

# Processamento de Sinais Biomédicos - Tarefa 2

## ECGLab

Fausto E. Brenner - 17/0009777

Lucas Ferreira - 17/0039641

Natália Backhaus - 16/0015766

### I. INTRODUÇÃO

Para este estudo, utilizaremos a *toolbox* ECGLab, de autoria do Prof. João Luiz Azevedo de Carvalho, para realizar a análise espectral de sinais de ECG do banco de dados *Fantasia*, hospedado no site [Physionet](#). Foram selecionados dois arquivos de pacientes do banco de dados: *f2y04m.mat* e *f2o04m.mat*.

#### A. ECGLab

Antes de aplicar as funcionalidades da *toolbox*, é interessante discorrer sobre seu funcionamento. Nosso objetivo final é realizar uma análise espectral do ECG. Para isso, passaremos por etapas iniciais de detecção de ondas R e estimação de intervalos RR para depois utilizar esses resultados na análise espectral do sinal R-R.

1) *ECGLabRR*: Para iniciar o módulo ECGLabRR basta digitar na tela de comando do Matlab

```
cd C:\ecglab
ecglabRR
```

Esse módulo recebe um arquivo *.mat* como entrada e aplica algoritmos para identificar as ondas R do ECG. Existe duas opções de algoritmo, um rápido e menos preciso e outro mais lento que gera resultados mais precisos. O algoritmo rápido se baseia em passar o sinal de ECG por uma série de filtros e amplificações para gerar um sinal de referência com picos dentro dos intervalos de interesse. Abaixo pode-se analisar um trecho do arquivo *ecglabRR\_marca\_ondar2.m* responsável por essa etapa. A Figura 1 apresenta um sinal de ECG de exemplo antes e após a aplicação do trecho de código destacado abaixo.

```
1 %filtra em 17Hz
2   ecg_filtrado=ecglabRR_filtra17hz(ecg_sinal,3,1.2); %filtra17hz(sinal,Q,ganho)
3
4 %deriva
5   %ecg_filtrado=1/8*filter([2 1 -1 -2],1,ecg_filtrado); %[1 -1]; %esta equacao esta errada
6
7 %derivada
8   ecg_filtrado=filter([1 -1],1,ecg_filtrado);
9 %passa baixas 30 Hz (para nao enriquecer ruido)
10  [B,A]=butter(8,30/(samplerate_ecg/2));
11  ecg_filtrado=filter(B,A,ecg_filtrado);
12  ecg_filtrado = ecg_filtrado /max(abs(ecg_filtrado)); %ganho
13
14 %eleva ao quadrado
15  ecg_filtrado=ecg_filtrado.^2;
16
17
18 %integrador (janela movel)
19  N=round(0.150*samplerate_ecg); ecg_filtrado=1/N*filter(ones(1,N),1,ecg_filtrado);
```

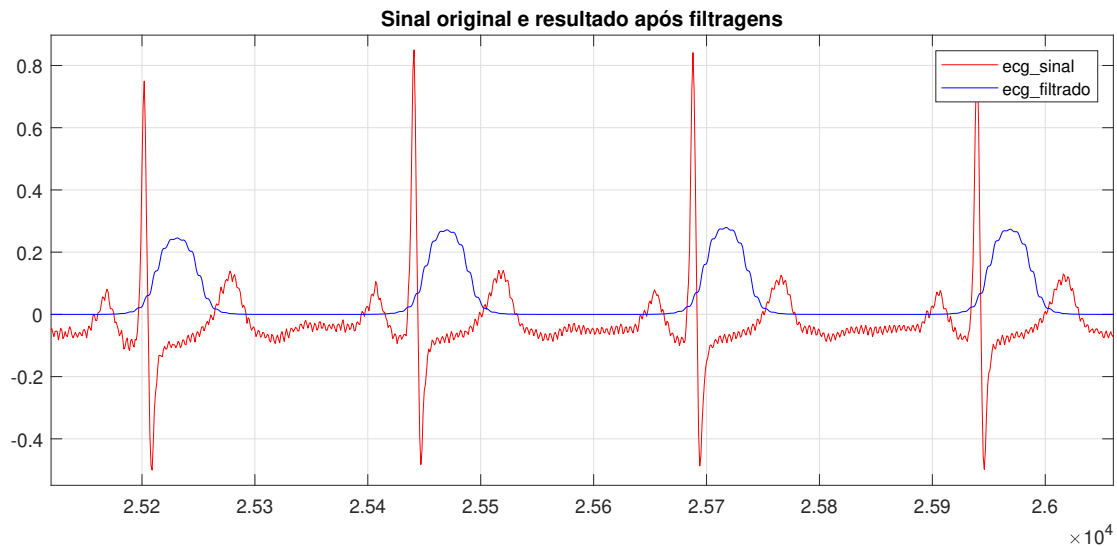


Figura 1: Sinal ECG original e o resultado após as filtrações do processo de detecção de onda R.

Então, o algoritmo `ecglabRR_marca_ondar2.m` percorre os sinais filtrado e original, comparando os valores dos picos dos sinais para identificar a localização dos picos (ondas R) no sinal original a partir de parâmetros determinados. Ao final da operação, é criado um arquivo `.irr`, contendo os valores dos intervalos entre ondas R. Esse arquivo será utilizado em nossa análise espectral.

2) *espectralRR*: Para iniciar o módulo `espectralRR` basta digitar na tela de comando do Matlab

```
cd C:\ecglab
espectralrr
```

Este módulo realiza o cálculo do espectrograma do sinal R-R gerado após a detecção das ondas R no módulo anterior. A função `espectralRR_espectrograma.m` realiza a estimação do espectrograma que, neste caso, pode ser feita com base em dois métodos, *AR model* e *Fourier Transform*, que iremos discutir a seguir.

O método descrito no programa como *AR model* se baseia em utilizar uma estimação de parâmetros de modelo auto regressivo (AR) por meio do método de Burg, utilizando a função `arburg()`. A partir desses parâmetros a PSD do sinal pode ser estimada a partir da resposta em frequência de um filtro utilizando a função `freqz()`. O trecho de código responsável pela estimação da PSD do sinal R-R pelo método AR pode ser visto abaixo.

```
1 [A, variancia] = arburg(intervaloRR, ordem_ar);
2 [H, F] = freqz(1, A, N, fs);
3 PSD = (abs(H).^2) * variancia * 1/fs; %malik, p.67
```

A outra forma de estimação é baseado na transformada rápida de Fourier (fft), representada no Matlab pela função `fft()`. Mais informações sobre o funcionamento desse algoritmo podem ser encontradas na [documentação do Matlab](#). O trecho do código que realiza a estimação do espectrograma do sinal a partir da transformada de Fourier pode ser visto a seguir.

```
1 PSD = (abs(fft(intervaloRR, N)).^2) / length(intervaloRR) / fs; %*fs
2 F = (0:fs/N:fs-fs/N)';
3 PSD = PSD(1:ceil(length(PSD)/2));
4 F = F(1:ceil(length(F)/2));
5 maxPSD = max(PSD);
6 minPSD = min(PSD(2:length(PSD)));
```

Após o cálculo da PSD do sinal, independente do método escolhido, calcula-se as áreas de muito baixa frequência (*very low frequency*), baixa frequência (*low frequency*) e alta frequência (*high frequency*) por meio do algoritmo `espectralRR_calcula_areas.m`. Essa rotina funciona de forma simples. Primeiramente, calcula a energia total presente em cada uma das regiões por meio do produto do eixo das frequências pelos valores de PSD de cada frequência dentro de determinado intervalo, como pode-se ver no código abaixo. Isso é a mesma coisa que calcular a área abaixo do gráfico da PSD encontrada.

```
1 % calcula a energia total (de 0 a hf2) em ms^2
2 aatotal=F(2)*sum(PSD(1:indice_hf-1));
3
4 %calcula a energia de freq. muito baixas (de 0 a vlf2)
5 aavlf=F(2)*sum(PSD(1:indice_vlf-1));
6
7 %calcula a area de baixas frequencias (de vlf2 a lf2)
8 aalf=F(2)*sum(PSD(indice_vlf:indice_lf-1));
9
10 %calcula a area de altas frequencias (de lf2 a hf2)
11 aahf=F(2)*sum(PSD(indice_lf:indice_hf-1));
```

Então, as áreas relativas são calculadas dividindo as áreas de cada setor pela área total. A razão LF/HF é calculada dividindo a área de baixa frequência pela área de alta frequência. As áreas normalizadas de baixa frequência e alta frequência são calculadas pela seguinte operação.

```
1 %calcula as areas normalizadas
2 anlf=aalf/(aalf+aahf);
3 anhf=aahf/(aalf+aahf);
```

## II. DESENVOLVIMENTO

### A. Instalação da Toolbox

A versão mais recente da *toolbox* foi baixada pelo link:

[ecglab\\_matlab2014a.zip.a](#)

O arquivo baixado veio com terminação `.zip.a`. Entendendo que isso se tratou de algum erro, alterou-se, manualmente, a nomeação do arquivo para ter terminação `.zip`. Então, o processo de instalação foi finalizado no diretório `C:\ecglab`. Adicionou-se, ao diretório instalado, o arquivo `rdsign212.mexw64`, conforme orientação da professora.

### B. Recuperação Dados de ECG

Os arquivos `f2y04m.mat` e `f2o04m.mat`, contendo dados de RESP, ECG e BP de dois pacientes foram baixados no site [Physionet](#). Utilizando a rotina `get5min`, desenvolvida na tarefa anterior, foi possível recuperar 5 minutos dos dados de ECG e a frequência de amostragem do arquivo. Esses dois dados foram salvos em um novo arquivo `.mat` nos padrões esperados pelo ECGLab. Para a formatação do arquivo `.mat` no formato esperado pelo ECGLab, utilizou-se o script `makeECGfile.m`, de autoria própria, gerando os arquivos `ecg_o.mat` e `ecg_y.mat`.

### C. Detecção IRR

O arquivo `.mat` gerado na etapa anterior foi utilizado como entrada no módulo ECGLabRR, para a detecção de onda R. Primeiramente, o algoritmo rápido foi aplicado no sinal, como o resultado já foi satisfatório, detectando todas as ondas R, não foi necessária a utilização do algoritmo mais lento. Analisou-se os pontos em busca de batimentos ectópicos, porém o *software* não detectou nenhum batimento anormal. O módulo `OutliersRR` também foi utilizado para verificar a presença de batimentos irregulares, porém todos os batimentos foram percebidos como normais (Figura 3). Ao final das análises, os arquivos `ecg_o.irr` e `ecg_y.irr` foram gerados contendo a série de intervalos R-R.

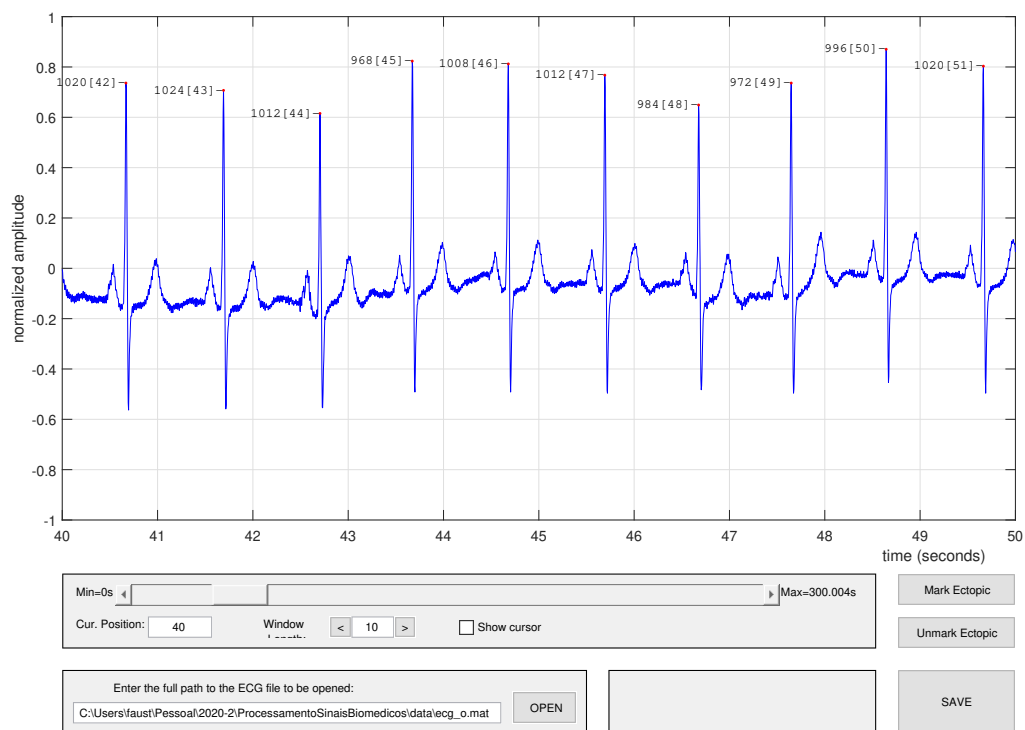


Figura 2: Módulo ECGLabRR após detecção de onda R do paciente f2o04m

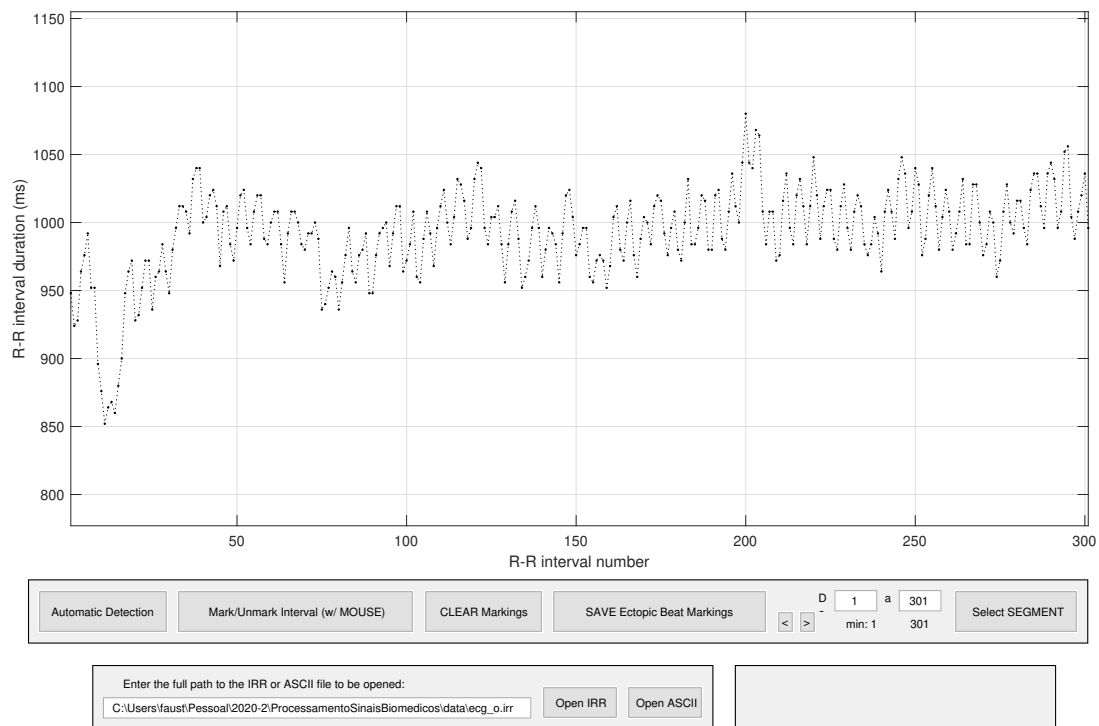


Figura 3: Módulo OutliersRR demonstrando que nenhum batimento apresentou características de ectopia

#### D. Análise no Domínio da Frequência

O arquivo `.irr` gerado ao final da etapa anterior foi utilizado como entrada no módulo `spectralRR`. O resultado, contendo ambos os métodos de cálculo, com janelamento retangular, 8192 pontos e ordem de AR de 150 (máx.), obteve-se o resultado da Figura 4. Fazendo alterações nesses valores, nota-se como a transformada de Fourier é mais sensível ao aumento de pontos, uma vez que ao aumentar esse valor, aumenta-se a resolução mas, por outro lado, o espectro de torna mais irregular. Quanto à variável "Ordem AR", quanto maior esse valor, mais próximo o modelo fica do espectro real do sinal. Com a configuração padrão, os limites das áreas de muito baixas, baixas e altas frequências são, respectivamente, 0.05, 0.15 e 0.4. Como o eixo das frequências está normalizado com base na frequência de amostragem (250 Hz no nosso caso), os limites em Hz são 12.5 Hz, 37.5 Hz e 100 Hz, respectivamente. Avaliando os valores de área obtidos, nota-se que existe uma maior concentração de energia nas frequências muito baixas, com cerca de 60% da energia total do sinal. Somando as áreas de muito baixa frequência e baixa frequência, conclui-se que cerca de 74% da energia está concentrada nessas frequências, contra 26% das altas frequências.

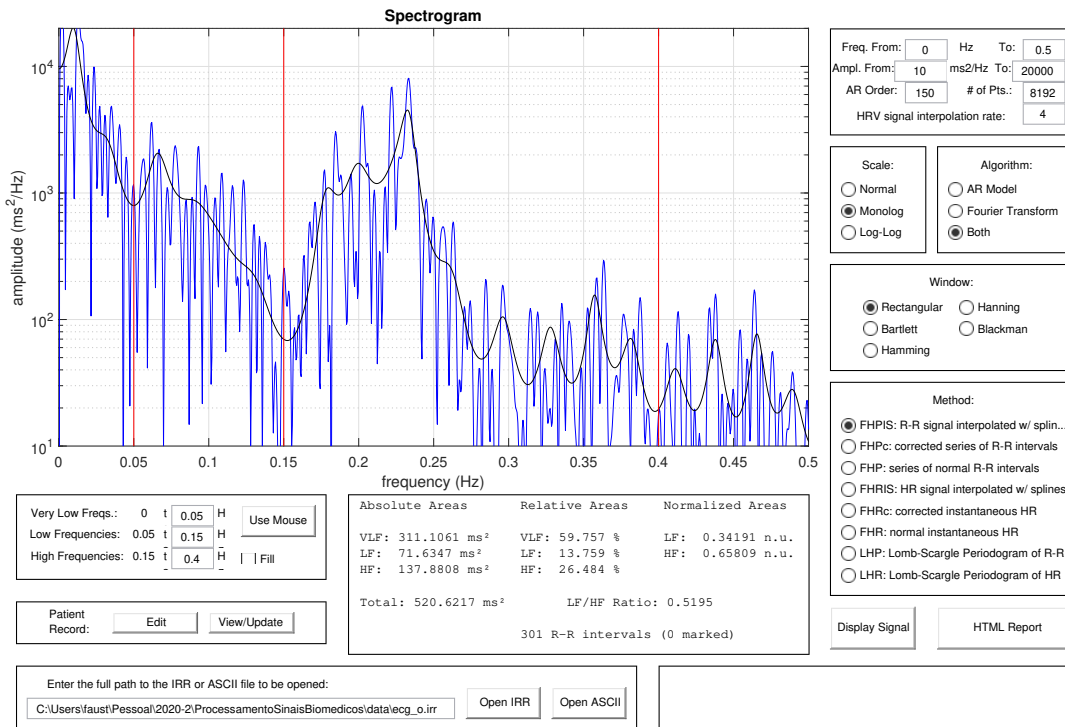


Figura 4: Módulo `spectralRR` apresentando o resultado da nossa análise espectral

### III. CONCLUSÃO

Nesta atividade, foi possível demonstrar a aplicabilidade da *toolbox* `ECGLab` para realizar o processamento e a análise de sinais de ECG. Realizou-se uma análise espectral completa do sinal, concluindo-se que existe uma maior concentração de energia nas baixas frequências do espectro do ECG.