



Aquisição e filtragem de sinais de ECG com Raspberry Pi e conversor AD

Acquisition and filtering of ECG signals with Raspberry Pi and AD converter

N. da S. Pinho*; D. de A. Gomes; D. de S. Lima

Instituto de Geociências e Engenharias, UNIFESSPA, 68506-410, Marabá-Pará, Brasil

**nielson360@gmail.com*

(Recebido em 18 de fevereiro de 2019; aceito em 03 de abril de 2019)

O eletrocardiograma (ECG) avalia a atividade elétrica do coração, e pode ser utilizado para avaliar se a atividade cardíaca está dentro da normalidade ou se há alterações nos músculos e nervos do coração. O ECG consiste na gravação do sinal elétrico gerado pela atividade cardíaca, no entanto quando esse sinal é gravado está sujeito a diversas interferências que causa variação na amplitude e/ou duração do sinal, levando a erros de diagnóstico. Para evitar essas interferências são utilizadas várias técnicas de filtragem, para este fim propomos um sistema de aquisição do sinal de ECG juntamente com uma filtragem digital. O sistema proposto utiliza um Raspberry Pi modelo B para fazer a filtragem digital, no qual implementamos um filtro digital passa-baixas com frequência de corte em 100 Hz do tipo Resposta ao Impulso Finita (RIF), utilizamos ainda uma janela de Hamming. Fizemos uma comparação entre o sinal filtrado e o sinal antes da filtragem e percebemos uma redução significativa na variabilidade do sinal, rendendo assim uma curva mais suave, dessa forma mostrando que foram eliminadas componentes de alta frequência.

Palavras-chave: Eletrocardiograma, Filtragem Digital, Raspberry Pi.

The electrocardiogram (ECG) evaluates the electrical activity of the heart, and can be used to evaluate if the heart activity is within normal or if there are changes in the muscles and nerves of the heart. The ECG consists of the recording of the electrical signal generated by the cardiac activity, however when this signal is recorded it is subject to several interferences that cause variation in the amplitude and / or duration of the signal, leading to diagnostic errors. To avoid such interference, several filtering techniques are used, for this purpose we propose an ECG signal acquisition system together with digital filtering. The proposed system uses a Raspberry Pi model B to do the digital filtering, in which we implemented a low pass digital filter with a cutoff frequency of 100 Hz of the Finite Impulse Response (FIR) type, we also used a Hamming window. We compared the filtered signal to the signal before filtering and realized a significant reduction in signal variability, thus yielding a smoother curve, thus showing that high frequency components were eliminated.

Keywords: Electrocardiogram, Filtering, Digital Filtering, Raspberry Pi.

1. INTRODUÇÃO

O monitoramento das variáveis fisiológicas do corpo humano é de essencial importância para avaliações periódica, e também é um fator de controle durante a execução de procedimentos críticos, como, por exemplo, cirurgias e avaliações pós-operatórias [6, 7]. O eletrocardiograma (ECG) é o registro da atividade elétrica do coração observada na superfície da pele [1], causada pela despolarização e repolarização do miocárdio em associação com as contrações dos átrios e ventrículos, que gera um sinal que pode ser capturado por meio de eletrodos.

O sinal capturado é fraco e de baixa frequência, suas frequências normais variam na faixa de 0.05 a 100Hz, principalmente na faixa de 0.05 a 35Hz, e sua amplitude varia de 10 μ V a 5mV, com um valor típico de 1mV. Sinais nesta faixa de amplitudes não podem ser gravados e exibidos com equipamentos comuns sem amplificação do sinal, e para usar um sinal de ECG para diagnóstico é necessário a sua filtragem [2].

O sinal do ECG típico possui cinco picos e vales, rotulados pelas letras P, Q, R, S e T. A Figura 1 exibe a forma típica desse sinal.

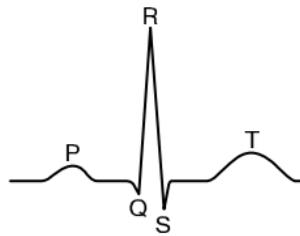


Figura 1: Forma de onda característica do ECG [3].

O ECG clínico padrão possui 12 derivações, ou seja, 12 diferenças de potencial usadas para coleta do sinal. Neste trabalho, a derivação utilizada foi a derivação I, na qual o braço esquerdo é conectado à entrada inversora do amplificador, enquanto que o braço direito é conectado na entrada não inversora, e a perna é utilizada como terra. A Figura 2 exibe como são feitas as conexões nesta derivação. Essa derivação faz parte das três derivações bipolares do ECG, que são as derivações clássicas do eletrocardiograma, descritas por Willem Einthoven (1860-1927) citada por Moukabary (2007) [15].

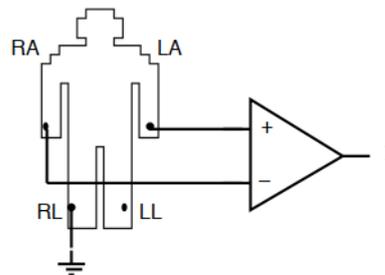


Figura 2: Derivação utilizada de Tompkins (1993) [4].

Em Gilani et al. (2018) [5] foram implementados e comparados 3 tipos de filtros para remoção do ruído causado pela interferência da linha de energia nos sinais de ECG. Foi removido o ruído nas frequências entre 47-53 Hz mantendo as frequências úteis do sinal de ECG nesse intervalo. Os dados de ECG foram retirados de repositórios *on-line* gratuitos e o ruído de interferência da linha de energia foi gerado e simulado usando a caixa de ferramentas de processamento de sinais do MATLAB. O sinal de ECG degradado foi filtrado usando os filtros Notch tradicional, Butterworth Band Stop de diferentes ordens, e o filtro adaptativo LMS (*Least Mean Square*). Os resultados mostraram que o filtro Notch não conseguiu remover a interferência em casos de variações na frequência da linha de energia. O filtro de parada de banda de alta ordem (5ª ordem) foi comparativamente melhor do que o filtro de parada de banda para uma ordem inferior, mas com a desvantagem de que ele requer mais período de processamento/convergência. Já o desempenho do LMS foi melhor em comparação com o resto dos filtros devido à sua natureza adaptativa.

Um dispositivo de ECG portátil em tempo real para detecção de arritmia usando Raspberry Pi foi desenvolvido em Valliappan et al. (2017) [14]. No trabalho, o CI (Circuito Integrado) AD8232 da SparkFun é utilizado para o condicionamento do sinal de ECG, um conversor de analógico para digital modelo MCP3008 faz a conversão do sinal, e um Raspberry Pi 3 B+ ficou responsável por amostrar e enviar o sinal por meio de WiFi. Um algoritmo de detecção de pico foi desenvolvido para obtenção de características usadas na classificação do sinal quanto ao tipo de arritmia e para exibir as informações do sinal ao usuário. O Desempenho do Algoritmo de Detecção de Pico foi testado utilizando amostras da base de dados pública MIT-BIH Arrhythmia Database, em que os testes revelaram que foi possível prever ocorrências de arritmia nessas amostras com alta precisão.

O Projeto e Implementação do Sistema de Monitoramento de ECG de Baixo Custo para o Paciente que Usa o Smartphone foi o tema do trabalho [1]. Nele, foi implementado um sistema de monitoramento de ECG de canal único, portátil e de baixo custo usando um smartphone com

sistema operacional Android e Arduino. O sistema proposto consiste em uma unidade de sensor constituída por eletrodos de ECG e cabos conectados ao paciente; uma unidade de condicionamento de sinal composta por amplificador, filtro e um circuito *leg driver*; uma fonte de alimentação e uma unidade de microcontrolador, juntamente com um módulo Bluetooth. Na etapa de amplificação foi feita uma amplificação em dois estágios, para evitar o risco de saturação do amplificador. Na filtragem, os dois filtros implementados, um passa alta com frequência de corte em 0.03Hz e outro passa baixa com frequência de corte em 159Hz, foram filtros de primeira ordem passivos. A fonte de alimentação foi construída utilizando duas baterias de 9V e dois reguladores de tensão LM7805. O Arduino foi usado para amostrar o sinal e enviar o sinal por meio do módulo bluetooth. Um smartphone foi utilizado para visualizar o sinal. O sinal foi salvo em um arquivo de texto e exibido no MATLAB R2014a para análise, que mostrou um pico em 50Hz. Na análise de custo do projeto foi calculado um custo total de USD (*United States Dollar*) 23.13. O sistema proposto funciona relativamente bem, mas a quantidade de componentes do circuito poderia ser diminuída com uma filtragem digital.

O presente trabalho propõe o desenvolvimento e implementação de um sistema (hardware e software) para a coleta do sinal de ECG via Raspberry Pi e conversor analógico/digital (AD), com filtragem digital. Optamos pela utilização do Raspberry Pi pelo seu poder de processamento e o uso separado de um conversor AD por motivos didáticos, pois assim o aluno teria a experiência de manipulação deste tipo de equipamento.

2. MATERIAL E MÉTODOS

Na Tabela 1 é exibida uma lista dos símbolos utilizadas neste trabalho.

Tabela 1: Lista dos símbolos utilizados.

Símbolo	Descrição
G, R_G	ganho obtido na amplificação do sinal e resistência utilizada na regulação do ganho de amplificação
R, V_{REF}	resolução da tensão lida pelo conversor AD e tensão de referência utilizada na conversão
$y(t), x(t), h(t)$	sinal filtrado, sinal que se deseja filtrar e resposta ao impulso do filtro, respectivamente, todos no domínio do tempo
$Y(f), X(f), H(f)$	espectro do sinal filtrado, do sinal original e do filtro, respectivamente
$y[n], x[n], h[n]$	sinal filtrado, sinal que se deseja filtrar e resposta ao impulso do filtro, respectivamente, todos no domínio do tempo discretizado
Ω, f, F_s	frequência do sinal discreto, frequência do sinal e frequência de amostragem
$E, h_d[n], M$	erro quadrático, filtro desejado e comprimento do filtro
$w[n]$	sinal do tipo janela retangular
$H_d(e^{j\Omega})$	comportamento de um filtro passa-baixas ideal no domínio da frequência
Ω_c	frequência de corte em radianos
$w_{Ham}[n]$	janela de Hamming

2.1. CIRCUITO DO SISTEMA

A coleta do sinal de ECG foi feita por um Raspberry Pi modelo B, que possui memória RAM de 512MB, processador com Clock de CPU (*Central Process Unit*) de 700MHz e 26 pinos GPIO (*General Purpose Input/Output*). O Raspberry Pi foi alimentado por uma fonte micro USB (*Universal Serial Bus*) de 5V. Para amplificação do sinal utilizamos um amplificador operacional da Analog Devices modelo AD620, e para a conversão do sinal analógico em digital utilizamos um dispositivo de 10 bits da Microchip, modelo MCP3008. O amplificador AD620 foi escolhido porque é um amplificador de instrumentação de baixo custo e alta precisão, com CMRR (*Common*

Mode Rejection Ratio) maior que 100dB próximo a frequência de 1kHz [12]. De acordo com o datasheet do AD620 o ganho é regulado por uma única resistência e a fórmula que determina o ganho é dada por

$$G = \frac{49,4k\Omega}{R_G} + 1 \quad (1)$$

onde G é o ganho e R_G é a resistência do componente utilizado para regular o ganho. Utilizamos um resistor de 150Ω o que nos proporcionou um ganho de aproximadamente 330 vezes.

O MCP3008 é o CI utilizado na conversão do sinal analógico para digital, que contém 16 pinos, dos quais 8 deles são canais de entrada de sinal, a sua resolução é de 10 bits o que pode proporcionar 1024 valores para representação das tensões lidas na entrada do conversor. A precisão desse conversor é de ± 1 LSB (*Least Significant Bit*), ou seja, de mais ou menos um bit menos significativo. As amostras do sinal convertido são enviadas por meio da interface de comunicação SPI (*Serial Peripheral Interface*). Este conversor pode trabalhar com uma frequência de amostragem de até 200ksps (200 mil amostras por segundo) com uma alimentação de 5V e 75ksps a 2,7V [13]. Para determinar a resolução da tensão lida pelo conversor, utilizou-se a relação retirada do datasheet do MCP3008 [13]:

$$R = \frac{V_{REF}}{1024} \quad (2)$$

em que a V_{REF} é a tensão de referência utilizada na conversão. Neste trabalho utilizamos o pino 3.3V do Raspberry Pi o que nos proporcionou uma resolução de 0,003223V.

O SPI é um protocolo de dados seriais síncronos utilizado para comunicação entre o microcontrolador e um ou mais periféricos. Este protocolo exige que um dos equipamentos seja o *master*, enquanto todos os outros são *slave*. No nosso trabalho o Master é o Raspberry Pi e o Slave é o MCP3008. Tal comunicação necessita de 4 pinos: MOSI (*Master Output Slave Input*), MISO (*Master Input Slave Output*), CLK (*Clock*) e CS (*Chip Select*). O MOSI é o pino de onde é enviado dados do Raspberry Pi para o MCP3008, e por meio dele é selecionado o canal de entrada do conversor que deve ser usado; o MISO é o pino utilizado para receber o resultado da conversão; CLK é o pino utilizado para definir o *clock* da comunicação (no Raspberry Pi essa velocidade pode ser de até 3,9 Mbits); e CS é o pino utilizado para selecionar o Slave que deve fazer a conversão. A Figura 3 exhibe o circuito montado para amplificação e conversão de analógico para digital do sinal.

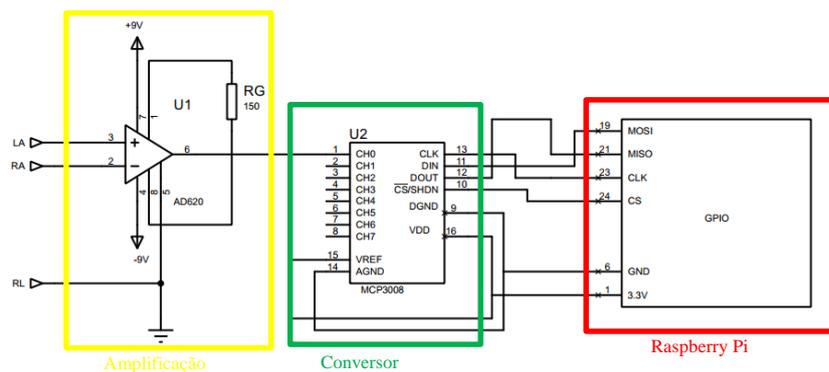


Figura 3: Circuito de amplificação e aquisição.

Foi implementado um algoritmo em linguagem C++ que faz a coleta e filtragem do sinal.

2.2. FILTRAGEM DIGITAL

Filtragem é o processo no qual selecionamos componentes espectrais de um sinal, e, conseqüentemente, descartamos outras [8]. Esta operação é requisitada em aplicações em que se deseja eliminar ruídos, ou destacar características de um sinal, por exemplo. De acordo com a região do espectro selecionada, um filtro pode ser dito passa-baixas, passa-altas, passa-faixa ou rejeita-faixa [9]. Neste trabalho utilizaremos um filtro passa-baixas, que seleciona as baixas frequências, a fim de eliminar a parcela de ruído nas altas frequências, presente na aquisição.

A filtragem é comumente representada no domínio só tempo como uma convolução entre o sinal que se deseja filtrar $x(t)$ e a resposta ao impulso do filtro $h(t)$, como em

$$y(t) = x(t) * h(t) \quad (3)$$

em que $y(t)$ representa o sinal $x(t)$ após a filtragem. Pela propriedade de convolução da Transformada de Fourier e citado por Oppenheim e Schaffer (2009) [8], esta convolução no tempo torna-se uma multiplicação no domínio espectral

$$Y(f) = X(f)H(f) \quad (4)$$

onde $Y(f)$, $X(f)$ e $H(f)$ representam os espectros do sinal filtrado, do sinal original e do filtro, respectivamente. Note que nesta representação, a resposta em frequência do filtro multiplica o espectro do sinal original, de forma que nas regiões em que $H(f)$ possui baixa magnitude, as componentes espectrais correspondentes em $X(f)$ serão atenuadas, e de modo contrário, as outras regiões do espectro serão selecionadas.

Este processo pode ser representado de maneira analógica (através de circuitos elétricos) ou digital (com algoritmos rodando em um computador). Para a aplicação em questão, optamos por aplicar apenas o filtro digital, dado que as características espectrais do sinal processado (o ECG) são bem documentadas na literatura, e assim podemos definir os parâmetros de um filtro digital de forma adequada a manter a essência do sinal manipulado [9]. Como adaptação a representação de sinais digitais, reescrevemos a formulação de filtragem trocando a dependência do tempo contínuo, para uma que leve em conta um tempo discretizado

$$y[n] = x[n] * h[n] \quad (5)$$

Os sinais discretos no tempo, possuem frequências numa escala de $\Omega = -\pi$ a $\Omega = \pi$, em que π corresponde à metade da frequência de amostragem. A equação que permite relacionar as frequências analógicas com as digitais é

$$\Omega = \frac{2\pi f}{F_s} \quad (6)$$

em que f é a frequência do sinal em Hz, e F_s é a frequência de amostragem.

Na categoria de filtros digitais, podemos ainda destacar dois grupos: os com Resposta ao Impulso Infinita (RII) e os com Resposta ao Impulso Finita (RIF). Os filtros RIF possuem algumas vantagens em relação aos RII:

1. Estes filtros tem memória limitada;
2. São sempre BIBO (*Bounded Input-Bounded Output*) estáveis;
3. Conseguem implementar um sistema com fase linear na região de banda passante, que é necessário para a realização de uma filtragem sem distorção [10].

No desenvolvimento de filtros utilizamos o um procedimento em duas etapas, que são a definição do filtro ideal que se deseja alcançar e aplicação de janelamento a este filtro desejado.

Deixe-nos representar o filtro desejado por $h_d[n]$, o qual possui características ideais como ganho constante na banda passante e banda de transição de largura zero. Tais características são

irrealizáveis, de forma que o filtro desenvolvido $h[n]$ de comprimento M será uma estimativa de $h_d[n]$.

O desvio nesta estimação pode ser avaliado com o erro quadrático

$$E = \sum_{-\infty}^{\infty} |h_d[n] - h[n]|^2 \quad (7)$$

Na equação acima os únicos parâmetros ajustáveis são os valores de $h[n]$, de forma que o erro será mínimo quando

$$h[n] = \begin{cases} h_d[n], & 0 \leq n \leq M \\ 0, & \text{outro caso} \end{cases} \quad (8)$$

Este procedimento pode ser representado pelo janelamento do sinal $h_d[n]$, como em

$$h[n] = h_d[n]w[n] \quad (9)$$

em que $w[n]$ é um sinal do tipo janela retangular dado por

$$w[n] = \begin{cases} 1, & 0 \leq n \leq M \\ 0, & \text{outro caso} \end{cases} \quad (10)$$

O próximo passo é determinar o comportamento no domínio do tempo do filtro desejado. O comportamento de um filtro passa-baixas ideal é definido no domínio da frequência como

$$H_d(e^{j\Omega}) = \begin{cases} e^{-j\Omega M/2}, & \Omega \leq \Omega_c \\ 0, & \text{fora} \end{cases} \quad (11)$$

em que Ω_c é a frequência de corte em radianos. A partir deste espectro podemos definir o seu comportamento no domínio do tempo aplicando a transformada inversa de Fourier, que rende

$$h_d[n] = \frac{\Omega_c}{\pi} \text{sinc} \left[\frac{\Omega_c}{\pi} \left(n - \frac{M}{2} \right) \right], \quad -\infty \leq n \leq \infty \quad (12)$$

Em seguida tomamos M amostras deste sinal utilizando janelamento. Na Eq. (9) utilizamos a janela retangular para representar esta ação de tomada de M amostras. Todavia, este tipo de janela não é o mais indicado para manipulação digital de sinais truncados, devido as suas características espectrais [10]. Para este tipo de aplicação, uma janela que apresenta resultados satisfatórios é a janela de Hamming, dada por:

$$w_{Ham}[n] = \begin{cases} 0,54 - 0,46 \cos \left(2\pi \frac{n}{M} \right), & 0 \leq n \leq M \\ 0, & \text{outro caso} \end{cases} \quad (13)$$

Finalmente, a estimativa do filtro pode ser representada como a multiplicação entre $h_d[n]$ e $w_{Ham}[n]$, ou seja:

$$h[n] = \begin{cases} \frac{\Omega_c}{\pi} \text{sinc} \left[\frac{\Omega_c}{\pi} \left(n - \frac{M}{2} \right) \right] \left[0,54 - 0,46 \cos \left(2\pi \frac{n}{M} \right) \right], & 0 \leq n \leq M \\ 0, & \text{outro caso} \end{cases} \quad (14)$$

3. RESULTADOS E DISCUSSÃO

Nesta seção apresentaremos os resultados do protótipo em relação a aquisição e filtragem do sinal de ECG.

A frequência de amostragem definida para a conversão analógico/digital foi de 3000 amostras por segundo, a qual, de acordo com Haykin e Veen (2002) [11] é controlada pela tensão fornecida ao pino V_{REF} do MCP3008. Nessas condições, o sistema é capaz de representar sinais com frequências de até 1500 Hz, ou seja, $F_s/2$. Tal valor de amostragem é suficiente para representar a parte essencial das características do sinal de ECG [8]. Na verdade, este valor é bem maior que o necessário para aquisição de ECG, mas como não aplicamos uma etapa de filtragem analógica, é aconselhável sobreamostrar o sinal para então aplicar a filtragem digital [9].

Para o filtro passa-baixas definimos uma frequência de corte de 100 Hz, ou, 0,209 rad, e ordem 100. As Figuras 4 e 5 mostram a resposta ao impulso e a resposta em frequência deste filtro, respectivamente. Note que no domínio do tempo este filtro se comporta como um sinal do tipo *sinc* escalado, como definido pela Eq. 14. Já no domínio da frequência podemos observar o seu formato que se aproxima de um pulso retangular (que é a forma do espectro do filtro desejado), no qual observamos que a banda passante (região de ganho unitário, ou de 0 db) é da largura 0,209 rad, indicando que as componentes espectrais nessa região serão mantidas, e as acima serão atenuadas.

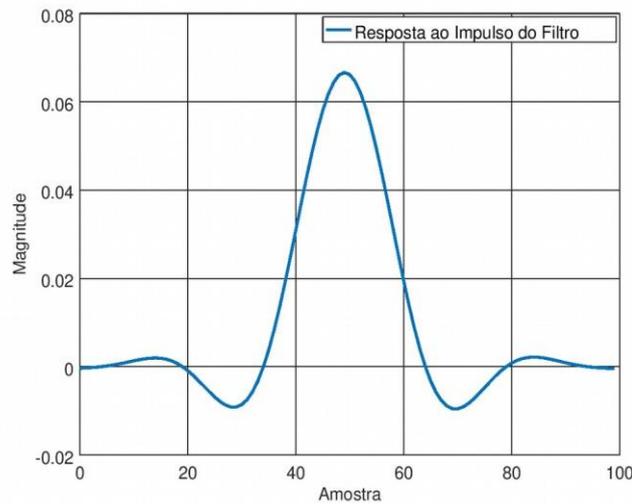


Figura 4: Resposta ao impulso do filtro passa-baixas.

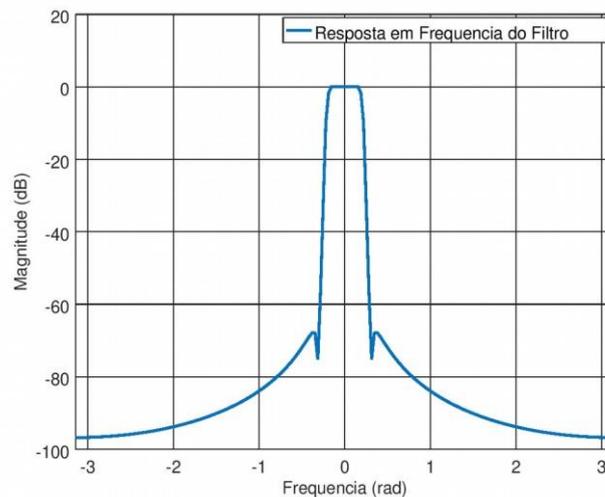


Figura 5: Resposta em frequência do filtro passa-baixas, com frequência de corte de 0,209 rad (100 Hz).

Para teste do sistema, executamos a medição de ECG com um aluno da UNIFESSPA (Universidade Federal do Sul e Sudeste do Pará). No protótipo desenvolvido precisamos acoplar

três eletrodos ao corpo da pessoa, um em cada pulso (configurados em modo diferencial) e outro na perna, como mostrado na Figura 6. Tal configuração permite ao sistema medir o sinal de ECG no pulso, tomando o sinal proveniente da perna com terra. O fato dos dois sensores do sinal de ECG estarem em modo diferencial é uma primeira medida para reduzir os efeitos de interferências.



Figura 6: Utilização do protótipo para medição de ECG em uma pessoa.

Na Figura 7 temos as 10000 amostras de uma aquisição de ECG realizada com este aluno. A curva neste gráfico possui as características gerais de um sinal de ECG, mas com um grande nível de degradação devido os ruídos presentes na aquisição. Após a aplicação do filtro o sinal de ECG passa a comportar como na Figura 8, na qual notamos uma redução significativa na variabilidade do sinal, rendendo assim uma curva mais suave. Este resultado confirma a efetividade da filtragem passa-baixas, a qual eliminou grande parte das oscilações rápidas devido as componentes espectrais de alta frequência do ruído.

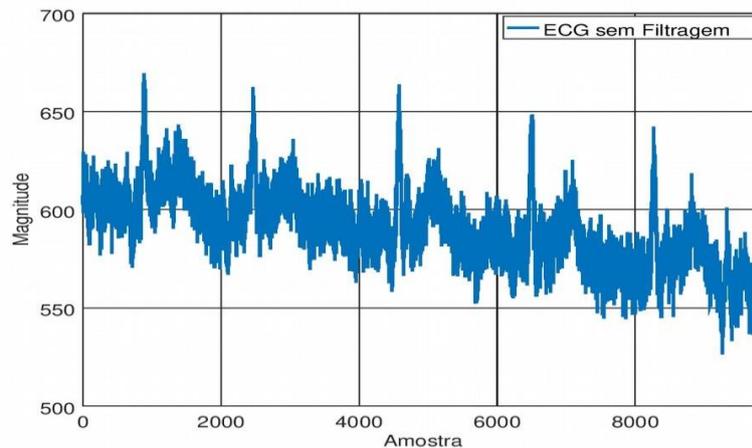


Figura 7: Sinal de ECG adquirido sem aplicação de filtro passa-baixas.

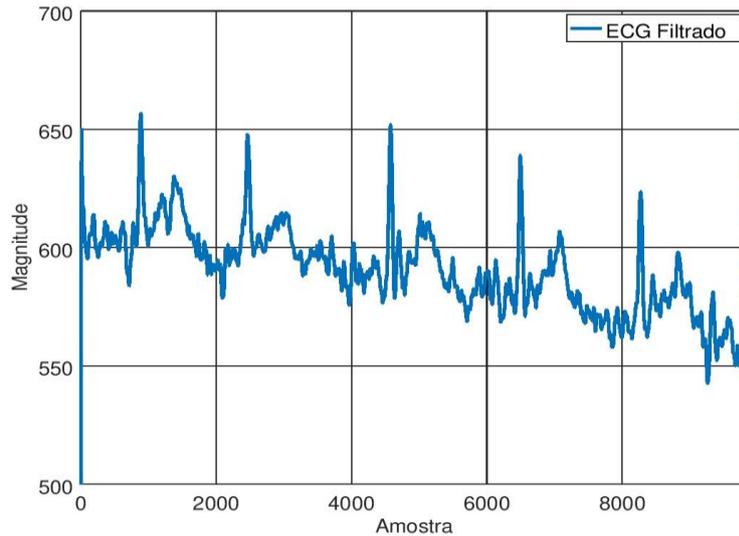


Figura 8: Sinal de ECG após a filtragem passa-baixas.

Apesar do efeito coerente da filtragem, avaliações futuras são necessárias para atestar que estes procedimentos mantêm as características do ECG que são importantes para diagnósticos feitos tanto por médicos especializados, como por algoritmos automáticos de classificação.

4. CONCLUSÃO

Neste trabalho apresentamos os passos iniciais do desenvolvimento de um aparelho capaz de medir ECG, com aplicação de filtragem digital, e possível utilização de algoritmos automáticos de detecção de características de ECG. Neste protótipo utilizamos um conversor analógico/digital separado do processador e não utilizamos filtragem analógica. O principal resultado alcançado nesta etapa do projeto foi a filtragem do sinal de ECG adquirido de uma pessoa, na qual observou-se uma redução significativa no nível de ruído do sinal após o processamento.

Os trabalhos futuros incluem a aplicação de algoritmos para detecção de frequência cardíaca e de detecção automática de anomalias cardíacas através do uso de modelos de inteligência artificial.

5. REFERÊNCIAS BIBLIOGRÁFICAS

1. Ahamed A, Hasan K, Alam S, editors. Design and implementation of low cost ECG monitoring system for the patient using smartphone. 2015 International Conference on Electrical & Electronic Engineering (ICEEE); 2015 Nov 4 - Nov 6; Rajshahi, Bangladesh: IEEE; 2015 Nov. 261p, doi: 10.1109/ICEEE.2015.7428272.
2. Al-Busaidi AM, Khriji L, editors. Digitally filtered ECG signal using low-cost microcontroller. 2013 International Conference on Control, Decision and Information Technologies (CoDIT); 2013 May 6 – May 8; Hammamet, Tunisia: IEEE; 2013 May. 259p, doi: 10.1109/CoDIT.2013.6689554.
3. Hampton JR. The ECG Made Easy. 6rd. ed. London, United Kingdom: Elsevier Health Sciences; 2003. 160 p.
4. Tompkins WJ. BIOMEDICAL DIGITAL SIGNAL PROCESSING: C-Language Examples and Laboratory Experiments for the IBM® PC. 6rd. ed. Upper Saddle River, Nova Jersey: Prentice Hall; 1993. 368 p.
5. Gilani SO, Ilyas Y, Jamil M. Power line noise removal from ECG signal using notch, band stop and adaptive filters. 2018 International Conference on Electronics, Information, and Communication (ICEIC); 2018 Jan 24 - Jan 27; Honolulu, HI, USA:IEEE; 2018 Jan. 1p, doi:10.23919/ELINFOCOM.2018.8330569
6. Yildirim Ö, Plawiak P, Tan RS, Acharya UR. Arrhythmia Detection Using Deep Convolutional Neural Network With Long Duration ECG Signals. *Comput Biol Med.* 2018 Nov;102:411–420, doi:10.1016/j.combiomed.2018.09.009.
7. Deopujari SP, Gaikwad A. Light Weight, Low Cost, Wearable Ecg Monitoring. *Int J Comput Eng Res.* 2013 Jul; 3(7):1-6.

8. Oppenheim AV, Schafer RW. Discrete-Time Signal Processing: Prentice-Hall Signal Processing Series. 3rd. ed. Upper Saddle River, NJ, USA: Pearson; 2009. 1144 p.
9. Sklar B. Digital Communications: Fundamentals and Applications. 2rd. ed. Upper Saddle River, N.J: Prentice Hall; 2017. 1104 p.
10. Molloy D. Exploring Raspberry Pi: interfacing to the real world with embedded Linux. 2rd. ed. Indianapolis, IN: John Wiley & Sons; 2016. 720 p.
11. Haykin S, Veen BV. Signals and Systems. 2rd. ed. New York: Wiley; 2002. 802 p.
12. Devices A. Low Cost Low Power Instrumentation Amplifier AD620 [Internet]. Available from: <<https://www.analog.com/media/en/technical-documentation/data-sheets/ad620.pdf>>.
13. Technology, M. MCP3004/3008 [Internet]. Available from: <<https://cdn-shop.adafruit.com/datasheets/MCP3008.pdf>>.
14. Valliappan C A, Balaji A, Thandayam SR, Piyush D, Baths V. A Portable Real Time ECG Device for Arrhythmia Detection Using Raspberry Pi. Wireless Mobile Communication and Healthcare (MobiHealth 2016). 2017 Jun;192:177-184, doi:10.1007/978-3-319-58877-3_24.
15. Moukabary T. Willem Einthoven (1860-1927): Father of electrocardiography. Cardiol J. 2007 Jan; 14(3):316-317.

APÊNDICE I

Apêndice 1 – Lista das siglas utilizadas.

Sigla	Descrição
ECG	Eletrocardiograma
LMS	<i>Least Mean Square</i>
CI	Circuito Integrado
USD	<i>United States Dollar</i>
AD	Analógico/Digital
CPU	<i>Central Process Unit</i>
GPIO	<i>General Purpose Input/Output</i>
USB	<i>Universal Serial Bus</i>
CMRR	<i>Common Mode Rejection Ratio</i>
LSB	<i>Least Significant Bit</i>
SPI	<i>Serial Peripheral Interface</i>
MOSI	<i>Master Output Slave Input</i>
MISO	<i>Master Input Slave Output</i>
CLK	<i>Clock</i>
CS	<i>Chip Select</i>
RII	Resposta ao Impulso Infinita
RIF	Resposta ao Impulso Finita
BIBO	<i>Bounded Input-Bounded Output</i>
UNIFESSPA	Universidade Federal do Sul e Sudeste do Pará

APÊNDICE II

```
#include <iostream>
#include <sstream>
#include <math.h>
#include "bus/SPIDevice.h"
#define SAMPLES 10000 //numero de amostras
#define M 100
#define OHMC 0.209 // frequencia de corte de 100 Hz cosiderando fs=3000 e espectro normalizado
#define PI 3.1415

using namespace std;
using namespace exploringRPi;

float sinc(float x){

    float sincx;
    if (x==0)
        x = (x+0.000001);
```

```

    sincx = sin((x*PI)/180)/((x*PI)/180);
    return sincx;
}

short combineValues(unsigned char upper, unsigned char lower){
    return ((short)upper<<8)|(short)lower;
}

int main(){
    SPIDevice *busDevice = new SPIDevice(0,0); //Using second SPI bus (both loaded)
    busDevice->setSpeed(240000); // Have access to SPI Device object
    busDevice->setMode(SPIDevice::MODE0);

    unsigned char send[3], receive[3];
    float samples[SAMPLES];

    send[0] = 0b00000001; // The Start Bit followed
    // Set the SGL/Diff and D mode -- e.g., 1000 means single ended CH0 value
    send[1] = 0b10000000; // The MSB is the Single/Diff bit and it is followed by 000 for CH0
    send[2] = 0; // This byte doesn't need to be set, just for a clear display
    //as linhas de 39 a 42 configuram a comunicação SPI
    float h[M+1]; //vetor com os coeficientes da janela
    for(int i=0; i<=M; i++){
        h[i]=(OHMC/PI)*sinc((OHMC/PI)*(i-M/2))*(0.54-(0.46*cos(((2*PI*i)/M)*(PI/180))));
    } //preenchimento da janela com M+1 amostras

    float buffer[M+1]; //criação do buffer com M+1 amostras
    for(int i=0; i<=M; i++){
        buffer[i]=0; //preenchimento do buffer com zeros
    }

    for(int i=0; i<SAMPLES; i++){
        for(int j=M+1; j>0;j--){
            buffer[j] = buffer[j-1];
        }

        busDevice->transfer(send, receive, 3); // prepara o conversor para fazer a amostragem de 1
        amostra
        buffer[0] = (short((receive[1]&0b00000011)<<8)|(short(receive[2]))); //ultima posição do
        buffer recebe a amostra corrente

        int samples_f = 0;

        for(int k=0; k<=M;k++){ //executa a filtragem das amostras do buffer atraves de
        uma convolução
            samples_f = samples_f + (buffer[k]*h[k]);
        }

        samples[i] = samples_f; //preenche a posição i com a amostra filtrada
    }

    for(int i=0; i<SAMPLES; i++){
        cout << i << " " << samples[i] << endl;
    }
}

```