

UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA ELETRÓNICA INDUSTRIAL
E COMPUTADORES

PROCESSAMENTO DIGITAL DE SINAL

Filtro Butterworth

Alunos

RAFAEL MELEIRO MARQUES
A74420

Professor

CARLOS MANUEL GREGÓRIO
SANTOS LIMA

19 Maio 2019

Conteúdo

Lista de Figuras	ii
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos	1
1.3 Estrutura do Relatório	1
2 Conceitos Teóricos	2
2.1 Decimação	2
3 Problema	4
4 Desenvolvimento em Matlab	5
4.1 Gravação do Áudio	5
4.2 Obtenção valores fundamentais da projeção do filtro	5
4.3 Criação do filtro apartir dos parâmetros	6
4.4 Função Bilinear e Filter	6
4.5 Downsample	6
4.6 Criação do objeto final	7
5 Testes e análise dos resultados	8
5.1 1º Teste	9
5.2 2º Teste	10
5.3 3º Teste	11
5.4 4º Teste	12
5.5 5º Teste	13
6 Conclusão	14

Lista de Figuras

1	Decimador	2
2	Processo de Decimação	3
3	Excerto de código para a criação do áudio amostrado	5
4	Excerto de código para projeção do filtro	5
5	Excerto de código para criação do filtro	6
6	Utilização do filtro	6
7	Downsample	6
8	Sinal filtrado	7
9	Sinal Original	8

1 Introdução

1.1 Motivação

Com a realização deste trabalho pretende-se desenvolver competências no software Matlab, bem como consolidação de conhecimentos relativamente a esta matéria.

Através da prática com a ferramenta Matlab será possível visualizar alguns dos conceitos aprendidos na Unidade Curricular.

1.2 Objetivos

Pretende-se implementar em Matlab um módulo que permita diminuir a frequência de amostragem por meios digitais evitando a ocorrência de *aliasing*, processo normalmente designado por *Downsampling*.

1.3 Estrutura do Relatório

Para cumprir os objetivos no relatório será dada uma breve explicação relativamente à decimação. Posteriormente será apresentado o problema proposto pelo docente e em seguida como será feita a implementação na ferramenta Matlab.

Para se conseguir verificar o correto funcionamento do filtro serão realizados testes para visualmente se verificar.

2 Conceitos Teóricos

2.1 Decimação

A decimação é a operação correspondente à redução da frequência de amostragem de um sinal digital, sendo este um sinal amostrado a partir de um sinal contínuo, a uma determinada frequência de amostragem. Considerando que muitas vezes o sinal original é amostrado a uma frequência de amostragem alta, através do processo de decimação somos capazes de reduzir as amostras, sendo o sinal original recuperável.

Este processo é vantajoso para compactar os dados, de modo a ser ocupada menos memória para armazenamento do sinal, sendo que o processo de decimação tem que cumprir determinados requisitos. Na realização desta operação, a única dificuldade é a possível ocorrência de aliasing, daí a utilização de um filtro “anti-aliasing” antes de ser efetuado o “downsample”.

A decimação é um processo reversível, isto é, realizada sem perda de informação, se $X(e^{j\omega})$ for zero fora do intervalo $[-\pi/M, \pi/M]$, ou seja, se for de banda limitada.

O processo de decimação é realizado pelo dispositivo decimador ilustrado na Figura 1. Este dispositivo tem como sinal de entrada o sinal $x[n]$ (sinal amostrado do sinal contínuo), e um fator M , que é responsável por gerar um sinal de saída $y[n]$, com uma taxa de amostragem M vezes menor que a taxa de amostragem do sinal amostrado na entrada, sendo a fórmula que descreve a seguinte:

$$X_d(k) = X(Mk)$$

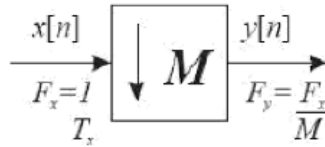


Figura 1: Decimador

Sabemos que reduzindo a frequência de amostragem através da seleção de valores $x[n]$ para múltiplos de M , o sinal resultante conterá “aliasing”, com duplicação em $F_x/2M$. Desta forma, para evitar o aliasing, é necessário reduzir a largura de banda de $x[n]$ para:

$$\Omega_{\max} = \pi/M.$$

Sendo assim, utiliza-se um filtro passa baixas (anti-aliasing) para eliminar o espectro $X(\Omega)$ no intervalo $\pi/M \leq \Omega \leq \pi$. Este filtro é caracterizado pela resposta ao impulso $h[n]$ e uma resposta de frequência $H_d(\Omega)$ que satisfaz a seguinte condição:

$$H_D(\Omega) = \begin{cases} 1, & |\Omega| \leq \pi D \\ 0 & \text{demais casos} \end{cases}$$

O processo de decimação está dividido em duas etapas como ilustrado na Figura 2. A primeira etapa é a redução das componentes do sinal com alta frequência com um filtro passa baixo; a segunda etapa realiza-se o “downsample” no sinal filtrado.

O processo de downsample sem a utilização prévia do filtro passa baixo faz com que as componentes do sinal de alta frequência sejam sobrepostos por outros dados, tornando-se mais propício que ocorra o aliasing, daí ser utilizado previamente um filtro passa baixo.

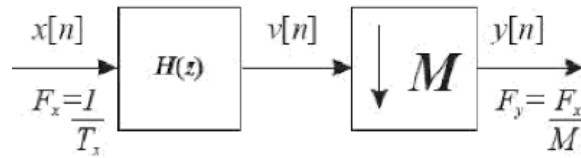


Figura 2: Processo de Decimação

3 Problema

Foi proposto a implementação de uma função cujos parâmetros de entrada é um segmento de áudio amostrado a 8 KHz e um factor de sub-amostragem do amostrador discreto. Esta função deve devolver um áudio amostrado a $\frac{Fs}{N}$.

Como filtro passa baixo será usado um filtro Butterworth com as seguintes características:

1. Ripple na banda passante de 40 dB
2. Ripple na banda de rejeição de 60 dB
3. Largura de banda de transição de 20 % da banda passante.

Este problema será resolvido com auxílio do programa Matlab, fazendo chamada recursiva da função de modo a gerar os diferentes valores do coeficiente de sub-amostragem, verificando sempre o áudio ao fim, para se poder identificar a perda sucessiva do conteúdo espectral.

4 Desenvolvimento em Matlab

De forma a cumprir o especificado pelo problema realizaram-se os seguintes passos:

4.1 Gravação do Áudio

Inicialmente foram executados os seguintes comandos de forma a gravar o áudio como se pode verificar na figura 3. Como resultado foi criado um ficheiro do tipo *.wav* que será usado posteriormente para a realização do estudo do sinal.

```
gravacao= audiorecorder;           %criação do objeto
disp('Start speaking.')
recordblocking(gravacao,4);        %gravação do sinal
disp('End of Recording.');
```



```
play(gravacao);                   %ouvir se a gravação ficou bem gravada
y = getaudiodata(gravacao);        %armazenar dados num array
audiowrite('som.wav',y,8000);      %escrever um ficheiro .wav com o array
[y,fs]=audioread('som.wav');       %leitura do ficheiro .wav para array
```

Figura 3: Excerto de código para a criação do áudio amostrado

4.2 Obtenção valores fundamentais da projeção do filtro

Com os dados fornecidos do ripple na banda passante, ripple na banda de rejeição e tendo a largura da banda de transição sabendo a banda passante obtem-se os valores necessário para fazer a projeção do filtro Butterworth. Pode-se observar a declaração destes valores na figura 4

```
rp=40;                             %ripple na banda passante
rs=60;                             %ripple na banda de rejeição
N=3;
wp=pi/N;                           %frequencia de corte
ws=wp+(0.2*wp);                    %frequencia de banda
Nd=2;                              %frequencia de sub-amostragem

[n,Wn] = buttord((2*fs*tan(wp/2))/Nd,(2*fs*tan(ws/2))/Nd,rp,rs,'s');
```

Figura 4: Excerto de código para projeção do filtro

4.3 Criação do filtro apartir dos parâmetros

Obtendo-se os parâmetros do filtro desejado presseguiu-se à sua criação. Com o comando butter (como verificamos na figura 5) obtemos os valores do numerador e denominador da função transferência do filtro analógico.

```
[b,a]=butter(n,Wn,'s');
```

Figura 5: Excerto de código para criação do filtro

4.4 Função Bilinear e Filter

Com os valores do numerador e do denominador, da função transferência do filtro analógico com auxílio à transformação bilinear, passamos diretamente da função de transferência do filtro analógico para a função de transferência do filtro digital para de seguida ser possível utilizar a função filter, função essa que permite aplicar o filtro ao sinal inicial. Ambas as implementações podem-se verificar na figura 6.

```
[bd,ad]=bilinear(b,a,fs);      %transformação bilinear  
  
sinalFiltrado=filter(bd,ad,y); %filtro
```

Figura 6: Utilização do filtro

4.5 Downsample

É efetuado o downsample do sinal filtrado pelo fator de subamostragem. Para tal, é chamada a função downsample (figura 7), e esta função recebe como parâmetros de entrada o sinal filtrado e o fator de subamostragem. Após isto mostramos o gráfico do sinal com o número de amostras já reduzido.

```
sinal_downsample=downsample(sinalFiltrado,Nd);  
figure  
plot(sinal_downsample)
```

Figura 7: Downsample

4.6 Criação do objeto final

Cria-se um objeto com o conteúdo do sinal após a decimação do mesmo, amostrado pela frequência de amostragem a dividir pelo fator de sub-amostragem, para que seja reproduzido no mesmo intervalo de tempo do sinal original.

```
player = audioplayer(sinal_downsample, fs);  
disp('Start playing');  
play(player);  
disp('End of playing');  
figure(4)  
subplot(2,1,1)
```

Figura 8: Sinal filtrado

5 Testes e análise dos resultados

Com o propósito de testar o trabalho até agora efetuado foram efetuados 5 testes diferentes. Em todos os testes a entrada é o mesmo audio mas com diferentes fatores de subamostragem, que alteram a compactação do sinal e também alterará a banda passante, para se poder verificar as mudanças que ocorrem ao áudio.

O áudio gravado foi retirado de uma música que continha um som de alta frequência a correr como ruído.

Para cada um dos testes realizados foram ilustrados os gráficos do sinal original, do sinal após filtragem e do sinal filtrado com aplicação do downsample.

A banda passante em cada teste foi considerada inicialmente $\pi/3$ e a banda de rejeição inicial foi a banda passante*1.2. A cada teste a banda passante e de rejeição variam por um factor de $1/N_d$. Inicialmente N_d será 1 e recursivamente será variado o valor de N_d .

O sinal original está representado na seguinte figura seguinte.

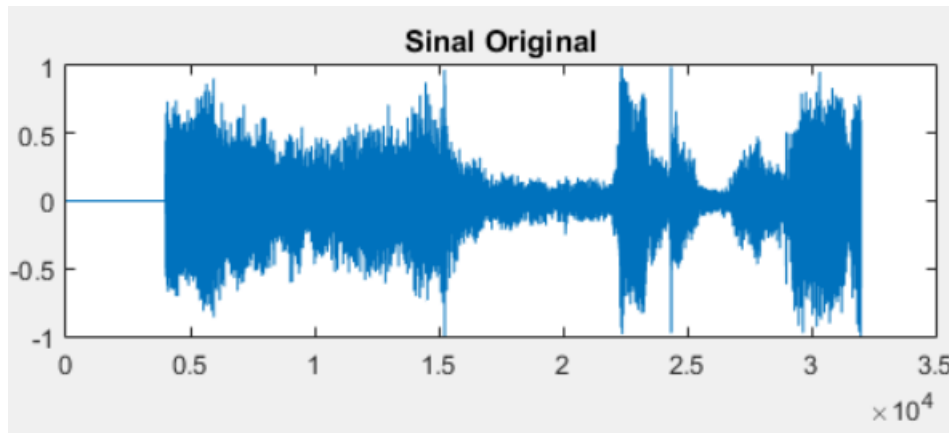
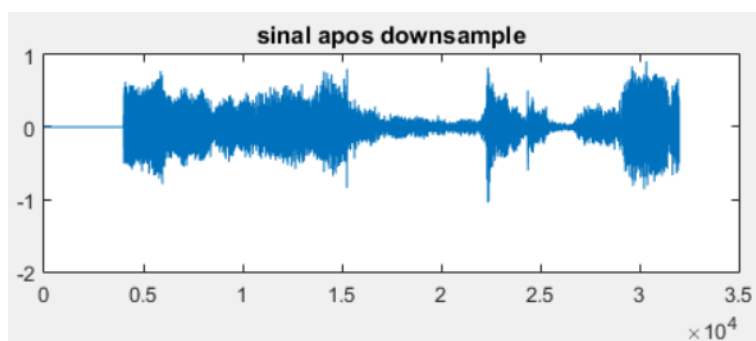
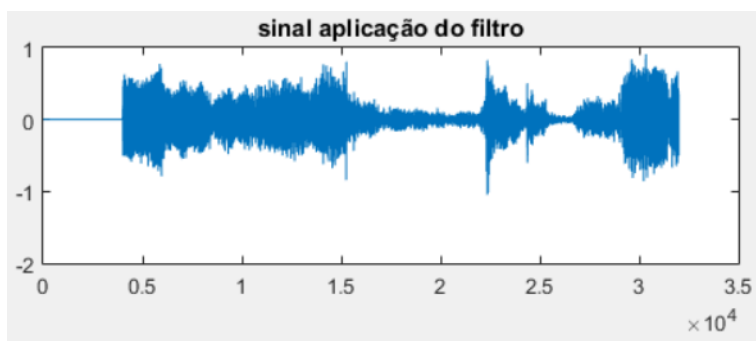
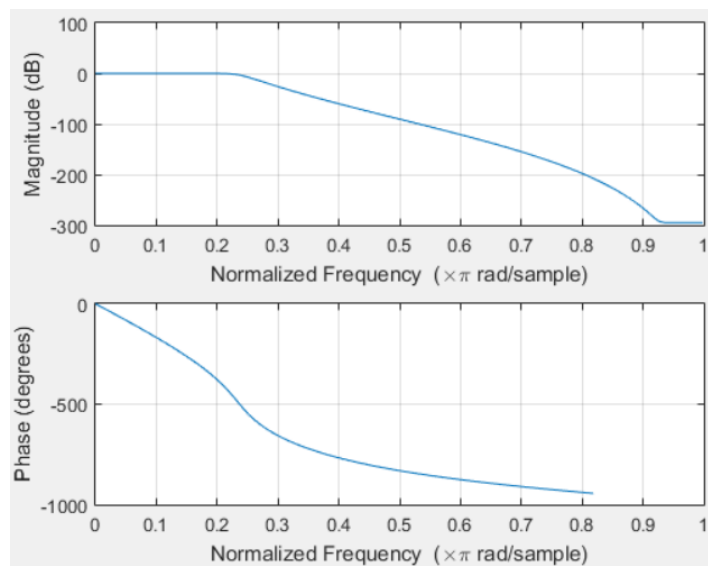


Figura 9: Sinal Original

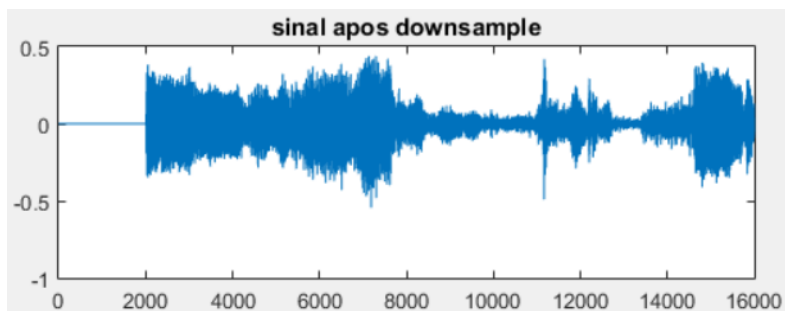
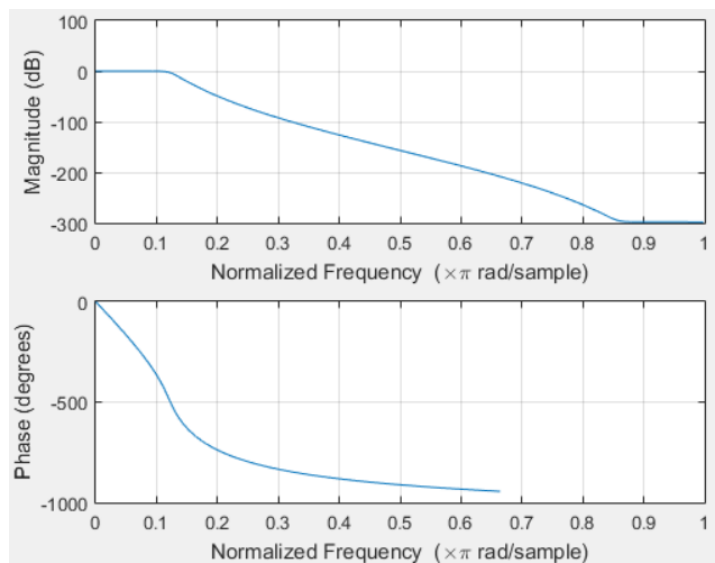
5.1 1º Teste

Para este foi atribuído à variável Nd o valor 1.



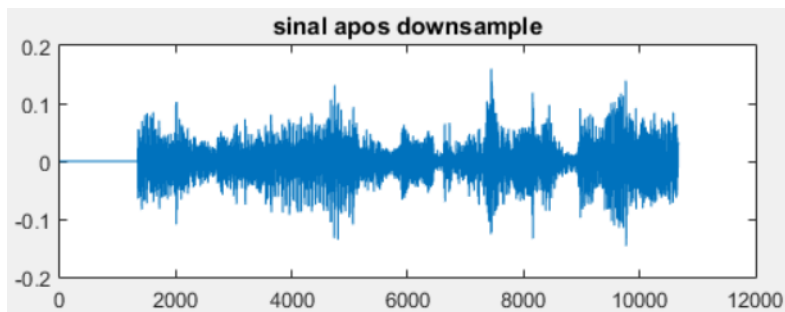
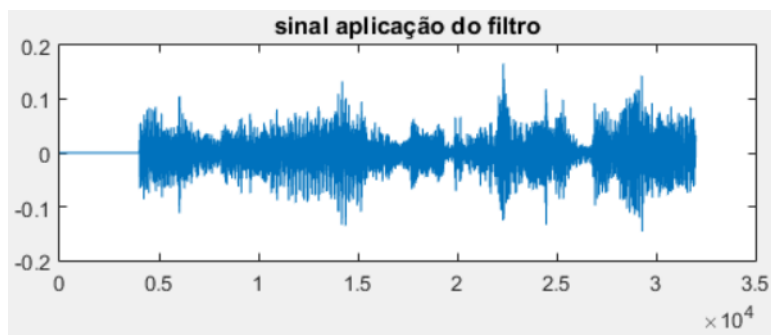
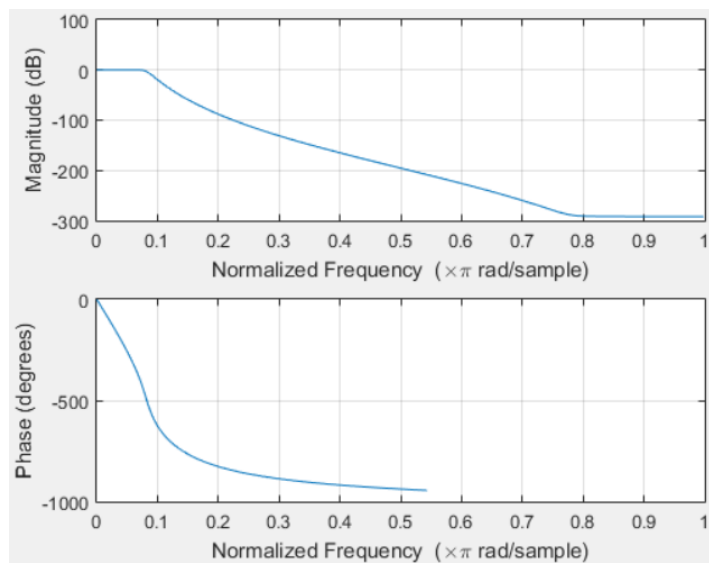
5.2 2º Teste

Para este foi atribuído à variável Nd o valor 2.



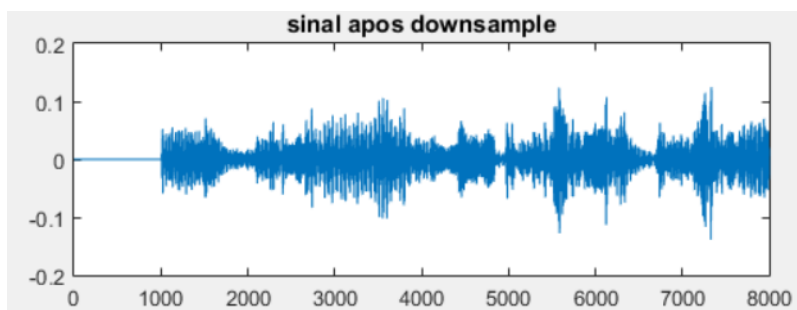
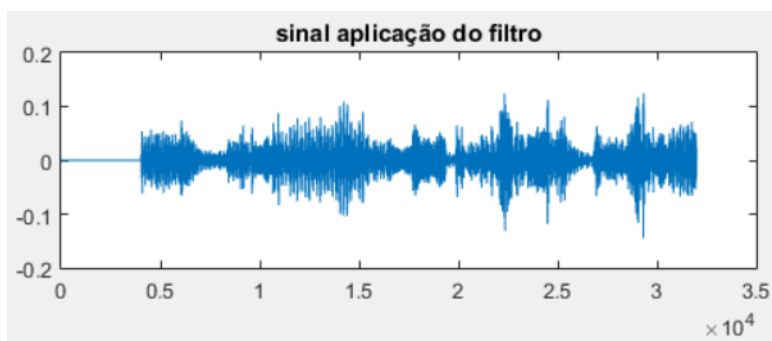
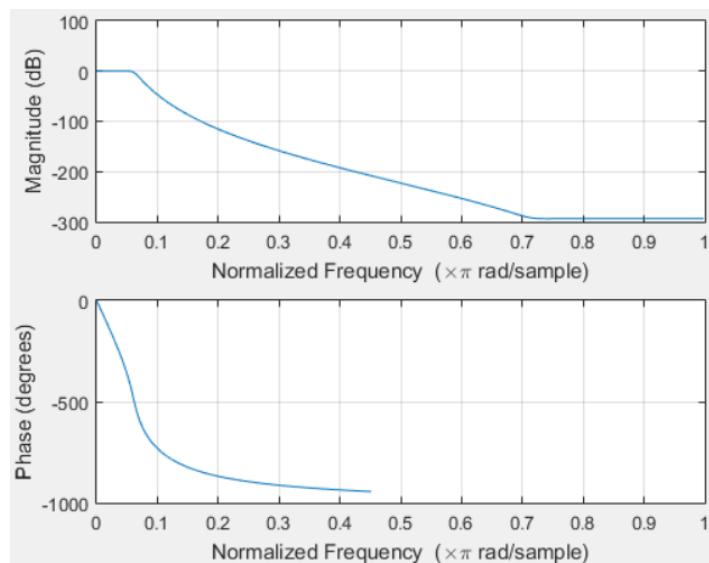
5.3 3º Teste

Para este foi atribuído à variável Nd o valor 3.



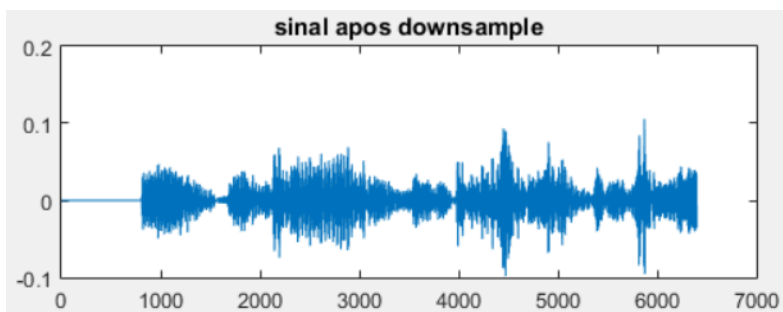
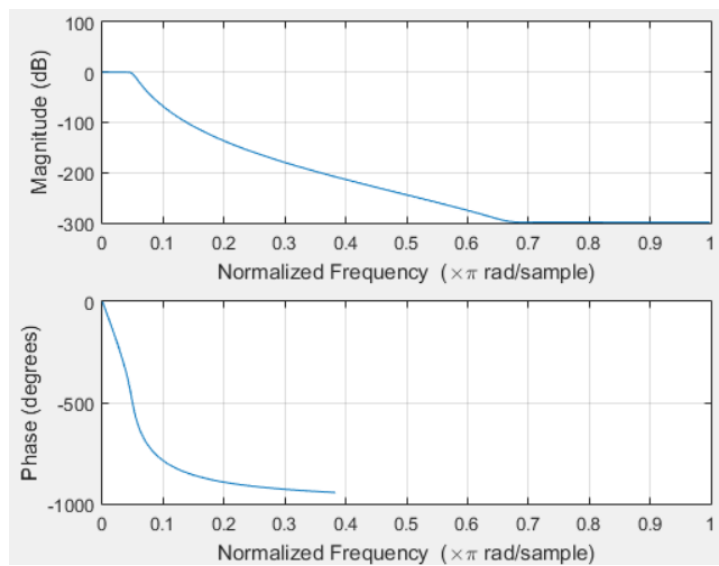
5.4 4º Teste

Para este foi atribuído à variável Nd o valor 4.



5.5 5º Teste

Para este foi atribuído à variável Nd o valor 5.



6 Conclusão

Com a realização deste trabalho foi possível desenvolver competências no software Matlab, bem como consolidação de conhecimentos relativamente a esta matéria.

Após os testes, conseguimos observar as diferenças do sinal inicial e do sinal após a decimação.

Com a variação da frequência de sub-amostragem e da banda passante observamos as perdas das frequências que esperaríamos perder. Além disto, a utilização do comando “downsample”, permitiu observar a compactação do sinal, o que facilita os cálculos e a menor utilização de memória para guardar os dados de um sinal.

Em suma, foi possível verificar com a realização deste trabalho todos os conceitos aprendidos em contexto de sala de aula, sendo que, com a utilização do software Matlab, os conteúdos esclarecidos.