

Multimédia e Novos Serviços

Lab 1

João Seixas - up201505648

José Pedro Borges - up201503603

Vicente Espinha - up201503764

**Projeto realizado no âmbito do
Mestrado Integrado em Engenharia Informática e
Computação**

26/02/2018

1ª Parte

Na primeira parte do trabalho é pedido que selecionemos um ficheiro de música mp3 e, utilizando o VLC, se converta o mesmo para dois ficheiros de formato WAV. Os ficheiros gerados deverão ter uma taxa de amostragem (Samplerate) de 11025 Hz e de 44100 Hz respetivamente. Estes devem ser reproduzidos, para se poder comparar a qualidade de cada um e notar as diferenças entre si.

No som gravado com Samplerate de 11025 Hz nota-se uma baixa qualidade, com um maior registo de notas graves e distorção de áudio. Isto é devido a um efeito de aliasing no som, causado por ter um Samplerate inferior a duas vezes a sua frequência máxima (teorema de Nyquist).

No som com Samplerate de 44100 Hz não existem diferenças audíveis relativamente ao som original. Isto porque a maioria dos ficheiros MP3 têm este Samplerate, e o que nós utilizamos não foi exceção.

2ª Parte

Nesta segunda parte do trabalho, foi pedido que se verificasse as alterações que ocorrem a ficheiros de som, quando sujeitos a sub-amostragem e interpolação, através de um script para o Matlab. Para isso, importou-se um ficheiro original para o Matlab ("bugsbunny1.wav"), e, de seguida, executou-se uma sub-amostragem (down sampling) do ficheiro original de som, em que se escutou o resultado obtido. Depois, foi executada uma interpolação a partir da sub-amostragem do ficheiro original de som, em que se escutou e gravou o resultado dessa operação. Por fim, calculou-se o erro quadrático médio obtido entre o som interpolado e o som original. Esta experiência foi realizada com uma sub-amostragem de 4 para 1 e de 2 para 1, e, com e sem filtro.

O som utilizado possui as seguintes características:

Filename: 'bugsbunny1.wav'

Compression Method: 'Uncompressed'

NumChannels: 1

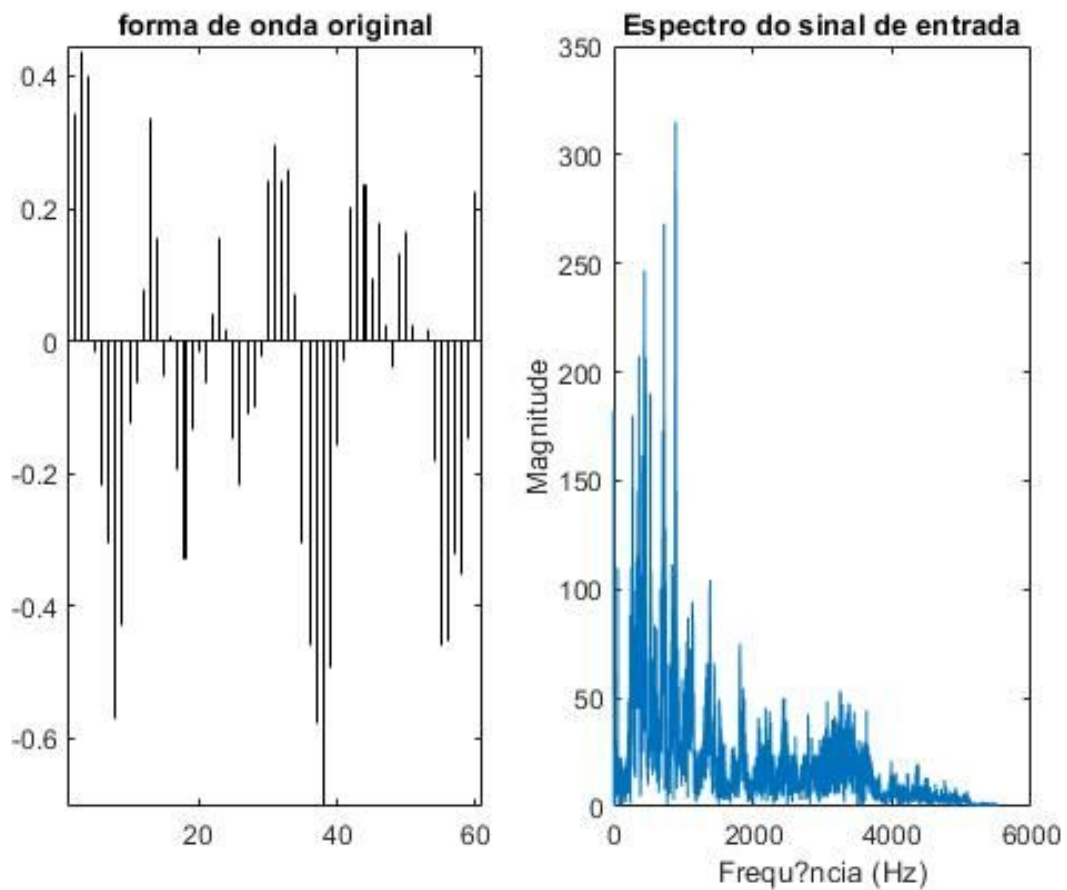
SampleRate: 11025

Total Samples: 47315

Duration: 4.2916

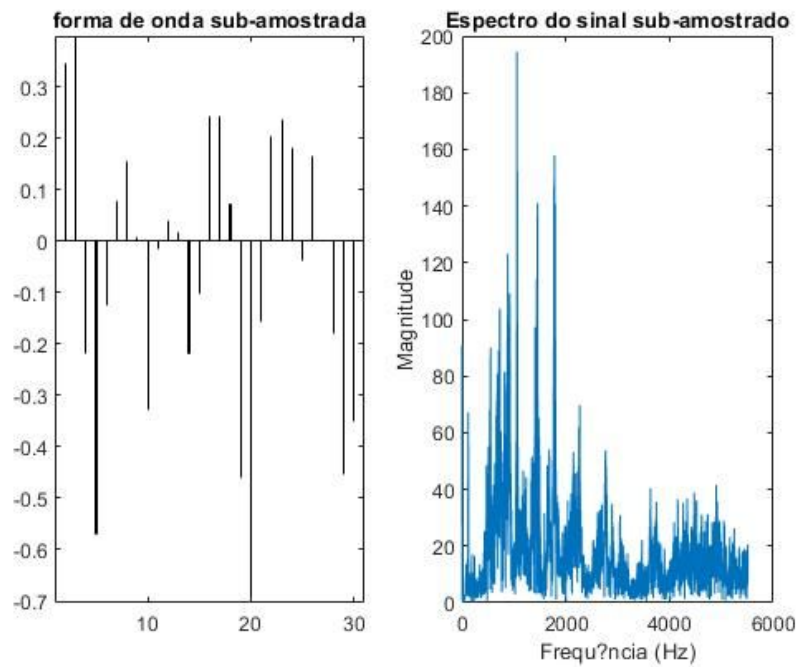
Bits Per Sample: 8

Uma sub-amostragem com um fator i significa que por cada i amostras na versão do som inicial apenas será mantida uma na nova versão. Para que a sub-amostragem seja feita de maneira a que não se introduza o efeito indesejável de aliasing, é necessário que o sinal não contenha componentes acima do limite da frequência de Nyquist. Para isto, antes de amostrar um sinal, utiliza-se um filtro anti-aliasing passa baixo para remover as frequências acima do limite.

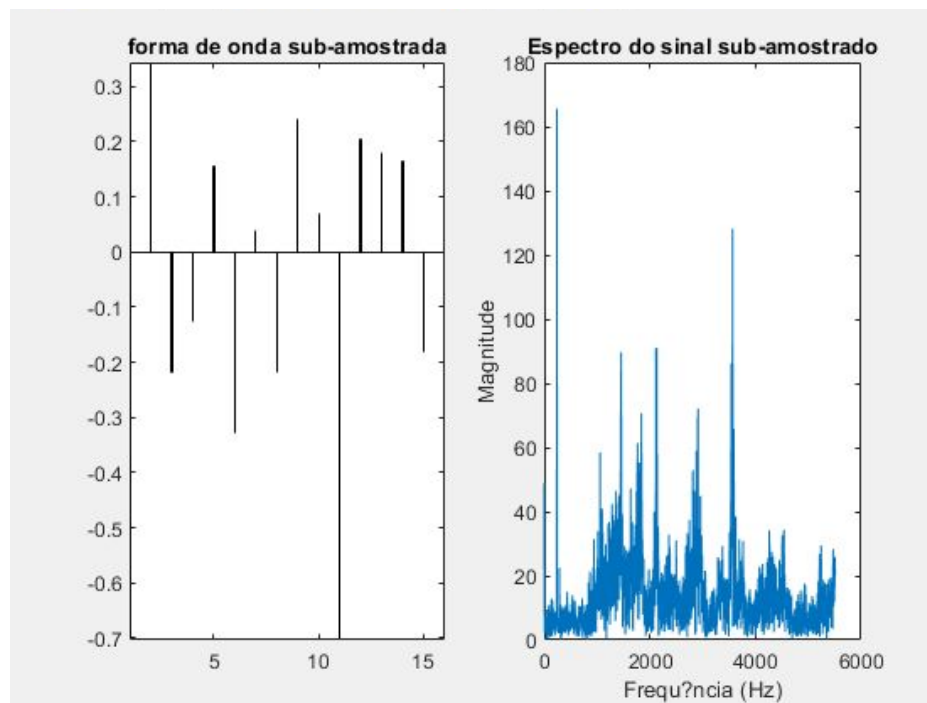


Som original

Sub-Amostragem



Som sub amostrado de Fator 2

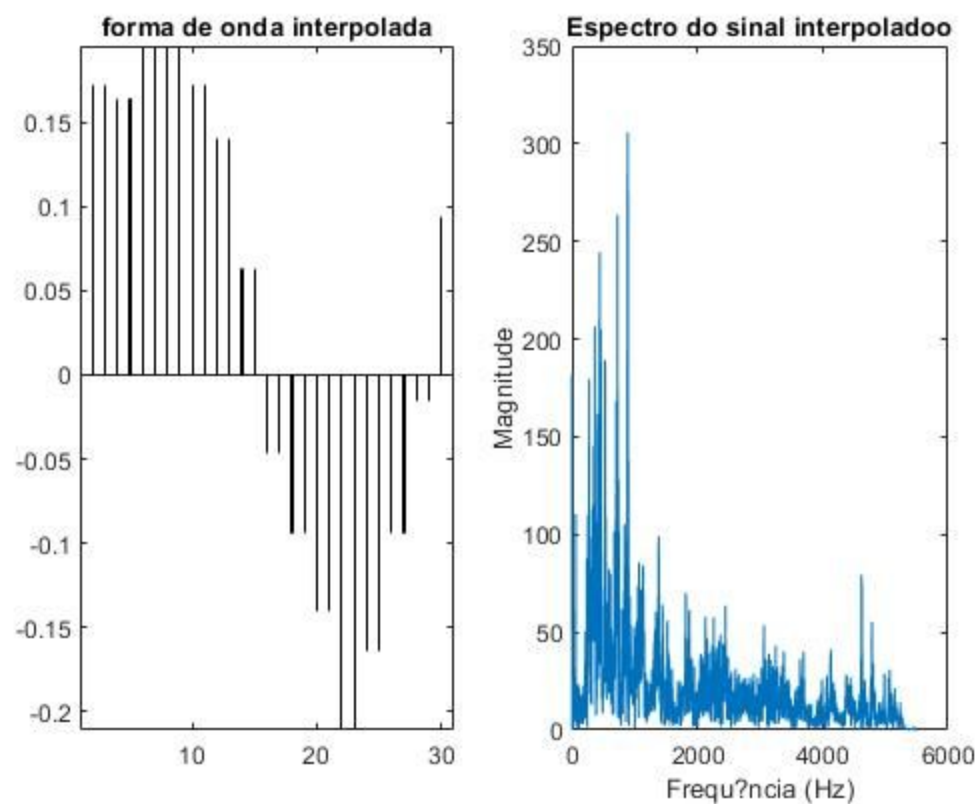


Som sub-amostrado de Fator 4

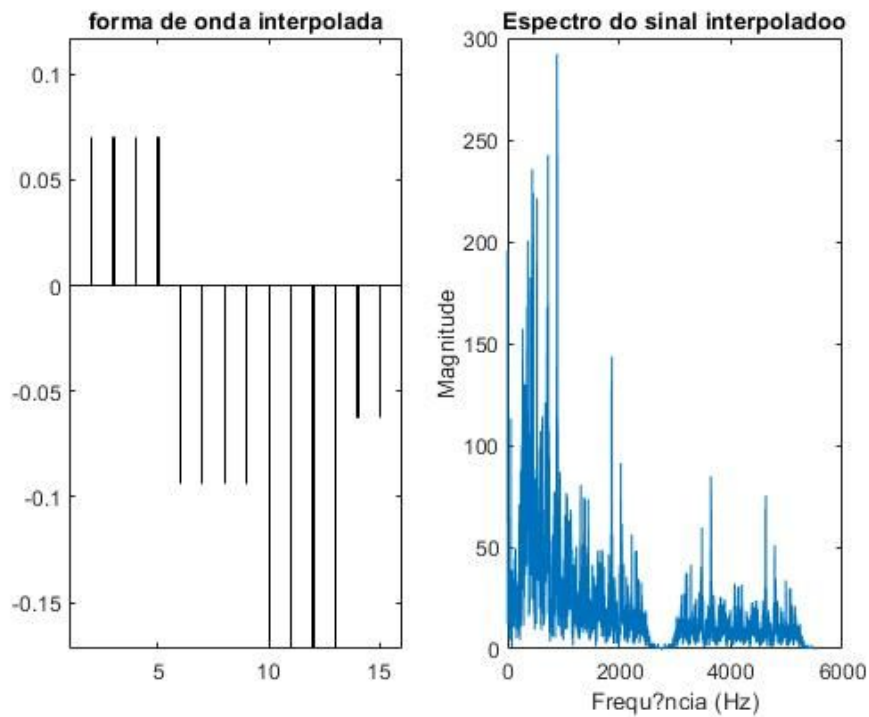
A sub-amostragem foi feita sem filtros e deu origem aos gráficos acima. A amostra de fator 4 tem metade das amostras de fator 2. É também possível observar que foram introduzidas frequências acima das do som original, dando origem a um efeito de aliasing que distorce o som. Isto é devido à não utilização de um filtro anti-aliasing, que iria cortar as frequências não desejadas.

Foi também realizado uma interpolação.

Interpolação



Som Interpolado de Fator 2



Som Interpolado de Fator 4

A Interpolação foi também realizada sem recurso a filtros, levando assim a frequências elevadas que previamente não existiam, tal como no caso da sub-amostragem. Pode-se, então, concluir que a utilização do filtro irá melhorar a qualidade do som gerado, graças à remoção de aliasing.

Assim, realizou-se a geração de sons utilizando filtros na expectativa de que a sua qualidade fosse melhorada.

Na página em baixo são apresentados os comandos utilizados e outputs obtidos na geração dos resultados acima obtidos.

```
>> amostragemInterp_semFiltro('bugsbunny1.wav','bugsbunny1Saida2.wav',2)

Importar o som original

Carregue numa tecla para continuar

O som sub-amostrado

Carregue numa tecla para continuar

O som interpolado

Carregue numa tecla para continuar

Erro entre o sinal original e o interpolado = 0.0056295

PSNR do sinal interpolado = 19.5319
```

Comandos utilizados em Filtro 2

```
>> amostragemInterp_semFiltro('bugsbunny1.wav','bugsbunny1Saida.wav',4)

Importar o som original

Carregue numa tecla para continuar

O som sub-amostrado

Carregue numa tecla para continuar

O som interpolado

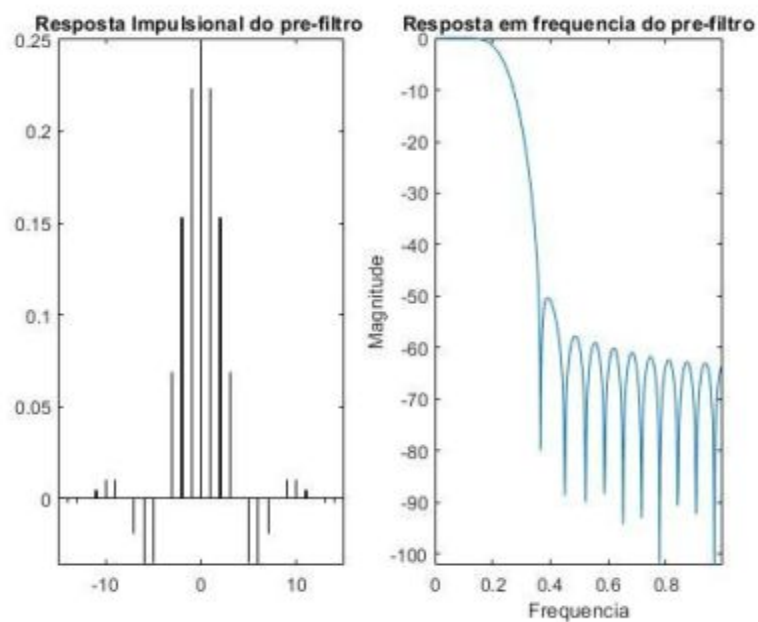
Carregue numa tecla para continuar

Erro entre o sinal original e o interpolado = 0.0186861

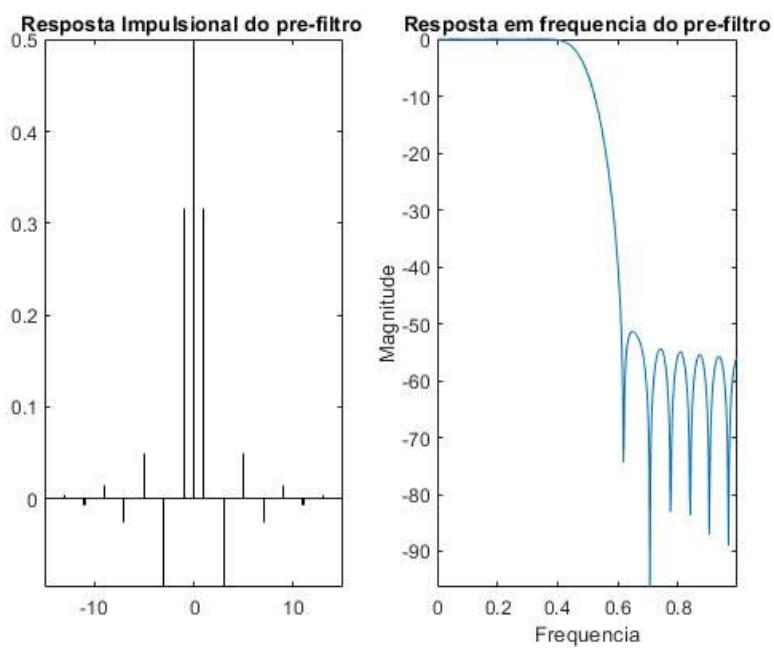
PSNR do sinal interpolado = 14.3214
```

Comandos utilizados em Filtro 4

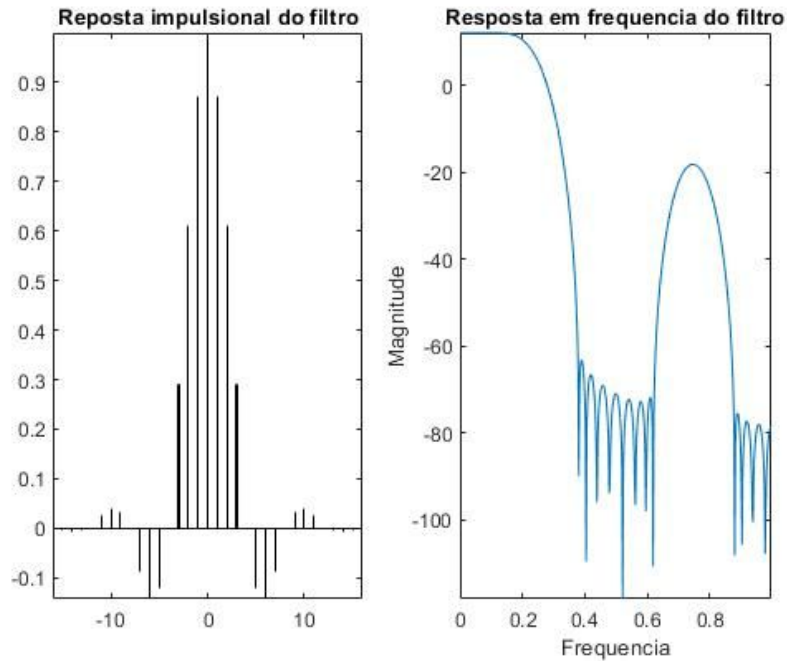
Utilização de Filtros



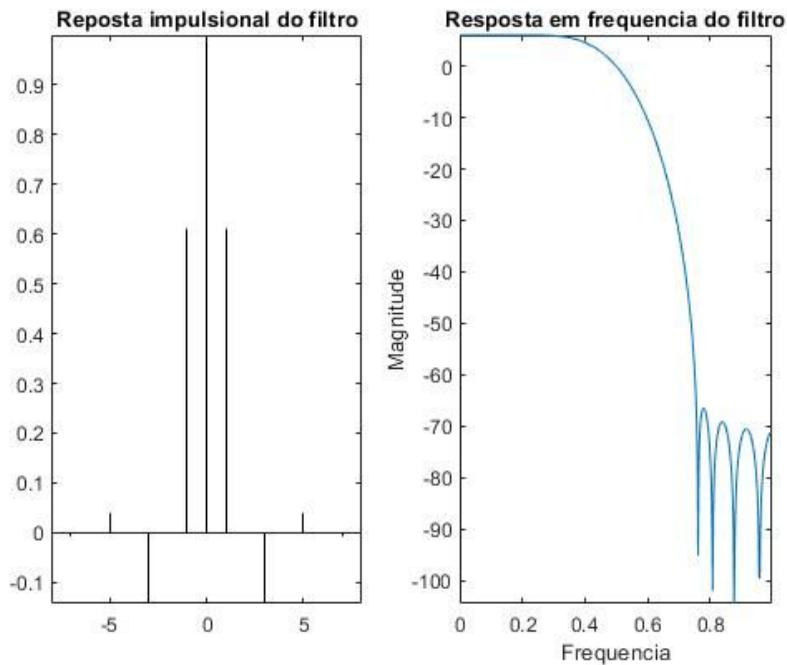
Pré-filtro de amostragem de fator 4



Pré-filtro de amostragem de fator 2



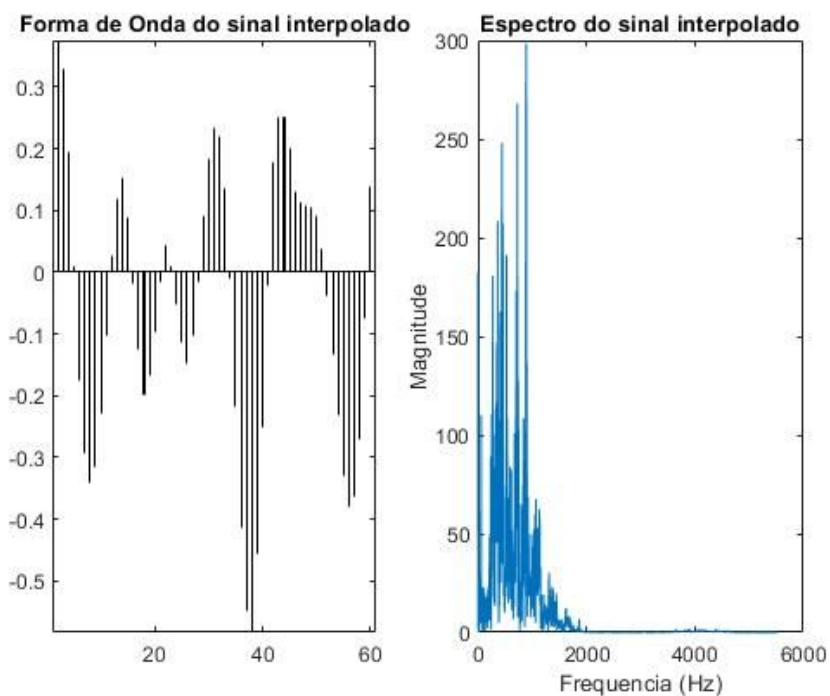
Filtro de fator de sub-amostragem 4



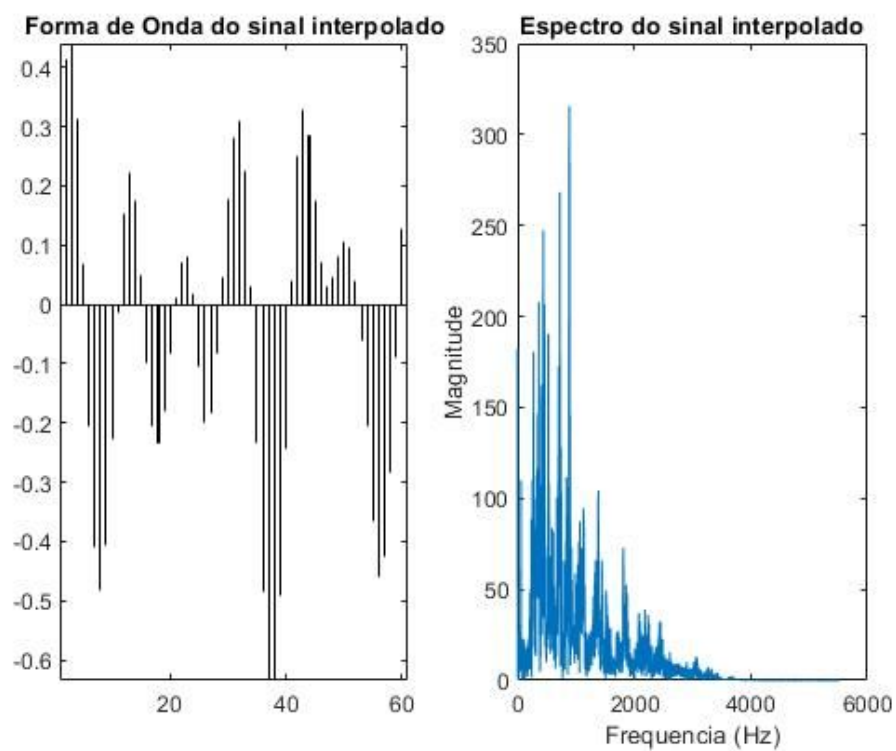
Filtro de fator de sub-amostragem 2

O filtro começa a cortar as frequências com magnitudes de, aproximadamente, -5 dB, que é a altura em que a curva começa a descair. O pré filtro de sub-amostragem 2 começa a cortar frequências acima dos 4000Hz e o pré filtro de sub-amostragem 4 começa a cortar a partir dos 2000Hz. Estes valores são aproximados.

Seguem-se os gráficos das interpolações com filtros.



Som interpolado de fator 4 com filtro



Som interpolado de fator 2 com filtro

Com base nos gráficos acima, pode-se verificar que as frequências mencionadas foram de facto cortadas do som gerado, contribuindo assim para a qualidade deste.

Segue-se um *print* da série de comandos e *outputs* realizados para gerar o resultado.

```
>> amostragemInterp_comFiltro('bugsbunny1.wav','bugsbunny1SaidaComFiltro4.wav',4)

Importar o sinal original para a variavel y

frequencia de amostragem = 11025 ; numero de amostras = 47312
Numero original de amostras no sinal de entrada = 47315

Prima uma tecla para continuar

O som sub-amostrado

Prima uma tecla para continuar

O pre-filtro utilizado

Prima uma tecla para continuar

O som interpolado

Prima uma tecla para continuar

O filtro utilizado na interpolacao

Prima uma tecla para continuar

Erro entre o sinal original e o interpolado = 0.00488846

PSNR do sinal interpolado = 20.1449
```

Resultados obtidos com fator 4 com filtro

```

>> amostragemInterp_comFiltro('bugsbunny1.wav','bugsbunny1SaidaComFiltro2.wav',2)

  Importar o sinal original para a variavel y

frequencia de amostragem = 11025 ; numero de amostras = 47314
Numero original de amostras no sinal de entrada = 47315

  Prima uma tecla para continuar

  O som sub-amostrado

  Prima uma tecla para continuar

  O pre-filtro utilizado

  Prima uma tecla para continuar

  O som interpolado

  Prima uma tecla para continuar

  O filtro utilizado na interpolacao

  Prima uma tecla para continuar

Erro entre o sinal original e o interpolado = 0.00204601

PSNR do sinal interpolado = 23.9276

```

Resultados obtidos com fator 2 com filtro

Foi também analisado o erro quadrático médio entre o sinal original e produzido e a PNSR.

ERRO

Fator de Amostragem	Com Filtro	Sem Filtro
2	0.00204601	0.0056295
4	0.00488846	0.0186861

PSNR

Fator de Amostragem	Com Filtro	Sem Filtro
2	23.9276	19.5319
4	20.1449	14.3214

Um valor reduzido de PSNR e um valor de Erro elevado é uma indicação subjetiva de má qualidade. Os valores obtidos vão de acordo com isto e com a análise feita anteriormente. Comparando os valores do fator de amostragem 2 com os de fator de amostragem 4, verifica-se que os de erro são menores e os de PSNR são maiores. Isto é devido ao facto de a sub-amostragem de fator 4 possuir menos amostras, contribuindo para uma menor qualidade de som. Comparando valores sem filtro com os com filtro, verifica-se também que os sem filtro têm maior valor de erro e um valor menor de PSNR.

Relativamente a uma análise mais subjetiva dos sons gerados, o grupo concorda que o som de sub-amostragem com filtro e fator 2 é o que melhor qualidade tem e que o sem filtro de sub-amostragem de fator 4 é o que pior qualidade tem.

3ª Parte

Nesta terceira parte, foi utilizado o programa “quant_uniform.m”, que permite a quantização de um sinal de áudio utilizando um número variável de níveis de quantização. Como sinal de entrada do programa, usou-se um som não comprimido com o formato WAV e, com pelo menos 16 bits de amostra e 44,1 kHz de frequência de amostragem. Para esta experiência, utilizou-se 256 e 16 níveis de quantização.

O som utilizado possui as seguintes características:

Filename: 'bugsbunnyexit_3.wav'

Compression Method: 'Uncompressed'

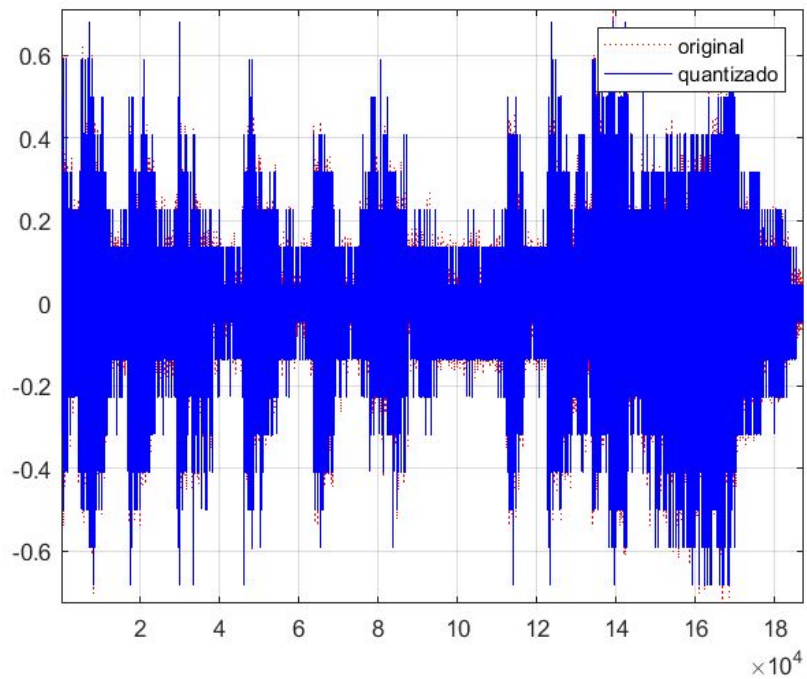
NumChannels: 1

SampleRate: 44100

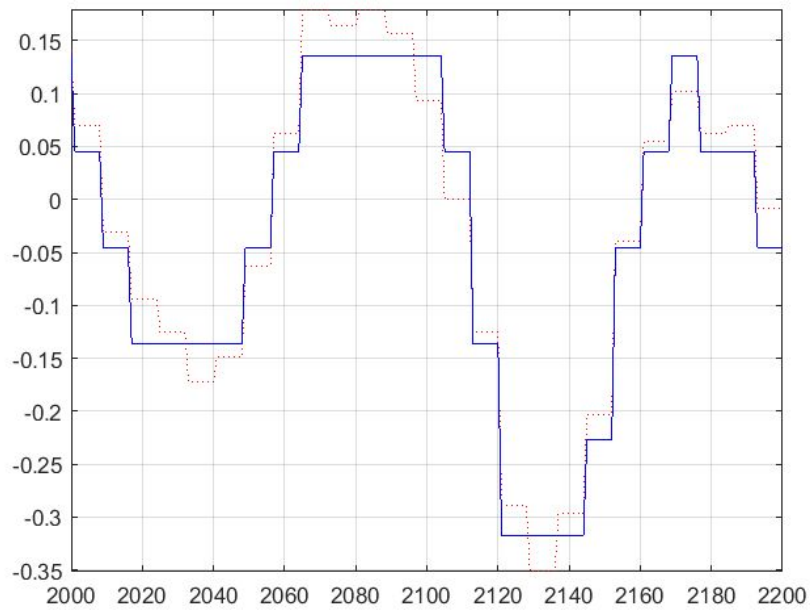
Total Samples: 187340

Duration: 4.2481

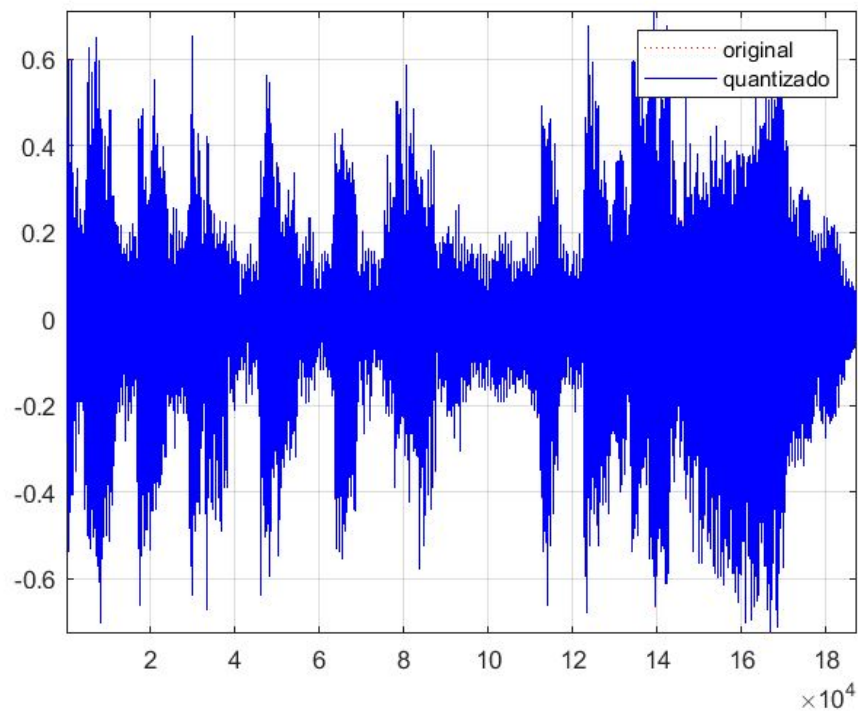
Bits Per Sample: 16



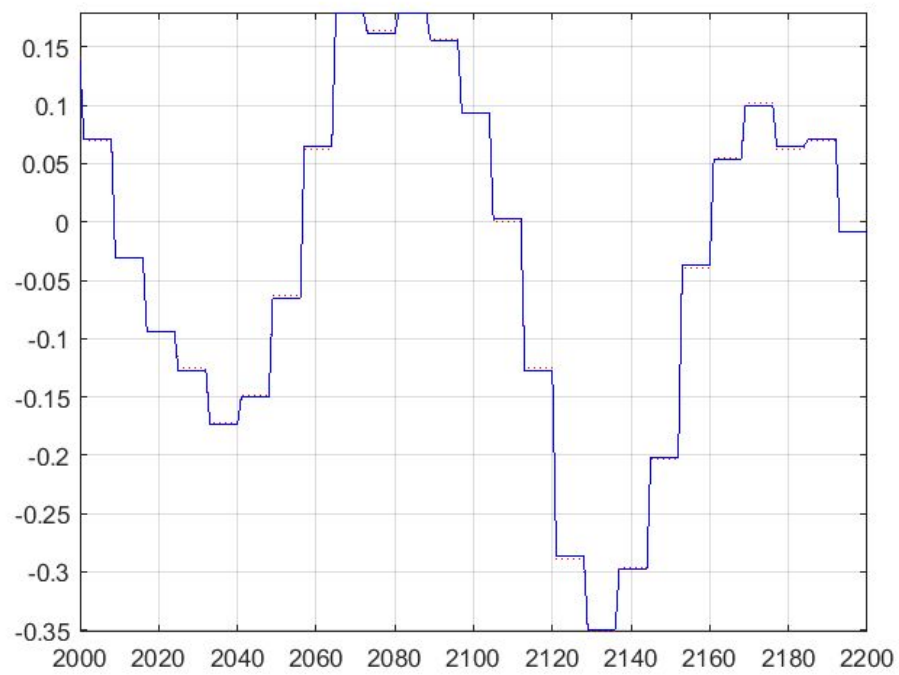
Programa utilizando 16 níveis de quantização (ao longo de todo o período)



Programa utilizando 16 níveis de quantização (num período em que $t \in [2000, 2200]$)



Programa utilizando 256 níveis de quantização (ao longo de todo o período)



Programa utilizando 256 níveis de quantização (num período em que $t \in [2000, 2200]$)

Resultados obtidos com 256 níveis de quantização

```
>> quant_uniform('bugsbunnyexit_3.wav','bugsbunnyexit_256.wav',256)

xmin =

    -0.7266
|
xmin,xmax,N,Q
    -0.7266    0.7266  256.0000    0.0057

Prima uma tecla para continuar
Erro entre o sinal original e o interpolado = 2.69943e-06

PSNR do sinal interpolado = 24.08
```

Resultados obtidos com 16 níveis de quantização

```
>> quant_uniform('bugsbunnyexit_3.wav','bugsbunnyexit_16.wav',16)

xmin =

    -0.7266

xmin,xmax,N,Q
    -0.7266    0.7266   16.0000    0.0908

Prima uma tecla para continuar
Erro entre o sinal original e o interpolado = 0.000684527

PSNR do sinal interpolado = -12.0024
```


Como podemos verificar nos gráficos entre 256 bits e 16 bits, quanto mais baixo o nível de quantização, maior será a diferença para o som original, uma vez que usa menos valores de amplitudes. O resultado obtido com o nível de quantização de 256 bits é bastante semelhante com som original, tendo um erro muito pequeno, ao passo que no de 16 bits já é perceptível a diferença para o som original, tendo um erro de quase 254 vezes maior que o erro do 256 bits, notando-se que não tem a mesma qualidade, contendo mais ruído.