo capítulo serão apresentados os trabalhos relacionados.

3 TRABALHOS RELACIONADOS

Este capítulo aborda certas aplicações que se assemelham ao trabalho proposto através de algumas características. A análise destes trabalhos teve como foco aqueles que utilizaram algum método de reconhecimento de locutor ou alguma forma de comunicação com um robô humanoide pela voz.

Analisaram-se cinco trabalhos relacionados. A Tabela 1 apresenta uma comparação entre as características do sistema proposto e dos trabalhos relacionados.

Tabela 1 - Comparativo das características dos trabalhos relacionados

Trabalho	**Extração caracteristicas**	**Clusterização**	**Tipo de reconhecimento**	**Plataforma**
Trabalho Proposto	MFCC	GMM	Independente de texto	DARwIn-OP
Shin *et al.* (2004)	API CMU Sphinx-4	API CMU Sphinx-4	Dependente de texto	Propósito geral
Farah *et al.* (2013)	MFCC, LPC	VQ	Dependente e Independente de texto	Propósito geral
Raja *et al.* (2006)	MFCC	VQ	Dependente de texto	Propósito geral
Fan *et al.* (2010)	LPCC	SVM	Independente de texto	Microprocessador ARM
Seddik *et al.* (2004)	MFCC	Rede Neural	Independente de texto	Propósito geral

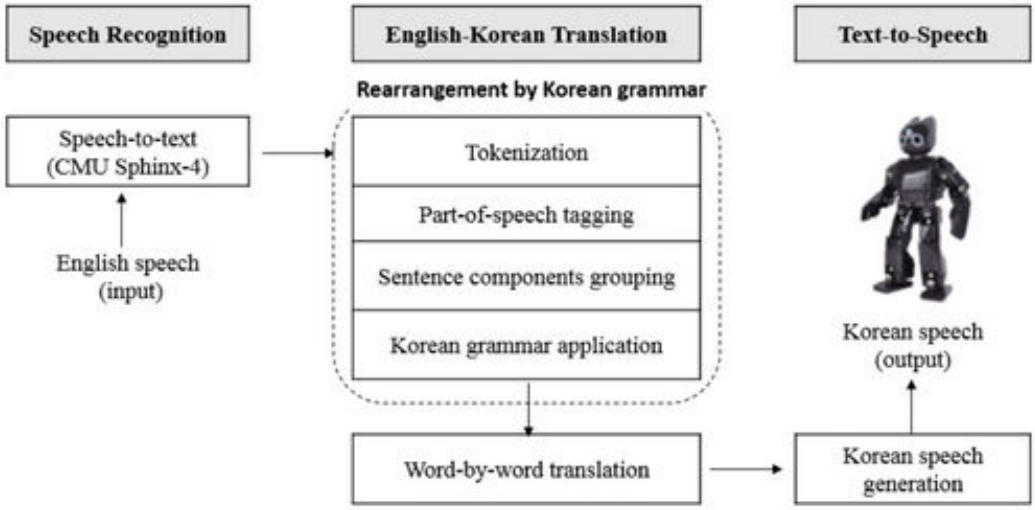
Fonte: O Autor

3.1 Speech-to-Speech Translation Humanoid Robot in Doctor´s Office

Em Shin *et al.* (2015) os autores propusseram um sistema de tradução de Inglês para Coreano embarcado em um robô humanóide DARwin-OP. Esta aplicação é voltada para a área de saúde, com o objetivo de traduzir para os médidos e enfermeiras os sintomas do pacientes que não fale Coreano. Utilizou-se um robô, pois este poderia facilmente se locomover pelo hospital abordando os pacientes e realizando uma triagem conforme a emergência.

O sistema proposto é dividido em três partes: reconhecimento da fala, tradução Inglês-Coreano e síntese da fala em Coreano, como observa-se na Figura 8. Para o reconhecimento da fala, os autores utilizaram a biblioteca *JAVA CMU Sphinx-4*, que permite treinar um conjunto de palavras para que sejam reconhecidas na fase de teste. Para determinar este conjunto de palavras, foi realizada uma pesquisa em parceria com um departamento de medicina para identificar as palavras-chaves dos sintomas de dor de cabeça, dor de estomago e gripe. Com isso, se tem um conjunto fixo e limitado de palavras que o sistema pode reconhecer.

Figura 8 - Módulos do sistema



Fonte: Adaptado de (SHIN; RAHMOUNI; SAYADI, 2015)

Para traduzir a fala do Inglês para Coreano são realizadas algumas etapas. Primeiramente, são identificadas as palavras que compõe a frase. Após, é aplicado algumas regras gramaticais a fim de ajustar corretamente o texto falado em Inglês para a estrutura da língua Coreana. Por fim, é realizada a tradução de cada uma dessas palavras para o Coreano com base no banco de palavras pré-cadastradas.

Para reproduzir o texto traduzido em Coreano não se utilizou um sintetizador de voz, mas sim um conjunto de arquivos de áudio em formato *mp3* para cada uma das palavras cadastradas.

O reconhecimento da voz e a tradução são realizados em um computador pessoal. Depois de realizadas essas tarefas é enviado uma série de comandos por uma conexão *TCP/IP* para o humanóide, no qual irá apenas reproduzir um arquivo MP3 para cada comando recebido. Cada um deste arquivos representa uma palavra que se deseja reproduzir.

3.2 Speaker Recognition System Using Mel-Frequency Ceptrum Coefficients, Linear Prediction Coding and Vector Quantization

O trabalho de Farah *et al.* (2013) tem como objetivo desenvolver um sistema de reconhecimento de locutor baseado nas técnicas de *Mel-Frequency Cepstrum Coefficients* (MFCC) e *Linear Prediction Coding* (LPC) para extração das características e *Vector Quantizantion* (VQ) para a formação dos modelos de cada locutor. O objetivo principal foi analisar o desempenho destas diferentes combinações de métodos.

A partir técnica de LPC, o objetivo dos autores é codificar o sinal de voz, baseando-se na premissa que a partir de algumas amostras do sinal é possível prever o seu comportamento. Para classificar este sinal de voz utiliza-se a técnica de VQ. Esta técnica é uma forma de compressão de dados, na qual o conjunto de dados do sinal de voz representado por *N* vetores é reduzido a *M* amostras ou *codeblocks*.

O trabalho faz uma comparação entre o desempenho obtido através da combinação das técnicas de LPC e VQ com MFCC e VQ empregadas tanto em um sistema dependente de texto como independente. O primeiro experimento avaliou o desempenho do sistema aplicando as técnicas de LPC e VQ. Nesta configuração para um dos cenários executados alcançou-se uma taxa de acerto de 74% para o reconhecimento dependente de texto e 36% para indepentende de texto.

Através da utilização de MFCC com VQ, os autores obtiveram um melhor desempenho em comparação ao uso da LPC. Alcançando uma taxa de acerto de 97% para o reconhecimento dependente de texto e 61% para independente de texto.

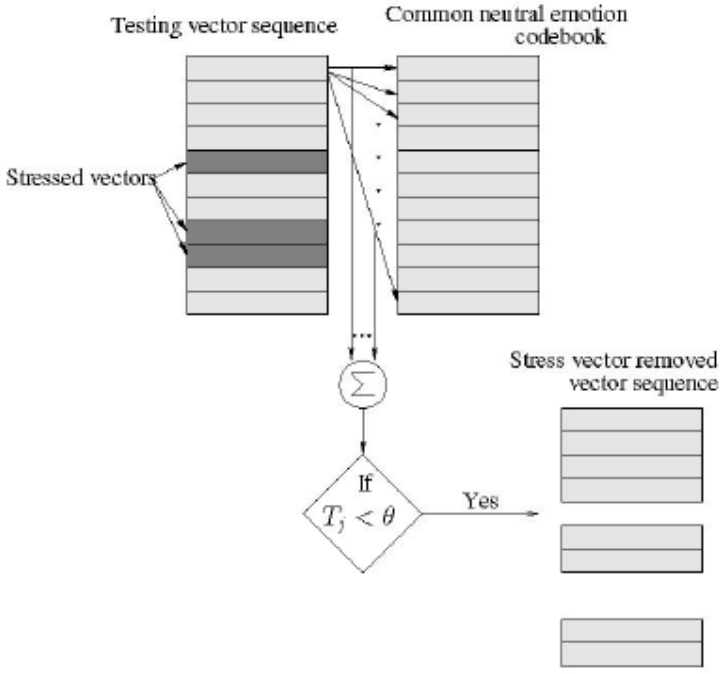
Ambas as configurações obtiveram um melhor desempenho nos teste realizados para um reconhecimento dependente de texto. Os autores atribuiram este resultado ao fato que, em um sistema dependente de texto tanto a base de treinamento como a de teste são iguais, aumentando as chances de identificar corretamente o locutor. Por fim, concluiu-se que a utilização dos MFCC para a extração das características do sinal de voz foi mais eficiente nos testes realizados que o uso do LPC.

3.3 Stress Compensation for Improvement in Speaker Recognition

Em Raja e Dandapat (2006), os autores propuseram em seu trabalho três diferentes técnicas para redução do efeito do estresse ou emoção na voz, com o objetivo de melhorar o reconhecimento de locutor. Os autores consideram o estresse como sendo um sinal de áudio gravado sobre qualquer condição que altere a produção da voz de uma condição neutra. As causas dessas variações podem estar relacionadas com fatores emocionais do locutor ou até mesmo fatores externos, como ruídos.

A primeira fase é responsável por comparar os vetores de caracteristicas da fase de teste com um conjunto de vetores padrões, calculados anteriormente em um ambiente neutro, ou seja, sem ruído. Com base nesta comparação, é a calculada a distância entre os dois vetores. Caso essa distância for maior que um determinado *threshold* de estresse, este vetor é separado dos demais. Este threshold foi determinado pelos autores atráves de experimentos. Na Figura 9 observa-se este fluxo da separação dos vetores com um nível superior de estresse.

Figura 9 - Identificação dos vetores estressados



Fonte: Adaptado de (RAJA; DANDAPAT, 2006)

Os vetores que tiveram um threshold de estresse maior que o limite estipulado são compensados. A compensação funciona atraves do ajustes dos valores destes vetores de tal forma que sejam neutralizados. Para isso, mais um threshold foi definido, visando identificar se o vetor estressado está demasiado excitado ou suprimido. Desta forma, a compensação será realizada pela subtração da média das distâncias em um vetor excitado ou pela soma da média das distâncias em um vetor suprimido.

As técnicas utilizadas para extração das caracteristicas e treinamentos dos modelos foram, respectivamente, MFCC e VQ. Com o sistema de compensação os autores conseguiram diminuir o efeito do estresse em um sinal alcançando uma taxa de reconhecimento de 92,5% em comparação a 84,65% sem a utilização do método.

3.4 Research and Realization of Speaker Recognition Based on Embedded System

O trabalho de Fan *et al.* (2010) propõe o desenvolvimento de uma aplicação de reconhecimento de locutor em um sistema embarcado. Para que isso seja possível, varias limitações de hardware devem ser analisadas, pois geralmente este tipo de sistema tem um alto custo computacional. Porém, com o avanço da microeletrônica e circuitos integrados, os processadores têm evoluído de tal forma que esse tipo de aplicação se torna possível.

O sistema de reconhecimento desevolvido utiliza a técnica de *Linear Prediction Cepstrum Coefficient* (LPCC) para a extração de características e *Support Vectors Machines* (SVM) para classificação dos padrões. LPCC é uma variação do método de *Linear Predictive Coding* (LPC), que analisa os coeficientes ceptrais do sinal. Enquanto, que SVM é uma técnica de aprendizado de máquina (FAN, 2010).

Essas duas técnicas foram embarcadas em um microprocessador *ARM,* com o sistema operacional *Linux*. Esta configuração foi capaz de oferecer todo o poder computacional necessário para esta aplicação, alcançando uma taxa de acerto de 88,44%.

3.5 Text Independent Speaker Recognition Using the Mel Frequenc Cepstral Coefficientes and a Neural Network Classifier

Em Seddik *et al.* (2004), os autores propusseram em seu trabalho um sistema de reconhecimento de locutor independente de texto baseado no uso dos Coeficientes Mel Ceptrais combinados com uma rede neural multicamada. O processo de extração das características do sinal de voz é realizado através da técnica de MFCC, permitindo dessa forma, diferenciar um locutor do outro.

Inicialmente, foi criado um banco de dados de teste e treinamento, com sentenças e fonemas extraídas de 30 locutores. Os autores definiram 7 sentenças de teste e 3 sentenças de aprendizado. Após cada locutor pronunciar estas sentenças, foram extraídos 9600 fonemas para a base de teste, e 4800 fonemas para a base de treinamento.

Para cada locutor são extraídos os fonemas e aplicado os filtros para a extração dos MFCC´s. Como resutado obteve-se uma matriz composta por todos os MFCC´s extraídos para cada fonema. Esta matriz gerada é utilizada como entrada para a rede neural. A rede neural foi treinada utilizando o algoritmo de *backpropagation* com a base de dados de treinamento e após, validada com a base de testes.

Os autores realizaram vários experimentos a fim de otimizar a rede neural, maximizando a taxa de reconhecimento. Entre algumas alterações realizadas na rede neural, pode-se citar, a alteração do número de neurônios em cada camada e diferentes coeficientes de treinamento.

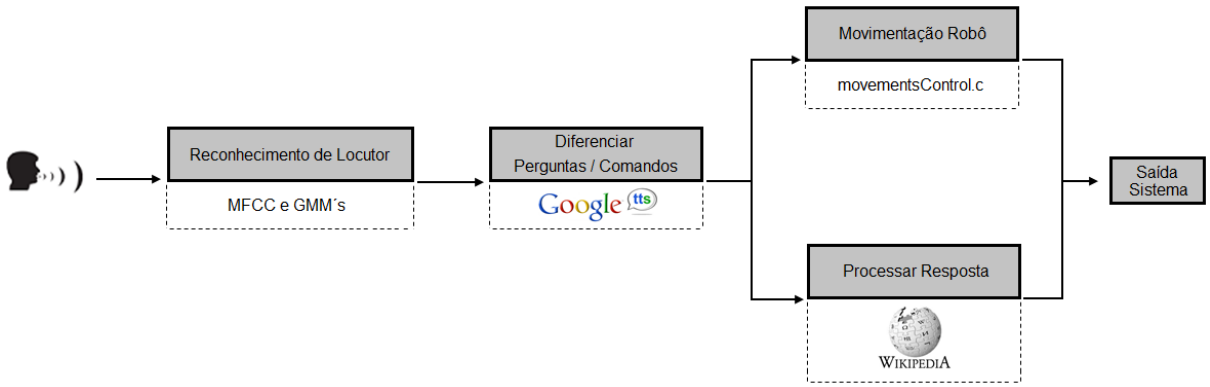
Após realizar diferentes testes, o melhor resultado encontrado foi uma taxa de reconhecimento igual a 77%, alcançada através de uma rede neural de 12 entrada, 45 neurônios na primeira camada e 4 neurônios de saída. Além disso, os autores concluíram que fonemas muito semelhantes não podem ser incluídos na fase de treinamento pois acabam confundindo a rede neural, levando a um resultado errado.

4 FERRAMENTA DESENVOLVIDA

Para alcançar os objetivos deste trabalho, desenvolveu-se uma ferramenta para realizar o reconhecimento de locutor e a comunicação com o robô humanóide. Este capítulo descreve esta ferramenta, seus módulos e as tecnologias utilizadas no seu desenvolvimento.

A ferramenta é composta por quatro diferentes módulos, cada um deles com objetivos específicos para que no final seja gerada uma saída adequada ao usuário. O módulo de reconhecimento de locutor implementa as técnicas de processamento de sinal discutidas nos capítulos anteriores para que, com base nos usuários já cadastrados, identifique ou não o locutor. Após a autenticação do usuário, o comando enviado ao robô é analisado de forma a identificar se é um comando de movimentação ou uma pergunta. Dessa forma, pode-se acionar o módulo de movimentação ou o módulo responsável pelo processamento de uma resposta para a pergunta interpretada. A Figura 10 apresenta a estrutura dos módulos e a relação entre eles.

Figura 10 – Estrutura da ferramenta desenvolvida



Fonte: O Autor

A entrada do software é o sinal de áudio produzido pelo usuário que é captado pelo microfone existente na estrutura do robô humanoide. A saída pode ser observada através da reação do robô ao comando interpretado e também pela interface gráfica do software desenvolvido. Essa interface exibe de forma simples todas as fases do processamento do sinal até a geração da saída.

A plataforma de desenvolvimento escolhida foi a linguagem *Python* e *C++*. Utilizou-se *Python* para o desenvolvimento dos módulos de captura da voz, reconhecimento de locutor e processamento da resposta. Para o controle dos movimentos do robô, optou-se pela linguagem *C++*, por oferecer um conjunto de bibliotecas que facilitam este controle.

4.1 Módulo Reconhecimento de Locutor

O módulo reconhecimento de locutor tem como objetivo reconhecer o usuário que está interagindo com o robô. Para isso, esse módulo utiliza as técnicas descritas no capítulo 2 para processar o áudio recebido como entrada e identificar o usuário. Após a identificação, uma camada de segurança é adicionada ao processamento. Apenas o usuário cadastrado como administrador tem permissão de interagir com o robô. Para os demais, o sistema irá alertar que a autenticação não foi completada com sucesso.

Esse reconhecimento é dividido em duas fases. A primeira fase é responsável por treinar o sistema com os áudios dos usuários a serem reconhecidos. Para cada usuário, existe um arquivo de áudio com uma narração qualquer gravada pelo mesmo. O sistema realiza a leitura desse arquivo, extraindo as suas características e realizando o treinamento de um GMM. Como resultado, para cada usuário a ser reconhecido, o sistema terá um modelo GMM armazenado, que representa as suas características vocais.

A segunda fase é responsável por comparar as características extraídas do sinal de áudio capturado em tempo de execução com a base de modelos cadastrada. Toda vez que o usuário se comunicar com o robô, são extraídas as características deste sinal de áudio capturado. Após, essas características são aplicadas em cada um dos modelos GMM cadastrados e o modelo que mais se aproximar dessas características será considerado o locutor.

4.2 Módulo Identificar Comandos/Perguntas

Após identificar o locutor, a próxima fase é interpretar o sinal de áudio para diferenciar comandos de perguntas. Para isso, utiliza-se a técnica de *speech-to-text* para transformar a linguagem em texto. Esta tarefa é realizada utilizando a *API* do *Google Translate.*

A diferenciação de comandos e perguntas é feita através da comparação do texto extraído com o conjunto pré-definido de comandos. Dessa forma, é possível identificar os comandos de movimentação do humanóide e encaminhar para o módulo responsável por executá-lo. Esta comparação não é realizada de forma absoluta, ou seja, o usuário não precisa pronunciar exatamente um comando específico, pois são utilizadas palavras chaves para identificar os diferentes comandos. Esta abordagem traz uma maior flexibilidade ao se comunicar com o humanoide, porém, aumenta a chance de perguntas que possuem as palavras chaves serem confundidas com comandos de movimentação.

A Tabela 2 apresenta a sequência de palavras chaves a serem identificadas na fala do locutor e o respectivo comando enviado para o módulo de movimentação. Através desta tabela pode-se visualizar o conjunto de comandos de movimentação aceitos pela ferramenta.

Tabela 2 - Lista de palavras chaves e comandos de movimentação

Palavras Chaves	Comando Enviado para o Módulo de Movimentação
Posição Inicial	Initial
Move braço esquerdo cima	Armleftup
Move braço direito cima	Armrightup
Move braço direito baixo	Armrightdown
Move cabeça cima	Headup
Move cabeça baixo	Headdown
Move cabeça esquerda	Headleft
Move cabeça direita	Headright
Caminhar frente	Walkup
Caminhar trás	Walkdown
Caminhar esquerda	Walkleft
Caminhar direita	Walkright
Chutar perna direita	Kickright
Chutar perna esquerda	Kickleft

Fonte: O Autor

Além de comandos que permitem a movimentação de diferentes partes do corpo do robô, criou-se o comando Initial, o qual move o robô para uma posição inicial pré-configurada. Na Figura 11 pode-se visualizar esta posição inicial.

Figura 11 - Posição inicial do robô humanoide

Fonte: O Autor

4.3 Módulo Movimentação Robô

O módulo de movimentação do robô é responsável por movimentar o humanoide de acordo com os comandos recebidos. Para realizar essa tarefa utiliza-se um conjunto de bibliotecas do DarWin-OP na linguagem C++. Estas bibliotecas oferecem métodos que facilitam a movimentação de cada um dos servo motores.

A comunicação com o módulo anterior ocorre através de uma conexão cliente-servidor. Esta conexão é necessária por se tratar de módulos independentes e desenvolvidos em plataformas diferentes. Este módulo permanece constantemente a espera de um comando para realizar a movimentação. Cada comando recebido ativa um método responsável por controlar os servo motores do robô produzindo o movimento desejado.

4.4 Módulo Processar Resposta

O objetivo deste módulo é gerar uma resposta adequada à pergunta feita pelo locutor de forma simples e flexível. Para tal, esta resposta pode ser gerada basicamente de duas formas. Primeiramente, deve-se verificar se a pergunta faz parte de um conjunto de perguntas pré-estabelecidas, para quais o sistema já possui uma resposta padrão. São perguntas básicas, como por exemplo, perguntar qual a hora, qual o dia ou pedir para o robô se apresentar. Para tais perguntas, o robô sempre apresentará o mesmo padrão de resposta.

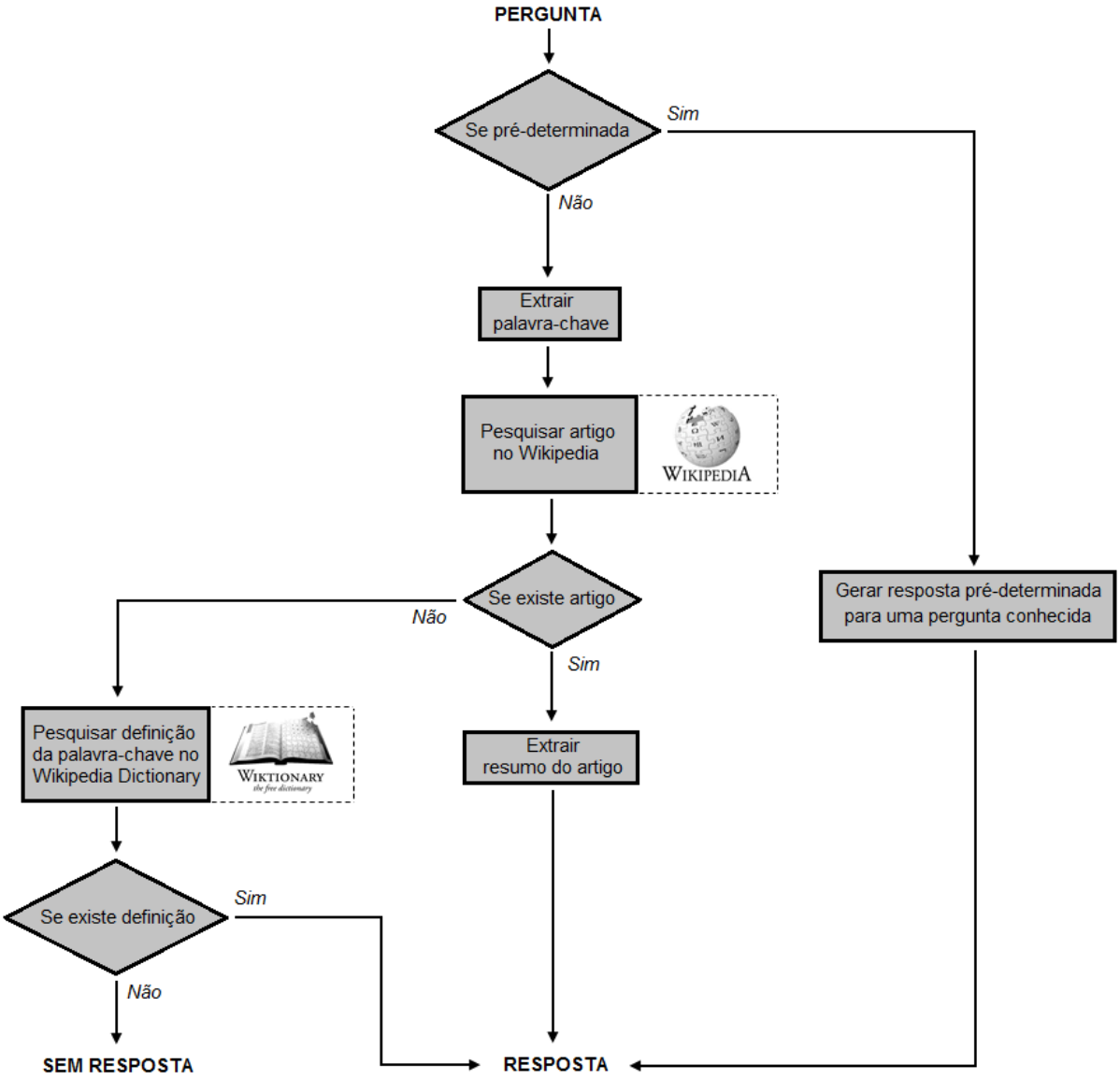
Quando tratar-se de uma pergunta desconhecida então o sistema tenta processar uma resposta apropriada. O processo para gerar esta resposta é dividido em duas etapas. Na primeira fase é necessário encontrar a palavra chave, isto é, o que deseja saber. Para tal, tem-se como premissa que o usuário irá formular a pergunta seguindo o padrão: "*Darwin, o que é <palavra-chave>"*. Pensando nisso, nesta etapa são excluídos do texto capturado, todos os artigos, algumas preposições e palavras como "Darwin", para que no fim reste apenas a palavra chave que representa o que o usuário deseja saber.

A segunda fase é responsável por gerar uma definição para a palavra chave identificada na fase anterior. Para tal, tenta-se encontrar um artigo na base do Wikipédia (WIKIPEDIA, 2015) sobre esta palavra chave. Caso exista este artigo, então parte da seção de resumo da página é extraída e esta passa a ser a resposta para a pergunta realizada. Porém, nem sempre é possível encontrar um artigo a respeito da palavra chave na base do Wikipédia. Nesses casos, tenta-se encontrar uma definição desta palavra no dicionário do Wikipédia, o *Wikipedia Dictionary*.

As consultas nas bases de artigos do Wikipédia e Wikipedia Dictionary são realizadas de formas diferentes. A primeira é através de uma API, também nomeada de Wikipedia (WIKIPEDIA PYTHON, 2015) que facilita a busca e extração de conteúdos dos artigos presentes no site. Já a extração do conteúdo do Wikipedia Dictionary é realizado através do acesso direto a página com a definição da palavra chave. Caso exista uma página no Wikipedia Dictionary a respeito desta palavra chave, então é extraído sua definição, através da respectiva *tag html* com este conteúdo.

Caso o sistema não encontre um artigo ou uma definição para a palavra chave, será emitido uma resposta padrão informando o usuário que não foi possível gerar uma saída adequada. Na Figura 12, apresenta as etapas de processamento da resposta.

Figura 12 - Etapas do processamento da resposta



Fonte: O Autor

Após gerar a resposta é necessário fazer uso da *API* de síntese de voz através do método de *text-to-speech* para transformar esta resposta processada em fala. Por fim, o arquivo de áudio com a narração da resposta pode ser executado através do *speaker* presente na estrutura do robô.

4.5 Interface

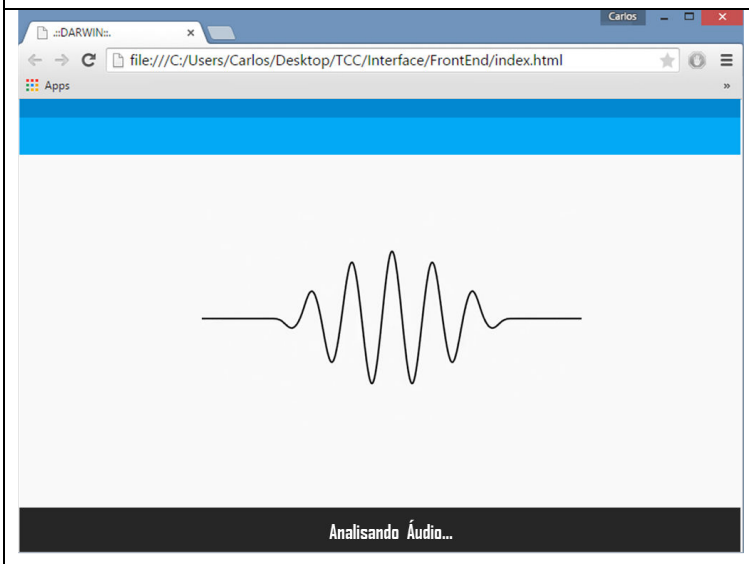
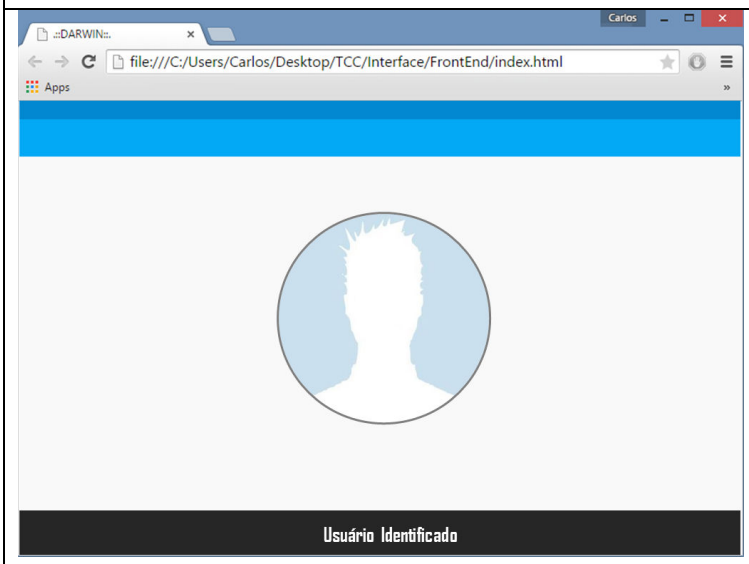
A ferramenta desenvolvida possui um interface de simples visualização e usabilidade. Projetada para exibir ao usuário, o status atual da aplicação permitindo acompanhar cada uma das fases, desde a entrada de áudio até a resposta emitida pelo robô humanóide. Desenvolvida em *HTML/CSS* permite a sua visualização através de um *browser* tanto em um computador pessoal, como em um dispositivel móvel.

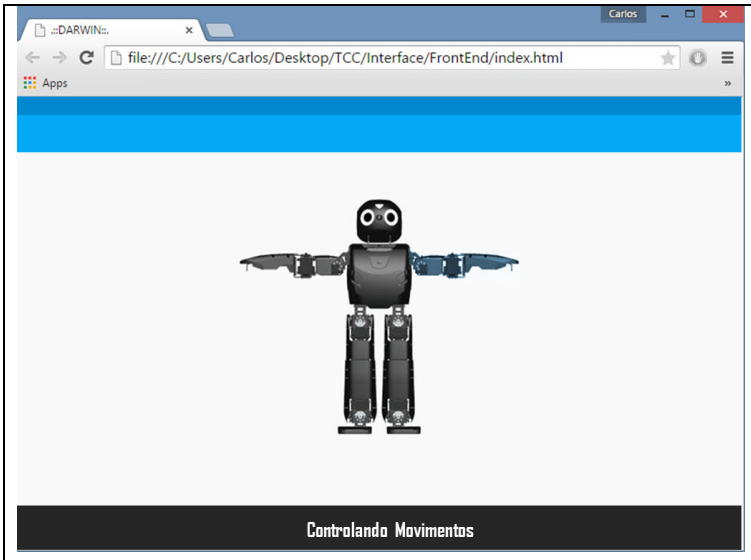
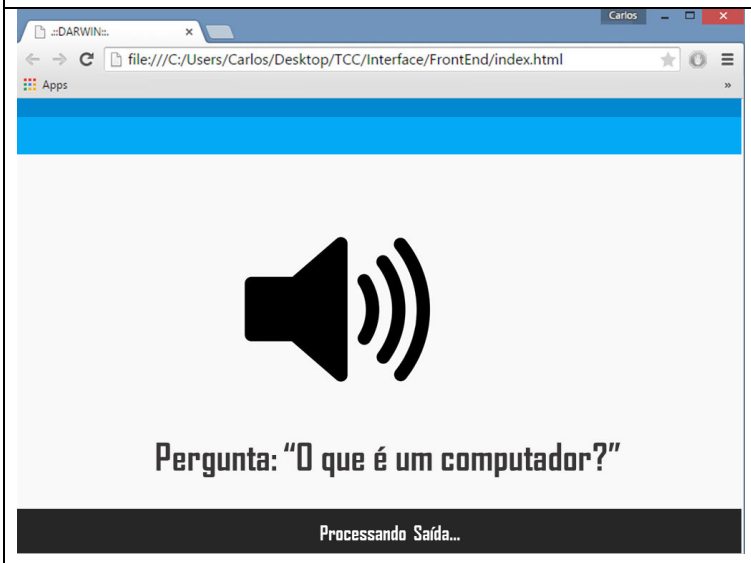
Inicialmente, a interface exibe a tela inicial, enquanto o sistema aguarda pela fala do usuário. A partir do momento que é capturado algum sinal de áudio, a tela de análise de sinal é exibida. O sistema exibe esta tela enquanto os métodos de reconhecimento de locutor e processamento da resposta são executados. Caso o locutor for reconhecido, uma foto do mesmo é exibida, informando ao usuário que a autenticação foi completada com sucesso, caso contrário, uma mensagem será exibida e o software voltará para a tela inicial à espera de um novo comando de voz.

A próxima etapa é dividida em duas diferentes tarefas: exibir para o usuário o movimento que o robô está executando ou emitir a resposta para a pergunta realizada. Para tal, sempre que for um comando de movimentação, uma tela com a representação física do humanóide será exibida, ressaltando a parte do corpo que está sendo controlada. No caso de perguntas, a tela exibirá a pergunta realizada pelo usuário e permanecerá nela enquanto o robô reproduz a resposta através da fala.

Após realizar todas essas etapas, a ferramenta exibe novamente a tela inicial a espera de um novo comando de voz. A Tabela 3 apresenta o fluxo de todas as telas da interface. Nela pode-se visualizar toda a seqüência de execução descrita anteriormente.

Tabela 3 - Fluxo de execução e telas da interface desenvolvida

	**Tela Inicial**
	**Análise do sinal de áudio**
	**Identificação do locutor**

	**Movimentação do robô humanóide**
	**Processar resposta**

Fonte: Do Autor

Este capítulo abordou detalhadamente como foi projetada e desenvolvida a ferramenta. As tecnologias utilizadas por cada módulo e a integração entre eles também foram abordados. No próximo capítulo serão apresentados os resultados.

5 RESULTADOS

Este capítulo aborda os testes realizados e os resultados obtidos neste Trabalho de Conclusão de Curso. Os testes foram realizados separadamente, sobre cada módulo do sistema e após sobre a ferramenta completa. As próximas seções apresentam a metodologia e resultados alcançados.

5.1 Reconhecimento de Locutor

A primeira fase do sistema de reconhecimento é a extração das características e treinamento dos modelos para cada locutor que se deseja reconhecer. Este treinamento é necessário para que o sistema possa comparar as características extraídas em tempo real com a base de treinamento. Para tal, foi necessário criar uma base de treinamento com arquivos de áudio gerados a partir da fala de cada locutor. A base de áudios de treinamento foi composta por cinco locutores de diferentes idades e sexo. As idades dos locutores variam de 4 a 65 anos, o que oferece uma boa heterogeneidade das características do sinal de voz.

O áudio de cada locutor foi capturado em um ambiente controlado. Para tal, criou-se um ambiente livre de ruídos externos, como por exemplo, música, barulho de carros, ou até mesmo outras pessoas falando. Optou-se por realizar a validação em um ambiente controlado, pois qualquer ruído pode interferir diretamente na extração de características da voz, diminuindo assim, a taxa de acerto.

Para capturar os áudios de treinamento, foi solicitado para cada locutor pronunciar um determinado texto durante aproximadamente 20 segundos. Foi realizado testes com gravações maiores, porem o desempenho diminuiu nesses cenários. Atribui-se esse comportamento ao fato de que os áudios de teste são muito curtos em relação aos áudios de treinamento, uma média de 5 segundos para realizar uma pergunta ou comando. Portanto, a quantidade de características extraídas do áudio de teste é muito menor, aumentando a probabilidade de encontrar características similares em outros locutores. Os melhores resultados foram encontrados nos cenários em que se usou arquivos de treinamentos com tamanho próximo aos arquivos de teste.

Por se tratar de um sistema de reconhecimento de locutor independente de texto, o texto pronunciado é completamente aleatório e não tem relação nunhuma com a pergunta ou comando a ser pronunciado pelo locutor na fase de teste. Cada um dos locutores recebeu um texto diferente.

Inicialmente o sistema realiza o treinamento de cada um dos cinco locutores, e após aguarda pela captura do áudio do usuáro de teste. Para avaliar o desempenho do módulo de reconhecimento de locutor, definiu-se uma métrica para a taxa de reconhecimento com sucesso. Essa taxa é definida pela razão entre o número de acertos pelo número total de testes realizados.

Para cada locutor da base de treinamento, realizou-se testes alcançando uma taxa de acerto geral de 72%. Essa taxa de acerto é justificada principalmente, por dois motivos. Primeiramente, pelo fato de se tratar de um sistema indendepente de texto. Em sistemas indendepentes de texto a taxa de acerto é prejudicada, pois a similaridade entre o modelo de teste e treinamento é muito menor do que em um sistema dependente de texto. Outro fator importante, é o cenário definido para a captura do áudio. Com amostras curtas de áudio de teste, o sistema não possui uma grande quantidade de características para aplicar sobre cada modelo.

5.2 Reconhecimento da Fala e Processamento da Resposta

O método de *speech-to-text*, utilizado atráves da *API* do *Google Translate*, também demonstrou um bom resultado no reconhecimento do texto pronunciado pelo locutor. A taxa de acerto da pergunta ou comando realizado alcançou 83%. Para alcançar esse resultado foi necessário pronunciar as palavras de forma clara e precisa. Por vezes, o sistema reconheceu apenas parte da frase pronunciada ou trocou as palavras com pronúncias parecidas.

A escolha da enciclopédia virtual *Wikipedia* para o processamento da resposta, trouxe ao sistema uma grande base de conhecimento, sendo possível gerar de forma simples e eficiente uma resposta ao locutor. O consulta a base do *Wikipedia Dictionary* como alternativa secundária de busca, aumentou a probabilidade de gerar com sucesso uma resposta.

O processo de descoberta da palavra chave, também se demonstrou bastante eficiente nos testes que o usuário seguiu o padrão de pergunta estipulado. Foi possível extraír a palavra chave para a busca na enciclopédia virtual.

5.3 Movimentação do robô e Interface

Através do software de simulação *Webots*, foi realizado o desenvolvido e simulação de todos os movimentos do robô. As bibliotecas de movimentação oferecidas por esta plataforma, facilitaram a implementação de cada movimento. O processo de simulação foi fiel ao comportamento real do robô, pois ao transportar o código fonte para robô se alcançou os mesmos resultados. Todos os movimentos programados foram executados com sucesso pelo humanoide.

A interface simples e objetiva do software auxiliou durante todas as fase de teste. A partir dela, foi possível acompanhar o estado atual de processamento dos módulos e visualizar os resultados. A escolha de desenvolvimento da interface em *HTML*, proporcionou uma maior mobilidade a ferramenta, permitido a sua visualização através de qualquer computador pessoal ou *smartfone*.

5.4 Ferramenta

No geral a ferramenta cumpriu o seu objetivo, proporcionando uma interação homem-máquina mais natural. Por vezes, foi possível identificar corretamente a pergunta ou comando pronunciado pelo usuário, porém não foi possível reconhecer corretamente o locutor e vice-versa. A maneira que são pronunciadas as palavras na hora da comunicação com o sistema, tem uma influência maior sobre o módulo de reconhecimento de perguntas e comandos do que sobre o módulo de reconhecimento de locutor. Uma fala não clara, prejudica o reconhecimento das palavras pronunciadas, porém não influência na extração das características do sinal de voz, permitindo o reconhecimento do locutor.

Por conta de limitações de espaço em disco e versão do sistema operacional do robô, não foi possível embarcar a ferramenta por completo. Por este motivo todos

os testes foram realizados em um computador pessoal, exceto os de movimentação que foram testados no próprio robô.

Para a instalação da ferramenta desenvolvida, é necessário instalar alguns pacotes adicionais. Estes pacotes são necessários para algumas funções matemáticas, execução dos arquivos de áudio e a busca dos artigos no *Wikipedia*. Não foi possível instalar estes pacotes na versão *Linux* de fábrica do robô, por se tratar de uma versão descontinuada e incompatível com esses pacotes. Para resolver este problema, tentou-se instalar uma versão mais atual do sistema operacional *Linux* no robô, porem com uma capacidade de disco de apenas 4 GB, não foi possível realizar esta instalação e também instalar todos os pacotes necessários. Por conta dessa dificuldade, optou-se por validar o sistema em um computador pessoal, o que não altera o comportamento e resultados do sistema, exceto em questões de desempenho.

6 CONCLUSÃO

No trabalho “Sistema de Reconhecimento de Locutor Integrado a comunicação e controle de um robô humanoide” foi proposto o desenvolvimento de um sistema para o reconhecimento de locutor alinhado a uma comunicação com um robô humanoide. A principal contribuição deste trabalho é realizar uma interação homem-máquina mais amigável com o robô humanoide, permitindo que o usuário estabeleça uma comunicação por voz de forma intuitiva.

Através dos testes realizados foi possível concluir os objetivos estabelecidos foram alcançados com sucesso. Além de identificar o locutor, o sistema foi capaz de processar a fala e gerar uma resposta adequada para a maioria dos testes realizados. O sucesso dessas etapas permitiu o uso da ferramenta pela voz, otimizando a experiência do usuário na comunicação homem-máquina.

A incidência de falsos positivos prejudicou o desempenho do sistema. Por vezes, não foi possível reconhecer corretamente o locutor. Em alguns desses casos, o locutor administrador foi reconhecido erroneamente, fazendo com que o sistema permitisse o acesso ao demais módulos e causando portanto, uma falha de segurança.

A forma com que a pergunta ou comando é pronunciada, influência diretamente no reconhecimento da fala. Uma voz clara e precisa aumenta as chances de reconhecer corretamente o texto falado pelo usuário. Porém, mesmo com uma fala clara, pode ocorrer erros neste processo. Um deles acontece ao substituir a palavra falada por outra, com uma pronúncia parecida. Palavras com pronúncia parecida podem ser confundidas pelo sistema, gerando uma resposta inesperada ao usuário.

O tempo de processamento para gerar uma resposta ao usuário está diretamente relacionado com a conexão a internet. É necessário fazer o upload e download de informações em três diferentes etapas, no processo de speech-to-text, processamento da resposta e text-to-speech. Por conta disso, o desempenho depende basicamente da conexão a internet.

Trabalhos futuros incluem a análise de outros métodos de reconhecimento de locutor, realizando uma comparação dos resultados sobre o cenário prosposto. Os trabalhos apresentados no Capítulo 3 apresentam algumas alternativas de métodos. Outra abordagem interessante seria melhorar o módulo de processamento de

perguntas, permitindo a entrada de perguntas mais complexas, tornando o sistema mais flexível. A combinação de outros mecanismos de busca, poderia ser explorado, a fim de gerar respostas mais precisas.

REFERÊNCIAS

AWAIS, M.; HENRICH, D. *Human-Robot Collaboration by Intention recognition using Probabilistic State Machines*. 19th International Workshop on Robotics in Alpe-Danube Region, Anais… p. 75-80, Budapest. 2010

DARwIn-OP. 2004-2015. *Apresenta informações, manuais e fórum sobre a plataforma*. Disponível em <http://support.robotis.com/en/product/darwin-op.htm/>. Acesso em: 17 de Junho de 2015.

DHINESH, G. R.; JAGADEESH, G. R.; Srikanthan, T. *A Low-Complexity Speaker-and-Work Recognition Application for Resource-Constrained Devices.* International Symposium on Electronic System Design, Anais… p. 335-340, Índia. 2011.

FAN, S.; SUN, H.; YU, M. *Research and Realization of Recognition Based on Embedded System.* International Conference On Computer Design and Applications, vol. 1, p. 402-406, China. 2010.

FARAH, S.; SHAMIM, A. *Speaker Recognition System Using Mel-Frequency Cepstrum Coefficients, Linear Coding and Vector Quantization.* Computer, Control & Communication, Anais… p. 1-5, Karachi. 2013.

GOOGLE TRANSLATE API. 2015. *Apresenta informações e manuais sobre a plataforma*. Disponível em <https://cloud.google.com/translate/docs>. Acesso em: 21 de Junho de 2015.

MATHWORKS. 2015. *Apresenta informações e manuais sobre a plataforma Matlab*. Disponível em < http://www.mathworks.com/>. Acesso em: 21 de Junho de 2015.

MOON, S. *The Expectation-maximization algorithm.* Signal Processing Magazine, vol. 13, p. 40-60, USA. 2002.

LAXMAN, S.; SASTRY, P. S. *Text-dependent speaker recognition using speaker specific compensation.* Conference On Convergent Technologies for the Asia-Pacific Region, vol. 1, p. 384-387, Índia. 2003.

OPPENHEIM, A. V.; SCHAFER, R. W. *Discrete-Time Signal Processing.* Prentice Hall*,* p. 796, 1999.

RABINER, L. R.; JUANG, B. *Fundamentals of Speech Recognition.* Prentice Hall*,* p. 493, 1993.

RAJA, G.; DANDAPAT, S. *Stress Compesantion for Improvement in Speaker Recognition.* India Conference, Anais… p. 1-5, New Delhi. 2006.

REYNOLDS, D. *Speaker identification and verification using Gaussian Mixture Models.* Speech Communications, vol. 17, p. 91-108, 1995

REYNOLDS, D.; HECK, L. P. *Automatic speaker recognition: recent progress, current applications and future trends.* American association for the advancement of science symposium. Washington. 2000

SEDDIK, H.; RAHMOUNI, A.; SAYADI, M. *Text independent speaker recognition using the Mel frequency cepstral coefficients and a neural network classifier.* First International Symposium on Control, Communications and Signal Processing, Anais… p. 631-634, Tunísia. 2004.

SHIN, S.; MATSON, E.; JINOK, P.; YANG, B. *Speech-to-speech translation humanoid robot in doctor´s office.* Automation, Robotics and Applications, Anais… p. 484-489, Queenstown. 2015.

WEBOTS. 1996-2014. *Apresenta informações, manuais e fórum sobre a ferramenta de simulação*. Disponível em <http://cyberbotics.com/>. Acesso em: 21 de Junho de 2015.

WIKIPEDIA. 2001-2015. *Uma enciclopédia livre*. Disponível em <https://wikipedia.org>. Acesso em: 21 de Junho de 2015.

WIKIPEDIA PYTHON. 2015. *Apresenta informações, manuais e fórum sobre a API*. Disponível em <https://pypi.python.org/pypi/wikipedia/>. Acesso em: 21 de Junho de 2015.