

IGOR KUCZUK MODENEZI

**Eletrocardiograma de baixo custo com análise de batimentos
cardíacos utilizando *Neuro-Fuzzy***

IGOR KUCZUK MODENEZI

**Eletrocardiograma de baixo custo com análise de batimentos
cardíacos utilizando Neuro-Fuzzy**

Trabalho de Graduação apresentado ao Conselho de Curso de Graduação em Engenharia Elétrica da Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Elétrica.

Orientadora: Prof^ª. Dr^ª. Paloma Maria Silva Rocha Rizol

Guaratinguetá - SP
2018

M689e	Modenezi, Igor Kuczuk Eletrocardiograma de baixo custo com análise de batimentos cardíacos utilizando Neuro-Fuzzy / Igor Kuczuk Modenezi – Guaratinguetá, 2018. 46 f : il. Bibliografia: f. 44-46 Trabalho de Graduação em Engenharia Elétrica – Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2018. Orientadora: Profª. Drª. Paloma Maria Silva Rocha Rizol 1. Arduino (Controlador programável). 2. Eletrocardiografia. 3. MATLAB (Programa de computador). I.Título. CDU 681.5
-------	---

Luciana Máximo
Bibliotecária CRB-8/3595

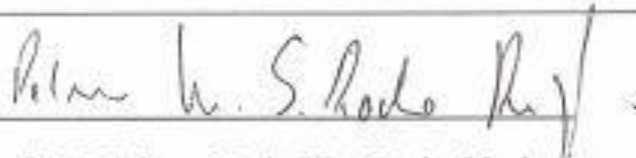
IGOR KUCZUK MODENEZI

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO
PARTE DO REQUISITO PARA A OBTENÇÃO DO DIPLOMA DE
"GRADUADO EM ENGENHARIA ELÉTRICA"

APROVADO EM SUA FORMA FINAL PELO CONSELHO DE CURSO DE
GRADUAÇÃO EM ENGENHARIA ELÉTRICA


Coordenador: Profª. Drª. Paloma Maria Silva Rocha Rizol

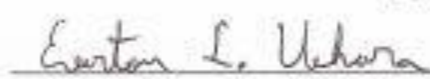
BANCA EXAMINADORA:


Prof. Drª. Paloma Maria Silva Rocha Rizol

Orientador/UNESP-FEG


Prof. Dr. Daniel Julien Barros da Silva Sampaio

UNESP-FEG


Everton Lincoln Uehara

Membro Externo

Dezembro 2018

DADOS CURRICULARES

IGOR KUCZUK MODENEZI

NASCIMENTO 12.12.1995 – São Paulo/SP

FILIAÇÃO Humberto Modenezi Neto
Marcia Kuczuk Modenezi

2014/2018 Engenharia Elétrica
FEG- Unesp

AGRADECIMENTOS

Em primeiro lugar, quero agradecer a minha família, que sempre me apoiou e esteve presente nos momentos difíceis. Sem eles, eu não estaria fazendo uma faculdade longe de casa e não chegaria tão longe.

Quero agradecer também os professores e funcionários da faculdade que tanto me ajudaram em especial a Paloma Rizol e o Daniel Julien Barros. Sua ajuda foi essencial para a minha formação, tanto pela ajuda motivacional quanto didática.

Tenho muito a agradecer a FEG Robótica, a minha família longe de casa. Aprendi muito e amadureci bastante convivendo com a equipe, fiz amizades que vou levar pra vida toda e passei por momentos inesquecíveis.

“Nossas virtudes e nossos defeitos são inseparáveis, como a força e a matéria. Quando se separam, o homem deixa de existir.”

Nikola Tesla

RESUMO

Doenças cardiovasculares são as principais causas de morte no mundo, sendo um grande custo para os cofres públicos. Dessa forma, uma detecção nos primeiros estágios é fundamental, podendo realizar tratamentos muito mais baratos. Isso, porém, requer a análise dos batimentos cardíacos por meio de eletrocardiogramas. Em muitos postos de saúde, esses aparelhos costumam ser mais antigos (utilizando papel e caneta para a marcação dos sinais) e requerem a presença de um cardiologista. Uma solução para o problema é o desenvolvimento de um eletrocardiograma de baixo custo, utilizando recursos *open source*, como o caso do Arduino e do AD8232, e um controlador *Fuzzy*, desenvolvido utilizando o ANFIS. Neste trabalho, foram considerados apenas os casos de isquemia que geram arritmia cardíaca para análise. Após o desenvolvimento do modelo no Matlab, foi realizada a medição utilizando um equipamento real para comprovação do funcionamento.

PALAVRAS-CHAVE: Arduino. Eletrocardiograma. AD8232. ANFIS. Matlab.

ABSTRACT

Cardiovascular diseases are some of the main causes of death in the world, generating a great cost to the public coffers. On account of this, the early detection is essential, allowing for cheaper treatments. This is possible through an analysis using an electrocardiogram. In health centers, this equipment are usually older (utilizing pen and paper to do the readings) and require the presence of a cardiologist. One solution to this problem is the development of a low-cost electrocardiogram, using open source resources, such as Arduino and AD8232, and a *Fuzzy* controller, developed through ANFIS. In this work, it was only considered for analysis ischemia that generates heart arrhythmia. After the development of the model in Matlab, measurements were conducted using real equipment to prove the usage.

KEYWORDS: Arduino. Electrocardiogram. AD8232. ANFIS. Matlab.

SUMÁRIO

1.	CONSIDERAÇÕES INICIAIS	9
1.1.	INTRODUÇÃO.....	9
1.2.	MOTIVAÇÃO	9
1.3.	OBJETIVO.....	9
1.4.	ETAPAS	10
2.	EMBASAMENTO TEÓRICO	11
2.1.	ARDUINO	11
2.2.	MÓDULO AD8232	11
2.3.	ELETROCARDÍOGRAFO.....	12
2.4.	FUNCIONAMENTO CARDÍACO.....	13
2.5.	ISQUEMIA CARDÍACA E ARRITMIA	13
2.6.	LÓGICA <i>FUZZY</i>	14
2.7.	REDES NEURAIS ARTIFICIAIS	16
2.8.	SISTEMA DE INFERÊNCIA ADAPTATIVO NEURO- <i>FUZZY</i> (ANFIS).....	17
3.	MODELO DO ELETROCARDÍOGRAFO	20
3.1.	LEVANTAMENTO DE PREÇO E ANÁLISE DE MERCADO	20
3.2.	MONTAGEM DO ECG	23
3.3.	DESENVOLVIMENTO DE UM SISTEMA ANFIS.....	25
4.	RESULTADOS	28
4.1.	AJUSTES DO SISTEMA ANFIS	28
4.2.	OBTENÇÃO DOS RESULTADOS.....	30
4.3.	ANÁLISE DE DADOS	33
5.	CONCLUSÕES	38
	REFERÊNCIAS	40
	ANEXO A – PROGRAMA DESENVOLVIDO NO PROCESSING	42

1. CONSIDERAÇÕES INICIAIS

1.1. INTRODUÇÃO

No Brasil, o sistema público de saúde (SUS) possui um gasto anual alto com doenças cardiovasculares. Um estudo, realizado em 2011, estima que o custo seja de R\$616.496,57 mensais, aproximadamente 12,26% da receita hospitalar (MORAIS, 2011). Apesar disso, os aparelhos de eletrocardiograma em ambulatório são geralmente antigos e analógicos, que utilizam papel milimetrado e caneta para registro dos batimentos cardíacos, acompanhado de um médico especializado.

Doenças cardiovasculares são as principais causas de mortalidade no mundo, segundo a Organização Pan-Americana de Saúde (OPAS), totalizando 17 milhões de fatalidades no ano de 2015(OPAS, 2017). De acordo com especialistas, 80% dos casos poderiam ser evitados se fossem observados com antecedência e realizados alguns tratamentos (GOVERNO DO BRASIL, 2017).

1.2. MOTIVAÇÃO

Considerando a situação da saúde pública no Brasil, uma forma de melhorar o quadro observado atualmente é o desenvolvimento de soluções de baixo custo que possam facilitar a identificação de doenças cardiovasculares. Dessa forma, garantindo um serviço ambulatório mais barato e que ainda serve para identificar os problemas.

Isso pode ser alcançado desenvolvendo um eletrocardiógrafo de baixo custo e que possa identificar problemas cardíacos mais simples, como arritmia. Assim, é possível realizar um pré-diagnóstico do paciente sem necessitar a presença do cardiologista, realizando uma triagem básica. Isso reduz o número de casos que precisam de análise do médico especialista, gerando uma economia considerável, uma vez que não seria necessário consulta dos casos que não apresentam doenças.

1.3. OBJETIVO

O objetivo do trabalho é desenvolver um eletrocardiógrafo de baixo custo utilizando Arduino e tecnologias *open-source* e, a partir dos dados medidos, utilizar o ANFIS para análise da leitura, de forma a identificar se o paciente possui arritmia cardíaca.

1.4. ETAPAS

O trabalho foi dividido em cinco capítulos, sendo cada uma dessas partes essenciais para o entendimento completo do estudo.

O primeiro capítulo apresenta a motivação e panorama da aplicação do trabalho.

O segundo capítulo apresenta a revisão teórica do trabalho, explicando brevemente os conceitos teóricos utilizados.

O terceiro capítulo apresenta a montagem do aparelho, o levantamento de preço e o *toolbox* utilizado para desenvolvimento do sistema ANFIS.

O quarto capítulo apresenta o funcionamento do eletrocardiograma, resultados apresentados no Matlab e programação adicional para atingir o objetivo desejado.

O quinto capítulo apresenta as conclusões do trabalho.

2. EMBASAMENTO TEÓRICO

2.1. ARDUINO

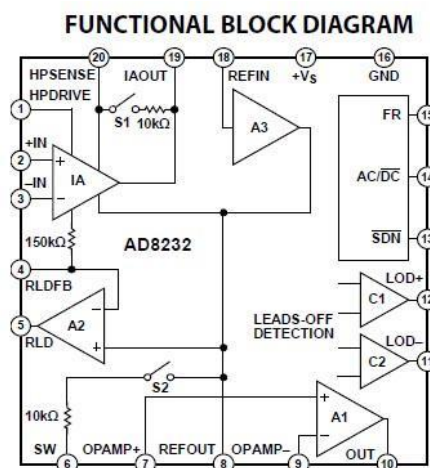
O Arduino é uma plataforma eletrônica *open-source*, desenvolvida com o intuito de facilitar o desenvolvimento de projetos para pessoas com pouco conhecimento de programação e eletrônica. Isso permite que tenha um alcance muito amplo e possa ser utilizado em uma série de aplicações de todas as áreas de conhecimento (ARDUINO, 2018).

Devido a sua natureza de acesso livre ao código e esquemático da placa eletrônica, é mais fácil desenvolver aplicações, já que se sabem previamente as suas funcionalidades e limitações. Outra vantagem é que isso permite um custo baixo, quando comparado com projetos que não são *open-source*. Esses detalhes são especialmente importantes para o desenvolvimento do eletrocardiógrafo proposto, uma vez que os principais objetivos são a acessibilidade e o baixo custo.

2.2. MÓDULO AD8232

O módulo AD8232 é um bloco de condicionamento de sinal integrado para eletrocardiograma e outras aplicações que utilizam medições biopotenciais. Assim como o Arduino, sua topologia é de fácil acesso, facilitando sua aplicação e adaptação para os projetos desejados. O diagrama de blocos funcional pode ser observado na figura 1.

Figura 1 – Diagrama de blocos funcional do AD8232



Fonte: Analog devices (2018)

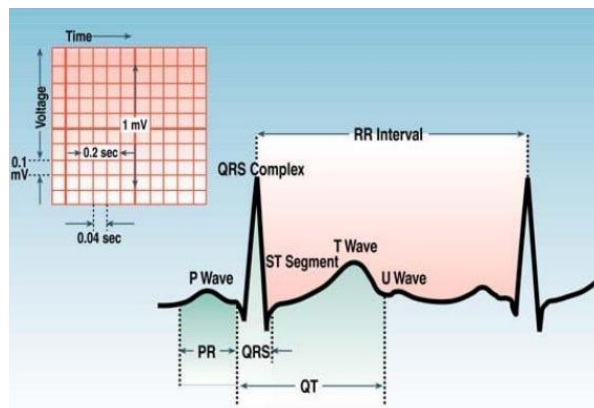
De acordo com seu *datasheet*, sua função é “extrair, amplificar, e filtrar pequenos sinais biopotenciais, na presença de condições de ruído” (ANALOG DEVICES, 2018).

Uma grande vantagem de seu uso (e o motivo de sua escolha) é sua natureza *open-source*, uma vez que sua topologia é possível de ser observada no *datasheet*. Dessa forma, o módulo selecionado possui uma série de revendedores e fabricantes, tanto nacionais (BAÚ DA ELETRÔNICA (2018)) como internacionais (SPARKFUN (2014)). Isso permitiu uma documentação mais abrangente de seu uso e competitividade de preço, além da facilidade de uso.

2.3. ELETROCARDIOGRAMA

O eletrocardiógrafo (ECG) é um aparelho de medição de batimentos cardíacos por meio do princípio da corrente elétrica que percorre o coração durante o funcionamento normal do órgão. Se o sinal apresentado for diferente do comportamento normal, a anomalia deve ser analisada para detectar se é algum tipo de doença. O modelo de leitura pode ser observado na figura 2.

Figura 2 – Leitura do batimento cardíaco em um ECG



Fonte: Yanovitz (2018)

Para a detecção de arritmia cardíaca, existem alguns valores a serem analisados. É possível detectar tal anomalia a partir do chamado intervalo RR, dando o valor instantâneo do batimento cardíaco do paciente. Tal método foi escolhido devido à precisão da detecção de arritmia, observável no estudo de análise de ECG realizado por Luz et al. (2015), que indica

pelo menos 90% de eficácia, aumentando de acordo com o método empreendido. Tal intervalo será apresentado no tópico que discute isquemia cardíaca.

2.4. FUNCIONAMENTO CARDÍACO

O funcionamento cardíaco normal dá-se por um conjunto de células capazes de “gerar estímulo elétrico espontaneamente” (SCAVANACCA,2011), chamados de nó sinoatrial ou nó sinusal, localizado junto ao átrio direito, próximo à veia cava superior.

Segundo Scavanacca, o impulso elétrico gerado por esse nó é propagado pelo tecido do átrio direito, que contrai sistematicamente, ação que resulta na contração também no átrio esquerdo. Como essas contrações são sincronizadas, as câmaras superiores são reduzidas, levando o sangue às camadas inferiores, denominadas de ventrículos.

Dessa forma, é necessário que haja a perfeita sincronia entre as contrações para o funcionamento adequado. Caso contrário, o batimento cardíaco será afetado, sinal de que há uma anomalia ou doença cardiovascular.

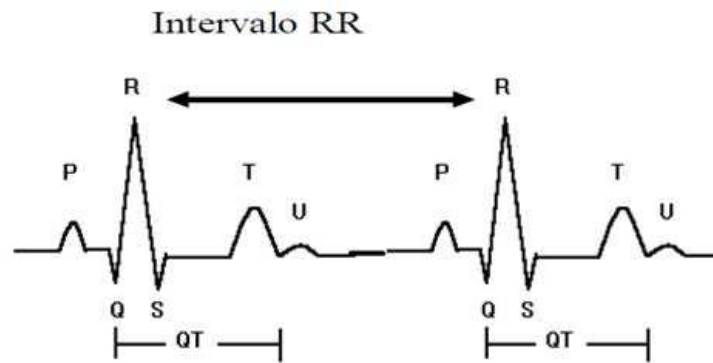
2.5. ISQUEMIA CARDÍACA E ARRITMIA

A isquemia é definida como a falta de suprimento de sangue a um músculo, resultando em baixa oxigenação. No caso da isquemia cardíaca, o problema ocorre devido à obstrução das veias. O resultado disso é um maior esforço cardíaco, observável por meio de arritmia cardíaca.

A arritmia cardíaca é definida como um descompasso dos batimentos cardíacos, fazendo com que aumente ou diminua a frequência. Considerando a isquemia, o problema observado é o maior esforço para realizar a contração muscular, observável na elevação da frequência cardíaca.

Para entender melhor o impacto no batimento cardíaco, deve-se considerar o seguinte gráfico:

Figura 3 – Padrão do batimento cardíaco



Fonte: Messias (2010)

A arritmia pode ser observada na alteração do sinal observado no intervalo QT. Para realizar a análise proposta por esse projeto, deve ser observado o intervalo RR, definido como ciclo ventricular. Ele é definido como o período entre os picos presente no trecho QRS (onde o ponto Q é o primeiro mínimo, o ponto R é o pico e o ponto S é o segundo mínimo).

Dessa forma, o intervalo RR reflete o intervalo do ciclo ventricular, podendo calcular o batimento cardíaco da seguinte forma:

$$BPM = \frac{60}{RR} \quad (1)$$

Onde o valor encontrado será a frequência cardíaca, dada em batimentos por minuto (BPM). Sabendo-se que a faixa considerada normal é entre 60 e 100 bpm, pode-se converter para obter a faixa de valores aceitáveis para o intervalo RR.

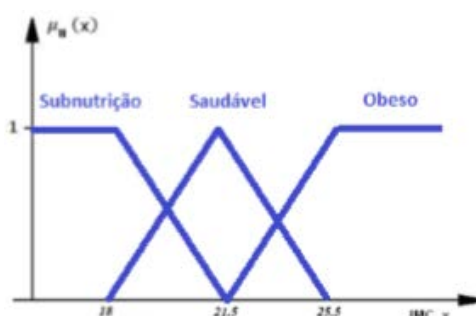
2.6. LÓGICA FUZZY

A lógica *Fuzzy* é representada por um conjunto de regras linguísticas. Assim, é realizado um conjunto de regras *se...então*, definido pelo usuário. “O resultado é um sistema de inferência baseado em regras, na qual a Teoria de Conjuntos *Fuzzy* e Lógica *Fuzzy* fornecem o ferramental matemático para se lidar com as tais regras linguísticas.”

A teoria de Conjuntos *Fuzzy*, concebida por L.A. Zaleh, permite o tratamento de informações de caráter vago ou impreciso, utilizando seus ferramentais matemáticos. Isso permite a utilização de incerteza de um dado conjunto, que será explicado melhor abaixo.

Na teoria de conjuntos clássicos, algo pode pertencer ou não a um conjunto. Na lógica *fuzzy*, o objeto possui um grau de pertinência ao conjunto, podendo pertencer parcialmente a um conjunto. De forma gráfica, a transição abrupta entre conjuntos é substituída por uma rampa de descida, que se sobrepõe aos outros em sua vizinhança. Isso pode ser compreendido melhor na figura 4.

Figura 4 – Exemplo de funções de pertinência



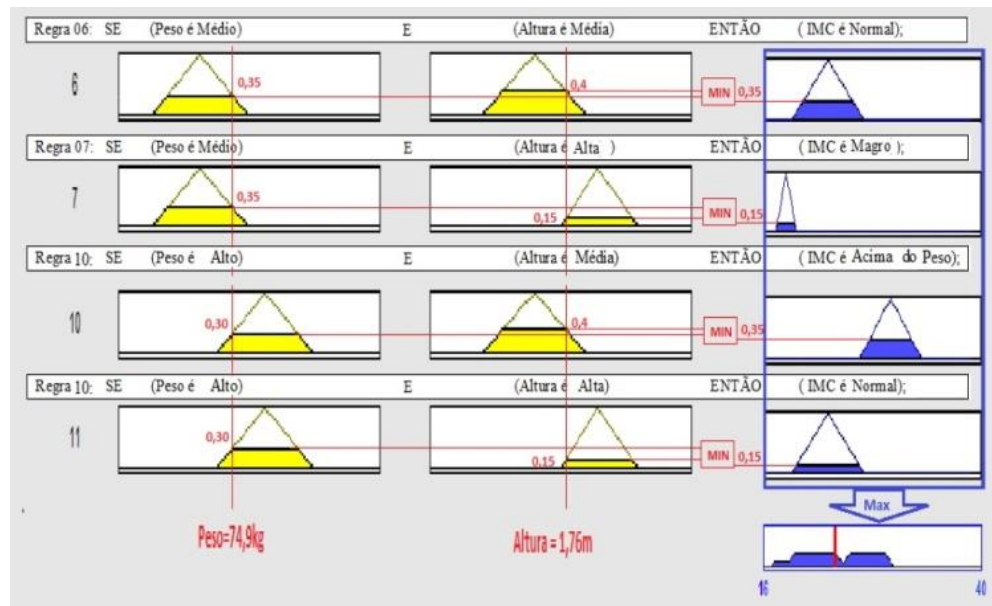
Fonte: Rizol, Dias (2014)

Assim, utilizando-se desse exemplo para explicar o funcionamento, o grau de pertinência (eixo das ordenadas no gráfico acima) é representado de acordo com o valor de IMC (eixo das abscissas).

Ainda utilizando-se do exemplo do IMC, é possível criar um sistema de conjuntos para realizar tal cálculo. Nesse caso, é necessário um conjunto de pesos e de alturas. Após definidos alguns conjuntos (alto, médio, baixo) e (gordo, médio, magro), são montados os conjuntos considerando grau de pertinência, como a figura 4. Para realizar o cálculo do IMC, é necessário realizar inferência. Para isso, existem muitos métodos (Mamdani, Takagi-Sugeno, entre outros).

Para realizar uma explicação breve, foi considerada a inferência Mamdani. Todos os pares de valores de altura e peso retornam áreas em seus determinados sistemas de conjuntos, ao considerar uma figura do valor zero de pertinência até o valor obtido no eixo das abscissas. Utilizando-se de um conjunto de regras *se...então*, é possível obter múltiplas áreas de cada variável de entrada. Em seguida, é analisado cada par de áreas para observar o mínimo grau de pertinência, que será o valor utilizado para definir a área no conjunto de saídas. Após determinadas todas as áreas possíveis, por meio das regras estabelecidas de *se...então*, as áreas dos conjuntos na saída são somados. O procedimento pode ser observado na figura 5.

Figura 5 – Regras do Sistema Fuzzy



Fonte: Rizol, Dias (2014)

Assim, o valor do centroide da área resultante (representado pelo marcador vermelho na figura do canto inferior direito) é o valor do IMC.

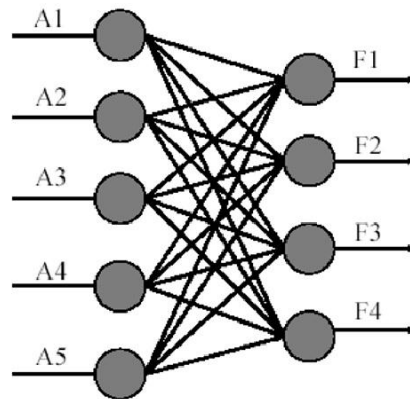
A partir desse exemplo simples, é possível perceber que um dado sistema físico, com conjuntos definidos de variáveis de entrada e de saída e regras *se...então* pode ter um resultado calculado de modo simples, quando utilizado ferramentas computacionais, como *Matlab* e semelhantes.

2.7. REDES NEURAIS ARTIFICIAIS

A base do modelo de redes neurais artificiais é inspirada pelo funcionamento neural do ser humano: uma série de sinapses conectam diversos neurônios, sendo que as informações percorrem várias camadas até que o processamento da informação seja devidamente realizado.

A modelagem matemática é de fácil compreensão: as redes neurais artificiais são compostas por várias unidades de processamento (análogos aos neurônios), conectadas por canais de comunicações (análogos às sinapses), geralmente associados a um peso. As unidades realizam processamento apenas dos dados de entrada, sendo que o comportamento inteligente é proveniente das interações entre as unidades. Um modelo genérico do sistema pode ser observado na figura

Figura 6 – Exemplo de redes neurais



Fonte: Rizol (2018)

Cada unidade realiza uma série de operações básicas: cada uma recebe um vetor de valores como entrada; cada um desses valores recebe um peso, que representa sua influência na saída da unidade e é realizada em seguida a soma ponderada. Se o valor exceder certo limite, a unidade produz um sinal de saída, que será considerado para o processamento total da rede.

Para a definição dos pesos, é utilizado um processo iterativo, por meio de uma classe de problemas pré-determinada. Assim, por meio de uma sequência de gerações, os pesos são definidos, considerando o escopo de entradas e saídas apresentadas.

Um problema apresentado, porém, é de que esses valores são de difícil acesso e leitura para o usuário. Assim, se for fornecido apenas o sistema, é complicado encontrar os valores que o treinaram, dificultando saber o escopo de atuação, caso não haja uma documentação.

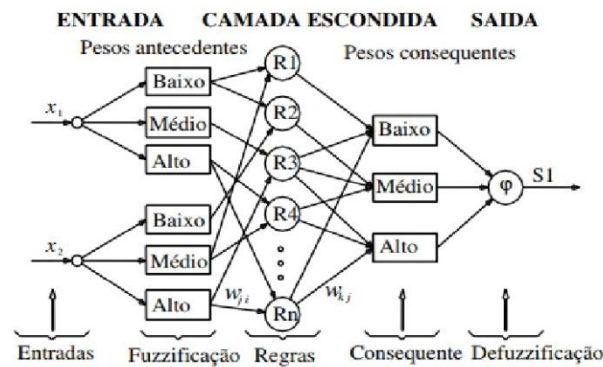
2.8. SISTEMA DE INFERÊNCIA ADAPTATIVO NEURO-FUZZY (ANFIS)

Os sistemas *Fuzzy* e redes neurais artificiais apresentam algumas diferenças quanto aos seus desenvolvimentos: os sistemas *Fuzzy* são adequados para a modelagem de controladores a partir do conhecimento explícito de quem o desenvolve, enquanto as redes neurais são baseadas no conhecimento implícito embutido em um conjunto de dados.

Dessa forma, o sistema *Fuzzy* é tão bom quanto o conhecimento do sistema por quem o desenvolve e a rede neural depende de um bom ajuste de seus parâmetros (número de neurônio, de camadas, entre outros fatores). Buscando usufruir das qualidades de cada sistema e minimizar suas deficiências, foram desenvolvidos modelos híbridos.

O sistema *Neuro-Fuzzy* é um sistema híbrido incorporado, ou seja, não é possível a separação entre os dois subsistemas, devido ao alto grau de hibridização. A arquitetura básica pode ser observada na figura 7.

Figura 7 – Arquitetura básica de um *Neuro-Fuzzy* genérico

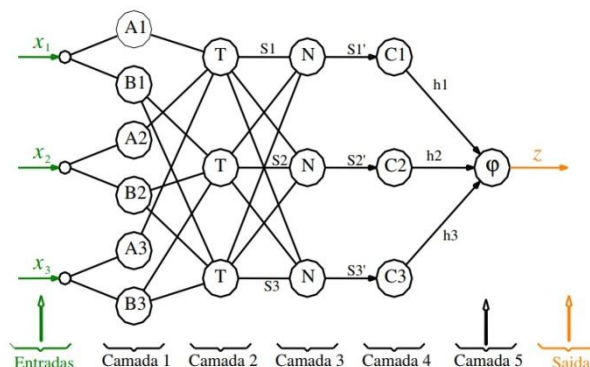


Fonte: Sanchez (2009)

A partir do conhecimento explícito do especialista unido do conhecimento implícito obtido por meio do conjunto de dados, o sistema *Neuro-Fuzzy* permite a extração do conhecimento baseada nas regras de inferência *Fuzzy*.

O sistema *Neuro-Fuzzy* ANFIS, criado por Robert Jang, é um sistema *Fuzzy* tipo Takagi-Sugeno com capacidade de aprendizado. A arquitetura em cinco camadas pode ser observada na figura 8.

Figura 8 – Arquitetura básica de um *Neuro-Fuzzy* ANFIS



Fonte: Sanchez (2009)

A seguir, será apresentada brevemente cada uma das camadas: as saídas da primeira camada são os graus de pertinência das entradas, de acordo com cada regra; na segunda

camada, calcula-se o grau de pertinência ao qual é submetido o consequente de cada regra; na terceira camada, é realizada a normalização dos graus de ativação das regras; na quarta camada, é realizado o produto da camada anterior pelo grau de ativação da consequente; e a saída da última camada é a defuzzificação do sistema, que fornece a saída do sistema ANFIS.

O processo é repetido por uma determinada quantidade de gerações (ou até alcançar um erro pré-determinado). Um cuidado que deve ser tomado, porém, é o chamado *overfitting*. Tal fenômeno ocorre quando é determinada uma quantidade muito alta de gerações, de forma que o erro seja muito baixo para os valores de treino, mas, ao testar com valores diferentes, apresentará um erro elevado. Isso ocorre porque a curva de resposta do sistema se aproxima muito da dos valores apresentados, algo que pode não ser desejado, uma vez que a curva de entradas genérica do sistema pode não se aproximar dessa.

3. MODELO DO ELETROCARDIÓGRAFO

3.1. LEVANTAMENTO DE PREÇO E ANÁLISE DE MERCADO

Para a montagem física do projeto, foram utilizados os seguintes materiais:

- *Protoboard*;
- Arduino Uno;
- Módulo AD8232;
- Conjunto de três eletrodos e um conector, para conexão no módulo;
- *Jumpers*, muito utilizados em projetos com Arduino.

Em relação ao custo, a *protoboard* e os *jumpers* foram comprados com um kit básico de Arduino, custando por volta de 30 reais e o Arduino Uno foi 50 reais. Quanto ao módulo AD8232 e os eletrodos, sua venda é realizada em um pacote único, pronto para uso. Para esse projeto, foi comprado na *AliExpress*, uma vez que é um produto prova de conceito, custando 30 reais. Se for comprado no Brasil, seu preço é mais elevado, sendo que o melhor preço encontrado foi de 90 reais, na loja digital “Baú Eletrônico”.

Assim, o preço final foi de 110 reais. No caso de realizar a compra inteiramente no Brasil, o custo é em torno de 170 reais. Pode ser observada abaixo a tabela comparativa (tabela 1).

Tabela 1 – Levantamento de custos para confecção do ECG proposto

Compra no Brasil e exterior		Compra exclusivamente no Brasil	
Descrição	Preço	Descrição	Preço
Arduino Uno	R\$50,00	Arduino Uno	R\$50,00
Protoboard e Jumpers	R\$30,00	Protoboard e Jumpers	R\$30,00
Módulo AD8232	R\$30,00	Módulo AD8232	R\$90,00
Total: R\$110,00		Total: R\$170,00	

Fonte: Produção do próprio autor

Pensando em uma montagem permanente, o custo pode ser reduzido substituindo a *protoboard* por uma placa de fenolite perfurada que, para a aplicação desejada, pode ser pequena. Um modelo de 7cmx5cm, encontrado no site “Eletrogate”, custa 2,90 reais. Nesse

caso, é necessário realizar solda dos componentes e fios para conectar ao Arduino. Como a utilização desses materiais é bem reduzida (cinco fios que precisam conectar ao Arduino, sendo que são vendidos em rolos de 200 m e estanho para poucas conexões), seu custo não foi considerado para o valor final. Assim, o custo (considerando a compra no *AliExpress*) é de 82,90 reais e, considerando a compra realizada apenas no Brasil, o custo é de 142,90 reais.

Em um comparativo com os produtos oferecidos no mercado, um eletrocardiógrafo digital, portátil e com armazenamento de informações é um produto de difícil acesso no Brasil, sendo que o produto que melhor atende essas características é o Wincardio, da Micromed. Seu preço estimado é de 8000 reais, a partir de estudo de valores no mercado. O valor unitário exato não foi encontrado, uma vez que a sua venda é realizada apenas para empresas de saúde. Uma imagem do produto pode ser observada na figura 9.

Figura 9 – Modelo de um Wincardio



Fonte: Micromed (2018)

Ao comparar os produtos, é possível perceber as similaridades: é necessário um software externo para sua utilização, tem armazenagem de dados dos batimentos cardíacos e é portátil. O Wincardio, porém, apresenta claras vantagens: 12 derivações, ao invés de uma apresentada pelo projeto, o que permite um sinal melhor e mais definido; apresentação de produto melhor e mais prática, uma vez que possui uma carcaça de proteção; uso médico certificado.

A disparidade entre os produtos, porém, faz com que seja possível observar que, apesar das limitações presentes no projeto de ECG de baixo custo, o seu custo é de 1,78% (considerando a compra sendo realizada inteiramente no Brasil) o do Wincardio. Considerando que seu uso é voltado para uma triagem inicial (e que é realizado a análise para a presença de arritmia sem a presença de um cardiologista), o custo benefício para sua utilização é muito grande, já que é possível confeccionar aproximadamente 56 unidades de baixo custo com o valor da compra de um Wincardio.

Segundo dados do Ministério de Planejamento (2018), há 31 UPAs em funcionamento no estado de São Paulo e mais 41 planejadas. Dessa forma, é possível atender todas as unidades construídas com pelo menos uma unidade de ECG de baixo custo. O impacto disso é o atendimento de toda a cobertura pública do estado de São Paulo pelo custo de apenas um aparelho de custo padrão no mercado.

Além do Wincardio, também foi encontrado um ECG portátil, da empresa Next. Esse produto apresenta o mesmo número de canais do projeto desenvolvido, mas não realiza análise e também necessita de um software para as medições. O ponto positivo é a praticidade de uso, uma vez que possui um display com o batimento cardíaco. Seu uso não é voltado para o meio hospitalar e sua venda é realizada por meio do “Mercado Livre”. Isso indica a procura de pessoas leigas para realização dos exames em casa, apesar de ainda ser necessário um especialista para avaliar a condição cardíaca. Uma foto do produto pode ser observada na figura 10.

Figura 10 – Modelo de um ECG Next



Fonte: Mercado livre (2018)

Buscando realizar uma comparação abrangente entre os eletrocardiógrafos presentes no mercado, foi encontrado um modelo muito parecido com os presentes em UPAs, para usar como base de preço. A grande diferença entre os outros é que necessita de um cardiologista presente na consulta para realizar as medições e análise dos dados observados.

Dessa forma, os preços dos produtos podem ser resumidos na tabela 2.

Tabela 2 – Comparação entre preços de ECGs presentes no mercado brasileiro

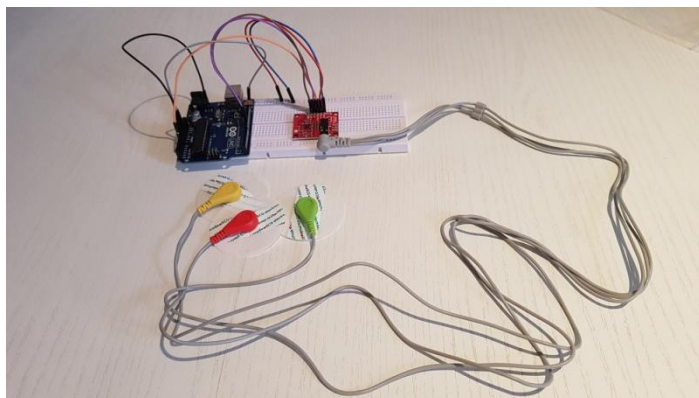
Descrição	Preço
ECG de baixo custo	R\$142,90
ECG Next (ECG digital simples)	R\$950,00
Wincardio (ECG digital)	R\$8000,00
Eletrocardiógrafo Interpretativo Digital	R\$6719,00
CardioCare 2000 (ECG comum)	

Fonte: Produção do próprio autor

3.2. MONTAGEM DO ECG

Como mencionado acima, a montagem física foi realizada utilizando uma *protoboard* e *jumpers*, uma vez que é uma prova de conceito e utiliza materiais que já tinham sido comprados para outras utilizações. A foto do projeto físico pode ser observada na figura 11.

Figura 11 – Montagem física do ECG de baixo custo

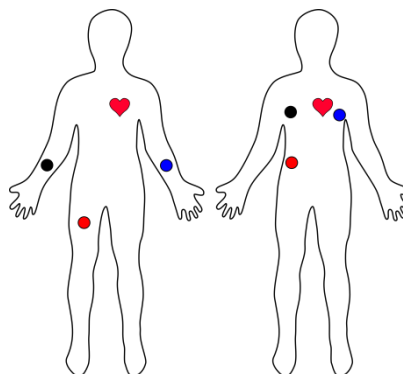


Fonte: Autoria própria

Para realizar a conexão entre o Arduino e o módulo AD8232, foram utilizadas as entradas digitais 11 para o LO+ e a 10 para o LO-, além da entrada analógica A0. As conexões nas entradas digitais são necessárias devido à detecção de pulsos elétricos em cada um dos eletrodos (um para cada lado e um de referência). A entrada analógica será responsável pela obtenção do sinal do batimento cardíaco, que será exibido na tela do computador conectado ao projeto.

Em primeiro lugar, devem-se colocar os eletrodos na superfície da pele, conforme a figura 12.

Figura 12 – Posicionamento dos eletrodos no corpo do paciente



Fonte: Sparkfun (2014)

No diagrama, a marcação preta é o eletrodo de referência, a azul é da lateral esquerda do corpo (braço esquerdo ou lateral esquerda do coração) e a vermelha é da lateral direita do corpo (perna direita ou lateral direita do coração). Devido a preocupações com eletrostática e danos aos componentes do projeto, foi utilizada uma pulseira antiestática.

Em seguida, foi desenvolvido um código para a IDE do Arduino, apresentado abaixo:

```
void setup()
{
  Serial.begin(9600);
  pinMode(11, INPUT);
  pinMode(10, INPUT);
}

void loop()
{
  if((digitalRead(10) == 1) || (digitalRead(11) == 1)){
    Serial.println('!');
  }
  else{
    Serial.println(analogRead(A0));
  }
  delay(1);
}
```

Esse código é utilizado para leitura das entradas, ou seja, obtenção dos dados provenientes do módulo AD 8232. Para a apresentação gráfica em tempo real, é necessária a utilização de um software externo, apontado pelo próprio tutorial oficial da Arduino: o Processing. Para isso, foi desenvolvido outro código apenas para esse processamento, conforme pode ser observado na íntegra no anexo A.

Assim, após compilar e executar o código na IDE Arduino, deve-se executar o código no Processing, de forma a obter um gráfico em tempo real.

3.3. DESENVOLVIMENTO DE UM SISTEMA ANFIS

Segundo um estudo comparativo entre métodos de análise utilizando inteligência artificial, foi observado que o método ANFIS é o que obtém melhor resultado, apresentando menos de 2% de erro. Assim, foi definido para esse projeto que tal sistema será utilizado, além de conhecimento prévio obtido durante a graduação.

Para isso, foi utilizado o *toolbox* de sistemas ANFIS do software Matlab versão 8.5 (2015a), uma vez que há licenças de uso nos computadores da faculdade e foi um programa muito utilizado ao decorrer da graduação, facilitando o seu uso.

Para isso, foi coletado um conjunto de dados para treino, definindo os intervalos de batimento cardíacos considerados saudáveis e os considerados com arritmia. As faixas consideradas de batimentos cardíacos e seus respectivos intervalos RR podem ser observados na tabela 3.

Tabela 3 – Valores de batimentos cardíacos normais e arrítmicos, com os seus devidos intervalos R-R

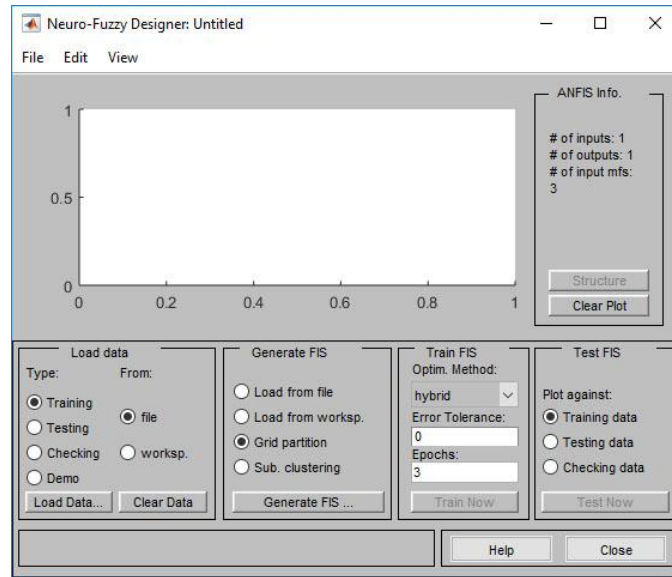
Leitura esperada	Batimento cardíaco por minuto	Intervalo R-R (milisegundos)
Normal	60 a 100 bpm	1000 ms a 600 ms
Arritmia	101 a 300 bpm	594 ms a 200 ms

Fonte: Produção do próprio autor

Em seguida, foi realizado um arquivo chamado *dadostreino.dat*, uma matriz 801x2, consistindo na faixa inteira aceitável de valores de intervalos R-R e a saída esperada, sendo zero o batimento cardíaco normal e um como o batimento cardíaco com arritmia.

Inicialmente abre-se o *toolbox* utilizando-se o comando *anfisedit*. Assim, a seguinte interface é aberta:

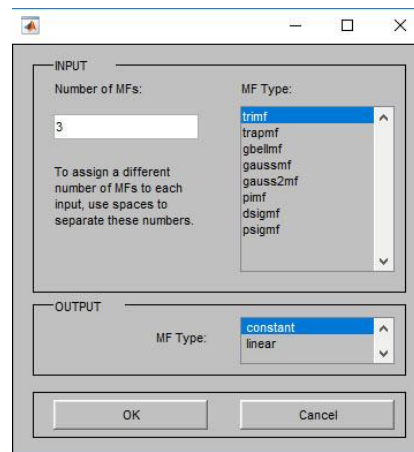
Figura 13 – Interface do *toolbox* do Neuro-Fuzzy Designer



Fonte: Autoria própria

Como o formato específico para abrir direto dessa janela utilizando a opção *file* é *.dat*, então é necessário ou realizar a conversão ou abrir de antemão (utilizando a opção *workspace*), na janela geral do Matlab. Em seguida, é selecionado “*Generate FIS...*”, que abrirá a seguinte janela:

Figura 14 – Escolha das características do sistema *Fuzzy*

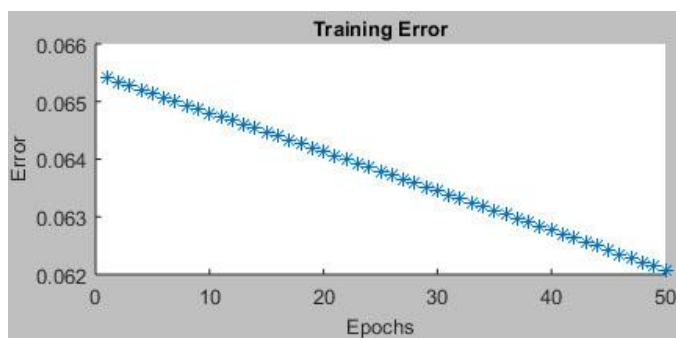


Fonte: Autoria própria

Aqui é decidida a topologia *Fuzzy* do sistema, com a quantidade de conjuntos de pertinência, formato das entradas e tipo de saída. Tais ajustes serão mais bem discutidos no Capítulo 4: Resultados.

Finalmente, é realizado o ajuste para a quantidade de épocas, método de análise do erro e limite de erro, para o treinamento do ANFIS. Assim, é realizado o treino e aparecerá na interface o erro de acordo com a quantidade de épocas, como pode ser observado no exemplo da figura 15.

Figura 15 – Erro durante o treino



Fonte: Autoria própria

Como a saída esperada é entre zero e um, o erro deve ser mantido baixo, de forma a evitar resultados falsos ou leituras equivocadas.

4. RESULTADOS

4.1. AJUSTES DO SISTEMA ANFIS

Como mencionado na tabela três, a faixa de valores de do período R-R considerados de uma pessoa normal está entre 1000 ms e 600 ms, enquanto a faixa de valores para pessoas com arritmia foi considerada entre 594 ms e 200 ms. Pode ser observado que há uma pequena quantidade de valores que está entre essas duas faixas. Como o limite de um para o outro é tênue e ainda seriam considerados no limite do aceitável, foram incluídos no conjunto de valores considerados normais. Também foi arbitrado que o valor 595 ms seria considerado anormal, de forma a facilitar a leitura dos dados.

De posse de todos os valores e da saída esperada, foi desenvolvido um arquivo *dados.dat*, de forma a reduzir a quantidade de passos necessários para realização do projeto, uma vez que foi necessário realizar múltiplos testes com os mesmos dados.

Após carregar o arquivo com os dados de treino, foi selecionado que as funções de pertinência serão do formato trapezoidal, o mais utilizado na literatura, quando se trata de Neuro-*Fuzzy*. Também foi decidido que a saída seria de formato constante, devido ao tipo de aplicação e que seriam utilizadas 20 épocas, valor arbitrado para garantir uma superfície de saída boa.

Realizados alguns testes, foi observado que a otimização por meio do método híbrido apresenta erros menores, sem a necessidade de utilizar uma quantidade de épocas grande.

Com essas informações definidas, a variável resultante a ser determinada era a quantidade das funções de pertinência da entrada. Para isso, foram realizados sucessivos testes, utilizando a comparação entre a curva obtida e os dados de treino para determinar se era necessário aumentar o número de funções de pertinência.

Treinando com três (o valor padrão definido pelo *toolbox*), é possível perceber que o formato está longe do desejado:

Figura 16 – Curva de resposta para três funções de pertinência



Fonte: Autoria própria

Em seguida foi treinado com quatro:

Figura 17 – Curva de resposta para quatro funções de pertinência

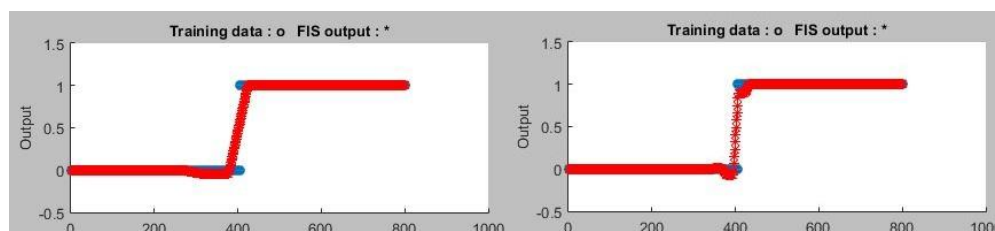


Fonte: Autoria própria

Após treinar com mais alguns valores, foi possível observar o surgimento de padrões. Para todo valor ímpar, o formato obtido é parecido com o obtido para três funções de pertinência; já os valores pares possuem o formato obtido com quatro. Como se deseja uma resposta com dois patamares (definidos como normal e arritmia), então o segundo padrão deve ser utilizado.

O que foi observado aumentando o valor foi que o tempo de transição entre patamares diminui à medida que aumentam as funções de pertinência. A partir de um ponto, porém, a diferença se tornou quase imperceptível quando comparado com os dados de entrada, mas com certeza ocorrendo *overfitting* da curva. O resultado para oito funções de pertinência pode ser comparado com o de trinta funções de pertinência, respectivamente, na figura 18.

Figura 18 – Curvas de resposta para oito e trinta funções de pertinência, respectivamente.



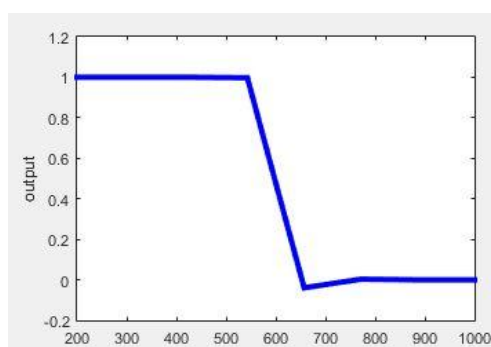
Fonte: Autoria própria

Comparando as duas curvas, é possível perceber que a alteração que ocorre é na faixa de transição entre patamares, tornando-se mais acentuada. Em um primeiro momento isso pode parecer o desejado, uma vez que responde exatamente como os dados de teste foram apresentados. O problema ocorre em uma leitura real.

Quando é realizado em condições normais, ocorrem variações do período, mesmo que mínimas. Para um sistema em que não ocorre uma rampa mais suave, uma pessoa com batimento cardíaco muito próximo do limite terá leituras que acusam arritmia, mesmo que tenha sido apenas uma variação de um período de amostragem. Assim, a curva de resposta desejada deve seguir o formato com uma rampa.

Assim, escolheu-se utilizar oito funções de pertinência, já que, a partir desse valor, as melhorias não foram acentuadas. A superfície de resposta do sistema pode ser observada figura 19.

Figura 19 – Superfície do controlador para oito funções de pertinência



Fonte: Autoria própria

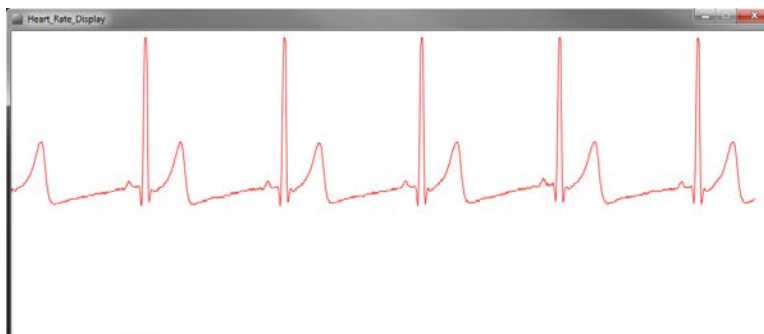
Obteve-se assim uma curva clara de valores, não ocorrendo sobrevalores e apresentando apenas uma pequena faixa com valores negativos, sendo próximo ao valor de - 0.04. Finalmente, exporta-se o modelo *Fuzzy* para uso como controlador no Simulink.

4.2. OBTENÇÃO DOS RESULTADOS

Para obtenção dos batimentos cardíacos do paciente, foi utilizado como parâmetro de medição o intervalo de batimentos R-R, como previamente citado. Isso foi realizado pela facilidade de observar o sinal, uma vez que o pico R não pode ser confundido com um ruído ou com alguma imprecisão do aparelho de leitura. Isso garante uma precisão maior de medição e análise de resultados.

O sinal encontrado durante a medição pode ser observado na figura 20.

Figura 20 – Batimentos cardíacos observados na tela



Fonte: Autoria própria

Aproveitando o uso do *Processing* para a apresentação dos dados na tela, foi realizada uma função para medição do período R-R. Após um conjunto de leituras realizadas, o usuário deve apertar “Enter” e o dado será salvo em um arquivo chamado “*medicoes.txt*”.

Inicialmente foi planejado armazenar todas as medições, mas foi encontrada uma limitação quanto à execução do programa. Para ser aceito a criação do arquivo, ele deve ser aberto e fechado. Quando é utilizado em um *loop* infinito, o compilador acusa o não fechamento do arquivo, mesmo se ele estivesse sendo realizado quando apertasse a tecla “Enter”. Apesar disso, é um funcionamento aceitável, uma vez que coleta 500 valores de intervalo.

Quanto ao código, foram realizadas algumas adições. Primeiramente, foi criada uma variável do tipo *PrintWriter*. No programa *Processing*, esse tipo de variável é utilizada para a exportação de informações para um arquivo texto. Em seguida, na função de *Setup* do programa, foi utilizado o seguinte comando:

```
output = createWriter("medicoes.txt");
```

Assim, é criado (ou sobescrito) um arquivo chamado “*medicoes.txt*” na pasta em que o programa está salvo.

Em seguida, foi adicionado uma chamada a uma função dentro da função *SerialEvent()*, realizando-a sempre que tiver uma leitura válida. Tal função foi chamada de *calculaRR()*, de forma a ser fácil de encontrar e compreender sua funcionalidade dentro do código do programa. O código será apresentado abaixo:

```
void calculaRR ()
{
    int beat_new = millis();
    int diff = beat_new - beat_old;
    beats[beatIndex] = diff;
    println(diff);
    beat_old = beat_new;
    beatIndex = (beatIndex + 1) % 500;
}
```

Com isso, é calculado cada intervalo RR válido e armazenado em um vetor, que foi definido previamente que teria dimensão de 500 valores de intervalo. Tendo realizado uma quantidade de medições, é necessário realizar a parada do programa e salvá-lo. Para isso, foi utilizada uma função de detecção de tecla pressionada. O código será apresentado abaixo:

```
void keyPressed()
{
    if(key == ENTER)
    {
        for(int i=0; i < 500;i++)
        {
            output.println(beats[i]);
        }
        output.flush();
        output.close();
        exit();
    }
}
```

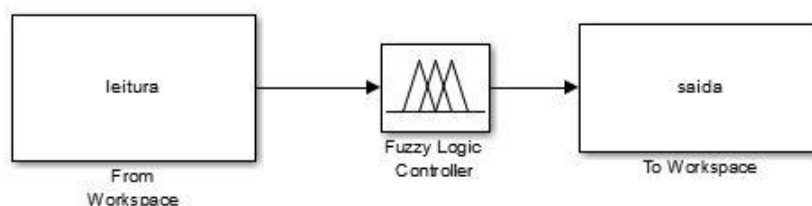
Inicialmente, foi planejado realizar todos os passos em um só programa. Porém, isso se provou complexo e desnecessário, uma vez que o passo a passo realizado é simples e o ferramental utilizado abrange exatamente o necessário.

Dessa forma, para a realização das medições, é necessário abrir o Processing, realizar as operações descritas acima e abrir o Matlab, para leitura e análise dos dados. Buscando facilitar para o usuário, é necessário apenas abrir dois arquivos (atentando ao fato que o programa do Arduino deve permanecer gravado): o de processamento e o *script* no Matlab. Deve-se também tomar o cuidado de estar tudo na mesma pasta, assim o arquivo “medicoes.txt” poderá ser criado em um programa e usado no outro.

4.3. ANÁLISE DE DADOS

De posse das leituras e do sistema ANFIS, foi realizado um modelo no “Simulink”, de forma a obter a resposta do sistema às entradas. Para isso, é necessário realizar a importação e adaptação do formato apresentado, de forma a conseguir utilizar como entrada do block “*Fuzzy Logic Controller*” e utilizar como variável importada no bloco “*From Workspace*” no “Simulink”. O sistema montado pode ser observado na figura 21.

Figura 21 – Sistema de simulação no ambiente Simulink



Fonte: Autoria própria

O funcionamento desse trecho é bem básico: a variável “leitura”, importada da área de trabalho do Matlab, contém as medições realizadas, que irão ser utilizadas como entrada do modelo *Fuzzy* desenvolvido anteriormente (exportado na etapa de desenvolvimento no *toolbox*), retornando as saídas do sistema (por meio da variável sabiamente nomeada como “saída”), sendo exportadas para a área de trabalho.

Para importar os dados, foi realizado o seguinte *script* no Matlab:

```

fileID = fopen('medicoes.txt','r');
dados = fscanf(fileID,'%d');
x=length(dados);

for i=1:x
    leitura(i,1)=i;
    leitura(i,2)=dados(i);
end

open Roda_Programa.slx

```

É importante observar que foi necessário criar uma marcação de tempo para cada medição, permitindo a simulação. Sem essa adaptação, ocorrem erros ao tentar simular o programa.

Como exemplo, será apresentado um conjunto de medições na figura 22.

Figura 22 – Conjunto de dados coletados com valores de tempo

	1	2
1	1	700
2	2	710
3	3	705
4	4	710
5	5	709
6	6	702
7	7	704
8	8	703
9	9	715
10	10	698
11	11	690
12	12	700

Fonte: Autoria própria

Essa leitura é de alguém com batimento cardíaco normal (porém um pouco elevado) de aproximadamente 85 bpm. Assim, é esperado que a saída do sistema fosse muito próxima de zero. Após a simulação, a resposta do sistema ao longo do tempo pode ser observada na figura 23:

Figura 23 – Conjunto de saídas observadas, por instante.

Time	Data:1
0	-0.0394
1	-0.0314
2	-0.0221
3	-0.0268
4	-0.0221
5	-0.0231
6	-0.0296
7	-0.0277
8	-0.0286
9	-0.0175
10	-0.0333
11	-0.0394
12	-0.0314

Fonte: Autoria própria

Como esperado, todos os valores de saída estão muito próximos de zero. Outro detalhe interessante de ser ressaltado que a leitura encontra-se exatamente na região em que os valores são negativos, que abrange aproximadamente entre 620 e 730 ms. Apesar de ser um conjunto de valores que ainda é considerado normal, pode ser considerado como um grupo que deve observar o motivo do batimento acelerado.

Considerando então todos os valores obtidos, foi realizado um *script* para exibição de uma mensagem, notificando se os batimentos cardíacos estão normais, acelerados ou arrítmicos, buscando cumprir o objetivo de acessibilidade, uma vez que o resultado apresentado não depende do usuário saber o tipo de saída esperada do sistema *Neuro-Fuzzy*.

O código final obtido, então, será esse:

```
clear
clc
threshold = 0.5;

fileID = fopen('medicoes.txt','r');
dados = fscanf(fileID,'%d');
x=length(dados);

for i=1:x
    leitura(i,1)=i;
    leitura(i,2)=dados(i);
```

```

end

open Roda_Programa.slx

open Le_Resultados.m

```

Onde o chamado “Roda_Programa.slx” é o modelo do sistema no Simulink e o *script* “Le_Resultados.m” é o que executa a exibição das mensagens. Deve-se observar que, para realizar o funcionamento normal do processo, é necessário seguir os seguintes passos:

- Executar o *script* acima;
- Rodar a simulação no Simulink;
- Executar o “Le_Resultados.m”.

Isso é necessário porque a saída do sistema simulado será a entrada de dados para a exibição das mensagens. Assim, alterar a ordem acarretará em um erro de o Matlab buscar uma variável inexistente.

O *script* “Le_Resultados.m” pode ser observado abaixo:

```

media=0;
for i=1:x
    media=media+saida(x);
end
media=media/x;

if(media<0)
    mydlg = warndlg('Batimento cardiaco acelerado!', 'Resultado');
elseif(media<threshold)
    mydlg = helpdlg('Batimento cardiaco normal', 'Resultado');
else
    mydlg = errordlg('Batimento cardiaco problematico!', 'Resultado');
end

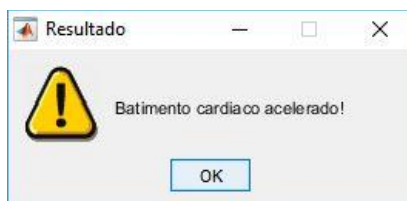
```

Buscando evitar a exibição de múltiplas mensagens na tela, foi realizada uma média dos valores coletados, com apenas uma mensagem apresentada. Além disso, foi definido um limiar (threshold = 0.5), considerado o limiar entre o batimento cardíaco normal e o com

arritmia. Considerando a curva de resposta do sistema, isso ocorre para valores abaixo de 600 ms, correspondendo com os resultados previstos na literatura.

No caso do exemplo usado, o que foi exibido na tela é apresentado na figura 24.

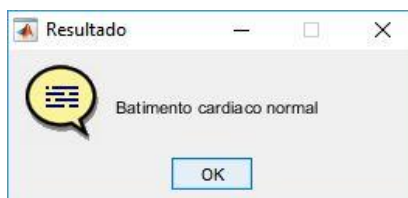
Figura 24 – Alerta de batimento cardíaco acelerado



Fonte: Autoria própria

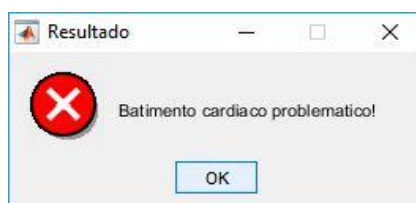
Foi realizada uma exibição gráfica simples de uma janela com o resultado, de forma a facilitar a visualização, uma vez que apenas uma mensagem no *prompt* de comandos do Matlab não é algo imediatamente claro ou fácil de perceber. Também foram utilizados tipos diferentes de símbolos, para alertar a cada um dos casos. As outras possibilidades podem ser observadas na figura 25.

Figura 25 – Mensagem de batimento cardíaco normal



Fonte: Autoria própria

Figura 26 – Mensagem de batimento cardíaco arritmico



Fonte: Autoria própria

5. CONCLUSÕES

No presente trabalho foi possível observar a utilização do sistema *Neuro-Fuzzy* em aplicado para análise do ECG. Considerando a situação atual do mercado de trabalho e do desenvolvimento de tecnologias, métodos de inteligência artificial mostram-se cada vez mais relevantes.

Utilizando-se o *toolbox* do Matlab, foi possível observar que, a partir dos dados coletados e a definição clara da saída desejada, a elaboração do modelo *Neuro-Fuzzy* ANFIS é muito simples, realizando ajustes mínimos até obter o resultado desejado. No caso apresentado, as leituras dos períodos RR permitem a avaliação do batimento cardíaco, classificando entre normal, acelerado ou fora do padrão, seguindo valores aceitos na literatura médica como consenso.

Isso foi realizado utilizando apenas um Arduino, um módulo *open source* de fácil comercialização, conhecimento sobre o modelo ANFIS e acesso a programas como Matlab. Dessa forma, é possível observar que um aparelho de baixo custo é possível e muito prático para usos em rede pública, para realizar a primeira análise do cliente. Isso é especialmente importante para o sistema de saúde brasileiro, já que existem problemas de verbas e atrasos em atendimentos. Considerando experiências vivenciadas em pronto atendimento, um ECG que permite a análise básica do batimento cardíaco permite que o cardiologista tenha sua atenção apenas em casos mais específicos ou que se sabe previamente o problema (como no caso de doenças cardíacas).

Quanto à utilização de sistemas de baixo custo, descobriu-se que não possui uma boa precisão, necessária para análises mais profundas. Isso pode ser resolvido confeccionando uma placa para obter um sinal melhor, assim como eletrodos de melhor qualidade. Porém, a proposta desse projeto foi utilizar produtos disponíveis de fácil acesso no mercado e que fossem acessíveis quanto a preço. Dessa forma, uma aplicação real necessita de adaptações que não foram discutidas aqui devido ao escopo definido.

A maior dificuldade do projeto foi a união de todas as partes. Pensou-se primeiro sobre o funcionamento de cada etapa, de forma modular e independente. Assim, foi possível realizar a solução dos problemas separadamente, garantindo-se que cada parte funciona sem as outras. Isso se provou indispensável no projeto, uma vez que a fonte do problema é muito clara e pode-se tratá-la diretamente.

Conclui-se, então, que um eletrocardiógrafo de baixo custo que utiliza um sistema *Neuro-Fuzzy* é possível de ser realizado e sua utilização é bem acessível. Apesar das

limitações e falhas, o projeto desenvolvido pode ser considerado um sucesso, já que funciona como uma boa primeira versão do produto.

REFERÊNCIAS

ANALOG DEVICES. **Single-lead, heart rate monitor front end**. Disponível em: <<https://www.analog.com/media/en/technical-documentation/data-sheets/AD8232.pdf>> Acesso em: 8 jun. 2018.

ARDUINO. **What is Arduino?** 2018. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>> Acessado em: 6 jun. 2018

BAÚ DA ELETRÔNICA. **Monitor de frequência cardíaca AD8232**. 2018. Disponível em: http://www.baudaeletronica.com.br/monitor-de-frequencia-cardiaca-ad8232.html?gclid=EAIaIQobChMIgZSRpqrL2gIVjYTIC1WbwDIEAkYBCABEgLWI_D_BwE >. Acesso em: 8 jun. 2018.

CAMPOS, M. M. de; SAITO, K. **Sistemas inteligentes em controle e automação de processos**. Rio de Janeiro: Ciência Moderna, 2004. 234 p.

CARVALHO, A. P. de L. F. **Redes neurais artificiais**. Disponível em: <<http://conteudo.icmc.usp.br/pessoas/andre/research/neural/>>. Acesso em: 06 jun. 2018.

GOVERNO DO BRASIL. **Doenças cardiovasculares são principal causa de morte no Brasil**. 2017. Disponível em: < <http://www.brasil.gov.br/noticias/saude/2017/09/doencas-cardiovasculares-sao-principal-cao-de-morte-no-mundo>>. Acesso em: 6 jun. 2018

LUZ, E. J. S. et al. **ECG-based heartbeat classification for arrhythmia detection: a survey**. 2015. 21 p. Trabalho de conclusão de curso (Graduação em Ciências da Computação)- UFOP, Ouro Preto, 2015. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0169260715003314>>. Acesso em: 06 ago. 2018.

MERCADO LIVRE. **Aparelho eletrocardiograma portátil ecg, software, bluetooth**. 2018. Disponível em: < https://produto.mercadolivre.com.br/MLB-996624222-aparelho-eletrocardiograma-portatil-ecg-software-bluetooth-_JM?matt_tool=55743738&matt_word&gclid=EAIaIQobChMIhc3ppeqV3wIVgxCRCh3ROArzEAYYASABEgI9Pfd_BwE&quantity=1> Acessado em: 16 ago. 2018.

MESSIAS, M. **Caracterização do comportamento caótico da variabilidade da frequência cardíaca (VFC) em jovens saudáveis**. 2010. Disponível em: <https://www.researchgate.net/figure/Figura-1-Exemplo-de-intervalo-RR-e-a-respectiva-serie-temporal-utilizada-para-analise-da_fig1_269673343>. Acesso em: 05 dez. 2018.

MICROMED. **Eletrocardiógrafo Wincardio**. 2018. Disponível em: <<https://micromed.ind.br/site/eletrocardiografo/wincardio/>> Acesso em: 16 ago. 2018

MINISTÉRIO DO PLANEJAMENTO. São Paulo. **UPA - unidade de pronto atendimento - São Paulo**. Disponível em: <<http://www.pac.gov.br/infraestrutura-social-e-urbana/upa-unidade-de-pronto-atendimento/sp>>. Acesso em: 06 jun. 2018.

MORAIS, M. G. T. et al. **Impacto das doenças cardiovasculares no serviço público**. 2011. 11p. Dissertação (Mestrado em Bases Gerais da Cirurgia) – Faculdade de Medicina de Botucatu, Universidade Estadual Paulista, Botucatu, 2011.

NEGNEVITSKY, M. **Artificial intelligence: a guide to intelligent systems**. Inglaterra: Pearson, 2002. 394 p.

ORGANIZAÇÃO PAN-AMERICANA DE SAÚDE (OPAS). **Doenças cardiovasculares**. 2017. Disponível em: <https://www.paho.org/bra/index.php?option=com_content&view=article&id=5253:doencas-cardiovasculares&Itemid=839> Acessado em: 6 jun. 2018

SANCHEZ, E. C. M. **Controle por aprendizado acelerado e neuro-fuzzy de sistemas servo** – hidráulicos de alta frequência. 2009. 130 f. Dissertação (Mestrado em Engenharia Mecânica) – Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2009.

RIZOL, P. M. S. R. **Material de aula introdução a inteligência artificial**. 2018.

RIZOL, P. M. S. R.; DIAS, R. A. **Desmistificando a lógica fuzzy**. 2014. 12 p. Artigo Acadêmico - UNESP – Univ. Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, Guaratinguetá, 2014.

SCANAVACCA, M.; VARELLA, D. **Arritmia cardíaca**. 2011. Disponível em: <<https://drauziovarella.uol.com.br/entrevistas-2/arritmia-cardiaca/>>. Acesso em: 06 ago. 2018.

SPARKFUN. (Estados Unidos). **AD8232 heart rate monitor hookup guide**. 2014. Disponível em: <<https://learn.sparkfun.com/tutorials/ad8232-heart-rate-monitor-hookup-guide/all>>. Acesso em: 08 jun. 2018.

UEHARA, L. E. **Implementação de neuro-fuzzy para detecção de isquemia em sinal ECG**. 2015. 8 f. Apresentação - UNESP – Univ. Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, Guaratinguetá, 2015.

YANOWITZ, Frank G. **Introduction to ECG interpretation**. 2018. Disponível em: <<https://ecg.utah.edu/pdf/>>. Acesso em: 06 ago. 2018.

ANEXO A – PROGRAMA DESENVOLVIDO NO PROCESSING

```

import processing.serial.*;

Serial myPort;
int xPos = 1;
float height_old = 0;
float height_new = 0;
float inByte = 0;
int BPM = 0;
int beat_old = 0;
int[] beats = new int[500];
int beatIndex;
float threshold = 620.0;
boolean belowThreshold = true;
PFont font;
PrintWriter output;

void setup ()
{
    // set the window size:
    size(1000, 400);

    println(Serial.list());
    myPort = new Serial(this, Serial.list()[0], 9600);
    myPort.bufferUntil('&apos;\n&apos;');
    background(0xff);
    font = createFont("Ariel", 12, true);

    output = createWriter("medicoes.txt");
}

void draw ()
{
    inByte = map(inByte, 0, 1023, 0, height);
    height_new = height - inByte;
    line(xPos - 1, height_old, xPos, height_new);
    height_old = height_new;

    if (xPos >= width)
    {
        xPos = 0;
        background(0xff);
    }
    else
    {
        xPos++;
    }

    if (millis() % 128 == 0)
    {
        fill(0xFF);
        rect(0, 0, 200, 20);
        fill(0x00);
        text("RR: " + inByte, 15, 10);
    }
}

void serialEvent (Serial myPort)

```

```

{
    String inString = myPort.readStringUntil('&apos;\n&apos;');

    if (inString != null)
    {
        inString = trim(inString);

        if (inString.equals("!"))
        {
            stroke(0, 0, 0xff);
            inByte = 512; //Flat line
        }
        else
        {
            stroke(0xff, 0, 0);
            inByte = float(inString);
            calculaRR();
        }
    }
}

void calculaRR ()
{
    int beat_new = millis();
    int diff = beat_new - beat_old;
    beats[beatIndex] = diff;
    println(diff);
    beat_old = beat_new;
    beatIndex = (beatIndex + 1) % 500;
}

void keyPressed()
{
    if(key == ENTER)
    {
        for(int i=0; i < 500;i++)
        {
            output.println(beats[i]);
        }
        output.flush();
        output.close();
        exit();
    }
}

```