

Aluno: Jedson Jhones Barbosa Alves

Exercício 1 – Conceitos de filtros

Filtros digitais (estudados no curso) são sistemas LIT que permitem passar certos componentes espectrais de um sinal de entrada (banda de passo), enquanto atenuam outros componentes em frequência (banda de rejeição).

- a) Classifique os tipos de filtros segundo a duração da resposta ao impulso; explique o que significa o termo usado em cada caso;

Podem ser classificados em duas categorias básicas:

1:IIR. Sistemas com resposta ao impulso infinita (IIR, do inglês Infinite Impulse Response)

O termo IIR, é devido ser um filtro digital com resposta ao impulso de duração infinita.

2:FIR. Sistemas com resposta ao impulso finita (FIR, do inglês Finite Impulse Response)

O termo FIR, é devido ser um filtro digital que apresenta duração finita em resposta a um impulso

- b) Comente com as suas palavras duas diferenças entre os tipos de filtros mencionados no item a);

A maior diferença entre eles, os filtros FIR e IIR, é a resposta ao impulso, que é finita no caso de FIR e infinita no caso de IIR.

Outra diferença é que os filtros IIR permite obter uma magnitude da resposta em frequência desejada com um número de coeficientes menores do que no caso FIR

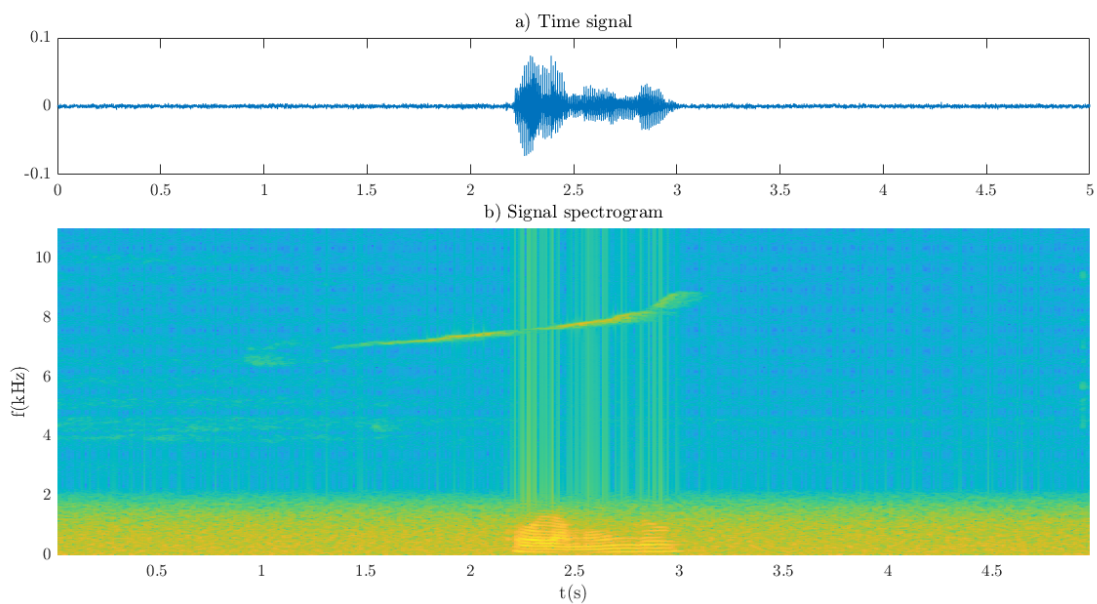
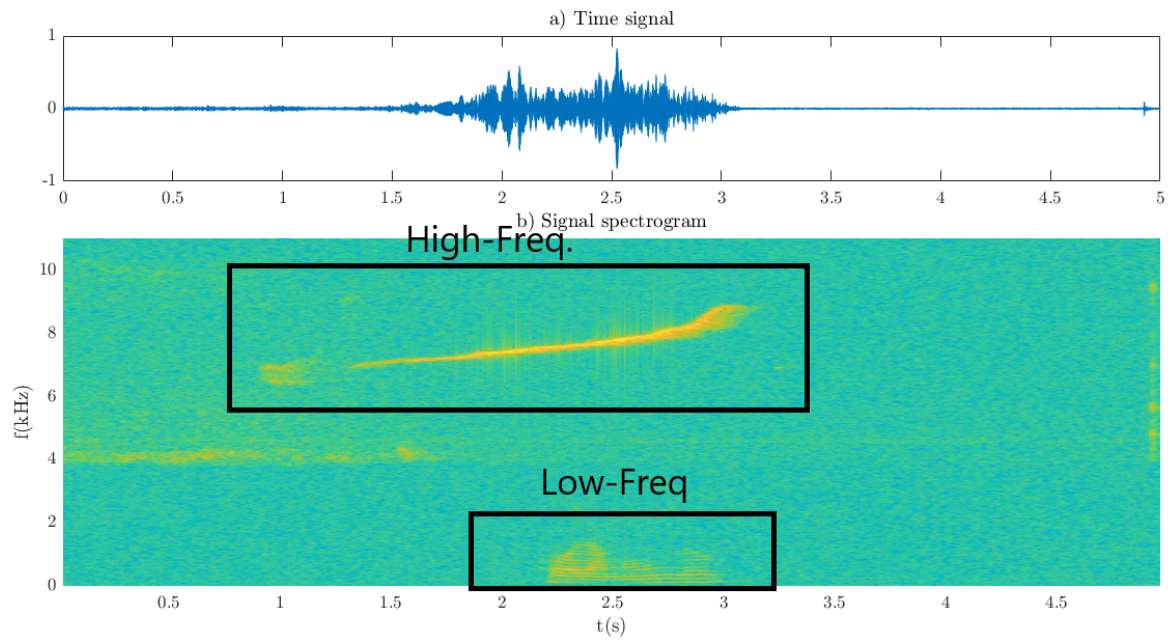
- c) Mencione pelo menos uma técnica de projetos de filtros para cada classe de filtro.

Para o caso do filtro IIR, pode ser projetado usando a Transformação Bilinear. E no caso do FIR, pode ser projetado usando métodos de janelamento.

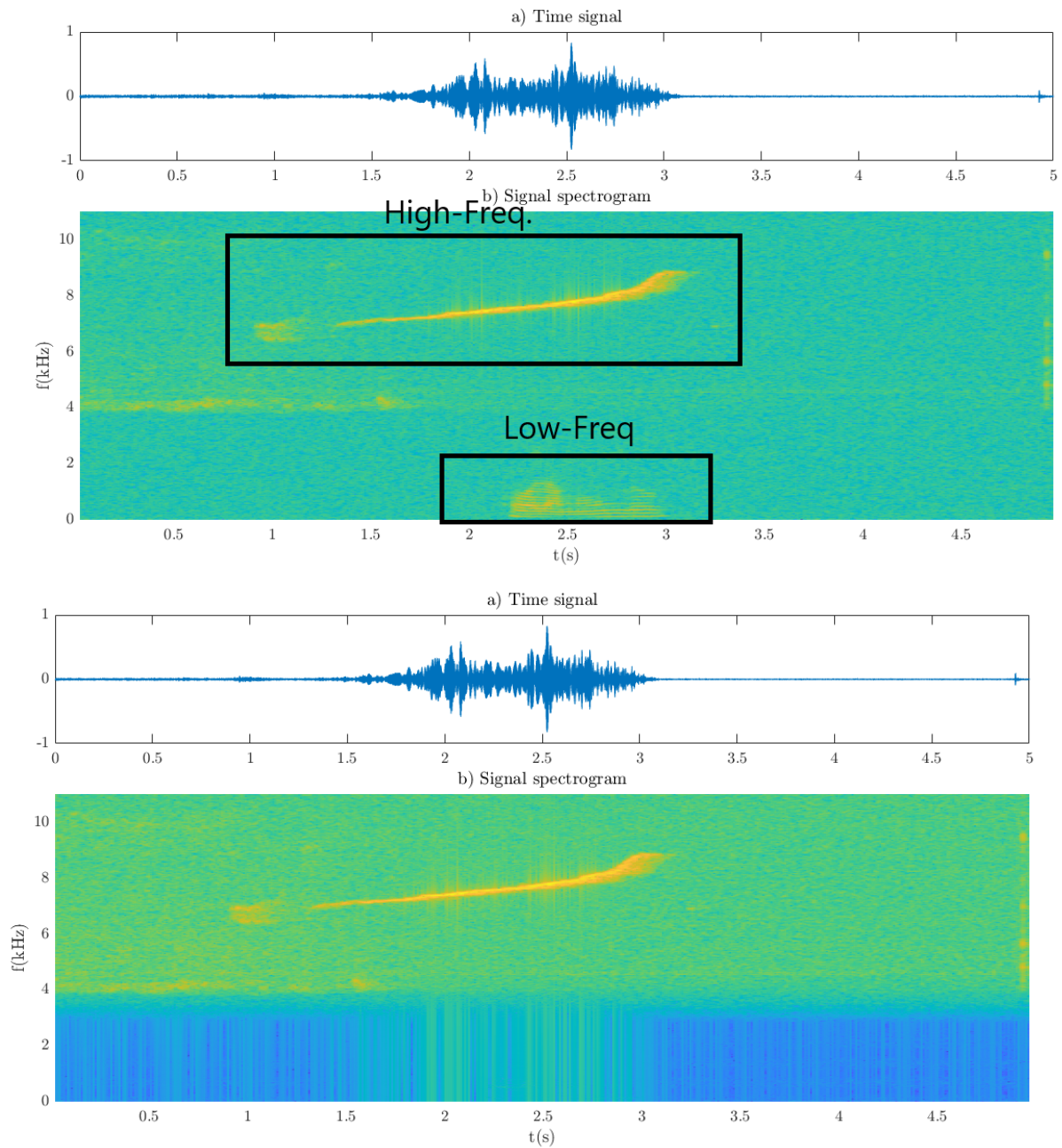
Exercício 2 – Filtragem usando filtro FIR

O sinal de áudio 'UnknownSound.wav' contém uma vocalização de sagui sobreposta a uma fala humana.

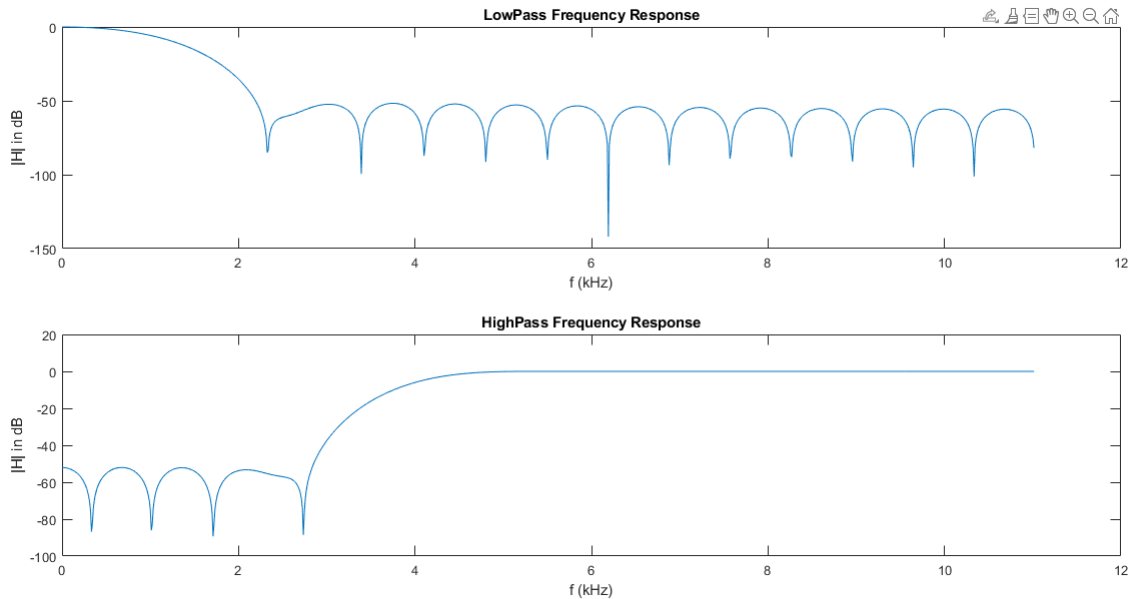
- a) Implementar um filtro FIR que permita recuperar o componente *Low-Freq* contido no sinal de áudio. Mostre uma figura similar à Figura 1, mas construída a partir do sinal recuperado (componente *Low-Freq*).



- b) Implementar um filtro FIR que permita recuperar o componente *High-Freq* contido no sinal de áudio. Mostre uma figura similar à Figura 1, mas construída a partir do sinal recuperado (componente *High-Freq*).



- c) Faça 2 gráficos na mesma figura e represente a resposta em frequência (valor absoluto em dB) dos filtros obtidos em a) e b). Discuta como a resposta em frequência obtida em cada caso se corresponde com o tipo da filtragem implementada.



Como sabemos Filtros FIR passa baixa, cortam as frequências acima da frequência de corte. E Filtros FIR passa alta, cortam os sinais abaixo da frequência de corte, permitindo somente as altas frequência. Que é exatamente o que está ocorrendo nos gráficos das respostas a frequência para LowPass e HighPass.

Onde o LowPass permite somente as baixas frequências, e o High somente as altas frequências.

Abaixo segue o código MATLAB para a questão 2

```
clearvars
close all
clc
```

```
% Abrindo o sinal de audio
FileName = 'UnknownSound.wav';
[x,fs] = audioread(FileName);
NSamples = length(x);
t = (0:NSamples-1)/fs;
```

```
% Plot do sinal e o spectrograma do sinal original
fig = figure('Position',[100 100 900 600],'color','w');
```

```
% Plot do sinal
ax1 = subplot(3,1,1);
plot(t,x);
tl = title('a) Time signal','Interpreter','latex');
set(ax1,'TickLabelInterpreter','latex','FontSize',13);
```

```
% Plot spectrogramaA
```

```
% pkg load signal; % % We need to uncommnet this in octave
nfft = 1024; % number fft points for spectrogram (STFT)
WLength = round(.05*fs); % Number of samples for each window segment
used in STFT
PercentOverlap = 0.8; % 80 percent overlap
```

```
ax2 = subplot(3,1,[2 3]);
[Pxx,tPxx,f] = My_STFT(x,fs,WLength,PercentOverlap);
Pxx_dB = 10*log10(Pxx);
surf(tPxx,f/1e3,Pxx_dB,'edgecolor','none');
axis tight, view(0,90)
```

```
tl = title('b) Signal spectrogram','Interpreter','latex');
xlabel('t(s)','Interpreter','latex');
ylabel('f(kHz)','Interpreter','latex');
set(ax2,'TickLabelInterpreter','latex','FontSize',13);
```

```
colorbar off
% print('-dpng','-r300', 'RawSignal.png');
```

```
%% Obtendo os coeficientes do filtro FII
%% Para Low-freq ((2)a))
```

```
fc = 1000; % fc = 1 kHz
wc = fc/(fs/2);
M = 31;
b = fir1(M,wc);
```

```
[H,fH] = freqz(b,1,1024,fs);
```

```
figure('color','w');
plot(fH/1000,20*log10(abs(H)));
xlabel('f (kHz)')
ylabel('|H| in dB')
title('LowPass Frequency Response')
```

```
% Plot do sinal de tempo e do spectrograma do sinal original
y = filter(b,1,x);
fig = figure('Position',[100 100 900 600],'color','w');
```

```
% Plot do sinal de tempo
ax1 = subplot(3,1,1);
plot(t,y);
```

```

tl = title('a) Time signal','Interpreter','latex');
set(ax1,'TickLabelInterpreter','latex','FontSize',13);

% Plot do spectrograma

nfft = 1024; % number fft points for spectrogram (STFT)
WLength = round(.05*fs); % Number of samples for each window segment
used in STFT
PercentOverlap = 0.8; % 80 percent overlap

ax2 = subplot(3,1,[2 3]);
[Pxx,tPxx,f] = My_STFT(y,fs,WLength,PercentOverlap);
Pxx_dB = 10*log10(Pxx);
surf(tPxx,f/1e3,Pxx_dB,'edgecolor','none');
axis tight, view(0,90)

tl = title('b) Signal spectrogram','Interpreter','latex');
xlabel('t(s)','Interpreter','latex');
ylabel('f(kHz)','Interpreter','latex');
set(ax2,'TickLabelInterpreter','latex','FontSize',13);

%% para o High-Freq ((2)B))
fc2 = 4000; % fc = 1 kHz
wn = fc2/(fs/2);
b2 = fir1(M,wn,'high')
[H2,fH2] = freqz(b2,1,1024,fs);
figure('color','w');
plot(fH2/1000,20*log10(abs(H2)));
xlabel('f (kHz)')
ylabel('|H| in dB')
title('HighPass Frequency Response')

% Plot time signal & spectrogram of original signal
y2 = filter(b2,1,x);
fig = figure('Position',[100 100 900 600],'color','w');

% Plot time signal
ax1 = subplot(3,1,1);
plot(t,y2);
tl = title('a) Time signal','Interpreter','latex');
set(ax1,'TickLabelInterpreter','latex','FontSize',13);

% Plot spectrogram
% pkg load signal; % % We need to uncommnet this in octave

```



```
nfft = 1024; % number fft points for spectrogram (STFT)
WLength = round(.05*fs); % Number of samples for each window segment
used in STFT
PercentOverlap = 0.8; % 80 percent overlap
```

```
ax2 = subplot(3,1,[2 3]);
[Pxx,tPxx,f] = My_STFT(y2,fs,WLength,PercentOverlap);
Pxx_dB = 10*log10(Pxx);
surf(tPxx,f/1e3,Pxx_dB,'edgecolor','none');
axis tight, view(0,90)
```

```
tl = title('b) Signal spectrogram','Interpreter','latex');
xlabel('t(s)','Interpreter','latex');
ylabel('f(kHz)','Interpreter','latex');
set(ax2,'TickLabelInterpreter','latex','FontSize',13);
```

```
%% Resposta a frequencia ((2)c)
figure('color','w');
ax1 = subplot(2,1,1);
plot(fH/1000,20*log10(abs(H)));
xlabel('f (kHz)')
ylabel('|H| in dB')
title('LowPass Frequency Response')
```

```
ax2 = subplot(2,1,2);
plot(fH2/1000,20*log10(abs(H2)));
xlabel('f (kHz)')
ylabel('|H| in dB')
title('HighPass Frequency Response')
```

Exercício 3 – Conceitos de Dizimação/Interpolação

Dizimação e **Interpolação** são duas técnicas muito úteis quando é necessário mudar a taxa de amostragem (f_s) por um fator inteiro (diminuir ou aumentar f_s respectivamente). Explique com as suas palavras:

- a) Diferença entre **Dizimação** e **Subamostragem**; quais efeitos indesejados poderiam acontecer ao subamostrar um sinal sem o devido pré-processamento?

De forma resumida, a dizimação possui um filtro passa baixa antes de fazer a subamostragem.

Subamostrar um sinal sem o devido pré-processamento pode acontecer o aliasing.

b) Como poderíamos mudar a taxa de amostragem por um fator não inteiro?

Utilizando a Re-amostragem, que é a combinação da dizimação + interpolação.

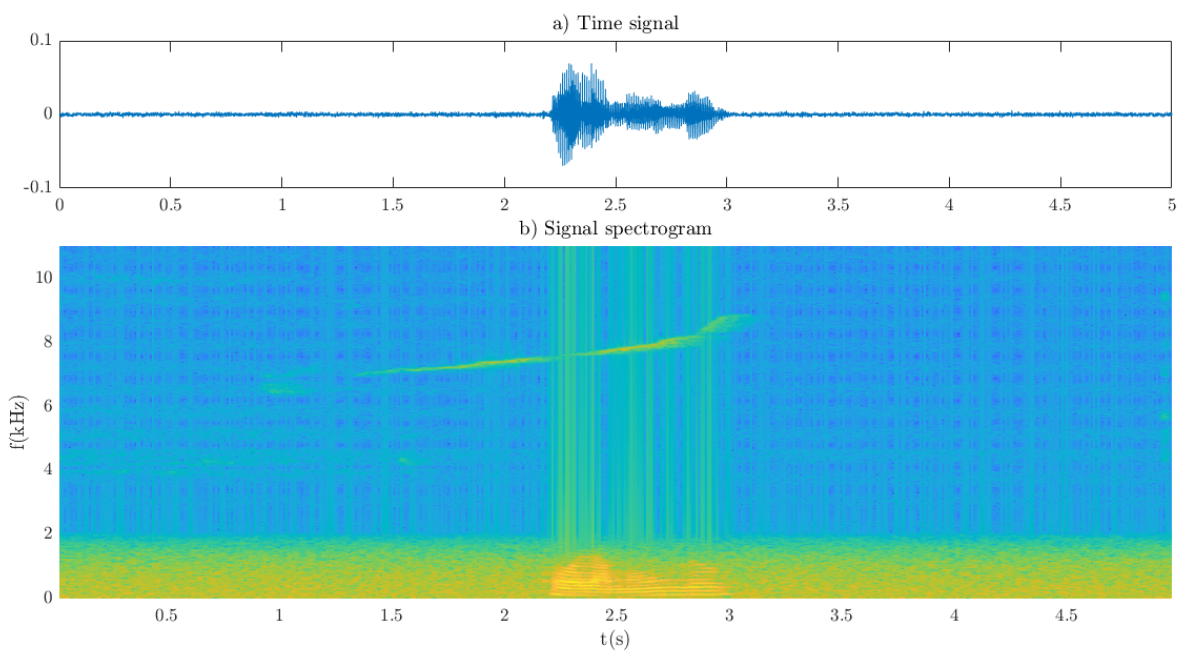
c) Os processos de **Dizimação** e **Interpolação** implementados na sua forma clássica não são eficientes. Mencione quais técnicas podem ser usadas para mudar a taxa de amostragem de forma eficiente.

Uma das técnicas que podem ser usadas para mudar a taxa de amostragem de forma eficiente são a decomposição polifásica.

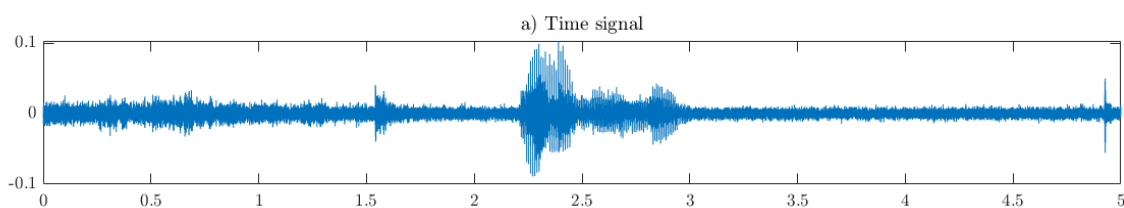
Exercício 4 – Mudança da frequência de amostragem

Dado o sinal de áudio apresentado no Exercício 2:

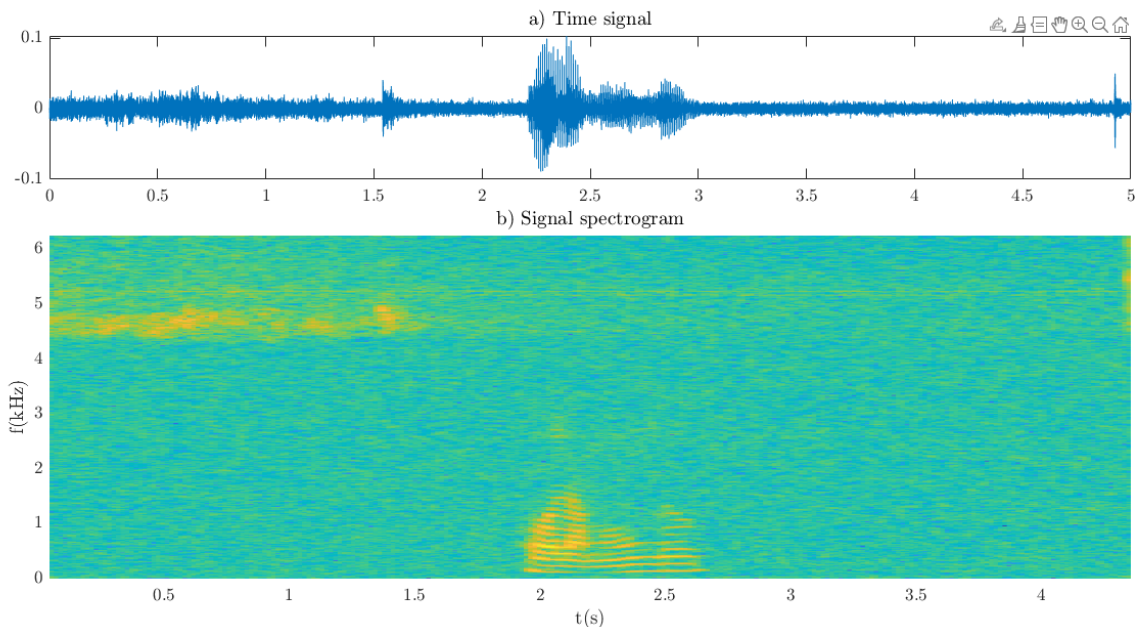
a) Determine um novo valor de frequência de amostragem que permita manter a informação do componente *Low-Freq*, mas que descarte o componente *High-Freq* do sinal analisado;



b) Implemente a mudança de taxa de amostragem projetada em a); obtenha o sinal correspondente a esta mudança de f_s ;



c) Mostre o resultado obtido em b) em uma figura similar à Figura 1.



Abaixo segue o código MATLAB para a questão 4.

```
clearvars
close all
clc
```

```
% Abrindo o sinal de audio
FileName = 'UnknownSound.wav';
[x,fs] = audioread(FileName);
```

```
NSamples = length(x);
t = (0:NSamples-1)/fs;
```

```
% Plot do sinal e o spectrograma do sinal original
fig = figure('Position',[100 100 900 600],'color','w');
```

```
% Plot do sinal
ax1 = subplot(3,1,1);
plot(t,x);
tl = title('a) Time signal','Interpreter','latex');
set(ax1,'TickLabelInterpreter','latex','FontSize',13);
```

```
% Plot spectrogramA
% pkg load signal; % % We need to uncommnet this in octave
```

```
nfft = 1024; % number fft points for spectrogram (STFT)
WLength = round(.05*fs); % Number of samples for each window segment
used in STFT
PercentOverlap = 0.8; % 80 percent overlap
```

```
ax2 = subplot(3,1,[2 3]);
[Pxx,tPxx,f] = My_STFT(x,fs,WLength,PercentOverlap);
Pxx_dB = 10*log10(Pxx);
surf(tPxx,f/1e3,Pxx_dB,'edgecolor','none');
axis tight, view(0,90)
```

```
tl = title('b) Signal spectrogram','Interpreter','latex');
xlabel('t(s)','Interpreter','latex');
ylabel('f(kHz)','Interpreter','latex');
set(ax2,'TickLabelInterpreter','latex','FontSize',13);
```

```
colorbar off
% print('-dpng','-r300', 'RawSignal.png');
```

```
%% Obtendo os coeficientes do filtro FII
%% ((4) a)
```

```
fc = 1000; % fc = 1 kHz
wc = fc/(25000/2);
M = 31;
b = fir1(M,wc);
```

```
[H,fH] = freqz(b,1,1024,fs);
```

```
% Plot do sinal de tempo e do spectrograma do sinal original
y = filter(b,1,x);
fig = figure('Position',[100 100 900 600],'color','w');
```

```
% Plot do sinal de tempo
ax1 = subplot(3,1,1);
plot(t,y);
tl = title('a) Time signal','Interpreter','latex');
set(ax1,'TickLabelInterpreter','latex','FontSize',13);
```

```
% Plot do spectrograma
```

```
nfft = 1024; % number fft points for spectrogram (STFT)
```

```
WLength = round(.05*fs); % Number of samples for each window segment  
used in STFT
```

```
PercentOverlap = 0.8; % 80 percent overlap
```

```
ax2 = subplot(3,1,[2 3]);  
[Pxx,tPxx,f] = My_STFT(y,fs,WLength,PercentOverlap);  
Pxx_dB = 10*log10(Pxx);  
surf(tPxx,f/1e3,Pxx_dB,'edgecolor','none');  
axis tight, view(0,90)
```

```
tl = title('b) Signal spectrogram','Interpreter','latex');  
xlabel('t(s)','Interpreter','latex');  
ylabel('f(kHz)','Interpreter','latex');  
set(ax2,'TickLabelInterpreter','latex','FontSize',13);
```

```
%% ((4) b e c) mudança na taxa de amostragem
```

```
r = 2;
```

```
fs1 = 25000/r;
```

```
t_dwn = t(1:r:end);
```

```
y_dwn = decimate(x,r,M,'fir'); % uses an FIR filter of order n.
```

```
% Plot time signal & spectrogram of original signal
```

```
fig = figure('Position',[100 100 900 600],'color','w');
```

```
% Plot time signal
```

```
ax1 = subplot(3,1,1);
```

```
plot(t_dwn,y_dwn);
```

```
tl = title('a) Time signal','Interpreter','latex');
```

```
set(ax1,'TickLabelInterpreter','latex','FontSize',13);
```

```
% Plot spectrogram
```

```
% pkg load signal; % % We need to uncommnet this in octave
```

```
nfft = 1024; % number fft points for spectrogram (STFT)
```

```
WLength = round(.05*fs); % Number of samples for each window segment  
used in STFT
```

```
PercentOverlap = 0.8; % 80 percent overlap
```

```
ax2 = subplot(3,1,[2 3]);  
[Pxx,tPxx,f] = My_STFT(y_dwn,fs1,WLength,PercentOverlap);  
Pxx_dB = 10*log10(Pxx);  
surf(tPxx,f/1e3,Pxx_dB,'edgecolor','none');  
axis tight, view(0,90)
```

```
tl = title('b) Signal spectrogram','Interpreter','latex');  
xlabel('t(s)','Interpreter','latex');  
ylabel('f(kHz)','Interpreter','latex');  
set(ax2,'TickLabelInterpreter','latex','FontSize',13);
```