



WILLIAM MACHADO

**SISTEMA DE IDENTIFICAÇÃO DE DIREÇÃO DE FONTE
SONORA PARA AUXILIAR DEFICIENTES AUDITIVOS**

LAVRAS – MG

2020

WILLIAM MACHADO

**SISTEMA DE IDENTIFICAÇÃO DE DIREÇÃO DE FONTE SONORA PARA
AUXILIAR DEFICIENTES AUDITIVOS**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação, área de concentração em Engenharia de Sistemas e Automação, para a obtenção do título de Mestre.

Prof. DSc. Wilian Soares Lacerda
Orientador

LAVRAS – MG
2020

**Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca
Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).**

Machado, William

Sistema de identificação de direção de fonte sonora para
auxiliar deficientes auditivos / William Machado. – Lavras :
UFLA, 2020.

120 p. :

Dissertação(mestrado acadêmico)–Universidade Federal de
Lavras, 2020.

Orientador: Prof. DSc. Wilian Soares Lacerda.

Bibliografia.

1. Rede Neural Artificial. 2. Detector de Sinal Sonoro.
3. Somatório das Diferenças Absolutas. 4. Acessibilidade. I.
Lacerda, Wilian Soares. II. Título.

WILLIAM MACHADO

**SISTEMA DE IDENTIFICAÇÃO DE DIREÇÃO DE FONTE
SONORA PARA AUXILIAR DEFICIENTES AUDITIVOS**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação, área de concentração em Engenharia de Sistemas e Automação, para a obtenção do título de Mestre.

APROVADA em 20 de fevereiro de 2020.

Prof. DSc. Erivelton Geraldo Nepomuceno UFSJ
Prof. DSc. Danton Diego Ferreira UFLA

Prof. DSc. Wilian Soares Lacerda
Orientador

LAVRAS – MG

2020

Dedico à minha família, pois sem eles este trabalho e muitos dos meus sonhos não se realizariam.

AGRADECIMENTOS

Agradeço, primeiramente, à Deus que me deu saúde e sabedoria para seguir sempre em frente. Pela força e ser meu guia em todos os momentos para concluir este trabalho.

Aos meus pais, Aroldo e Nicésia e meus avós, Aroldo e Haidê, pelo apoio e incentivo em todos os momentos em minha vida, por acreditarem em mim, e não medirem esforços para a concretização dos meus sonhos.

Ao meu professor, orientador e amigo Wilian Soares Lacerda, que sempre foi um companheiro desde a graduação, pela paciência e orientação. Pelos puxões de orelha, ensinamentos e toda a experiência compartilhada comigo nesse tempo.

Ao meu primo Guilherme pelas ajudas nos feriados e finais de semana, mesmo na correria do dia-a-dia me ajudando nas dúvidas de programação e desenvolvimento dos algoritmos.

Ao meu irmão Frederico, e todos os meus amigos que estiveram em meu caminho, compartilhando esse momento direta ou indiretamente.

À minha namorada Laura, pela paciência e companheirismo, pelo apoio em vários momentos de dificuldade durante minha pós-graduação.

Enfim, agradeço a todas as pessoas que fizeram parte dessa etapa decisiva em minha vida. Muito obrigado por tudo!

"Pouco conhecimento faz com que as pessoas se sintam orgulhosas. Muito conhecimento, que se sintam humildes. É assim que as espigas sem grãos erguem desdenhosamente a cabeça para o céu, enquanto que as cheias as baixam para a terra."

(Leonardo da Vinci)

RESUMO

Existem milhares de pessoas no mundo portadoras de deficiência e que possuem necessidades especiais. Com o desenvolvimento urbano, surgem as dificuldades de locomoção nas vias públicas e de acesso aos transportes públicos. Cada tipo de deficiência exige certo tipo de adaptação para o indivíduo poder realizar suas tarefas diárias. Um problema muito comum no dia-a-dia dos deficientes auditivos é a dificuldade de detectar a direção de sinais sonoros provenientes de fontes sonoras. Muitas vezes, os deficientes auditivos não conseguem distinguir de qual lado (direito ou esquerdo) um alerta sonoro foi emitido. Com o objetivo de oferecer uma alternativa para este problema, o presente trabalho propõe um sistema que auxilia o deficiente auditivo. O sistema baseia-se no defasamento temporal do sinal sonoro captado por dois sensores de som (microfones), onde cada sensor é localizado no ombro do usuário. Os sinais sonoros captados do ambiente pelos sensores terão um defasamento temporal entre si devido à diferença da distância da fonte sonora com os sensores (microfones). Este defasamento será calculado pelo software executado em um microcontrolador que determinará a direção (direita ou esquerda) da fonte sonora em relação ao usuário de acordo com o defasamento obtido. O microcontrolador, de posse desta informação, ativará o motor vibracall colocados em cada ombro do usuário para alertá-lo da direção proveniente do som. Os resultados obtidos através da simulação mostraram-se promissores, uma vez que a Rede Neural *One Step Secant (OSS)* apresentou 96,3% de acertos na detecção da direção de diferentes tipos de fontes de sinal sonoro. O teste da simulação do algoritmo utilizando Somatório das Diferenças Absolutas apresentou 100% de acerto e os testes em ambientes reais mostraram uma taxa de acerto de até 85,5%. Deste modo, o desenvolvimento do trabalho preza pela acessibilidade, pelo seu caráter social, baixo custo e inclusivo, pois com ele é possível tornar acessível e auxiliar o deficiente auditivo na locomoção no ambiente urbano evitando acidentes, além de proporcionar conforto e sensação de pertencimento dos espaços nos quais todos circulam.

Palavras-chave: Rede Neural Artificial, Detector de Sinal Sonoro, Somatório das Diferenças Absolutas, Acessibilidade, Fonte Sonora, Direção Sonora.

ABSTRACT

There are thousands of people in the world with disabilities who have special needs. With urban development, difficulties arise in locomotion on public roads and access to public transport. Each type of disability requires a certain type of adaptation for the individual to be able to perform their daily tasks. A very common problem in the day-to-day life of the hearing impaired is the difficulty in detecting the direction of sound signals from sound sources. Often, the hearing impaired cannot distinguish which side (right or left) an audible alert was issued. In order to offer an alternative to this problem, the present work proposes a system that assists the hearing impaired. The system is based on the time lag of the sound signal captured by two sound sensors (microphones), where each sensor is located on the user's shoulder. The sound signals captured from the surroundings by the sensors will have a time lag between them due to the difference in distance between the sound source and the sensors (microphones). This lag will be calculated by the software running on a microcontroller that will determine the direction (right or left) of the sound source in relation to the user according to the acquired lag. The microcontroller, in use of this information, will activate the vibracall engine placed on each shoulder of the user to alert him of the direction coming of the sound. The results obtained through the simulation proved to be promising, since the Neural Network *One Step Secant (OSS)* presented 96.3% of accuracy in detecting the direction of different types of sound sources. The algorithm simulation test using the Sum of Absolute Differences showed 100% accuracy and the tests in real environments showed a accuracy rate of up to 85.5%. Thus, the development of work values accessibility, its social character, low cost and inclusive, as it is possible to make accessible and assist the hearing impaired in locomotion in the urban environment, avoiding accidents, in addition to providing comfort and a sense of belonging to spaces in which everyone circulates.

Keywords: Artificial Neural Network, Sound Detector, Sum of the Absolute Differences, Accessibility, Sound Source, Sound Direction.

LISTA DE FIGURAS

Figura 2.1 – Tipos de ondas sonoras	18
Figura 2.2 – Defasamento de sinais	20
Figura 2.3 – Exemplos de sinais: (a) sinal com energia finita (b) sinal com potência finita	25
Figura 2.4 – Exemplo da amostragem de um sinal	28
Figura 2.5 – Respostas ideais em frequência para os quatros tipos de filtro	32
Figura 2.6 – Filtro Passa-baixa	33
Figura 2.7 – Resposta ideal e real em frequência de um filtro passa-baixa	33
Figura 2.8 – Angulo entre dois vetores	35
Figura 2.9 – Comparação com os lados de um triângulo	36
Figura 2.10 – Linhas de microcontroladores 8 bits da Microchip	40
Figura 2.11 – Esquemático PIC16F688	40
Figura 2.12 – Diagrama de blocos do PIC16F688	42
Figura 2.13 – Interface gráfica do MikroC	43
Figura 2.14 – Modelo proposto por McCulloch e Pitts	45
Figura 2.15 – Camadas de uma Rede Neural	46
Figura 2.16 – Feed-Forward	47
Figura 2.17 – Cascade-Foward	48
Figura 2.18 – Motor vibracall de um celular	50
Figura 2.19 – Motor vibracall de um controle de videogame	51
Figura 2.20 – Formatos do microfone	51
Figura 2.21 – Funcionamento do Microfone de Eletreto	54
Figura 2.22 – Partes do microfone de eletreto	54
Figura 2.23 – Acurácia e Precisão	56
Figura 2.24 – Exemplo de Gráfico ROC	60
Figura 3.1 – Funcionamento do Protótipo	61
Figura 3.2 – Esboço do sistema a ser construido	62
Figura 3.3 – Exemplo de sinal obtido representando som de uma sirene policial	64
Figura 3.4 – Defasamento de sinais	65
Figura 3.5 – Possíveis posições da fonte sonora	67
Figura 3.6 – Fluxograma geral da Simulação	68
Figura 3.7 – Vetores de amostras	69

Figura 3.8 – Matriz de amostras	70
Figura 3.9 – Modelo da rede neural no Matlab®	71
Figura 3.10 – Matriz de saída desejada da RNA	71
Figura 3.11 – Vetores com a amostragem dos sinais	72
Figura 3.12 – Representação dos sinais adquiridos pelos microfones direito e esquerdo na simulação	73
Figura 3.13 – Matriz de amostras	74
Figura 3.14 – Deslocamento do vetor durante o cálculo do SDA	75
Figura 3.15 – Gráfico gerado pelo algoritmo SDA	76
Figura 3.16 – Diagrama do Hardware	77
Figura 3.17 – MAX9814	78
Figura 3.18 – Microcontrolador PIC16F688	80
Figura 3.19 – Motor VibraCall	80
Figura 3.20 – Esquemático da placa central contendo microcontrolador	81
Figura 3.21 – Esquemático da placa que será utilizada em cada ombro	82
Figura 3.22 – Fluxograma do algoritmo de simulação no Microcontrolador	83
Figura 4.1 – Matriz de confusão das etapas do algoritmo GDX	86
Figura 4.2 – Curva ROC do Algoritmo GDX	87
Figura 4.3 – Matriz de confusão das etapas do algoritmo OSS	88
Figura 4.4 – Curva ROC do Algoritmo OSS	89
Figura 4.5 – Matriz de confusão do Algoritmo SDA	90
Figura 4.6 – Curva ROC do algoritmo SDA	90
Figura 4.7 – Placa central contendo Microcontrolador	91
Figura 4.8 – Placa utilizada em cada ombro contendo microfone e motor vibracall	91
Figura 4.9 – Protótipo, composto pelo PIC16F688, Microfone, Motor vibracall, Bateria.	92
Figura 4.10 – Sinal adquirido em um ambiente controlado	93
Figura 4.11 – Posições utilizadas para o teste real	94

LISTA DE TABELAS

Tabela 2.1 – Resumo das características dos filtros	32
Tabela 2.2 – Teorema 1 do Produto Escalar	35
Tabela 2.3 – Matriz de Confusão	58
Tabela 2.4 – Valores Coeficiente Kappa	59
Tabela 4.1 – EMQ da rede <i>Feed-Forward Back-Propagation</i> usando dados de teste . . .	85
Tabela 4.2 – EMQ para topologia de rede <i>Cascade-Forward Back-Propagation</i>	87
Tabela 4.3 – Matriz de Confusão dos resultados obtidos com o som do Tambor	94
Tabela 4.4 – Matriz de Confusão dos resultados obtidos com o som do Apito	95
Tabela 4.5 – Matriz de Confusão dos resultados obtidos com o som senoidal de 500Hz gerados por Celular	95
Tabela 4.6 – Matriz de Confusão dos resultados obtidos com o som da Gaita	95
Tabela 4.7 – Matriz de Confusão dos resultados obtidos com o som do Carro	96
Tabela 4.8 – Matriz de Confusão dos resultados obtidos com o som do violão acorde Dó	96
Tabela 4.9 – Matriz de Confusão dos resultados obtidos com o som de palmas	97
Tabela 4.10 – Custo total e dos componentes utilizados no protótipo	97

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Trabalhos Relacionados	13
1.2	Objetivos	14
1.2.1	Objetivo Geral	14
1.2.2	Objetivos Específicos	15
1.3	Justificativa deste trabalho	15
1.4	Estrutura do Trabalho	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Deficiência Auditiva	17
2.2	Ondas Sonoras	18
2.2.1	Propagação Sonora	20
2.2.2	Percepção da direção sonora	22
2.3	Sinais e Sistemas	22
2.3.1	Tamanho do Sinal	23
2.3.1.1	Energia e Potência do Sinal	23
2.3.2	Processamento de sinais	26
2.3.3	Amostragem de sinais	27
2.3.4	Transformada de Fourier	28
2.3.5	Aliasing e Teorema de Nyquist	29
2.3.6	Filtros	30
2.3.6.1	Filtros Passa-baixa	32
2.4	Produto vetorial	34
2.5	Somatório das Diferenças Absolutas de números reais	37
2.5.1	Algoritmo do Somatório das Diferenças Absolutas - SDA	38
2.6	Sistemas Embarcados	38
2.6.1	Microcontrolador	39
2.6.2	Microcontrolador PIC16F688	40
2.7	Compilador MikroC	42
2.8	Redes Neurais Artificiais	44
2.8.1	Neurônio Artificial	44
2.8.2	Arquitetura da Rede Neural	45

2.8.3	Topologia de uma Rede Neural	46
2.8.4	Processos de aprendizado	48
2.9	Motor de Corrente Contínua	49
2.9.1	Motor VibraCall	50
2.10	Microfone	51
2.10.1	Microfone de Eletreto	53
2.11	Métodos de análise de desempenho	55
2.11.1	Acurácia e precisão	55
2.11.2	Erro Quadrático Médio	57
2.11.3	Matriz de Confusão	57
2.11.4	Coefficiente Kappa	58
2.11.5	Gráfico ROC	59
3	METODOLOGIA	61
3.1	Base de Dados	63
3.1.1	Pré-processamento dos Dados	64
3.2	Simulação do detector de direção de som no PC	67
3.2.1	Rede Neural Artificial	68
3.2.2	Somatório das Diferenças Absolutas - SDA	72
3.3	Desenvolvimento do Hardware do Sistema Embarcado	77
3.3.1	Módulo Amplificador Microfone Eletreto MAX9814	77
3.3.2	Filtro passa-baixa	78
3.3.3	Microcontrolador PIC16F688	79
3.3.4	Motor VibraCall	80
3.3.5	Desenvolvimento do esquemático eletrônico	80
3.4	Desenvolvimento do Programa para Microcontrolador	82
3.4.1	Algoritmo SDA	84
3.5	Métricas de Desempenho	84
4	RESULTADOS	85
4.1	Resultados da RNA	85
4.2	Resultados do SDA	89
4.3	Implementação real no Microcontrolador	91
4.4	Teste em ambiente real	93

4.5	Custos	97
5	CONCLUSÃO	99
5.1	Propostas de continuidade	100
	REFERÊNCIAS	101
	APENDICE A – Script do Matlab que implementa o detector de direção do som utilizando RNA	108
	APENDICE B – Simulação do SDA no Matlab	110
	APENDICE C – Simulação do SDA em C	114
	APENDICE D – Implementação do SDA no PIC16F688	116

1 INTRODUÇÃO

A comunidade de surdos luta há anos contra preconceitos. Os surdos são tratados como marginalizados por uma sociedade opressora que não leva em consideração suas necessidades. Através de uma série de equívocos e ações, esses comportamentos injustos pela sociedade levam os surdos ao isolamento e à exclusão. Lenta e dolorosamente, os surdos fizeram algum progresso para serem considerados membros de uma minoria com necessidades especiais, cuja inclusão exige dimensões sociais, políticas e de cidadania (ANTUNES et al., 2011).

Existem milhares de pessoas portadoras de deficiência auditiva, com a expansão das cidades e a urbanização, principalmente, o aumento de automóveis e equipamentos mecânicos tornou o trânsito desorganizado e sujeito a diversos tipos de imprevistos que podem criar problemas para as pessoas, especialmente para os deficientes. O monitoramento do tráfego é um serviço essencial para tais situações. Atualmente, o monitoramento do tráfego é realizado usando técnicas de processamento de vídeo. É fácil analisar o número de veículos que cruzam em uma junção particular. Observa-se que análises de vídeo por si só não são suficientes para o monitoramento do tráfego (LUITEL et al., 2016). Assim, surgem as dificuldades de locomoção dos portadores de deficiência nas vias públicas e de acesso ao transporte público, uma vez que cada tipo de deficiência exige certo tipo de adaptação para as tarefas diárias possam ser realizadas.

Devido a esta desorganização, um dos desafios enfrentados pelos deficientes auditivos é a percepção do sinal sonoro, que pode ser o som do motor de um veículo que se aproxima, uma buzina, um apito de trem, entre outros sons de trânsito. A acessibilidade vem ganhando destaque em várias áreas, há tendência para se concentrar na acessibilidade, em vez de usabilidade da tecnologia para pessoas com deficiência, embora ambas sejam essenciais e complementares, e devem ser consideradas parte de boas práticas de *design* (HERSH, 2008). Com isso, vêm ganhando mais transparência no âmbito acadêmico, de acordo com Santos et al. (2017), a usabilidade e a acessibilidade, que são características a quem deve ser dada especial importância neste processo. Deste modo, as tecnologias que são aplicadas na acessibilidade auxiliam os deficientes e também os indivíduos que não possuem deficiência.

A maioria das tecnologias de detecção de acidentes é utilizada em veículos. De acordo com Rivas et al. (2017), os sistemas de assistência à condução implementados em veículos modernos podem ser aprimorados através do intercâmbio de informações com a rede de sensores. Os requisitos de complexidade dos sistemas dentro do veículo podem ser reduzidos, deixando

tarefas específicas, como a detecção de atropelamento ou o aviso de limite de velocidade a ser executado pela infraestrutura. As tecnologias de transporte visam interconectar veículos e infraestrutura rodoviária para ter informações precisas sobre as condições da estrada o tempo todo. Muitas montadoras e outras empresas estão trabalhando para as tecnologias carro-para-X, onde o X significa carros ou infraestrutura. Nem sempre visam o sistema para ser utilizado por um indivíduo, como pedestre ou ciclista, a fim de evitar acidentes.

Visando melhorar a qualidade de vida das pessoas, principalmente deficientes auditivos, o presente trabalho apresenta o desenvolvimento de um protótipo de um sistema embarcado com microcontrolador e software capaz de detectar a direção da fonte de um sinal sonoro a partir do defasamento entre os sinais captados, e que alerta o usuário a respeito de possíveis perigos que podem ocorrer.

1.1 Trabalhos Relacionados

Buscas na literatura da área para verificar a existência de algum sistema relacionado ao trabalho proposto foram realizadas, encontrando poucos trabalhos e alguns sistemas como sensor de ré veicular e alarme de presença, que é ativado sempre que detecta algum movimento. Estes dispositivos podem ser adquiridos em lojas de eletrônicos, em geral.

Diversos trabalhos já utilizaram inteligência computacional para realizar o processamento de som. O trabalho, proposto por Yu et al. (2016), utiliza uma antena para simular o ouvido humano e detectar micro-ondas sonoras. Uma outra técnica amplamente utilizada é a rede neural, como o trabalho desenvolvido por Vesperini et al. (2016), onde treinou uma rede neural artificial multicamadas alimentada por GCC-PHAT (Transformação de Fase de Correlação Cruzada, em português) com o objetivo de identificar autofalantes ligados em uma sala.

O trabalho proposto por Armansyah et al. (2016), apresenta um algoritmo para localização da fonte sonora, com um *design Very-Large-Scale Integration* (VLSI) para localização da fonte sonora. O sistema criado suporta reconhecimento sonoro no espaço tridimensional, utilizando 4 sensores e a técnica computacional *Time Difference Of Arrival* (TDOA). O sistema é implementado no *Altera Cyclone II FPGA* com codificação *Verilog HDL*.

Os trabalhos apresentados por Luz (2016) e Andrade (2017) exibem métodos práticos variados para se determinar a defasagem entre dois sinais, busca expor os meios necessários para a utilização de cada um, suas vantagens e desvantagens, além de propor combinações entre si e a consequente composição de novos métodos com o mesmo objetivo. Além disso, os

trabalhos demonstram como utilizar a técnica apropriada para exprimir na forma exponencial os sinais analógicos antes de serem processados pelos métodos digitais.

Da mesma forma, os trabalhos apresentados por Silva (2014) e Almeida (2015) explicam formas de como determinar a direção de chegada de um sinal (DOA) aplicado para antenas. Para tanto, descreve de forma sucinta tais métodos, além de expor exemplos práticos com a ajuda de um microcontrolador Arduino e algoritmos computacionais via MatLab.

Há outra aplicação, proposta por Umezawa et al. (2017), que consiste em uma técnica de detecção de veículos silenciosos, como híbridos e elétricos. O funcionamento se porta da seguinte maneira: o veículo transmite um sinal acústico modulado digitalmente que contém a sua identidade – ID – ao transitar, assim o dispositivo móvel do pedestre que estiver com ele registra o sinal no momento em que o veículo se aproxima, alertando a aproximação do veículo em tempo real, medindo o efeito Doppler em cada sinal.

O trabalho proposto por Okuyama et al. (2002), propõe um sistema para detecção de ondas sonoras utilizando 4 microfones, um modelo parecido com as boias utilizadas nos mares, para detecção do terremoto. O autor cria uma simulação na qual o sistema é capaz de determinar a posição da fonte sonora em um ambiente tridimensional, assim podendo encontrar a fonte do terremoto.

Há outras aplicações utilizadas no ambiente urbano, que funcionam como detecção de veículos e também para detecção de acidentes no trânsito, nas quais são utilizados referenciais estáticos tais como câmeras em postes, torres, e há outros ainda que utilizam aviões e drones para análise do tráfego (KURNIANGGORO et al., 2014; ELMIKATY; STATHAKI, 2017a; ELMIKATY; STATHAKI, 2017b; YOON et al., 2016; LAI et al., 2016; BOROWIK et al., 2017).

1.2 Objetivos

Nesta seção serão abordados o objetivo geral e os objetivos específicos do trabalho.

1.2.1 Objetivo Geral

O presente trabalho tem por objetivo principal desenvolver um protótipo de baixo custo para detectar a direção de uma fonte de sinal sonoro a partir do defasamento entre os sinais captados por dois sensores (microfones). Este equipamento poderá ser utilizado por deficientes auditivos para alerta de provável perigo.

1.2.2 Objetivos Específicos

Os objetivos específicos abordados são::

- implementar algoritmos de cálculo de defasamento de sinais utilizando RNA (Rede Neural Artificial) e SDA (Somatório das Diferenças Absolutas);
- testar os algoritmos simulando defasamento de sinais de som obtidos de uma base de dados;
- montar e testar um protótipo que implementa uma das técnicas de cálculo de defasamento de sinais.

1.3 Justificativa deste trabalho

No cenário atual, a acessibilidade dos deficientes auditivos é um grande desafio. De acordo com a Organização Mundial de Saúde (OMS), a quantidade de deficientes auditivos no mundo é de aproximadamente 360 milhões. Estima-se que, um terço desta população, ou um total de 165 milhões é formado por pessoas acima de 65 anos e 32 milhões por crianças e adolescentes com idade igual ou inferior a 15 anos (PONDÉ, 2009; PORSINAL, 2013; EXAME, 2016).

No Brasil, de acordo com o censo de 2010 realizado pelo Instituto Brasileiro de Geografia e Estatística (IBGE), 9,7 milhões de pessoas têm deficiência auditiva. Desses, 2,2 milhões apresentam deficiência auditiva severa, situação em que há uma perda entre 70 e 90 decibéis (dB), cerca de um milhão são jovens de até 19 anos (BRASIL, 2016).

Uma das principais dificuldades dos deficientes auditivos é a identificação da direção sonora, até mesmo para deficientes auditivos que realizam cirurgia para implantes cocleares. Nos trabalhos propostos por (NEUMAN et al., 2007; HOESEL et al., 2002), os autores comparam a precisão da identificação da direção sonora em pacientes com implante coclear bilateral. Foram testadas em três condições auditivas (implantes cocleares bilaterais, implante coclear esquerdo e implante coclear direito), e os sinais utilizados foram com ruídos rosas e estímulos de fala.

Com esse grande número de pessoas enfrentando diversos desafios, surge a possibilidade de desenvolver um sistema para detectar a direção de uma fonte de sinal sonoro a partir de sinais defasados e utilizando técnicas de processamento de sinais. A redução das dificuldades enfrentadas por deficientes auditivos no cotidiano e a melhora da qualidade de vida justifica-se, portanto, voltar a atenção para esse grupo social.

1.4 Estrutura do Trabalho

Este trabalho está organizado em cinco capítulos divididos da seguinte maneira: introdução, fundamentação teórica, metodologia, resultados e conclusão, seguido pelas referências bibliográficas. O primeiro capítulo apresenta a introdução na qual contém os objetivos e a justificativa deste projeto. O segundo apresenta a fundamentação teórica, mostrando os conceitos importantes dos componentes e materiais que serão utilizados para o entendimento do trabalho de uma forma geral. O terceiro apresenta os materiais e métodos empregados no desenvolvimento do sistema proposto. Na sequência, o capítulo 4 apresenta os resultados obtidos e, por fim, o capítulo 5 dispõe a conclusão e sugestões para trabalhos futuros. Posteriormente, as referências bibliográficas que foram consultadas para o desenvolvimento do projeto são listadas. Por fim, os apêndices apresentam os algoritmos desenvolvidos.

2 FUNDAMENTAÇÃO TEÓRICA

O presente capítulo apresenta uma revisão dos assuntos que possuem uma relação com o trabalho.

2.1 Deficiência Auditiva

Para Santos (2012), a deficiência auditiva é o nome usado para identificar perda de audição ou diminuição na capacidade de escutar os sons, ou seja, qualquer problema relacionado à audição, que ocorre em alguma parte do ouvido.

Em muitos casos, o diagnóstico médico consegue identificar a causa mais provável da perda auditiva, mas nem sempre isso é possível. A ocorrência de gestações e partos com histórico complicado, bem como a manifestação de doenças maternas no período próximo ao nascimento da criança, podem inviabilizar a identificação da causa. Em cerca de 50% dos casos, a origem da deficiência auditiva é atribuída a “causas desconhecidas”. Quando se consegue descobrir a causa, o mais frequente é que ela se deva a doenças hereditárias, rubéola materna e meningite (REDONDO; CARVALHO, 2000).

De acordo com Pagliuca et al. (2007), a audição é o sentido por meio do qual se percebem os sons. No caso dos deficientes auditivos, estes podem ser classificados em surdos totais e surdos parciais. O totalmente surdo é aquele em que a audição não é funcional na vida comum; e parcialmente surdo é aquele em que a audição, embora deficiente, é funcional, com ou sem prótese auditiva. Esse tipo de deficiência é bastante frequente, pois segundo estimativas da Organização Mundial de Saúde (OMS), cerca de 15% da população brasileira é portadora de alguma deficiência auditiva.

Conforme Redondo e Carvalho (2000), a deficiência auditiva traz muitas limitações para o desenvolvimento do indivíduo. A sociedade conhece bem pouco os portadores de deficiência. Esse desconhecimento se reflete por exemplo na ausência de estatísticas brasileiras tanto a respeito de seu número real quanto às formas de assistência disponíveis, de sua integração social e de sua inclusão no mercado de trabalho.

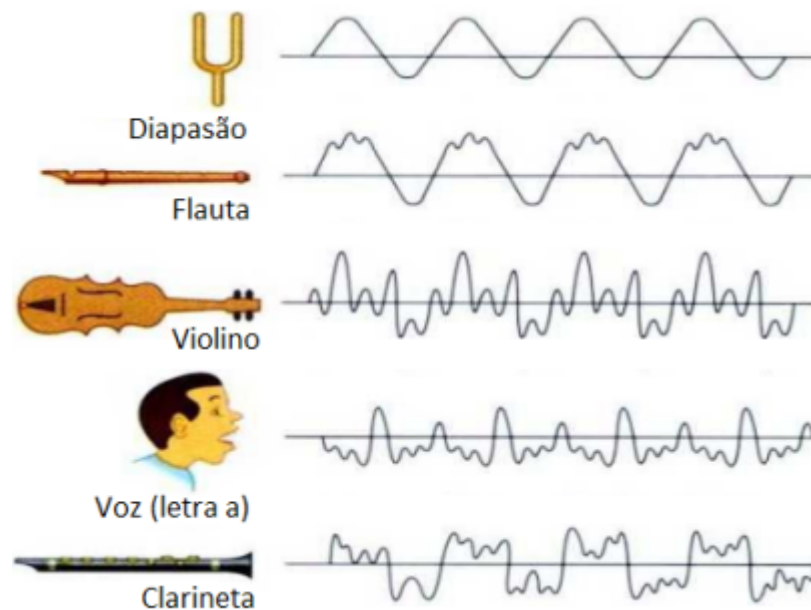
Como citado por Pagliuca et al. (2007), a deficiência auditiva é tratada por muitos autores como o tipo de deficiência de mais difícil convívio com o restante da sociedade. Ao se comparar as pessoas com deficiência física, auditiva e visual, o deficiente auditivo é o que enfrenta maior dificuldade de inclusão na sociedade, porquanto a audição é o sentido essencial

para a aquisição e uso da linguagem. Segundo os autores, em face da invisibilidade da sua limitação, muitas vezes este deficiente é estigmatizado como revoltado e dissimulado.

2.2 Ondas Sonoras

A grosso modo, o som pode ser conceituado como ondas longitudinais que se propagam mecanicamente através de um meio elástico. Geralmente, o som é gerado pela vibração de um corpo. Essa vibração varia a pressão em torno do corpo, gerando expansão e/ou compressão do meio que o rodeia. O que distingue um instrumento de outro ou até mesmo a voz de cada pessoa é uma característica do som chamada timbre. Ela é marcada pelo formato da onda sonora, mostrado na Figura 2.1 e é responsável por essa textura do som, algo místico de cada instrumento, que traz toda essa essência para as músicas e composições (SANTOS, 2016).

Figura 2.1 – Tipos de ondas sonoras



Fonte: (SANTOS, 2016)

Para Nascimento e Gobara (2009), o som deve ser tratado como uma onda. Define-se onda como uma perturbação que se propaga transportando energia, sem envolver transporte de matéria. Ela pode ser classificada quanto a sua direção de vibração, natureza de vibração e grau de liberdade para a propagação. Denota-se que o som é uma onda mecânica, longitudinal, tridimensional e periódica.

Nascimento e Gobara (2009), também, definem algumas classificações das ondas sonoras, destacando as características que se atribuem a elas, a saber:

- **Natureza de Vibração:** Quanto à natureza de vibração, as ondas podem ser mecânicas ou eletromagnéticas. As ondas mecânicas precisam de um meio para se propagar. Ex.: som, terremotos. Porém, as ondas eletromagnéticas não precisam de um meio para se propagar e possuem velocidade igual a 300.000 km/s no vácuo. Ex.: Raio X, ondas de rádio, micro-ondas, luz visível, etc.
- **Direção de Vibração:** As ondas podem ser classificadas como transversais ou longitudinais. As transversais são aquelas cuja direção de vibração é perpendicular a propagação da onda. Já as ondas ditas longitudinais, possuem a mesma direção de vibração da propagação da onda. Ex.: O som.
- **Grau de Liberdade para a Propagação:** As ondas podem se propagar em uma, duas ou três dimensões. Esta característica é utilizada para classificá-las quanto ao grau de liberdade para a propagação. Quando uma onda se propaga apenas em uma dimensão, ela é classificada como unidimensional. Quando uma onda se propaga em duas dimensões, ela é classificada como bidimensional. Quando uma ela se propaga em três dimensões, é classificada como tridimensional.

Para Marcelo (2006), a propagação do som ao ar livre é normalmente considerada em três fases: a fonte sonora, a trajetória e o receptor. Primeiramente, a fonte emite uma certa potência sonora, gerando um nível sonoro que pode ser medido nas imediações da fonte. A partir daí, o nível sonoro é atenuado à medida que o som se propaga, entre a fonte e o receptor, ao longo de determinada trajetória.

A autora constata que o nível sonoro vai se reduzindo com a distância, à medida que o som vai se distanciando da fonte. A absorção do ar atmosférico atenua o som ao longo de sua trajetória. Reflexões no solo, áreas densamente arborizadas, barreiras naturais ou artificiais interferem na propagação do som, atenuando, assim, seu nível de intensidade.

A velocidade do som no ar independe da pressão atmosférica, pois esta é constante para uma determinada temperatura. Porém, alterando-se o nível térmico do ar, alterna-se também a sua densidade e, conseqüentemente, a velocidade do som no seu meio (SILVA, 2005).

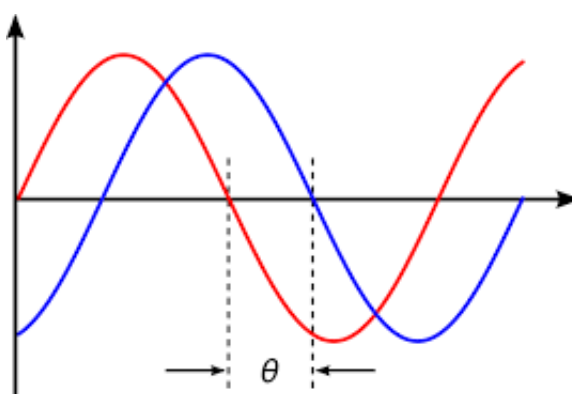
Quando ocorre de o gradiente de temperatura ser nulo, isto é, a temperatura permanecer constante nas camadas mais altas e mais baixas da atmosfera, não haverá nenhuma alteração na propagação do som, por causa da temperatura. Esta constância, na prática, só é possível dentro de uma faixa limitada ou ambiente artificialmente condicionado (MARCELO, 2006).

O deslocamento do ar também influencia na propagação do som em seu meio. Se o som se propaga no sentido e na direção do vento, os efeitos se somam e a velocidade final da propagação é resultante da soma dos valores da velocidade do som mais a do vento. Por sua vez, caso o som se propague na mesma direção do vento, porém em sentido inverso, a resultante será a diferença dos valores das suas velocidades (SILVA, 2005).

Para Silva (2005), os sons podem advir de duas formas distintas: sons originados no ar, que são produzidos dentro da massa de ar de um ambiente, aberto ou fechado, e que se transmitem integralmente até o ouvinte; ou sons originados por impacto, que são produzidos pelo choque de elementos sólidos ou líquidos e pelo ar, sendo transmitidos até o nosso corpo sob a forma de vibração ou de sensação sonora.

Um exemplo de sinal defasado é apresentado na Figura 2.2, onde o Θ representa a distância do defasamento entre dois sinais.

Figura 2.2 – Defasamento de sinais



Fonte: Elaborado pelo autor

2.2.1 Propagação Sonora

De acordo com Freitas (2005), os antigos não parecem ter se ocupado muito com a questão, um dos poucos a tratar da propagação do som foi Aristóteles que afirmava – erroneamente – que os sons agudos viajavam mais rápido que os graves. As primeiras medidas da velocidade do som tiveram que esperar pela Renascença, e foram realizadas em 1635 pelo filósofo francês Pierre Gassendi. Ele observou o disparo de um canhão localizado a uma distância conhecida, e mediu o tempo entre as chegadas do clarão e do som. Com isso, ele obteve 478 m/s para a velocidade do som, um valor um tanto alto. Gassendi também mostrou de forma conclusiva que os sons graves e agudos têm a mesma velocidade, contradizendo Aristóteles. O próprio Newton fez uma medida da velocidade do som cronometrando ecos no pátio do Trinity College. Em

1738 uma equipe da Academia de Ciências Parisiense, usando disparos de canhão como Gas-sendi, encontrou um resultado notadamente preciso: 343 m/s, a uma temperatura de 20°C. A preocupação com a temperatura é importante, pois mais ou menos à mesma época Branconi demonstrou que a velocidade do som aumenta com a temperatura (Equação 2.1). Freitas (2005) afirma que as medidas modernas dão para a velocidade do som a 0°C o valor:

$$c_0 = 331,45 \pm 0,05 \text{ m/s} \quad (2.1)$$

O valor acima é válido para o ar seco em condições normais de pressão (Equação 2.2). A uma temperatura absoluta T a velocidade do som é:

$$c = c_0 \sqrt{T/T_0} \quad (2.2)$$

em que $T_0 = 273,15 \text{ K}$ é a temperatura Kelvin correspondente a 0°C à 20°C a velocidade prevista pela equação acima é 343,4 m/s, o que mostra a qualidade da medida francesa de 1738. A temperaturas próximas da ambiente, a velocidade do som cresce aproximadamente 0,6 m/s por grau centígrado (FREITAS, 2005).

Certos projetos não desconsideram a presença de umidade no ar, pois a umidade interfere aumentando a velocidade do som, mas o efeito é bem menor que o efeito da temperatura, como é proposto por Bohn (1988), mostrando a influência dos fatores na velocidade do som.

A velocidade do som, também pode ser obtida através de outros métodos, como por exemplo, através de observações por ressonância, assim estipulando dois métodos:

- dada a frequência, determina-se o comprimento de onda;
- dado o comprimento de onda, determina-se a frequência.

Nos dois casos a velocidade do som será dada por (Equação 2.3):

$$c = \lambda \cdot f \quad (2.3)$$

onde λ é o comprimento de onda, f a frequência e c a velocidade do som.

As medidas por ressonância têm como principal vantagem o fato de serem fáceis de realizar. Sua desvantagem é que envolvem conceitos pouco familiares aos alunos do Ensino Médio, como ressonância, condições de contorno, ondas estacionárias, etc. (FREITAS, 2005).

2.2.2 Percepção da direção sonora

A percepção da direção dos sons é um fenômeno que ocorre de maneira inconsciente e está presente desde cedo no ciclo de vida humano, por ser uma função relacionada à sobrevivência: por volta de seis meses de idade um bebê normal já é capaz de localizar sons a sua volta (OLIVEIRA et al., 2008).

Para Oliveira et al. (2008), na percepção da direção do som, deve haver a discriminação de diferenças muito sutis de intensidade, frequência e tempo. Geralmente, a localização sonora espacial depende da análise das diferenças com relação a esses parâmetros entre os sons que chegam às duas orelhas.

A localização de fontes sonoras acontece, sobretudo, pela influência de características físicas da onda sonora, que geram diferenças no espectro dessa onda entre as orelhas, ou seja, as chamadas diferenças interaurais. Assim, dois fatores principais aparecem, ou seja, as diferenças de fase entre as orelhas, denominada diferença de tempo interaural (DTI) e as diferenças de intensidade entre as orelhas, denominada diferença de intensidade interaural (DII) (OLIVEIRA et al., 2008).

Rodrigues et al. (2009), expõem que pessoas com audição normal percebem e discriminam melhor os sons justamente por possuírem duas orelhas. Dessa forma, pode-se afirmar que pessoas com deficiência auditiva bilateral irão ouvir melhor se estiverem utilizando próteses bilaterais. Mas nem sempre ouvir com duas próteses auditivas retroauriculares ou intra-aurais possibilita ou melhora a localização sonora.

Por ser a localização de fontes sonoras uma função importante para o desempenho auditivo do indivíduo, uma vez que, em algumas situações, sobretudo fora do campo visual, é o único sentido presente para a percepção da distância e que a sua falta implica, muitas vezes, no aumento dos riscos de acidentes (RODRIGUES et al., 2009).

2.3 Sinais e Sistemas

Um sinal é um conjunto de dados ou informações. Como por exemplo, tem-se um sinal de telefone ou televisão, o registro de vendas de uma corporação ou os valores de fechamento de uma bolsa de negócios. Em todos esses exemplos, os sinais são funções da variável independente tempo, entretanto este nem sempre é o caso. Quando uma carga elétrica é distribuída

sobre um corpo, por exemplo, o sinal é a densidade de carga, ou seja, uma função do espaço ao invés do tempo (LATHI, 2006).

Os sinais podem ser posteriormente processados por sistemas, os quais podem modificar os sinais ou extrair informações adicionais deles. Por exemplo, um operador de artilharia antiaérea pode querer saber a posição futura de um alvo hostil que está sendo seguido por seu radar. Conhecendo o sinal do radar, ele sabe a posição passada e a velocidade do alvo. Através do processamento do sinal do radar – a entrada –, ele pode estimar a posição futura do alvo. Portanto, um sistema é uma entidade que processa um conjunto de sinais – entradas – resultando em um outro conjunto de sinais – saídas. Um sistema pode ser construído com componentes físicos, elétricos, mecânicos ou sistemas hidráulicos – realização em *hardware* – ou pode ser um algoritmo que calcula uma saída de um sinal de entrada (realização em *software*) (LATHI, 2006).

2.3.1 Tamanho do Sinal

Para Lathi (2006), o tamanho de qualquer entidade é um número que indica a largura ou seu comprimento. Genericamente falando, a amplitude do sinal varia com o tempo. Como um sinal que existe em um certo intervalo de tempo com amplitude variante pode ser medido por um número que irá indicar o seu tamanho ou a sua força? Tal medida deve considerar não apenas a amplitude do sinal, mas também sua duração. Por exemplo, se quiser utilizar um único número V como medida do tamanho de um ser humano, deve considerar não somente seu peso, mas também sua altura. Se levar em consideração que a forma da pessoa é um cilindro cuja variável é o raio r – o qual varia com a altura h –, então uma possível medida de tamanho de uma pessoa de altura H é o volume V da pessoa dada por

$$V = \pi \int_0^H r(h)^2 dh \quad (2.4)$$

2.3.1.1 Energia e Potência do Sinal

Toda atividade física é medida por uma transferência de energia. Nenhum sistema físico real pode responder a um estímulo a não ser que ele tenha energia. É importante estabelecer certa terminologia para descrever a energia e a potência dos sinais. No estudo sobre sinais em sistemas, os sinais são normalmente tratados como abstrações matemáticas e, muitas vezes, a sua significância física é ignorada ou irrelevante em favor da simplicidade da análise. Sinais

típicos em sistemas elétricos seriam tensões e correntes, mas poderiam ser carga ou campo elétrico ou alguma outra quantidade física. Em outros tipos de sistemas, um sinal poderia ser uma força, uma temperatura, uma concentração química, um fluxo de nêutrons e assim por diante. Devido aos diversos tipos de sinais físicos existentes, passíveis de serem operados por sistemas, o termo energia de sinal foi definido. Assim, a energia de sinal – para se contrapor à energia somente de um sinal – é definida como a área sob a magnitude do sinal ao quadrado (ROBERTS, 2009).

Ao argumentar desta forma, podemos considerar a área abaixo do sinal $x(t)$ como uma possível medida de seu tamanho, pois a área irá considerar não somente a amplitude, mas também sua duração. Entretanto, esta medida ainda é defeituosa, pois mesmo para um sinal grande $x(t)$, suas áreas positivas e negativas podem se cancelar, indicando um sinal de tamanho pequeno. Esta dificuldade pode ser corrigida pela definição do tamanho do sinal como área debaixo de $x(t)^2$ a qual é sempre positiva. Podemos chamar essa medida de energia do sinal E_x , definida (para um sinal real) por

$$E_x = \int_{-\infty}^{\infty} x(t)^2 dt \quad (2.5)$$

Essa definição pode ser generalizada para um sinal complexo $x(t)$, sendo dada por

$$E_x = \int_{-\infty}^{\infty} |x(t)|^2 dt \quad (2.6)$$

Também existem outras possíveis medidas de tamanho de um sinal tal como a área sob $|x|^2$ (LATHI, 2006).

Portanto, as unidade da energia de sinal dependem das suas unidades. Se a unidade do sinal é o volt (V), a sua energia é expressa em $V^2 \cdot s$. Portanto, a energia de sinal é definida dessa forma para ser proporcional à energia física real presente em um sinal, mas que não necessariamente seja igual àquela energia física. Em relação ao sinal de tensão $v(t)$ presente nos terminais de um resistor R , a energia real entregue ao resistor pela tensão aplicada seria dada por

$$Energia = \int_{-\infty}^{\infty} \frac{|v(t)|^2}{R} dt = \frac{1}{R} \int_{-\infty}^{\infty} |v(t)|^2 dt = \frac{E_v}{R} \quad (2.7)$$

Pela definição, energia de sinal é proporcional à energia existente e à constante de proporcionalidade que, nesse caso, é R . Para tipos distintos de sinal, a constante de proporcionalidade será diferente. Em muitos tipos de análises de sistemas, o uso da energia de sinal é mais conveniente do que o uso da energia física real disponível (ROBERTS, 2009).

Para muitos sinais encontrados na análise de sinais e sistemas, a integral expressa na Equação 2.5 não converge porque a energia do sinal é infinita. Isso ocorre, normalmente, devido ao sinal que não é limitado no tempo. Um exemplo de sinal com energia infinita seria o sinal senoidal $x(t) = A\cos(2\pi f_0 t)$, $A \neq 0$. A energia de sinal é infinita porque, sobre um intervalo de tempo infinito, a área sob o quadrado deste sinal é infinita (ROBERTS, 2009).

Por este motivo, a energia do sinal deve ser finita para que seja uma medida significativa do seu tamanho. Uma condição necessária para que a energia seja finita é a amplitude do sinal $\rightarrow 0$ quando $|t| \rightarrow \infty$ (Figura 2.3a) caso contrario a integral da Equação 2.5 não irá convergir.

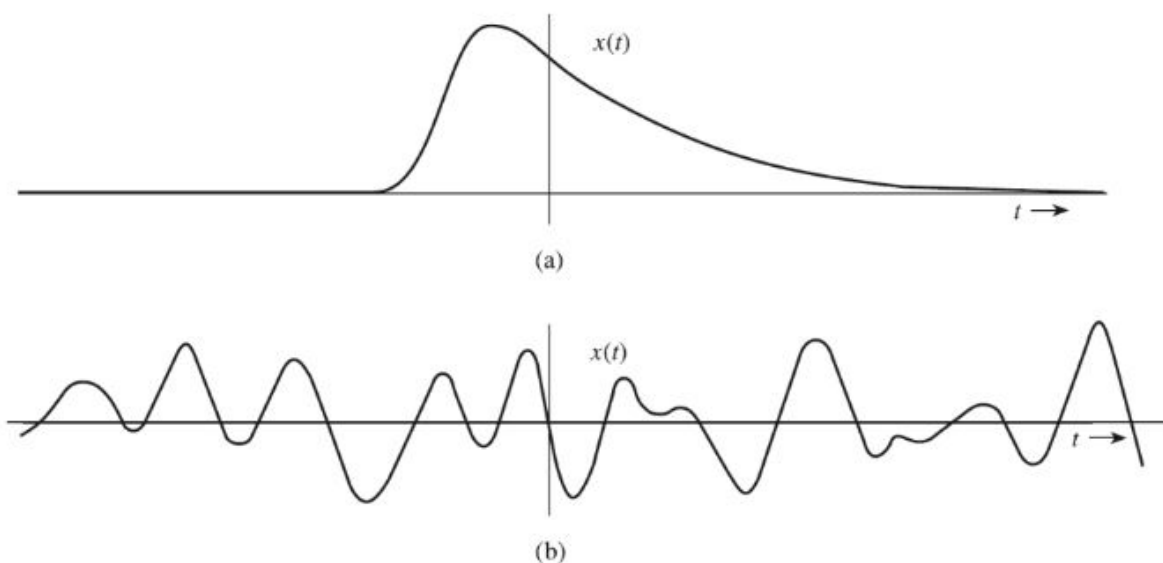
De acordo com Lathi (2006), quando a amplitude do sinal $x(t)$ não $\rightarrow 0$ quando $|t| \rightarrow \infty$ (Figura 2.3b), a energia do sinal é infinita. Uma medida mais significativa do tamanho do sinal neste caso é a energia média, se ela existir. Esta média é chamada de *potência do sinal*. Para um sinal $x(t)$, sua potência P_x é definida por:

$$P_x = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t)^2 dt \quad (2.8)$$

Podemos generalizar esta definição para um sinal complexo $x(t)$, sendo dada por:

$$P_x = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} |x(t)|^2 dt \quad (2.9)$$

Figura 2.3 – Exemplos de sinais: (a) sinal com energia finita (b) sinal com potência finita



Fonte: (LATHI, 2006)

Observe que a potência do sinal P_x é uma média temporal do quadrado da amplitude do sinal, ou seja, o valor *médio quadrático* de $x(t)$. De fato, a raiz quadrada de P_x é o que já conhecido valor *rms* (raiz média quadrática) de $x(t)$.

Geralmente, a média de uma entidade ao longo de um grande intervalo de tempo aproximando do infinito existe se a entidade for periódica ou possuir uma regularidade estatística. Se tal condição não for satisfeita, a média não existirá. Por exemplo, um sinal em ramá, $x(t) = t$, aumenta indefinidamente quando $|t| \rightarrow \infty$ e nem energia nem a potência existirão para este sinal. Entretanto, a função degrau unitário, a qual não é periódica nem possui regularidade estatística, possui uma potência finita (LATHI, 2006).

Para sinais periódicos, a determinação da potência de um sinal média pode ser mais simples. O valor médio de qualquer função periódica é a media sobre algum intervalo. Sendo assim, desde que o quadrado de uma função periódica seja também periódico, para sinais periódicos, tem-se

$$P_x = \frac{1}{T} \int_{t_0}^{t_0+T} |x|^2(t) dt = \frac{1}{T} \int_T |x(t)|^2 dt \quad (2.10)$$

em que a notação \int_T significa a mesma coisa que $\int_{t_0}^{t_0+T}$ para uma escolha arbitrária de t_0 onde T pode ser qualquer período de $|x(t)|^2$ (ROBERTS, 2009).

Sinais que têm energia de sinal finita são denominados sinais de energia e os que têm energia de sinal infinita, mas potência de sinal média finita, são denominados sinais de potência. Nenhum sinal físico real pode de fato ter energia em quantidade infinita ou potência média infinita, uma vez que não existe energia ou potência suficientemente disponível no universo. Porém, ao analisar frequentemente sinais que, de acordo com suas definições matemáticas estritas, têm energia infinita, como uma senoide, por exemplo (ROBERTS, 2009).

2.3.2 Processamento de sinais

O processamento digital de sinais estuda as regras que governam os sinais que são funções de variáveis discretas, assim como os sistemas usados para processá-los. Da mesma forma, lida com os aspectos envolvidos, que são funções de variáveis contínuas utilizando técnicas digitais.

No processamento analógico de sinais, toma-se um sinal que varia continuamente, representando uma quantidade física que varia ininterruptamente, e passa-se por um sistema que

modifica o sinal com um certo propósito. Essa modificação também é, em geral, continuamente variável por natureza, isto é, pode ser descrita por equações diferenciais.

Alternativamente no processamento digital de sinais, processa sequências de números usando algum tipo de hardware digital. Normalmente, é chamado as sequências de números de sinais digitais ou sinais no tempo discreto. O poder de processamento digital de sinais decorre do fato de que, uma vez que uma sequência de números esteja disponível para o hardware digital apropriado, podemos efetuar qualquer forma de processamento numérico sobre eles (DINIZ et al., 2014).

2.3.3 Amostragem de sinais

De acordo com Santos (2019), a amostragem de sinais nada mais é do que a coleta de pontos, ao longo do tempo, de forma que seja possível reproduzir um sinal contínuo em um sinal discreto. Em outras palavras, um sinal analógico é transformado em um digital. Um sinal discreto é representado por uma sequência numérica, que pode ser considerada uma função contínua de uma variável discreta. Em muitos casos, um sinal $x(n)$ no tempo discreto consiste em amostras de um sinal $x_a(t)$ no tempo contínuo, isto é,

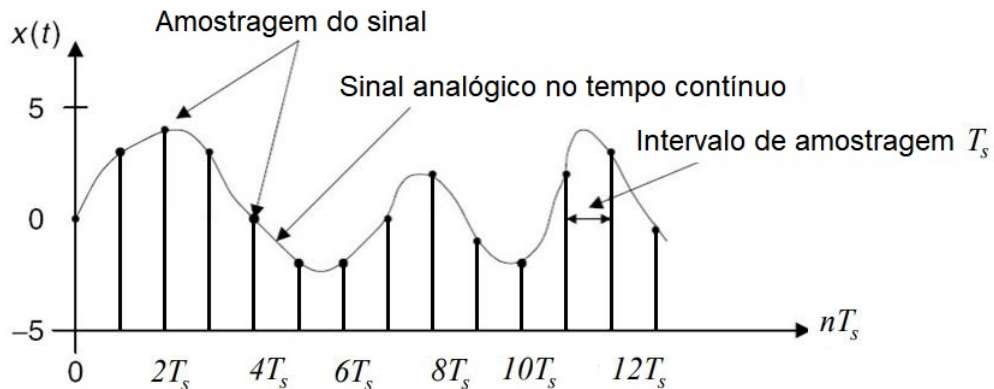
$$x[n] = x_c(nT_s) \quad (2.11)$$

onde T_s é o período de amostragem e $f_s = 1/T_s$ representa a frequência de amostragem.

Se tem necessidade de processar o sinal no tempo contínuo $x_a(t)$ usando um sistema no tempo discreto, então primeiramente precisa convertê-lo conforme a equação 2.11, processar digitalmente a entrada no tempo discreto e por fim converter a saída no tempo discreto de volta ao domínio do tempo contínuo. Portanto, para essa operação seja efetiva, é essencial que tenha a capacidade de restaurar um sinal no tempo contínuo a partir de suas amostras (DINIZ et al., 2014).

A Figura 2.4, representa um exemplo de amostragem do sinal no tempo contínuo a partir das amostras discretas.

Figura 2.4 – Exemplo da amostragem de um sinal



Fonte: (VILLANUEVA, 2015)

2.3.4 Transformada de Fourier

A resposta de um sistema linear e invariante no tempo de uma função senoidal também é uma senoide de mesma frequência, porém a amplitude e a fase podem estar alteradas. Por conta deste fato, assumindo a linearidade do sistema, é interessante decompor sinais em somas de senoides de diversas frequências, de maneira que cada componente possa ser tratado de forma independente (SANTOS, 2019).

Qualquer sequência discreta $x[n]$ pode ser escrita como uma soma de senoides, com frequências adequadas (BOYCE; DIPRIMA, 2015). Este processo é análogo à decomposição de um vetor em componentes x , y e z . Tais componentes podem ser tratados separadamente, entretanto carregam ainda informações sobre o vetor em si. No domínio da frequência, é possível obter informações importantes sobre o sinal analisado que não são facilmente vistas no domínio do tempo (SANTOS, 2019).

Uma função aperiódica de duração finita pode ser vista como uma função periódica com período infinito. Supondo-se que o período T tende a infinito, ω_1 tende a zero, o intervalo entre as frequências harmônicas também tende a zero e a função no domínio da frequência é contínua. A Transformada de Fourier mapeia sinais aperiódicos no tempo para o domínio da frequência, resultando em um espectro contínuo (ANTUNES et al., 2014).

A transformada de Fourier é definida como:

$$X(\omega) = F(x(t)) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt \quad (2.12)$$

A transformada inversa de Fourier calcula a função contínua, de período infinito, $f(t)$, no domínio do tempo.

$$x(t) = F^{-1}(X(j\omega)) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} d\omega \quad (2.13)$$

O operador F significa “transformada de Fourier de” e o operador F^{-1} significa “transformada inversa de Fourier de”. Essas duas definições da transformada de Fourier são as mais comumente utilizadas na engenharia. A primeira (Equação 2.12) é escrita em termos de frequência cíclica f e tem a vantagem de ser bastante simétrica. As transformadas direta e inversa são praticamente iguais. Somente o sinal no expoente e a variável de integração mudam. Essa é a forma adotada mais frequente em análise de sistemas de comunicação, na óptica de Fourier e em processamento de imagens.

A segunda definição (Equação 2.13) é escrita em termos da variável de frequência angular ω em lugar da frequência cíclica f . A frequência angular possui uma relação com as contantes de tempo e com as frequências de ressonância de sistemas reais um tanto quanto mais direta e, conseqüentemente, as transformadas de certas funções de sistemas são um tanto mais simples com essa adoção. Essa é a forma mais comumente usada em análise de sistemas de controle. Qualquer uma das definições pode ser convertida na outra por meio da seguinte relação $\omega = 2\pi f$ (ROBERTS, 2009).

Em conformidade com o autor, o sinal $x(t)$ é dito estar no domínio do tempo, pois seu argumento funcional t representa tempo, e a função da transformada $X(f)$ ou $X(j\omega)$ é dita estar no domínio da frequência porque seu argumento funcional f ou ω representa a frequência. A frequência cíclica é o recíproco do tempo e a frequência angular é proporcional ao recíproco do tempo.

2.3.5 *Aliasing* e Teorema de Nyquist

Quando um sinal é adquirido, é necessário utilizar uma frequência de amostragem adequada, de forma que a sua representação digital esteja o mais próximo possível do sinal original. Quando a amostragem é insuficiente, um sinal de frequência f_0 acaba sendo representado como se fosse um sinal de frequência $f_1 < f_0$. Esse fenômeno é conhecido como *Aliasing*.

Segundo Santos (2019), uma forma de explicar o fenômeno é analisar o tempo entre os intervalos das amostras. Considera-se um sinal senoidal $x[n] = x_c(nT_s) = A \cos(\omega n T_s + \phi)$, onde

ω é a frequência angular. Como $\omega = 2\pi f$, podemos escrever:

$$A\cos(\omega nT_s) = A\cos(2\pi f nT_s) \quad (2.14)$$

Tomando como exemplo um sinal com frequência de 60 Hz, adquirido a frequências de amostragem $f_{s1} = 30$ Hz, $f_{s2} = 60$ Hz e $f_{s3} = 240$ Hz, teremos:

$$\frac{1}{f_1} = \frac{1}{30} = 33,3ms \quad (2.15)$$

$$\frac{1}{f_2} = \frac{1}{60} = 16,6ms \quad (2.16)$$

$$\frac{1}{f_3} = \frac{1}{240} = 4,16ms \quad (2.17)$$

Como o tempo T_1 é maior que o tempo para um ciclo do sinal, a frequência f_1 não é adequada para amostrar um sinal de 60 Hz. Outra forma de explicar sobre o *aliasing* é analisando o sinal no domínio da frequência. Quando é realizado a transformada de Fourier, passando o sinal do domínio do tempo para o domínio da frequência, o sinal real é transformado em um sinal complexo. Por conta disso, seus valores são conjugados complexos, o que faz com que tenha-se um sinal espelhado em torno do eixo vertical, conhecido como *alias*. Caso a frequência de amostragem não seja adequada, o sinal no domínio da frequência indicará uma frequência menor do que a original.

Conforme Diniz et al. (2014), Teorema de Nyquist, descreve que um sinal $x_a(t)$ no tempo contínuo tem largura de faixa limitada, isto é, sua transformada de Fourier é tal que $X_a(j\Omega) = 0$ para $|\Omega| > \Omega_c$, então $x_a(t)$ pode ser completamente recuperado a partir do sinal no tempo discreto $x(n) = x_a(nT)$ se a frequência de amostragem Ω_s satisfaz $\Omega_s > 2\Omega_c$

2.3.6 Filtros

O conceito de filtros tem sido uma parte integral da evolução da engenharia elétrica, desde o início. Várias conquistas tecnológicas não teriam sido possíveis sem os filtros elétricos. Devido a esse papel proeminente dos filtros, muito esforço tem sido despendido em sua teoria, em seu projeto e na sua construção (SADIKU; MUSA, 2014).

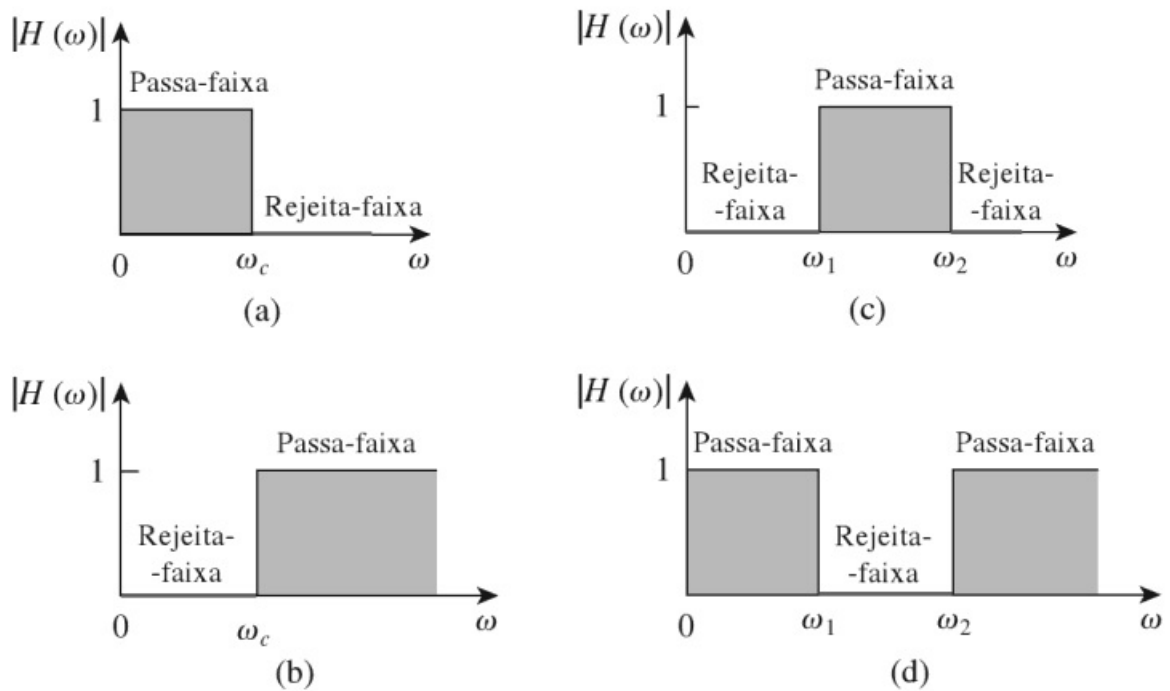
Os filtros são constituídos por elementos como resistência, capacitor e indutor, modelados e projetados para eliminar ou minimizar as amplitudes harmônicas indesejadas do sistema elétrico, podendo ser conectados em paralelo ou em série com o circuito (DUGAN et al., 2012).

Como um dispositivo seletor de frequência, um filtro pode ser utilizado para limitar o espectro de frequência de um sinal a uma banda específica de frequências. Os filtros são os circuitos utilizados em rádio e TV para nos permitir selecionar um desejado sinal dentre múltiplos sinais de radiodifusão no ambiente. De acordo com Francis (2019), a aplicação de filtros serve para obter dados mais precisos, minimizando os efeitos dos ruídos presentes nas altas frequências.

Segundo Sadiku e Musa (2014), conforme mostrado na Figura(2.5), existem quatro tipos de filtros:

1. Um filtro passa-baixa passa baixas frequências e retém altas frequências, conforme mostrado na Figura 2.5(a).
2. Um filtro passa-alta passa altas frequências e rejeita baixas frequências, conforme mostrado na Figura 2.5(b).
3. Um filtro passa-faixa passa frequências dentro de uma banda e bloqueia ou atenua frequências fora dessa banda, conforme mostrado idealmente na Figura 2.5(c).
4. Um filtro rejeita-faixa passa frequências fora de uma banda e bloqueia ou atenua frequências dentro da banda, conforme mostrado idealmente na Figura 2.5(d).

Figura 2.5 – Respostas ideais em frequência para os quatros tipos de filtro



Fonte: (SADIKU; MUSA, 2014)

Um resumo das características desses filtros é apresentado na Tabela 2.1. Deve ser enfatizado que as características na Tabela 2.1 são somente válidas para filtros simples e não se deve ter a impressão que somente filtros como esses existem.

Tabela 2.1 – Resumo das características dos filtros

Tipo de filtro	$H(0)$	$H(\infty)$	$H(\omega_c)$ ou $H(\omega_0)$
Passa-baixa	1	0	$1/\sqrt{2}$
Passa-alta	0	1	$1/\sqrt{2}$
Passa-faixa	0	0	1
Rejeita-faixa	1	1	0

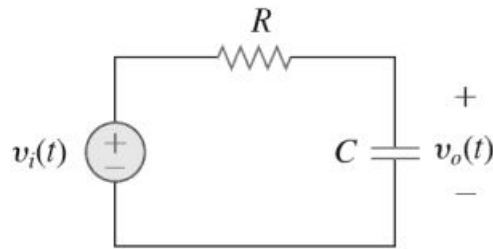
Fonte: Modificado de Sadiku e Musa (2014)

Note que ω_c é a frequência de corte para os filtros passa-baixa e passa-alta, ω_0 é a frequência central para os filtros passa-faixa e rejeita-faixa.

2.3.6.1 Filtros Passa-baixa

Um típico filtro passa-baixa é formado quando a saída de um circuito RC em série é tomada dos terminais do capacitor, conforme mostrado na Figura 2.6.

Figura 2.6 – Filtro Passa-baixa



Fonte: (SADIKU; MUSA, 2014)

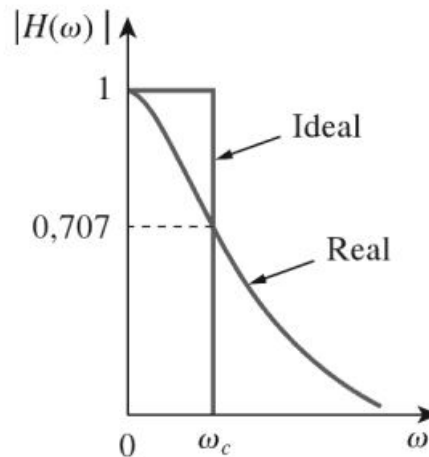
A função de transferência é

$$H(\omega) = V_0/V_i = \frac{1/j\omega C}{R + 1/j\omega C} \quad (2.18)$$

$$H(\omega) = \frac{1}{1 + j\omega RC} \quad (2.19)$$

Observe que $H(0) = 1$, $H(\infty) = 0$. O gráfico de $|H(\omega)|$ é mostrado na Figura 2.7, onde as características ideais são também apresentadas.

Figura 2.7 – Resposta ideal e real em frequência de um filtro passa-baixa



Fonte: (SADIKU; MUSA, 2014)

A frequência de meia potência, a qual é equivalente à frequência de corte no diagrama de Bode, mas no contexto de filtros, é usualmente conhecida como frequência de corte ω_c e é obtida fazendo-se a magnitude $H(\omega)$ igual a $1/\sqrt{2}$, isto é,

$$H(\omega_c) = \frac{1}{\sqrt{1 + \omega_c^2 R^2 C^2}} = \frac{1}{\sqrt{2}} \quad (2.20)$$

em que

$$\omega_c = \frac{1}{RC} \quad (2.21)$$

A frequência de corte é também chamada de frequência “*rolloff*”. Assim, um filtro passa-baixa é projetado para passar somente frequências de zero até a frequência de corte ω_c . Um circuito passa-baixa pode também ser formado quando a saída de um circuito RL em série é tomada nos terminais do resistor. Obviamente, existem muitos outros circuitos para os filtros passa-baixa (SADIKU; MUSA, 2014).

2.4 Produto vetorial

O produto entre dois vetores pode ser usado para determinar o ângulo entre as linhas de ação desses vetores e determinar a componente de um vetor que age na direção do outro. Tanto na estática quanto em outras disciplinas, frequentemente precisaremos determinar tais informações, e o produto escalar fornece um modo fácil de fazê-lo, especialmente para vetores em três dimensões (PLESHA et al., 2014).

Segundo Rogawski e Adams (2018), o produto escalar é uma das mais importantes operações vetoriais, desempenhando um papel em quase todos os aspectos do Cálculo e várias variáveis.

O produto escalar $\mathbf{v} \cdot \mathbf{w}$ de dois vetores $\mathbf{v} = \{v_1, v_2, v_3\}$, $\mathbf{w} = \{w_1, w_2, w_3\}$,

$$\mathbf{v} \cdot \mathbf{w} = v_1 w_1 + v_2 w_2 + v_3 w_3 \quad (2.22)$$

Em palavras, para calcular o produto escalar, multiplica e soma os componentes correspondentes.

O produto escalar de vetores $\mathbf{v} = \{v_1, v_2\}$, $\mathbf{w} = \{w_1, w_2\}$ em \mathbb{R}^2 é definido analogamente:

$$\mathbf{v} \cdot \mathbf{w} = v_1 w_1 + v_2 w_2 \quad (2.23)$$

Para Rogawski e Adams (2018), o produto escalar tem uma relação muito próxima com o ângulo entre \mathbf{v} e \mathbf{w} . Antes de entrar nisso, descrevemos algumas propriedades elementares do produto escalar.

Em primeiro lugar, o produto escalar é comutativo: $\mathbf{v} \cdot \mathbf{w} = \mathbf{w} \cdot \mathbf{v}$, porque podemos multiplicar os componentes em qualquer ordem. Em segundo lugar, o produto escalar de um vetor

consigo mesmo é o quadrado do comprimento do vetor: se $\mathbf{v} = \{v_1, v_2, v_3\}$, então

$$\mathbf{v} \cdot \mathbf{v} = v_1v_1 + v_2v_2 + v_3v_3 = v_1^2 + v_2^2 + v_3^2 = \|\mathbf{v}\|^2 \quad (2.24)$$

O produto escalar também satisfaz a distributividade e uma propriedade com escalares, resumidas na Tabela 2.2.

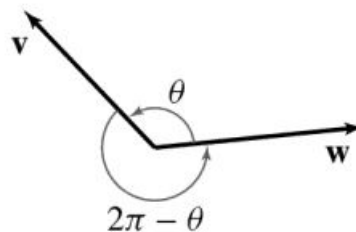
Tabela 2.2 – Teorema 1 do Produto Escalar

Teorema 1	Propriedades do produto escalar
(i) Nulo	$0 \cdot v = v \cdot 0 = 0$
(ii) Comutatividade:	$v \cdot w = w \cdot v$
(iii) Tirando escalares para fora:	$(\lambda v) \cdot w = v \cdot (\lambda w) = \lambda(v \cdot w)$
(iv) Distributividade:	$u \cdot (v + w) = u \cdot v + u \cdot w$ $(v + w) \cdot u = v \cdot u + w \cdot u$
(v) Relação com o comprimento:	$v \cdot v = \ v\ ^2$

Fonte: (ROGAWSKI; ADAMS, 2018)

Como mencionado pelos autores Rogawski e Adams (2018), o produto escalar $v \cdot w$ está relacionado com o ângulo θ entre v e w . Esse ângulo θ não está definido de maneira única porque, como vemos na Figura 2.8, θ e $2\pi - \theta$ podem, ambos, servir como ângulo entre v e w . Além disso, qualquer múltiplo de 2π pode ser somado a θ .

Figura 2.8 – Angulo entre dois vetores



Fonte: (ROGAWSKI; ADAMS, 2018)

Todos esses ângulos têm o mesmo cosseno, portanto não importa qual ângulo é utilizado. No entanto, será adotado a seguinte convenção:

O ângulo entre dois vetores é escolhido para satisfazer $0 \leq \theta \leq \pi$

Seja θ o ângulo entre dois vetores não nulos v e w . Então:

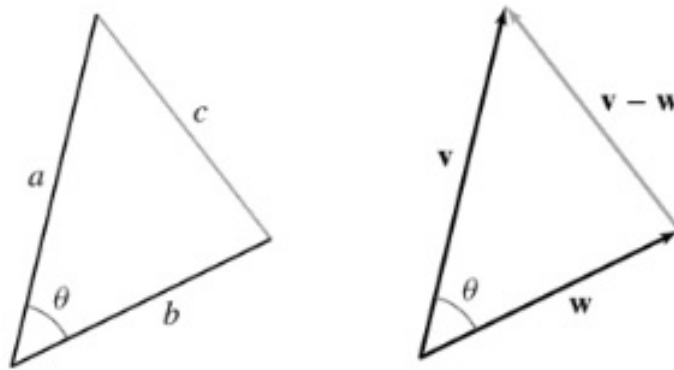
$$v \cdot w = \|v\| \|w\| \cos\theta \quad (2.25)$$

ou

$$\cos\theta = \frac{v \cdot w}{\|v\| \|w\|} \quad (2.26)$$

De acordo com a lei dos cossenos, os três lados de um triângulo satisfazem como mostrado na Figura 2.9.

Figura 2.9 – Comparação com os lados de um triângulo



Fonte: (ROGAWSKI; ADAMS, 2018)

Aplicando a lei dos cossenos tem-se:

$$c^2 = a^2 + b^2 - 2ab\cos\theta \quad (2.27)$$

Se dois lados do triângulo são v e w , então o terceiro lado é $v - w$, como na Figura 2.9, e pela lei dos cossenos,

$$\|v - w\|^2 = \|v\|^2 + \|w\|^2 - 2\cos\theta \|v\| \|w\| \quad (2.28)$$

Agora, pela propriedade (v) do Teorema 1 (Tabela 2.2) e a distributividade

$$\|v - w\|^2 = (v - w) \cdot (v - w) = v \cdot v - 2v \cdot w + w \cdot w = \|v\|^2 + \|w\|^2 - 2v \cdot w \quad (2.29)$$

Comparando a Equação 2.28 com a Equação 2.30, obtem-se

$$-2\cos\theta \|v\| \|w\| = -2v \cdot w \quad (2.30)$$

decorre a Equação 2.25 ou Equação 2.33.

Por definição de arco cosseno, o ângulo $\theta = \arccos x$ é o ângulo do intervalo $[0, \pi]$ satisfazendo $\cos\theta = x$. Assim, para vetores não nulos v e w , tem-se

$$\cos\theta = \frac{v \cdot w}{\|v\| \|w\|} \quad (2.31)$$

ou

$$\theta = \arccos\left(\frac{v \cdot w}{\|v\| \|w\|}\right) \quad (2.32)$$

2.5 Somatório das Diferenças Absolutas de números reais

A diferença absoluta de dois números reais x, y é dada por $|x - y|$. O valor absoluto da sua diferença (AUBYN et al., 2006; STEWART, 2010; WEISSTEIN, 2019) descreve a distância sobre a linha real entre os pontos correspondentes ao x e y . É um caso especial do espaço L^p para todos $1 \leq p \leq \infty$ e é a métrica padrão usada para ambos conjuntos de números racionais \mathbb{Q} e o conjunto de números reais \mathbb{R} (RUMSEY; FRANZIN, 2018; BACHA, 2004).

Como em qualquer métrica, as propriedades apoiam que:

- $|x - y| \geq 0$, já que o valor absoluto é sempre não-negativo.
- $|x - y| = 0$ se, e somente se, $x = y$.
- $|x - y| = |y - x|$, simetria ou comutatividade.
- $|x - z| \leq |x - y| + |y - z|$, desigualdade triangular, no caso de a diferença absoluta, a igualdade é apoiada se, e somente se, $x \leq y \leq z$.

Assim, o somatório da diferença absoluta de números reais é dado por:

$$\sum_{i=1}^n |x_i - y_i| \quad (2.33)$$

2.5.1 Algoritmo do Somatório das Diferenças Absolutas - SDA

O SDA é um algoritmo de correspondência que computa a diferença de todos os pontos da janela definida e em seguida soma o módulo de todos os resultados dessas diferenças. O valor mínimo desta função denota qual das janelas melhor se correlacionou. A disparidade é então calculada como a diferença horizontal entre as coordenadas dos pontos (NISHIMURA, 2008).

O SDA, também chamado SAD (Sum of the Absolute Differences), possui algoritmos desenvolvidos utilizando suas variações propostas pelos autores Cancellier et al. (2017), Zhu et al. (2005), Trudeau et al. (2015). O SDA é utilizado para casos onde o tempo de processamento é crítico segundo França (2003). Também é um dos métodos aplicados em processamento de sinais e reconhecimento de imagens onde pode ser utilizado para calcular a semelhança entre dois fragmentos da cena (JACOBI et al., 2006).

A estrutura proposta pelo autor Jacobi et al. (2006) para calcular o SDA foi escolhida devido a similaridade com o trabalho, pois o autor implementa em uma FPGA para obter processamento em tempo real, e é representada na Equação 2.34.

$$SDA = \sum_{x=1}^n \sum_{y=1}^n |I'_{x,y} - I_{x,y}| \quad (2.34)$$

2.6 Sistemas Embarcados

Wilmshurst (2009) define um sistema embarcado como um sistema cuja principal função não é computacional, mas que é controlado por um computador embarcado ou incorporado dentro dele. O autor afirma que os sistemas embarcados estão presentes em tudo, como em casas, escritórios, fábricas, carros, hospitais etc. Os sistemas podem ser muito diferentes entre eles, mas a característica principal de um sistema embarcado está presente em todos.

No entanto, Noergaard (2012), define que um sistema embarcado é um sistema de computação com alguma aplicação específica, diferenciando-se de computadores pessoais ou supercomputadores. O autor demonstra que uma definição de “sistemas embarcados” é fluida e muito difícil de definir, pois o sistema evolui constantemente com os avanços da tecnologia. O autor afirma ainda que não existe só uma definição para o sistema, mas algumas descrições sobre o sistema embarcado, a saber:

- Sistemas embarcados são mais limitados em hardware e/ou software do que um computador;
- Um sistema embarcado é construído para executar poucas funções específicas;
- Um sistema embarcado é um sistema computacional com uma qualidade e confiabilidade maior que outros tipos de sistemas.

2.6.1 Microcontrolador

Um microcontrolador (MCU) é um “computador-num-chip”, contendo um processador, uma memória e periféricos de entrada/saída. O MCU é um microprocessador que pode ser programado para funções específicas, em contraste com outros microprocessadores de propósito geral – como os utilizados nos PCs. Eles são embarcados no interior de algum outro dispositivo – geralmente um produto comercializado – para que possam controlar as funções ou as ações do produto. Um outro nome para o microcontrolador, portanto, é controlador embutido.

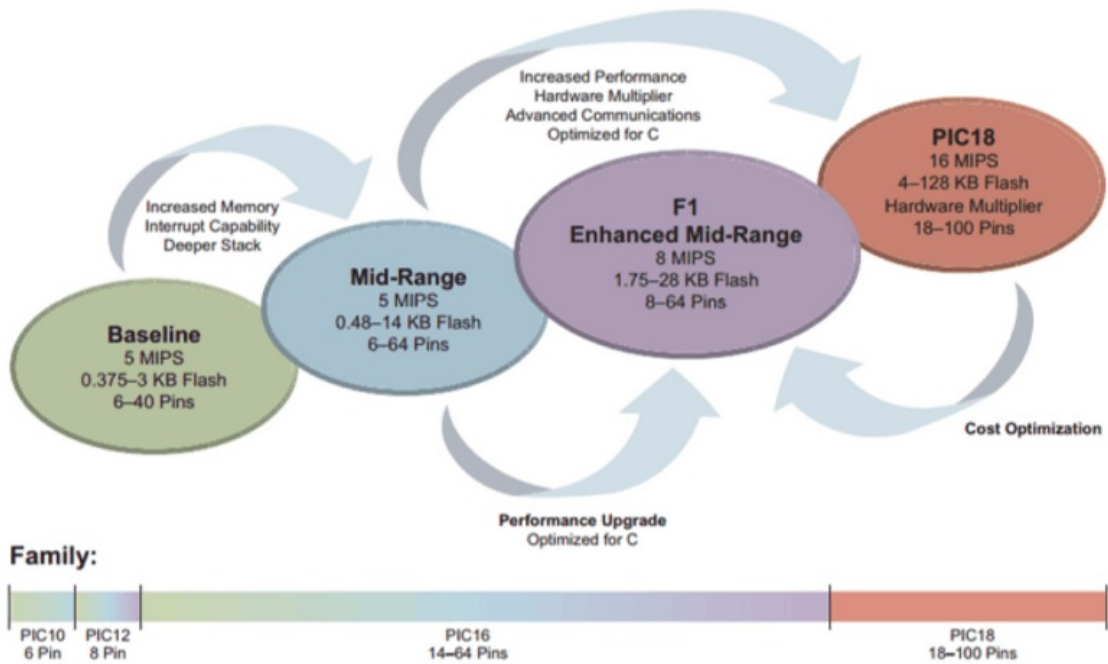
Os microcontroladores diferenciam-se dos processadores, pois além dos componentes lógicos e aritméticos usuais de um microprocessador de uso geral, os MCUs integram elementos adicionais em sua estrutura interna, como memória de leitura e escrita para armazenamento de dados, memória somente de leitura para armazenamento de programas, Electrically-Erasable Programmable Read-Only Memory (EEPROM) para armazenamento permanente de dados, dispositivos periféricos como conversores analógico/digitais (ADC, em inglês), conversores digitais/analógicos (DAC, em inglês) em alguns casos.

Os microcontroladores são largamente utilizados em diversos aparelhos e produtos hoje em dia, sem eles os produtos e as funcionalidades existentes hoje não existiriam, pois a capacidade que apresentam de gerenciar e otimizar a utilização de dispositivos é considerável.

Como neste trabalho foram testados e utilizados apenas microcontroladores da Microchip, então será apresentado somente este MCU, e na próxima subseção será explicado sobre o microcontrolador da família PIC16.

De acordo com Souza (2014), as linhas de microcontroladores de 8 bits da Microchip é dividida em quatro grupos: Baseline, Mid-Range, Enhanced Mid-Range e PIC18, como mostrado na Figura 2.10:

Figura 2.10 – Linhas de microcontroladores 8 bits da Microchip

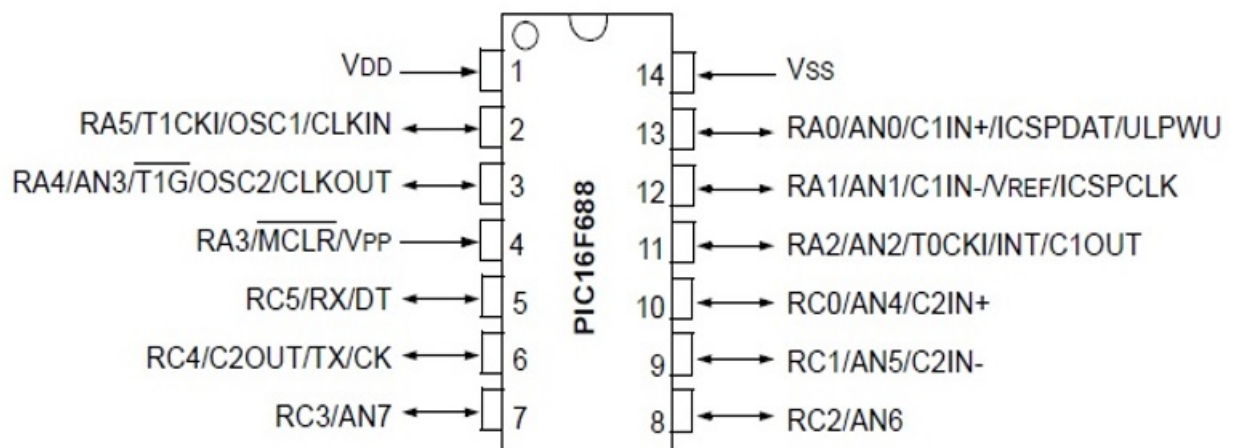


Fonte: (SOUZA, 2014)

2.6.2 Microcontrolador PIC16F688

O PIC16F688 é um microcontrolador da família PIC16 da Microchip e este pode funcionar a uma frequência de até 20MHz. Possui 14 pinos como é mostrado na Figura 2.11:

Figura 2.11 – Esquemático PIC16F688



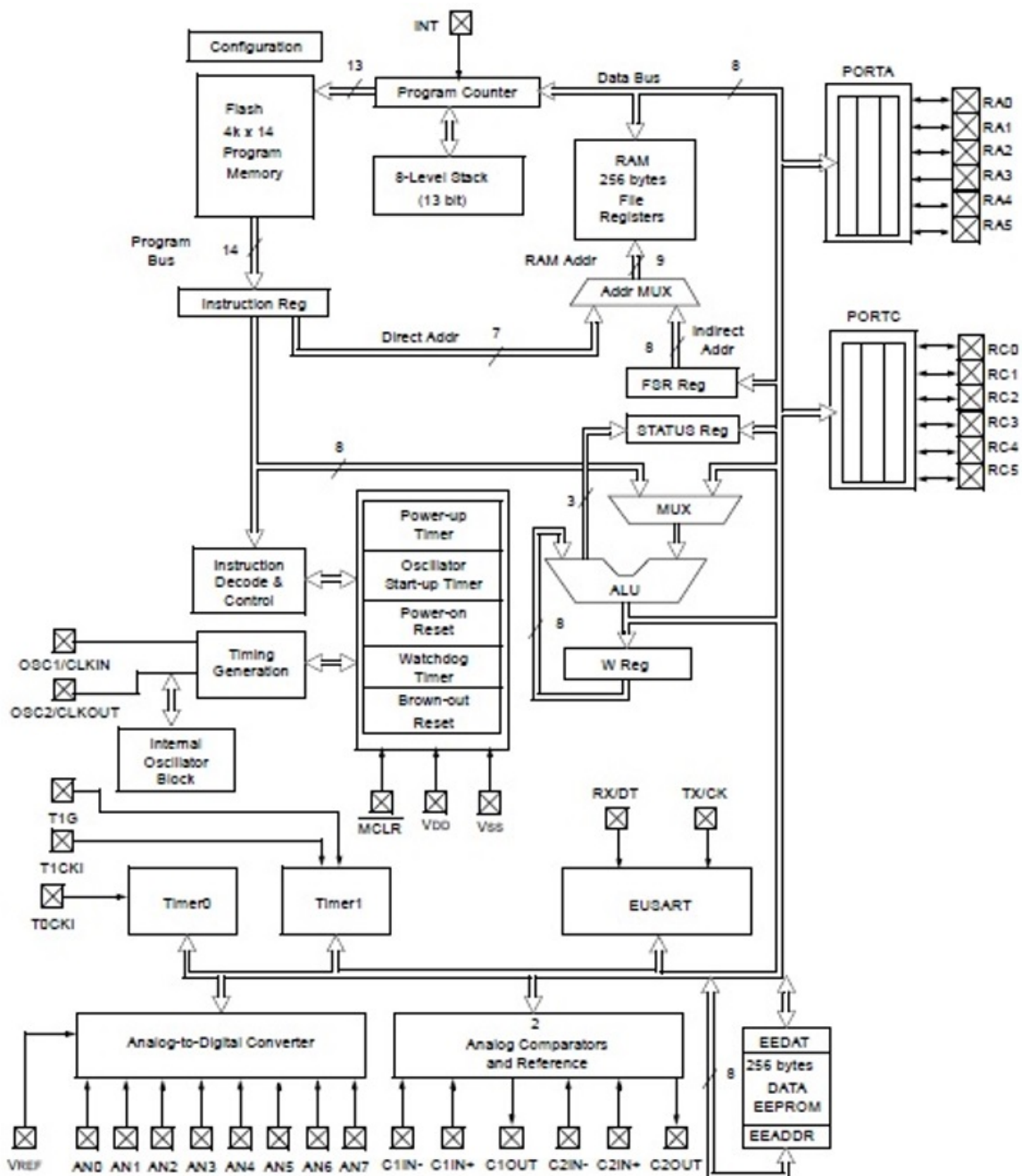
Fonte: (MICROCHIP, 2009)

Características importantes desse Microcontrolador de acordo com Microchip (2009) são:

- CPU RISC de alta-performance;
- Opera em correntes típicas de $11 \mu A$, 2V;
- Opera em até 20MHz com cristal externo;
- Economia de energia no modo Sleep;
- Resistente a altas temperaturas;
- Power-on Reset;
- Power-up Timer;
- Watchdog Timer melhorado em baixa-corrente;
- Proteção de código programável;
- Memória Flash/EEPROM de alta durabilidade;
- 12 pinos individuais de I/O;
- Conversor A/D;
- Módulo USART.

A Figura 2.12, representa o diagrama de bloco do PIC16F688, é mostrado as respectivas entradas e saídas, os pinos, registradores, e as configurações gerais:

Figura 2.12 – Diagrama de blocos do PIC16F688



Fonte: (MICROCHIP, 2009)

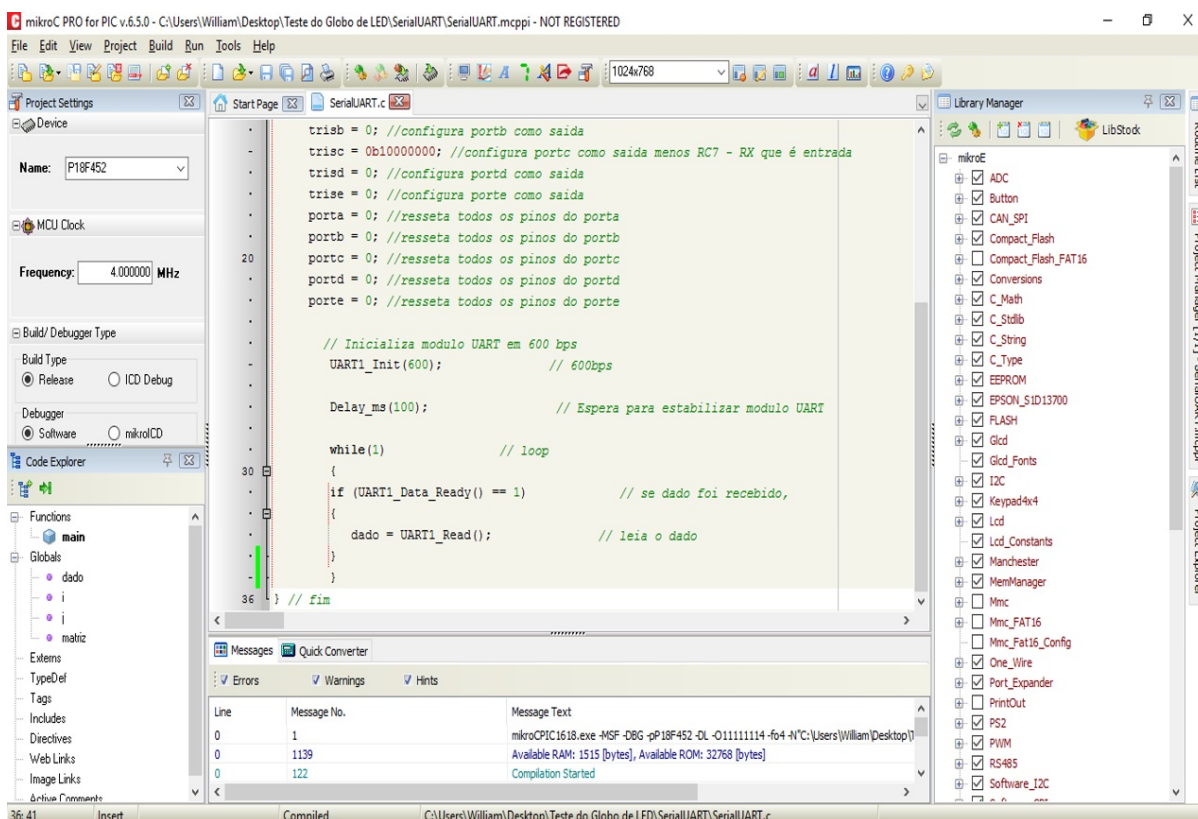
2.7 Compilador MikroC

Segundo MikroElektronika (2018), MikroC para PIC é um compilador cheio de recursos de ANSI C para dispositivos PIC da Microchip. O compilador é a melhor solução para o desenvolvimento de código para dispositivos PIC, pois dispõe de Integrated Development Environment (IDE) intuitivo, poderoso compilador com otimizações avançadas, muitas bibliotecas

de hardware e software, e ferramentas adicionais para ajudar no trabalho a ser desenvolvido. O compilador vem com arquivo de ajuda abrangente e muitos exemplos prontos para serem usados pelo usuário a fim de ele começar em qualquer momento o desenvolvimento. O compilador inclui atualizações grátis e um suporte técnico, de modo que pode contar com a ajuda durante o seu desenvolvimento. O autor afirma que o MikroC possui algumas características relevantes, a saber:

- O IDE melhor, mais rápido e mais produtivo;
- IDE envolvente, com vários recursos, códigos para utilizar, listas de rotinas, bibliotecas úteis, e muitos outros dispositivos;
- Suporte para várias ferramentas integradas e adicionais;
- Manual compreensivo para ajuda;
- Suporte para softwares adicionais.

Figura 2.13 – Interface gráfica do MikroC



Fonte: Elaborado pelo Autor

Como mostrado na Figura 2.13, a interface gráfica do MikroC possui vários acessórios que ajudam o programador para a construção do código para microcontrolador. No canto superior esquerdo da imagem, estão as configurações do projeto que está sendo criado, embaixo está o explorador do código, no centro está o código para a execução da tarefa determinada, embaixo estão as mensagens que são exibidas quando o código é compilado, como mensagens de erro, mostrando quanto de memória foi utilizada e quanto ainda está disponível. Por fim, no lado direito, está a biblioteca que contém todas as funções, que o programador pode acessar para saber sobre determinada função e onde deve ser mais adequado usá-la.

2.8 Redes Neurais Artificiais

Apesar da complexidade das redes neurais não permitir uma única definição, as linhas seguintes seguem como uma tentativa de algumas definições ou interpretações do que seja realmente uma rede neural.

Uma rede neural artificial é composta por várias unidades de processamento, cujo funcionamento é simples. Geralmente, essas unidades são conectadas por canais de comunicação que estão associados a determinado peso. As unidades fazem operações apenas sobre seus dados locais, que são entradas recebidas pelas suas conexões. O comportamento inteligente de uma Rede Neural Artificial vem das interações entre as unidades de processamento da rede.

Assim como o sistema nervoso é composto por bilhões de células nervosas, a rede neural artificial também seria formada por unidades que nada mais são que pequenos módulos que simulam o funcionamento de um neurônio. Estes módulos devem funcionar de acordo com os elementos em que foram inspirados, recebendo e retransmitindo informações.

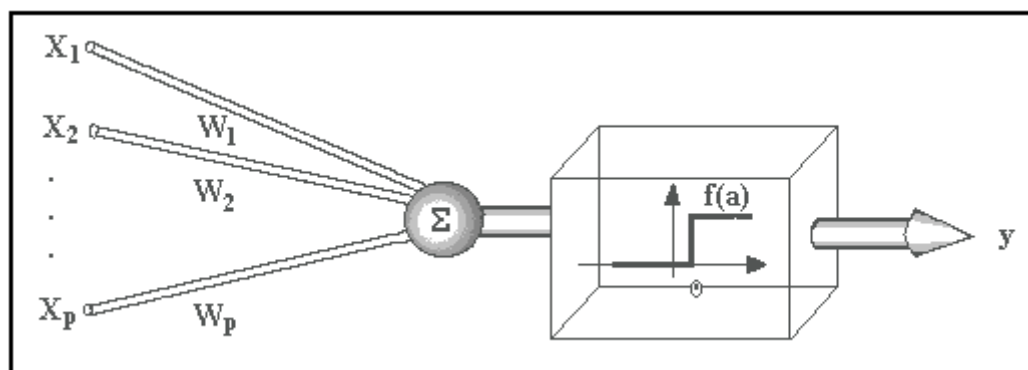
2.8.1 Neurônio Artificial

Um neurônio biológico pode ser interpretado como um circuito de entradas binárias combinadas por uma soma ponderada produzindo uma entrada efetiva. Proposto por McCulloch e Pitts (1943) (Figura 2.14), o processamento pode ser resumido da seguinte maneira:

- Sinais são apresentados à entrada;
- Cada sinal é multiplicado por um número, ou peso, que indica sua influência na saída da unidade;
- É feita a soma ponderada dos sinais que produz um nível de atividade;

- Se este nível de atividade exceder um certo limite (*threshold*) a unidade produz uma determinada resposta de saída.

Figura 2.14 – Modelo proposto por McCullock e Pitts



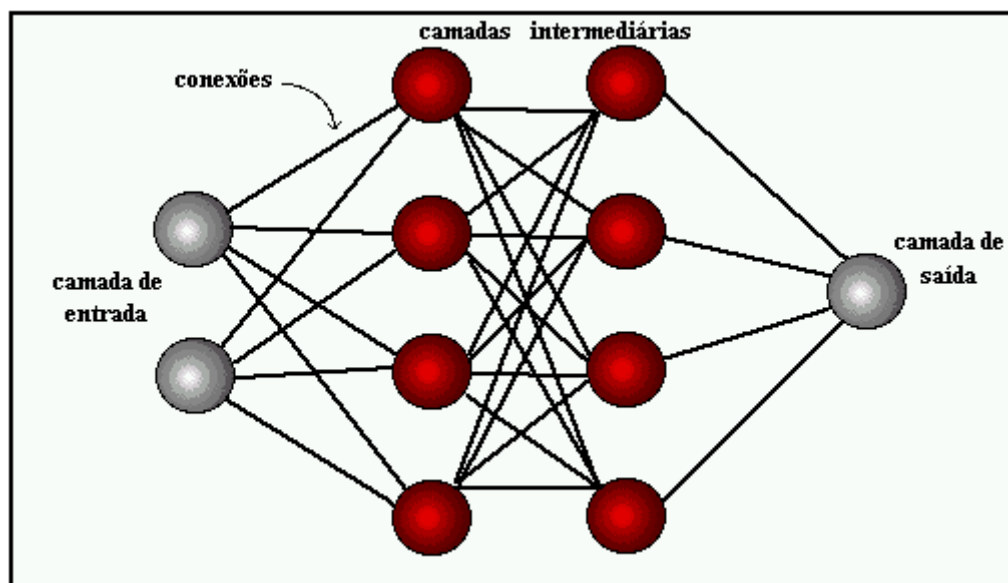
Fonte: (CARVALHO, 2009)

No modelo geral de neurônio, como mostrado na Figura 2.14, as entradas W_i , são combinadas usando uma função f , para produzir um estado de ativação do neurônio, as entradas chegam através dos dendritos e tem um peso atribuído pela sinapse (CARVALHO, 2009).

2.8.2 Arquitetura da Rede Neural

Arquiteturas neurais são tipicamente organizadas em camadas, com unidades que podem estar conectadas às unidades da camada posterior (Figura 2.15). A camada mais à esquerda da rede é chamada de camada de entrada e os neurônios dentro dela são chamados de neurônios de entrada. A camada mais à direita ou a saída contém os neurônios de saída. A camada intermediária é chamada de camada oculta, já que os neurônios dela não são entradas ou saídas.

Figura 2.15 – Camadas de uma Rede Neural



Fonte: (CARVALHO, 2009)

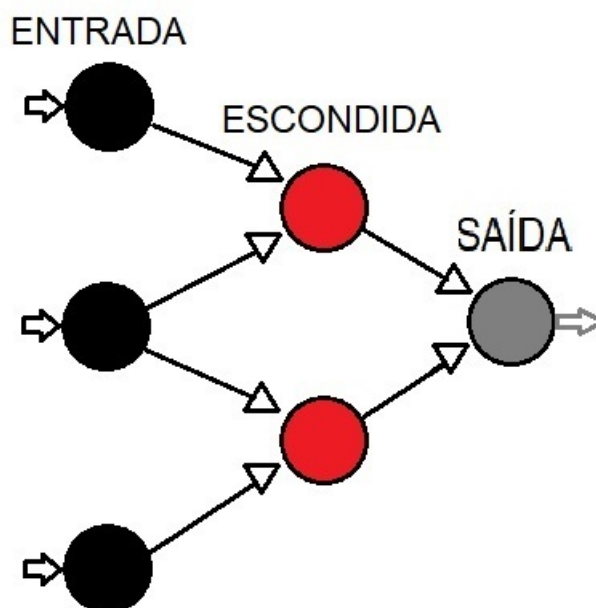
Usualmente as camadas são classificadas em três grupos como mostrado na Figura 2.15:

- Camada de Entrada: Onde os padrões são apresentados à rede;
- Camada Intermediária ou Escondida: Onde é feito a maior parte do processamento;
- Camada de Saída: onde o resultado final é apresentado.

2.8.3 Topologia de uma Rede Neural

Existem diferentes topologias de rede neural artificial, como por exemplo a *Feed-Forward* e *Cascade-Forward Back-Propagation*. As redes neurais do tipo *Feed-Forward*, mostradas na Figura 2.16, são redes cujas camadas se conectam as próximas sem retro-alimentação. Sendo assim, a informação é propagada pela rede em apenas uma direção e uma camada pode atualizar apenas o peso da próxima camada.

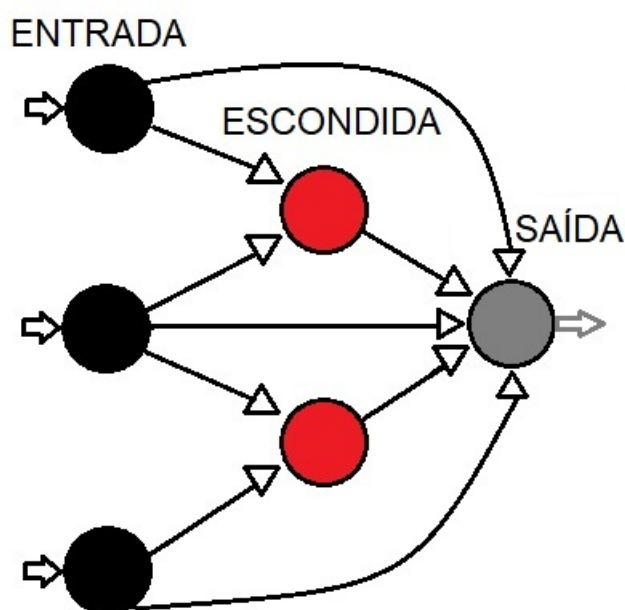
Figura 2.16 – Feed-Forward



Fonte: Elaborado pelo autor

A topologia de rede *Cascade-Forward Back-Propagation* são semelhantes às redes *Feed-Forward*, mas incluem uma conexão de peso da entrada para cada camada e de cada camada para as camadas sucessivas. Por exemplo, uma rede de três camadas tem conexões da camada 1 para a camada 2, da camada 2 para a camada 3 e da camada 1 para a camada 3 conforme mostrado na Figura 2.17. A rede de três camadas também possui conexões da entrada para todas as três camadas. As conexões adicionais podem melhorar a velocidade na qual a rede aprende o relacionamento desejado (BEALE et al., 2018).

Figura 2.17 – Cascade-Foward



Fonte: Elaborado pelo autor

2.8.4 Processos de aprendizado

A propriedade mais importante das redes neurais é a habilidade de aprender de seu ambiente e com isso melhorar seu desempenho. O aprendizado ocorre quando a rede neural atinge uma solução generalizada para uma classe de problemas. Isso é feito através de um processo iterativo de ajustes aplicado a seus pesos, ou seja, o treinamento.

Nesse contexto, existem os seguintes paradigmas de aprendizado:

- **Aprendizado Supervisionado:** quando é utilizado um agente externo que indica à rede a resposta desejada para o padrão de entrada;
- **Aprendizado Não Supervisionado (auto-organização):** quando não existe um agente externo indicando a resposta desejada para os padrões de entrada;
- **Aprendizado por Reforço:** quando um crítico externo avalia a resposta fornecida pela rede.

A maioria dos modelos de redes neurais possui alguma regra de treinamento, onde os pesos de suas conexões são ajustados de acordo com os padrões apresentados. Em outras palavras, elas aprendem através de exemplos.

2.9 Motor de Corrente Contínua

De acordo com a SIEMENS (2018), as máquinas de corrente contínua podem ser utilizadas tanto como motor quanto como gerador. Porém, uma vez que as fontes retificadoras de potência podem gerar tensão contínua de maneira controlada a partir da rede alternada, pode-se considerar que, atualmente, a operação como gerador fica limitada aos instantes de frenagem e de reversão de um motor.

Em conformidade com a SIEMENS (2018), o desenvolvimento das técnicas de acionamento de corrente alternada (CA) e a viabilidade econômica têm favorecido a substituição dos motores de corrente contínua (CC) pelos motores de indução acionados por inversores de frequência. Apesar disso, devido a suas características e vantagens, que serão analisadas adiante, o motor CC ainda se mostra a melhor opção em inúmeras aplicações, tais como:

- Máquinas de Papel
- Bobinadeiras e desbobinadeiras
- Laminadores
- Máquinas de Impressão
- Extrusoras
- Prensas
- Elevadores
- Movimentação e Elevação de Cargas
- Moinhos de rolos
- Indústria de Borracha
- Mesa de testes de motores

O motor de corrente contínua é composto de duas estruturas magnéticas:

- Estator (enrolamento de campo ou ímã permanente);
- Rotor (enrolamento de armadura).

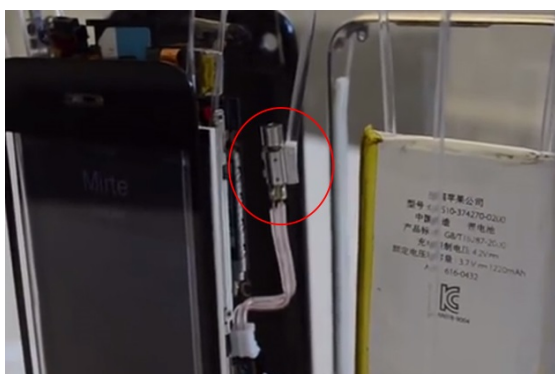
O estator é composto de uma estrutura ferromagnética com polos salientes aos quais são enroladas as bobinas que formam o campo ou um ímã permanente. O rotor é um eletroímã constituído de um núcleo de ferro com enrolamentos em sua superfície que são alimentados por um sistema mecânico de comutação. O sistema é formado por um comutador, solidário ao eixo do rotor, que possui uma superfície cilíndrica com diversas lâminas às quais são conectados os enrolamentos do rotor; e por escovas fixas, que exercem pressão sobre o comutador e que são ligadas aos terminais de alimentação. O propósito do comutador é o de inverter a corrente na fase de rotação apropriada de forma que o conjugado desenvolvido seja sempre na mesma direção (SIEMENS, 2018).

2.9.1 Motor VibraCall

O motor VibraCall, de corrente contínua, é um motor que simplesmente se liga na bateria para funcionar. Ele não apresenta possibilidade de controle de rotação, podendo estar apenas ligado e desligado. Pode-se considerar certa variação de RPM com diferentes tensões. Assim, ele é um dispositivo que vibra ao receber uma corrente elétrica.

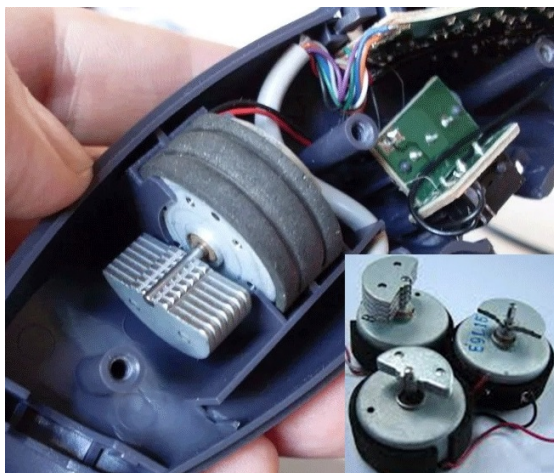
O dispositivo tem em média 15 x 5 milímetros e um eixo de metal em formato de meia-lua numa das pontas, podendo ser alimentado por uma bateria de 5V. Pela praticidade, o motor é muito utilizado em equipamentos eletrônicos, principalmente em aparelhos celulares (Figura 2.18), e controles de videogame (Figura 2.19).

Figura 2.18 – Motor vibracall de um celular



Fonte: (ALVES, 2013)

Figura 2.19 – Motor vibracall de um controle de videogame



Fonte: (RETROGAMES, 2018)

2.10 Microfone

Um microfone é considerado um transdutor. Um transdutor é um dispositivo que recebe um tipo de energia e o converte – ou traduz – em outro. Assim, o microfone recebe a energia acústica e o "traduz" em energia elétrica capaz de trafegar pelos cabos e ser processado e amplificado pelos aparelhos, de outro modo, em outras palavras, ele é o termo utilizado para falar dos elementos que transformam o som em energia elétrica, isto é, um dispositivo eletromecânico que transforma as vibrações mecânicas em corrente elétrica. Ele pode se apresentar em diversos tamanhos, tipos e formatos (Figura 2.20).

Figura 2.20 – Formatos do microfone



Fonte: (SILVA, 2018)

Algumas principais características do microfone de acordo com Oliveira (2010), são:

- *Resposta em frequência*: define como um determinado microfone responde às fontes sonoras com diferentes valores de frequência.
- *Impedância*: de uma forma geral, o valor de impedância do microfone define a carga que ele irá representar para o seu circuito de pré-amplificação. Este parâmetro também é importante para definir o comprimento do cabo de ligação do microfone. Os microfones de baixa impedância, inferior a 600 ohm, permitem a ligação através de cabos com grande comprimento – como 100m ou mais – sem perdas de sinal significativo. Os de alta impedância, com valores na ordem de 5000 ohm, limitam a utilização de cabos curtos – menor que 3 m.
- *Sensibilidade*: é a relação entre o nível elétrico de saída do microfone e a pressão sonora incidente. Este parâmetro mede o valor do sinal de saída – voltagem ou corrente – que o microfone produz a partir de uma pressão sonora, caracterizando assim sua eficiência.
- *Ruído de fundo*: em função da tecnologia de construção do microfone, sinais ruidosos de fundo podem ser inerentemente gerados. O próprio circuito de pré-amplificação necessário pode gerar ruídos.
- *Directividade*: é a característica que indica o padrão de sensibilidade de um microfone em função da direção ou posição da fonte sonora. Ela é, usualmente, indicada por um diagrama polar, onde o comprimento do raio numa direção – ângulo – indica a resposta em decibéis do microfone. Com respeito a sua directividade, um microfone pode ser:
 - *Omnidirecional*: é aquele que capta o som da fonte não importando a direção em que este chegue a sua cápsula. Observa-se que a sensibilidade pode variar em função da frequência do sinal sonoro, por ser comum a sensibilidade omnidirecional para baixas frequências e diferente sensibilidade para altas.
 - *Cardióides*: é aquele que capta com maior eficácia os sons emitidos a sua frente. À medida que a fonte sonora se desloca do eixo central do microfone, sua sensibilidade é reduzida. Seu diagrama polar aproxima-se de um formato de coração.
 - *Bi-direcionais / em forma de oito* : é aquele que capta o som igualmente no eixo da cápsula (0° e 180°), reduzindo sua sensibilidade em outros ângulos. Aplica-se muito bem em entrevistas.

Outras características de directividade podem definir outros tipos de microfone além dos acima descritos. Dentre eles, ressalta o “*shotgun*” que é uma microfone com a sensibilidade altamente focada em uma única direção (OLIVEIRA, 2010).

No microfone existe um diafragma, protegido por uma cobertura perfurada que permite a passagem do som e recebe as vibrações sonoras longitudinais. Quando alguma vibração do ar atinge o diafragma, ele a transmite para um sistema elétrico, que pode ser uma bobina móvel, um capacitor ou grãos de carvão, dependendo do tipo de microfone utilizado (SILVA, 2018).

No microfone de bobina móvel, a bobina é fixada na parte interna do diafragma e encontra-se próxima a um ímã permanente, cuja função é produzir um campo magnético na região onde está a bobina. A vibração do diafragma move a bobina, de acordo com a intensidade das ondas sonoras. Em função do campo do ímã e do movimento da bobina, uma corrente elétrica é induzida, seguindo o padrão das ondas sonoras que atingiram o diafragma. Por essa razão, as oscilações da corrente elétrica na bobina correspondem às vibrações do som que produziram seu movimento, isto é, o som foi convertido em corrente elétrica (SILVA, 2018).

No microfone de capacitor, uma de suas placas, que é móvel, é conectada ao diafragma, de modo que as vibrações sonoras possam ser transmitidas para ele. O capacitor é mantido carregado através de uma bateria. Vibrando solidariamente ao diafragma, a placa responde aos impulsos sonoros, variando sua distância com a outra placa, com isso altera a capacitância do capacitor. A mudança da capacitância produz uma corrente elétrica no circuito, que novamente varia segundo o padrão das vibrações sonoras originais (SILVA, 2018).

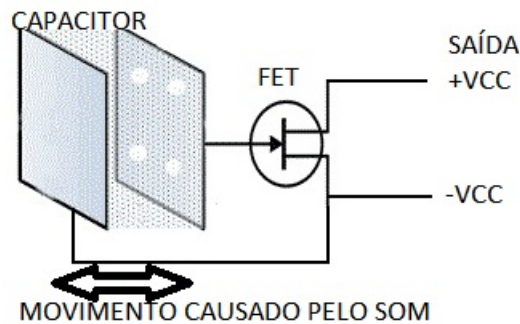
Por fim, o microfone de grãos de carvão, no qual as vibrações sonoras captadas pelo diafragma chegam até os grãos, fazendo variar a distância média entre eles. Esse conjunto de grãos de carvão é parte de um circuito elétrico, alimentado por uma fonte. Quando a distância média entre os grãos varia, altera-se a resistência elétrica do circuito que provoca uma corrente elétrica variável. Como nos tipos de microfone anteriores, o resultado final é a transformação das vibrações mecânicas do som em oscilações de uma corrente elétrica (SILVA, 2018).

2.10.1 Microfone de Eletreto

Os microfones de eletreto, pelo seu baixo custo, excelente sensibilidade e pequeno tamanho, são úteis numa infinidade de aplicações práticas. O microfone de eletreto é um tipo de microfone condensador que elimina a necessidade de uma fonte de tensão para se polarizar.

A polarização é obtida pelo uso de um material quase permanentemente carregado no dipolo capacitor, o eletreto, como é mostrado na Figura 2.21 (OLIVEIRA, 2010).

Figura 2.21 – Funcionamento do Microfone de Eletreto

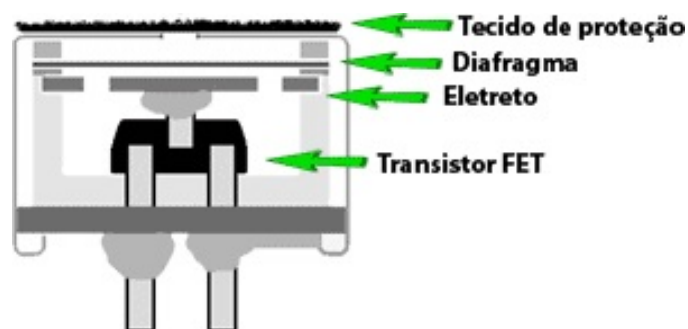


Fonte: Modificado de Eletronica (2018)

Oliveira (2010) alega que o microfone de eletreto é seguramente o tipo mais utilizado. Isto se deve a sua simplicidade de construção, por conseguinte ele está presente em diversos produtos populares, como por exemplo, câmeras de vídeo, celulares e computadores.

Diferentemente de outros microfones condensadores, os microfones de eletreto não requerem voltagem de polarização, mas eles contêm um pré-amplificador integrado de baixa potência. Este pré-amplificador é alimentado externamente – *phantom power* – ou por uma pilha simples de 1,5volts dentro do próprio microfone. O pré-amplificador é formado por um único transistor de efeito de campo (FET) como é mostrado na Figura 2.22 (OLIVEIRA, 2010).

Figura 2.22 – Partes do microfone de eletreto



Fonte: (ELETRONICA, 2018)

Para Oliveira (2010), existem três principais tipos de microfone de eletreto, dependendo de onde o material eletreto é colocado na cápsula:

- *Diafragma de eletreto - Foil-type or diaphragm-type*: um filme de material eletreto é usado como o próprio diafragma. Este é o tipo mais comum, mas também o de menor qualidade, uma vez que o material eletreto não faz um diafragma bom;

- *Eletreto trazeiro - Back electret*: um filme de eletreto é aplicado à placa trazeira da cápsula do microfone e o diafragma é feito de um material descarregado que pode ser mecanicamente mais adequado para o projeto do transdutor;
- *Eletreto frontal - Front electret*: neste novo tipo, a placa trazeira é eliminada do projeto, e o condensador é formado pelo diafragma e a superfície interna da cápsula. O filme de eletreto é aderido à cobertura frontal interna e o diafragma metalizado é conectado à entrada do *Field Effect Transistor* (FET) (Transistor de Efeito de Campo, em português).

2.11 Métodos de análise de desempenho

As análises de desempenho dos algoritmos têm como objetivos: agilizar a tomada de decisão usando informações confiáveis, verificar se o resultado esperado está sendo atingido, e identificar pontos de melhoria nas atividades executadas para se alcançar o resultado esperado.

A seguir serão apresentados algumas métricas de análise de desempenho que serão utilizadas neste trabalho.

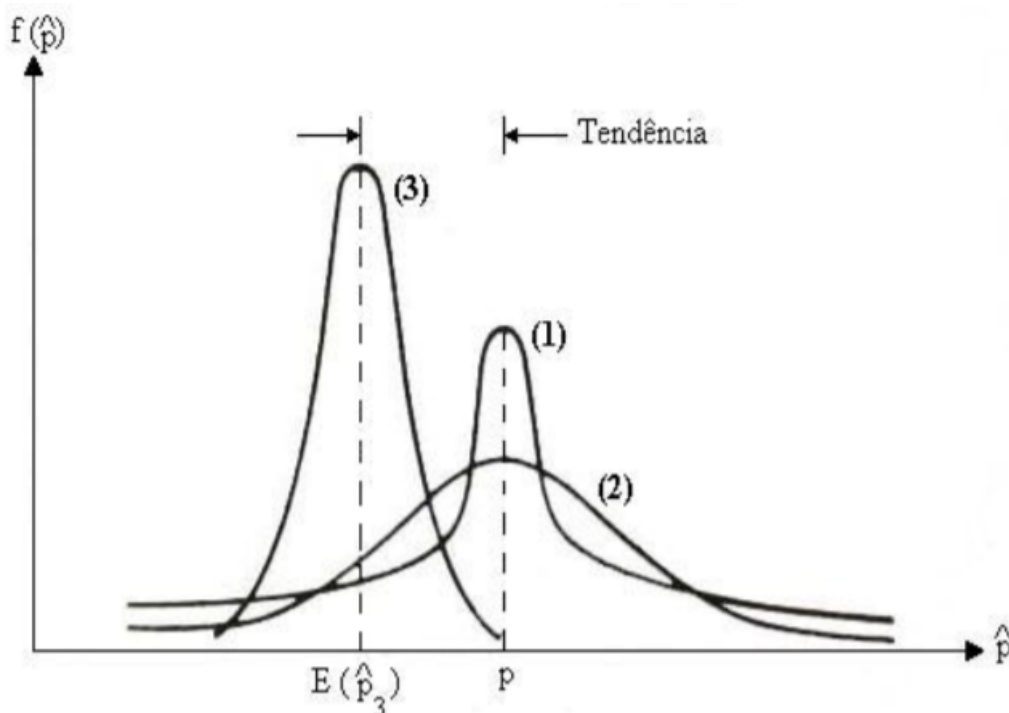
2.11.1 Acurácia e precisão

Mikhail e Ackermann (1976) apresentam acurácia como sendo o grau de proximidade de uma estimativa com seu parâmetro – ou valor verdadeiro –, enquanto precisão expressa o grau de consistência da grandeza medida com sua média. De acordo com os autores, acurácia reflete a proximidade de uma grandeza estatística ao valor do parâmetro para o qual ela foi estimada e que precisão está diretamente ligada à dispersão da distribuição das observações.

Precisão pode ser definida como o grau de conformidade entre as séries de observações da mesma variável aleatória, e que a dispersão da distribuição de probabilidade é um indicador da precisão (MIKHAIL; ACKERMANN, 1976).

Analisando a Figura 2.23, a estimativa p_2 é a menos precisa e que a estimativa p_3 é a mais precisa. Em termos de acurácia, ratificam que ela pode ser definida como o grau de proximidade que uma estimativa tem de seu parâmetro, ou seja, proximidade do valor verdadeiro. Acrescentam que p_1 e p_2 são igualmente acurados, pois estão assumindo p como referência, mas que nenhum deles é tão preciso quanto p_3 . Alguns problemas começam a aparecer a partir desta última afirmativa. Trata-se, no entanto, de uma contradição que pode ser facilmente solucionada (MONICO et al., 2009).

Figura 2.23 – Acurácia e Precisão



Fonte: (MONICO et al., 2009)

Para MONICO et al. (2009), a diferença entre precisão e acurácia advém da presença de erros sistemáticos, que se manifestam como uma tendência constante ou variável com tempo, afetando a estimativa – como o assunto deste artigo trata de uma tendência constante, doravante usaremos tendência apenas com esse sentido. À precisão se consideram apenas os efeitos aleatórios, enquanto à acurácia incluem-se não só os efeitos aleatórios, mas também os sistemáticos (MIKHAIL; ACKERMANN, 1976).

Uma medida simples de calcular é verificar a porcentagem de acertos do classificador. Por exemplo, se das 286 amostras da base, o classificador acertou 230, teremos uma acurácia final de $(230/286) \times 100 = 80,42\%$. Baseado nessa acurácia, não podemos dizer que este classificador é bom ou ruim. Uma melhor análise dos resultados devem ser feitas, e entender o que é importante para o problema em questão. Apenas o valor da acurácia não é suficiente para declarar a qualidade do classificador.

Outra medida de acurácia é apresentado por Mikhail e Ackermann (1976), que é proposta por Gauss, denominada erro quadrático médio (EQM), ou em inglês *mean square error* (MSE).

2.11.2 Erro Quadrático Médio

O Erro Quadrático Médio (EQM) é um critério para selecionar um estimador adequado, ou seja, em modelos estatísticos, os modeladores devem escolher entre vários estimadores do potencial. Em termos práticos, ele é igual à soma da variância e a tendenciosidade do quadrado do estimador. Um estimador é usado para deduzir o valor de um parâmetro desconhecido em um modelo estatístico. Por sua vez, tendência é a diferença entre o valor esperado do estimador e o verdadeiro valor do parâmetro estimado.

O EQM é obtido através da diferença entre o valor obtido pela estimação e o valor real ponderado pelo número de termos (Equação 2.35).

$$EQM = \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}, \quad (2.35)$$

onde \hat{y} é o valor estimado, y é o valor real e n é o número de termos.

2.11.3 Matriz de Confusão

A matriz de confusão é uma forma intuitiva de verificar como o classificador está comportando, verificando-se o desempenho do algoritmo de aprendizado, principalmente em algoritmos supervisionados.

De acordo com International Business Machines Corporation IBM (2018), em cada seção da visualização da Matriz de Confusão, o número de classes previstas, o número de classificações corretas e uma matriz de confusão são exibidos. A visualização da Matriz de Confusão fornece as seguintes seções:

- Matriz de Confusão para dados de treinamento
- Matriz de Confusão para dados de validação

Nesta técnica, as linhas da matriz representam as instâncias de uma classe de saída ou real e as colunas representam a classe prevista ou esperada, isto é, cada linha representa a classe das amostras que o classificador retornou e cada coluna representa a classe real.

Tabela 2.3 – Matriz de Confusão

	Positivo	Negativo	Total
Positivo	VP	FN	VP+FN
Negativo	FP	VN	FP+VN
Total	VP	VN+FN	VN+VP+FP+FN

Fonte: Modificado de Scalco (2016)

Existe uma nomenclatura específica para lidar com os acertos e erros de classificadores, como mostrado na tabela 2.3. As amostras preditas corretamente são denominadas como **Verdadeiros Positivos (VP)**, pois o resultado do classificador respondeu a pergunta em questão com *Positivo* e é correta *Verdadeira*. As amostras classificadas com *Positivo*, mas na verdade é *Negativo*, são denominadas **Falsos Positivos (FP)**. De maneira similar, na segunda linha da tabela possuem amostras classificadas como *Negativo*, mas que na verdade é *Positivo*. Essas são os **Falsos Negativos (FN)**. Por fim, as amostras *Negativas* classificadas corretamente são os **Verdadeiros Negativos (VN)**, pois neste caso a resposta do classificador é correta para classe *Negativa* em questão.

2.11.4 Coeficiente Kappa

Cohen (1960) desenvolveu um coeficiente de concordância - denominado Kappa - para escalas nominais que mede a relação de concordância. Essa medida de concordância usa todas as células da matriz de confusão, não apenas elementos diagonais. A estimativa de Kappa (K) - Equação 2.36 - é a proporção de concordância após acordo de acaso é removida da consideração,

$$\hat{K} = \frac{(p_o - p_c)}{1 - p_c} \quad (2.36)$$

no qual,

p_o = proporção de unidades corretas,

p_c = proporção de unidades que podem ter chances corretas, e

$p_o = \sum p_{ij}, p_c = \sum (p_{i+} + p_{+i}), p_{ij} = X_{ij}/N$

representa somatório sobre o índice.

onde N = numero total de contagens na matriz, e

X_{ij} = numero de contagens no ij -ésima célula.

Cohen (1960) mostra que, $\hat{K} = 0$ quando os resultados foram obtido por mero acaso. Valores positivos de Kappa representa maiores confiabilidade dos dados; Valores negativos de Kappa possuem uma chance menor de confiabilidade. O limite superior do Coeficiente Kappa (+1.00) ocorre apenas quando possui uma aceitação perfeita. O limite inferior do Coeficiente Kappa depende das distribuições marginais e isso não possui vantagens práticas (Tabela 2.4).

Tabela 2.4 – Valores Coeficiente Kappa

Coeficiente Kappa	Resultados
< 0,40	Pobre
0,40 - 0,75	Satisfatório a Bom
> 0,75	Excelente

Fonte: Modificado de Fonseca et al. (2007)

2.11.5 Gráfico ROC

O gráfico ROC (*Receiver Operating Characteristic*) é um método gráfico amplamente utilizado para avaliar o desempenho de classificadores. Segundo Prati et al. (2008), a gráfico ROC é baseado na probabilidade de detecção, ou taxa de verdadeiros positivos, e na probabilidade de falsos alarmes, ou taxa de falsos positivos.

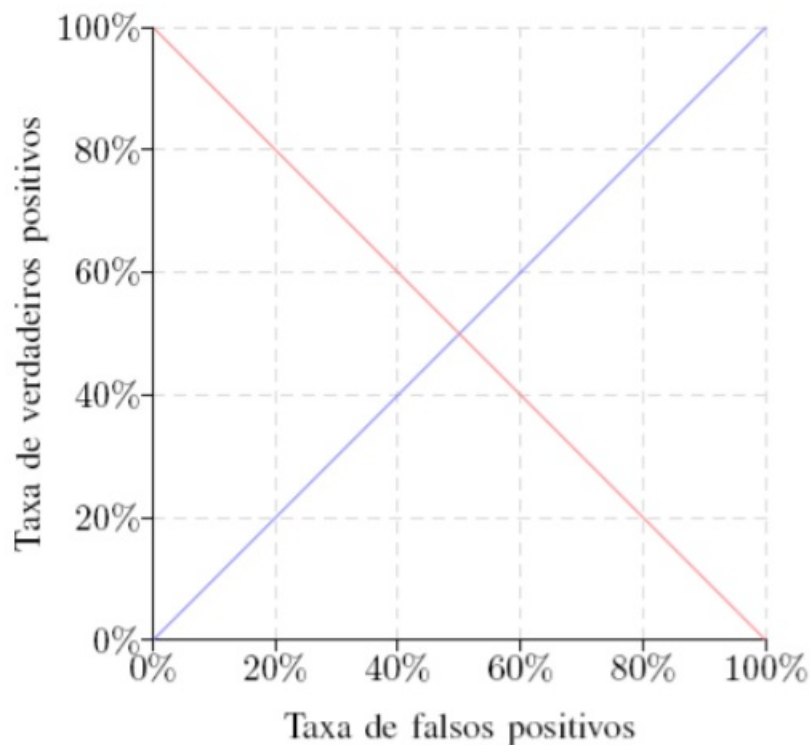
A construção de um gráfico ROC dá-se pela disposição da taxa de falsos positivos no eixo X e pela taxa de verdadeiros positivos no eixo Y (SCALCO, 2016).

De acordo com Prati et al. (2008), a interpretação do gráfico ROC é feita, baseando-se na Figura 2.24 e seguindo as seguintes diretrizes:

- Ponto (0,0): o modelo assume a estratégia de nunca classificar um exemplo como positivo, ou seja, não apresenta falsos positivos, mas também não classificam nenhum verdadeiro positivo.
- Ponto (1,1): assume a estratégia inversa do Ponto (0,0), ou seja, sempre classifica um exemplo como positivo.
- Ponto (0,1): representa o modelo de classificação perfeito, todos os exemplos positivos e negativos são, corretamente, classificados
- Ponto (1,0): representa o inverso do ponto (0,1), sempre faz as predições de forma incorreta.

Conforme Scalco (2016), pela Figura 2.24, percebe-se que os modelos mais próximos ao canto inferior esquerdo fazem uma classificação positiva somente se têm grande segurança na classificação, pois esses modelos podem ser considerados “conservadores”. Modelos desse tipo, geralmente, apresentam poucos falsos positivos, mas têm baixas taxas de verdadeiros positivos. Por outro lado, modelos próximos ao canto superior direito são considerados “liberais”, pois classificam a classe positiva com maior frequência, de tal maneira que realizam a classificação das amostras positivas, corretamente, porém com altas taxas de falsos positivos (PRATI et al., 2008).

Figura 2.24 – Exemplo de Gráfico ROC



Fonte: (PRATI et al., 2008)

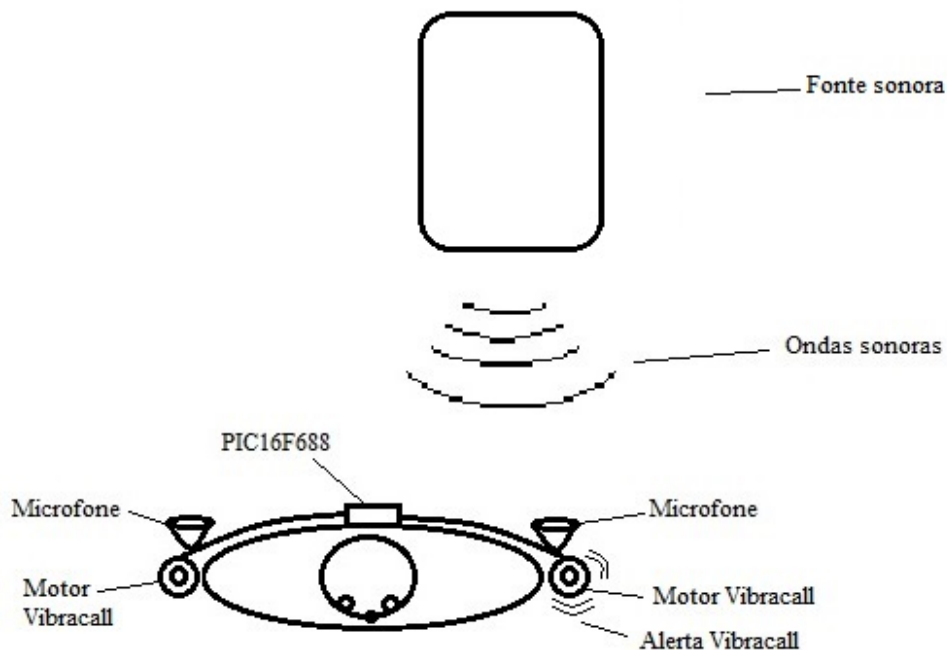
Analisando a Figura 2.24, em conformidade com Prati et al. (2008), a linha diagonal ascendente (Pontos (0,0) e (1,1)), é possível determinar que os pontos pertencentes ao triângulo superior esquerdo a essa diagonal representam modelos que trazem resultados melhores que o aleatório entretanto, no triângulo inferior direito, o resultado é o inverso. Já na diagonal descendente (Pontos (0,1) e (1,0), diagonal em vermelho) são representados os modelos de classificação que classificam, com eficácia equivalente, ambas as classes. À esquerda dessa diagonal são representados os modelos que desempenham melhor para a classe negativa, e à direita, os modelos que desempenham melhor para a classe positiva.

3 METODOLOGIA

Neste capítulo são abordados os materiais e métodos utilizados para a elaboração do projeto. Inclui a coleta dos dados, o desenvolvimento das técnicas de processamento de sinais para desenvolver as funções do sistema proposto e as etapas necessárias para a construção do mesmo.

A Figura 3.1 mostra a ideia principal do projeto. Quando a fonte sonora se aproxima pelo lado esquerdo do usuário, ele recebe o aviso através do motor *vibracall* localizado no seu ombro esquerdo, alertando-o da aproximação. O mesmo acontece quando a fonte sonora se aproxima pelo lado direito, que neste caso ativa o motor *vibracall* direito. A identificação de qual lado está a fonte sonora (direita ou esquerda) é realizada por meio do software executado pelo microcontrolador (PIC16F688).

Figura 3.1 – Funcionamento do Protótipo



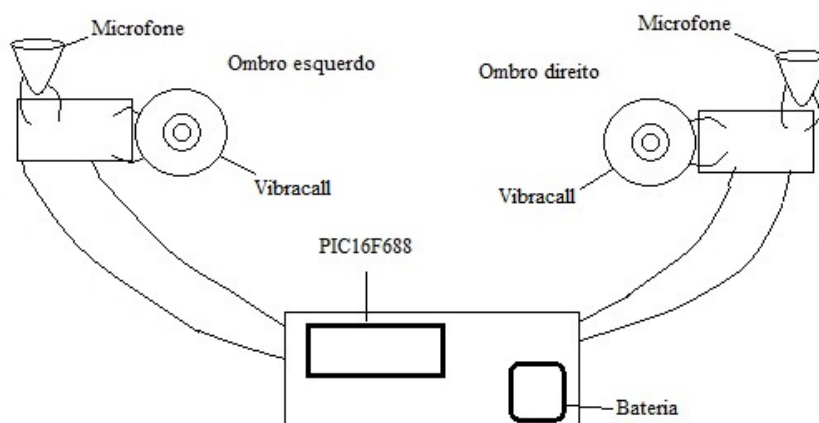
Fonte: Elaborado pelo Autor

O resultado final é uma vestimenta utilizada pelo usuário, que contém uma placa de PCI, integrando um microfone e um motor *vibracall* em cada ombro. No centro contém o microcontrolador PIC16F688, conectado com os motores e os microfones através de fios, para a comunicação e passagem de dados entre eles.

Sempre que aproximar alguma fonte sonora, os dois microfones captarão as ondas sonoras e enviarão os sinais para o microcontrolador PIC16F688. O software implementado no microcontrolador é responsável por identificar a direção dos sons através do defasamento dos sinais captados pelos microfones, acionará o motor *vibracall* mais próximo da fonte sonora, alertando o usuário a direção na qual a fonte sonora se aproxima.

Para o entendimento do protótipo é mostrado na Figura 3.2 o esboço do sistema. É exibido os principais componentes do projeto, como os motores *vibracall*, sensores (microfones), o microcontrolador e a fonte de energia que será uma bateria.

Figura 3.2 – Esboço do sistema a ser construído



Fonte: Elaborado pelo autor

Para melhor entendimento do trabalho desenvolvido, a metodologia utilizada foi dividida em 5 etapas:

1 - A primeira etapa foi a coleta dos dados, na qual foram adquiridos diversos tipos de sons em base de dados da web para simulação das técnicas utilizadas. Foi realizado o pré-processamento dos sons, efetuando os testes, gerando os atrasos e executando as adaptações necessárias.

2 - Na segunda etapa, simulações foram realizadas em software no PC para testes do método proposto, quando os sons coletados foram utilizados. Nesta etapa, a técnica de Rede Neural Artificial e também o algoritmo utilizando Somatório das Diferenças Absolutas foram implementados e testados em plataforma do MatLab[®] para detectar a direção de sinal sonoro por meio da defasagem entre dois sinais.

3 - A terceira etapa compreendeu o desenvolvimento da parte eletrônica, na qual foram selecionados os componentes específicos como o *vibracall*, o microfone, o microcontrolador PIC, circuito pré-amplificador, e tipo de fonte de alimentação. Em seguida, foi realizado o projeto esquemático no software Proteus[®]. De posse do projeto eletrônico completo, foi realizada a sua construção.

4 - A quarta etapa envolveu o desenvolvimento do software para o microcontrolador utilizando o compilador MikroC da empresa MikroElectronica. Nesta etapa, foram efetuados ajustes no hardware e software para adequar e garantir o funcionamento do sistema considerando as suas limitações.

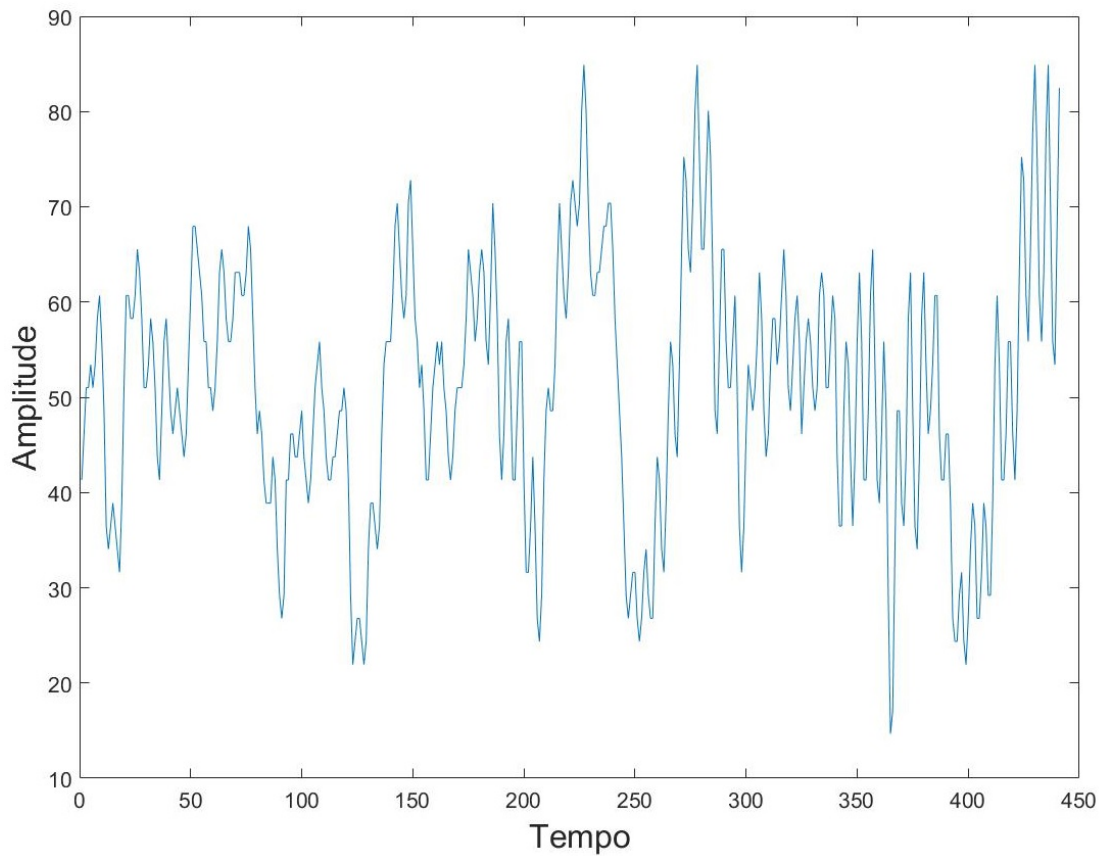
5 - Na quinta e última etapa, foram realizados os testes de funcionamento do sistema composto pelo hardware e software embarcados.

A seguir, cada etapa é detalhada.

3.1 Base de Dados

Para implementar o método de identificação da direção da fonte sonora, inicialmente foram realizadas simulações em software. Para isto, foi necessário a obtenção de dados que foram obtidos no repositório do site <<http://www.get-sounds.com/>> que contém diversos sons como: buzina, barulho de motor, derrapagem de pneu, trem, carro, ônibus, avião, e até sons de animais entre outros. Foram obtidos 40 diferentes tipos de sons em arquivo WAVE do repositório, cuja frequência de amostragem é de 22,225kHz. Na Figura 3.3 é apresentado a forma de onda de um sinal adquirido do repositório que representa o som de uma sirene de uma viatura policial.

Figura 3.3 – Exemplo de sinal obtido representando som de uma sirene policial

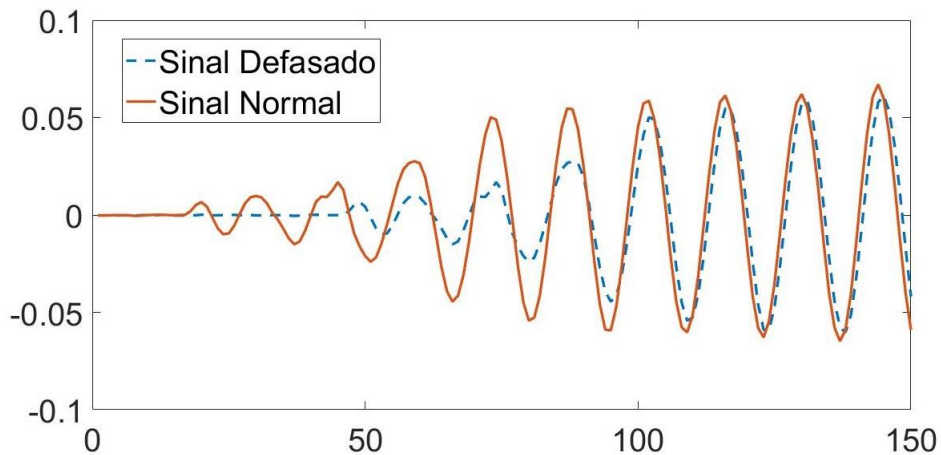


Fonte: Elaborado pelo autor

3.1.1 Pré-processamento dos Dados

Para realizar a simulação, foi considerado que os dois microfones do sistema proposto detectam o mesmo sinal sonoro, porém defasados no tempo devido à distância entre eles. Assim, a Figura 3.4 ilustra um exemplo de um sinal sonoro e seu respectivo defasamento: o sinal em vermelho representa o som captado pelo microfone direito enquanto o sinal azul representa o sinal captado pelo microfone esquerdo, indicando que a fonte do som está mais próxima do microfone direito.

Figura 3.4 – Defasamento de sinais



Fonte: Elaborado pelo autor

Para gerar os sinais defasados a serem utilizados na simulação, inicialmente foi medido a distância entre os ombros de uma pessoa que será utilizado como separação entre os sensores de som (microfones). Desta forma, constatou-se que a largura média estimada entre o ombro direito até o ombro esquerdo é de 30cm. A partir dessa informação, foram definidos possíveis atrasos para gerar os sinais defasados. Deste modo, realizaram-se os cálculos para determinar a quantidade de atrasos entre os microfones utilizando a fórmula da velocidade do som no ambiente dado por:

$$v = \frac{d}{t} \quad (3.1)$$

onde v é a velocidade do som no ambiente, d a distância do som até a fonte e t o tempo de propagação até o sensor. Utilizando a velocidade média do som $v = 343\text{m/s}$ (STEWART, 2009; BAUER et al., 2013; MENEZES, 2004), e a distância entre os microfones $d = 30\text{cm}$, para calcular o tempo t , para maior distância entre os microfones, aplicando a Equação 3.1, obtem-se

$$343\text{m/s} = \frac{30\text{cm}}{t} \quad (3.2)$$

desta maneira, tem-se o tempo máximo de atraso em

$$t \approx 0,87\text{ms} \quad (3.3)$$

Desta forma, no pior caso quando a fonte sonora estiver na linha reta que une os dois ombros, o tempo máximo de atraso entre os sinais captados é de 0,87ms.

Após cálculo do tempo máximo, é necessário calcular o número de amostras dos sinais que correspondem ao máximo de atraso, considerando que a taxa de aquisição do sinal coletado foi de 22,255kHz. A frequência (f) indica o número de ocorrências de um evento - ciclo ou oscilações - em um determinado intervalo de tempo. Esse tempo recebe o nome de período (T), e assim o cálculo do período é dado por:

$$T = \frac{1}{f} \quad (3.4)$$

onde T é o período e f a frequência do sinal. Assim:

$$T = \frac{1}{22255Hz} \quad (3.5)$$

deste modo, o intervalo de uma amostragem é:

$$T \approx 45\mu s \quad (3.6)$$

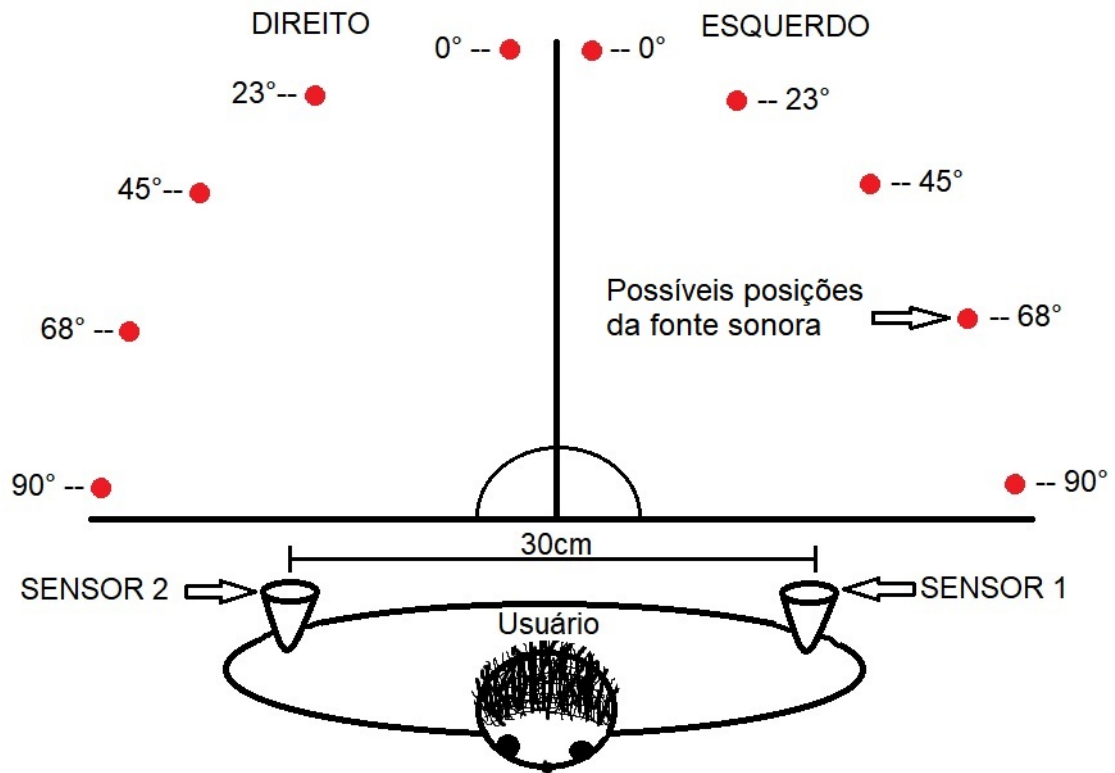
ou seja, uma amostra do sinal é adquirido a cada $45\mu s$. Para calcular o número máximo de amostras de atraso, tem-se:

$$Amostras = \frac{0,87ms}{45\mu s} \approx 18 \quad (3.7)$$

Então, o número máximo de amostras de atraso entre os sinais captados pelos microfones direito e esquerdo será de 18 amostras. Assim, para simular o atraso temporal entre os sinais captados pelos microfones basta atrasar um sinal dele mesmo no intervalo {-18 até +18} amostras, simulando que a fonte sonora está mais a esquerda ou mais a direita do usuário.

Os círculos vermelhos mostrados na Figura 3.5 são alguns dos possíveis ângulos no plano que correspondem as possíveis direções da fonte sonora em relação ao usuário. Em cada posição, a fonte sonora pode estar mais próxima ou mais afastada do usuário, pois o sistema irá identificar apenas o lado - direito ou esquerdo - em que a fonte sonora está em relação ao usuário.

Figura 3.5 – Possíveis posições da fonte sonora



Fonte: Elaborado pelo autor

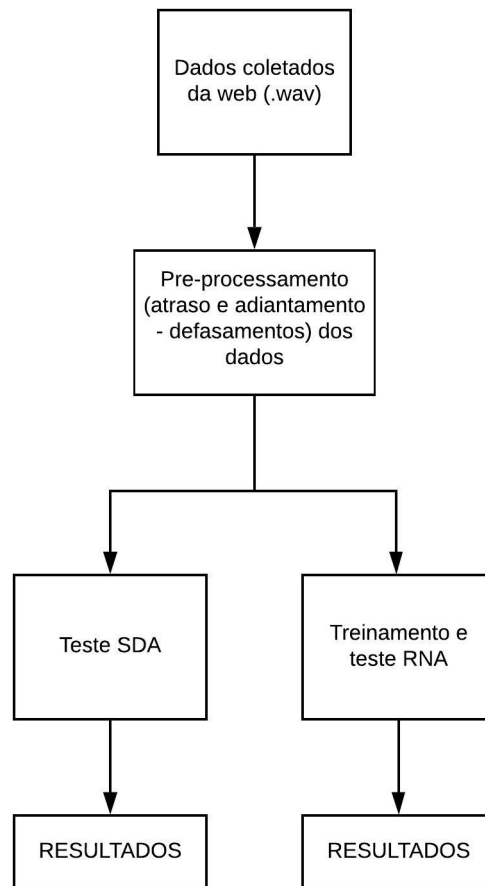
Cada posição da fonte sonora será representada pelo seu atraso associado, deste modo cria-se os vetores que contém as amostras para cada possível direção em que se encontra a fonte sonora.

3.2 Simulação do detector de direção de som no PC

Para o desenvolvimento do sistema de detecção, foi necessário a simulação em computador para a confirmação de que o método escolhido funcionaria no microcontrolador. Desta forma, desenvolveu-se a simulação no Matlab[®] com a finalidade de elaborar o sistema funcional.

Deste modo, buscou-se o desenvolvimento da Rede Neural Artificial aplicada ao problema de classificação, em que necessita-se identificar a direção – direita ou esquerda – da fonte sonora. Posteriormente, foi também testada a solução utilizando a técnica do Somatório das Diferenças Absolutas. Nas seções a seguir, cada método simulado é descrito. Na Figura 3.6 é apresentado um fluxograma simplificado da etapa de simulação.

Figura 3.6 – Fluxograma geral da Simulação

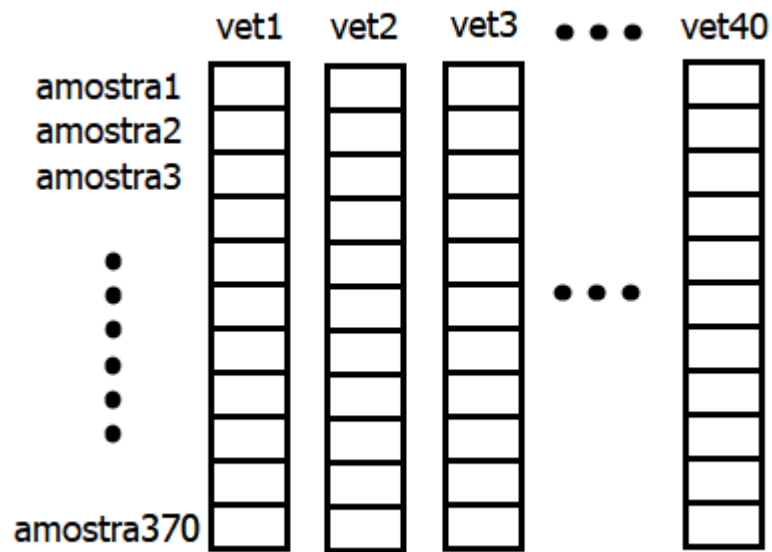


Fonte: Elaborado pelo autor

3.2.1 Rede Neural Artificial

Inicialmente, foram definidos os dados para treinamento e utilização da Rede Neural Artificial (RNA) da seguinte forma: cada arquivo exemplo de som foi armazenado em um vetor de amostras. Foram adquiridos 40 sons diferentes, totalizam 40 vetores, onde cada vetor possui 370 posições – referente à 370 amostras –, como é mostrado na Figura 3.7.

Figura 3.7 – Vetores de amostras

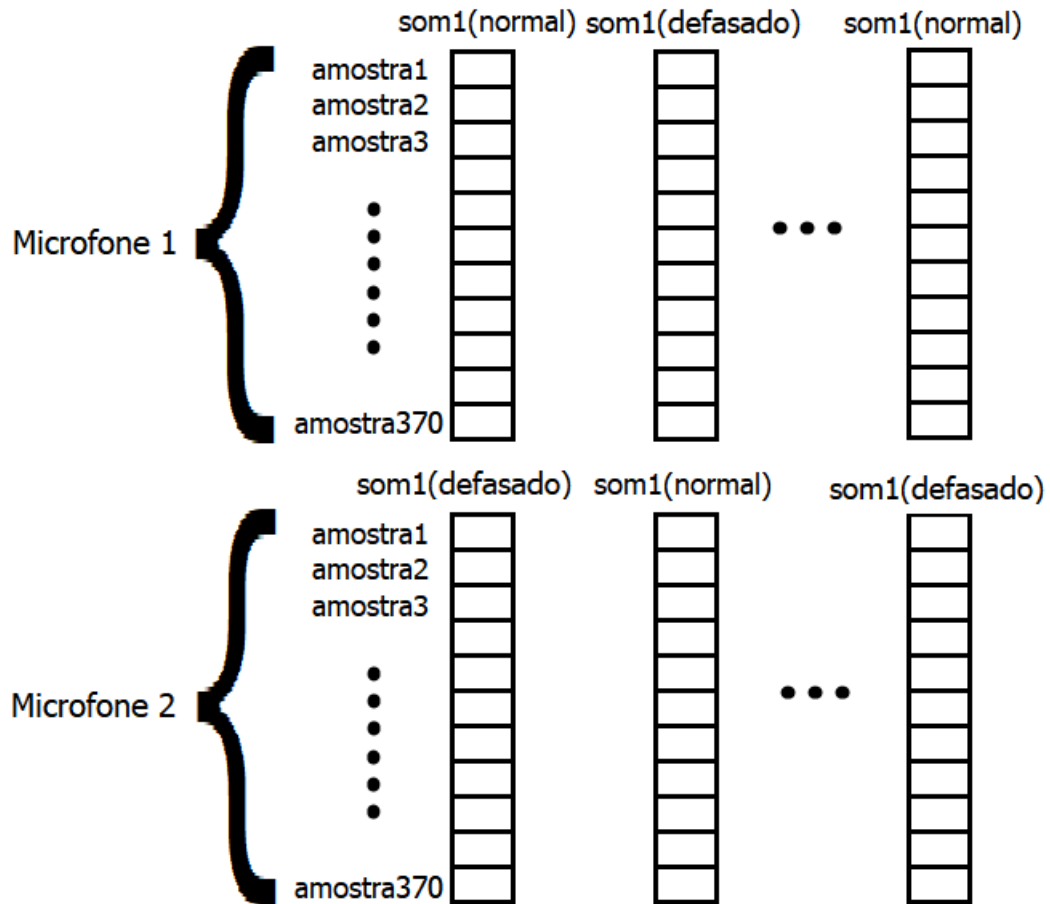


Fonte: Elaborado pelo autor

Para cada vetor de som estabelecido são gerados 6 novos vetores do mesmo som porém defasados. Assim, foram elaborados: 1 vetor normal e 6 vetores com defasamentos, 3 pela direita e 3 pela esquerda de modo que o defesamento tenha: -18, -12, -6, +6, +12, +18 amostras.

Por fim, os vetores são armazenados em uma matriz, da seguinte maneira: cada coluna da matriz possui um vetor defasado e um vetor sem defasamento do mesmo som correspondendo aos canais de som direito e esquerdo adquiridos. Como foram escolhidos 40 vetores de som inicialmente, então a matriz possui 240 colunas - representando um sinal sonoro defasado pela direita ou pela esquerda -, por 720 linhas - 370 amostras por canal -. A Figura 3.8 ilustra a matriz de dados obtida que é utilizada para treinar e testar a RNA.

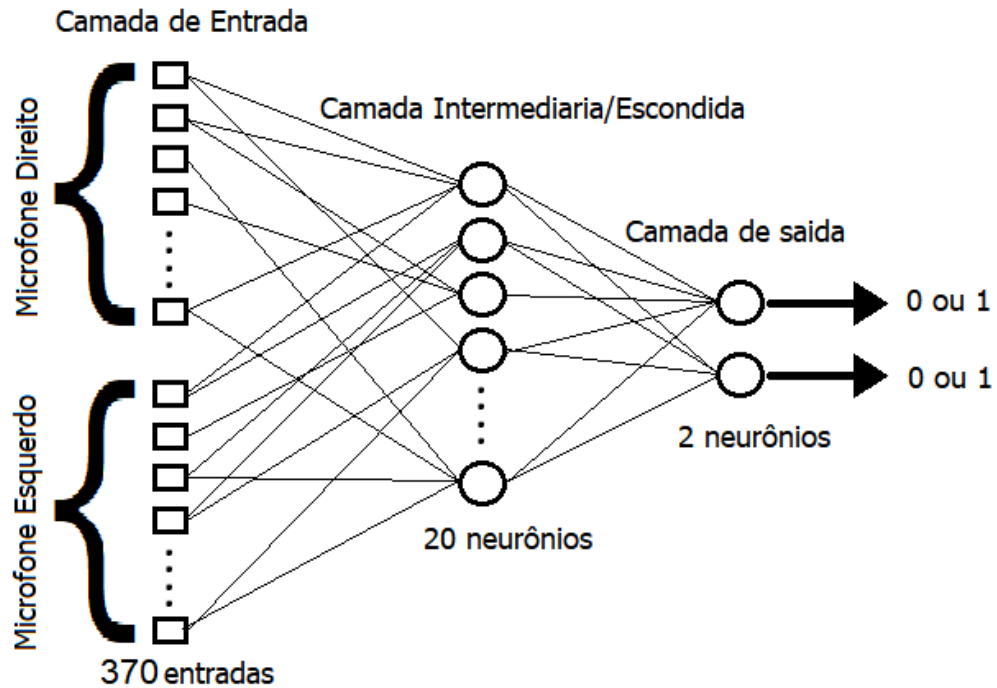
Figura 3.8 – Matriz de amostras



Fonte: Elaborado pelo autor

Para a escolha da Rede Neural Artificial, foram testadas diversas redes neurais e o melhor resultado foi obtido com as seguintes propriedades: 3 camadas, sendo uma camada de entrada composta por 720 entradas (representando os dois canais de som direito e esquerdo), uma camada escondida composta por 20 neurônios e uma camada de saída composta por 2 neurônios (representando as saídas Direita e Esquerda) como é mostrado na Figura 3.9. O treinamento foi realizado com limite de até 1000 épocas no software Matlab[®] utilizando o Toolbox ANN.

Figura 3.9 – Modelo da rede neural no Matlab®



Fonte: Elaborado pelo autor

Para realizar o treinamento supervisionado da RNA e testes de desempenho, além dos dados de entrada descritos acima, a RNA foi treinada e testada com a saída desejada organizada em uma outra matriz da seguinte forma: uma matriz de 2 linhas e 240 colunas em que cada linha representa a direção - primeira linha representa o lado direito, a segunda linha representa o lado esquerdo -, possuindo valores binários 0 ou 1, representado na Figura 3.10. Caso a primeira linha tenha o valor 1 e a segunda linha o valor 0, isso significa que o sinal está mais próximo do lado direito.

Figura 3.10 – Matriz de saída desejada da RNA

0	0	0	0	1	1	1	1
1	1	1	1	0	0	0	0

...

Fonte: Elaborado pelo autor

A fim de garantir um controle e avaliação da capacidade de generalização da RNA, os dados organizados em matrizes de entrada e saída desejada foram divididos aleatoriamente em conjunto de treinamento, validação e teste na seguinte proporção: 70%, 15%, 15%, que são valores *default* do Matlab®.

Foram implementadas duas topologias de RNA disponíveis no Toolbox ANN do Matlab[®], sendo elas: *Cascade-Foward* e *Feed-Forward*. Adicionalmente, quatro algoritmos de treinamentos foram executados para posterior comparação dos resultados, sendo eles: *One-Step-Secant (NOSS)*, *Gradient Descent with Momentum (GDM)*, *Gradient Descent with Momentum Adaptive LR (GDX)* e *Gradient Descent with Adaptive Learning Rate (GDA)*.

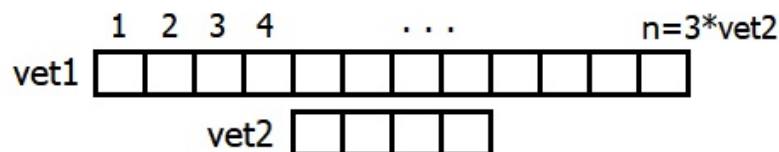
No Apêndice A é apresentado o *script* do Matlab[®] que implementa e testa a RNA.

3.2.2 Somatório das Diferenças Absolutas - SDA

Os dados utilizados para testar a técnica do Somatório das Diferenças Absolutas (SDA) foram os mesmos utilizados para a RNA. Para implementação da técnica, foi necessário gerar dois vetores de amostras, onde cada vetor representa os dados captados por um microfone (direito e esquerdo).

A coleta dos sinais é realizada de maneira que durante a aquisição das amostras, os dados do primeiro sinal é salvo no vetor 1 enquanto os dados do segundo sinal serão adquiridos a partir de quando o vetor 1 atingir um terço do seu tamanho total, e então salvos no vetor 2, como é mostrado na Figura 3.11. Isto foi necessário para a verificação do atraso pelo método SDA pois é feito o deslocamento (atraso e adiantamento) do vetor 2 em relação ao vetor 1 para realizar o cálculo do somatório das diferenças absolutas dos dois vetores a cada deslocamento. Após o cálculo, é realizada uma comparação para identificar o valor mínimo obtido para o SDA em determinado defasamento, e assim determinar o defasamento existente entre os dois sinais.

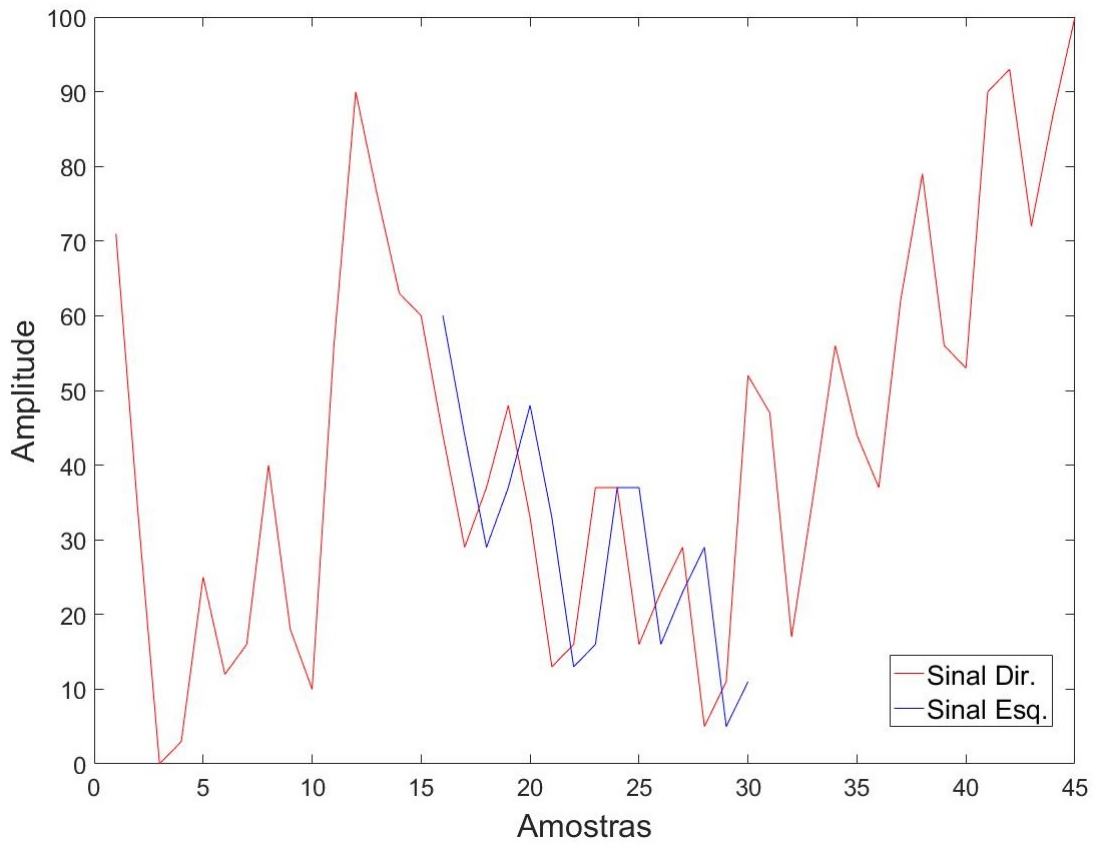
Figura 3.11 – Vetores com a amostragem dos sinais



Fonte: Elaborado pelo autor

Deste modo, para a simulação da técnica SDA foram criados vetores de sons com atrasos e adiantamentos para cada possível direção da fonte sonora, variando os atrasos entre os intervalos de -18 a +18 amostras. A Figura 3.12 apresenta dois sinais simulados correspondentes aos captados por dois microfones diferentes no qual o sinal azul (canal esquerdo) está uma amostra atrasada em relação ao sinal vermelho (canal direito).

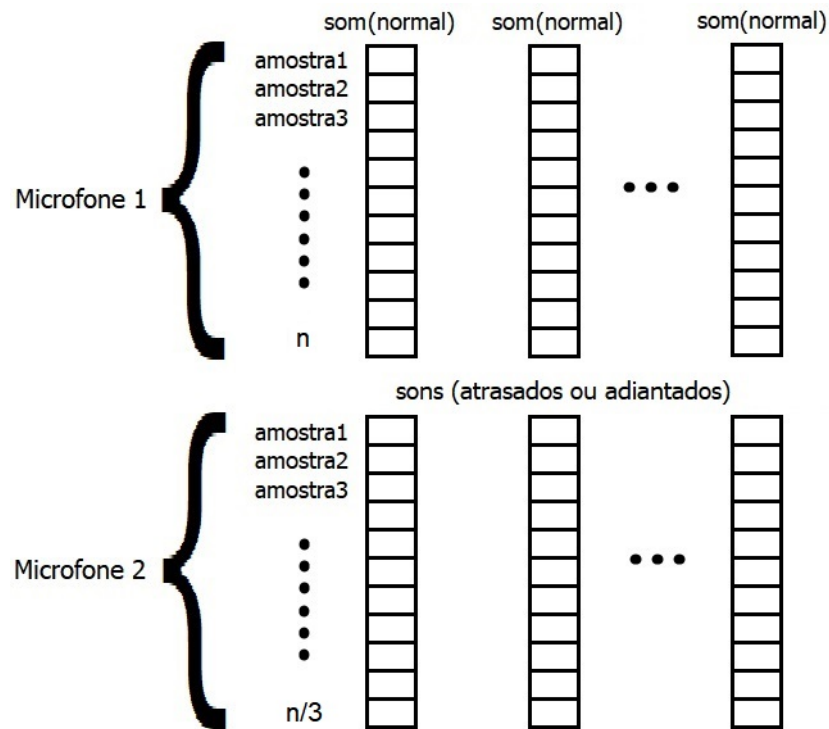
Figura 3.12 – Representação dos sinais adquiridos pelos microfones direito e esquerdo na simulação



Fonte: Elaborado pelo autor

Assim, cada vetor gerado pelas amostras dos sons originais possui seu respectivo vetor com atraso ou adiantado. Assim, cada par de vetor (normal e atrasado/adiantado) é armazenado de forma aleatória em uma matriz para a simulação, como é mostrado na Figura 3.13.

Figura 3.13 – Matriz de amostras

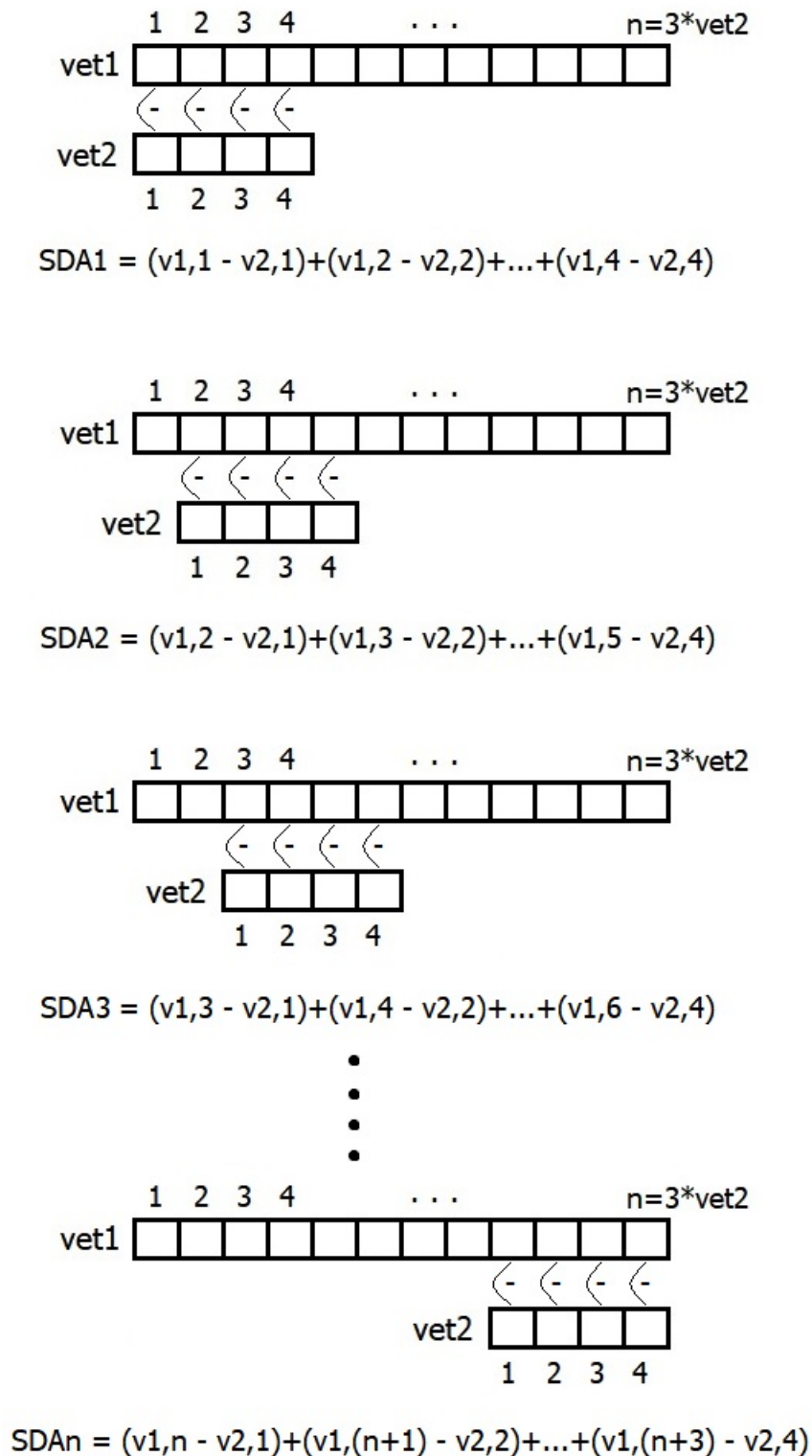


Fonte: Elaborado pelo autor

O vetor 1 são os dados adquiridos pelo microfone 1 (direito). O vetor 2 são os dados adquiridos pelo microfone 2 (esquerdo). O vetor 1 é 3 vezes maior que o vetor 2 porque o vetor 2 é que será deslocado (atrasado ou adiantado) para o cálculo do SDA.

O cálculo do SDA é realizado da seguinte forma: o vetor 2 é deslocado para o início do vetor 1 de forma que a primeira posição do vetor 2 esteja paralelo com a primeira posição do vetor 1. Assim, o valor da primeira posição do vetor 2 será subtraído pelo valor da primeira posição do vetor 1. Em seguida, o valor da segunda posição do vetor 1 é subtraído pela segunda posição do vetor 2 e o resultado é somado com o resultado da primeira subtração. Após o cálculo percorrer todo o vetor 2, o mesmo é deslocado uma posição e é realizado novamente o somatório, como é mostrado na Figura 3.14.

Figura 3.14 – Deslocamento do vetor durante o cálculo do SDA



Fonte: Elaborado pelo autor

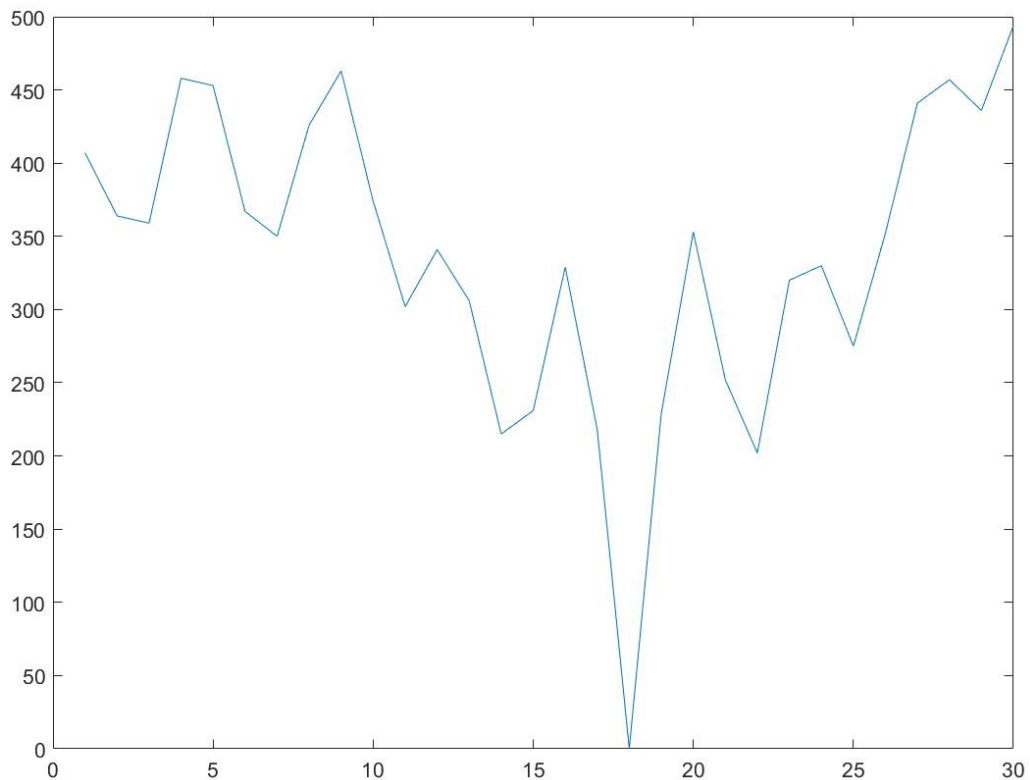
Após o cálculo efetuado pelo algoritmo do SDA, é realizada a verificação para encontrar o valor mínimo. Assim, a cada cálculo de um novo SDA, o novo valor encontrado é comparado

com o valor atual, e se for maior ele é descartado, mantendo o valor atual. Caso o novo SDA seja menor que o valor atual, então ele é armazenado substituindo o SDA atual.

Após realizado todo processo e o deslocamento do vetor 2 atingir o final do vetor 1, tem-se o menor valor calculado do SDA e a posição do vetor 2 no qual foi obtida. Através da posição obtida para o valor mínimo do SDA, é determinado em qual direção está localizada a fonte sonora: direita ou esquerda.

Utilizando a Figura 3.12 como exemplo para os sinais captados pelos microfones direito e esquerdo, e aplicando o SDA tem-se como resultado o gráfico mostrado na Figura 3.15, onde visivelmente é possível identificar o valor mínimo encontrado para o SDA na posição 18, consequentemente identificando a direção em que o sinal sonoro se encontra.

Figura 3.15 – Gráfico gerado pelo algoritmo SDA



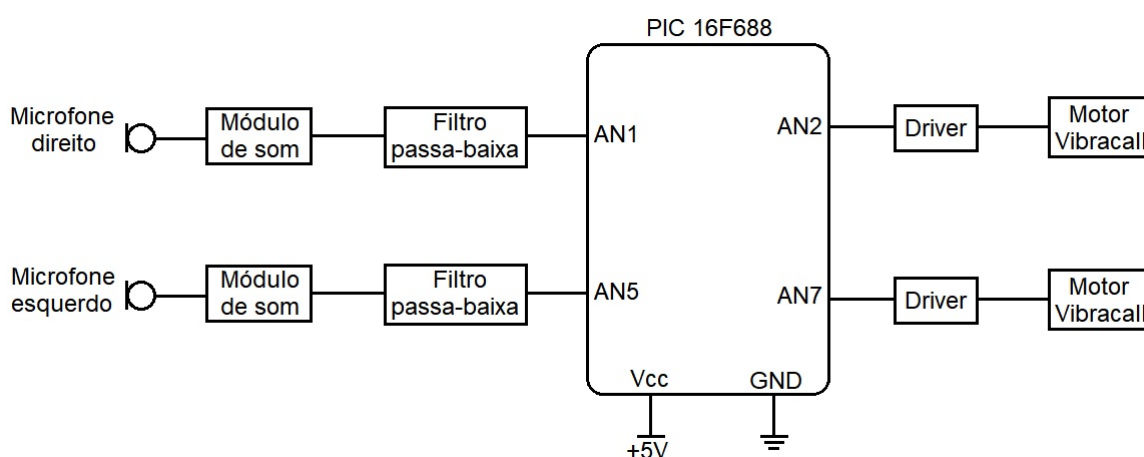
Fonte: Elaborado pelo autor

No Apêndice B é apresentado o *script* do Matlab[®] que implementa o algoritmo SDA responsável pela detecção da direção do sinal sonoro pela defasagem entre os sinais de entrada.

3.3 Desenvolvimento do Hardware do Sistema Embarcado

Nessa seção são mostrados os componentes eletrônicos que foram utilizados, o desenvolvimento do sistema embarcado e suas especificações. Apresenta-se também os motivos que levaram a selecionar tais componentes e a construção dos circuitos. Na Figura 3.16 é apresentado um diagrama de blocos simplificado do hardware do sistema embarcado desenvolvido.

Figura 3.16 – Diagrama do Hardware



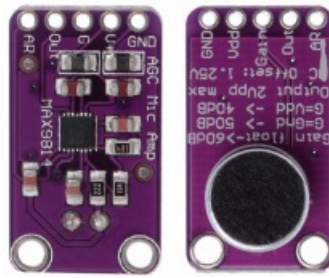
Fonte: Elaborado pelo autor

A seguir, nas próximas subseções são explicadas cada parte do hardware.

3.3.1 Módulo Amplificador Microfone Eletreto MAX9814

O Módulo Amplificador Microfone Eletreto MAX9814 foi utilizado pelo tamanho ser ideal para o sistema, pois ele contém um microfone de eletreto e inclui um circuito amplificador, como mostra a Figura 3.17. Ele dispõe de um baixo consumo de corrente de aproximadamente 3mA e é capaz de funcionar quando alimentado com uma tensão entre 2,7V - 5,5V, além de possuir alta sensibilidade e ser ajustável, tornando-o, assim, ideal para o projeto, pois a identificação do som e a qualidade são importantes para o sistema.

Figura 3.17 – MAX9814



Fonte: Elaborado pelo autor

3.3.2 Filtro passa-baixa

Devido a limitação do conversor analógico digital (CAD) do microcontrolador PIC16F688, em que necessita de aproximadamente $50\mu s$ para realizar a conversão do sinal analógico para digital de cada canal utilizado, desta forma tornou-se necessário a utilização do filtro. Com a utilização de 2 canais de entrada, onde cada canal recebe o sinal de cada microfone, foi necessário determinar a frequência máxima para conversão.

Como cada canal do CAD necessita de $50\mu s$ em cada conversão, tem-se:

$$50\mu s + 50\mu s = 100\mu s \quad (3.8)$$

que representa o tempo total gasto para converter os 2 canais.

Então, para calcular a frequência de amostragem máxima, tem-se

$$f_a = \frac{1}{T} \quad (3.9)$$

onde f_a é a frequência de amostragem e T é o período. Desta maneira:

$$f_a = \frac{1}{100\mu s} = 10kHz \quad (3.10)$$

O teorema de Nyquist, define que:

$$f_a > 2f_s \quad (3.11)$$

ou

$$f_s < \frac{f_a}{2} \quad (3.12)$$

onde f_s é a máxima frequência do sinal amostrado.

Aplicando o teorema, tem-se:

$$f_s < \frac{10kHz}{2} \quad (3.13)$$

ou

$$f_s < 5kHz \quad (3.14)$$

Assim, pelo teorema de Nyquist, a máxima frequência de sinal de entrada é de até 5kHz. Deste modo, optou-se pela utilização do filtro passa-baixa de primeira ordem com frequência de corte próxima em 500Hz (dez vezes menor que a frequência máxima de aquisição) devido a sua simplicidade (utiliza apenas 1 capacitor e 1 resistor) para “eliminar” as frequências acima da desejada.

O cálculo da frequência de corte (f_c) do filtro de primeira ordem é dado por:

$$f_c = \frac{1}{2\pi RC} \quad (3.15)$$

onde R é o valor do resistor e C o valor do capacitor.

O filtro foi calculado estimando o valor do capacitor, e então calculou-se o valor do resistor. Na Equação 3.16 é mostrado o cálculo utilizando os valores dos componentes escolhidos para o filtro RC.

$$f_c = \frac{1}{2\pi 1k\Omega * 330nF} = 482Hz \cong 500Hz \quad (3.16)$$

3.3.3 Microcontrolador PIC16F688

A partir de buscas realizadas com diversos microcontroladores PIC da Microchip, constatou-se que deverá ser utilizado um microcontrolador com poucos pinos, uma vez que são necessários apenas 4 pinos, 2 para cada dispositivo em cada ombro e que deve apresentar boa capacidade de armazenamento/processamento para conter a rede neural. Assim foi escolhido o PIC16F688 que contém 14 pinos – Figura 3.18 – e buscando aproveitar o maior número de pinos, definiu-se da seguinte maneira:

- 2 pinos para o cristal;
- 1 pino para o botão reset;
- 2 contendo uma saída e entrada para ombro direito;
- 2 contendo uma saída e entrada para ombro esquerdo;
- 1 pino para o LED que é utilizado para testes.

Figura 3.18 – Microcontrolador PIC16F688



Fonte: Elaborado pelo autor

3.3.4 Motor VibraCall

O motor *vibracall* mostrado na Figura 3.19, é um motor de pequenas dimensões, utilizado em celulares para vibrarem quando é recebida uma ligação ou notificações como uma SMS (mensagens de texto), podendo ser utilizado em diversas aplicações em placas contendo Arduíno, microcontroladores PIC, ARM (*Acorn RISC Machine*), entre outros. O motor possui dimensões de 11x5x4mm, e pesa em média 1g, funciona com corrente elétrica entre 0,09A e 0,2A e tensão de 1,5V à 5V.

Figura 3.19 – Motor VibraCall



Fonte: Elaborado pelo autor

3.3.5 Desenvolvimento do esquemático eletrônico

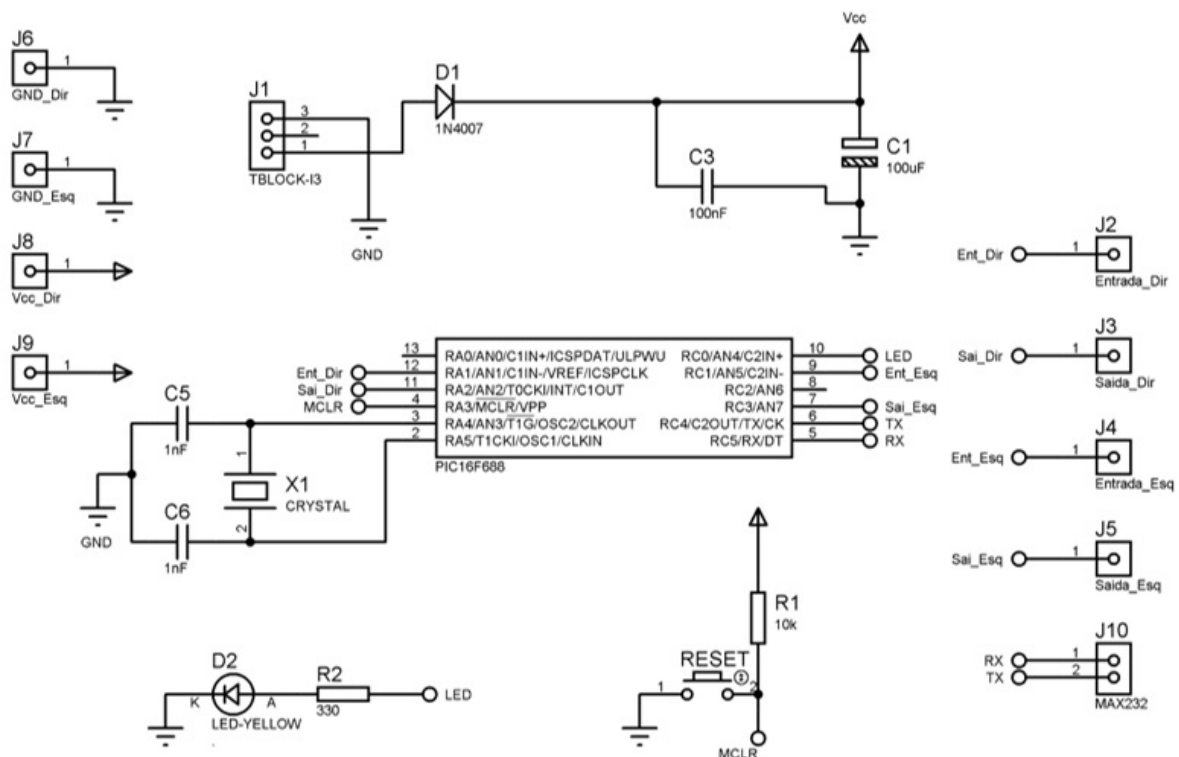
Para o desenvolvimento do esquemático eletrônico do protótipo, foi utilizado a versão de estudante do software Proteus[®]. Este é um software de baixo custo que oferece facilidades

para desenhos esquemáticos e simulação de funcionamento - Simulation Program with Integrated Circuits Emphasis - SPICE - e layout de Placa de Circuito Impresso (PCI). O módulo de simulação inclui simulação do microcontrolador PIC16FXX, PIC18FXX, diversas outras famílias PIC e o PIC16F é a mesma utilizada no projeto.

O software possui uma gama de modelos periféricos interativos para ser utilizado que inclui telas LCD, teclados e terminais RS-232. O Proteus[®] é desenvolvido para design de placas de circuito impresso e projetos eletrônicos, também é constituído pelo Intelligent Schematic Input System (ISIS) – para captura de esquemas, simulação SPICE e simulação de microcontroladores da família PIC, com voltímetro, osciloscópio e gerador de funções – e pelo Advanced Routing and Editing Software (ARES) – que serve para criar layouts de placas de circuito impresso –, que foi utilizado para criar o esquemático do projeto.

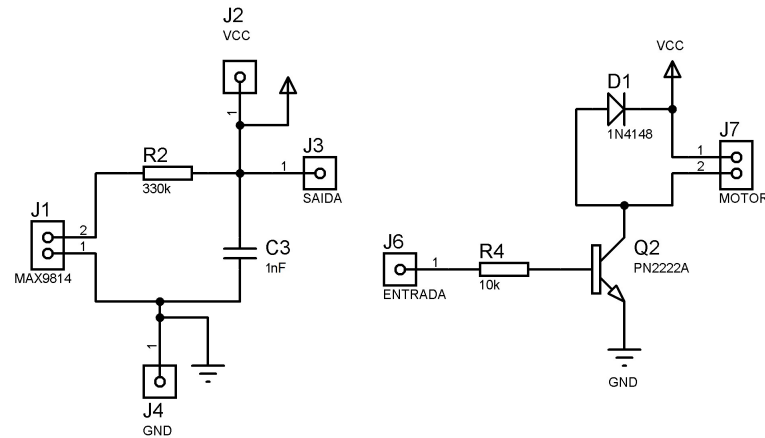
Por isso, uma das vantagens para a utilização deste programa, é que o usuário pode simular circuitos e criar protótipos. A Figura 3.20 exibe o esquemático desenvolvido para a placa central, que contém o microcontrolador. A Figura 3.21 mostra o esquemático das placas que serão utilizadas em cada ombro.

Figura 3.20 – Esquemático da placa central contendo microcontrolador



Fonte: Elaborado pelo autor

Figura 3.21 – Esquemático da placa que será utilizada em cada ombro



Fonte: Elaborado pelo autor

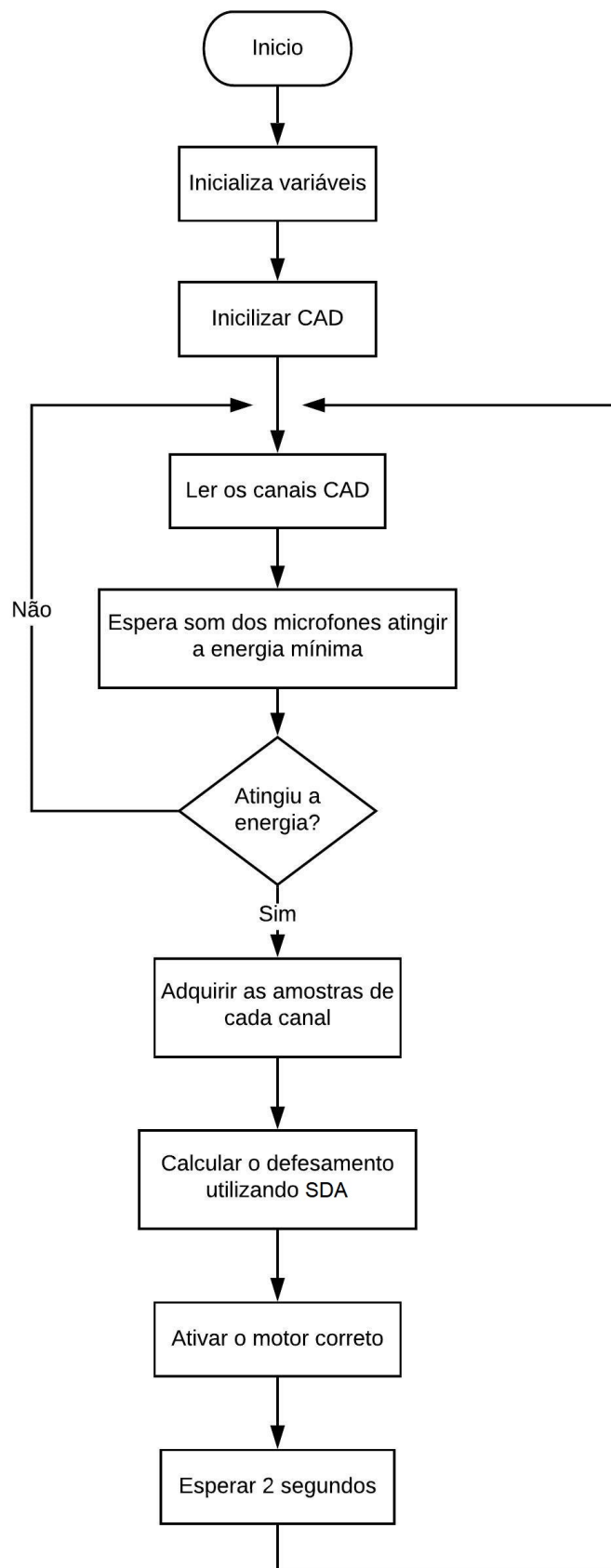
3.4 Desenvolvimento do Programa para Microcontrolador

Devido à limitação do microcontrolador escolhido (PIC 16F688) que possui apenas 256 bytes de memória de dados e 7k posições de memória de programa, não foi possível implementar a Rede Neural Artificial simulada anteriormente. Devido a esse impedimento, a técnica implementada no microcontrolador para a detecção da direção da fonte sonora foi o Somatório das Diferenças Absolutas (SDA), visto que o algoritmo coube na memória de programa do microcontrolador e contém menos cálculos comparado a RNA.

Assim, foi desenvolvido o programa em C baseado no algoritmo SDA implementado no Matlab[®] descrito anteriormente com algumas adaptações. Foi utilizado a IDE MikroC[®], que é uma IDE para programação da Mikroelektronika para microcontroladores.

O fluxograma mostrado na Figura 3.22 apresenta a sequência de tarefas necessárias para o microcontrolador executar. O fluxograma para o microcontrolador difere um pouco da simulação no Matlab[®], dado que é necessário inicializar as variáveis padrões, os pinos e também o conversor analógico digital (CAD) que irá realizar a aquisição dos sinais que serão expostos no protótipo.

Figura 3.22 – Fluxograma do algoritmo de simulação no Microcontrolador



Fonte: Elaborado pelo autor

3.4.1 Algoritmo SDA

O algoritmo implementado em C, ambientado para Microcontrolador, foi adaptado do mesmo implementado na simulação do Matlab[®] apresentado em 3.2.2. Foi necessário a adaptação, devido às particularidades da linguagem, como bibliotecas específicas para as funções utilizadas que diferem das linguagens.

Os algoritmos se encontram nos apêndices, onde o Apendice C, apresenta o algoritmo testado em linguagem C, no compilador DevC++. O Apendice D apresenta o algoritmo implementado no microcontrolador, que é utilizado no protótipo e testado em ambientes reais.

3.5 Métricas de Desempenho

Para avaliação de desempenho dos algoritmos utilizados na simulação, foram empregadas quatro métricas que são amplamente utilizadas na área de aprendizado de máquinas, sendo elas: o erro médio quadrático (EMQ), a curva ROC, a matriz de confusão e o índice Kappa.

Os testes realizados no protótipo foram realizados em um ambiente controlado, buscando evitar os ruídos de fundo que pudessem interferir no funcionamento dos sistemas devido à sensibilidade dos sensores. Foram utilizados vários objetos que geram sons como: tambor, voz humana, flauta, gaita, batida de palmas, barulho de impactos como uma bola batendo no chão, dentre outros. Após a coleta dos resultados, estes foram avaliados utilizando as métricas estatísticas.

4 RESULTADOS

Neste capítulo são apresentados os resultados obtidos das simulações dos métodos utilizando Rede Neural Artificial (RNA) e Somatório das Diferenças Absolutas (SDA) para detecção da direção da fonte sonora, como também os resultados obtidos utilizando o protótipo desenvolvido.

4.1 Resultados da RNA

A Rede Neural Artificial (RNA) foi implementada e treinada conforme descrito na seção 3.2.1. A Tabela 4.1 apresenta os resultados de teste obtidos para a topologia de rede *Feed-Forward Back-Propagation* treinada utilizando diferentes tipos de algoritmos de treinamento disponibilizados no Toolbox de Redes Neurais Artificiais do Matlab®.

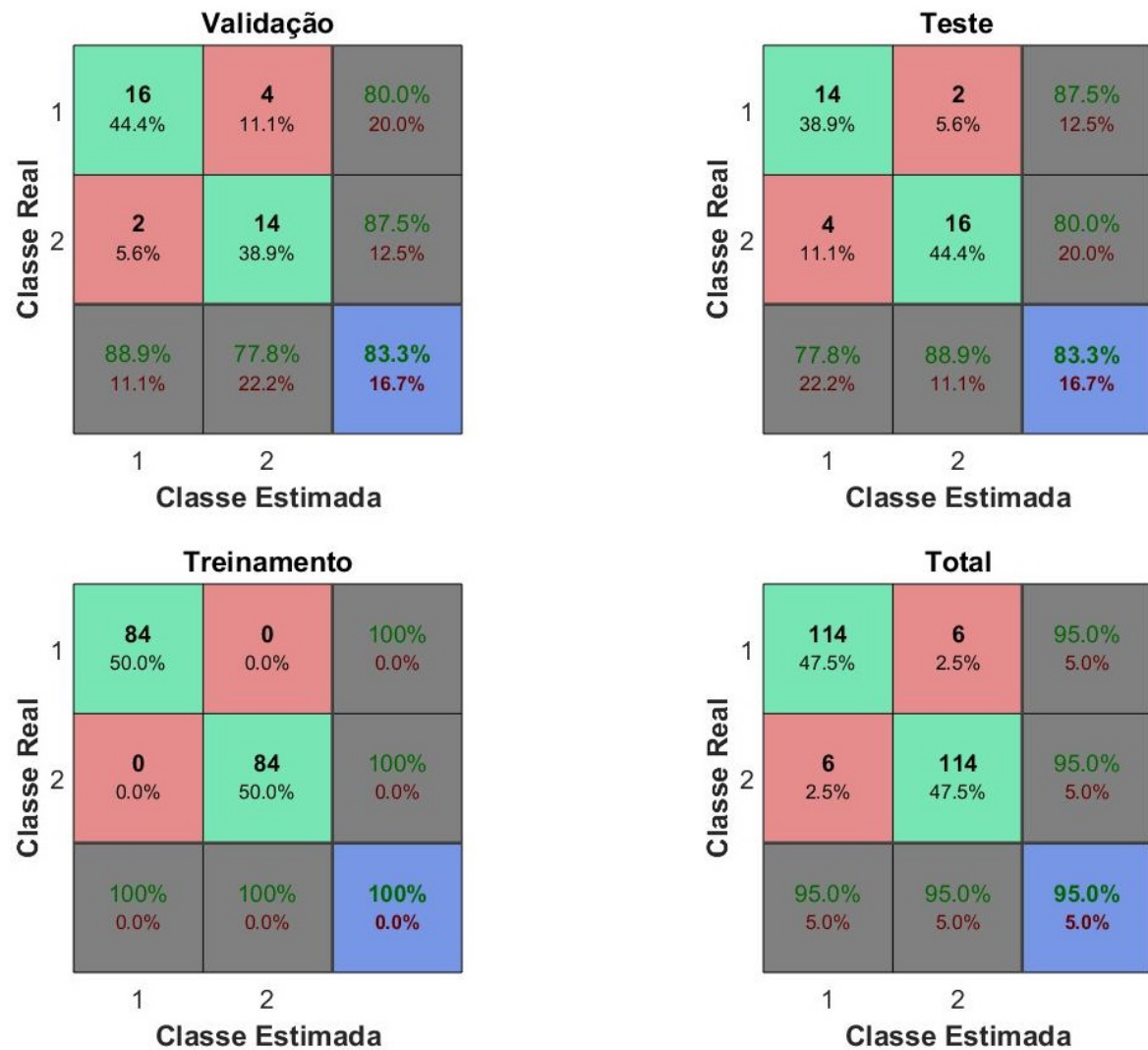
Tabela 4.1 – EMQ da rede *Feed-Forward Back-Propagation* usando dados de teste

Tipo de Rede	Algoritmo de Treinamento	EMQ		Iterações [n Épocas]	
		Média	Desvio Padrão	Média	Desvio Padrão
Feed-Foward BackPropagation	Train NOSS	0.1707	0.0503	30	26
	Train GDX	0.1597	0.0382	44	38
	Train GDA	0.5119	0.0421	88	70
	Train GDM	0.2752	0.0859	362	255

Analisando a Tabela 4.1, os melhores resultados com os dados de teste utilizando a topologia de rede *Feed-Forward Back-Propagation* foram obtidos com o algoritmo *GDX*. A média do erro médio quadrático (EMQ) obtida foi de 0.1597 com 44 épocas de treinamento.

As matrizes de confusão mostradas na Figura 4.1 apresentam os resultados das classificações de todos os dados gerados utilizando a RNA treinada com o algoritmo *GDX*. Analisando a matriz de confusão total, a RNA classificou 114 instâncias como “1” (direito) que realmente são “1” (direito), 114 instâncias como “2” (esquerdo) que são “2” (esquerdo), 6 instância como “2” (esquerdo) que na verdade é “1” (direito), e 6 instâncias como “1” (direito) que na verdade são “2” (esquerdo), e assim obtendo 95.0% de acertos.

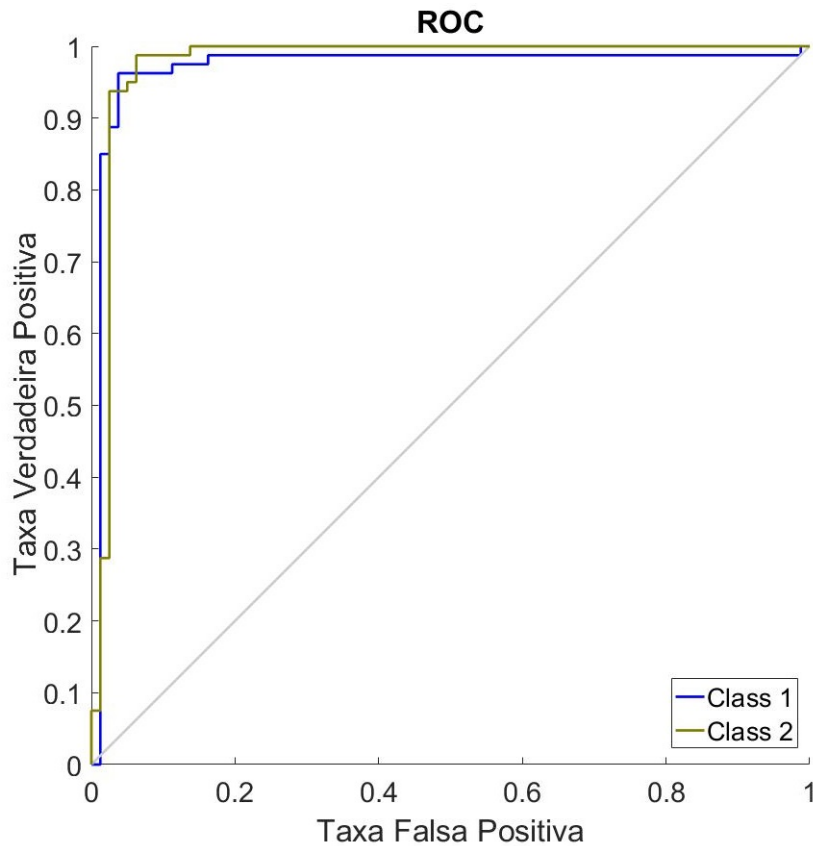
Figura 4.1 – Matriz de confusão das etapas do algoritmo GDX



Fonte: Elaborado pelo autor

Analisando a Curva ROC do resultado total obtido com a RNA treinada com o algoritmo GDX, mostrado na Figura 4.2, conclui-se que a RNA obteve um bom desempenho pois esteve próximo do ponto (0, 1), ou (0%, 100%) que representa um bom resultado.

Figura 4.2 – Curva ROC do Algoritmo GDX



Fonte: Elaborado pelo autor

Além da topologia de rede *Feed-Forward Back-Propagation* descrita anteriormente, foi também implementado a RNA com a topologia *Cascade-Forward Back-Propagation* disponível no Toolbox do Matlab e treinada com o mesmo conjunto de dados. A Tabela 4.2 mostra os resultados obtidos para esta topologia de rede utilizando diferentes tipos de algoritmos de treinamento.

Tabela 4.2 – EMQ para topologia de rede *Cascade-Forward Back-Propagation*

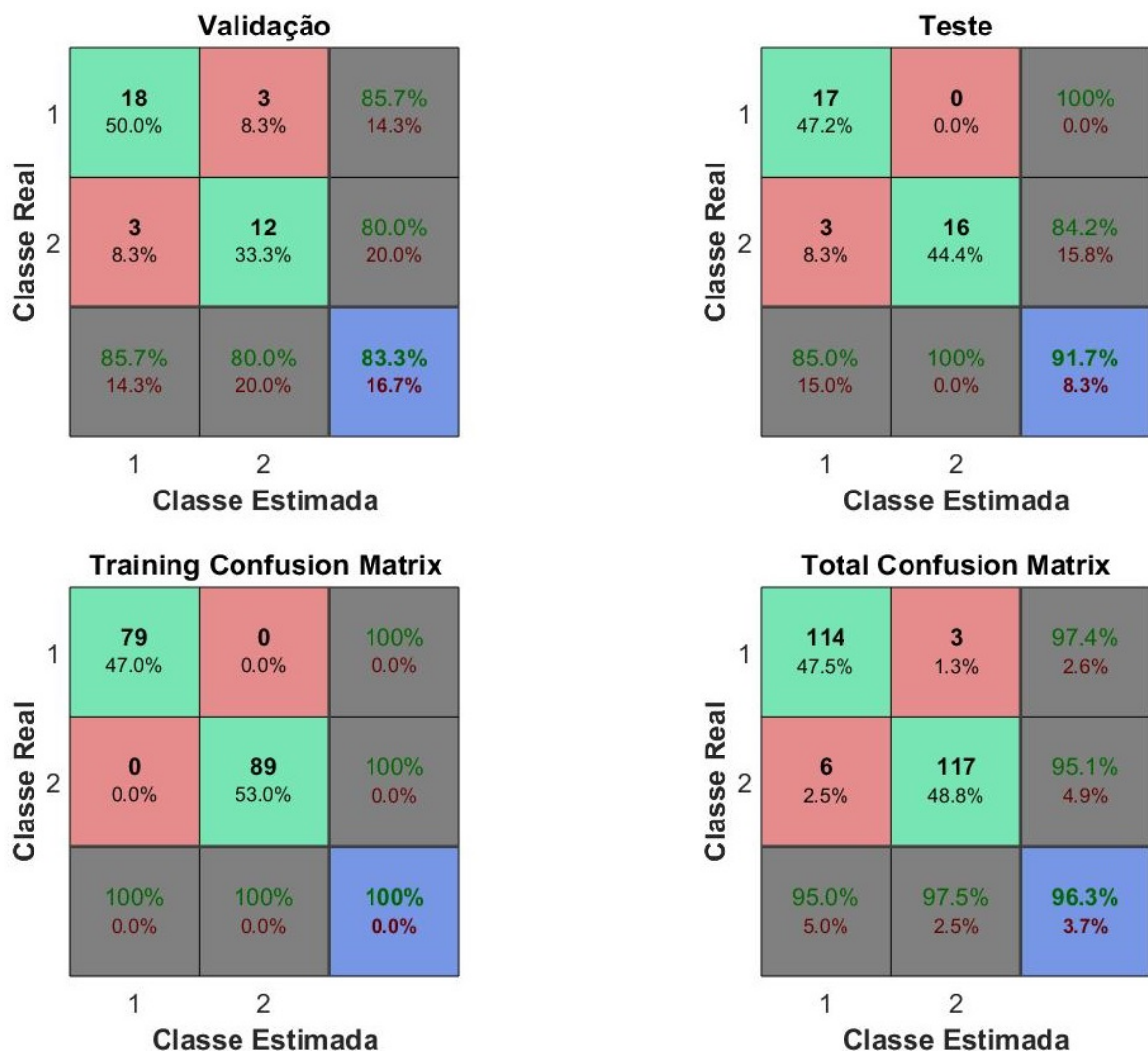
Tipo de Rede	Algoritmo de Treinamento	EMQ		Iterações [n Épocas]	
		Média	Desvio Padrão	Média	Desvio Padrão
Cascade-Foward BackPropagation	Train NOSS	0.2014	0.1589	71	37
	Train GDX	0.2159	0.1155	45	30
	Train GDA	0.4883	0.1631	422	161
	Train GDM	0.515	0.1526	345	261

Os melhores resultados obtidos com a topologia de rede *Cascade-Foward Back-Propagation*, foi o algoritmo de treinamento *One-Step-Secant (OSS)*. A média do erro médio quadrático (EMQ) obtida foi de 0.2014 com 71 épocas de treinamento.

A Figura 4.3 apresentam os resultados de classificações na forma de matriz de confusão obtida pela rede *Cascade-Foward Back-Propagation* treinada com o algoritmo *One-Step-Secant (OSS)*.

Analisando a matriz de confusão total, a RNA classificou 114 instâncias como “1” (direito) que realmente são “1” (direito), 117 instancias como “2” (esquerdo) que são “2” (esquerdo), 6 instância como “2” (esquerdo) que na verdade é “1” (direito), e 3 instâncias como “1”(direito) que na verdade são “2” (esquerdo), e assim obtendo 96.3% de acertos, e os erros foram 3.7%.

Figura 4.3 – Matriz de confusão das etapas do algoritmo OSS

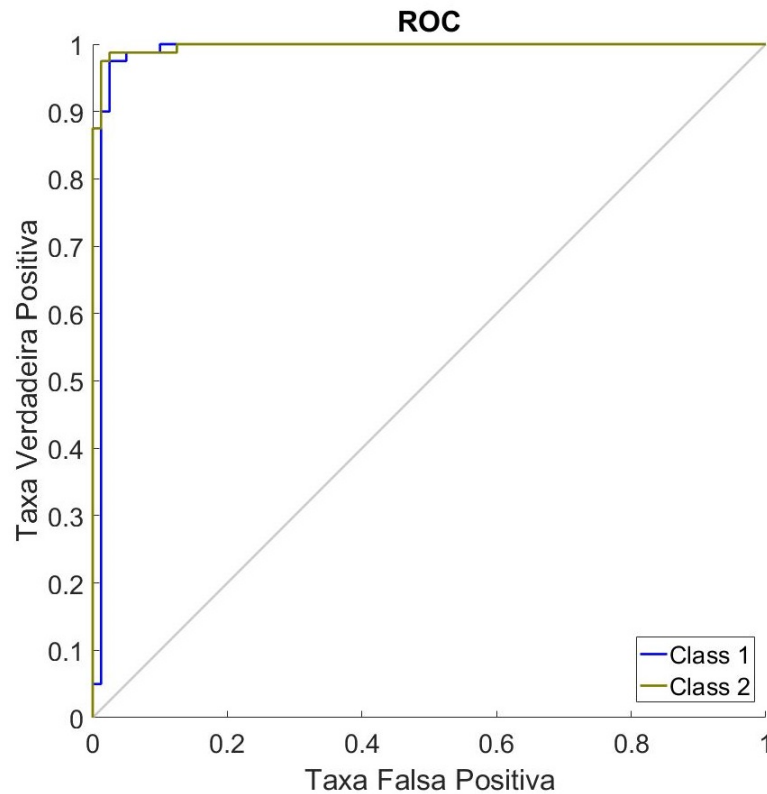


Fonte: Elaborado pelo autor

A Figura 4.4 mostra a curva ROC do total para o algoritmo de treinamento OSS, que obteve o melhor desempenho. Analisando a Curva ROC do algoritmo OSS, mostrado na Figura

4.4, o algoritmo obteve um desempenho ótimo, pois esteve próximo do ponto (0, 1), ou (0%, 100%) que representa o modelo perfeito, isto é, quase não obteve erros na classificação.

Figura 4.4 – Curva ROC do Algoritmo OSS



4.2 Resultados do SDA

Para analisar o funcionamento do algoritmo SDA na detecção da direção da fonte sonora, executou-se os testes com os mesmos 40 sons utilizados para treinar e avaliar a Rede Neural Artificial. A Figura 4.5 apresenta os resultados obtidos (identificação da direção do som: 1 - direito, 2 - esquerdo) utilizando o método SDA simulado no Matlab. Analisando a matriz de confusão, o método SDA simulado classificou 120 instâncias como “1” (direito) que realmente são “1” (direito), 120 instâncias como “2” (esquerdo) que são “2” (esquerdo), nenhuma instância como “2” (esquerdo) que na verdade é “1” (direito), e nenhuma instâncias como “1” (direito) que na verdade são “2” (esquerdo), e assim obtendo 100% de acertos.

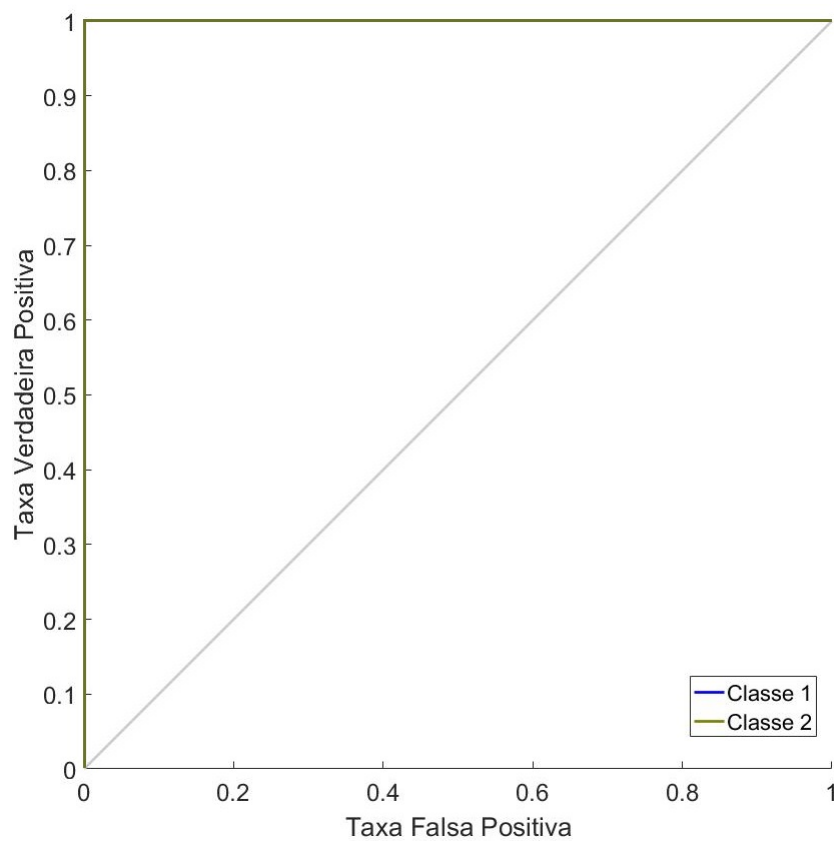
Figura 4.5 – Matriz de confusão do Algoritmo SDA

Classe Real	1	120 50.0%	0 0.0%	100% 0.0%
	2	0 0.0%	120 50.0%	100% 0.0%
		1	2	
		Classe Estimada		

Fonte: Elaborado pelo autor

A Figura 4.6 mostra a curva ROC do algoritmo SDA, que obteve um ótimo desempenho devido a taxa de 100% de acerto.

Figura 4.6 – Curva ROC do algoritmo SDA

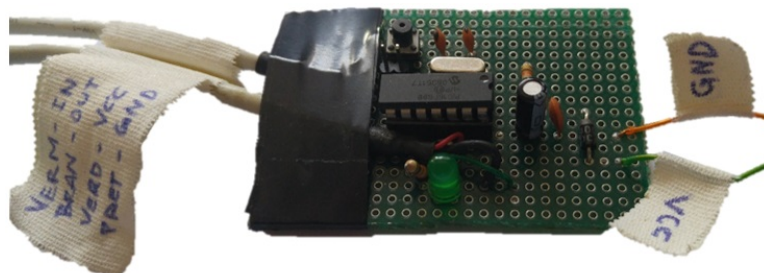


Fonte: Elaborado pelo autor

4.3 Implementação real no Microcontrolador

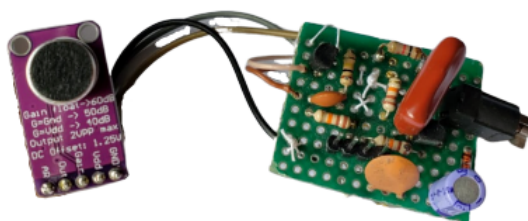
As Figuras 4.7 e 4.8 apresentam o resultados da montagem final de cada parte do protótipo. A Figura 4.7 é a placa central e a Figura 4.8 mostra a placa que será utilizada no ombro do usuário.

Figura 4.7 – Placa central contendo Microcontrolador



Fonte: Elaborado pelo autor

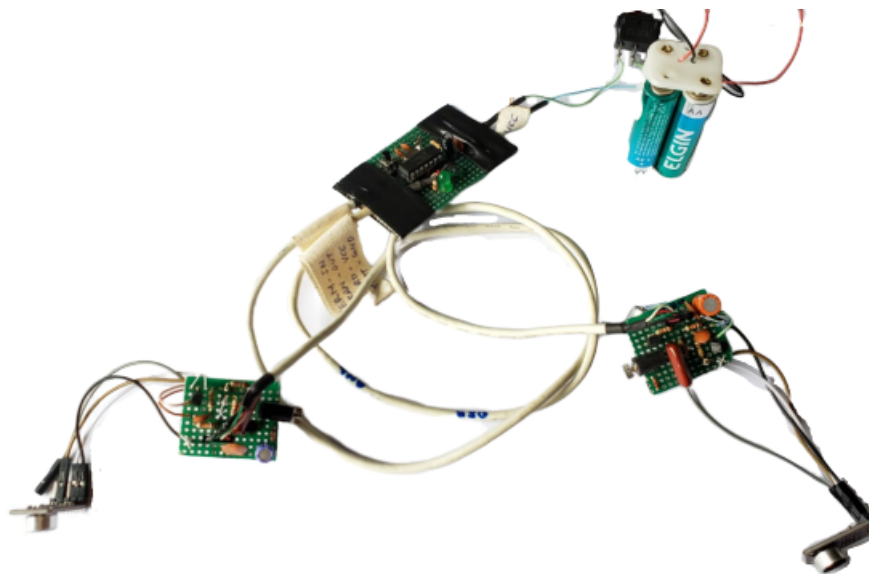
Figura 4.8 – Placa utilizada em cada ombro contendo microfone e motor vibracall



Fonte: Elaborado pelo autor

O protótipo completo pode ser visto na Figura 4.9, composto por todos os componentes apresentados. Na placa maior encontra-se o microcontrolador PIC16F688, o cristal oscilador, e demais componentes do circuito. Nas outras duas placas menores, encontram-se o microfone, motor vibracall, e os fios para a ligação entre as placas, junto com a fonte composta por 4 pilhas AA de 1,5V.

Figura 4.9 – Protótipo, composto pelo PIC16F688, Microfone, Motor vibracall, Bateria.

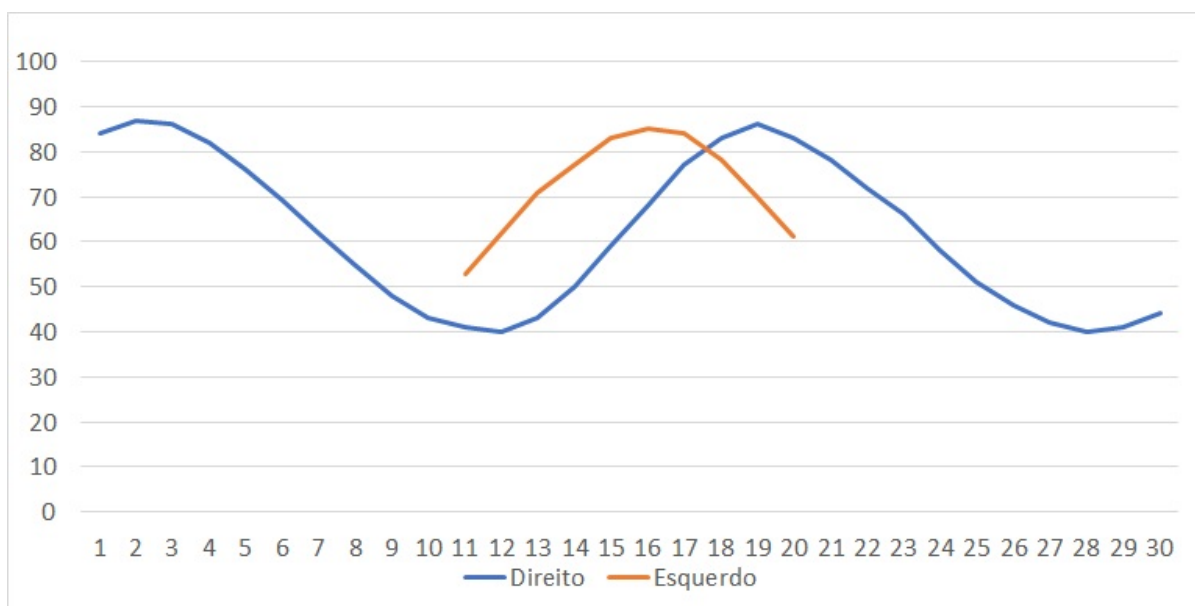


Fonte: Elaborado pelo Autor

Para analisar o funcionamento do sistema em conjunto com o software gravado no microcontrolador, realizou-se testes capturando os sinais através dos sensores (microfones) e transmitindo através da interface RS232 para o computador. Utilizando as portas RC4 e RC5 do microcontrolador com as funcionalidades RX e TX para comunicação serial, os dados foram recebidos e exibidos por meio gráfico utilizando o software Excel.

Assim, foi preparado um ambiente isolado sem som de fundo, aproximando de um local ideal sem ruídos externos. O sistema conseguiu capturar os sinais exibidos na Figura 4.10, com baixa taxa de ruído, quando foi emitido um som de um aparelho celular com frequência de 500Hz. Visualmente é possível identificar o defasamento, onde a linha laranja representando o sensor do lado esquerdo, que está adiantado em relação ao lado direito (linha azul), desse modo a fonte sonora está mais próxima do lado esquerdo.

Figura 4.10 – Sinal adquirido em um ambiente controlado



Fonte: Elaborado pelo Autor

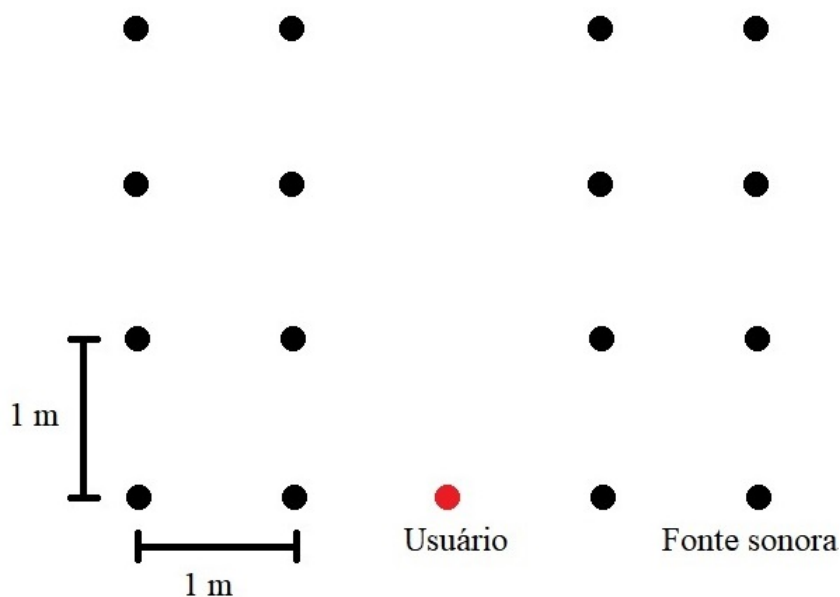
Com os sinais capturados, foram realizados ajustes, como o tempo de aquisição das amostras, o intervalo de captura de cada amostra, e o cálculo para o identificar o limiar da energia mínima para iniciar a aquisição dos dados.

4.4 Teste em ambiente real

Após a construção do protótipo, foram realizados os testes utilizando sons reais, no qual o usuário utiliza a vestimenta contendo o microcontrolador para testar o sistema em um ambiente real.

Para realizar o teste de desempenho, foram reproduzidos os sons a diferentes distâncias e posições do protótipo conforme Figura 4.11. Cada som foi gerado 10 vezes em cada posição, totalizando em 160 execuções. Os testes foram realizados em um ambiente controlado, com isolamento de ruídos e ecos.

Figura 4.11 – Posições utilizadas para o teste real



Fonte: Elaborado pelo Autor

A Figura 4.11, mostra o padrão utilizado para realizar os testes, onde o círculo preto representa as posições onde foi posicionado o objeto sonoro e assim reproduzindo o som do mesmo, e o círculo vermelho representa a posição fixa do usuário utilizando o protótipo.

A Tabela 4.3 apresenta a matriz de confusão dos resultados experimentais obtidos com o protótipo, quando foi utilizado um instrumento de som tipo tambor como fonte sonora. O índice de acertos encontrado foi de 70,6%. O índice Kappa correspondente foi de 0,41, indicando que o sistema possui uma concordância satisfatória.

Tabela 4.3 – Matriz de Confusão dos resultados obtidos com o som do Tambor

	Direito	Esquerdo	
Direito	55	22	
Esquerdo	25	58	
Acurácia	-	-	70,6%

Fonte: Elaborado pelo Autor

A Tabela 4.4 apresenta a matriz de confusão dos resultados experimentais obtidos com o protótipo, quando foi utilizado um instrumento de som tipo apito como fonte sonora. O índice de acertos encontrado foi de 48,8%. O índice Kappa calculado é -0,03, indicando que o sistema possui uma concordância péssima. Isto é justificado porque o som do apito possui uma

frequência bem acima da frequência de corte do filtro passa baixa do protótipo, que é utilizado devido a baixa taxa do conversor ADC do PIC.

Tabela 4.4 – Matriz de Confusão dos resultados obtidos com o som do Apito

	Direito	Esquerdo	
Direito	41	43	
Esquerdo	39	37	
Acurácia	-	-	48,8%

Fonte: Elaborado pelo Autor

A Tabela 4.5 apresenta a matriz de confusão dos resultados experimentais obtidos com o protótipo quando foi utilizado um celular para gerar um som por meio de aplicativo. Definiu-se a frequência de som gerada em 500Hz (exatamente a frequência de corte do filtro) com formato senoidal. O índice de acertos obtido foi de 85,5%. O índice Kappa calculado é 0,73, indicando que o sistema possui uma concordância boa.

Tabela 4.5 – Matriz de Confusão dos resultados obtidos com o som senoidal de 500Hz gerados por Celular

	Direito	Esquerdo	
Direito	68	9	
Esquerdo	12	71	
Acurácia	-	-	85,5%

Fonte: Elaborado pelo Autor

A Tabela 4.6 apresenta a matriz de confusão dos resultados experimentais obtidos com o protótipo, quando foi utilizado um instrumento de som tipo gaita como fonte sonora, onde utilizou apenas os sons mais graves, ou seja, sons de baixa frequências. O índice de acertos obtido foi de 70,6%. O índice Kappa calculado é 0,41, indicando que o sistema possui uma concordância satisfatória.

Tabela 4.6 – Matriz de Confusão dos resultados obtidos com o som da Gaita

	Direito	Esquerdo	
Direito	55	22	
Esquerdo	25	58	
Acurácia	-	-	70,6%

Fonte: Elaborado pelo Autor

A Tabela 4.7 apresenta a matriz de confusão dos resultados experimentais obtidos com o protótipo, quando foi utilizado um carro como fonte sonora, o carro em questão possui escapamento esportivo. Realizou-se os testes com o automóvel em um ambiente aberto, pois não foi possível colocar em um ambiente fechado a fim de evitar ruídos. O índice de acertos obtido foi de 77,5%. O índice Kappa calculado é 0,55, indicando que o sistema possui uma concordância boa.

Tabela 4.7 – Matriz de Confusão dos resultados obtidos com o som do Carro

	Direito	Esquerdo	
Direito	60	16	
Esquerdo	20	64	
Acurácia	-	-	77,5%

Fonte: Elaborado pelo Autor

A Tabela 4.8 apresenta a matriz de confusão dos resultados experimentais obtidos com o protótipo, quando foi utilizado um violão como fonte sonora, apenas o acorde Dó maior (D) foi utilizada. O índice de acertos calculado é 70%. O índice Kappa calculado é 0,4, indicando que o sistema possui uma concordância satisfatória.

Tabela 4.8 – Matriz de Confusão dos resultados obtidos com o som do violão acorde Dó

	Direito	Esquerdo	
Direito	53	21	
Esquerdo	27	59	
Acurácia	-	-	70%

Fonte: Elaborado pelo Autor

A Tabela 4.9 apresenta a matriz de confusão dos resultados experimentais obtidos com o protótipo, quando foi utilizado o efeito sonoro ao bater palmas. O índice de acertos obtido foi de 70,6%. O índice Kappa calculado é 0,41, indicando que o sistema possui uma concordância satisfatória.

Tabela 4.9 – Matriz de Confusão dos resultados obtidos com o som de palmas

	Direito	Esquerdo	
Direito	56	23	
Esquerdo	24	57	
Acurácia	-	-	70,6%

Fonte: Elaborado pelo Autor

Os resultados obtidos pelo protótipo para alguns objetos utilizados como fontes sonoras tiveram baixa acurácia como os sons gerados pelo apito e pela gaita (sons agudos) devido a alta frequência do sinal ultrapassando o limite de 500Hz imposto pelo filtro passa-baixa. Os demais sons utilizados possuem uma frequência menor e assim obteve-se uma melhor acurácia.

Os resultados obtidos pela simulação dos métodos escolhidos (RNA e SDA) foram bem melhores que os resultados obtidos pelo protótipo, indicando que o sistema real possui características que não foram levadas em conta na simulação, privilegiando a simulação.

4.5 Custos

A Tabela 4.10, apresenta o custo total gasto na construção do protótipo, bem como os custos de cada componente que foram utilizados.

Tabela 4.10 – Custo total e dos componentes utilizados no protótipo

Componente	Quantidade	Valor Unitário	Valor Total
Microcontrolador 16F688	1	R\$ 26,00	R\$ 26,00
Módulo Amplificador MAX9814	2	R\$ 17,00	R\$ 34,00
Motor VibraCall	2	R\$ 10,00	R\$ 20,00
Componentes eletrônicos (Cristal, capacitor, resistor, pushbotton, fios)	-	-	R\$ 28,00
LED	3	R\$ 1,00	R\$ 3,00
Pack de 4 Pilhas AA	1	R\$ 9,90	R\$ 9,90
Total			R\$ 120,90

Os aparelhos auditivos comerciais possuem valores em torno de R\$500,00 e podem ultrapassar até R\$ 4 mil a unidade. O orçamento é realizado de acordo com o aparelho que se adapta melhor ao usuário, levando em consideração o grau da perda auditiva do usuário e também considerando o tipo de deficiência.

Além dos aparelhos auditivos, de acordo com o grau da perda auditiva e o problema apresentado no ouvido do deficiente auditivo em que a perda auditiva é muito severa, há as cirurgias de implantes cocleares (também chamados de Ouvidos Biônicos). No Brasil, o de-

ficiente auditivo pode conseguir realizar pelo SUS (Sistema Único de Saúde), mas o custo do implante em clínicas particulares chega a R\$ 40 mil.

Assim, como mostrado na Tabela 4.10, o objetivo em desenvolver o protótipo de baixo custo, foi satisfatório, uma vez que o custo total foi de apenas R\$ 120,90, tornando mais acessível para a população em geral.

5 CONCLUSÃO

A presente dissertação, teve como principal objetivo a criação de um protótipo de baixo custo para detecção da direção sonora. Utilizou-se o microcontrolador PIC16F688, levando em consideração o baixo custo e a facilidade de manuseio e programação.

A implementação de métodos de processamento de sinais digitais permitiram o cálculo da defasagem entre dois sinais e/ou determinar a direção de chegada de um sinal. Assim, foi possível a criação do protótipo contendo o hardware e o software que fosse eficiente na detecção das direções de fontes sonoras de frequência até 500Hz devido a limitação do microcontrolador PIC16F688.

Algumas topologias de redes neurais artificiais foram implementadas para detecção da direção da fonte sonora e comparadas com o objetivo de encontrar a topologia mais adequada. O algoritmo de treinamento que obteve um melhor desempenho foi *One Step Secant* com a topologia *Cascade-Forward Back-Propagation*. A matriz de confusão obtida para este algoritmo mostra uma taxa de acerto de 96,3%, enquanto para os demais algoritmos como *Gradient Descent with Momentum & Adaptive LR (GDX)* apresentou uma acurácia de 95,6%.

A simulação do método de cálculo do defasamento entre dois sinais utilizando SDA obteve 100% de taxa de acerto. Por isto, esta técnica foi utilizada no protótipo para a detecção da direção da fonte sonora, além de apresentar maior facilidade de implementação que uma RNA.

O teste do protótipo em ambiente real controlado utilizando a técnica SDA para determinação da direção do som obteve uma taxa de acerto de até 85,5% dependendo da frequência do sinal da fonte sonora. Assim, o protótipo mostrou-se adequado para o objetivo proposto, sendo possível melhorar a faixa de frequência de entrada por meio de melhorias.

A acessibilidade para os deficientes auditivos é essencial e só será possível quando compreenderem suas peculiaridades e respeitarem sua deficiência. Como este trabalho tem por finalidade amenizar as barreiras existentes à pessoa surda e promover o bem-estar e facilidade de locomoção no dia a dia, espera-se que as instituições em geral, sejam elas públicas, privadas, filantrópicas, compreendam o surdo enquanto sujeito de direitos perfeitamente “normal”, que precisa ser atendido em seus direitos plenamente.

Levando em consideração os desafios da inclusão e das acessibilidade, buscou-se com a proposta por uma sociedade mais igualitária e humana. Essa perspectiva leva em conta os espaços, os ambientes e os recursos que devem ser acessíveis para todos os cidadãos, propor-

cionando oportunidade iguais para todos. Assim, o desenvolvimento deste trabalho preza pela acessibilidade, pelo seu caráter social e inclusivo, pois com ele é possível auxiliar o deficiente auditivo na locomoção no ambiente urbano evitando acidentes, além de proporcionar conforto e sensação de pertencimento dos espaços nos quais todos circulam.

5.1 Propostas de continuidade

Para trabalhos futuros, sugere-se implementar outros métodos como a Rede Neural Artificial no microcontrolador. Deste modo, é necessária a utilização de um microcontrolador com maior capacidade de processamento e armazenamento. Também podem ser utilizados outros dispositivos para sistemas embarcados contendo maior capacidade de armazenamento e processamento como Arduino Mega, Raspberry Pi 3, Rock Pi 4B, Banana Pi M2 Zero, Nvidia Jetson Nano, Asus Tinker Board, Gigabyte GA-SBCAP3350, microcontroladores PIC da família 18F, microcontroladores da Texas Instruments como MSP430 LaunchPad, dentre outros.

Sugere-se também a implementação de outros métodos de detecção de defasagem entre sinais como Sistemas Fuzzy, outras topologias de Redes Neurais Artificiais e outros métodos de treinamento.

Sugere-se também a aplicação do sistema desenvolvido em outras áreas como: câmera inteligente que segue um objeto sonoro; aplicações bélicas para detectar a direção de um disparo efetuado por uma arma, explosões, até mesmo mísseis; sensores sonoros para equipamentos de segurança para carro ou residências.

REFERÊNCIAS

- ALMEIDA, J. F. S. de. **Algoritmos de Estimação da Direção de Chegada de Sinal**. Dissertação (Mestrado) — Instituto Superior de Engenharia de Lisboa, Nov 2015. Engenharia de Eletrônica e Telecomunicações e de Computadores.
- ALVES, P. **iPhone ‘explodido’ por estudante revela funcionamento do vibracall**. 2013. TechTudo, Informática. Disponível em: <<https://www.techtudo.com.br/noticias/noticia/2013/11/iphone-explodido-por-estudante-revela-funcionamento-do-vibracall.html>>. Acesso em: 30/09/2018.
- ANDRADE, B. C. de. **Determinação da defasagem entre sinais utilizando métodos digitais de processamento**. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro / Escola Politécnica, 2017. Curso de Engenharia Eletrônica e de Computação.
- ANTUNES, D. R.; GUIMARÃES, C.; GARCÍA, L. S.; OLIVEIRA, L. E. S.; FERNANDES, S. A framework to support development of sign language human-computer interaction: Building tools for effective information access and inclusion of the deaf. In: **Fifth International Conference On Research Challenges In Information Science**. Gosier, France: [s.n.], 2011. p. 1–12. ISSN 2151-1349.
- ANTUNES, F.; LEÃO, R.; SAMPAIO, R. **Harmônicos em Sistemas Elétricos**. Elsevier Editora Ltda., 2014. ISBN 9788535274400. Disponível em: <<https://books.google.com.br/books?id=NtNeAwAAQBAJ>>. Acesso em: 05/05/2019.
- ARMANSYAH, R. F.; IKRAM, F. D.; NOLIKA, S. S.; ADIONO, T. Efficient sound-source localization system using low cost tdoa computation. In: **2016 International Symposium on Electronics and Smart Devices (ISESD)**. Bandung, Indonesia: [s.n.], 2016. p. 315–319.
- AUBYN, A. S.; FIGUEIREDO, M. C.; LOURA, L. de; RIBEIRO, L.; VIEGAS, F. **Números Reais**. Dissertação (Mestrado) — Grupo de Matemática da Universidade Técnica de Lisboa, mai 2006. Disponível em: <<http://preprint.math.ist.utl.pt/files/ppgmutlreais.pdf>>. Acesso em: 16/09/2019.
- BACHA, C. **Macroeconomia aplicada à análise da economia brasileira**. Edusp, 2004. (Acadêmica (São Paulo, Brazil)). ISBN 9788531408311. Disponível em: <<https://books.google.com.br/books?id=iI4w7qm-HjsC>>. Acesso em: 30/11/2019.
- BAUER, W.; WESTFALL, G.; DIAS, H. **Física para Universitários - Relatividade, Oscilações, Ondas e Calor**. McGraw Hill Brasil, 2013. ISBN 9788580551600. Disponível em: <<https://books.google.com.br/books?id=796EMutjMtYC>>. Acesso em: 19/08/2019.
- BEALE, M. H.; HAGAN, M. T.; DEMUTH, H. B. **Neural Network Toolbox™ User's Guide**. 1st. ed. 3 Apple Hill Drive Natick, MA 01760-2098: MathWorks, 2018. 558 p.
- BOHN, D. A. Environmental effects on the speed of sound. **Journal of the Audio Engineering Society**, v. 36, n. 4, p. 7–11, April 1988.
- BOROWIK, B.; BOROWIK, B.; KARPINSKYI, V.; KOCHAN, R.; WIECLAW, L. Microcontroller pic based traffic light system with collision detection. In: **9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)**. Bucharest, Romania: [s.n.], 2017. v. 1, p. 118–123.

BOYCE, W. E.; DIPRIMA, R. C. **Equações Diferenciais Elementares e Problemas de Valores de Contorno**. 10. ed. Brasil: LTC, 2015. 680 p. ISBN 9788521627357.

BRASIL. **Apesar de avanços, surdos ainda enfrentam barreiras de acessibilidade**. 2016. Portal Brasil. Disponível em: <<http://www.brasil.gov.br/cidadania-e-justica/2016/09/apesar-de-avancos-surdos-ainda-enfrentam-barreiras-de-acessibilidade>>. Acesso em: 11/01/2018.

CANCELLIER, L.; SEIDEL, I.; GUNTZEL, J. L. A. Algoritmo de eliminações sucessivas em níveis baseado na soma das diferenças transformadas absolutas. In: **Anais do XXXVI Concurso de Trabalhos de Iniciação Científica da SBC**. Porto Alegre, RS, Brasil: SBC, 2017. Disponível em: <<https://sol.sbc.org.br/index.php/ctic/article/view/3242>>.

CARVALHO, A. P. L. F. **Redes Neurais Artificiais**. 2009. Tutorial Introductório sobre Redes Neurais Artificiais em especial sobre as Redes Multi Layer Perceptron treinadas com BackPropagation. Disponível em: <<http://conteudo.icmc.usp.br/pessoas/andre/research/neural/>>. Acesso em: 11/01/2018.

COHEN, J. A Coefficient of Agreement for Nominal Scales. **Educational and Psychological Measurement**, v. 20, n. 1, p. 37, 1960.

DINIZ, P.; SILVA, E. da; NETTO, S. **Processamento Digital de Sinais: Projeto e Análise de Sistemas**. 2. ed. Bookman Editora, 2014. ISBN 9788582601242. Disponível em: <<https://books.google.com.br/books?id=HoWaAgAAQBAJ>>. Acesso em: 22/10/2019.

DUGAN, R. C.; MCGRANAGHAN, M. F.; SANTOSO, S.; BEATY, H. W. **Electrical Power Systems Quality - 3.ed.** [S.l.]: McGraw-Hill Education, 2012. ISBN 9780071761550.

ELETRONICA, N. **Microfone de eletreto**. 2018. Microfone de eletreto. Disponível em: <<http://blog.novaeletronica.com.br/microfone-de-eletreto/>>. Acesso em: 30/09/2018.

ELMIKATY, M.; STATHAKI, T. Detection of cars in complex urban areas. In: **Fifteenth IAPR International Conference on Machine Vision Applications (MVA)**. [S.l.: s.n.], 2017. p. 105–108.

ELMIKATY, M.; STATHAKI, T. Detection of cars in high-resolution aerial images of complex urban environments. **IEEE Transactions on Geoscience and Remote Sensing**, v. 55, n. 10, p. 5913–5924, Oct 2017. ISSN 0196-2892.

EXAME. **OMS afirma que 60% dos casos de surdez podem ser prevenidos**. 2016. Dos % dos casos evitáveis, 31% se devem a doenças como sarampo, caxumba, rubéola ou meningite. Disponível em: <<https://exame.abril.com.br/mundo/oms-afirma-que-60-dos-casos-de-surdez-podem-ser-prevenidos/>>. Acesso em: 11/01/2018.

FONSECA, R.; SILVA, P.; SILVA, R. Acordo inter-juízes: O caso do coeficiente kappa. **Instituto Superior de Psicologia Aplicada**, v. 5, n. 3, p. 81–90, 2007.

FRANÇA, J. A. de. **Estereoscópica para Aplicações em Robótica Móvel**. Dissertação (Mestrado) — Uiniversidade Federal de Santa Catarina, 2003. Disponível em: <<http://www.uel.br/pessoal/josealexandre/doutorado/franca03desenvolvimento.pdf>>. Acesso em: 01/09/2019.

- FRANCIS, L. T. **Análise e Desenvolvimento de um Sistema de Controle para Integração Eficiente da Iluminação Natural e Artificial**. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, jul 2019.
- FREITAS, M. A. B. **Medindo a Velocidade do Som com o Microfone do PC**. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, jul 2005.
- HERSH, M. Accessibility and usability of virtual learning environments. In: **2008 Eighth IEEE International Conference on Advanced Learning Technologies**. [S.l.: s.n.], 2008. p. 991–992. ISSN 2161-3761.
- HOESEL, R.; RAMSDEN, R.; ODRISCOLL, M. Sound-direction identification, interaural time delay discrimination, and speech intelligibility advantages in noise for a bilateral cochlear implant user. **Ear and hearing**, v. 23, p. 137–49, 05 2002.
- IBM. **Visualização da Matriz de Confusão**. 2018. IBM Knowledge Center - o local da documentação de produto IBM. Disponível em: <<https://www.ibm.com/support/knowledgecenter/pt-br/>>. Acesso em: 22/03/2018.
- JACOBI, R. P.; CARDOSO, R. B.; BORGES, G. A. Voc: a reconfigurable matrix for stereo vision processing. In: **Proceedings 20th IEEE International Parallel Distributed Processing Symposium**. [S.l.: s.n.], 2006. p. 6 pp.–. ISSN 1530-2075.
- KURNIANGGORO, L.; WAHYONO; HERNÁNDEZ, D. C.; JO, K. H. Camera and laser range finder fusion for real-time car detection. In: **IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society**. [S.l.: s.n.], 2014. p. 3419–3424. ISSN 1553-572X.
- LAI, Y. K.; HUANG, Y. H.; HWANG, C. M. Front moving object detection for car collision avoidance applications. In: **IEEE International Conference on Consumer Electronics (ICCE)**. [S.l.: s.n.], 2016. p. 367–368.
- LATHI, B. **Sinais e Sistemas Lineares - 2.ed.** Bookman, 2006. ISBN 9788577803910. Disponível em: <<https://books.google.com.br/books?id=S806DQAAQBAJ>>. Acesso em: 02/10/2019.
- LUITEL, B.; MURTHY, Y. V. S.; KOOLAGUDI, S. G. Sound event detection in urban soundscape using two-level classification. In: **IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)**. [S.l.: s.n.], 2016. p. 259–263.
- LUZ, G. B. B. **Determinação da Defasagem entre Sinais: Comparação e Análise de Métodos Práticos Variados**. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro / Escola Politécnica, 2016. Curso de Engenharia Eletrônica e de Computação.
- MARCELO, C. B. **Sons e Formas - As barreiras Acústicas na Atenuação do Ruído na Cidade**. Dissertação (Mestrado) — Universidade Presbiteriana Mackenzie, nov 2006.
- MCCULLOCH, W. S.; PITTS, W. **A logical calculus of the ideas immanent in nervous activity**. 1943. 115–133 p. Disponível em: <<https://doi.org/10.1007/BF02478259>>. Acesso em: 10/09/2019.
- MENEZES, F. **A Acústica Musical em Palavras e Sons**. Ateliê, 2004. ISBN 9788574802169. Disponível em: <<https://books.google.com.br/books?id=OX7aAC1RgcAC>>.

- MICROCHIP. 14-Pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology.** 2009. PIC16F688 Data Sheet. Disponível em: <<http://www.microchip.com/doclisting/TechDoc.aspx?type=datasheet&productfamily=8-BIT>>. Acesso em: 16/01/2018.
- MIKHAIL, E. M.; ACKERMANN, F. E. **Observations and least squares.** [S.l.]: New York, 1976. 497 p. ISBN 9780700224814.
- MIKROELEKTRONIKA. **MikroC.** 2018. MikroC PRO for PIC. Disponível em: <<https://www.mikroe.com/mikroc-pic>>. Acesso em: 22/01/2018.
- MONICO, J. F. G.; PÓZ, A. P. D.; GALO, M.; SANTOS, M. C. D.; OLIVEIRA, L. C. D. Acurácia e precisão: Revendo os conceitos de forma acurada. **Boletim de Ciências Geodésicas**, v. 15, n. 3, p. 469–483, jul-set 2009. ISSN 1982-2170.
- NASCIMENTO, C. S. do; GOBARA, S. T. A contextualizaÇÃo do ensino de ondas sonoras por meio do corpo humano. **VIIEnpec, Encontro Nacional de Pesquisa em Educação em Ciências**, nov. 2009. ISSN 21766940.
- NEUMAN, A. C.; HARAVON, A.; SISLIAN, N.; WALTZMAN, S. B. Sound-direction identification with bilateral cochlear implants. **Ear and hearing**, v. 28, p. 73–82, 02 2007.
- NISHIMURA, C. M. O. **Análise Comparativa de Algoritmos de Correlação Local Baseados em Intensidade Luminosa.** Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, 2008. Disponível em: <https://www.teses.usp.br/teses/disponiveis/3/3152/tde-14082008-082214/publico/Claudio_Nishimura_Dissertacao_ed_Revisada.pdf>. Acesso em: 11/07/2019.
- NOERGAARD, T. **Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers.** 2nd. ed. England: Newnes, 2012. 672 p. ISBN 9780123821966.
- OKUYAMA, F.; TAKAYAMA, J. .; OHYAMA, S.; KOBAYASHI, A. A study on determination of a sound wave propagation direction for tracing a sound source. In: **Proceedings of the 41st SICE Annual Conference. SICE 2002.** [S.l.: s.n.], 2002. v. 2, p. 1102–1104 vol.2.
- OLIVEIRA, A. C. de; ROCHA, L. P.; LIMA, V. P.; NETTO, L. M.; MENEZES, P. de L.; COLAFÊMINA, J. F. Localização de fontes sonoras: a importância das diferenças dos limiares auditivos interaurais. **Sociedade Brasileira de Fonoaudiologia**, n. 13, p. 7–11, fev 2008.
- OLIVEIRA, J. R. de. A aplicação de microfones em projeto digitais. **DCA-FEEC-Unicamp**, p. 7, Abril 2010.
- PAGLIUCA, L. M. F.; FIÚZA, N. L. G.; REBOUÇAS, C. B. de A. Aspectos da comunicação da enfermeira com o deficiente auditivo. **Revista da Escola de Enfermagem da USP**, v. 41, n. 3, p. 411–418, set. 2007. Disponível em: <<https://www.revistas.usp.br/reusp/article/view/41636>>.
- PLESHA, M.; GRAY, G.; COSTANZO, F. **Mecânica para Engenharia: Estática.** AMGH Editora, 2014. ISBN 9788565837309. Disponível em: <<https://books.google.com.br/books?id=Gc5IAgAAQBAJ>>. Acesso em: 22/11/2019.
- PONDÉ, L. F. **42 milhões de pessoas no mundo têm deficiência auditiva de moderada a profunda.** 2009. Íntegras. Disponível em: <<http://integras.blogspot.com.br/2009/02/42-milhoes-de-pessoas-no-mundo-tem.html>>. Acesso em: 11/01/2018.

PORSINAL. **Segundo a OMS 360 milhões de pessoas no mundo sofrem de perda auditiva incapacitante**. 2013. Porsinal. Disponível em: <<http://www.porsinal.pt/index.php?ps=destaques&idt=not&iddest=154>>. Acesso em: 11/01/2018.

PRATI, R. C.; BATISTA, G. E. A. P. A.; MONARD, M. C. Curvas roc para avaliação de classificadores. In: . [S.l.: s.n.], 2008. v. 6, n. 2, p. 215–222.

REDONDO, M. C. da F.; CARVALHO, J. M. **Deficiência Auditiva**. 1. ed. Brasília: MEC, 2000. 64 p. Secretaria de Educação a Distância. ISSN 1518-4706.

RETROGAMES. **Motor de Vibração para Controle Ps1 Ps2 Ps3 Xbox 360**. 2018. Mercado Livre, Motores. Disponível em: <https://produto.mercadolivre.com.br/MLB-853405321-motor-de-vibraco-para-controle-ps1-ps2-ps3-xbox-360-promo-_JM>. Acesso em: 30/09/2018.

RIVAS, J.; WUNDERLICH, R.; HEINEN, S. J. Road vibrations as a source to detect the presence and speed of vehicles. **IEEE Sensors Journal**, v. 17, n. 2, p. 377–385, Jan 2017. ISSN 1530-437X.

ROBERTS, M. **Fundamentos de Sinais e Sistemas**. McGraw Hill Brasil, 2009. ISBN 9788563308573. Disponível em: <<https://books.google.com.br/books?id=hjK5egvmsdsC>>. Acesso em: 20/06/2019.

RODRIGUES, P. C.; OLIVEIRA, A. C. de; COLAFÊMINA, J. F.; MENEZES, P. de L. Localização sonora em usuários de aparelhos de amplificação sonora individual. **CEFAC**, dez. 2009.

ROGAWSKI, J.; ADAMS, C. **Cálculo: Volume 2**. Bookman Editora, 2018. ISBN 9788582604588. Disponível em: <<https://books.google.com.br/books?id=19pPDwAAQBAJ>>. Acesso em: 30/05/2019.

RUMSEY, D.; FRANZIN, L. **Estatística II Para Leigos**. Alta Books, 2018. (Para Leigos). ISBN 9788550805702. Disponível em: <<https://books.google.com.br/books?id=N3lvDwAAQBAJ>>. Acesso em: 30/03/2019.

SADIKU, C. M. N. O.; MUSA, S. **Análise de Circuitos Elétricos com Aplicações**. AMGH Editora, 2014. ISBN 9788580553031. Disponível em: <<https://books.google.com.br/books?id=d2Q3AgAAQBAJ>>. Acesso em: 30/11/2019.

SANTOS, G. R. **A propagação das ondas sonoras nos diferentes tipos de instrumentos, física, interpretação e emoção**. 2016. Música, ondas sonoras e instrumentos musicais. Disponível em: <<https://theinstrumentsshow.wordpress.com/2016/05/24/musica-ondas-sonoras-e-instrumentos-musicais/>>. Acesso em: 30/11/2018.

SANTOS, L. de S. Deficiência auditiva. **Revista Científica Eletrônica de Ciências Sociais Aplicadas da Eduvale**, n. 07, p. 7, Novembro 2012. ISSN 1806-6283.

SANTOS, L. R. dos. **Algoritmo da Conformação de Feixes Aplicado a Sonares para Detecção de Fontes Sonoras em Ambiente Submarino**. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, mar 2019.

SANTOS, M. A. M.; PIEDADE, M. B.; TÁVORA, V. N. e. Hospital center website: Usability, accessibility and mobile optimization. In: **24º Encontro Português de Computação Gráfica e Interação (EPCGI)**. [S.l.: s.n.], 2017. p. 1–8.

SCALCO, H. N. **Sistema de Detecção de Intrusão em Redes de Computadores com Técnicas de Inteligência Computacional**. Dissertação (Mestrado) — Universidade Federal de Lavras, nov 2016.

SIEMENS. **MOTORES DE CORRENTE CONTÍNUA**. 2018. Motores de Corrente Contínua: Guia rápido para uma especificação precisa. Disponível em: <http://www.marioloureiro.net/tecnica/electrif/Motores_CC_ind1.pdf>. Acesso em: 30/09/2018.

SILVA, B. F. da. **Determinação da Direção de Chegada (DOA) de um Sinal Sonoro usando o microcontrolador Arduino**. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro / Escola Politécnica, Agosto 2014. Curso de Engenharia Eletrônica e de Computação.

SILVA, D. C. M. da. **Microfone**. 2018. Brasil Escola, Microfone. Disponível em: <<https://brasilecola.uol.com.br/fisica/microfone.htm>>. Acesso em: 30/09/2018.

SILVA, P. **Acústica Arquitetônica Condicionamento de Ar**. 4. ed. Belo Horizonte: EDTAL EMPRESA TERMO ACUSTICA LTDA, 2005. 368 p. ISBN 9788010048601.

SOUZA, F. **Microcontroladores Baseline PIC**. 2014. Microchip - Microcontroladores PIC de 8 bits. Disponível em: <<https://www.embarcados.com.br/pic/>>. Acesso em: 20/01/2018.

STEWART, J. **Calculus: Concepts and Contexts**. [S.l.]: Cengage Learning, 2009. ISBN 9780495557425.

STEWART, J. **Calculus: Concepts and Contexts**. Cengage Learning, 2010. (Available 2010 Titles Enhanced Web Assign Series). ISBN 9780495557425. Disponível em: <https://books.google.com.br/books?id=Vou3MZu_7tcC>. Acesso em: 11/10/2019.

TRUDEAU, L.; COULOMBE, S.; DESROSIERS, C. An adaptive search ordering for rate-constrained successive elimination algorithms. In: **2015 IEEE International Conference on Image Processing (ICIP)**. [S.l.: s.n.], 2015. p. 207–211. ISSN null.

UMEZAWA, Y.; EBIHARA, T.; MIZUTANI, K.; WAKATSUKI, N. Detection of individual vehicles approaching for pedestrians using digital modulated warning sounds. In: **IEEE 6th Global Conference on Consumer Electronics (GCCE)**. [S.l.: s.n.], 2017. p. 1–5.

VESPERINI, F.; VECCHIOTTI, P.; PRINCIPI, E.; SQUARTINI, S.; PIAZZA, F. A neural network based algorithm for speaker localization in a multi-room environment. In: **IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)**. [S.l.: s.n.], 2016. p. 1–6.

VILLANUEVA, J. M. M. **Amostragem dos Sinais**. 2015. Amostragem dos Sinais. Disponível em: <<http://www.cear.ufpb.br/juan/wp-content/uploads/2016/08/Aula-5-Teorema-de-Amostragem.pdf>>. Acesso em: 10/12/2019.

WEISSTEIN, E. W. **Absolute Difference**. 2019. From MathWorld—A Wolfram Web Resource. Disponível em: <<http://mathworld.wolfram.com/AbsoluteDifference.html>>. Acesso em: 17/09/2019.

WILMSHURST, T. **Designing Embedded Systems with PIC Microcontrollers: Principles and Applications**. 2nd. ed. England: Newnes, 2009. 704 p. ISBN 9780080961842.

YOON, J.; KIM, I.; CHUNG, W.; KIM, D. Fast and accurate car detection in drone-view. In: **IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)**. [S.l.: s.n.], 2016. p. 1–3.

YU, X.; ZHOU, R.; ZHANG, H.; XIN, H. A microwave direction of arrival estimation technique using a single antenna. **IEEE Transactions on Antennas and Propagation**, v. 64, n. 7, p. 3189–3195, July 2016. ISSN 0018-926X.

ZHU, C.; QI, W.-S.; SER, W. Predictive fine granularity successive elimination for fast optimal block-matching motion estimation. **IEEE Transactions on Image Processing**, v. 14, n. 2, p. 213–221, Feb 2005. ISSN 1941-0042.

APÊNDICE A – Script do Matlab que implementa o detector de direção do som utilizando RNA

```

morekeywords
1 %
2 %% William Machado
3 %% Lavras - 05/01/2019 (atualizado em 01/07/2019)
4 %% Engenharia de sistemas e automacao
5
6 % Algoritmo de Rede Neural
7 % O script assume as variaveis que sao definidas:
8 % Xp - input data.
9 % Tp - target data.
10 x = Xp;
11 t = Tp;
12
13 % Ajusta parametros para treinamento
14 %Lista de todas funcoes de treinamento utilizadas: help nntrain
15 %ntrainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation
16 %trainFcn = 'traingd'; % Gradient descent backpropagation.
17 %trainFcn = 'traingdm'; % Gradient descent with momentum.
18 trainFcn = 'trainnoss'; % One Step Secant.
19
20 %Definir as camadas escondidas
21 hiddenLayerSize = 20;
22 net = feedforwardnet(hiddenLayerSize);
23
24 % Configuracao de divisao dos dados para Treinamento, Validacao e Teste
25 % Para a lista de todas as funcoes: help nndivide
26 net.trainParam.epochs = 1000; % numero maximo de epocas para treinamento
27 net.divideFcn = 'dividerand'; % Divide os dados aleatoriamente
28 net.divideMode = 'sample'; % Divide por cada amostra
29 net.divideParam.trainRatio = 70/100;
30 net.divideParam.valRatio = 15/100;
31 net.divideParam.testRatio = 15/100;
32
33 %Escolher a funcao de perfomance
34 %Para a lista de todas as funcoes: help nnperformance
35 net.performFcn = 'mse'; % Mean Square Error
36
37 % Treinar a Rede
38 [net,tr] = train(net,x,t);
39
40 % Testar a Rede
41 y = net(x);
42 e = gsubtract(t,y);
43 performance = perform(net,t,y);

```



```

44 tind = vec2ind(t);
45 yind = vec2ind(y);
46 percentErrors = sum(tind ~= yind)/numel(tind);
47
48 % Recalcula a performance de Treinamento, Validacao e Teste
49 trainTargets = t .* tr.trainMask{1};
50 valTargets = t .* tr.valMask{1};
51 testTargets = t .* tr.testMask{1};
52 trainPerformance = perform(net,trainTargets,y);
53 valPerformance = perform(net,valTargets,y);
54 testPerformance = perform(net,testTargets,y);
55
56 % Vizualizar a Rede
57 view(net);
58
59 % Obtem os valores de pesos e bias em um unico vetor
60 wb = getwb(net);
61
62 % Separa os valores dos pesos e bias
63 [b,IW,LW] = separatewb(net,wb);
64 %disp(b);
65 %disp(IW);
66 %disp(LW);
67
68 % Converte de celula para vetor
69 b = cell2mat(b);
70 %disp('Bias de todas camadas =');
71 %disp(b);
72
73 % Converte de celula para vetor
74 Wescondida = cell2mat(IW);
75 %disp('Pesos da camada escondida =');
76
77 Wescondida = Wescondida';
78
79 % Converte de celula para vetor
80 Wsaida = cell2mat(LW);
81 %disp('Pesos da camada de saida =');
82 %disp(Wsaida);
83
84 %Plotar as funcoes
85 figure, plotperform(tr)
86 figure, plottrainstate(tr)
87 figure, ploterrhist(e)
88 figure, plotconfusion(valTargets, y, 'Validation',testTargets, y, 'Test',
    trainTargets, y, 'Training', t, y, 'Total')
89 figure, plotroc(t,y)

```

APÊNDICE B – Simulação do SDA no Matlab

```

breaklines
1 % MATLAB R2016a
2 % Data: 01/09/2019
3 % William Machado
4 % Engenharia de Sistemas e automacao
5 % Teste determinacao da direcao do som por defasamento utilizando
6 %diferenca entre os dois sinais deslocados no tempo
7 % Exemplos de som baixado do site:
8 % http://www.get-sounds.com/
9
10 clc;
11 clear;
12 close all
13
14 disp('Programa para determinar a direcao sonora: esquerdo ou direito.')

```

```

45 [testeSom19]=audioread('C:\Users\willi\Desktop\Sons\FAST_TRN.wav');
46 [testeSom20]=audioread('C:\Users\willi\Desktop\Sons\FIGHTER.wav');
47 [testeSom21]=audioread('C:\Users\willi\Desktop\Sons\FIGHTERP.wav');
48 [testeSom22]=audioread('C:\Users\willi\Desktop\Sons\FIRE_ENG.wav');
49 [testeSom23]=audioread('C:\Users\willi\Desktop\Sons\FIRE_TRK.wav');
50 [testeSom24]=audioread('C:\Users\willi\Desktop\Sons\GAS_TRBP.wav');
51 [testeSom25]=audioread('C:\Users\willi\Desktop\Sons\GRINDGRS.wav');
52 [testeSom26]=audioread('C:\Users\willi\Desktop\Sons\HELICPTI.wav');
53 [testeSom27]=audioread('C:\Users\willi\Desktop\Sons\HIGH_SPD.wav');
54 [testeSom28]=audioread('C:\Users\willi\Desktop\Sons\HOT_ROD.wav');
55 [testeSom29]=audioread('C:\Users\willi\Desktop\Sons\IDLING_A.wav');
56 [testeSom30]=audioread('C:\Users\willi\Desktop\Sons\INCOMING.wav');
57 [testeSom31]=audioread('C:\Users\willi\Desktop\Sons\JAGUAR.wav');
58 [testeSom32]=audioread('C:\Users\willi\Desktop\Sons\JET_FLY.wav');
59 [testeSom33]=audioread('C:\Users\willi\Desktop\Sons\JETS.wav');
60 [testeSom34]=audioread('C:\Users\willi\Desktop\Sons\LAND.wav');
61 [testeSom35]=audioread('C:\Users\willi\Desktop\Sons\MANIAC.wav');
62 [testeSom36]=audioread('C:\Users\willi\Desktop\Sons\MOTORCYC.wav');
63 [testeSom37]=audioread('C:\Users\willi\Desktop\Sons\OLD_DSLB.wav');
64 [testeSom38]=audioread('C:\Users\willi\Desktop\Sons\OLD_TRCT.wav');
65 [testeSom39]=audioread('C:\Users\willi\Desktop\Sons\POL_CAR.wav');
66 [testeSom40]=audioread('C:\Users\willi\Desktop\Sons\SKIDDNG.wav');
67 [testeSom41]=audioread('C:\Users\willi\Desktop\Sons\SPRTSCRA.wav');
68 [testeSom42]=audioread('C:\Users\willi\Desktop\Sons\tAXI_HRN.wav');
69 [testeSom43]=audioread('C:\Users\willi\Desktop\Sons\tIN_LV.wav');
70 [testeSom44]=audioread('C:\Users\willi\Desktop\Sons\tINY_TRN.wav');
71 [testeSom45]=audioread('C:\Users\willi\Desktop\Sons\tIRE_BLW.wav');
72 [testeSom46]=audioread('C:\Users\willi\Desktop\Sons\tRUCK_HO.wav');
73 [testeSom47]=audioread('C:\Users\willi\Desktop\Sons\tRUCK_PS.wav');
74 [testeSom48]=audioread('C:\Users\willi\Desktop\Sons\tUGBOAT.wav');
75 [testeSom49]=audioread('C:\Users\willi\Desktop\Sons\tV6_REV.wav');
76 [testeSom50]=audioread('C:\Users\willi\Desktop\Sons\tV8_REV.wav');
77 [testeSom51]=audioread('C:\Users\willi\Desktop\Sons\tURBULEN.wav');
78
79 s = testeSom33;
80
81 % Inicio para analisar:
82 % a partir de um limiar, para evitar pegar silencio:
83 s_max = max(s);
84 limiar = 0.1 * s_max; % limiar de 10% do valor maximo do som
85 for h = 20:length(s) % comeca a partir da amostra 20
86     if s(h) > limiar
87         inicio = h;
88         break;
89     end;
90 end
91

```

```

92 % seleciona apenas uma quantidade de amostras a partir de um inicio,
93 % e cria dois vetores defasados com o som original;
94 % 4.5ms de tempo equivale a 100 amostras
95 X = s(inicio:(inicio+amostrasX-1));
96 Y = s((inicio+atraso+amostrasY):(inicio+2*amostrasY+atraso-1));
97
98 % normalizar para ficar igual ao CAD do microcontrolador:
99 % valores entre 0 e 100
100 smin = min([min(X'); min(Y')]);
101 smax = max([max(X'); max(Y')]);
102
103 % coloca X entre 0 e 1:
104 X = (X - smin)/(smax - smin);
105 % coloca X entre 0 e 100
106 X = round(X*100);
107
108 %normalizar o som
109 % coloca Y entre 0 e 1:
110 Y = (Y - smin)/(smax - smin);
111 % coloca Y entre 0 e 100
112 Y = round(Y*100);
113
114 %plotar o grafico
115 figure(1);
116 plot(X, 'r');
117 hold on
118 plot((amostrasY+[1:length(Y)]), Y, 'b');
119
120 %declaracao dos valores das variaveis
121 Dif_max = 0;
122 Dif_min = 60000;
123 pontoMin = 1;
124 pontoMax = 1;
125
126 %SDA
127 for i=1:(length(X)-amostrasY)
128     %calcula do SDA
129     Diferenca=sum(abs(X(i:(i+amostrasY-1))-Y)');
130     disp('Diferenca')
131     disp(Diferenca)
132     grafico_diferenca(i)= Diferenca;
133
134     % Descubre se a diferenca e' minima
135     if (Dif_min > Diferenca)
136         Dif_min = Diferenca;
137         pontoMin = i;
138     end;

```

```
139 end
140
141 figure(2);
142 plot(grafico_diferenca)
143 disp('Ponto min:');
144 disp(pontoMin);
145 disp('Lado que vem o som:');
146 if (pontoMin > amostrasY)
147     disp('Lado direito')
148 else
149     disp('Lado esquerdo')
150 end
```

APÊNDICE C – Simulação do SDA em C

```

breaklines
1
2 /*
3 *   WILLIAM MACHADO
4 *   Lavras - 20/09/2019
5 *   ENGENHARIA DE SISTEMAS E AUTOMACAO
6 *   ALGORITMO PARA DETECCAO DA DIRECAO SONORA
7 */
8
9 #include <stdio.h>
10 #include <conio.h>
11 #include <math.h>
12
13 #define TamY 30
14 #define TamX TamY*3
15 #define TamXmenosY TamX-TamY
16
17 int main(){
18 //Vetores de exemplo para teste
19   char X[TamX] = {69, 69, 69, 67, 75, 83, 63, 33, 33, 41, 52, 63, 49, 29,
20                 56, 89, 96, 100, 86, 73, 71, 71, 75, 77, 63, 47, 20, 0, 3, 10, 23,
21                 38, 56, 68, 66, 60, 45, 49, 76, 73, 32, 33, 71, 74, 42, 21, 7, 10,
22                 23, 37, 50, 60, 67, 77, 85, 80, 71, 55, 37, 35, 34, 34, 34, 26, 17,
23                 18, 20, 15, 11, 11, 10, 5, 2, 10, 17, 17, 17, 10, 8, 12, 17, 21, 21,
24                 12, 14, 27, 46, 69, 84, 87};
25   char Y[TamY] = {10, 23, 38, 56, 68, 66, 60, 45, 49, 76, 73, 32, 33, 71,
26                 74, 42, 21, 7, 10, 23, 37, 50, 60, 67, 77, 85, 80, 71, 55, 37};
27   char diferenca = 0;
28   int Dif_min = 60000;
29   int pontoMin = 0, k =0, n, i=0;
30
31 //Loop para calcular SDA
32   while (i < TamXmenosY){
33     int SDA=0;
34 //laco onde desloca o vetor Y
35     for(n = i; n < i+TamY; n++){
36       if (k > TamY-1){
37         k =0;
38       }
39       printf("_____loop %d_____\\n", i);
40 //calcula a diferenca absoluta entre os vetores
41       diferenca = abs(X[n] - Y[k]);
42       printf("calculado da diferenca abs %d e: %d\\n", i, diferenca);
43 //somatorio das diferencas absolutas
44       SDA = SDA + diferenca;

```

```
39     printf("valor Y = k %d\n", k);
40     printf("valor X = n %d\n", n);
41     printf("valor loop = i %d\n", i);
42     printf("somatorio da diferenca SDA %d e: %d\n", i, SDA);
43     printf("-----\n");
44     k++;
45 }
46 //Descobre se a diferenca encontrada e menor que a proxima e salva a
    posicao
47     if (Dif_min > SDA){
48         Dif_min = SDA;
49         pontoMin = i;
50     }
51     printf("o ponto min e: %d\n", pontoMin);
52     i ++;
53 }
54 //verificar qual lado o som chega
55 printf("Lado que vem o som:");
56 if (pontoMin > TamY){
57     printf("Lado Direito");
58 }else{
59     printf("Lado Esquerdo");
60 }
61 }
```

APÊNDICE D – Implementação do SDA no PIC16F688

```

breaklines
1 /*
2 *   WILLIAM MACHADO
3 *   Lavras - 27/09/2019
4 *   ENGENHARIA DE SISTEMAS E AUTOMACAO
5 *   DETECCAO DE DIRECAO SONORA USANDO SDA
6 */
7 // Pinos do microcontrolador 16F688:
8 /* RC4 - TX      OUT
9   RC5 - RX      IN
10  RA1/AN1 - ENTR. DIREITA
11  RA2/AN2 - SAIDA DIREITA
12  RC0/AN4 - LED
13  RC1/AN5 - ENTR. ESQUERDA
14  RC3/AN7 - SAIDA ESQUERDA
15 */
16 //***** definicao de variaveis *****
17 unsigned int valor_AD = 0;      // variavel que armazena resultado CAD
18 //unsigned int uart_rd;
19 #define temp1  1000
20 #define temp2  2000
21 #define atraso1 40
22 #define atraso2 10
23 #define atrasoY 40
24 #define TamY 10           //tamanho do vetor Y
25 #define TamX 30          //tamanho do vetor X = 3*Y
26 #define TamXmenosY 20 // tamanho de X - tamanho de Y
27
28 unsigned int X[TamX];
29 unsigned int Y[TamY];
30 int diferenca = 0;
31 unsigned static int i=0, k =0, n;
32 int Dif_min = 60000;
33 unsigned int pontoMin = 0;
34 unsigned int SDA=0;
35 unsigned int valor_AD_D = 0, valor_AD_E =0;
36 unsigned long int energia =0;
37 long int val_canalum =0, val_canalcinco =0;
38
39 //***** programa principal *****
40 void main() {
41 /* B7:0 - seleciona entre analogico ou digital a funcao nos pinos
42 *   1 = Entrada Analogica **(Pino e definido como entrada analogica)
43 *   0 = Digital I/O.

```



```

44 *    **OBS: Colocando o pino no analogico desativa automaticamente o
      circuito de
45 *    entrada digital, fracos pull-ups, e interrupt-on-change se existir.
46 *    O TRIS bit correspondente Deve ser ajustado para o modo de entrada
47 *    para permitir o controle externo da tensao no pino.
48 */
49 ANSEL = 0b00100010;
50 // Inicialmente deve-se configurar os registradores de controle do CAD
51 /* Configura e define pinos de A/D - ADCON0
52 *    B7 - 1: justificado a direita //A justificacao ajuda quando
      desejamos trabalhar
53 *    com apenas 8 bits na conversao ao inves de 10.
54 *    Neste caso justificamos a esquerda onde utilizariamos
55 *    todos os bits em ADRESH e desprezariamos os bits em ADRESL.
56 *    0: justificado a esquerda
57 *    B6 - tensao de referencia
58 *    1: Vref (externo)
59 *    0: Vdd (interno)
60 *    B5 - NC
61 *    B4:2 - Selecao do canal
62 *           000 = Canal 00 (AN0)
63 *           001 = Canal 01 (AN1)
64 *           010 = Canal 02 (AN2)
65 *           011 = Canal 03 (AN3)
66 *           100 = Canal 04 (AN4)
67 *           101 = Canal 05 (AN5)
68 *           110 = Canal 06 (AN6)
69 *           111 = Canal 07 (AN7)
70 *    B1 - Status da conversao
71 *           0 - Conversao completa ou nao iniciada;
72 *           1 - Conversao iniciada. O bit e limpo automaticamente
73 *           quando completar a conversao.
74 *    B0 - habilita o modulo ADC
75 *           0: Conversor desligado e sem consumo de corrente
76 *           1: Conversor ligado
77 */
78 ADCON0 = 0b10010101; //justificado a direita,
79 // tensao de referencia interna,
80 // selecao do canal AN5, conversor ligado
81
82 /* Configura A/D - ADCON1
83 *    B7 - Nc
84 *    B6:4 - Selecao do clock da conversao
85 *           000 = FOSC/2
86 *           001 = FOSC/8
87 *           010 = FOSC/32

```

```

88 *     x11 = FRC (clock derivado de um oscilador interno dedicado = 500 kHz
      max)
89 *     100 = FOSC/4
90 *     101 = FOSC/16
91 *     110 = FOSC/64
92 *     B3:0 - NC
93 */
94     ADCON1 = 0b00100000; // selecao de clock - FOSC/32
95 /*
96 *     RA1/AN1 - ENTR. DIREITA    -- microfone
97 *     RA2/AN2 - SAIDA DIREITA   -- motor
98 *     RC0/AN4 - LED
99 *     RC1/AN5 - ENTR. ESQUERDA -- microfone
100 *    RC3/AN7 - SAIDA ESQUERDA  -- motor
101 */
102     //todos os pinos do porta (AN0-AN4) e porte (AN5-AN7) como i/o de uso
      geral
103 //define porta RA1(AN1) como entrada, RA2 como saida (motor), restante como
      entrada
104     trisa = 0b11111011;
105 //define RC1(AN5) como entrada, RC0 (led) saida, RC3 como saida (motor),
106     trisc = 0b11110110; // outros pinos como entrada
107     porta = 0; //reseta todos os pinos do porta
108     portc = 0; //reseta todos os pinos do portc
109     portc.rc0 = 1; //liga o led
110
111     // Inicializa conversor A/D
112     ADC_Init();
113     //delay_ms(100); //100 milisegundo iniciar ADC
114
115     //loop execucao do programa
116     while(1){
117         //delay_ms(200);
118         energia =0;
119 //calculando energia do sinal a cada 2000 amostras coletada, equivale a 200
      ms
120         for(i=0; i < 2000; i++){
121             // capturando as amostras canal 1
122             val_canalum = ADC_read(1)>>2; //eliminar 2 bits
123             val_canalum = val_canalum - 65;
124             val_canalum = val_canalum*val_canalum;
125             // capturando as amostras canal 5
126             val_canalcinco= ADC_read(5)>>2; //eliminar 2 bits
127             val_canalcinco= val_canalcinco - 65;
128             val_canalcinco = val_canalcinco*val_canalcinco;
129             // somatorio energia
130             energia = val_canalum + val_canalcinco + energia;

```

```

131     }
132     if (energia > 60000){
133         portc.rc0 = 0; //desliga o led
134         for(i=0; i < TamY; i= i+1){ //demora 50us para cada
            aquisicao
135             X[i] = ADC_read(1)>>2; //elimina 2 bits
136             delay_us(50); //50 us
137         }
138         k=0;
139         for(i=0; i < TamY; i= i+1){
140             /*           B4:2 - Selecao do canal
141             *           000 = Canal 00 (AN0)
142             *           001 = Canal 01 (AN1)
143             *           010 = Canal 02 (AN2)
144             *           011 = Canal 03 (AN3)
145             *           100 = Canal 04 (AN4)
146             *           101 = Canal 05 (AN5)
147             *           110 = Canal 06 (AN6)
148             *           111 = Canal 07 (AN7)
149             */
150             // ADCON0 = 0b10000101; //mudar canal AN1
151             // delay_us(10); //atraso 10us para mudar o canal
152             k = i + TamY;
153             X[k] = ADC_read(1)>>2;
154             // ADCON0 = 0b10010101; //mudar canal AN5
155             // delay_us(10); //atraso 10us para mudar o canal
156             Y[i] = ADC_read(5)>>2;
157             // delay_us(5);
158         }
159         k=0;
160         for(i=0; i < TamY; i= i+1){ //demora 50us para cada
            aquisicao
161             k = i + (2*TamY);
162             X[k] = ADC_read(1)>>2;
163             delay_us(50); //50 us
164         }
165         Dif_min = 60000;
166         i =0;
167         //laco que vai percorrer todo vetor
168         while (i < TamXmenosY){
169             SDA = 0;
170             k =0;
171             //laco onde desloca o vetor Y
172             for(n = i; n < (i+TamY); n = n +1){
173                 //if (k > (TamY-1)){
174                 // k =0;
175                 //}

```

```
176         //calcula a diferenca absoluta entre os vetores
177         diferenca = abs(X[n] - Y[k]);
178
179         //somatorio das diferencas absolutas
180         SDA = SDA + diferenca;
181         k = k+1;
182     }
183     // Descobre se a diferenca encontrada e menor que a
184     // proxima
185     //e salva a posicao
186     if (Dif_min > SDA){
187         Dif_min = SDA;
188         pontoMin = i;
189     }
190     i = i +1;
191 } //fim while
192
193 //verificar qual lado o som chega e ativa o microfone
194 if (pontoMin > TamY){
195     portc.RC3 = 1; //ligar motor esquerdo
196     delay_ms(temp1);
197     portc.RC3 = 0; // desligar motor esquerdo
198     delay_ms(temp1);
199 }else{
200     porta.RA2 = 1; //ligar o motor Direito
201     delay_ms(temp1); //temp1 = 1s
202     porta.RA2 = 0; //desliga o motor Direito
203     delay_ms(temp1);
204 }
205 portc.rc0 = 1; //liga o led
206
207 }//fim if
208     delay_ms(1000);
209 } //fim while infinito
210 } //fim programa
```