

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E ELETRÔNICA
CURSO DE ENGENHARIA ELETRÔNICA

FELIPPE VANROO SILVEIRA

**ESTUDO COMPARATIVO DA EFICÁCIA DE
REMOÇÃO DE RUÍDO EM SINAIS DE
ELETROCARDIOGRAMA UTILIZANDO
DIFERENTES FILTROS DIGITAIS**

FLORIANÓPOLIS — SANTA CATARINA
2019

FELIPPE VANROO SILVEIRA

**ESTUDO COMPARATIVO DA EFICÁCIA DE
REMOÇÃO DE RUÍDO EM SINAIS DE
ELETROCARDIOGRAMA UTILIZANDO
DIFERENTES FILTROS DIGITAIS**

Trabalho de Conclusão de
Curso submetido ao Curso de
Engenharia Eletrônica da Uni-
versidade Federal de Santa
Catarina para obtenção do
Grau de Bacharel em Enge-
nharia Eletrônica

Orientador: Prof. Dr. Jeffer-
son Luiz Brum Marques

FLORIANÓPOLIS — SANTA CATARINA

2019

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Vanroo Silveira, Felipe

Estudo comparativo da eficácia de remoção de ruído
em sinais de eletrocardiograma utilizando
diferentes filtros digitais / Felipe Vanroo
Silveira ; orientador, Jefferson Luiz Brum Marques,
2019.

92 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro
Tecnológico, Graduação em Engenharia Eletrônica,
Florianópolis, 2019.

Inclui referências.

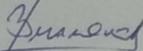
1. Engenharia Eletrônica. 2. Eletrocardiograma.
3. Processamento digital de sinais. 4. Filtro. 5.
Relação sinal-ruído. I. Luiz Brum Marques, Jefferson.
II. Universidade Federal de Santa Catarina.
Graduação em Engenharia Eletrônica. III. Título.

Felippe Vanroo Silveira

**ESTUDO COMPARATIVO DA EFICÁCIA DE REMOÇÃO DE
RUÍDO DE SINAIS DE ELETROCARDIOGRAMA
UTILIZANDO DIFERENTES FILTROS DIGITAIS**

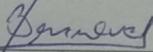
Este Trabalho foi julgado adequado para obtenção do Título de Bacharel
em Engenharia Eletrônica e aprovado em sua forma final pela Banca
Examinadora

Florianópolis, 11 de junho de 2019.

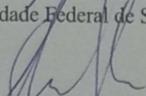


Prof. Jefferson Luiz Brum Marques, PhD
Coordenador do Curso

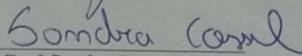
Banca Examinadora:



Prof. Jefferson Luiz Brum Marques, PhD
Orientador
Universidade Federal de Santa Catarina



Prof. Glauco Cardozo, MSc
Instituto Federal de Santa Catarina



Eng. Sandra Consul, MSc
Universidade Federal de Santa Catarina

Dedico este trabalho à minha mãe, pelo eterno apoio.

Agradecimentos

Ao Prof. Dr. Jefferson Luiz Brum Marques pelo apoio na construção deste projeto. Todas as críticas, sugestões e correções foram muito importantes durante este período atribulado do conhecido TCC.

Ao Prof. Dr. Márcio Holsbach Costa, por ter me inspirado nas aulas de Sistemas Lineares, tanto em relação a este projeto quanto em relação às demonstrações de como ser um bom professor e profissional.

Aos colegas de curso, em especial Lucas, Pedro, Fernando, Cíntia, Marcos, Samuel, Natan, assim como vários outros, pelos momentos de ajuda e descontração durante a graduação.

Aos amigos que fiz quando morei no Japão, Vizinho, Peludão, Ariel, Buxexa, Mateus, Baiano, PNC, Schwaiz, Barney, Daniel, Marcial e, em especial, nosso amigo Aurélio, que nos deixou antes do tempo.

À WEG, na figura do meu supervisor Diogo, que me acolheu durante 7 meses para realização do estágio. Todo conhecimento adquirido durante este tempo será sempre muito importante para minha vida profissional.

À minha noiva Daniela, pelo apoio e companheirismo que duram já quase 10 anos. Este apoio foi fundamental durante toda a graduação, principalmente no período de TCC. Obrigado por tudo.

Aos meus pais, pelo suporte financeiro e emocional. Sem isto, nunca seria possível realizar este projeto ou até mesmo poder concluir o curso de graduação.

A todos que de alguma forma contribuíram para a realização deste trabalho.

Part of the journey is the end

TONY STARK, 2019

Resumo

Os sinais biomédicos são sinais que carregam informações que tem como origem os seres vivos. A análise destes sinais é importante para diversas áreas na Medicina, como a obtenção de diagnósticos com rigor e clareza. Um dos sinais mais importantes é o eletrocardiograma, que reproduz a atividade elétrica do coração. Este tipo de informação, no entanto, geralmente está corrompida por ruídos, que podem ter origem tanto interna (próprio corpo, outros sinais biomédicos) quanto externa (aparelhos eletrônicos, rede elétrica). Assim, faz-se necessário o uso de processos para remoção das partes indesejadas do sinal. O método mais utilizado é o processamento digital de sinais, que manipula a informação utilizando algoritmos específicos. Este trabalho trata de uma das etapas do processamento digital, chamada de filtragem. Múltiplos sinais de eletrocardiograma, vindos de um banco de dados público, são, primeiramente, corrompidos por um ruído simulados da rede elétrica e do próprio corpo dos indivíduos. Em seguida, os sinais são processados por diferentes tipos de filtros, sendo comparadas suas performances utilizando métricas de relação sinal-ruído e tensão pico a pico. O filtro que apresentou o melhor desempenho foi o Equiripple, ocasionando perdas de aproximadamente 16,44% na relação sinal-ruído e 17,05% na tensão pico-a-pico, em média.

Palavras-chave: Eletrocardiograma. Processamento digital de sinais. Filtro. Relação sinal-ruído.

Abstract

BIOSIGNALS are signals that carry information that have living beings as the origin. Analysing these signals is important for many fields in Medicine, such as the obtention of diagnosis with clarity and accuracy. One of the most important biosignals is the electrocardiogram, since it shows the heart's electrical activity. This kind of information, however, is often corrupted by noises that originates internally (e.g., body movement, another biosignal) as well as externally (e.g., electronic devices, electrical grid). Therefore, it becomes necessary the use of specific procedures to remove the unwanted parts of the signal. The most used method is the digital signal processing, that manipulates signals applying specific algorithms. This paper illustrates the differences regarding several methods of filtering, one of the digital processing stages. Multiple electrocardiogram signals, from a public database, are firstly corrupted with simulated electrical grid and body movement noises. Posteriorly, these signals are processed by different kinds of filters, being compared with each other, using signal-to-noise ratio and peak-to-peak voltage metrics. The filter with the best performance was the Equiripple, causing 16,44% signal-to-noise ratio and 17,05% peak-to-peak voltage average losses.

Keywords: Electrocardiogram. Digital signal processing. Filter. Signal-to-noise ratio.

Lista de ilustrações

Figura 1 – Diagrama de blocos de um sistema padrão para aquisição e processamento de sinais biomédicos	26
Figura 2 – Sinal de ECG e sua origem, dividido em ciclos	30
Figura 3 – Representação de um sinal de ECG com seus pontos e intervalos de interesse	31
Figura 4 – Posicionamento dos eletrodos do plano horizontal para obtenção do sinal de ECG	32
Figura 5 – Posicionamento dos eletrodos do plano vertical para obtenção do sinal de ECG	33
Figura 6 – Disposição dos triângulos de Einthoven	34
Figura 7 – Faixa de frequências para métodos de aquisição do ECG	35
Figura 8 – Exemplo do processo de amostragem, com sinal original em cima e sinal amostrado embaixo	38
Figura 9 – Perda de informação ao utilizar f_s menor que o necessário, resultando em sinal amostrado alterado	38
Figura 10 – Sinal de EEG com ruído em 50Hz na esquerda e sinal amostrado em 80Hz com ruído em 30Hz na direita	39
Figura 11 – Comparação do sinal quantizado para diferentes valores de n	40
Figura 12 – Especificações relativas a um LPF não-ideal	44
Figura 13 – <i>Prompt</i> de comando do Cygwin	52
Figura 14 – <i>Prompt</i> de comando do Cygwin X	53
Figura 15 – Visualização do sinal de ECG no Cygwin X	53

Figura 16 – Gravação de um sinal de ECG da base de dados	55
Figura 17 – Comparação do sinal de ECG original e com adição de ruído	56
Figura 18 – Comparação do sinal de ECG original e fil- trado para o filtro FIR Equiripple	58
Figura 19 – Comparação do sinal de ECG original e fil- trado para o filtro FIR Kaiser	61
Figura 20 – Comparação do sinal de ECG original e fil- trado para o filtro FIR Bartlett	62
Figura 21 – Comparação do sinal de ECG original e fil- trado para o filtro FIR Hamming	62
Figura 22 – Comparação do sinal de ECG original e fil- trado para o filtro IIR Butterworth	63
Figura 23 – Comparação do sinal de ECG original e fil- trado para o filtro IIR Chebyshev I	63
Figura 24 – Comparação do sinal de ECG original e fil- trado para o filtro IIR Chebyshev II	64
Figura 25 – Comparação do sinal de ECG original e fil- trado para o filtro IIR Elíptico	64

Lista de tabelas

Tabela 1 – Diferenças entre os tipos de filtro FIR e IIR .	43
Tabela 2 – Tipos de filtros escolhidos	57
Tabela 3 – Valores de SNR e V_{pp} do sinal original e dos sinais filtrados	65
Tabela 4 – Perdas dos ECGs filtrados em relação ao ori- ginal, em porcentagem	65

Lista de abreviaturas e siglas

A/D	Analógico/Digital
CC	Corrente Contínua
dB	Decibéis
DFT	<i>Discrete Fourier Transform</i> – Transformada de Fourier Discreta
ECG	Eletrocardiograma
EEG	Eletroencefalograma
EMG	Eletromiograma
EQM	Erro Quadrático Médio
f_c	Frequência de corte
FC	Frequência cardíaca
FFT	<i>Fast Fourier Transform</i> – Transformada Rápida de Fourier
FIR	<i>Finite Impulse Response</i> – Resposta ao Impulso Finita
FPA	Filtro Passa-Alta
FPB	Filtro Passa-Baixa
FPF	Filtro Passa-Faixa

FRF	Filtro Rejeita-Faixa
f_s	Frequência de amostragem
Hz	Hertz
IIR	<i>Infinite Impulse Response</i> – Resposta ao Impulso Infinita
LMS	<i>Least Mean Square</i> – Mínimo Quadrado Médio
m	Mili
PDS	Processamento Digital de Sinais
s	Segundos
SNR	<i>Signal to Noise Ratio</i> – Relação sinal-ruído
STFT	<i>Short Time Fourier Transform</i> – Transformada de Fourier de Curto Tempo
T	Tempo
T_s	Intervalo de amostragem
V	Volts
V_{pp}	Tensão Pico a Pico

Lista de símbolos

δ	Letra grega delta
Δ	Letra grega Delta
ω	Letra grega omega
σ	Letra grega sigma

Sumário

1	INTRODUÇÃO	25
1.1	OBJETIVOS	27
1.1.1	Objetivo geral	27
1.1.2	Objetivos específicos	27
2	FUNDAMENTAÇÃO TEÓRICA	29
2.1	ELETROCARDIOGRAMA	29
2.2	PROCESSAMENTO DIGITAL DE SINAIS	36
2.2.1	Amostragem	37
2.2.2	Quantização	39
2.2.3	Filtros Digitais	41
2.3	TRABALHOS RELACIONADOS	45
3	METODOLOGIA	47
3.1	PLATAFORMA PHYSIONET	47
3.2	BANCO DE DADOS	48
3.3	CYGWIN	51
3.4	PROCESSAMENTO NO MATLAB	54
3.4.1	Transformação dos arquivos	54
3.4.2	Adição de ruídos	55
3.4.3	Geração do sinal de ECG ruidoso	56
3.4.4	Criação e aplicação dos filtros	57
3.4.5	Escolha dos parâmetros de avaliação	58
4	RESULTADOS	61
5	DISCUSSÃO DOS RESULTADOS	67

6	CONCLUSÃO	69
6.1	SUGESTÕES PARA TRABALHOS FUTUROS	70
	REFERÊNCIAS	71
	ANEXO A – CÓDIGO DA FUNÇÃO DO MA- TLAB <i>WFDB2MAT</i>	75
	ANEXO B – CÓDIGO DA FUNÇÃO DO MA- TLAB <i>RDMAT</i>	77
	ANEXO C – EXEMPLO DE FUNÇÃO JA- NELA	81
	APÊNDICE A – CÓDIGO DO MATLAB PARA PROCESSAMENTO DOS SINAIS, CRIAÇÃO DOS GRÁ- FICOS E CÁLCULO DOS PARÂMETROS	83

1 INTRODUÇÃO

A Engenharia Biomédica é a área da Engenharia que aplica princípios de elétrica, eletrônica, mecânica e química para entender e controlar sistemas biológicos, fechando a lacuna existente entre tecnologia e a área da Medicina. Dentre as atividades que desempenham o profissional da função, podem ser listadas a criação de próteses, modelagem de sistemas do corpo humano, instrumentação para medicina esportiva e elaboração de *software* para pesquisa e análise de dados médicos, entre outros.

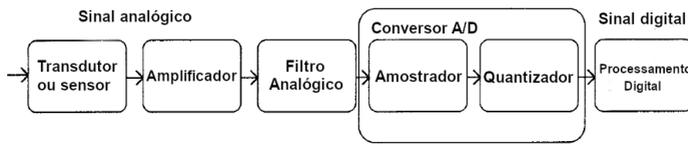
Uma das áreas mais importantes da medicina, que trata de diagnósticos, utiliza os conhecimentos advindos da Engenharia Biomédica em duas frentes: *hardware* e *software*. Em relação ao *hardware*, são criados equipamentos para aquisição de sinais biomédicos, assim como para análise de exames utilizando os sinais obtidos. Para a parte de *software*, sistemas de processamento de imagens e sinais são desenvolvidos para maior eficácia em tomadas de decisão com relação à diagnósticos.

Os sinais biomédicos carregam informações relativas à compreensão de mecanismos vivos, que servem principalmente para extrair dados biológicos de sistemas que estão sendo investigados ou analisados (COHEN, 2015; CERUTTI; BIANCHI; MAINARDI, 2015). Estes sinais podem ser classificados de acordo com a sua origem, que pode ser bioacústica, bioelétrica, biomecânica, biomagnética (COHEN, 2015).

A aquisição destes sinais dá-se por meio de sensores,

que estão presentes em sistemas de medições diversos. Além disso, também há a presença, nestes sistemas, de transdutores para transformação da grandeza medida em um sinal elétrico. Este sinal, que geralmente é analógico, passa por outras etapas antes de ser possível realizar uma análise sendo elas amplificação, filtragem, conversão analógico/digital (A/D) e processamento digital. Estas etapas estão ilustradas na Figura 1.

Figura 1 – Diagrama de blocos de um sistema padrão para aquisição e processamento de sinais biomédicos



Fonte: Adaptado de [CERUTTI; BIANCHI; MAINARDI, 2015](#)

Uma das etapas, o processamento digital de sinais (PDS ou DSP - *Digital Signal Processing*), manipula e/ou transforma os sinais analógicos que foram convertidos em digitais para uma melhor qualidade de análise. Este processo faz-se necessário porque os sinais biomédicos sofrem limitações vindas de várias fontes de ruído devido à sua característica de possuir baixas amplitudes. Além disso, dependendo da origem biológica da medição, o sinal a ser analisado pode sofrer influência proveniente de outro lugar do próprio corpo, como uma medida de eletroencefalograma (EEG) sofrer interferência da medida de um eletromiograma (EMG) ([NAJARIAN; SPLINTER, 2012](#)).

Desta maneira, utilizando o processamento de sinais, diminui-se o ruído presente no sistema para ser possível obter

informações com maior rigor, assim como extraí-las para a análise desejada. As ferramentas de PDS, no entanto, demandam certo conhecimento para utilização correta, tanto para a escolha do *software* adequado quanto para a correta escolha dos filtros e algoritmos para cada tipo de sinal (SMITH, 1999).

Este trabalho foca em um estudo comparativo do desempenho de diferentes tipos de filtros para remoção de ruídos em um tipo específico de sinal biomédico, o eletrocardiograma (ECG), utilizando técnicas de PDS. As amostras foram corrompidas com ruídos provenientes de duas fontes encontradas normalmente em análise de sinais: ruído de base, com origem na interação do sistema de medição com o paciente; e ruído da rede, com origem na oscilação natural do sinal da rede elétrica.

1.1 OBJETIVOS

1.1.1 Objetivo geral

O objetivo deste trabalho é testar a performance de diferentes tipos de filtros em um sinal de ECG corrompido com ruídos e escolher qual elimina as distorções de forma mais eficaz.

1.1.2 Objetivos específicos

- Aprofundamento dos conhecimentos relativos à sinais biomédicos, especialmente sobre o ECG;
- Realizar filtragem em sinais de ECG, desenvolvendo os algoritmos para tal;
- Escolher os tipos de filtros a serem utilizados para processamento do sinal de ECG;

- Verificar e escolher os parâmetros de comparação de melhor filtragem para cada tipo de processamento e ruído, em especial aqueles relativos aos sinais de ECG;
- Comparar o desempenho dos sinais resultantes, após filtragem, em relação ao tipo de filtro utilizado, a partir das métricas escolhidas.

2 FUNDAMENTAÇÃO TEÓ- RICA

2.1 ELETROCARDIOGRAMA

O eletrocardiograma é a manifestação elétrica da atividade contrativa do coração durante certo período de tempo (RAJNI; KAUR, 2013), podendo ser mensurada e gravada utilizando eletrodos de superfície. Estes movimentos do coração podem ser divididos em ciclos, em que cada um representa uma movimentação de diferentes partes do coração, como pode ser visto na Figura 2.

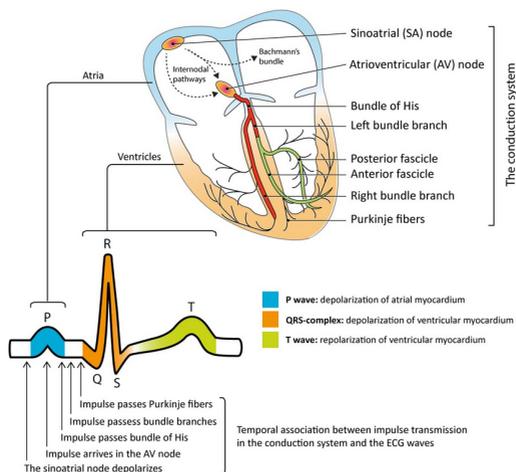
Os ciclos do sinal de ECG são comumente decompostos em três partes:

- **Onda P:** deflexão que representa a despolarização atrial;
- **Complexo QRS:** junção de três ondas, representando a despolarização ventricular;
- **Onda T:** representa a repolarização ventricular.

Há ainda pontos que não são tão utilizados quanto os elencados acima, mas também possuem importância para diagnósticos de certas doenças ou condições:

- **Ponto J:** a abnormalidade deste ponto pode ser utilizada para diagnóstico de doenças como pericardite e hipotermia;
- **Onda U:** presente mesmo em corações saudáveis, variações fora do normal em seu padrão são indicativo de condições como hipertensão e hipertireoidismo.

Figura 2 – Sinal de ECG e sua origem, dividido em ciclos



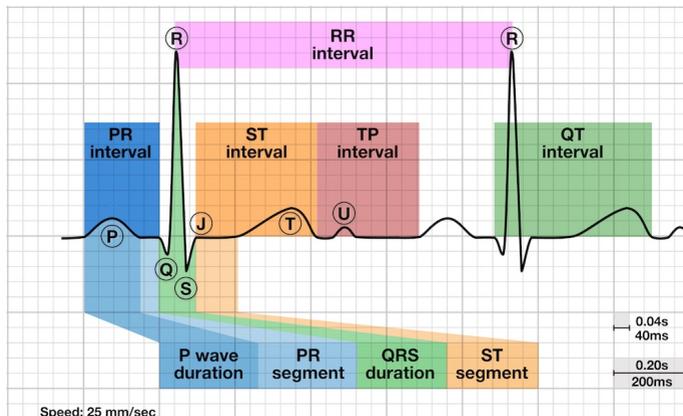
Fonte: RAWSHANI, 2018, disponível em <https://ecgwaves.com/introduction-electrocardiography-ecg-book/>. Acesso em 15/10/2018

A representação de uma onda normal presente em um exame de eletrocardiograma é mostrada na Figura 3. Como pode ser visto, além da divisão dos pontos e complexos, é comum destacar no sinal de ECG certos intervalos, com o propósito de, novamente, auxiliar na obtenção de diagnósticos. Existem

valores destes intervalos que podem ser considerados dentro da normalidade para indicar um exame de uma pessoa saudável (CHRISTENSEN, 2014):

- **Intervalo PR:** tempo de condução atrioventricular, durando de 120 a 200 milissegundos (ms);
- **Intervalo QRS:** tempo de despolarização ventricular, durando de 80 a 120ms;
- **Intervalo QT:** tempo de despolarização e repolarização ventricular, durando de 350 a 460ms;
- **Segmento ST:** intervalo entre a despolarização ventricular e a sua repolarização, durando de 80 a 120ms.

Figura 3 – Representação de um sinal de ECG com seus pontos e intervalos de interesse

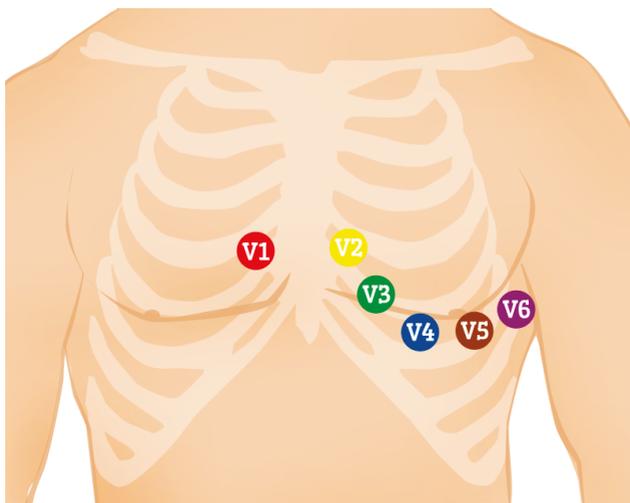


Fonte: BURNS, 2018, disponível em <https://litfl.com/qt-interval-ecg-library/>. Acesso em 15/10/2018

O dado mais utilizado no dia-a-dia, também mostrado na Figura 3, é o intervalo RR, que é a diferença de tempo entre dois picos R consecutivos. Este parâmetro serve para determinar a frequência cardíaca (FC) aproximada do indivíduo, onde o tempo (T) do intervalo RR é medido em segundos e utilizado na fórmula $FC = 60/T$.

A aquisição dos sinais de ECG é feita, normalmente, por uma técnica que utiliza 10 eletrodos e 12 *leads* (ligações) para formar o sinal resultante. O posicionamento dos eletrodos é dividido em duas frentes: plano horizontal e vertical. Os eletrodos do plano horizontal, que são 6 e unipolares, devem ser posicionados no peito do paciente de acordo com as posições da Figura 4.

Figura 4 – Posicionamento dos eletrodos do plano horizontal para obtenção do sinal de ECG

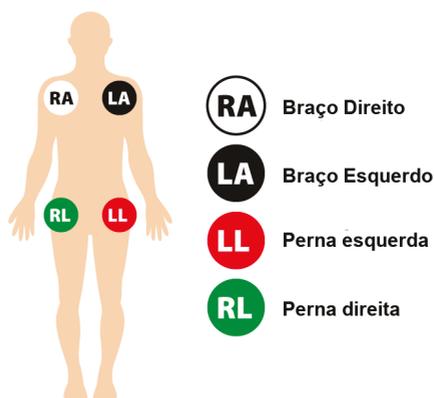


Fonte: Disponível em <https://www.cablesandsensors.eu/pages/12-lead-ecg-placement-guide-with-illustrations>. Acesso em 15/10/2018

Já para os 4 eletrodos do plano vertical, o seu posicionamento é feito nos braços (entre os ombros e os cotovelos) e nas pernas (entre o torso e os tornozelos), como mostra a Figura 5. O princípio de medição é baseado no triângulo de Einthoven, conforme a Figura 6, que é formado pelos dois braços e pela perna esquerda (neste caso, o eletrodo da perna direita serve para reduzir a interferência). O modo de medição é dividido em duas maneiras:

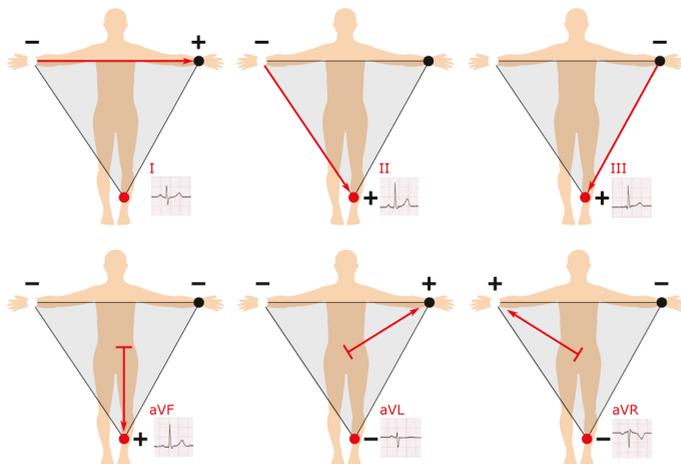
1. **Leads I, II e III**: medição pela diferença de potencial entre dois eletrodos;
2. **Leads aVR, aVL e aVF**: medição pela diferença de potencial entre um eletrodo e o coração como polo negativo.

Figura 5 – Posicionamento dos eletrodos do plano vertical para obtenção do sinal de ECG



Fonte: Adaptado de <https://www.cablesandsensors.eu/pages/12-lead-ecg-placement-guide-with-illustrations>. Acesso em 16/10/2018

Figura 6 – Disposição dos triângulos de Einthoven



Fonte: Disponível em <https://www.cablesandsensors.eu/pages/12-lead-ecg-placement-guide-with-illustrations>. Acesso em 16/10/2018

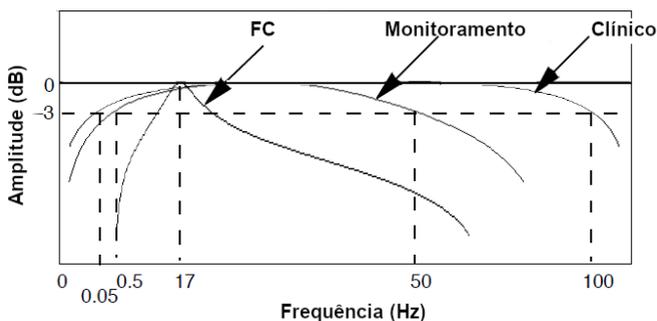
O sinal de ECG possui, como característica física, uma faixa dinâmica de amplitude entre 1 e 10 milivolts (mV) (COHEN, 2015). A largura de banda ocupada pelo sinal depende de sua aplicação (TOMPKINS, 2000):

- Para aplicações clínicas, é comum utilizar uma banda de 0,05 até 100 Hz, como o exame que utiliza 12 *leads*, que foi explicado previamente;
- Utilizando banda de 0,5 até 50 Hz, o sinal é adquirido com o objetivo de monitorar pacientes em ambulatórios;

- Outra faixa de frequência bastante utilizada está centrada em torno de 17 Hz, que elimina as ondas P e T, aumentando a relação sinal-ruído (SNR) do complexo QRS para detectar a FC;
- Algumas aplicações, que requerem medições de eventos específicos, como a ocorrência de potenciais tardios, podem utilizar uma banda com frequência máxima de até 500 Hz.

As faixas de frequência para algumas das situações citadas estão expostas na Figura 7. A vantagem das limitações de banda para as aplicações de ECG é a atenuação de ruídos advindos da tanto da própria aquisição do sinal (eletrodos de superfície) quanto da rede elétrica (50 ou 60Hz, dependendo do país).

Figura 7 – Faixa de frequências para métodos de aquisição do ECG



Fonte: Adaptado de [TOMPKINS, 2000](#)

2.2 PROCESSAMENTO DIGITAL DE SINAIS

Como já explicado, os sinais biomédicos possuem baixa amplitude e, por causa disto, sofrem muita influência de ruídos, que podem vir de várias fontes, tanto internas como externas. Assim, faz-se necessário um processamento com o objetivo de realçar a informação realmente relevante e extrair os parâmetros necessários corretamente, os quais serão utilizados para terapias, diagnósticos ou estudos acadêmicos.

Um processamento pode envolver: amplificação, filtragem, cálculo de média, estimação, entre outros. Existem duas maneiras de realizar estes tipos de procedimentos: analógico ou digital. O avanço recorrente das inovações tecnológicas relacionadas à computação faz o processamento digital ser mais flexível que o analógico. A performance do tipo digital geralmente é mais poderosa por ser mais facilmente implementável, não ser afetada por fatores como temperatura e envelhecimento de componentes e, além disso, uma modificação ou ajuste de parâmetros é feita de maneira mais simples, porque envolve uma mudança em *software* e não em *hardware*.

Assim, fica claro que utilizar as ferramentas digitais como bloco principal de um projeto é mais vantajoso. Para ser possível realizar o PDS, é necessária a execução de certos passos, como mostra a Figura 1. Primeiramente, nas etapas analógicas, a aquisição é feita por sensores, em sinais elétricos, enquanto que sinais não-elétricos são transformados em elétricos por um transdutor. Em seguida, um bloco de amplificação analógica é utilizado com o propósito de atender as especificações do *hardware* que fará o PDS.

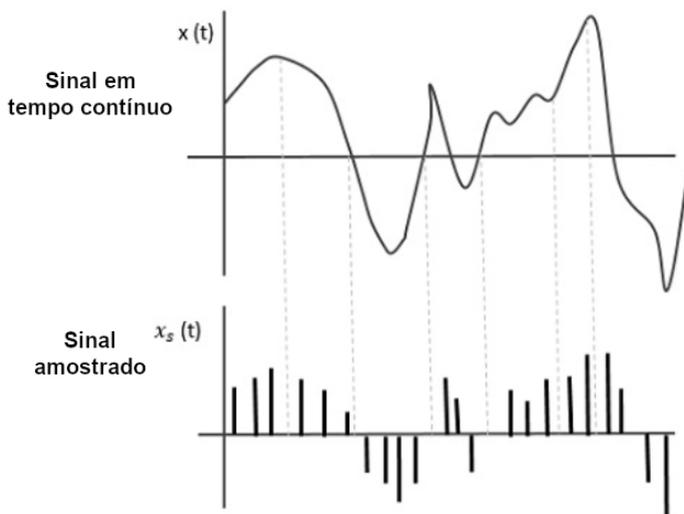
Em seguida, um filtro analógico é utilizado para, além de também atender as especificações do *hardware* de PDS, reduzir a quantidade de ruído do sinal e limitar sua banda, já que um sinal ilimitado em banda dificilmente é encontrado na prática. Como o sinal de ECG é do tipo contínuo, deve-se fazer este processo antes da conversão A/D com o propósito de evitar o efeito do *aliasing*, que pode surgir posteriormente. Por causa deste efeito indesejado, pode ocorrer uma sobreposição de informações, distorcendo o sinal original e, conseqüentemente, ocasionar uma perda de informações.

2.2.1 Amostragem

A primeira etapa em uma conversão A/D chama-se amostragem. Aplica-se este processo em sinais contínuos para serem transformados em sinais discretos, mantendo os valores de amplitude, de acordo com o intervalo de amostragem (T_s) que é escolhido. A seleção da taxa de amostragem deve levar em conta a de largura de banda imposta anteriormente pela filtragem analógica, assim como a frequência máxima do sinal. O processo está demonstrado na Figura 8.

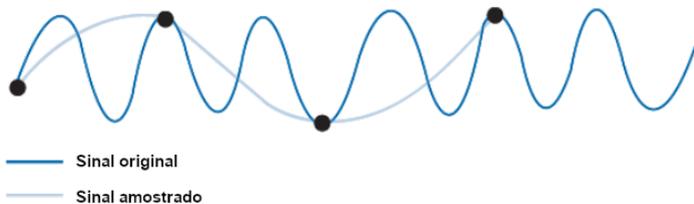
Para ser possível utilizar e, posteriormente, recuperar a informação original contida no sinal adquirido, é necessário que a taxa de amostragem (f_s) seja, no mínimo, maior que o dobro da frequência máxima do sinal a ser realizada a conversão A/D, de acordo com o Teorema da Amostragem de Nyquist-Shannon (SHANNON, 1949). Se a f_s escolhida for menor que o valor necessário, ocorre o fenômeno chamado de *undersampling*, em que informações importantes são perdidas pela ocorrência do *aliasing*, como exemplificado na Figura 9.

Figura 8 – Exemplo do processo de amostragem, com sinal original em cima e sinal amostrado embaixo



Fonte: Adaptado de https://www.tutorialspoint.com/digital_communication/
Acesso em 18/10/2018

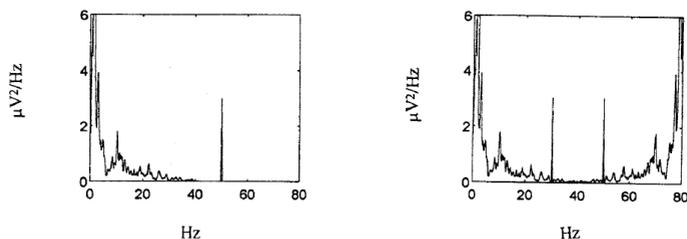
Figura 9 – Perda de informação ao utilizar f_s menor que o necessário, resultando em sinal amostrado alterado



Fonte: Adaptado de <http://www.ni.com/newsletter/50078/en/>
Acesso em 18/10/2018

Um exemplo em que se pode perceber a importância tanto da filtragem analógica quanto a escolha correta da f_s , utilizando uma situação real, está presente na Figura 10. Neste caso, tem-se um sinal de EEG com frequências de interesse de 0 a 40Hz, assim como um ruído em 50Hz (lado esquerdo da Figura). Se for realizada uma amostragem com f_s de 80Hz sem a filtragem analógica, o ruído estará presente na frequência de 30Hz do sinal amostrado (lado direito da Figura), comprometendo a leitura da banda de diagnóstico das ondas β do EEG .

Figura 10 – Sinal de EEG com ruído em 50Hz na esquerda e sinal amostrado em 80Hz com ruído em 30Hz na direita



Fonte: [CERUTTI; BIANCHI; MAINARDI, 2015](#)

Para combater o efeito do *aliasing*, sugere-se utilizar uma f_s que seja pelo menos 4 a 5 vezes a frequência máxima do sinal a ser analisado ([SEMMLOW, 2017](#)). Isto nem sempre é possível, no entanto, devido às limitações de *hardware* que é usado para fazer o PDS.

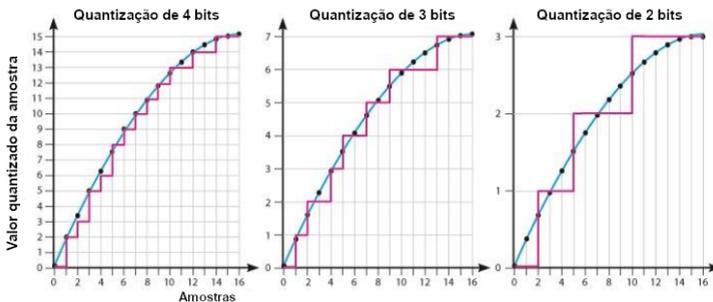
2.2.2 Quantização

A quantização é o próximo passo da conversão A/D, no qual as amplitudes das amostras são discretizadas para um

conjunto finito de valores. Estes possíveis valores são chamados de níveis de quantização, que são definidos de acordo com o intervalo do sinal que passou pela amostragem. O número de níveis disponíveis para a quantização irá depender do *hardware* utilizado, que terá uma limitação descrita no *datasheet* referente ao número de *bits* para conversão A/D.

Os conversores A/D que possuem n *bits*, terá 2^n valores possíveis. Assim, se o conversor tiver 4 *bits*, terá $2^4 = 16$ valores ou níveis possíveis; se tiver 8 *bits* terá $2^8 = 256$ níveis possíveis. Como pode-se perceber, quanto maior a resolução do conversor, maior será o número de valores que o sinal poderá assumir, aumentando assim a exatidão do sinal quantizado. Um exemplo de quantização comparando o efeito de diferentes valores de n *bits* é mostrado na Figura 11.

Figura 11 – Comparação do sinal quantizado para diferentes valores de n



Fonte: Adaptado de <https://airfreshener.club/quotes/analogue-digital-sampling.html>
Acesso em 18/10/2018

O efeito indesejado que este processo traz é chamado de

ruído de quantização. Este erro é influenciado pela diferença do valor real do sinal e seu valor quantizado. De acordo com (CERUTTI; BIANCHI; MAINARDI, 2015), a fórmula geral utilizada para estimar a SNR levando em conta o erro de quantização, em função do número de *bits* n , que é

$$\text{SNR} = 6,02n + 10,79 + 10 \log_{10}(\sigma_x^2) \quad (1)$$

onde σ_x^2 é a variância do sinal quantizado. Assim, de acordo com a fórmula, a SNR aumenta quanto maior é o número de *bits*. Isto é facilmente verificável, visto que, quanto mais níveis de quantização existirem, menores serão as diferenças entre os valores quantizados para os valores originais.

2.2.3 Filtros Digitais

Filtragem é um processo que tem o objetivo de separar ou restaurar sinais. A função de separar sinais é utilizada quando há alguma degradação por interferência ou ruídos, como o caso do EEG sofrer interferência do EMG. Já a função de restaurar sinais surge quando há alguma distorção, como no caso de uma fotografia ser tirada com o foco errado, por exemplo.

Apesar de possuírem os erros e ruídos mencionados anteriormente, os filtros digitais possuem vantagens sobre os analógicos:

- A ordem pode ser aumentada para valores que seriam impraticáveis com filtros analógicos, pois nestes filtros o espaço físico aumentaria demasiadamente;

- São feitos utilizando *software*, sendo mais facilmente projetados, implementados, testados e modificados;
- Não estão sujeitos à variações externas, como mudanças de temperatura;
- Tanto os dados filtrados quanto os não-filtrados podem ser armazenados, sem perigo de perda de dados;
- A performance não é modificada de um processo para o outro;
- O mesmo filtro pode ser utilizado para várias entradas, sem necessidade de aumento de *hardware*;
- Características de resposta em frequência ou em amplitude não possíveis em filtros analógicos, como transições mais agudas na banda de transição;
- Trabalham bem com sinais de baixa frequência, como certas bandas de diagnóstico de EEG. Além disso, estão expandindo para áreas que eram dominadas por filtros analógicos, como faixas de radiofrequência;
- Podem auto ajustar de acordo com a necessidade em relação ao tempo (filtros adaptativos).

Uma divisão comumente utilizada para caracterizar os filtros digitais é relacionado a sua resposta ao impulso. Existem dois tipos de filtro nessa classificação: resposta finita ao impulso (FIR) ou resposta infinita ao impulso (IIR). Cada tipo possui características que fazem seu uso mais indicado para certos casos ou aplicações, como mostra a Tabela 1.

Tabela 1 – Diferenças entre os tipos de filtro FIR e IIR

	FIR	IIR
Implementação	Convolução (saída atual depende somente das entradas)	Recursividade (saída atual depende tanto das entradas quanto das saídas anteriores)
Estabilidade	Sempre são estáveis	Podem ser instáveis
Ordem	Número de coeficientes; maior que a ordem de um filtro IIR com mesmo desempenho	Número de saídas anteriores presentes na atual; menor que filtros FIR para o mesmo desempenho
Fase	Sempre são estáveis	Podem ser instáveis
Histórico	Não possuem ligação com filtros analógicos	Derivam de filtros analógicos
Atraso de grupo	Igual para todas as frequências	Depende da frequência

Fonte: Autor, 2019

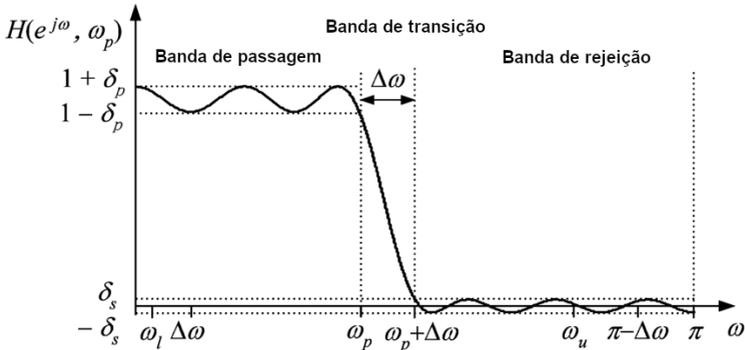
Independente do tipo de filtro escolhido, é necessário projetá-lo de acordo com as especificações requeridas no projeto. Primeiramente, é determinado qual tipo de filtro é necessário, havendo quatro principais: passa-baixa (FPB), passa-alta (FPA), rejeita-faixa (FRF) e passa-faixa (FPF). Os dois primeiros tipos possuem uma frequência de corte (f_c) que permitem a passagem de frequências abaixo (FPB) e acima (FPA) da f_c ; já os outros dois ou rejeitam (FRF) ou permitem (FPF) uma faixa de frequência

entre duas f_c .

Em seguida, devem ser escolhidas as especificações práticas relativas à frequências e amplitudes limites de filtros não-ideais, como mostrado na Figura 12. As especificações comumente utilizadas são:

- δ_p – Máxima magnitude do *ripple* da banda de passagem;
- δ_s – Máxima magnitude do *ripple* da banda de rejeição;
- ω_p – Limite de frequência da banda de passagem;
- $\omega_p + \Delta\omega$ – Limite de frequência da banda de rejeição.

Figura 12 – Especificações relativas a um LPF não-ideal



Fonte: Adaptado de YU; SHI; LIM, 2009

Há também a f_c do filtro que, em projetos analógicos, é o ponto onde a amplitude é reduzida para 70,7% de seu valor

original. Em filtros digitais, no entanto, não há uma padronização deste valor, com valores de 50% até 99% de redução no ponto da f_c sendo observados (SMITH, 1999). Outros parâmetros utilizados são A_p e A_s , que derivam de δ_p e δ_s . Esses valores de amplitude são mensurados em decibéis (dB), com expressões mostradas nas Equações 2 e 3:

$$A_p = 20 \log_{10} (1 + \delta_p) \quad (2)$$

$$A_s = -20 \log_{10} (\delta_s) \quad (3)$$

As não-idealidades descritas surgem devido à impossibilidade do ganho ser unitário (na banda de passagem) ou zero (na banda de rejeição), assim como ser impossível obter uma transição instantânea entre as bandas do filtro.

2.3 TRABALHOS RELACIONADOS

Em RAJNI; KAUR (2013), é descrita uma visão geral sobre análises e processamentos em sinais de ECG. O autor primeiramente descreve o sinal fisicamente, assim como caracteriza algumas abnormalidades de acordo com certos intervalos e características do sinal. Em seguida, alguns métodos existentes de processamentos de sinais são explicados, como a transformada de Fourier rápida (FFT)/curta (STFT) e a transformada Wavelet, assim como as vantagens e desvantagens de cada método.

O estudo descrito por AHAMMED (2018) trata da remoção de ruídos adicionado a um sinal ECG, mais especificamente a remoção de uma linha de base de 0,5Hz e um ruído da rede elétrica de 50Hz. O metodologia utilizada consiste em uma comparação entre a efetividade de utilizar um filtro *notch* (um tipo

específico de FRF) e um filtro adaptativo do tipo Mínimo Quadrado Médio (LMS). Utilizando SNR e Erro Quadrático Médio (EQM) como métricas de desempenho, chegou-se a conclusão de que o filtro LMS remove o ruído com mais eficácia.

No trabalho desenvolvido por BHOGESHWAR; SONI; BANSAL (2014) foram utilizados várias técnicas de filtragem para remoção de três tipos de ruídos, aleatório, ruído branco e interferência em 50Hz, com a ajuda da *toolbox* Simulink presente no *software* MATLAB. O autor confere a efetividade das técnicas de filtragem de acordo com os tipos de filtros FIR e IIR, utilizando SNR como método de comparação. Todos os filtros utilizados foram do tipo LPF com f_c de 100Hz e f_s de 1000Hz.

Seguindo a mesma linha do artigo citado acima está o trabalho de SAXENA et al. (2018), que analisa a performance de diferentes filtros FIR e IIR removendo o ruído de 50Hz sinais de ECG. A ferramenta utilizada para remoção do ruído é novamente o MATLAB, porém sem auxílio do Simulink, sendo implementado com um algoritmo de LMS.

3 METODOLOGIA

3.1 PLATAFORMA PHYSIONET

PhysioNet (GOLDBERGER et al., 2000) é um site que acumula três plataformas distintas:

- **PhysioToolkit:** uma biblioteca de técnicas de processamento digital implementada em código aberto. A plataforma inclui *software* para análises de sinais biomédicos, assim como cria novas bases de dados e implementa modos de exibição e caracterização de sinais de maneira interativa;
- **PhysioBank:** uma coleção de bancos de dados de sinais biomédicos para utilização em pesquisas. Os sinais arquivados foram coletados tanto de pessoas saudáveis quanto de pessoas com algum tipo de doença ou condição especial de saúde, como epilepsia, parada cardíaca, infarto do miocárdio, entre outros. Outros tipos de dados coletados têm como fonte experimentos *in vitro* e *in vivo*, assim como algoritmos de origem psicológica;
- **PhysioNet:** com o mesmo nome do site, esta plataforma proporciona um link dinâmico entre os dados e a comunidade acadêmica utilizando um fórum *online* para a disseminação de sinais biomédicos e *software* relacionados. Além disso, a plataforma exerce a função

de ser um lugar comum para análise de dados, que podem ser discutidos, avaliados, revistos e examinados em detalhes pela comunidade. Outras funcionalidades incluem tutoriais para utilização e aproveitamento do que está disponível no site, assim como a correção de possíveis erros em projetos e implementações de algoritmos relacionados a resultados discrepantes.

Neste trabalho, as 3 plataformas foram utilizadas. Primeiramente, PhysioBank foi consultado com o objetivo de analisar os bancos de dados de sinais disponíveis para análise e, posteriormente, escolher um deles que incluía sinais de ECG adequados para a pesquisa desenvolvida.

Em seguida, a plataforma PhysioToolkit foi utilizada para baixar os *software* necessários para a análise dos sinais de ECG, assim como a utilização de *toolkits* para utilização em *software* externos, auxiliando nos procedimentos para processamento dos sinais.

Por fim, a plataforma PhysioNet foi utilizada como um local para tirar dúvidas sobre a maneira de manipular corretamente os sinais, que por vezes estão em extensões não-convencionais e portanto, necessitam de algum tipo de manipulação para serem lidos e utilizados no processamento.

3.2 BANCO DE DADOS

Para a realização deste trabalho, o banco de dados escolhido, dentre os inúmeros incluídos na plataforma PhysioBank, chama-se “*Data for development and evaluation of ECG-based apnea detectors*” (PENZEL et al., 2000). Este conjunto de dados

foi escolhido pela praticidade de ter os sinais de 12 *leads* unidos no mesmo arquivo, formando o sinal de ECG conhecido da Figura 3. Desta maneira, não é necessário fazer o processamento em 12 arquivos de uma mesma amostra, mas sim em apenas 1 arquivo.

Este banco de dado inclui 70 sinais de ECG durante o sono dos indivíduos do estudo, assim como seus dados respiratórios. Para este trabalho, no entanto, serão utilizados somente os sinais de ECG para análise e processamento. As gravações foram obtidas em dois períodos de tempo diferentes: o primeiro grupo de dados foi colhido entre 1993 e 1995 de 9 pessoas em 3 dias diferentes, resultando em 27 gravações; o segundo foi colhido entre 1998 e 1999, com 23 pessoas e 43 gravações.

As 70 gravações colhidas tem duração entre 401 e 578 minutos, foram amostradas com uma f_s de 100Hz e quantizadas utilizando 16 *bits* de resolução. Para este tipo de estudo, de acordo com (ZIEMSEN; GASCH; RUEDIGER, 2008), este valor de f_s não exerce influência significativa com relação à perda de dados. Além disso, de acordo com (TURCHIELLO, 2014), valores aceitáveis para a quantização de um sinal de ECG estão na faixa de 10 a 12 *bits*.

Os indivíduos e gravações foram divididos em três grupos, de acordo com a influência da apneia em relação ao seu sono:

- **Classe A:** mais de 100 minutos de episódios de desordem respiratória, contendo 40 gravações de 15 homens e 1 mulher, com idade média de 50 anos (de 29 até 63 anos);
- **Classe B:** entre 5 e 100 minutos de episódios de desordem respiratória, contendo 10 gravações de 4

homens e 1 mulher, com idade média de 46 anos (de 39 até 53 anos);

- **Classe C:** menos de 5 minutos de episódios de disordem respiratória, contendo 20 gravações de de 6 homens e 5 mulheres, com idade média de 33 anos (de 27 até 42 anos).

Para este trabalho, foram escolhidos aleatoriamente 4 sinais da classe A e 2 sinais das classes B e C, totalizando 8 sinais. Todas as informações das gravações dos indivíduos, inclusive sobre as classes, podem ser consultadas pelo arquivo *additional-information.txt* presente na base de dados. Nele, há uma tabela que relaciona o nome dos arquivos com as características da gravação.

Cada gravação produziu mais de um arquivo, que tem suas características dependendo da extensão¹:

- *.apn*: arquivos de anotação binários, contendo observações para cada minuto de cada gravação indicando a presença ou não de apneia naquele momento
- *.dat*: arquivos que contêm os sinais de ECG digitalizados, com as características de gravação descritas anteriormente, sendo os arquivos presentes nos bancos de dados da PhysioBank;
- *.hea*: arquivos de cabeçalho que contem informações que especificam o nome e formato dos arquivos asso-

¹ Estes são apenas alguns tipos de arquivos que podem aparecer. Outros bancos de dados podem possuir arquivos com extensões diferentes.

ciados aos sinais, sendo necessários para o funcionamento correto dos *software* que PhysioBank disponibiliza;

- *.qrs*: arquivos binários gerados automaticamente, contendo a localização dos complexos QRS nos arquivos *.dat*;
- *.xws*: arquivos que possibilitam a visualização das formas de ondas sem fazer *download*, a partir de *software* disponíveis na PhysioToolkit.

Existem, também, arquivos extras para certas gravações: aqueles com nome de final *r* possuem os sinais de referentes à respiração dos indivíduos, separados por canal; e aqueles com nome de final *er* possuem os sinais de respiração juntamente com o sinal de ECG para uma visualização conjunta.

3.3 CYGWIN

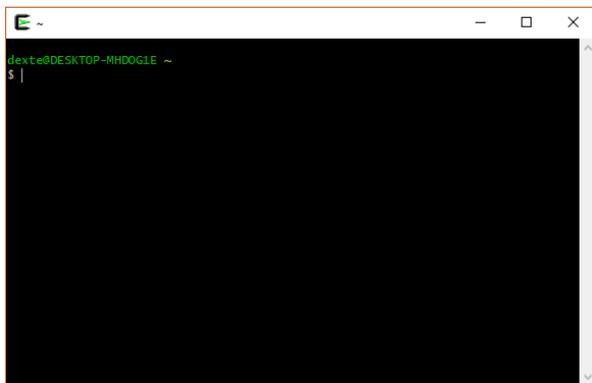
Cygwin é um software livre que fornece um ambiente inspirado em Unix, assim como uma gama de *software* baseados em Unix. Utilizando uma biblioteca de emulação de sistema operacional, Cygwin utiliza linhas de comando para conectar o usuário aos bancos de dados presentes na plataforma PhysioBank, assim como às bibliotecas presentes em PhysioToolkit.

Instalando o programa e os pacotes necessários, de acordo com as instruções da PhysioNet, ao iniciar o programa, é aberto um ambiente de linha de comando, como mostra a Figura 13. Este é o ambiente inicial já preparado para receber os comandos para estudo dos sinais biomédicos.

Algumas das várias opções incluem traçar os gráficos disponíveis para o banco de dados escolhido, mostrar as amostras e suas anotações em formato de texto, exportar os arquivos com extensões *.csv* e *.dat*, entre outras escolhas.

Os comandos que podem ser utilizados estão disponíveis para consulta na PhysioNet. Um exemplo de utilização do terminal seria a visualização da quantidade de picos R, assim como os momentos em que eles aparecem no arquivo “a01”, digitando o comando *sumann -r apnea-ecg/a01 -a qrs*. Este processo acontece sem realizar o *download* dos arquivos.

Figura 13 – *Prompt* de comando do Cygwin

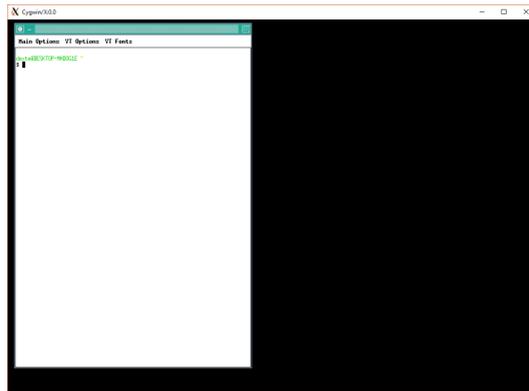


Fonte: Autor, 2019

Outra utilidade é a visualização dos sinais escolhidos sem a necessidade de fazer o *download* dos arquivos. Para isto, é necessária a inicialização do servidor de nome “Cygwin X”, que é o ambiente gráfico incluso no Cygwin. É necessário executar o comando *startx* para inicialização do pacote. É aberta uma nova janela, com representação na Figura 14.

A partir desta nova janela pode-se inserir comandos para visualização dos sinais da PhysioBank. Para o sinal “a01” contido na base de dados, o comando *wave -r apnea-ecg/01* é utilizado. O ECG, em uma janela de 10s, está representado na Figura 15.

Figura 14 – *Prompt* de comando do Cygwin X



Fonte: Autor, 2019

Figura 15 – Visualização do sinal de ECG no Cygwin X



Fonte: Autor, 2019

3.4 PROCESSAMENTO NO MATLAB

3.4.1 Transformação dos arquivos

O *software* escolhido para realizar o PDS neste trabalho foi o MATLAB (MathWorks Inc, versão 2017b). Este *software* dispõe de uma vasta biblioteca de manipulação de matrizes, visualização de funções, implementação de algoritmos, assim como várias ferramentas de PDS necessárias para cumprir os objetivos explicitados anteriormente.

Para conseguir manipular os sinais, primeiramente é necessário converter os arquivos para outras extensões, já que o MATLAB não consegue processar os arquivos originalmente presentes na base de dados. Assim, é necessário o *download* da *toolbox* de código aberto WFDB (SILVA; MOODY, 2014), presente na PhysioToolkit.

Esta *toolbox* contém as funções disponíveis também no Cygwin e Cygwin X, assim como funções extras para visualização dos arquivos manipulados, conversores de extensões de arquivos (sem prejuízo aos dados contidos) e funções para cálculo de intervalos característicos do ECG, como a FC e intervalo QRS.

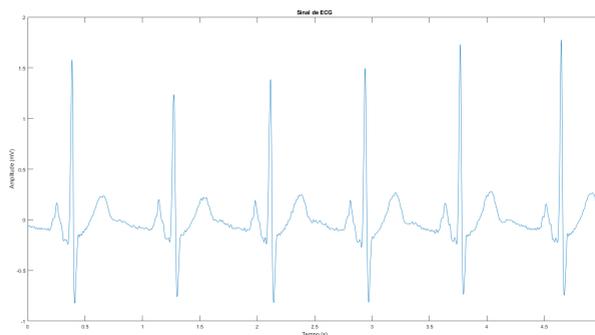
A primeira função a ser utilizada é *wfdb2mat*, que transforma os arquivos de extensão *.dat* em *.mat*, reconhecida nativamente pelo MATLAB, a partir das informações do arquivo *.hea*. O código desta função está disponível no Anexo A. Juntamente com o arquivo *.mat* é criado um novo arquivo *.hea*, com as informações do sinal convertido.

O próximo passo é a execução da função *rdmat* que, utilizando as informações contidas no novo arquivo *.hea* gerado,

transporta o conteúdo do arquivo para variáveis internas do MATLAB, permitindo assim a manipulação do sinal.

O padrão de execução da função é $[tm, signal] = rdmatrix(recordName)$, onde tm é o tempo da gravação em s; $signal$ é a variável com a amplitude das amostras em mV (no caso da base de dados escolhida); e $recordName$ é o arquivo escolhido para manipulação. O código da função está disponível no Anexo B. O sinal de ECG é mostrado na Figura 16.

Figura 16 – Gravação de um sinal de ECG da base de dados



Fonte: Autor, 2019

3.4.2 Adição de ruídos

Os ruídos que serão adicionados ao sinal são os chamados *baseline wander* (excursão da linha de base) e *powerline* (linha de rede). O primeiro ruído se refere a um sinal senoidal com uma frequência muito baixa (parecendo um sinal de CC); e o segundo ruído se refere à senoide do ruído da rede elétrica.

Para este trabalho, o ruído de base adicionado tem como

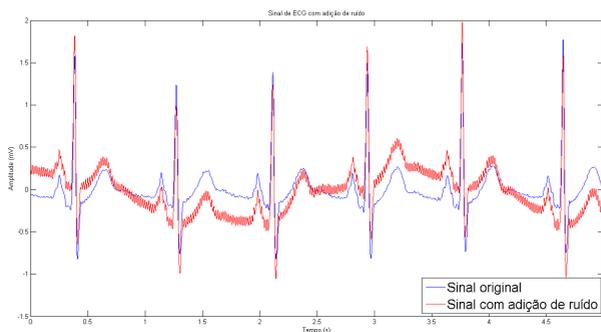
característica uma frequência de 0,3Hz com tensão de pico de 0,3V, enquanto que o ruído da rede tem frequência de 60Hz (frequência da rede no Brasil) com tensão de pico de 0,05V.

3.4.3 Geração do sinal de ECG ruidoso

Para adicionar os ruídos ao sinal de ECG, primeiramente, é necessário fazer uma interpolação do sinal original da base de dados. Isto é necessário porque, ao construir o ruído de 60Hz, uma f_s de 100Hz não obedeceria o Teorema da Amostragem de Nyquist-Shannon. Assim, para obtermos uma boa resolução do sinal de 60Hz, uma f_s de 1000Hz foi escolhida. A interpolação, do tipo cúbica, é realizada pela função *spline* do MATLAB.

Com o sinal interpolado, foram criadas as senoides e somadas ao sinal de ECG. Um exemplo comparativo entre o sinal original e o sinal com adição de ruído pode ser visto na Figura 17.

Figura 17 – Comparação do sinal de ECG original e com adição de ruído



Fonte: Autor, 2019

3.4.4 Criação e aplicação dos filtros

Para remoção dos ruídos nos sinais de ECG foram escolhidos 4 tipos de filtros FIR e 4 tipos de filtros IIR, sem nenhuma ordem específica de importância. Os filtros escolhidos estão presentes na Tabela 2.

Tabela 2 – Tipos de filtros escolhidos

FIR	IIR
Equiripple	Butterworth
Kaiser	Chebyshev Tipo I
Bartlett	Chebyshev Tipo II
Hamming	Elíptico

Fonte: Autor, 2019

A criação dos parâmetros dos filtros foi realizada com o auxílio da ferramenta *Filter Designer* presente no MATLAB. Com esta ferramenta, é possível gerar as funções criadoras dos filtros de maneira automatizada (assim como seus coeficientes), utilizando os valores das frequências do ruído caracterizado previamente.

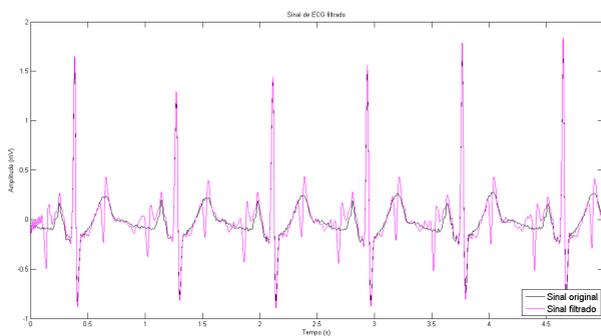
Os tipos de filtros a serem utilizados são do tipo FPA e FRF. O FPA é usado para remoção do ruído de base, enquanto que o FRF é usado para remoção do ruído da rede. Neste caso específico de FRF, que tem como banda de rejeição projetada uma frequência única, pode ser chamado de filtro *notch*.

Para o FPA, foram escolhidas a f_c de 1,5Hz, f_{stop} de 0,3Hz e f_{pass} de 2Hz. Já para o FRF, foram escolhidas as f_c/f_{pass} de 58,5Hz e 61,5Hz, com f_{stop} de 59,9Hz e 60,1Hz.

Com os filtros construídos, é utilizada a função *filtfilt* para aplicação da filtragem no ECG. Esta diferença em utilização

das funções se deve ao modo de como cada tipo de filtro gera um atraso de grupo de maneira diferente, como explicado anteriormente. Um exemplo de resultado final do ECG já filtrado pode ser visto na Figura 18.

Figura 18 – Comparação do sinal de ECG original e filtrado para o filtro FIR Equiripple



Fonte: Autor, 2019

Como haveriam muitas imagens produzidas pelas análises realizadas, será escolhido o arquivo “a01” para demonstrar o funcionamento dos filtros, comparando os resultados com o sinal original de ECG.

3.4.5 Escolha dos parâmetros de avaliação

Quando tratamos de sinais que são de difícil análise tanto no domínio do tempo quanto no da frequência, como um sinal biomédico, pode ser difícil por vezes escolher o método de avaliação de performance dos métodos de filtragem utilizados.

Um dos métodos mais utilizados para verificação de per-

formance é a potência do sinal, assim como a SNR, de acordo com artigos na área de processamento de sinais biomédicos de REJA; MURTUZAAND; ROY, 2018; AHAMMED, 2018 e SINGH; AYUB; SAINI, 2013. Assim, parece natural utilizar as mesmas métricas neste trabalho.

Para cálculo da potência de um sinal discreto, utiliza-se a Equação 4:

$$P = \frac{1}{L} \sum_{n=1}^{n=L} |x[n]|^2 \quad (4)$$

onde L é o número de amostras do sinal e $x(n)$ o valor da amostra n (que varia de 1 até L).

Já para o cálculo da SNR, utiliza-se a Equação 5:

$$\text{SNR} = \frac{P_{\text{signal}}}{P_{\text{noise}}} \quad \text{ou} \quad \text{SNR}_{\text{dB}} = 10 \log_{10} \left(\frac{P_{\text{signal}}}{P_{\text{noise}}} \right) \quad (5)$$

Este tipo de operação é feita no MATLAB utilizando a função ***norm*** para o somatório das amostras e ***length*** para cálculo do número de amostras do sinal.

Além disso, calcula-se a tensão pico a pico (V_{pp}) como parâmetro secundário, com sua fórmula presente na Equação 6:

$$V_{pp}(x) = \max(x) - \min(x) \quad (6)$$

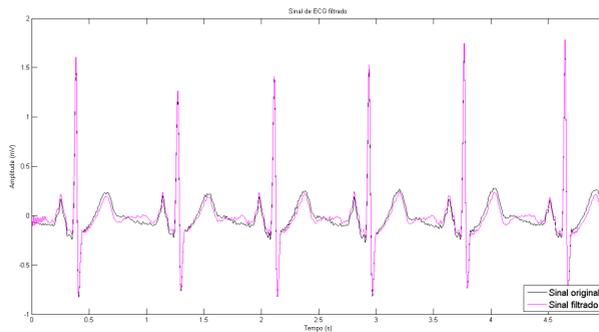
O código desenvolvido para a realização dos procedimentos descritos está disponível no Apêndice A.

4 RESULTADOS

Os arquivos escolhidos foram rodados nos códigos do Anexo A e do Apêndice A. Os valores de V_{pp} e SNR resultantes de cada ECG após a filtragem foram somados e, então, uma operação de média aritmética foi realizada, com o objetivo de medir o desempenho médio de cada filtro, comparando-os entre si.

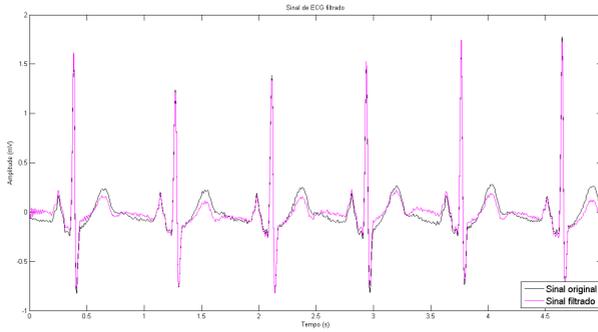
Como frisado anteriormente, as Figuras a seguir representam os resultados para o sinal “a01”. O resultado para o filtro FIR Equiripple já foi mostrado na Figura 18. O restante dos sinais filtrados está presente nas Figuras 19 a 25.

Figura 19 – Comparação do sinal de ECG original e filtrado para o filtro FIR Kaiser



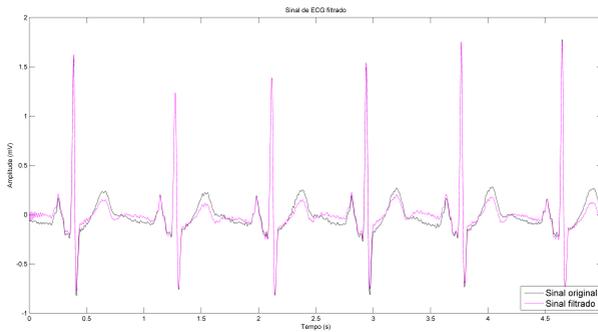
Fonte: Autor, 2019

Figura 20 – Comparação do sinal de ECG original e filtrado para o filtro FIR Bartlett



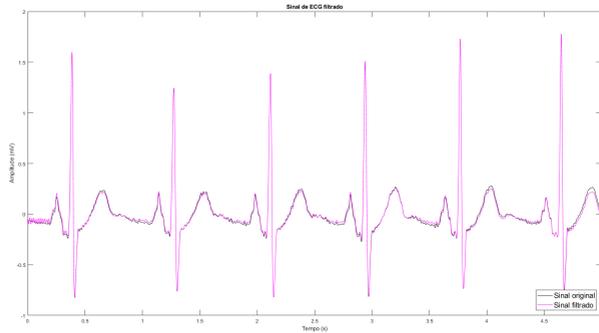
Fonte: Autor, 2019

Figura 21 – Comparação do sinal de ECG original e filtrado para o filtro FIR Hamming



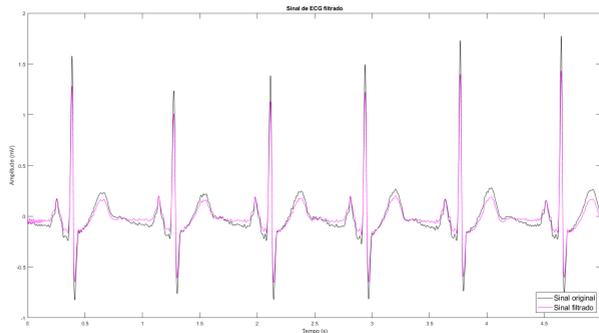
Fonte: Autor, 2019

Figura 22 – Comparação do sinal de ECG original e filtrado para o filtro IIR Butterworth



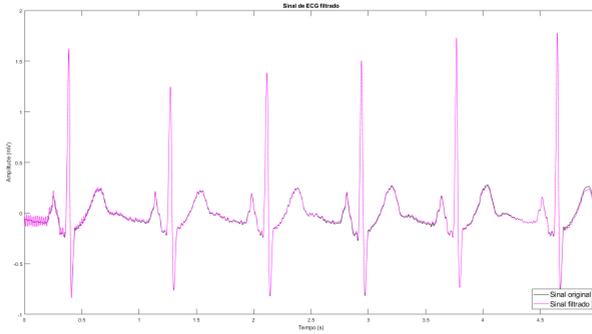
Fonte: Autor, 2019

Figura 23 – Comparação do sinal de ECG original e filtrado para o filtro IIR Chebyshev I



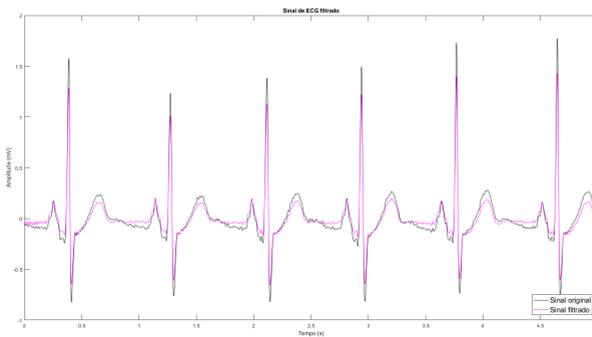
Fonte: Autor, 2019

Figura 24 – Comparação do sinal de ECG original e filtrado para o filtro IIR Chebyshev II



Fonte: Autor, 2019

Figura 25 – Comparação do sinal de ECG original e filtrado para o filtro IIR Elíptico



Fonte: Autor, 2019

Os resultados dos processamentos realizados estão presentes nas Tabelas 3 e 4.

Tabela 3 – Valores de SNR e V_{pp} do sinal original e dos sinais filtrados

Tipo de filtro	V_{pp} do ECG original (mV)	V_{pp} do ECG filtrado (mV)	SNR do sinal original (dB)	SNR do sinal filtrado (dB)
Bartlett (FIR)		6,7457		1,3763
Butterworth (IIR)		7,4106		1,9164
Chebyshev I (IIR)		6,8916		2,0036
Chebyshev II (IIR)	9,0733	7,0809	3,3958	2,0538
Elíptico (IIR)		7,3929		2,0958
Equiripple (FIR)		7,5262		2,6156
Hamming (FIR)		6,7443		1,4086
Kaiser (FIR)		6,9692		1,8303

Fonte: Autor, 2019

Tabela 4 – Perdas dos ECGs filtrados em relação ao original, em porcentagem

Tipo de filtro	Perdas em relação à V_{pp} original	Perdas em relação à SNR original
Bartlett (FIR)	25,65%	37,19%
Butterworth (IIR)	18,32%	28,87%
Chebyshev I (IIR)	24,04%	27,42%
Chebyshev II (IIR)	21,95%	26,58%
Elíptico (IIR)	18,52%	25,86%
Equiripple (FIR)	17,05%	16,44%
Hamming (FIR)	25,67%	36,72%
Kaiser (FIR)	23,19%	30,26%

Fonte: Autor, 2019

5 DISCUSSÃO DOS RESULTADOS

De acordo com a análise realizada, é possível concluir que o tipo de filtro influi diretamente no resultado do processo de filtragem. A diferença mais evidente neste trabalho é o contraste de desempenho entre os diferentes tipos de filtros FIR.

O filtro Equiripple teve eficácia muito superior aos outros filtros FIR utilizados. Isto possivelmente se deve ao fato de que a construção do filtro Equiripple é feita de maneira diferente dos outros, com relação ao método de construção do algoritmo do filtro.

De acordo com os dados da Tabela 3, o filtro que resultou em um sinal com a maior SNR depois do processamento foi o FIR Equiripple, representando uma perda de aproximadamente 16,44% (linear) ou 0,78dB (logarítmica) na SNR em relação ao ECG original.

Na análise de V_{pp} , todos eles ficaram dentro da faixa dinâmica de amplitude considerada normal de 1 a 10mV (COHEN, 2015). Como foi realizada uma média de valores, não é possível obter uma análise única de cada sinal analisado. O algoritmo do Apêndice A, no entanto, pode ser rodado para analisar cada sinal separadamente, para verificar possíveis desvios tanto no formato quanto nas amplitudes dos sinais de ECG.

Com relação ao valor da V_{pp} , o filtro que mais preservou

o sinal foi novamente o FIR Equiripple, com perda de aproximadamente 17,05% de amplitude em relação ao sinal original.

Uma informação que pode ser extraída é que, excluindo o filtro Equiripple, os filtros que tiveram melhor eficácia foram os do tipo IIR, sendo eles Elíptico, Chebyshev II, Butterworth e Chebyshev I, em ordem decrescente em relação ao parâmetro da SNR. O mesmo fenômeno ocorre com relação aos valores de V_{pp} , apenas mudando a ordem dos filtros Butterworth e Chebyshev I.

Além disso, excluindo novamente os filtros Equiripple e do tipo IIR, o restante das filtragens obtiveram perdas de aproximadamente $1/3$ em relação à SNR e $1/4$ em relação à V_{pp} . Esta diminuição dos valores pode influenciar em análises médicas para diagnóstico de certas doenças, por exemplo.

6 CONCLUSÃO

Sendo o filtro que apresentou melhor eficácia, o *design* do filtro Equiripple é feito utilizando um método iterativo, no qual a Transformada de Fourier Discreta (DFT) é realizada em um algoritmo inicial do filtro, que é então ajustado de acordo com os requerimentos do filtro e então é realizada uma DFT inversa. No domínio do tempo os coeficientes da função são limitados à ordem do filtro, para em seguida ser realizada mais uma operação de DFT e feito um ajuste final.

Os filtros FIR, tirando o Equiripple, são criados por um método de janelamento, que consiste primeiramente na criação de um filtro IIR ideal, que sofre então um truncamento ao ser multiplicado por uma função janela, sendo definida por ter valores iguais a zero após determinado intervalo, normalmente simétrica e de valor máximo na sua metade. Um exemplo deste tipo de função está presente no Anexo C.

Os filtros IIR apresentaram eficácias similares, devido ao fato de que todos são construídos utilizando o mesmo método, que consiste em primeiro basear o filtro desejado em um protótipo de filtro analógico (que também possui resposta ao impulso infinita) com as mesmas especificações. Em seguida, é realizado um escalonamento das faixas de frequências resultantes para ficarem iguais às desejadas. Por fim, o filtro analógico projetado é convertido em digital, processo geralmente feito por um método conhecido como transformação bilinear.

As métricas de desempenho escolhidas conseguiram cumprir o papel desejado, que era salientar as semelhanças e diferenças dentre vários métodos de filtragem diferentes, assim como permitir a escolha de um filtro mais adequado para um ECG, que neste caso foi o do tipo FIR Equiripple. Como os sinais depois da filtragem são muito semelhantes entre si, não é possível identificar muitas diferenças visuais nas figuras resultantes.

6.1 SUGESTÕES PARA TRABALHOS FUTUROS

- Utilização de mais tipos diferentes de filtros para análise, como o filtro de Média Móvel e LMS, por exemplo;
- Adição de mais métricas de desempenho, como o EQM, para uma análise ainda mais sólida na escolha do filtro de maior performance;
- Analisar uma possível otimização do código para diminuição do tempo de processamento, que dura em torno de 1 minuto por sinal;
- Desenvolvimento de uma interface gráfica que exiba os sinais de saída e entrada do projeto, assim como os resultados de performance obtidos.

REFERÊNCIAS

- 12-LEAD ECG Placement Guide with Illustrations. 2011. <https://www.cablesandsensors.eu/pages/12-lead-ecg-placement-guide-with-illustrations>. Online, acesso em 09/12/2018.
- AHAMMED, K. Removal of Structured Noise and Base Line Wander From ECG Signals via LMS Adaptive and Fixed Notch Filter. **European Journal of Engineering Research and Science**, v. 3, p. 12–15, 2018.
- ANALOG Digital Sampling. 2018. <https://airfreshener.club/quotes/analogue-digital-sampling.html>. Online, acesso em 13/12/2018.
- BHOGESHWAR, S. S.; SONI, M. K.; BANSAL, D. Design of Simulink Model to denoise ECG signal using various IIR & FIR Filters. In: **2014 International Conference on Reliability Optimization and Information Technology (ICROIT)**. India: IEEE, 2014. v. 1, p. 477–483.
- BOURKE, P. **Windows**. 1998. <http://paulbourke.net/miscellaneous/windows/>. Online, acesso em 20/01/2019.
- BREAK Nyquist Criterion Using Undersampling Technique. 2007. <http://www.ni.com/newsletter/50078/en/>. Online, acesso em 10/12/2018.
- BURNS, E. **QT Interval**. 2018. <https://litfl.com/qt-interval-ecg-library/>. Online, acesso em 09/12/2018.
- CERUTTI, S.; BIANCHI, A. T.; MAINARDI, L. T. Digital Biomedical Signal Acquisition and Processing. In: BRONZINO, J. D. (Ed.). **The Biomedical Engineering Handbook**. 4. ed. Boca Raton: CRC Press LLC, 2015.

CHRISTENSEN, B. **Normal Electrocardiography (ECG) Intervals**. 2014.

<https://medicine.medscape.com/article/2172196-overview>.

Online, acesso em 09/12/2018.

COHEN, A. Biomedical Signals: Origin and Dynamic Characteristics; Frequency-Domain Analysis. In: BRONZINO, J. D. (Ed.). **The Biomedical Engineering Handbook**. 4. ed. Boca Raton: CRC Press LLC, 2015.

DIGITAL Communication Tutorial. 2018.

https://www.tutorialspoint.com/digital_communication/.

Online, acesso em 13/12/2018.

GOLDBERGER, A. L.; AMARAL, L. A. N.; GLASS, L.; HAUSDORFF, J. M.; IVANOV, P. C.; MARK, R. G.; MIETUS, J. E.; MOODY, G. B.; PENG, C.-K.; STANLEY, H. E. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. **Circulation**, v. 101, p. e215–e220, 2000.

NAJARIAN, K.; SPLINTER, R. **Biomedical Signal and Image Processing**. 2. ed. Boca Raton: CRC Press, 2012.

PENZEL, T.; MOODY, G. B.; MARK, R. G.; GOLDBERGER, A. L.; PETER, J. H. The apnea-ECG database. In: **Computers in Cardiology 2000**. Cambridge: IEEE, 2000. v. 27, p. 255–258.

RAJNI, R.; KAUR, I. Electrocardiogram Signal Analysis - An Overview. **International Journal of Computer Applications**, v. 84, p. 22–25, 2013.

RAWSHANI, A. **Clinical eletrocardiography and ECG interpretation**. 2018. <https://ecgwaves.com/introduction-electrocardiography-ecg-book/>. Online, acesso em 09/12/2018.

REJA, M. I.; MURTUZAAND, M. G.; ROY, R. Combinations of Different FIR Windows for Removal of Baseline and Power Line Noise from Electrocardiogram. **Modern Applied Science**, v. 12, 2018.

- SAXENA, C.; SHARMA, A.; SRIVASTAV, R.; GUPTA, H. K. Denoising of ECG Signals Using FIR & IIR Filter: A Performance Analysis. **International Journal of Engineering and Technology**, v. 7, p. 1–5, 2018.
- SEMMLOW, J. **Circuits, Signals, and Systems for Bioengineers: A MATLAB-Based Introduction**. 3. ed. New Jersey: Elsevier, 2017.
- SHANNON, C. E. Communication in the Presence of Noise. **Proceedings of the IRE**, v. 37, p. 10–21, 1949.
- SILVA, I.; MOODY, G. B. An Open-source Toolbox for Analysing and Processing Physionet Databases in MATLAB and Octave. **Journal of Open Research Software**, v. 2, 2014.
- SINGH, N.; AYUB, S.; SAINI, J. P. Design of Digital IIR Filter for Noise Reduction in ECG Signal. In: **2013 5th International Conference and Computational Intelligence and Communication Networks**. [S.l.: s.n.], 2013. p. 171–176.
- SMITH, S. W. **The Scientist and Engineer's Guide to Digital Signal Processing**. 2. ed. San Diego: California Technical Publishing, 1999.
- TOMPKINS, W. J. **Biomedical Digital Signal Processing**. 1. ed. New Jersey: Prentice Hall, 2000.
- TURCHIELLO, G. de M. **Plataforma Visual de Processamento de Sinais Biomédicos**. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, 2014.
- YU, Y. J.; SHI, D.; LIM, Y. C. Low-Complexity Design of VariableBandedge Linear Phase FIR Filters With Sharp Transition Band. **IEEE Transactions on Signal Processing**, v. 57, p. 1328 – 1338, 2009.
- ZIEMSEN, T.; GASCH, J.; RUEDIGER, H. Influence of ECG Sampling Frequency on Spectral Analysis of RR Intervals and Baroreflex Sensitivity Using the EUROBAVAR Data set.

Journal of Clinical Monitoring and Computing, v. 22, p. 159-168, 2008.

ANEXO A – Código da função do MATLAB *wfdb2mat*

```

function wfdb2mat(varargin)
%
% wfdb2mat(recordName, signalList, N, NO)
%
% Wrapper to WFDB WFDB2MAT:
%   http://physionet.org/physiotools/wag/wfdb2m-1.htm
%
% Converts a WFDB-compatible signal file to MATLAB/Octave *.mat file.
% The output files are recordName.mat and recordName.heh. The standard
%   output of
% WFDB2MAT will be saved in a file recordName.info.
%
% Required Parameters:
%
% recorName
%   String specifying the name of the record in the WFDB path or
%   in the current directory.
%
% Optional Parameters are:
%
% signalList
%   A Mx1 array of integers. Read only the signals (columns)
%   named in the signalList (default: read all signals).
% N
%   A 1x1 integer specifying the sample number at which to stop
%   reading the
%   record file (default read all the samples = N).
% NO
%   A 1x1 integer specifying the sample number at which to start
%   reading the
%   record file (default 1 = first sample).
%
% NOTE:
%   You can use the WFDB2MAT command in order to convert the record
%   data into a *.mat file,
%   which can then be loaded into MATLAB/Octave's workspace using
%   the LOAD command.
%   This will load the signal data in raw units (use RDMAT to load
%   the signal in physical units).
%
% Written by Ikaro Silva, 2014

```

```

% Last Modified: September 15, 2014
% Version 0.1
%
% Since 0.9.7
%
% %Example:
% wfdb2mat('mitdb/200')
% [tm,signal,Fs,labels]=rdmat('200m');
%
%
% See also RDSAMP, RDMAT

%endOfHelp

persistent javaWfdbExec config
if isempty(javaWfdbExec)
    [javaWfdbExec,config]=getWfdbClass('wfdb2mat');
end

%Set default pararameter values
inputs={'recordName','signalList','N','NO'};
signalList=[];
N=[];
NO=1;
for n=1:nargin
    if (~isempty(varargin{n}))
        eval([inputs{n} '='varargin{n};'])
    end
end

wfdb_argument={'-r',recordName,'-f',[ 's' num2str(NO-1)]};

if (~isempty(N))
    wfdb_argument{end+1}='-t';
    wfdb_argument{end+1}=[ 's' num2str(N)];
end

if (~isempty(signalList))
    wfdb_argument{end+1}='-s ';
    %-1 is necessary because WFDB is 0 based indexed.
    for sInd=1:length(signalList)
        wfdb_argument{end+1}=[num2str(signalList(sInd)-1)];
    end
end

data=javaWfdbExec.execToStringList(wfdb_argument);

```

ANEXO B – Código da função do MATLAB *rdmat*

```

function varargout=rdmat(varargin)
%
% [tm,signal,Fs,signinfo]=rdmat(recordName)
%
% Import a signal in physical units from a *.mat file generated by
%   WFDB2MAT.
% Required Parameters:
%
% recorName
%   String specifying the name of the *.mat file.
%
% Outputs are:
%
% tm
%   A Nx1 array of doubles specifying the time in seconds.
% signal
%   A NxM matrix of doubles contain the signals in physical units.
% Fs
%   A 1x1 integer specifying the sampling frequency in Hz for the
%   entire record.
%signinfo
%   A LxN cell array specifying the signal signinfo. Currently it is
%   a
%   structure with the following fields:
%
%       signinfo.Units
%       signinfo.Baseline
%       signinfo.Gain
%       signinfo.Description
%
% NOTE:
%   You can use the WFDB2MAT command in order to convert the record
%   data into a *.mat file,
%   which can then be loaded into MATLAB/Octave's workspace using
%   the LOAD command.
%   This sequence of procedures is quicker (by several orders of
%   magnitude) than calling RDSAMP.
%   The LOAD command will load the signal data in raw units, use
%   RDMAT to load the signal in physical units.
%
% KNOWN LIMITATIONS:

```

```

%      This function currently does support several of the features
described
%      in the WFDB record format (such as multiresolution signals) :
%      http://www.physionet.org/physiotools/wag/header-5.htm
%      If you are not sure that the record (or database format) you are
reading is
%      supported, you can do an integrity check by comparing the output
with RDSAMP:

%
%      [tm,signal,Fs,signifo]=rdmat('200m');
%      [tm2,signal2]=rdsamp('200m');
%      if(sum(abs(signal-signal2)) !=0);
%          error('Record not compatible with RDMAT');
%      end
%
%
%
% Written by Ikaro Silva, 2014
% Last Modified: November 26, 2014
% Version 1.2
%
% Since 0.9.7
%
% %Example:
% wfdb2mat('mitdb/200')
%tic;[tm,signal,Fs,signifo]=rdmat('200m');toc
%tic;[signal2]=rdsamp('200m');toc
% sum(abs(signal-signal2))
%
%
% See also RDSAMP, WFDB2MAT

%endOfHelp

%Set default pararameter values
inputs={'recordName'};
defGain=200; %Default value for missing gains

for n=1:nargin
    if(~isempty(varargin{n}))
        eval([inputs{n} '='varargin{n};'])
    end
end

outputs={'tm','val','Fs','signifo'};
fid = fopen([recordName, '.hea'], 'rt');
if(fid==-1)
    error(['Could not open file: ' recordName '.hea !'])
end

%Following the documentation described in :
%http://www.physionet.org/physiotools/wag/header-5.htm

```

```
%to parse the header file

%Skip any comment lines
str=fgetl(fid);
while(strcmp(str,'#'))
    str=fgetl(fid);
end

%Process Record Line Info
info=textscan(str,'%s %d %f %d %s %s');
M=info{2}; %Number of signals present
Fs=info{3};

%Process Signal Specification lines. Assumes no comments between lines.
siginfo(M)=struct();
for m = 1:M
    str=fgetl(fid);
    info=textscan(str,'%s %s %s %d %d %f %d %d %s');
    fmt=info{2}{:};
    gain=info{3}{:};

    %Get Signal Units if present
    ind=strfind(gain,'/');
    if(~isempty(ind))
        siginfo(m).Units=gain(ind+1:end);
        gain=gain(1:ind-1);
    end

    %Get Signal Baseline if present
    ind=strfind(gain,'(');
    if(~isempty(ind))
        ind2=strfind(gain,')');
        siginfo(m).Baseline=str2num(gain(ind+1:ind2-1));
        gain=gain(1:ind-1);
    else
        %If Baseline is missing, set it equal to ADC Zero
        adc_zero=info{5};
        if(~isempty(adc_zero))
            siginfo(m).Baseline=double(adc_zero);
        else
            error('Could not obtain signal baseline');
        end
    end

    %Get Signal Gain
    gain=str2num(gain);
    if(gain==0)
        %Set gain to default value in this case
        gain=defGain;
    end
    siginfo(m).Gain=double(gain);
end
```

```

%Get Signal Descriptor
siginfo(m).Description=info{9}{:};

% Store format for later
siginfo(m).fmt=fmt(1:strfind(fmt,'+')-1);

end
fclose(fid);

load([recordName '.mat']);

for m = 1:M
    % Interpreting digital values of byte offset format 80
    if strcmp(siginfo(m).fmt, '80')
        val(m,:)=val(m,:)-128;
        wfdbNaN=-128;
    elseif strcmp(siginfo(m).fmt, '16')
        wfdbNaN=-32768;
    else
        wfdbNaN=-2147483648;
    end

    % Fill in NaNs before subtracting and dividing.
    val(m, val(m,:)==wfdbNaN)=nan;

    %Convert from digital units to physical units.
    % Mapping should be similar to that of rdsamp.c
    % http://www.physionet.org/physiotools/wfdb/app/rdsamp.c
    val(m, :) = (val(m, :) - siginfo(m).Baseline) / siginfo(m).Gain;
end

%Reshape to the Toolbox's standard format
val=val';

%Generate time vector
N=size(val,1);
tm =linspace(0,(N-1)/Fs,N);

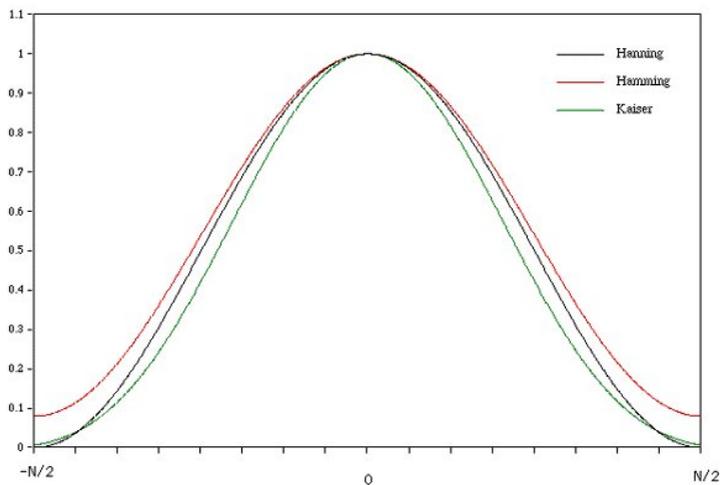
for n=1:nargout
    eval(['varargout{n}=' outputs{n} ' ;'])
end

end

```

ANEXO C – Exemplo de função janela

Figura 26 – Funções janelas para criação de filtros FIR



Fonte: Disponível em <http://paulbourke.net/miscellaneous/windows/>
Acesso em 20/11/2018

APÊNDICE A – Código do MATLAB para processamento dos sinais, criação dos gráficos e cálculo dos parâmetros

```

%-----
%   AUTOR: Felipe Vanroo Silveira
%   CURSO: Engenharia Eletronica - UFSC
%
% DESCRICAO: Arquivo para realizacao de filtragem em sinais
%            de ECG e calculo da SNR
%
%-----

clear;
clc;
close;

%%
%Escolha do sinal

caminho = pwd;
caminho = what(caminho);
caminho = [caminho.mat];
[sinal, escolha] = listdlg('PromptString', 'Escolha um arquivo:', '
    SelectionMode', 'single', 'ListString', caminho);
caminho = string(caminho);
[filepath, name, ext] = fileparts(caminho(sinal, 1));

%%
%Propriedades e adicao de ruido

Fs = 1000; %frequencia de amostragem
[t, signal] = rdmat(char(name)); %importacao do arquivo para variaveis
internas
ts = (0:1/1000:(size(signal,1)-1)/100); %criacao do vetor de tempo
ecg = spline(t, signal, ts); %mudanca da frequencia de amostragem do
sinal

```

```

f_baseline = 0.3; %frequencia do ruido de base
f_60hz = 60; %frequencia do ruido de rede
coeff_60hz = 0.05; %amplitude do ruido de base
coeff_baseline = 0.3; %amplitude do ruido de base
signal_60hz = cos(2*pi*f_60hz*ts);
signal_baseline = cos(2*pi*f_baseline*ts);
noise = coeff_60hz*signal_60hz + coeff_baseline*signal_baseline; %sinal
    de ruido
ecg_noise = ecg + noise;

%%
%Imagem do sinal com e sem ruido

figure('Name', 'Sinal de ECG original e corrompido', 'NumberTitle', 'off
');
plot(ts(1:5000), ecg(1:5000), 'b');
hold on
plot(ts(1:5000), ecg_noise(1:5000), 'r');
hold off
title('Sinal de ECG com adicao de ruido');
legend({'Sinal original', 'Sinal com adicao de ruido'}, 'location', '
southeast', 'FontSize', 20);
ylabel('Amplitude (mV)');
xlabel('Tempo (s)');

%%
%Parametros de filtros FIR Highpass

fstop_baseline = 0.3;
fpass_baseline = 2;
fc_baseline = 1.5;
Wstop_baseline = 1;
Wpass_baseline = 0.5;
dens = 20;
N = 500;
flag = 'noscale';
beta = 0.5;

%%
%Filtros FIR Highpass
b_baseline_equiripple = firpm(N, [0 fstop_baseline fpass_baseline Fs
/2]/(Fs/2), ...
    [0 0 1 1], [Wstop_baseline Wpass_baseline], {dens});

win_baseline_kaiser = kaiser(N+1, beta);
b_baseline_kaiser = fir1(N, fc_baseline/(Fs/2), 'high',
    win_baseline_kaiser, flag);

win_baseline_bartlett = bartlett(N+1);
b_baseline_bartlett = fir1(N, fc_baseline/(Fs/2), 'high',
    win_baseline_bartlett, flag);

```

```

win_baseline_hamming = hamming(N+1);
b_baseline_hamming = fir1(N, fc_baseline/(Fs/2), 'high',
    win_baseline_hamming, flag);

%%
%Parametros de filtros filtros FIR Bandstop

fpass1_60hz = 58.5;
fstop1_60hz = 59.9;
fstop2_60hz = 60.1;
fpass2_60hz = 61.5;
Wpass1_60hz = 0.5;
Wstop_60hz = 20;
Wpass2_60hz = 0.5;
N = 500;
fc1_60hz = 58.5;
fc2_60hz = 61.5;

%%
%Filtros FIR Bandstop

b_60hz_equiripple = firpm(N, [0 fpass1_60hz fstop1_60hz fstop2_60hz
    fpass2_60hz Fs/2]/(Fs/2), [1 1 0 0 1 ...
    1], [Wpass1_60hz Wstop_60hz Wpass2_60hz], {dens});

win_60hz_kaiser = kaiser(N+1, beta);
b_60hz_kaiser = fir1(N, [fc1_60hz fc2_60hz]/(Fs/2), 'stop',
    win_60hz_kaiser, flag);

win_60hz_bartlett = bartlett(N+1);
b_60hz_bartlett = fir1(N, [fc1_60hz fc2_60hz]/(Fs/2), 'stop',
    win_60hz_bartlett, flag);

win_60hz_hamming = hamming(N+1);
b_60hz_hamming = fir1(N, [fc1_60hz fc2_60hz]/(Fs/2), 'stop',
    win_60hz_hamming, flag);

%%
%Parametros de filtros IIR Highpass

Astop_baseline = 20;
Apass_baseline = 0.5;
match = 'stopband';

%%
%Filtros IIR Highpass

Hd_baseline_butter = designfilt('highpassiir', 'StopbandFrequency',
    fstop_baseline, ...
    'PassbandFrequency', fpass_baseline, 'StopbandAttenuation',

```

```

    Astop_baseline, ...
    'PassbandRipple', Apass_baseline, 'SampleRate', Fs, 'DesignMethod',
    'butter');

Hd_baseline_cheby1 = designfilt('highpassiir', 'StopbandFrequency',
    fstop_baseline, ...
    'PassbandFrequency', fpass_baseline, 'StopbandAttenuation',
    Astop_baseline, ...
    'PassbandRipple', Apass_baseline, 'SampleRate', Fs, 'DesignMethod',
    'cheby1');

Hd_baseline_cheby2 = designfilt('highpassiir', 'StopbandFrequency',
    fstop_baseline, ...
    'PassbandFrequency', fpass_baseline, 'StopbandAttenuation',
    Astop_baseline, ...
    'PassbandRipple', Apass_baseline, 'SampleRate', Fs, 'DesignMethod',
    'cheby2');

Hd_baseline_ellip = designfilt('highpassiir', 'StopbandFrequency',
    fstop_baseline, ...
    'PassbandFrequency', fpass_baseline, 'StopbandAttenuation',
    Astop_baseline, ...
    'PassbandRipple', Apass_baseline, 'SampleRate', Fs, 'DesignMethod',
    'ellip');

%%
%%Parametros de filtros IIR Bandstop

Apass1_60hz = 0.5;
Astop_60hz = 20;
Apass2_60hz = 0.5;

%%

Hd_60hz_butter = designfilt('bandstopiir', 'PassbandFrequency1',
    fpass1_60hz, ...
    'StopbandFrequency1', fstop1_60hz, 'StopbandFrequency2', fstop2_60hz
    , ...
    'PassbandFrequency2', fpass2_60hz, 'PassbandRipple1', Apass1_60hz,
    ...
    'StopbandAttenuation', Astop_60hz, 'PassbandRipple2', Apass2_60hz,
    ...
    'SampleRate', Fs, 'DesignMethod', 'butter');

Hd_60hz_cheby1 = designfilt('bandstopiir', 'PassbandFrequency1',
    fpass1_60hz, ...
    'StopbandFrequency1', fstop1_60hz, 'StopbandFrequency2', fstop2_60hz
    , ...
    'PassbandFrequency2', fpass2_60hz, 'PassbandRipple1', Apass1_60hz,
    ...
    'StopbandAttenuation', Astop_60hz, 'PassbandRipple2', Apass2_60hz,

```

```
    ...
    'SampleRate', Fs, 'DesignMethod', 'cheby1');

Hd_60hz_cheby2 = designfilt('bandstopiir', 'PassbandFrequency1',
    fpass1_60hz, ...
    'StopbandFrequency1', fstop1_60hz, 'StopbandFrequency2', fstop2_60hz
    , ...
    'PassbandFrequency2', fpass2_60hz, 'PassbandRipple1', Apass1_60hz,
    ...
    'StopbandAttenuation', Astop_60hz, 'PassbandRipple2', Apass2_60hz,
    ...
    'SampleRate', Fs, 'DesignMethod', 'cheby2');

Hd_60hz_ellip = designfilt('bandstopiir', 'PassbandFrequency1',
    fpass1_60hz, ...
    'StopbandFrequency1', fstop1_60hz, 'StopbandFrequency2', fstop2_60hz
    , ...
    'PassbandFrequency2', fpass2_60hz, 'PassbandRipple1', Apass1_60hz,
    ...
    'StopbandAttenuation', Astop_60hz, 'PassbandRipple2', Apass2_60hz,
    ...
    'SampleRate', Fs, 'DesignMethod', 'ellip');

%%
%Aplicacao dos filtros FIR

ecg_equiripple = filtfilt(b_60hz_equiripple, 1, filtfilt(
    b_baseline_equiripple, 1, ecg_noise));

ecg_kaiser = filtfilt(b_60hz_kaiser, 1, filtfilt(b_baseline_kaiser, 1,
    ecg_noise));

ecg_bartlett = filtfilt(b_60hz_bartlett, 1, filtfilt(b_baseline_bartlett
    , 1, ecg_noise));

ecg_hamming = filtfilt(b_60hz_hamming, 1, filtfilt(b_baseline_hamming,
    1, ecg_noise));

%%
%Aplicacao dos filtros IIR

ecg_butter = filtfilt(Hd_60hz_butter, filtfilt(Hd_baseline_butter,
    ecg_noise));

ecg_cheby1 = filtfilt(Hd_60hz_cheby1, filtfilt(Hd_baseline_cheby1,
    ecg_noise));

ecg_cheby2 = filtfilt(Hd_60hz_cheby2, filtfilt(Hd_baseline_cheby2,
    ecg_noise));

ecg_ellip = filtfilt(Hd_60hz_ellip, filtfilt(Hd_baseline_ellip,
```

```
    ecg_noise));

%%
%Sinal Equiripple

figure('Name', 'Sinal de ECG original e filtrado', 'NumberTitle', 'off')
;
plot(ts(1:5000), ecg(1:5000), 'k');
hold on
plot(ts(1:5000), ecg_equiripple(1:5000), 'm');
hold off
title('Sinal de ECG filtrado');
legend({'Sinal original', 'Sinal filtrado'}, 'location', 'southeast', '
    FontSize', 14);
ylabel('Amplitude (mV)');
xlabel('Tempo (s)');

%%
%Sinal Kaiser

figure('Name', 'Sinal de ECG original e filtrado', 'NumberTitle', 'off')
;
plot(ts(1:5000), ecg(1:5000), 'k');
hold on
plot(ts(1:5000), ecg_kaiser(1:5000), 'm');
hold off
title('Sinal de ECG filtrado');
legend({'Sinal original', 'Sinal filtrado'}, 'location', 'southeast', '
    FontSize', 14);
ylabel('Amplitude (mV)');
xlabel('Tempo (s)');

%%
%Sinal Bartlett

figure('Name', 'Sinal de ECG original e filtrado', 'NumberTitle', 'off')
;
plot(ts(1:5000), ecg(1:5000), 'k');
hold on
plot(ts(1:5000), ecg_bartlett(1:5000), 'm');
hold off
title('Sinal de ECG filtrado');
legend({'Sinal original', 'Sinal filtrado'}, 'location', 'southeast', '
    FontSize', 14);
ylabel('Amplitude (mV)');
xlabel('Tempo (s)');

%%
%Sinal Hamming

figure('Name', 'Sinal de ECG original e filtrado', 'NumberTitle', 'off')
```

```
    ;
    plot(ts(1:5000), ecg(1:5000), 'k');
    hold on
    plot(ts(1:5000), ecg_hamming(1:5000), 'm');
    hold off
    title('Sinal de ECG filtrado');
    legend({'Sinal original', 'Sinal filtrado'}, 'location', 'southeast', '
        FontSize', 14);
    ylabel('Amplitude (mV)');
    xlabel('Tempo (s)');

%%
%Sinal Butter

figure('Name', 'Sinal de ECG original e filtrado', 'NumberTitle', 'off')
;
plot(ts(1:5000), ecg(1:5000), 'k');
hold on
plot(ts(1:5000), ecg_butter(1:5000), 'm');
hold off
title('Sinal de ECG filtrado');
legend({'Sinal original', 'Sinal filtrado'}, 'location', 'southeast', '
    FontSize', 14);
ylabel('Amplitude (mV)');
xlabel('Tempo (s)');

%%
%Sinal Chebyshev I

figure('Name', 'Sinal de ECG original e filtrado', 'NumberTitle', 'off')
;
plot(ts(1:5000), ecg(1:5000), 'k');
hold on
plot(ts(1:5000), ecg_cheby1(1:5000), 'm');
hold off
title('Sinal de ECG filtrado');
legend({'Sinal original', 'Sinal filtrado'}, 'location', 'southeast', '
    FontSize', 14);
ylabel('Amplitude (mV)');
xlabel('Tempo (s)');

%%
%Sinal Chebyshev II

figure('Name', 'Sinal de ECG original e filtrado', 'NumberTitle', 'off')
;
plot(ts(1:5000), ecg(1:5000), 'k');
hold on
plot(ts(1:5000), ecg_cheby2(1:5000), 'm');
hold off
title('Sinal de ECG filtrado');
```

```

legend({'Sinal original', 'Sinal filtrado'}, 'location', 'southeast', '
      FontSize', 14);
ylabel('Amplitude (mV)');
xlabel('Tempo (s)');

%%
%Sinal Elíptico

figure('Name', 'Sinal de ECG original e filtrado', 'NumberTitle', 'off')
;
plot(ts(1:5000), ecg(1:5000), 'k');
hold on
plot(ts(1:5000), ecg_ellip(1:5000), 'm');
hold off
title('Sinal de ECG filtrado');
legend({'Sinal original', 'Sinal filtrado'}, 'location', 'southeast', '
      FontSize', 14);
ylabel('Amplitude (mV)');
xlabel('Tempo (s)');

%%
% Cálculo dos valores de potencia

L = length(ecg);
P = @(x) (norm(x).^2)/L;
P_log = @(x) 10*log10(x);

P_ecg = P(ecg);
P_ecg_db = P_log(P_ecg);

P_noise = P(noise);
P_noise_db = P_log(P_noise);

P_equiripple = P(ecg_equiripple);
P_equiripple_db = P_log(P_equiripple);

P_kaiser = P(ecg_kaiser);
P_kaiser_db = P_log(P_kaiser);

P_bartlett = P(ecg_bartlett);
P_bartlett_db = P_log(P_bartlett);

P_hamming = P(ecg_hamming);
P_hamming_db = P_log(P_hamming);

P_butter = P(ecg_butter);
P_butter_db = P_log(P_butter);

P_cheby1 = P(ecg_cheby1);
P_cheby1_db = P_log(P_cheby1);

```

```
P_cheby2 = P(ecg_cheby2);
P_cheby2_db = P_log(P_cheby2);

P_ellip = P(ecg_ellip);
P_ellip_db = P_log(P_ellip);

%%
%Calculo dos valores de SNR

SNR = @(x,y) x/y;
SNR_db = @(x,y) x - y;

SNR_ecg = SNR(P_ecg, P_noise);
SNR_ecg_db = SNR_db(P_ecg_db, P_noise_db);

SNR_equiripple = SNR(P_equiripple, P_noise);
SNR_equiripple_db = SNR_db(P_equiripple_db, P_noise_db);

SNR_kaiser = SNR(P_kaiser, P_noise);
SNR_kaiser_db = SNR_db(P_kaiser_db, P_noise_db);

SNR_bartlett = SNR(P_bartlett, P_noise);
SNR_bartlett_db = SNR_db(P_bartlett_db, P_noise_db);

SNR_hamming = SNR(P_hamming, P_noise);
SNR_hamming_db = SNR_db(P_hamming_db, P_noise_db);

SNR_butter = SNR(P_butter, P_noise);
SNR_butter_db = SNR_db(P_butter_db, P_noise_db);

SNR_cheby1 = SNR(P_cheby1, P_noise);
SNR_cheby1_db = SNR_db(P_cheby1_db, P_noise_db);

SNR_cheby2 = SNR(P_cheby2, P_noise);
SNR_cheby2_db = SNR_db(P_cheby2_db, P_noise_db);

SNR_ellip = SNR(P_ellip, P_noise);
SNR_ellip_db = SNR_db(P_ellip_db, P_noise_db);

%%
%Calculo dos valores de tensao pico a pico

vpp = @(x) max(x) - min(x);

vpp_ecg = vpp(ecg);

vpp_equiripple = vpp(ecg_equiripple);
```

```
vpp_kaiser = vpp(ecg_kaiser);  
vpp_bartlett = vpp(ecg_bartlett);  
vpp_hamming = vpp(ecg_hamming);  
vpp_butter = vpp(ecg_butter);  
vpp_cheby1 = vpp(ecg_cheby1);  
vpp_cheby2 = vpp(ecg_cheby2);  
vpp_ellip = vpp(ecg_ellip);  
  
%%  
%Salvando variaveis de resultados  
  
savefile_snr = strcat('SNR_', name, '.mat');  
save(savefile_snr, 'SNR*');  
  
savefile_vpp = strcat('Vpp_', name, '.mat');  
save(savefile_vpp, 'vpp*');
```