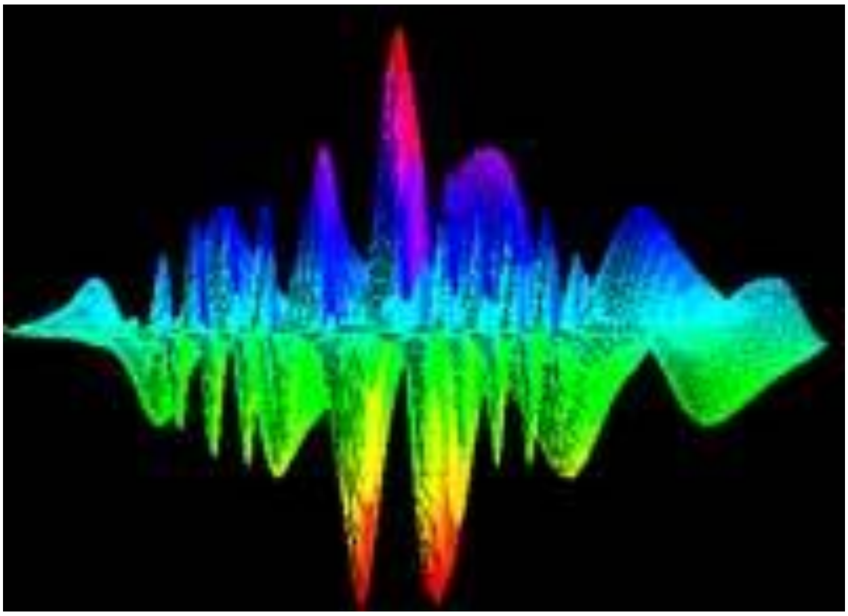
# Análise e Transformação de Dados

## Trabalho Prático 3

### Relatório



**Realizado Por**
João Miguel Rodrigues Jesus nº2008111667
Rosa Manuela Rodrigues de Faria nº 2005128014

## Exercício 1

**1.1.** Sendo este o grupo nº 23, teremos como expressão do sinal

$$x(t) = -1 + 3\sin(30\pi t) + 4\sin\left(12\pi t - \frac{\pi}{4}\right)\cos(21\pi t)$$
$$= -1 + 3\sin(30\pi t) - 2\sin\left(\frac{\pi}{4} - 33\pi t\right) - 2\sin\left(\frac{\pi}{4} + 9\pi t\right)$$
$$= -1 + 3\cos\left(30\pi t - \frac{\pi}{2}\right) + 2\cos\left(33\pi t + \frac{\pi}{4}\right) - 2\cos\left(9\pi t + \frac{\pi}{4}\right)$$

Assim, teremos $\omega \in \{0, 9\pi, 30\pi, 33\pi\} rad/s$
e $\omega_0 = mdc(0, 9\pi, 30\pi, 33\pi) = 3\pi\ rad/s$, com $f_0 = \frac{\omega_0}{2\pi} = \frac{3}{2} Hz$

A condição para escolher uma frequência de amostragem, segundo o Teorema de Nyquist, é dada por $\omega_s > 2\omega_{max} = 66\pi \Longrightarrow f_s > 33\ Hz$

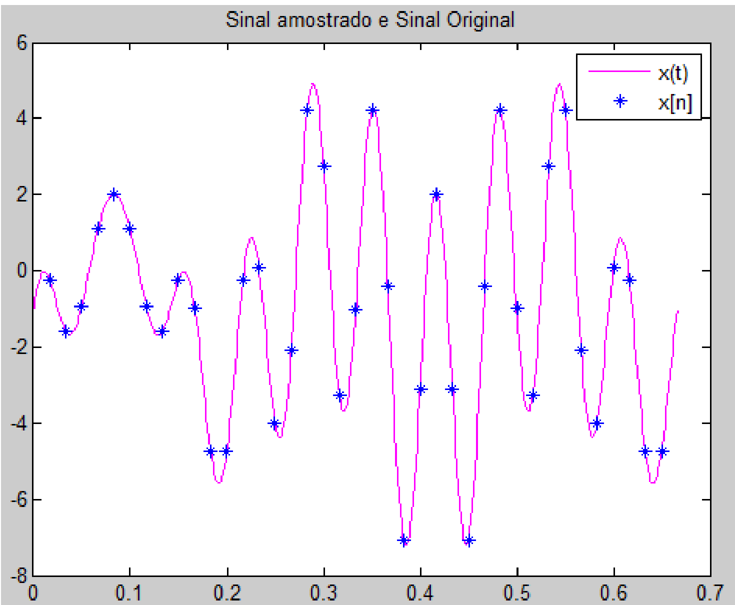
Assim, podemos escolher $f_s = 60\ Hz \Leftrightarrow T_s = {}^{1}/_{60}$, já que este valor cumpre o Teorema de Nyquist e é múltiplo de $f_0$.

Ficamos então com

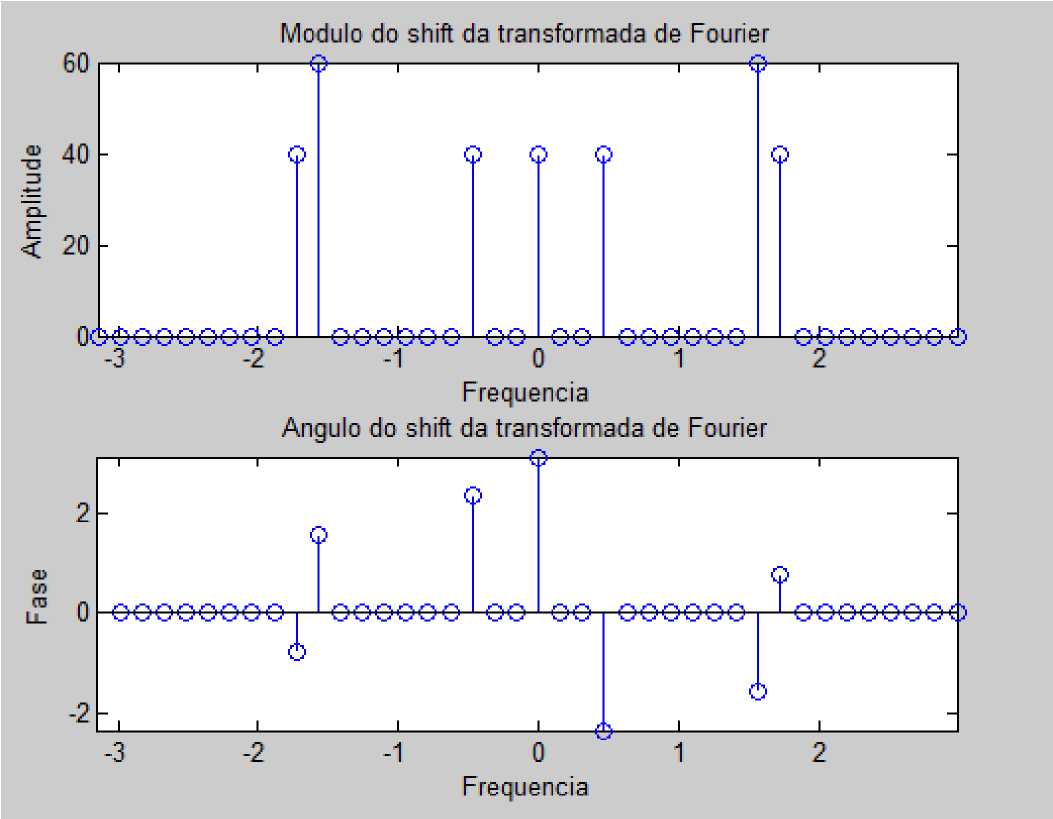
$$x[n] = -1 + 3\cos\left(\frac{1}{2}\pi n - \frac{\pi}{2}\right) + 2\cos\left(\frac{11}{20}\pi n + \frac{\pi}{4}\right) - 2\cos\left(\frac{3}{20}\pi n + \frac{\pi}{4}\right)$$

**1.2.** Como mencionado anteriormente, as frequências angulares do sinal contínuo $x(t)$ são $\omega \in \{0, 9\pi, 30\pi, 33\pi\} rad/s$ e a sua frequência e período fundamental são, respectivamente, $\omega_0 = 3\pi\ rad/s$ e $T_0 = \frac{2\pi}{\omega_0} = \frac{2}{3}\ s$.
Para o sinal de tempo discreto teremos $\Omega = \frac{\omega}{f_s} \in \left\{0, \frac{3\pi}{20}, \frac{\pi}{2}, \frac{11\pi}{20}\right\} rad/s$, $\Omega_0 = \frac{\pi}{20}\ rad/s$ e $N = \frac{2\pi}{\Omega_0} = 40$

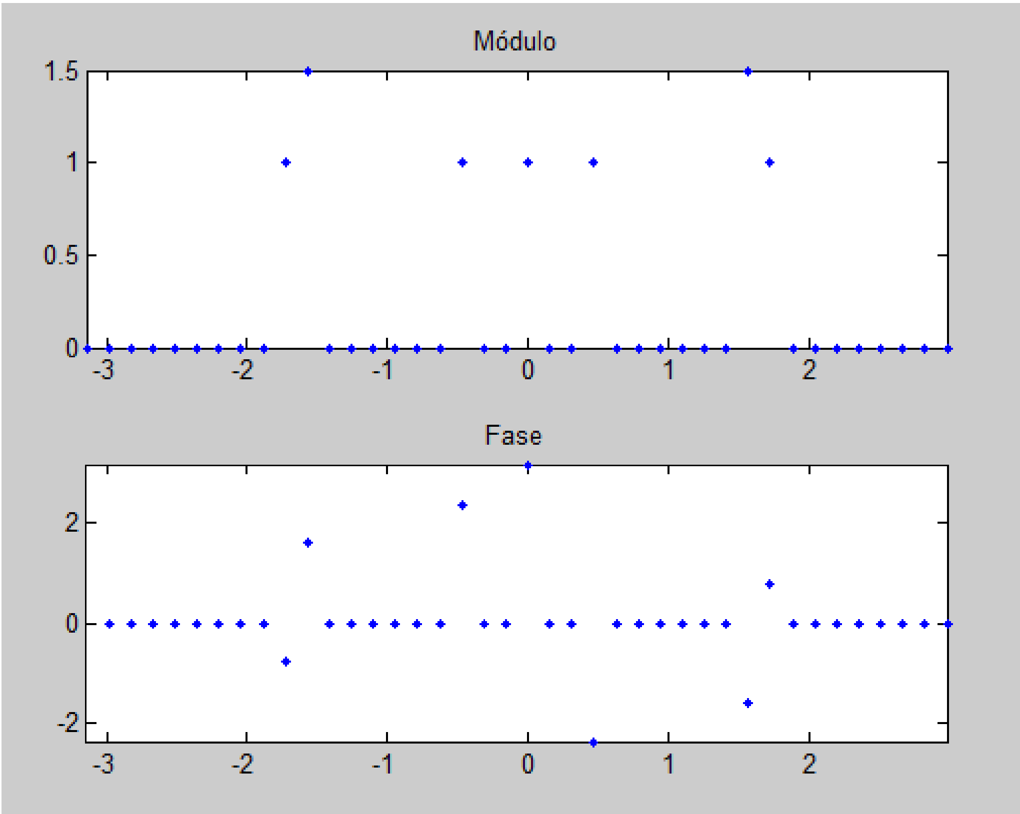
**1.3.**



### 1.4.



Modulo do shift da transformada de Fourier
Amplitude
Frequencia
Angulo do shift da transformada de Fourier
Fase
Frequencia

Módulo
Fase

**1.5.** Para determinar os coeficientes $c_m$ foi utilizada a fórmula

$$c_m = \frac{X}{N}$$

Obtendo os coeficientes não negativos de $c_m$:

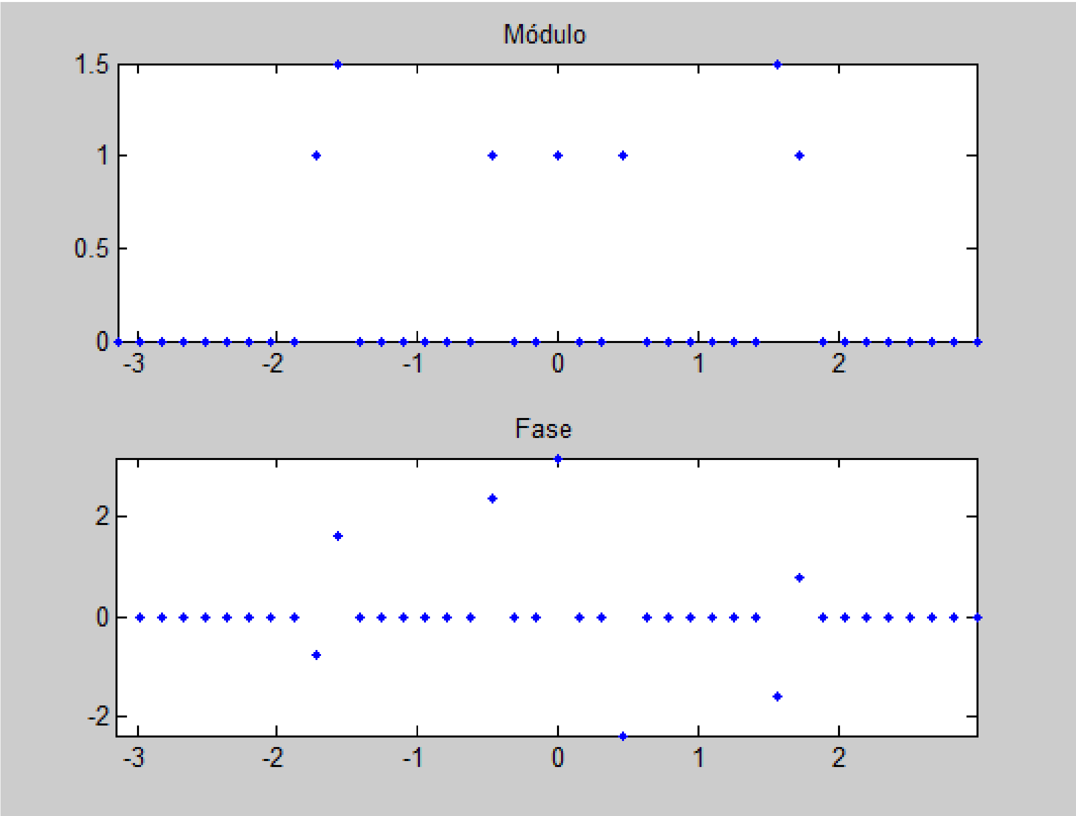
$$c_0 = -1$$

$$c_3 = -0.0711 + 0.0711i \qquad c_{-3} = -0.0711 - 0.0711i$$

$$c_{10} = -7.7272 \times 10^{-15} + 1.5i \qquad c_{-10} = -7.7272 \times 10^{-15} - 1.5i$$

$$c_{11} = 0.0711 + 0.0711i \qquad c_{-11} = 0.0711 + 0.0711i$$

A representação gráfica destes coeficientes resultou nos seguintes gráficos:




**1.6.** Para determinar os coeficientes da Série de Fourier a partir dos coeficientes da Série Complexa temos as seguintes expressões:

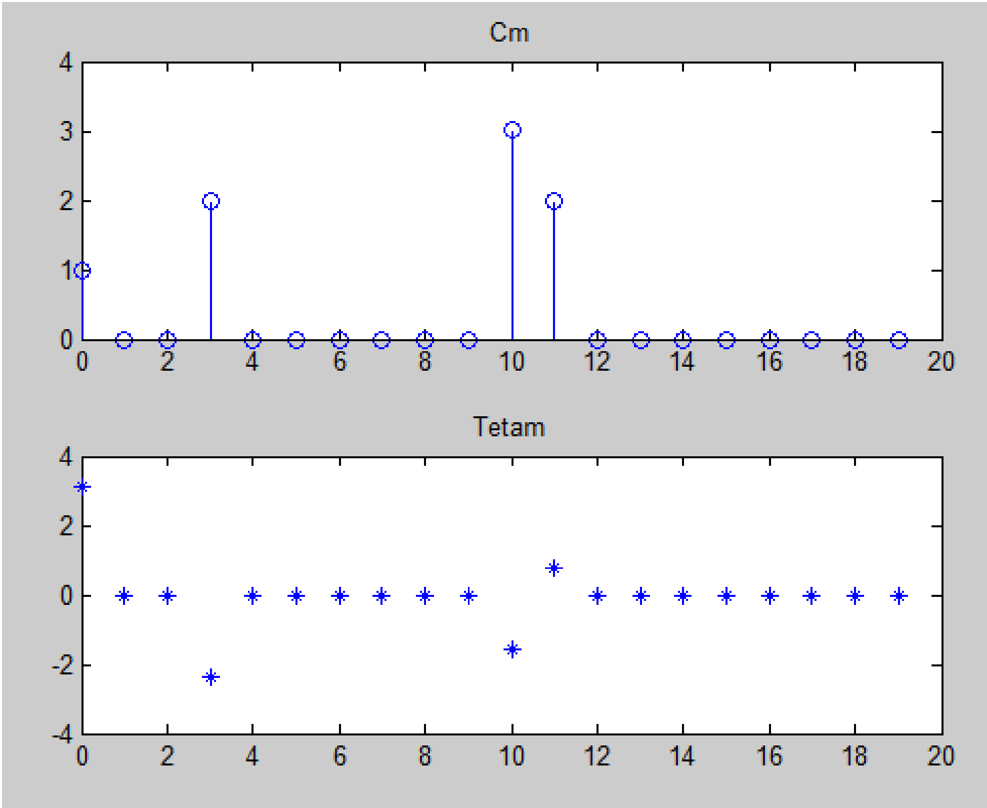
$$C_m = 2|c_m|, m > 0 \qquad C_0 = |c_0| \qquad \theta_m = \theta_{cm}\ m \geq 0$$

Aplicando-as obteremos

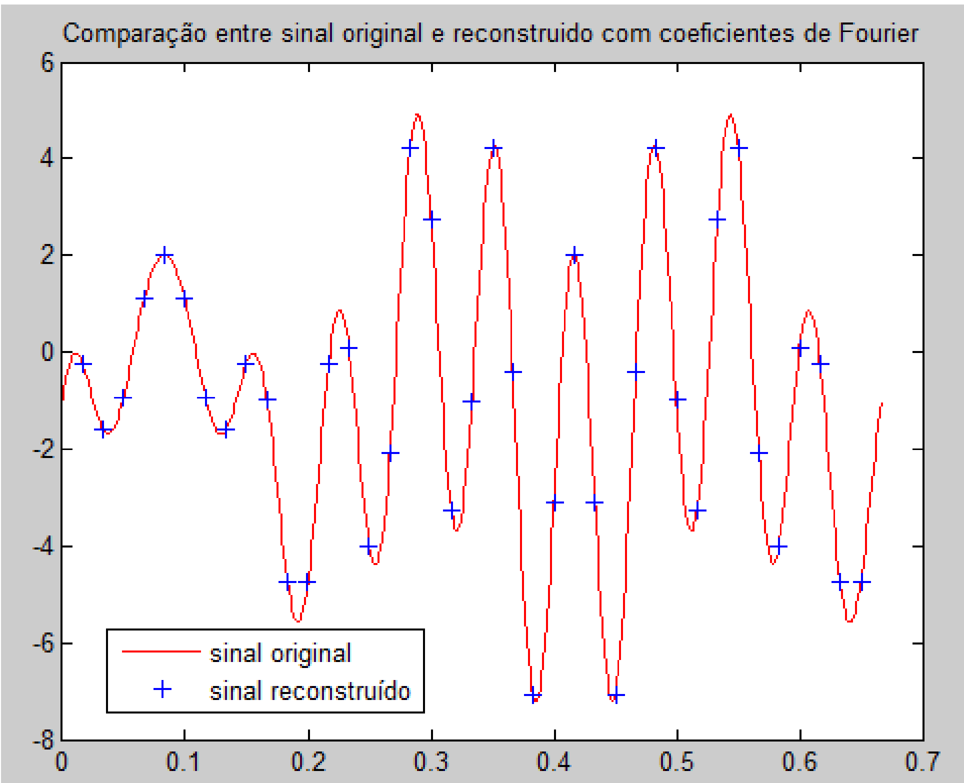
$$\theta_m = \left[\pi, 0,0, -\frac{3\pi}{4}, 0,0,0,0,0,0, -\frac{\pi}{2}, \frac{\pi}{4}\right]$$

$$C_m = [1,0,0,2,0,0,0,0,0,0,3,2]$$

Segue-se a representação gráfica destes.



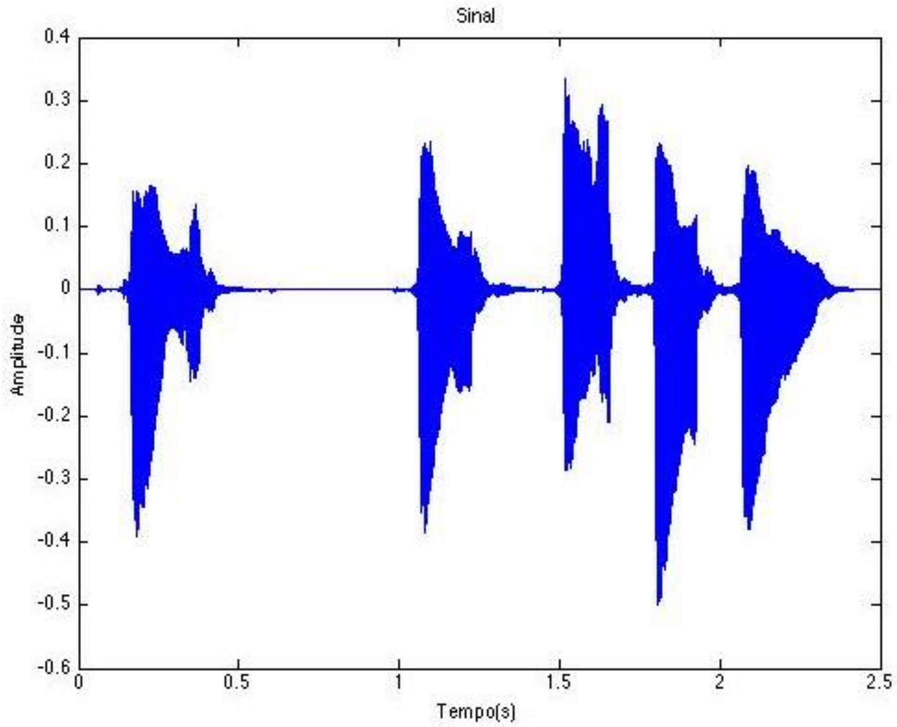

**1.7.** Ao reconstruirmos o sinal podemos ver que os sinais são perfeitamente coincidentes, como previsto teoricamente.



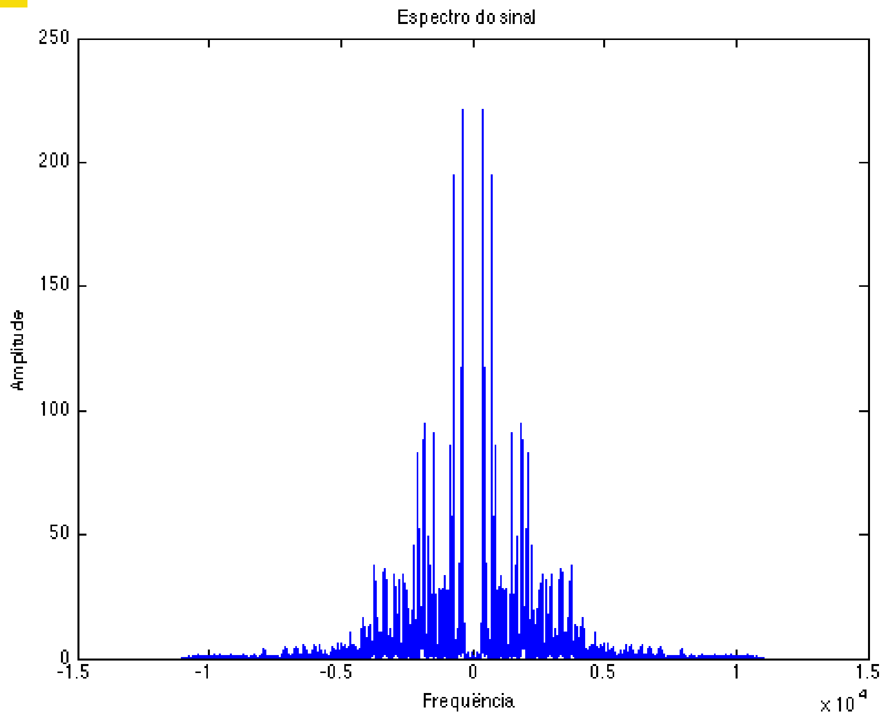
## Exercício 2

### 2.1.

Ao lermos o ficheiro "saxriff.wav" através da função *wavread* do MATLAB, iremos obter o sinal *x*, a frequência $f_S$ e o *bitrate nbits* do som lido.




Depois para representarmos o sinal graficamente, utilizamos a função *wavplot* criada após para resolver os problemas 4 e 5 (já que todos estes necessitam dos mesmos gráficos), onde calculámos a Transformada discreta de Fourier com os valores obtidos anteriormente, sendo o espectro do sinal representado em função da frequência.



### 2.2.

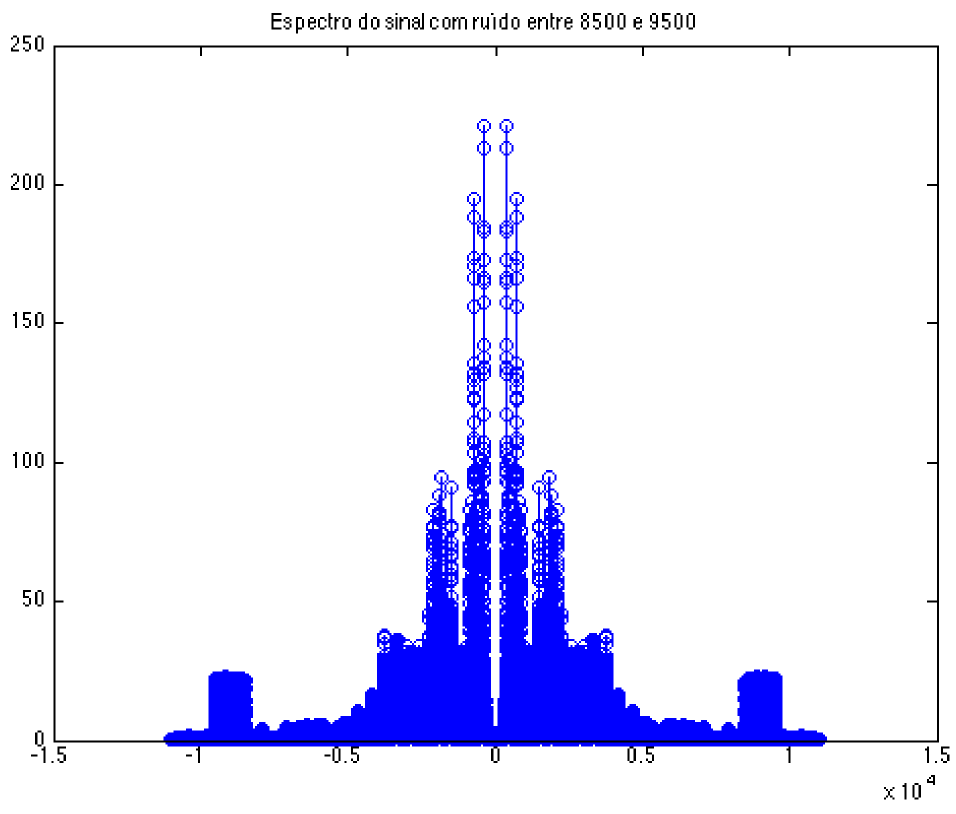
Para calcularmos a amplitude máxima da DFT e a sua frequência máxima iremos buscar os valores que devolvemos na função criada para calcular a Transformada Discreta de Fourier, e procuramos saber o valor máximo absoluto que corresponde à amplitude máxima do sinal aplicando a função *max* à transformada de Fourier. A frequência máxima corresponderá à frequência cujo índice no vector é o mesmo que o índice na transformada.

### 2.3.

Neste exercício iremos introduzir ruído uniforme fora da banda de frequência do sinal, ou seja, entre 8.5 – 9.5 KHz; para isso iremos criar índices negativos e positivos entre a frequência máxima e a frequência mínima, sendo que os índices positivos têm de estar entre *fmin* e *fmax* e os índices negativos entre - *fmax* e – *fmin*.

Tendo agora os índices, iremos obter os valores absolutos a partir de números imaginários, sendo que para definir a amplitude máxima do espectro do ruído temos de utilizar apenas 10% (0.1) da amplitude máxima do espectro do sinal. Definimos assim no MATLAB a variável `xabs = 0.1*maxamp*rand(size(ind_pos))`, em que rand calcula um valor aleatório até ao tamanho do índice positivo.

De seguida pretendemos determinar o ângulo que irá ser calculado a partir de `xang = 2*pi*rand(size(ind_pos))-pi`. Com isto podemos agora definir os valores positivos e negativos para que se possa representar o espectro do sinal com ruído. Depois, ao adicionarmos ao índice de valores do espectro do sinal os valores do índice do espectro do ruído, iremos ter um espectro do sinal com ruído entre os 8.5 e 9.5 KHz:



### 2.4.

Para obtermos o sinal com ruído no domínio temporal, temos de calcular a inversa da Transformada de Fourier Discreta para os índices calculados anteriormente do sinal no MATLAB a partir de `xr = real(ifft(ifftshift(Xr)))`, ou seja, reconstruímos o sinal com ruído, para poder ser comparado ao sinal original através da função do MATLAB *wavplay*.

### 2.5.

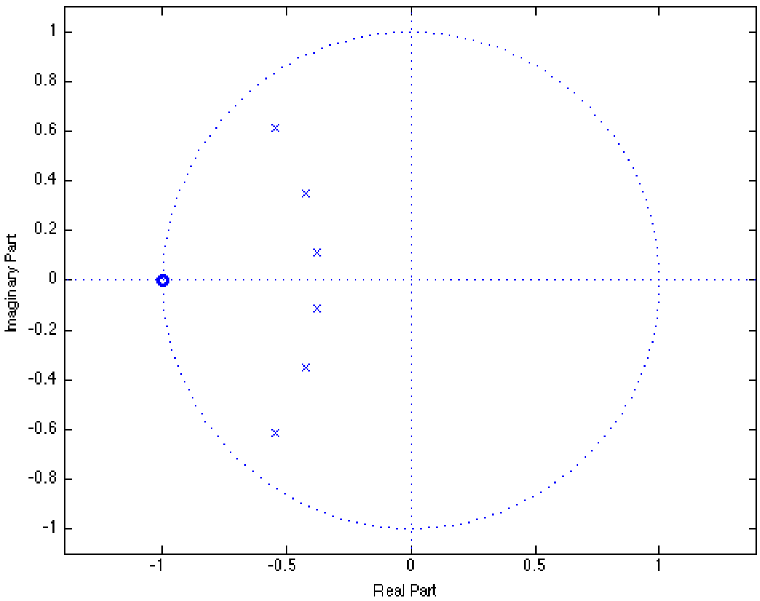
Anteriormente foi recriado um sinal com ruído entre os 8.5 KHz e os 9.5 KHZ, mas para eliminar esse efeito teremos de implementar um filtro de Butterworth de ordem 6 com uma frequência de corte ($f_c$) a rondar os 8 KHz, sendo ω(w) dado por $2*f_c / f_s$. Assim obtemos todos os elementos essenciais para criar o filtro através da função do MATLAB `butter(6, wn)`, que nos irá devolver os coeficientes do filtro, tanto o numerador b como o denominador a. Para filtrarmos agora o sinal com ruído, temos de aplicar a função *filter* do MATLAB, através dos coeficientes anteriormente obtidos e com a inversa de *fft* obtendo assim o sinal filtrado. Por fim reconstruímos os sinal aplicando *fftshift(fft(xfiltrado))* para recriarmos o sinal filtrado. Calculando os pólos e os zeros dos coeficientes obtivemos os seguintes valores:

| Pólos (a): | Zeros (b): |
|---|---|
| -0.5440 + 0.6129i | -1.0048 |
| -0.5440 - 0.6129i | -1.0024 + 0.0042i |
| -0.4236 + 0.3493i | -1.0024 - 0.0042i |
| -0.4236 - 0.3493i | -0.9976 + 0.0042i |
| -0.3756 + 0.1134i | -0.9976 - 0.0042i |
| -0.3756 - 0.1134i | -0.9952 |

Por fim obtemos os coeficientes do numerador e do denominador de $G(Z) = \frac{Y(z)}{X(z)}$

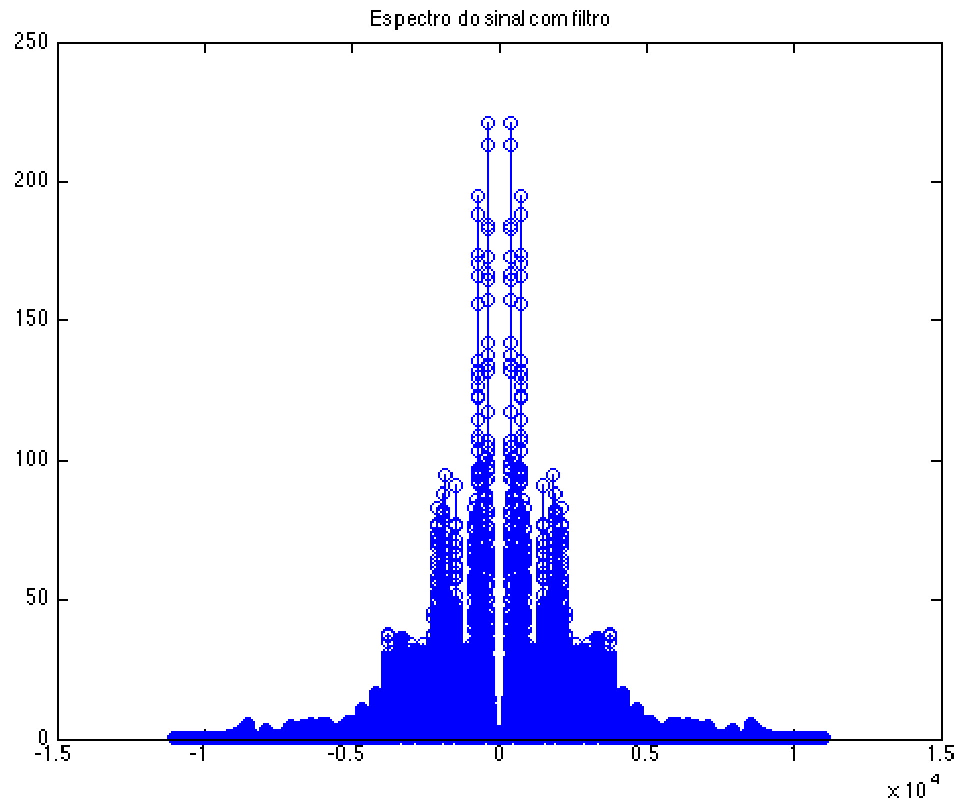
Coeficientes do numerador: [0.1765, 1.0589, 2.6473, 3.5298, 2.6473, 1.0589, 0.1765]

Coeficientes do denominador: [1.0000, 2.6863, 3.5022, 2.6179, 1.1677, 0.2901, 0.0312]



## 2.6.

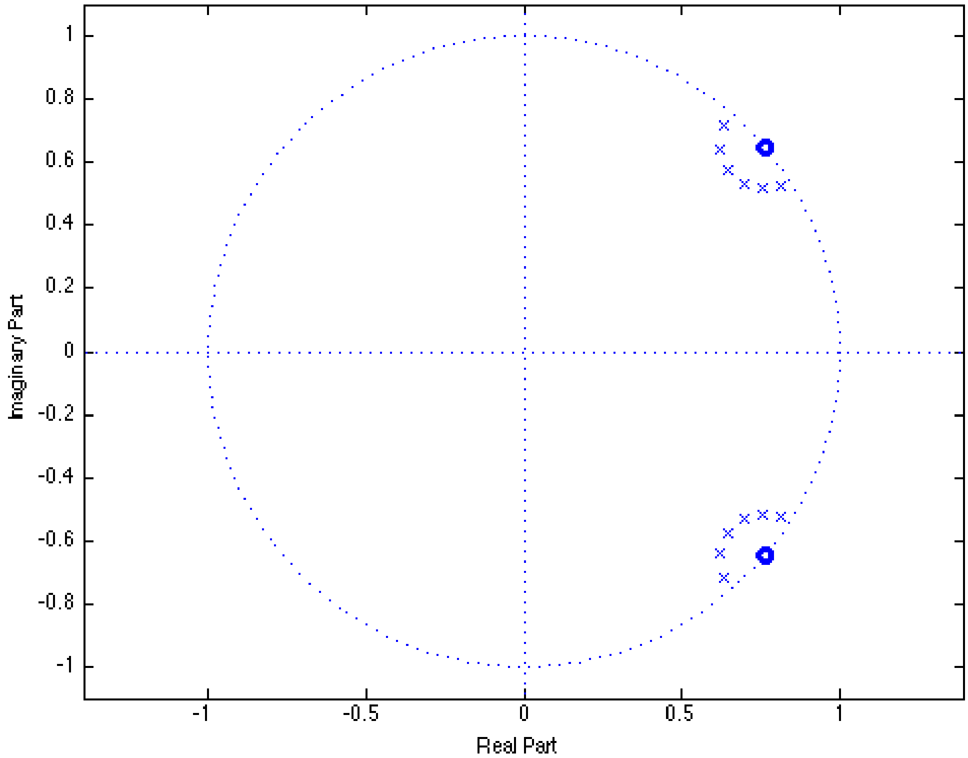
Neste exercício iremos repetir todos os passos anteriormente realizados, alterando apenas o ruído para valores entre 2 e os 3 KHz.




| Pólos: | Zeros: |
|---|---|
| 0.6335 + 0.7194i | 0.7678 + 0.6503i |
| 0.6335 - 0.7194i | 0.7678 - 0.6503i |
| 0.8150 + 0.5258i | 0.7612 + 0.6502i |
| 0.8150 - 0.5258i | 0.7612 - 0.6502i |
| 0.6205 + 0.6425i | 0.7712 + 0.6446i |
| 0.6205 - 0.6425i | 0.7712 - 0.6446i |
| 0.7561 + 0.5157i | 0.7580 + 0.6445i |
| 0.7561 - 0.5157i | 0.7580 - 0.6445i |
| 0.6456 + 0.5749i | 0.7679 + 0.6389i |
| 0.6456 - 0.5749i | 0.7679 - 0.6389i |
| 0.6960 + 0.5312i | 0.7613 + 0.6389i |
| 0.6960 - 0.5312i | 0.7613 - 0.6389i |

Coeficientes do numerador de G(Z): [0.5753 -5.2782 23.6292 -67.5290 136.5175 -205.0524 234.2814 -205.0524 136.5175 -67.5290 23.6292 -5.2782 0.5753]

Coeficientes do denominador de G(Z): [1.0000 -8.3335 33.8853 -88.0133 161.8510 -221.3534 230.5152 -184.0844 111.9360 -50.6196 16.2070 -3.3150 0.3310]




## Conclusão

Ao compararmos o espectro do sinal com ruído entre os 8.5 a 9.5 KHz com o espectro do sinal com ruído entre os 2 e os 3 KHz verificamos que o primeiro apenas afecta o sinal por diminuir a qualidade do som, mas que no segundo caso o sinal é muito afectado, já que uma vez removido algum ruído nessas zonas estaremos também a remover frequências do sinal.

## Exercício 3

### 3.1.

No exercício 3.1, para ler a imagem pretendida no MATLAB, utilizámos a função *imread* que recebe a imagem a ser lida, e devolve uma matriz com os tamanhos da imagem h*w, em que cada valor dessa matriz contém um índice referente ao índice de cores da imagem, que é também devolvido nesta função na forma de vector.

### 3.2

No exercício 3.2, através da leitura da matriz da imagem lida e do vector com as cores das imagens, podemos recriar a imagem através da função do MATLAB *imshow* onde obtivemos a seguinte imagem:

De seguida, para calcularmos as frequências da imagem, utilizamos as funções *fftshift(fft2())*, onde obtivemos a componente de frequência 0 no meio do espectro de forma a visualizar a transformada discreta de Fourier.

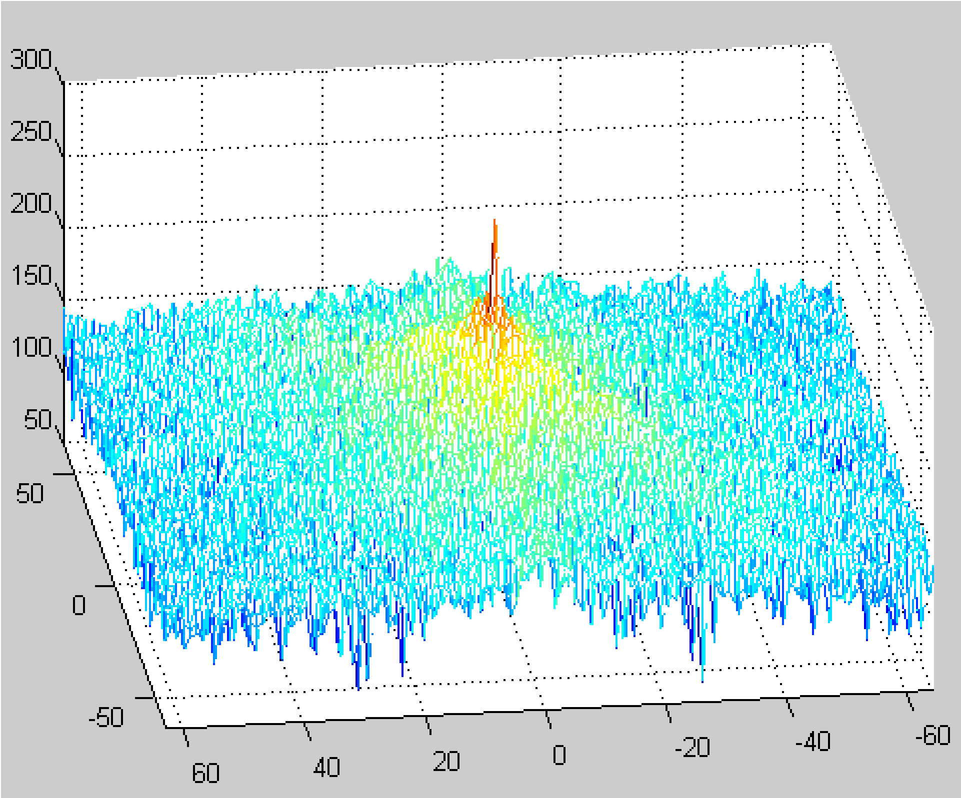
Como precisamos do tamanho da matriz da imagem após este passo, utilizámos a função *size* que nos vai devolver o número de linhas e o número de colunas; estas irão ser utilizadas para calcular o eixo de x e o eixo de y (tanto para os valores pares como para os ímpares).

### 3.3

Ajustando os eixos de x e de y para os valores de frequência obtidos anteriormente pela imagem, iremos representar numa malha de pontos os valores absolutos das frequências dos pontos da imagem, nos eixos acima mencionados.

Já para encontrarmos a cor média da imagem, teremos de procurar no ponto da malha onde a frequência é de (0,0) onde obtivemos um valor de 124.2999.

A malha de pontos toma a seguinte forma:




### 3.4-8

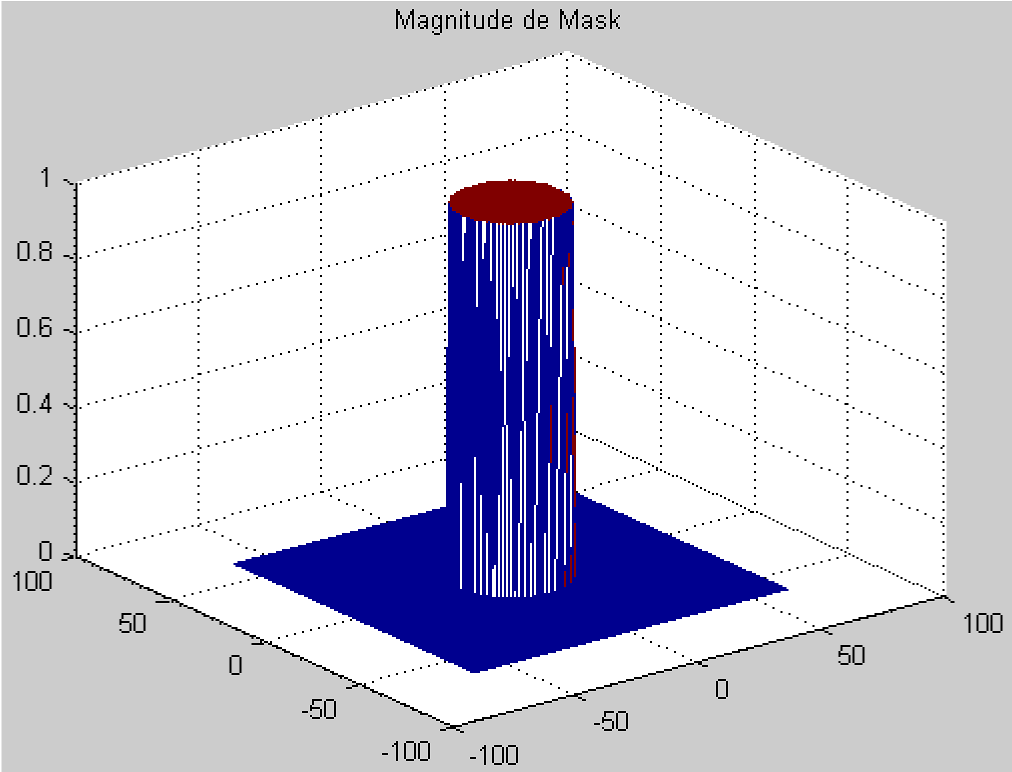
Neste exercício pretende-se criar um filtro ideal, de modo a que o utilizador possa escolher entre um filtro Passa – Baixo e um filtra Passo – Alto. Para isso começamos por criar um vector preenchido com zeros com o tamanho da matriz dos pontos da imagem anteriormente calculados com a transformada discreta de Fourier. De seguida damos a opção ao utilizador de escolher no menu entre esses filtros disponíveis e pedimos ainda a frequência de corte para os limites do filtro ideal na linha de comandos.

Depois de escolhidos os parâmetros anteriormente descritos, iremos percorrer a matriz ponto a ponto para calcular a distância entre esse ponto e o ponto de cor média, sendo que, se este for menor que a frequência de corte, então o vector *mask* nesse ponto vai ser uniformizado.

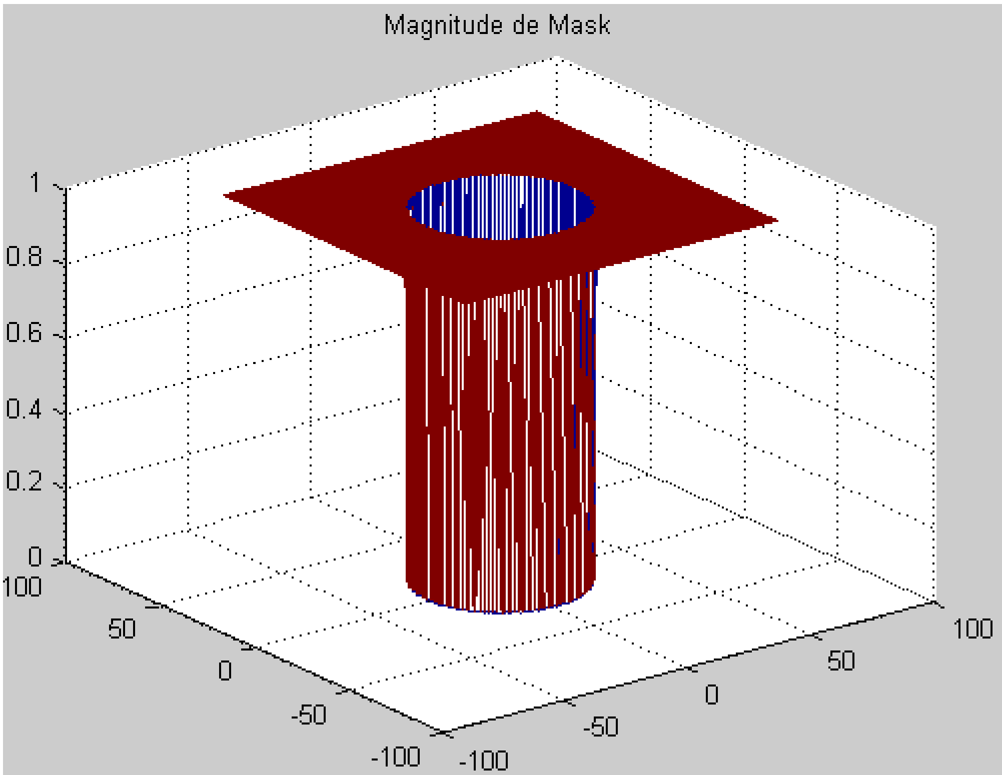
Caso o filtro seja = 2 (Passa-alto) ir-se-á subtrair os valores uniformizados a *mask* para obtermos os valores que não são uniformizados, usando uma constante = 10 para multiplicar pela imagem filtrada.

Ao apresentarmos a malha de pontos segundo os cálculos obtidos anteriormente, obtivemos uma malha de pontos que corresponde a Magnitude dos espectros do vector do filtro ideal.

**Passa-Baixo com $f_c$ = 20**

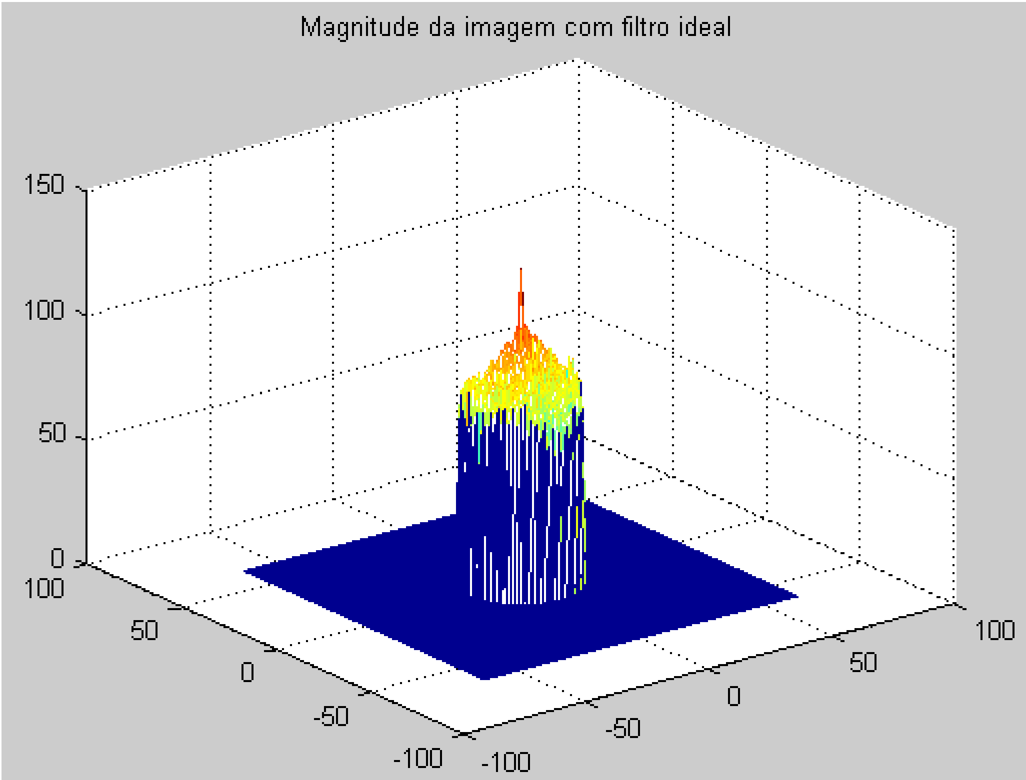



**Passa-Alto $f_c$ = 30**

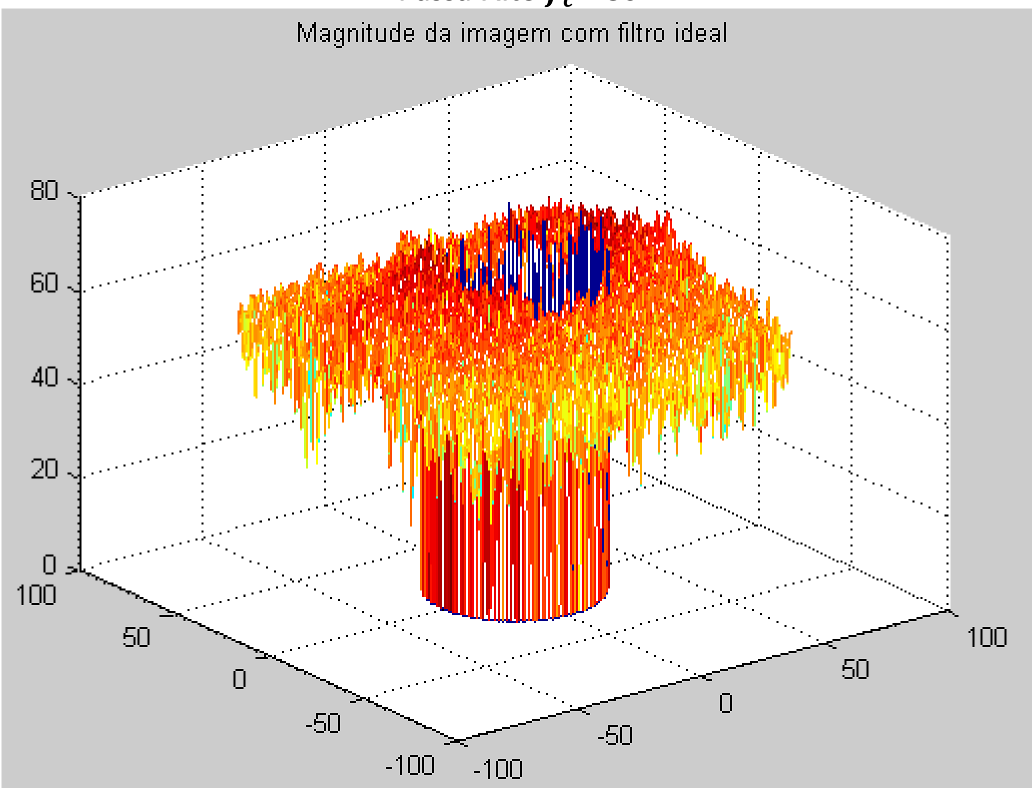



Onde a área a vermelho corresponde à área da máscara que se pretende guardar e azul a área que se pretende remover. Ao multiplicarmos a convolução entre a imagem original e a imagem da máscara, obteremos uma malha de pontos da imagem transformada filtrada, com as áreas conforme o filtro acima indicado guardadas ou removidas. Obtivemos os seguintes resultados:

**Passa-Baixo com $f_c$ = 20**

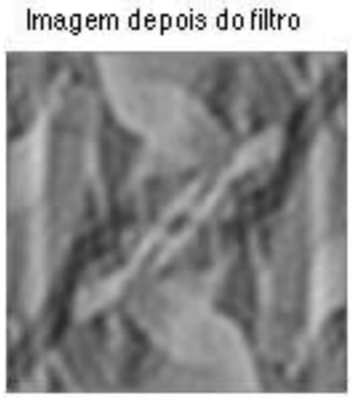



**Passa-Alto $f_c$ = 30**



Depois de aplicada a Inversa da Transformada Discreta de Fourier à imagem filtrada e de reconstruir a imagem proveniente desses cálculos obtivemos as seguintes imagens filtradas:

**Passa-Baixo com $f_c$ = 20**




**Passa-Alto $f_c$ = 30**


## 3.9

Ao analisarmos as imagens podemos verificar que, no filtro passa-baixo, as transições entre as cores são atenuadas, tal como esperado, já que as frequências mais altas indicam grandes transições, e são precisamente essas que são removidas.

No caso do filtro passa-alto verificamos o caso contrário, em que apenas as frequências mais altas se mantêm, e portanto são realçados os contornos da imagem.

## Exercício 4

Para a resolução deste exercício, tal como dos exercícios 2 e 5, foram criadas 2 funções que serão necessárias em várias alíneas: *wavplot.m* e *STFT.m*.

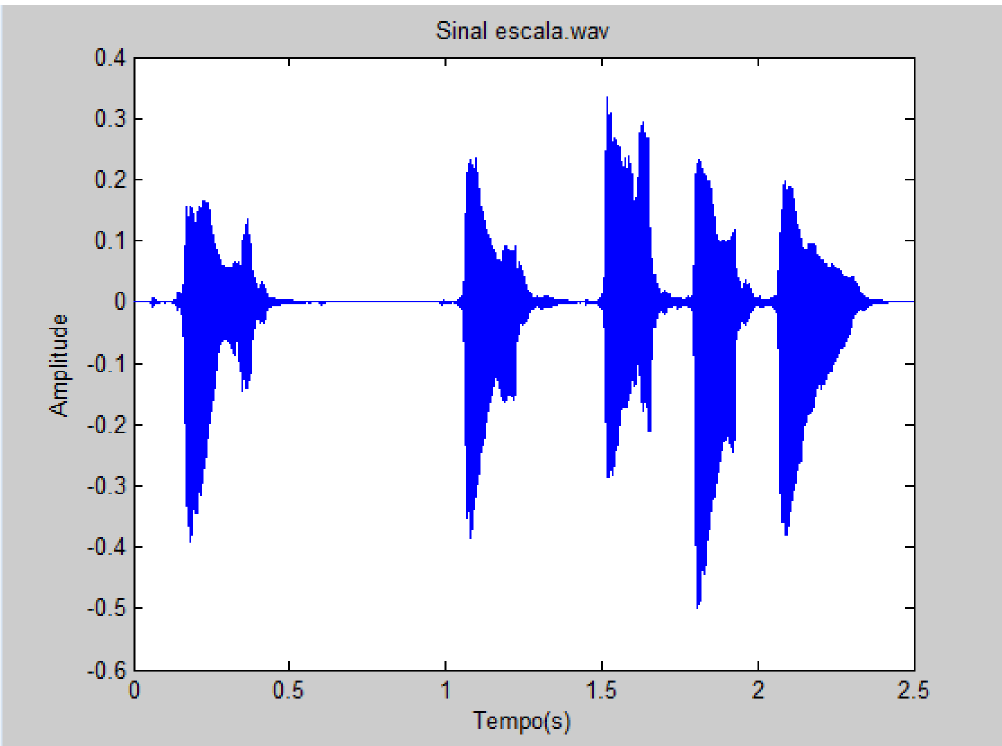
A função *wavplot.m* tem como entrada um sinal *xt* e a sua frequência *fs*, e irá simplesmente fazer o *plot* do sinal e do seu espectro em magnitude. Para fazer o primeiro gráfico calcula-se simplesmente a duração do sinal para definir o eixo xx, e faz-se o *plot* de *xt*. No segundo caso é feito um *shift* à transformada de Fourier do sinal, e é desenhado o gráfico da sua magnitude.

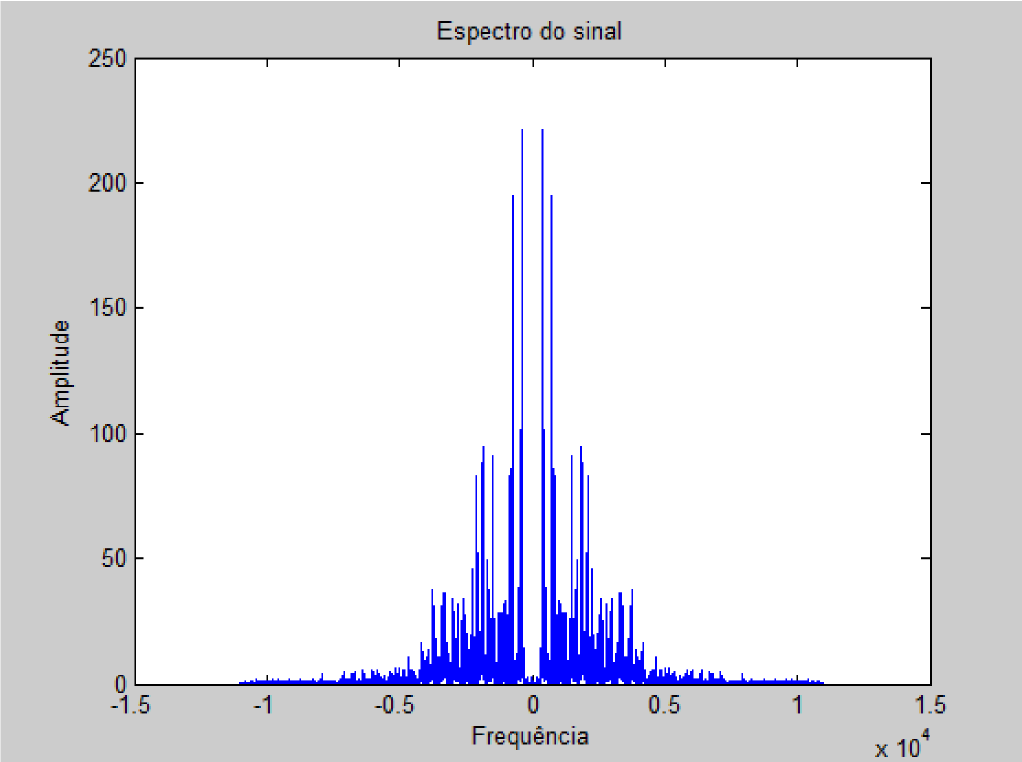
Por sua vez a função STFT.m recebe como parâmetros de entrada o sinal, a sua frequência fs, uma flag para o caso de querermos reconstruir o sinal, a janela a usar e a sobreposição da mesma; esta função devolve um vector com as frequências fundamentais do sinal calculadas pela Short-Time Fourier Transform, e o sinal aproximado por estas, caso a flag seja diferente de 0.

Nesta segunda função as janelas e sobreposição são primeiro convertidas para intervalos de frequências, permitindo assim o cálculo do domínio das frequências fundamentais.

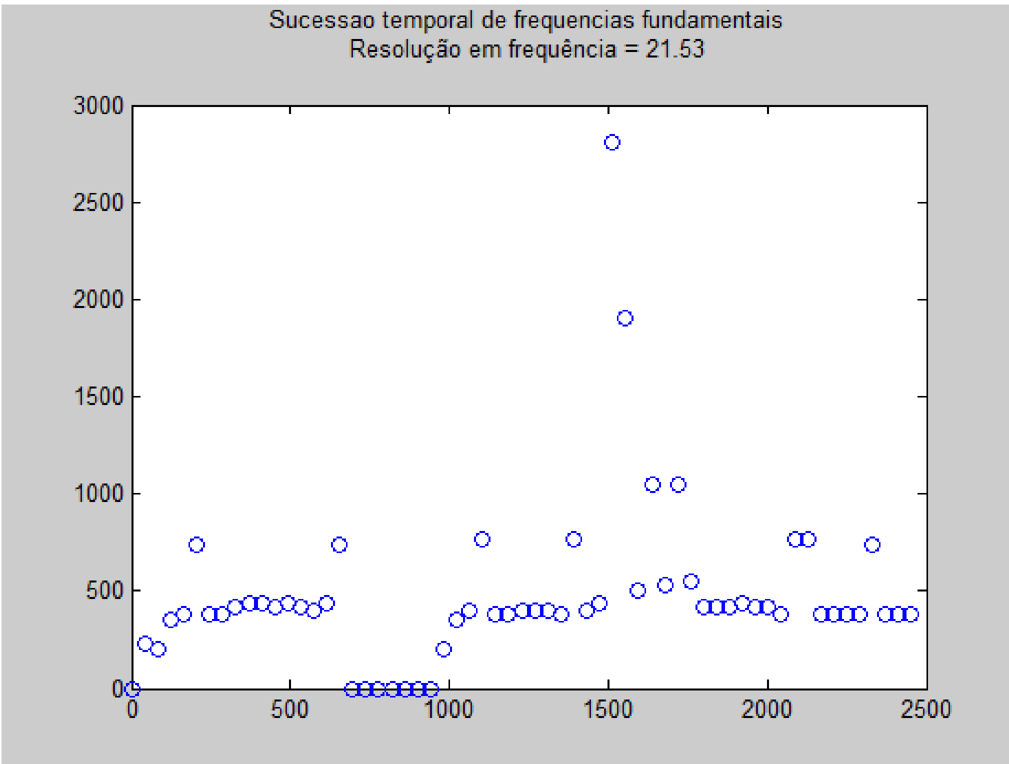
Seguidamente entra-se num ciclo, no qual se definem os limites máximo e mínimo da janela, se selecciona a parte pretendida do sinal, e se calcula a sua janela de Hamming e respectiva transformada. Calcula-se depois o máximo das frequências obtidas, que vão no fim do ciclo ser filtradas, para serem só guardadas as frequências acima dos 100 Hz.

### 4.1.

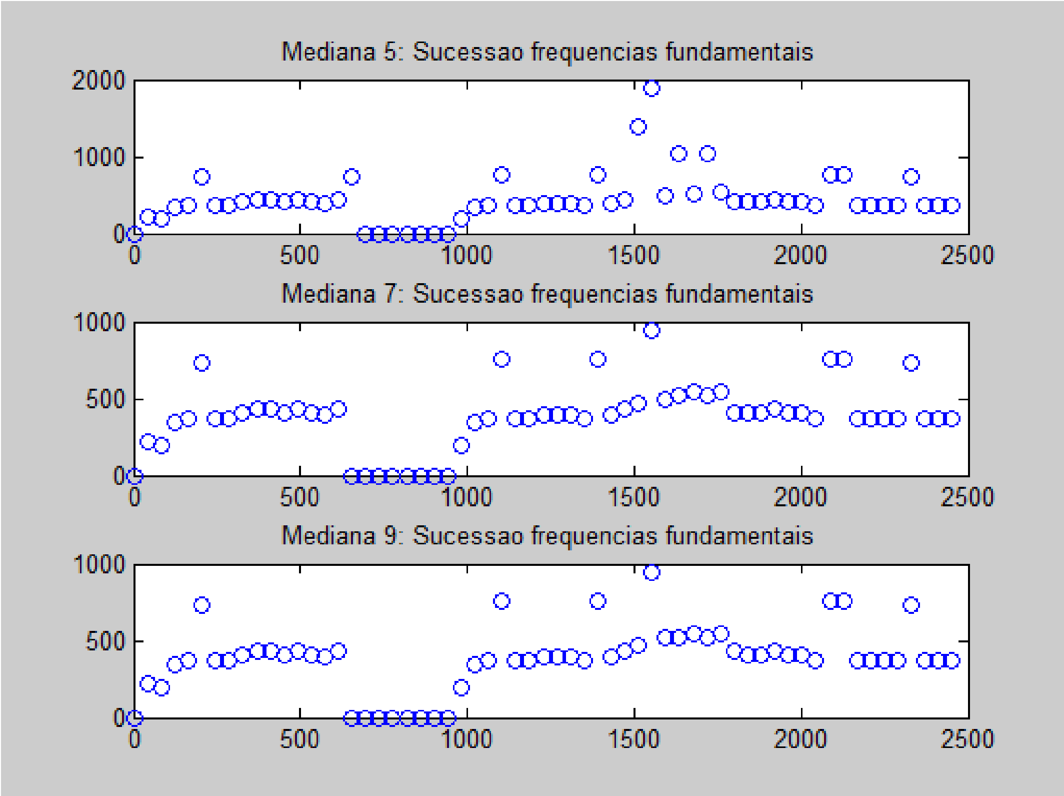



Espectro do sinal
Amplitude
Frequência
x 10^4
0
50
100
150
200
250
-1.5
-1
-0.5
0
0.5
1
1.5

### 4.2.

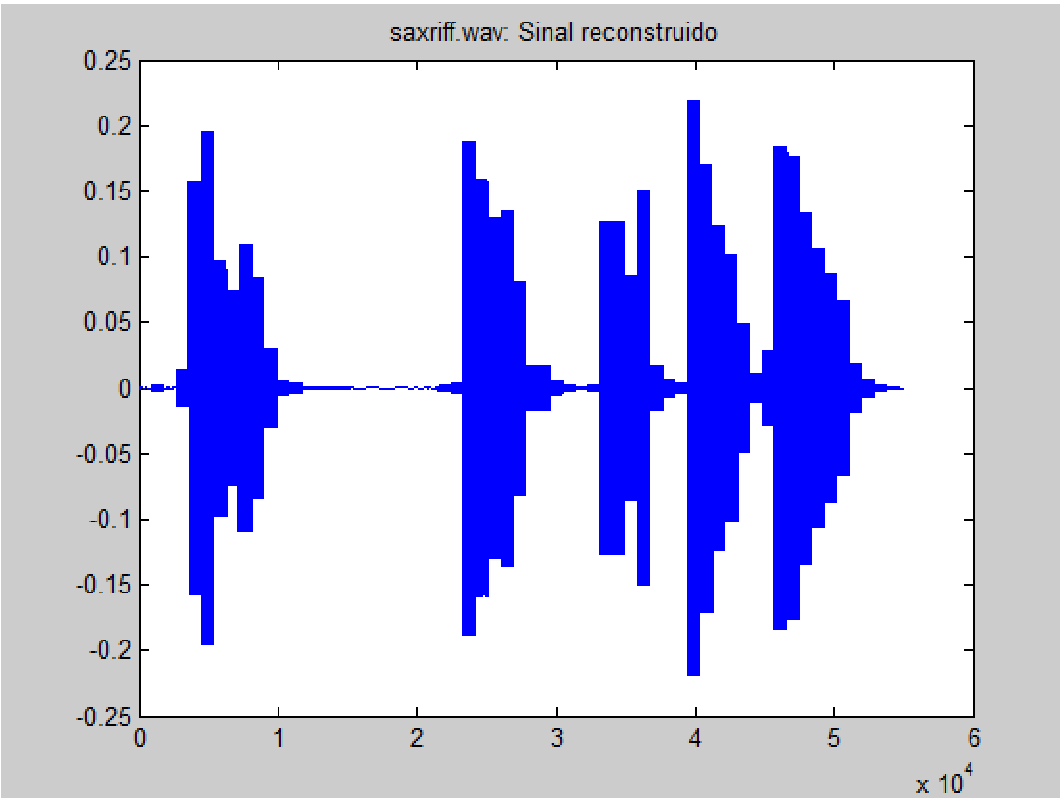

Sucessao temporal de frequencias fundamentais
Resolução em frequência = 21.53
0
500
1000
1500
2000
2500
3000
0
500
1000
1500
2000
2500

## 4.3.

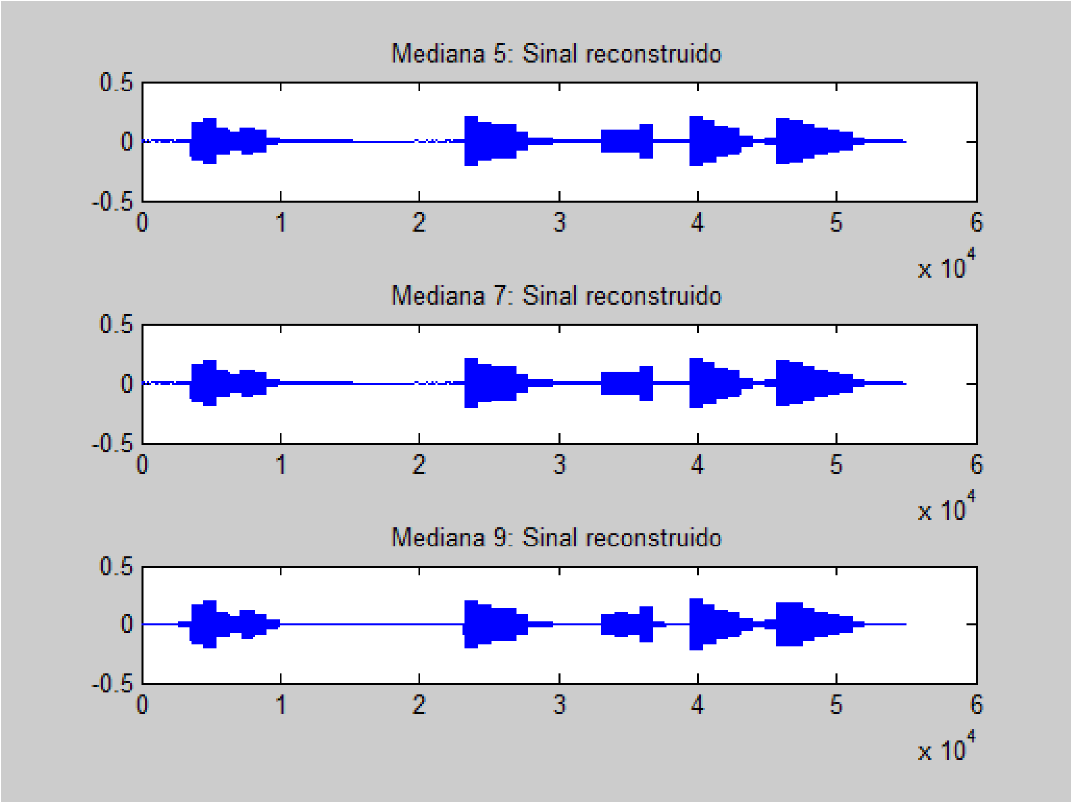


Mediana 5: Sucessao frequencias fundamentais
2000
1000
0
0
500
1000
1500
2000
2500
Mediana 7: Sucessao frequencias fundamentais
1000
500
0
0
500
1000
1500
2000
2500
Mediana 9: Sucessao frequencias fundamentais
1000
500
0
0
500
1000
1500
2000
2500


## 4.4



saxriff.wav: Sinal reconstruido
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
0
1
2
3
4
5
6
x $10^4$




### 4.5.

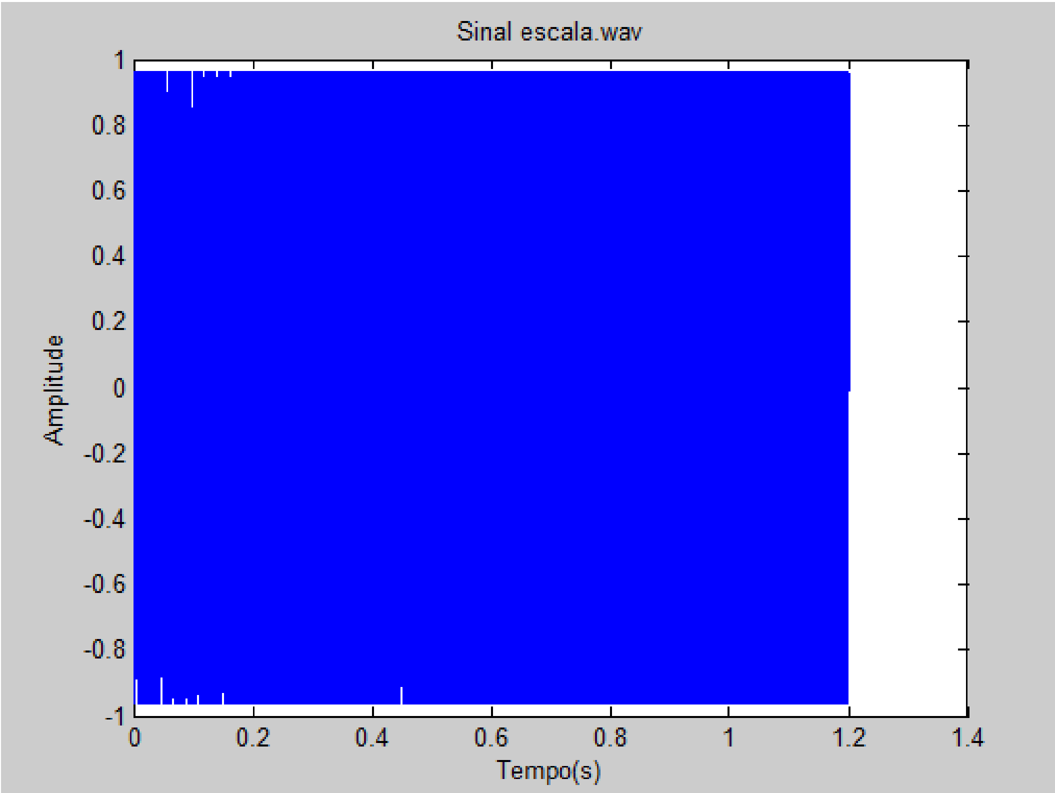
Ouvindo os sinais reconstruídos pode-se verificar que, como seria de esperar, ao sintetizar os sons se perde alguma informação. Ao tocar um saxofone ocorrem sempre variações de frequências, fazendo essa variação parte do estilo inerente do saxofone. Estas variações devem-se a alterações na respiração e às características do instrumento.

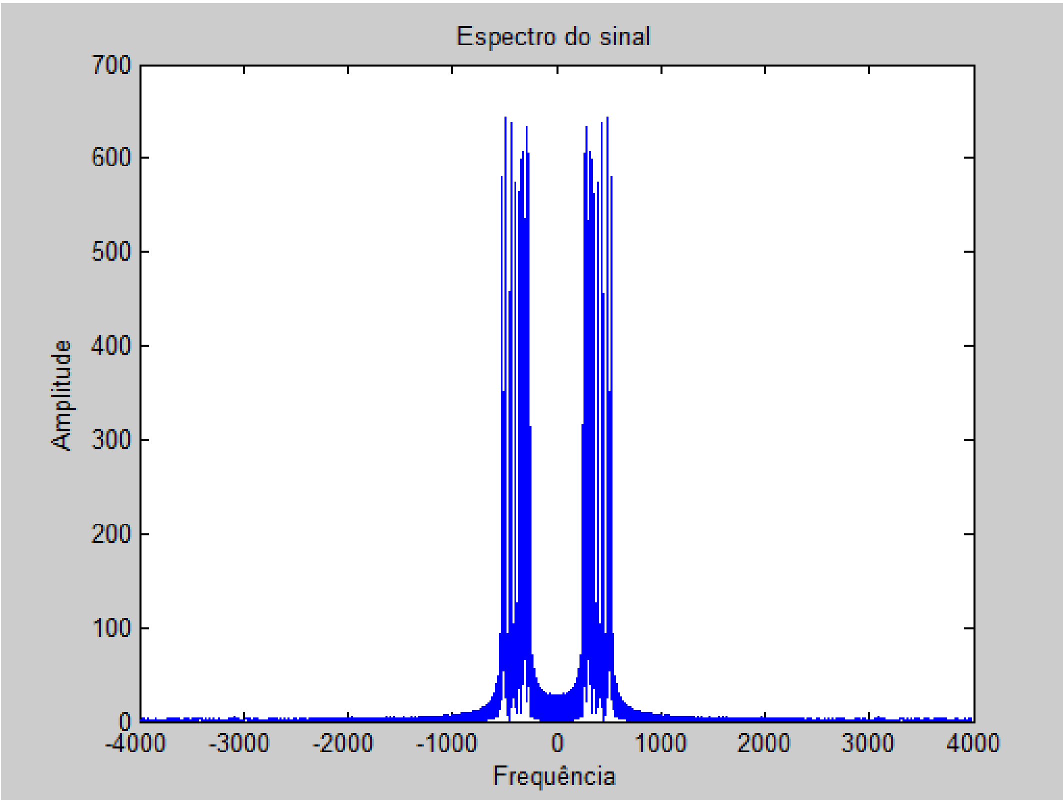
Ao sintetizar um som com tais características é normal registarem-se frequências que não correspondem às notas musicais pretendidas, e que, após uma reconstrução do sinal, serão salientadas, tendo a mesma intensidade que as frequências pretendidas, o que é prejudicial à análise da informação.

Graficamente pode-se observar que o sinal foi claramente reconstruído em janelas discretas com uma dimensão considerável em relação ao sinal inicial, que, apesar de discreto, tinha uma frequência de amostragem com que não se perdia informação.

## Exercício 5

### 5.1

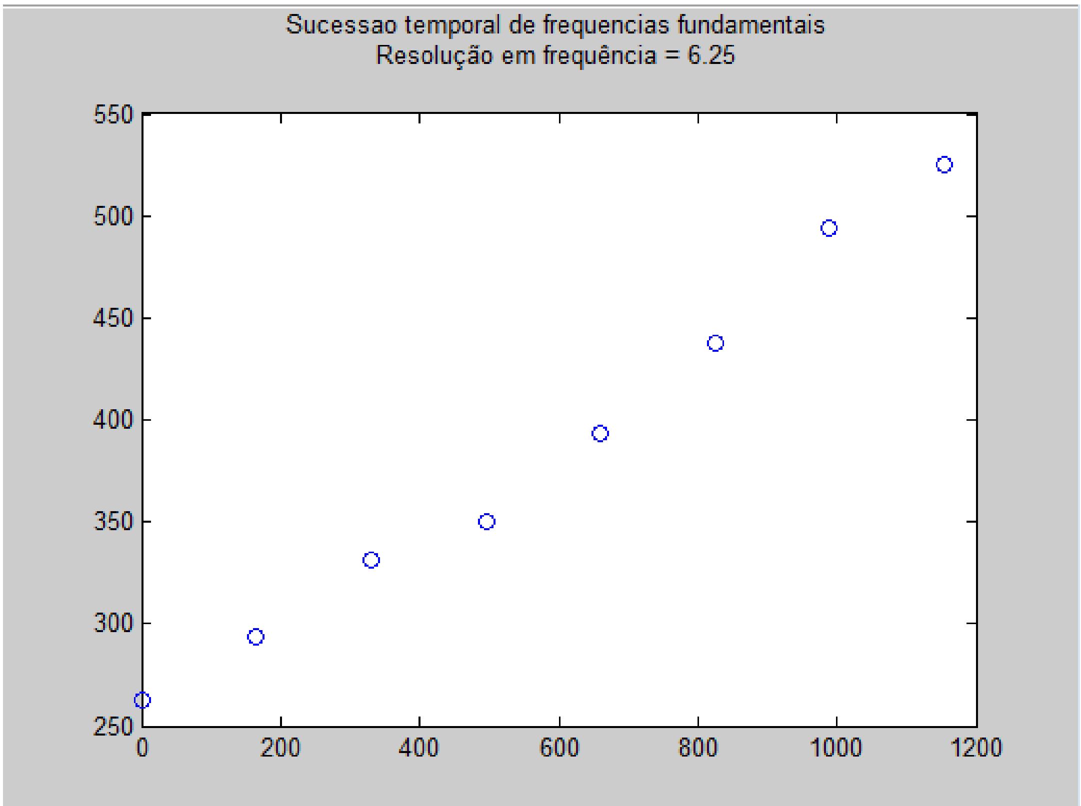


### 5.2

Após vários testes concluiu-se que a melhor janela a usar para encontrar as verdadeiras frequências seria de 160 ms, com sobreposição de 12.5 ms, dado que outras janelas ou repetiam a mesma nota, ou não detectavam uma nota (normalmente o último Dó).

Neste caso foi feita uma função *notasmusicais.m* que detecta qual a nota correspondente a uma frequência num vector destas. Para facilitar esta detecção foram calculados os pontos médios entre as várias frequências, de modo a só termos de detectar o último valor estritamente maior que a frequência fundamental, tendo de reduzir previamente a frequência à oitava principal. Os nomes das notas são também armazenados, usando-se a nomenclatura germânica para a sua classificação.

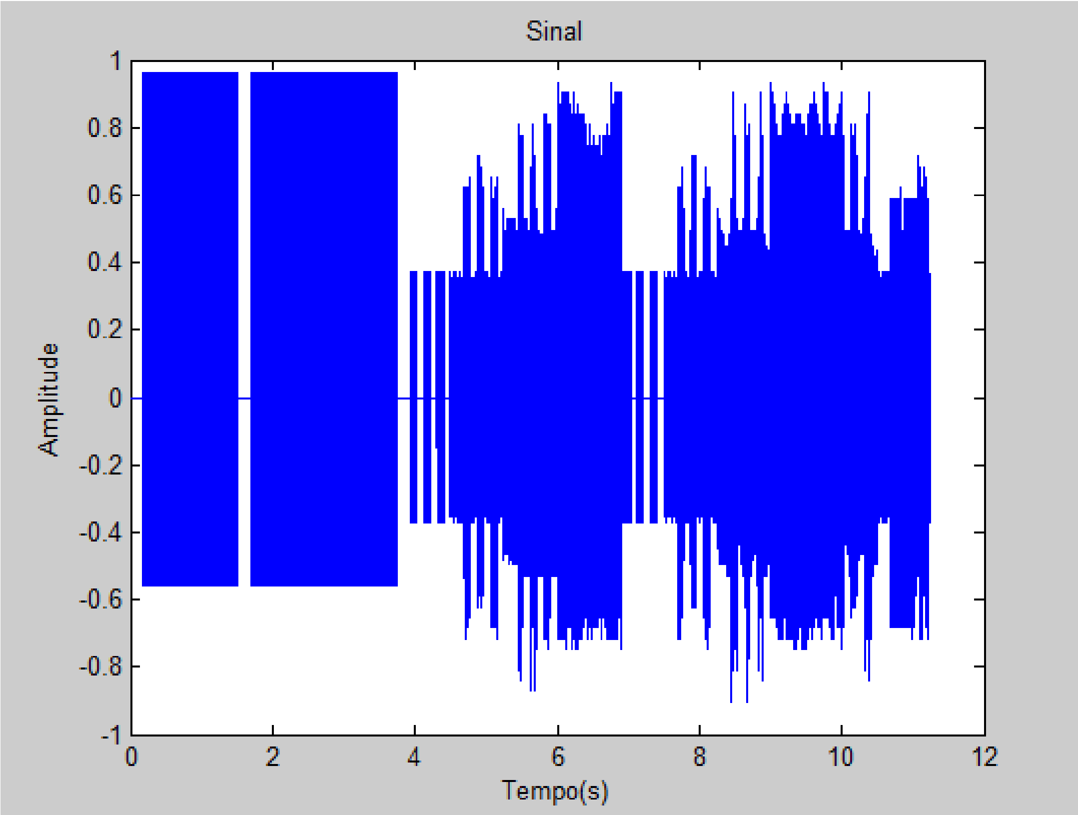


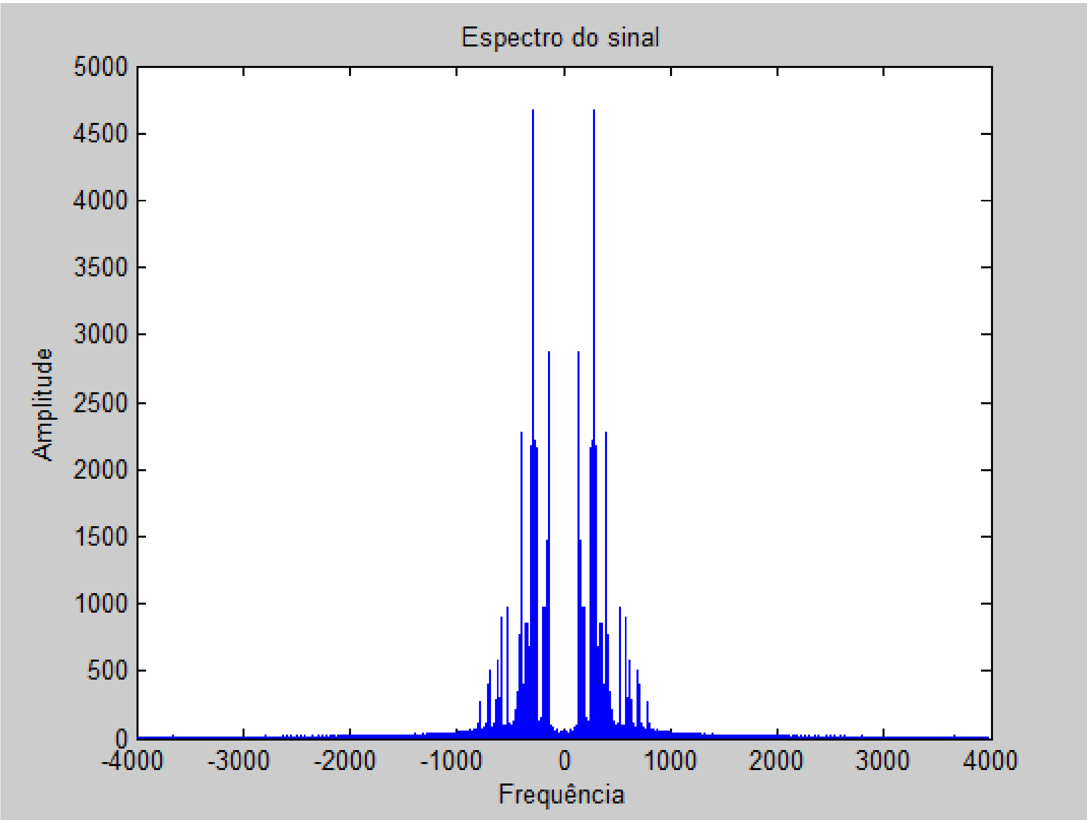

```
escala.wav: Por favor insira o tamanho da janela em ms: 160
escala.wav: Por favor insira o tamanho da sobreposição em ms: 12.5

      C        D        E        F        G        A        B        C
 262.50   293.75   331.25   350.00   393.75   437.50   493.75   262.50 >>
```

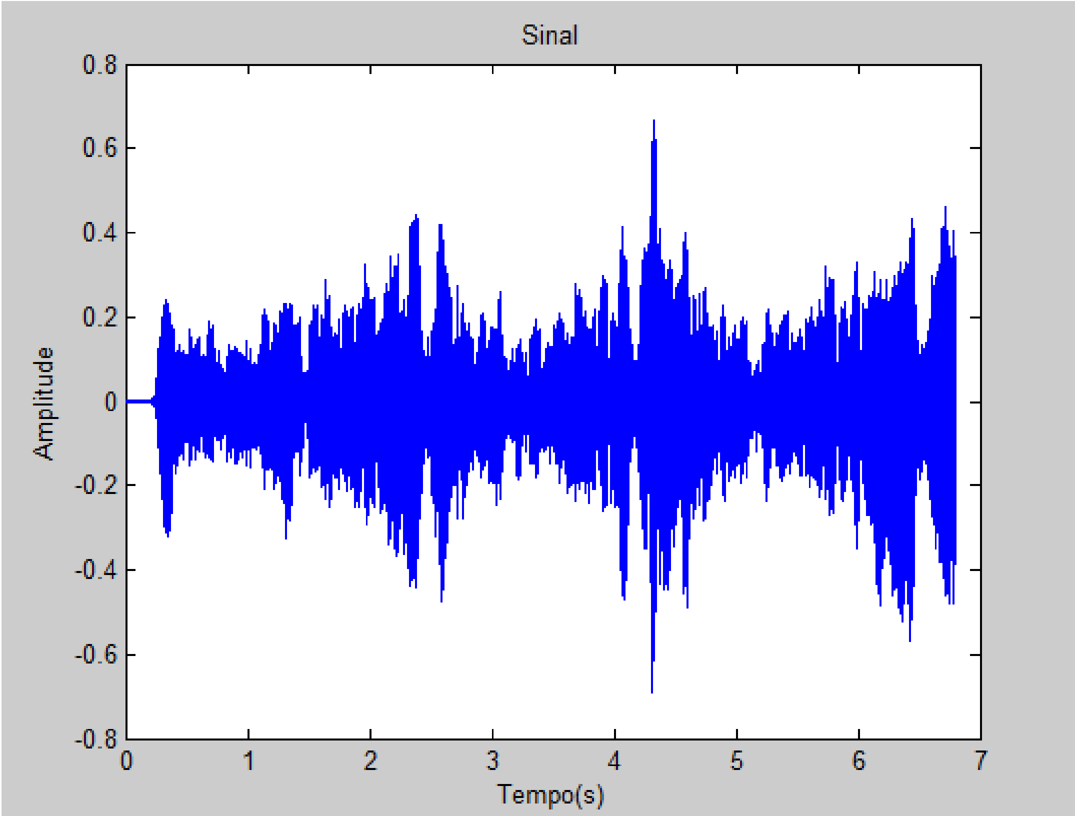
## 5.3.

**Piano.wav**

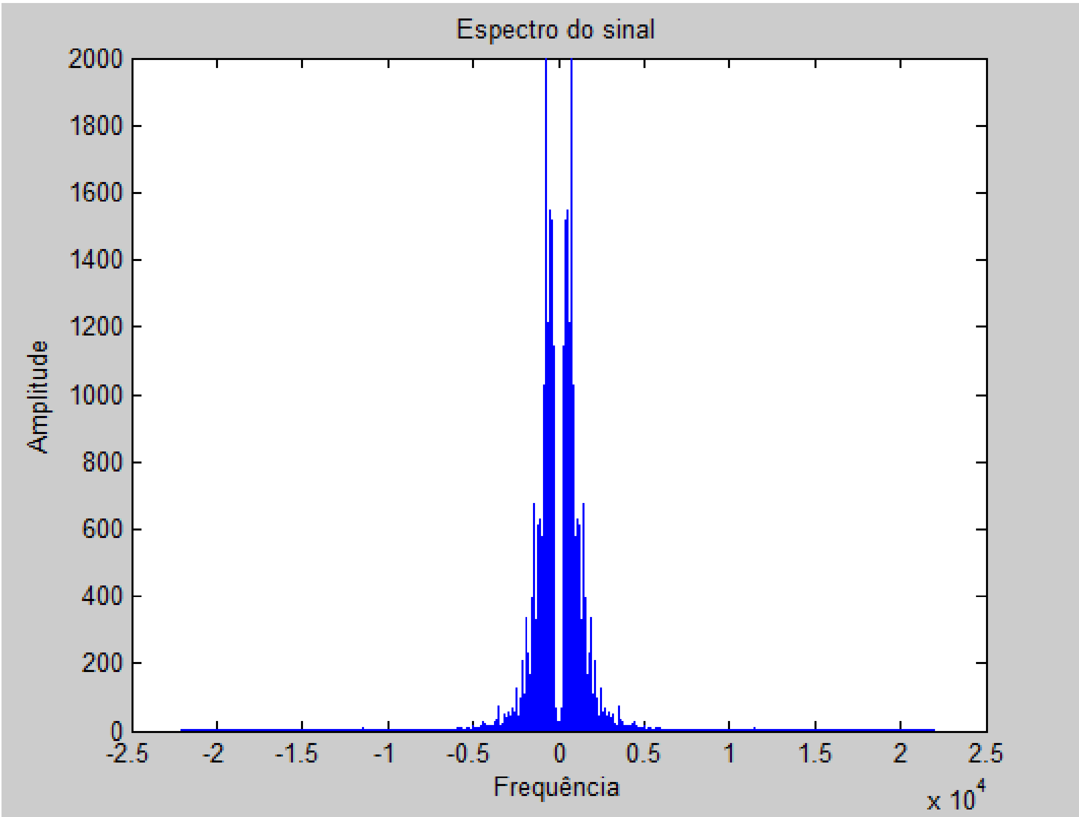

Sinal
Amplitude
Tempo(s)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
2
4
6
8
10
12


Espectro do sinal
Amplitude
Frequência
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000

**Flauta.wav**



Sinal
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
Amplitude
0
1
2
3
4
5
6
7
Tempo(s)

Espectro do sinal
2000
1800
1600
1400
1200
1000
800
600
400
200
0
Amplitude
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
Frequência
x 10^4

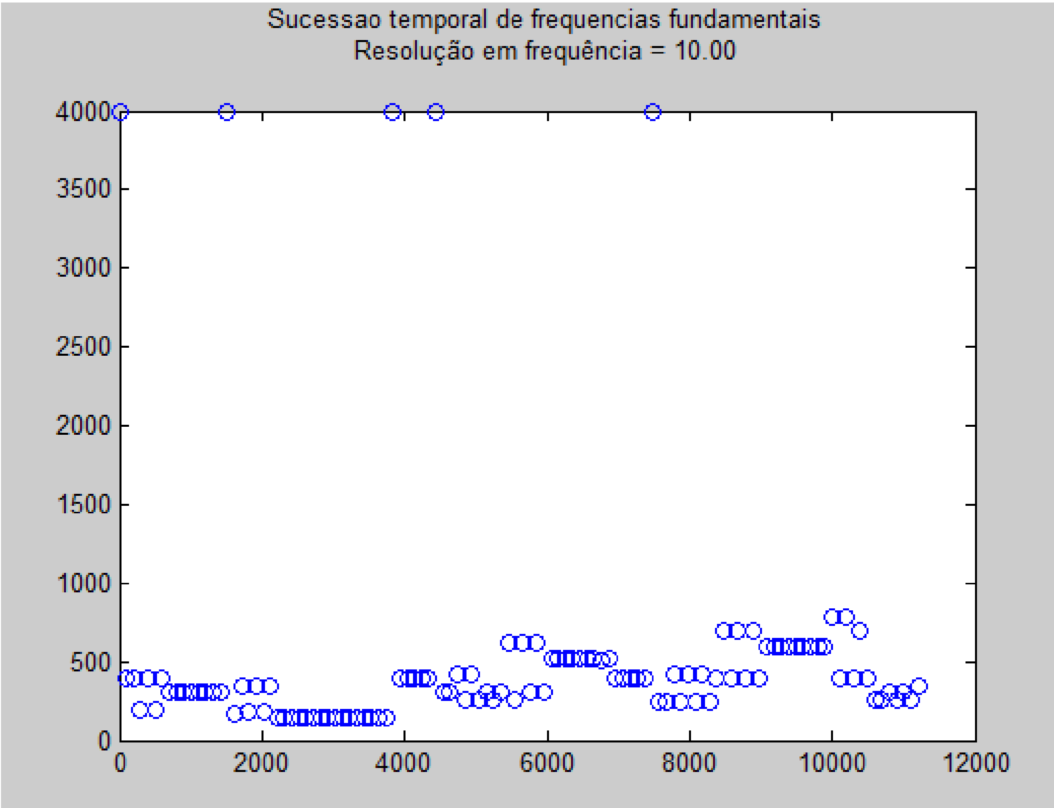
## 5.4. e 5.5

**Piano.wav**

No caso do ficheiro piano.wav, após serem testados vários valores para janelas e sobreposições, concluiu-se que, em termos auditivos, a janela mais fiel seria de 100 ms sem sobreposição, já que detecta a grande maioria das frequências, apesar de detectar mais frequências do que o necessário. Detectar as frequências deste sinal é particularmente difícil, já que temos duas ou mais notas tocadas ao mesmo tempo, e portanto haverá frequências simultâneas.

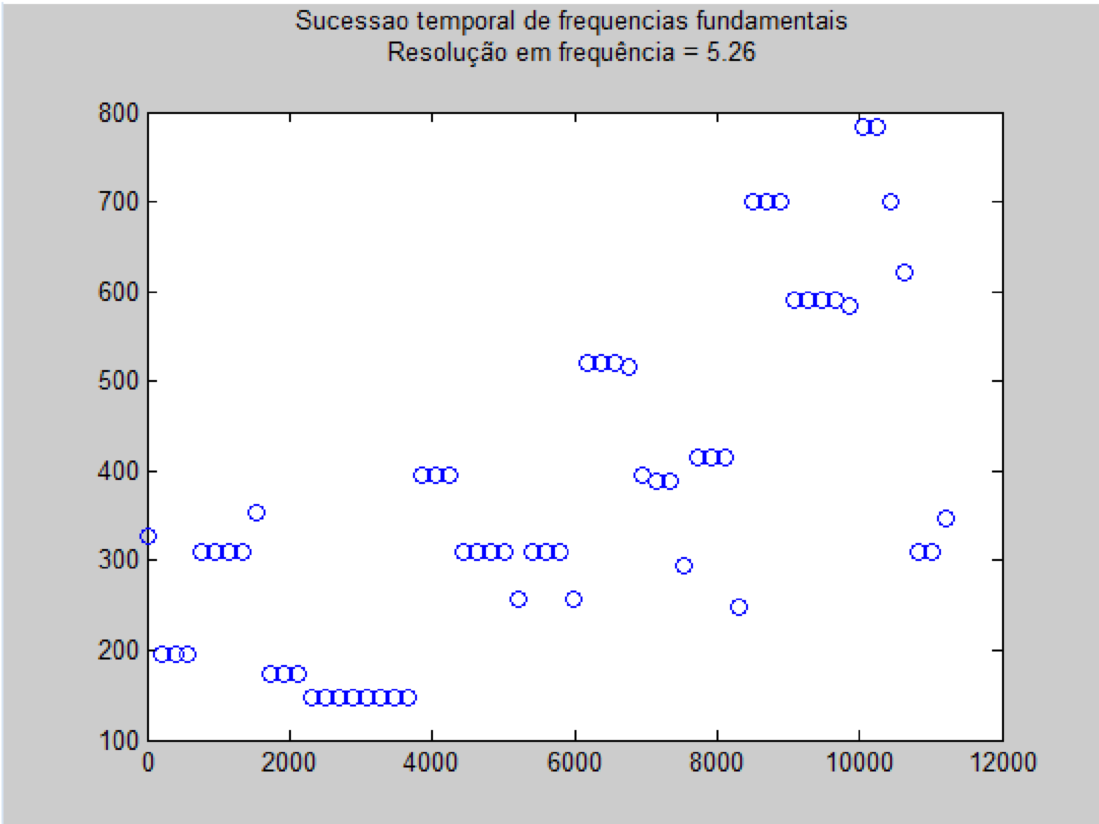
Foi também encontrada a janela de 190 ms, que não detecta frequências a mais, e é bastante fiel ao sinal na parte final da música, sendo no entanto muito difícil distinguir a melodia no resto do sinal.

**Janela: 100 ms** **Sobreposição: 0 ms**




B G G G G G G D# D# D# D# D# D# D# D# B F F F# F F# F D D D D D D D D D D D D D D
D B G G G G G B D# D# G# C G# C D# C D# D# C D# D# D# D# C C C C C C C C C G G G G G
B B B G# B G# B G# B G F G F G F G D D D D D D D D D G G G G F G C C D# C D# C F
498.75 400.00 390.00 400.00 390.00 400.00 390.00 310.00 310.00 310.00 310.00
310.00 310.00 310.00 310.00 498.75 340.00 350.00 360.00 350.00 360.00 350.00
300.00 300.00 300.00 300.00 300.00 300.00 300.00 300.00 300.00 300.00 300.00
300.00 300.00 300.00 300.00 300.00 498.75 390.00 390.00 390.00 390.00 390.00
498.75 310.00 310.00 420.00 260.00 420.00 260.00 310.00 260.00 310.00 310.00
260.00 310.00 310.00 310.00 310.00 260.00 260.00 260.00 260.00 260.00 260.00
260.00 255.00 260.00 390.00 390.00 390.00 390.00 390.00 498.75 500.00 500.00
420.00 500.00 420.00 500.00 420.00 480.00 390.00 350.00 390.00 350.00 390.00
350.00 390.00 295.00 295.00 295.00 295.00 295.00 295.00 295.00 300.00 295.00
390.00 390.00 390.00 390.00 350.00 390.00 260.00 260.00 310.00 260.00 310.00
260.00 350.00

**Janela: 190 ms** **Sobreposição: 0 ms**



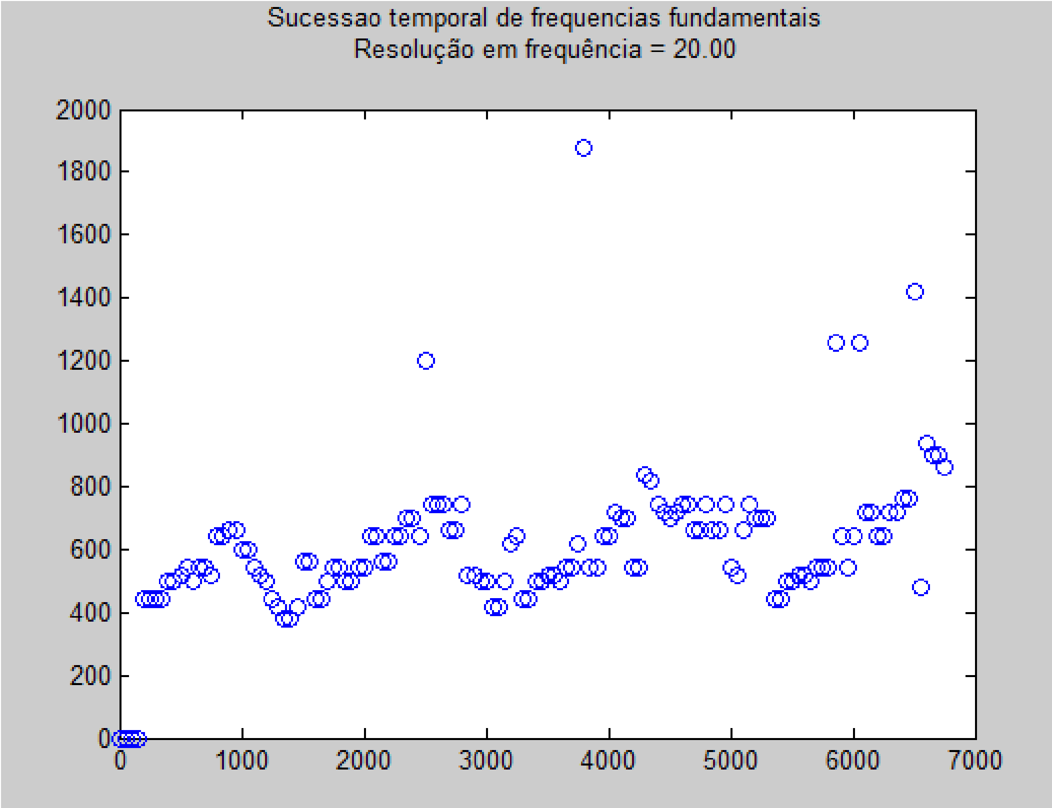

E G G G D# D# D# D# F F F F D D D D D D D D G G G D# D# D# D# C D# D# D# C C C C C G G G D G# G# G# B F F F D D D D D G G F D# D# D# F

326.32 389.47 389.47 389.47 310.53 310.53 310.53 310.53 352.63 347.37 347.37 347.37 294.74 294.74 294.74 294.74 294.74 294.74 294.74 294.74 394.74 394.74 394.74 310.53 310.53 310.53 310.53 257.89 310.53 310.53 310.53 257.89 260.53 260.53 260.53 257.89 394.74 389.47 389.47 294.74 415.79 415.79 415.79 494.74 350.00 350.00 350.00 294.74 294.74 294.74 294.74 292.11 392.11 392.11 350.00 310.53 310.53 310.53 347.37

## Flauta.wav

Já neste caso foi bastante mais difícil encontrar uma janela satisfatória, dado que a melodia é intrincada, tal como a cadência das notas e o seu ritmo. Assim, foram testadas várias janelas, que se podem ver de seguida. Conclui-se que a janela mais adequada será de 100 ms, com sobreposição de 12.5 ms, já que as frequências não aparecem muito repetidas (como nos casos de janelas menores), parecendo, no entanto, que grande parte das frequências é detectada. É no entanto muito difícil perceber se sonoramente está “correcto”, dado que, ao reduzir as frequências à mesma oitava, e com a variedade rítmica e melódica desta peça musical, é quase impossível fazer a comparação auditiva dos dois sons.

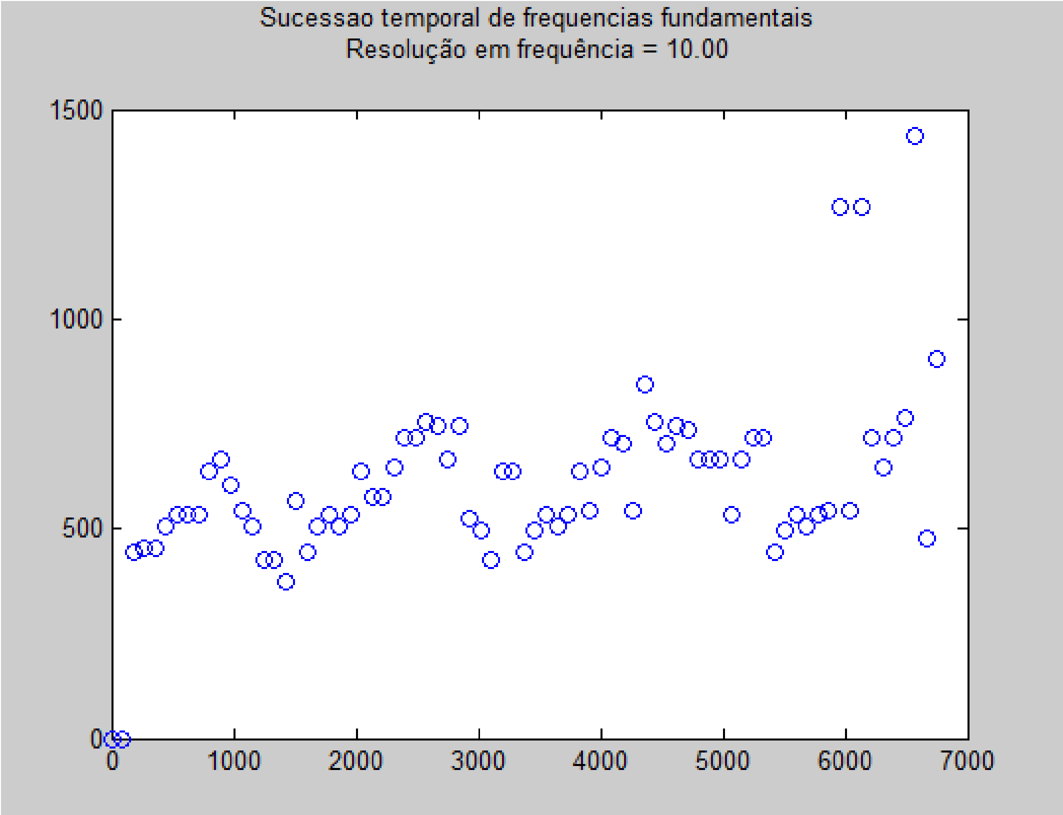
**Janela: 50 ms** **Sobreposição: 0 ms**




A A A A B B C C# B C# C# C D# D# E E D D C# C B A G# F# F# G# C# C# A A B C# C# B B C# C# D# D# C# C# D# D# F F D# D F# F# F# E E F# C C B B G# G# B D# D# A A B B C C B C# C# D# A# C# C# D# D# F# F F C# C# G# G# F# F# F F# F# F# E E F# E E F# C# C E F# F F F A A B B C C B C# C# C# D# D# C# D# D# F# F# D# D# F# F# F# F# F B A# A A A

440.00 440.00 440.00 440.00 500.00 500.00 260.00 270.00 500.00 270.00 270.00 260.00 320.00 320.00 330.00 330.00 300.00 300.00 270.00 260.00 500.00 440.00 420.00 380.00 380.00 420.00 280.00 280.00 440.00 440.00 500.00 270.00 270.00 500.00 500.00 270.00 270.00 320.00 320.00 280.00 280.00 320.00 320.00 350.00 350.00 320.00 300.00 370.00 370.00 370.00 330.00 330.00 370.00 260.00 260.00 500.00 500.00 420.00 420.00 500.00 310.00 320.00 440.00 440.00 500.00 500.00 260.00 260.00 500.00 270.00 270.00 310.00 470.00 270.00 270.00 320.00 320.00 360.00 350.00 350.00 270.00 270.00 420.00 410.00 370.00 360.00 350.00 360.00 370.00 370.00 330.00 330.00 370.00 330.00 330.00 370.00 270.00 260.00 330.00 370.00 350.00 350.00 350.00 440.00 440.00 500.00 500.00 260.00 260.00 500.00 270.00 270.00 270.00 315.00 320.00 270.00 320.00 315.00 360.00 360.00 320.00 320.00 360.00 360.00 380.00 380.00 355.00 480.00 470.00 450.00 450.00 430.00

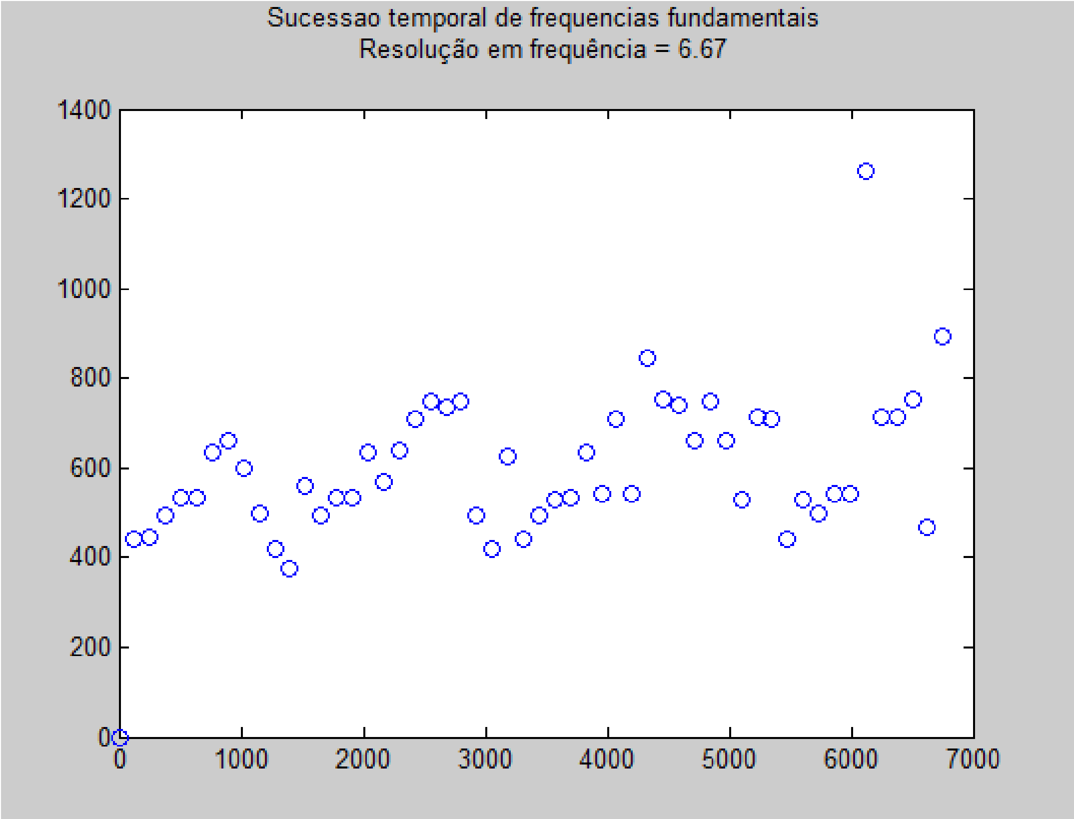
**Janela: 100 ms** **Sobreposição: 12.5 ms**




A A# A# B C C C D# E D# C# B G# G# F# C# A B C B C D# D D E F F F# F# E F# C B G# D# D# A B C B C D# C# E F F C# G# F# F F# F# E E E C E F F A B C B C C# D# C# D# F E F G F A# A

445.00 455.00 455.00 505.00 267.50 267.50 267.50 317.50 332.50 302.50 272.50 505.00 425.00 425.00 375.00 282.50 445.00 505.00 267.50 505.00 267.50 317.50 287.50 287.50 322.50 357.50 357.50 377.50 372.50 332.50 372.50 262.50 495.00 425.00 317.50 317.50 445.00 495.00 267.50 505.00 267.50 317.50 272.50 322.50 357.50 352.50 272.50 422.50 377.50 352.50 372.50 367.50 332.50 332.50 332.50 267.50 332.50 357.50 357.50 445.00 495.00 267.50 505.00 267.50 272.50 316.25 272.50 316.25 357.50 322.50 357.50 382.50 358.75 475.00 452.50

**Janela: 150 ms** **Sobreposição: 25 ms**




A A B C C D# E D B G# F# C# B C C D# C# D# F F# F# F# B G# D# A B C C D# C# F C# G# F# F# E F# E C F F A C B C# C# D# F F F# A# A
440.00 446.67 493.33 266.67 266.67 316.67 330.00 300.00 500.00 420.00
373.33 280.00 493.33 266.67 266.67 316.67 283.33 320.00 353.33 373.33
366.67 373.33 493.33 420.00 313.33 440.00 493.33 263.33 266.67 316.67
270.00 353.33 270.00 423.33 376.67 370.00 330.00 373.33 330.00 263.33
356.67 353.33 440.00 263.33 500.00 270.00 270.00 315.00 356.67 356.67
376.67 466.67 446.67