

Informática Médica



TRABALHO PRÁTICO DE PROCESSAMENTO DE SINAL E IMAGEM BIOMÉDICA I

Trabalho Prático I

José André Gomes Pinto nº 13273

Daniel Simão de Carvalho nº 13286

Índice

Introdução	3
Exercício 1	4
Exercício 2	7
Exercício 3	8
Conclusão	12

Introdução

No âmbito da unidade curricular de Processamento de Sinal e Imagem Biomédica do primeiro semestre do terceiro ano curricular do curso de Informática Médica do Instituto Politécnico do Cavado e do Ave, no ano letivo 2018/2019 foi-nos proposto, pela referida unidade curricular, a elaboração de um projeto para colocar em prática todos os conhecimentos adquiridos nas aulas.

Este trabalho surge no contexto de avaliação continua da unidade curricular, sendo esta avaliação composta pelos testes e pelos trabalhos práticos onde se engloba este projeto agora elaborado. Com este projeto prático pretendemos colocar em prática e aprofundar todos os nossos conhecimentos adquiridos nas aulas de modo a sermos melhor sucedidos em trabalhos futuros.

O trabalho elaborado e executado pelo grupo, tem como objetivo principal elaborar um processamento de um sinal biomédico recolhido através de um eletroencefalograma (EEG) referente à resposta evocada visual gravada pelo córtex visual após um estímulo.

Para a realização deste trabalho prático foram tomados em conta os principais objetivos apontados pelo respetivo docente da unidade curricular, de modo a elaborar o trabalho prático da melhor forma e que este seja uma mais valia para trabalhos futuros e também para uma futura vida profissional.

Para elaborar este projeto foram utilizadas duas ferramentas utilizadas em contexto de aula, sendo elas o Matlab e o Octave. Com o auxílio destas duas ferramentas foram elaborados scripts para a realização de cada uma das questões propostas neste trabalho prático.

Exercício 1

Considerando o sinal da variável `sem_ruido` (intervalo de amostragem é 5 ms), apresente um script em MATLAB, devidamente comentado, que obedeça aos seguintes requisitos:

a) Calcule e apresente na linha comandos as seguintes medições do sinal:

Neste exercício, antes do cálculo da média, valor eficaz, variância e desvio padrão, executamos sempre o 'load' de `biossinal` dado pelo docente. De seguida calculamos o 'length' do sinal 'sem ruido' (numero de amostras) que será importante para a criação do vetor de tempo entre 0 e 99, contendo, desta forma, 100 amostras.

De seguida executamos os seguintes comandos:

- '`mean(sem_ruido)`' – para o cálculo da média
- '`sqrt(mean(sem_ruido.^2))`' – para o cálculo do valor eficaz
- '`(sum((sem_ruido-mean(sem_ruido)).^2) / (N-1))`' – para o calculo da variância
- '`sqrt(variância)`' – para o cálculo do Desvio padrão
- '`disp()`' – comando que permite que o resultado final seja apresentado na linha de comandos.

As imagens seguintes mostram o sucesso na execução dos comandos.

i. Média:

```

1 - clear all;
2 - close all;
3
4 - load('biossinal.mat'); %Carrega o ficheiro biossinal.mat fornecido para a realização do trabalho
5
6 - fs = 1/0.005; %Frequencia do sinal com intervalo de amostragem 5ms
7 - N = 100; %length(sem_ruido) = 100; logo N = 100; Numero de amostras
8 - t = (0:N-1)/fs; %Cria o vetor de tempo para valores entre 0 e N-1 (0 a 99)
9
10
11
12 %Calcular a Média do sinal%
13
14 - Media = mean(sem_ruido); %Calculo da media do sinal
15 - disp(Media); %Apresenta na consola o valor da variavel Media (Media do sinal)

>> Exerciciol_TrabalhoPratico
-0.0968
  
```

Figura 1 – Média do Sinal “Sem_Ruido”;

ii. Valor Eficaz (RMS):

```

FILE NAVIGATE EDIT BREAKPOINTS RUN
Exercicio1_TrabalhoPratico.m
1- clear all;
2- close all;
3
4- load('biossinal.mat'); %Carrega o ficheiro biossinal.mat fornecido para a realização do trabalho
5
6- fs = 1/0.005; %Frequencia do sinal com intervalo de amostragem 5ms
7- N = 100; %length(sem_ruido) = 100; logo N = 100; Numero de amostras
8- t = (0:N-1)/fs; %Cria o vetor de tempo para valores entre 0 e N-1 (0 a 99)
9
10
11
12 %Calcular o Valor Eficaz (RMS) do sinal%
13
14- RMS = sqrt(mean(sem_ruido.^2)); %Calculo do valor eficaz do sinal
15- disp(RMS); %Apresenta na consola o valor da variavel RMS (Valor eficaz)

>> Exercicio1_TrabalhoPratico
0.3360

```

Figura 2 – Valor Eficaz (RMS) do Sinal “Sem_Ruido”;

iii. Variância e Desvio Padrão:

```

FILE NAVIGATE EDIT BREAKPOINTS RUN
Exercicio1_TrabalhoPratico.m
1- clear all;
2- close all;
3
4- load('biossinal.mat'); %Carrega o ficheiro biossinal.mat fornecido para a realização do trabalho
5
6- fs = 1/0.005; %Frequencia do sinal com intervalo de amostragem 5ms
7- N = 100; %length(sem_ruido) = 100; logo N = 100; Numero de amostras
8- t = (0:N-1)/fs; %Cria o vetor de tempo para valores entre 0 e N-1 (0 a 99)
9
10
11
12 %Calcular Variancia e Desvio Padrão do sinal%
13
14- Variancia = (sum((sem_ruido-mean(sem_ruido)).^2)/(N-1)); %Calcula a Variancia do sinal
15- disp(Variancia); %Apresenta o valor da Variancia na consola
16- Desvio_Padrao = sqrt(Variancia); %Calcula o Desvio Padrão atraves da raiz quadrada da variancia
17- disp(Desvio_Padrao); %Apresenta o valor do Desvio Padrão na consola

>> Exercicio1_TrabalhoPratico
0.1046
0.3234

```

Figura 3 – Variância e Desvio Padrão do Sinal “Sem_Ruido”;

b) Apresente graficamente o sinal `sem_ruido` em função do tempo, bem como as medidas calculadas na alínea anterior.

A fim de apresentar o sinal `sem_ruido` em função do tempo, bem como todas as medidas anteriores, após o cálculo das medidas, executamos o comando `plot([t(1) t(end)], "medida a desenhar", 'cor')` a fim de gerar graficamente a medida pretendida com a cor e formato pretendido. O script abaixo demonstra o plot feito à média, média mais desvio padrão, média menos desvio padrão, variância e valor eficaz. De seguida colocamos um título, labels, para o eixo do x e y, e uma legenda de cores para facilitar, deste modo, a identificação das medidas.

```
clear all;
close all;
load('biossinal.mat'); %Carrega o ficheiro biossinal.mat fornecido para a realização do trabalho

fs = 1/0.005; %Frequencia do sinal com intervalo de amostragem 5ms
N = 100; %length(sem_ruido) = 100; logo N = 100; Numero de amostras
t = (0:N-1)/fs; %Cria o vetor de tempo para valores entre 0 e N-1 (0 a 99)

%Apresentar Graficamente o sinal com as medidas calculadas%
figure(1); %Cria a figura para ser gerado o grafico
plot(t, sem_ruido, 'k'); %Gera o grafico do sinal "sem_ruido" em função do tempo com a cor preta
hold on;
Media = mean(sem_ruido); %Calculo da media do sinal
RMS = sqrt(mean(sem_ruido.^2)); %Calculo do valor eficaz do sinal
Variancia = (sum((sem_ruido-mean(sem_ruido)).^2)/(N-1)); %Calcula a Variancia do sinal
Desvio_Padiao = sqrt(Variancia); %Calcula o Desvio Padrão através da raiz quadrada da variancia

plot([t(1) t(end)],[Desvio_Padiao+Media Desvio_Padiao+Media],'-r'); %Gera no grafico Média+Desvio Padrão a Vermelho
plot([t(1) t(end)],[-Desvio_Padiao+Media -Desvio_Padiao+Media],'-g'); %Gera no gráfico Média-Desvio Padrão a Verde
plot([t(1) t(end)],[Variancia Variancia],'-c'); %Gera no gráfico Variancia a azul claro
plot([t(1) t(end)],[RMS RMS],'-b'); %Gera no gráfico o Valor Eficaz (RMS) a azul escuro
plot([t(1) t(end)],[Media Media],'-m'); %Gera no grafico Média a magenta
title('Biossinal (sem-ruido)'); %Insere no gráfico o título do mesmo
xlabel('Tempo t(s)'); %Insere no gráfico a legenda do eixo dos XX
ylabel('x(t)'); %Insere no gráfico a legenda do eixo dos YY
legend('Biossinal','Média+Desvio Padrão', 'Média-Desvio Padrão', 'Variancia', 'Valor Eficaz (RMS)', 'Média');
%Insere a legenda das linhas utilizadas
```

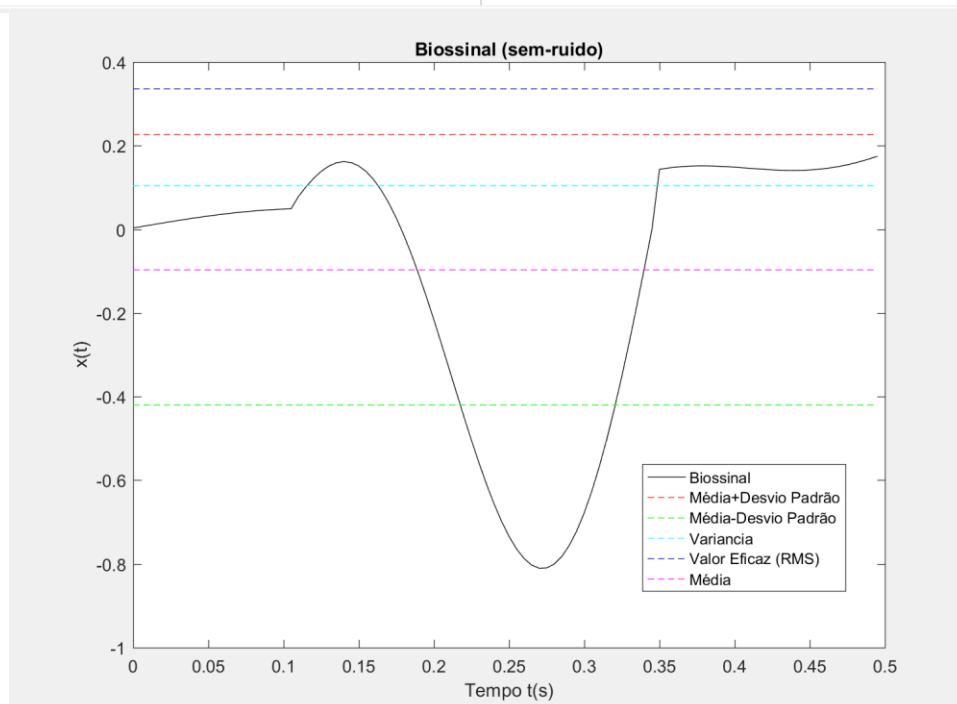


Figura 4 – Script de resolução da alínea b) e gráfico do Sinal “Sem_Ruido” com as medidas calculadas;

Exercicio 2

Apresente um script em MATLAB, devidamente comentado, que calcule e apresente graficamente o espectro do sinal `sem_ruido`. O espectro deverá ser apresentado em magnitude e em fase. O gráfico de fase deverá ser apresentado sem wrap-around.

A fim de calcular o espectro do sinal apresentado em magnitude, implementamos o comando `fft(sem_ruido)`, Fast Fourier Transform. Este permite calcular o espectro do sinal pela transformada de Fourier (Espectro). De seguida aplicamos o comando `abs` ao espectro anteriormente calculado obtendo-se, deste modo, o espectro do sinal apresentado em magnitude. Por último, a fim de obter o espectro do sinal apresentado em fase, utilizamos o comando `unwrap(angle(Espectro))`. O comando `angle` permite calcular o ângulo da fase do espectro. Este pode apresentar problemas pelo facto do seu resultado estar limitado entre $-\pi$ e $+\pi$ sendo que coloca os valores maiores dentro desse limite. Deste modo, um desvio de fase de $3\pi/2$ será igual a $\pi/2$, a $5\pi/2$ e por aí adiante. Este facto faz com que quando o espectro do sinal resulta de desvios grandes de fase, seja produzido saltos sobre a curva de fase (wrap-around). A fim de obter o espectro de fase correto (sem saltos maiores do que π), ou seja, sem 'wrap-around', utiliza-se o comando `unwrap` que permite a verificação da existência de saltos maiores do que π , subtraindo 2π sempre que a verificação seja positiva.

No final fizemos o plot do espectro apresentado em magnitude e em fase. As seguintes imagens mostram o script utilizado o seu resultado gráfico.

```
clear all;
close all;

load('biossinal.mat'); %Carrega o ficheiro biossinal.mat fornecido para a realização do trabalho

fs = 1/0.005; %Frequencia do sinal com intervalo de amostragem 5ms
N = 100; %length(sem_ruido) = 100; logo N = 100; Numero de amostras
t = (0:N-1)/fs; %Cria o vetor de tempo para valores entre 0 e N-1 (0 a 99)
f = (0:N-1)*fs/(N-1); %Cria o vetor de frequencia

%Calculo do Espectro, Magnitude e Fase do Sinal%
Espectro = fft(sem_ruido); %Calcula o espectro do sinal pela transformada de Fourier
Magnitude = abs(Espectro); %Calcula a magnitude do Espectro
Phase = unwrap(angle(Espectro)); %Calcula a Fase do Espectro
Phase = Phase * 360/(2*pi); %Unwrap da fase do Espectro

%Graficos do Sinal em Fase e Magnitude%
subplot(1,2,1); %Cria um subplot com uma linha e duas colunas na primeira posição
plot(f,Magnitude); %Cria o plot da Magnitude do Sinal "sem_ruido"
title('Magnitude do Sinal'); %Legenda da titulo do plot de magnitude
xlabel('Frequência (Hz)'); %Legenda do eixo do XX
ylabel('Magnitude'); %Legenda do eixo do YY
subplot(1,2,2); %Cria um subplot com uma linha e duas colunas na segunda posição
plot(f,Phase); %Cria o plot da Fase do Sinal "sem_ruido"
title('Fase do Sinal'); %Legenda da titulo do plot de fase
xlabel('Frequência (Hz)'); %Legenda do eixo do XX
ylabel('Fase'); %Legenda do eixo do YY
```

Figura 5 – Script de resolução do exercício 2 do espectro de sinal em fase e magnitude;

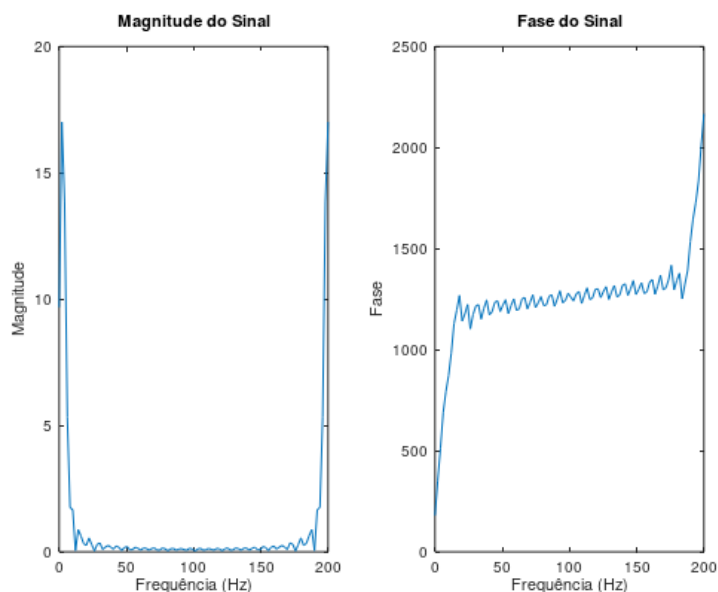


Figura 6 – Gráfico do Espectro do Sinal em magnitude e fase;

Exercicio 3

Apresente um script em MATLAB, devidamente comentado, que calcule a média das múltiplas respostas evocadas individuais. Divida o sinal (use variável `biossinal`) em segmentos de 100 pontos cada (intervalo de amostragem é 5 ms) e apresente os seguintes resultados:

a) Apresente graficamente medidas de “Ensemble Averaging” para 25, 100 e 1000 respostas e compare os 3 sinais de média com o sinal `sem_ruido`. Comente os resultados.

Nestes exercícios, para além dos comandos utilizados nas alíneas anteriores, utilizamos o `'vec2mat (biossinal, 100)'`. Este permite a conversão do vector `'biossinal'` para uma matriz com 100 colunas resultando uma matriz com 1000 linhas e 100 colunas, como demonstra o comando `'size(novo)'`. De seguida codificamos uma condição que nos retornava o vetor tempo correto conforme o numero de linhas e de colunas: se o numero de linhas fosse menor que o numero e colunas, faz-se a transposta de `'novo'` e o vetor tempo seria `'t=(0:NL-1)/fs'`, caso contrário o vetor de tempo seria `'t=(0:NC-1)/fs'`. De seguida representamos graficamente o sinal sem ruído, a média da matriz `'novo'` e o Ensemble averaging para 25, 100 e 1000 respostas.

Os Resultados permitem concluir que quantas mais respostas houver no Ensemble Averaging, mais preciso é o sinal, ou seja, com menos ruído este se torna. As representações gráficas obtidas mostram claramente que, comparando com o sinal sem ruído: o Ensemble Averaging para 25 contém bastante ruído, sendo, portanto, o sinal pouco claro; o Ensemble averaging para 100 já se encontra com menos ruído do que o anterior;

O Ensemble averaging para 1000 é um sinal bastante próximo do sinal da média da matriz e, consequentemente já é um sinal que se assemelha bastante ao sinal sem ruído, apesar de haver sempre ruído associado.

Neste Exercício conseguimos constatar que o Ensemble averaging é, de facto uma técnica de processamento de sinal bastante poderosa para reduzir o ruído aquando da possibilidade de obtenção de várias observações do mesmo sinal. Este consiste no cálculo da média de várias medições e, de seguida no cálculo da média das médias anteriormente calculadas. Assim é possível reduzir o ruído do sinal.

As seguintes imagens mostram o script e representações gráficas que resultaram deste exercício:

```
clear all;
close all;

load biossinal; % Carrega o ficheiro biossinal.mat fornecido para a realização do trabalho ;
fs = 1/.005; % Frequencia do sinal com intervalo de amostragem 5ms
N = length(sem_ruído); % Numero de amostras
t_sem_ruído = (0:N-1)/fs; % Cria o vetor de tempo
novo = vec2mat(biossinal,100); % converte o vector biossinal numa matriz com 100 colunas
[NL,NC] = size(novo); % retorna o tamanho da matriz novo

if NL < NC % numero de linhas < que o numero de colunas
novo = novo'; % faz se a transposta da matriz novo
t = (0:NL-1)/fs; % Cria o vetor de tempo
else
t = (0:NC-1)/fs; % Cria o vector de tempo
end

figure(1);
subplot(321);
plot(t_sem_ruído,sem_ruído); % representa graficamente o sinal sem ruído
title('Sem ruído');
subplot(322);
avg1 = mean(novo); % calculo da média da matriz novo
plot(t,avg1); % representa graficamente a média de 'novo'
title('Média');
subplot(323);
avg2 = mean(novo(1:25,:)); % calcula as medidas de 'Ensemble Averaging' para 25 respostas
plot(t,avg2);
title('Ensemble Averaging para 25 respostas');
subplot(324);
avg3 = mean(novo(1:100,:)); % calcula as medidas de 'Ensemble Averaging' para 100 respostas
plot(t,avg3);
title('Ensemble Averaging para 100 respostas');
subplot(325);
avg4 = mean(novo(1:1000,:)); % calcula as medidas de 'Ensemble Averaging' para 1000 respostas
plot(t,avg4);
title('Ensemble Averaging para 1000 respostas');
```

Figura 7 – Script da resolução de “Ensemble Averaging” para 25, 100 e 1000 respostas;

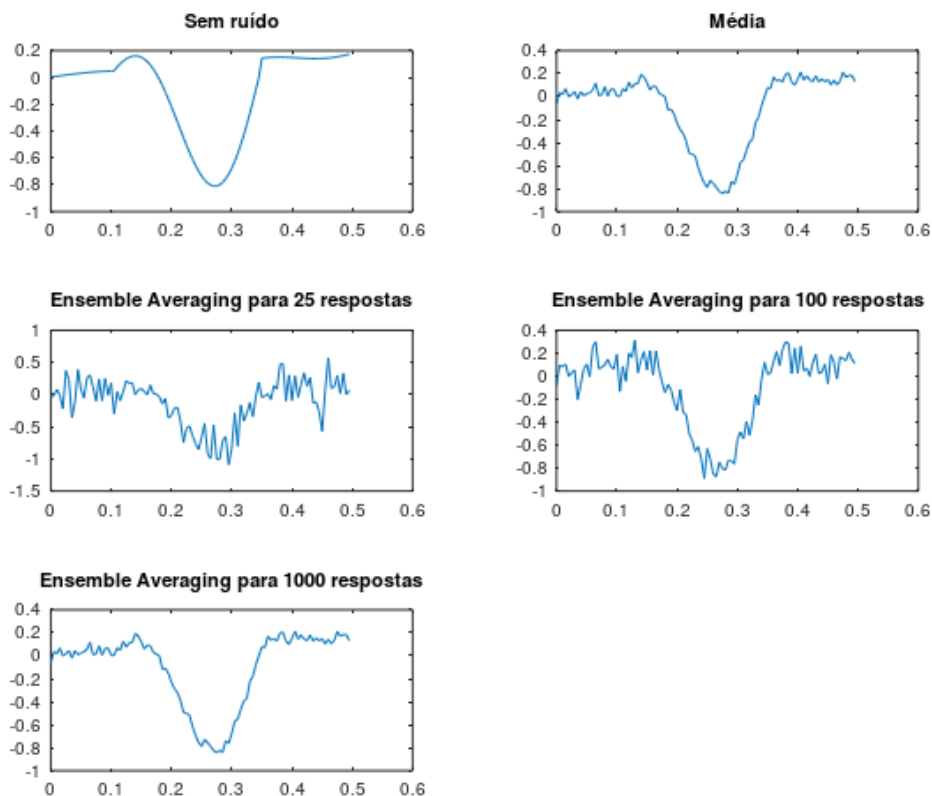


Figura 8 – Gráfico das medidas de “Ensemble Averaging” para 25, 100 e 1000 respostas;

b) Calcule o espectro de potências dos 3 sinais de média e de uma resposta individual, comparando-os graficamente. O que conclui quanto à utilidade da medida “Ensemble Averaging” para a avaliação da relação sinal-ruído?

Neste exercício, executamos o cálculo do espectro de potência que consiste no quadrado da magnitude da transformada de Fourier. Colocando os sinais calculados na alínea a) ao quadrado e retirando o valor absoluto desses, obtemos o espectro de potência dos sinais. Para obter o espectro de potência duma resposta individual, escolhemos a última linha do sinal, ou seja, ‘(end-1):end’), e executamos o mesmo processo anterior.

Podemos concluir que o “Ensemble Averaging” é uma técnica de redução de ruído eficaz, como explicado anteriormente, e que é compatível no cálculo do espectro de potência. Graficamente, conseguimos constatar que, no espectro de potência, o sinal fica com menos ruído à medida que se incrementa várias medições da mesma fonte, ou seja, o gráfico da resposta individual é bastante afetado com ruído, o que não acontece no gráfico que contém 1000 Respostas.

```

41 %Alinea B%
42
43 f=(0:N-1)*(fs/N); % alcance da frequencia
44 YI1 = abs((fft(mean(novo(1:25,:))))).^2); % calculo do espectro de potencia - 25 respostas
45 YI2 = abs((fft(mean(novo(1:100,:))))).^2); % calculo do espectro de potencia - 100 respostas
46 YI3 = abs((fft(mean(novo(1:1000,:))))).^2); % calculo do espectro de potencia - 1000 respostas
47 YI4 = abs((fft(mean(novo(end-1:end,:))))).^2); %Calculo do espectro de potencia de resposta individual
48
49 figure(2);
50 subplot(221);
51 plot(f(1:N),YI1(1:end),'k'); % representação grafica do espectro de potencia YI1 a preto
52 xlabel('Frequency (hz)');
53 ylabel('Power Spectrum');
54 title('25 Respostas');
55 subplot(222);
56 plot(f(1:N),YI2(1:end),'r'); % representação grafica do espectro de potencia YI2 a vermelho
57 xlabel('Frequency (hz)');
58 ylabel('Power Spectrum');
59 title('100 Respostas');
60 subplot(223);
61 plot(f(1:N),YI3(1:end),'b'); % representação grafica do espectro de potencia YI3 a azul
62 xlabel('Frequency (hz)');
63 ylabel('Power Spectrum');
64 title('1000 Respostas');
65 subplot(224);
66 plot(f(1:N),YI4(1:end),'c'); % representação grafica do espectro de potencia YI4 a azul
67 xlabel('Frequency (hz)');
68 ylabel('Power Spectrum');
69 title('Resposta Individual');

```

Figura 9 – Script da resolução do escetro de potencia para as varias respostas;

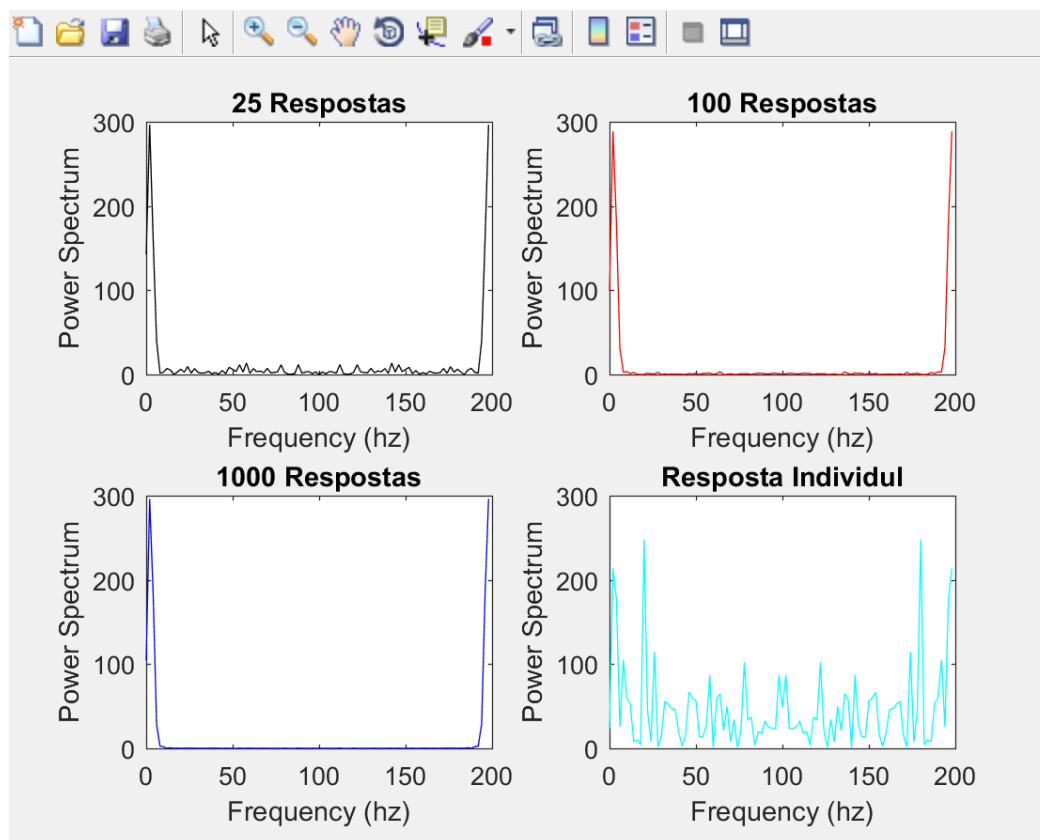


Figura 10 – Gráfico do espectro de potencia para as varias respostas;

Conclusão

No decorrer do trabalho, foi colocado em prática todos os conhecimentos que foram lecionados e aprendidos ao longo das aulas desta unidade curricular, deste modo sentimos que de um modo geral estamos satisfeitos com os resultados obtidos, pensamos também que cumprimos com os objetivos propostos para este trabalho. Este projeto que incluiu a realização de processamento de sinal utilizando as ferramentas do Matlab e Octave, fez-nos colocar em prática não só os conhecimentos apreendidos nas aulas práticas, mas também nas aulas onde a matéria nos era exposta.

Depois deste trabalho, saímos com o sentimento de uma maior sabedoria, uma vez que o trabalho não se focava apenas em colocar em prática os conhecimentos obtidos, mas sim, ir em busca de mais conhecimentos. Julgamos que este trabalho contribuiu para um melhor espírito crítico sobre a matéria, o que é fundamental para sucesso futuro.

Ao longo do trabalho foram surgindo algumas dificuldades, o que nos levou em busca de soluções, talvez nem sempre a melhor, mas foi um desafio conseguir concluir este trabalho.

Apesar das dificuldades encontradas, de um modo geral, estamos satisfeitos com o resultado obtido, esperamos, pois, que seja uma mais-valia para um futuro que nos espera.