Mestrado Integrado em Engenharia Eletrónica Industrial e Computadores

2018/2019

**Relatório do Primeiro Trabalho**

**de**

**Processamento Digital de Sinal**

Método de Filtragem: Janela de Kaiser

Sérgio Baixo - A76513

Índice

Introdução**3**

Procedimentos**4**

Janela de Kaiser**6**

Downsampling(decimação)**7**

Resultados Obtidos**9**

Conclusão**10**

Introdução

O objetivo da realização deste trabalho seria implementar um conjunto de funções em Matlab que permitissem aplicar a um sinal sonoro, amostrado a 8kHz, um conjunto de algoritmos que permitam diminuir a frequência de amostragem(fs) evitando assim, a ocorrência do fenómeno de aliasing, gerado pela aplicação de um filtro, neste caso de um filtro passa-baixo com a janela de Kaiser aplicada.

A realização do downsampling permite-nos, para além de eliminar a ocorrência de aliasing, obter um sinal mais compactado, ocupando menos espaço em memória e contribuindo para um número inferior de cálculos em processamento de sinal.

As funções de Matlab utilizadas foram a ‘audiorecording’, ‘getaudiodata’ e ‘sound’ para gravar o som a partir do microfone, armazenar os dados do mesmo sinal sonoro numa variável e reproduzir o mesmo, respetivamente. ‘kaiserord’, ‘fir1’ e ‘filter’ para calcular os parâmetros da janela de kaiser, criar o filtro e filtrar o sinal sonoro a partir do filtro criado, respetivamente. Todas estas funções serão explicadas detalhadamente ao longo deste relatório.

Procedimentos

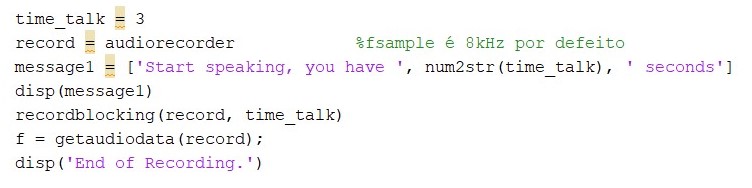
Primeiramente é definida uma variável ‘time\_talk’ que corresponde ao número de segundos que o utilizador dispõe para gravar a voz. É mostrada uma mensagem que se destina a informar o utilizador do tempo que dispõe para gravar a sua voz. Terminado esse tempo, os dados da gravação são armazenados numa variável como se fosse um array numérico. Por defeito a frequência de amostragem da gravação é de 8kHz.

Figura 1- Gravação de áudio

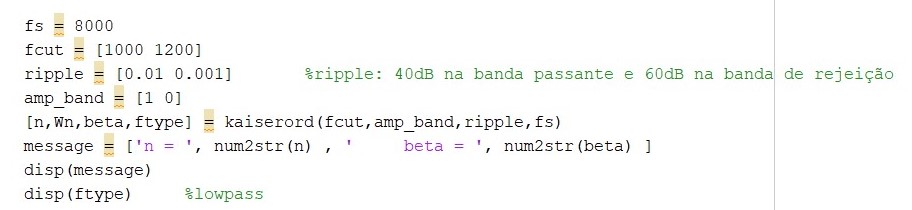
Segue-se a parte da elaboração do filtro. Começou-se por definir as variáveis de entrada da função ‘kaiserord’ associada ao método da janela de Kaiser. Estas variáveis são a frequência de sampling, um array com a banda passante e a banda de rejeição, que corresponde á banda passante mais 20%. Um array com a amplitude da banda, outro array com os ripples da banda passante(40dB) e da banda de rejeição(60dB) e a frequência de sampling(8kHz). Esta função devolve como parâmetros de saída os parâmetros da janela de Kaiser, são eles a ordem do filtro ,n, a frequência normalizada, wn e o fator de shape, beta, para além de devolver o tipo de filtro ftype, que neste caso se tratará de um filtro low-pass. É imprimido na consola a ordem do filtro e o tipo de filtro bem como o valor de beta através da instrução ‘disp’.

Figura 2-Janela de Kaiser

O segmento que se segue consiste no design do filtro através da função ‘fir1’. Esta função tem como parâmetros de entrada os parâmetros de saída que a função ‘kaiserord’ devolve, bem como o tipo de janela que queremos utilizar, que neste caso seria a keyword ‘kaiser’.

A função devolve-nos um conjunto de valores numéricos que representam os valores do gráfico do filtro. Ora, de forma a podermos observar o gráfico proveniente do design do filtro, optei por utilizar a instrução ‘wvtool’ para conseguir visualizar a resposta do filtro tanto no domínio da frequência como no domínio do tempo. Aliada a esta última foi também utilizada a instrução ‘freqz’ apenas para que fosse possível observar a resposta do filtro em termos de fase.  
O passo seguinte será a filtragem do sinal sonoro pelo filtro desenvolvido. Para lograr isto usa-se a instrução ‘filter’ que tem como parâmetros de entrada o sinal gravado e o filtro criado.

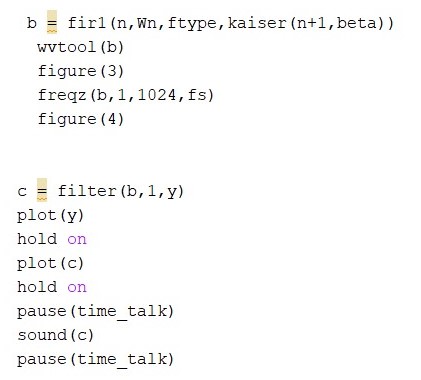


Figura 3- Design do filtro e aplicação do mesmo ao sinal sonoro

Por último fez-se o downsampling do resultado da saída do filtro recorrendo á função ‘downsample’ dentro de um ciclo ‘for’. Esta função irá dividir a frequência de amostragem do sinal pelo fator de decimação(n), fator esse que é igual ao intervalo de valores definidos no ciclo ‘for’. Ou seja, começando com um fator de decimação de 2 numa primeira iteração, a função irá dividir o atual sample rate(fs) por 2, e teremos assim um sinal decimado com metade do sample rate que neste caso passará a ser 4kHz. O ciclo é executado para n=2, n=3, n=4, obtendo três sample rates diferentes e como tal, três gráficos diferentes todos no mesmo ‘plot’, permitindo assim a possibilidade de comparar os diferentes resultados para os diferentes fatores de decimação. O sinal é então reproduzido através do comando ‘sound’ no final de cada iteração, podendo-se ouvir as diversas alterações que a decimação provocou no sinal filtrado.   
Podemos concluir, com isto, que o downsampling ou decimação impede a ocorrência de aliasing e também possibilita a compactação do sinal, reduzindo o número de amostras do nosso sinal.

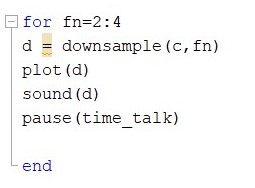


Figura 4- Downsample

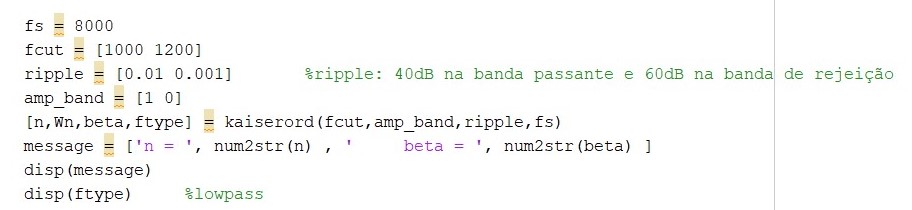
Janela de Kaiser

Figura 5-Janela de Kaiser

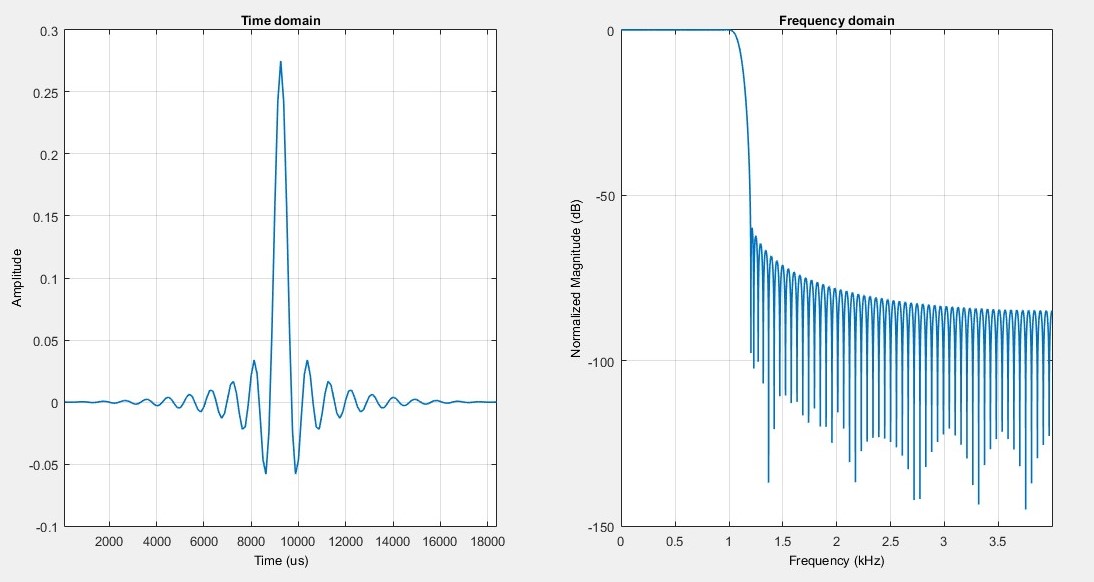


Figura 6-Janela de Kaiser no domínio do tempo e da frequência

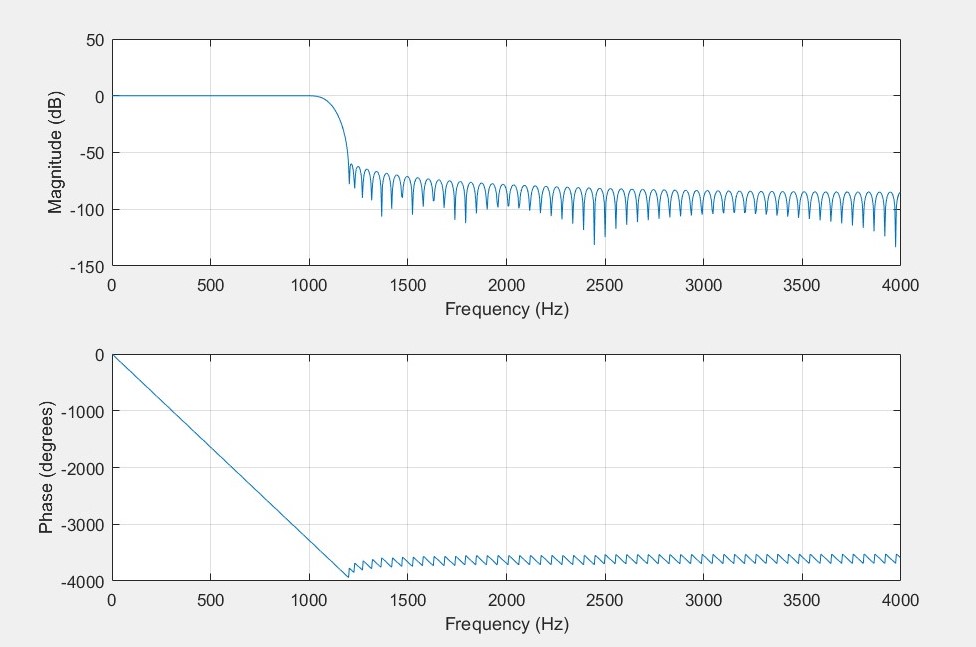


Figura 7-Janela de Kaiser em magnitude e em fase

Downsampling(decimação)

A decimação consiste na operação correspondente á redução da frequência de amostragem do sinal contínuo. Na realização desta operação, a única consequência é a possível ocorrência de aliasing. A mesma trata-se de uma operação reversível, ou seja, pode ser realizada sem quaisquer perda de informação, com a vantagem de compactação de sinal.

Esta redução da frequência de sample mencionada acima é feita dividindo a frequência normalizada por um fator inteiro M. Á medida que se aumenta este fator, a frequência de sample vai diminuir e vamos obter um sinal cada vez mais compactado.

De modo a evitar a ocorrência de aliasing, no processo de decimação, devemos garantir que wn < π/M. Se esta condição não se verificar, irá ocorrer aliasing, no entanto, em certas aplicações, pode ser desprezado. Em outros casos, o downsampling pode ser feito sem aliasing se for reduzida a largura de banda do sinal antes de fazer downsampling

A decimação deve ser executada após o filtro digital, de modo a eliminar o aliasing.

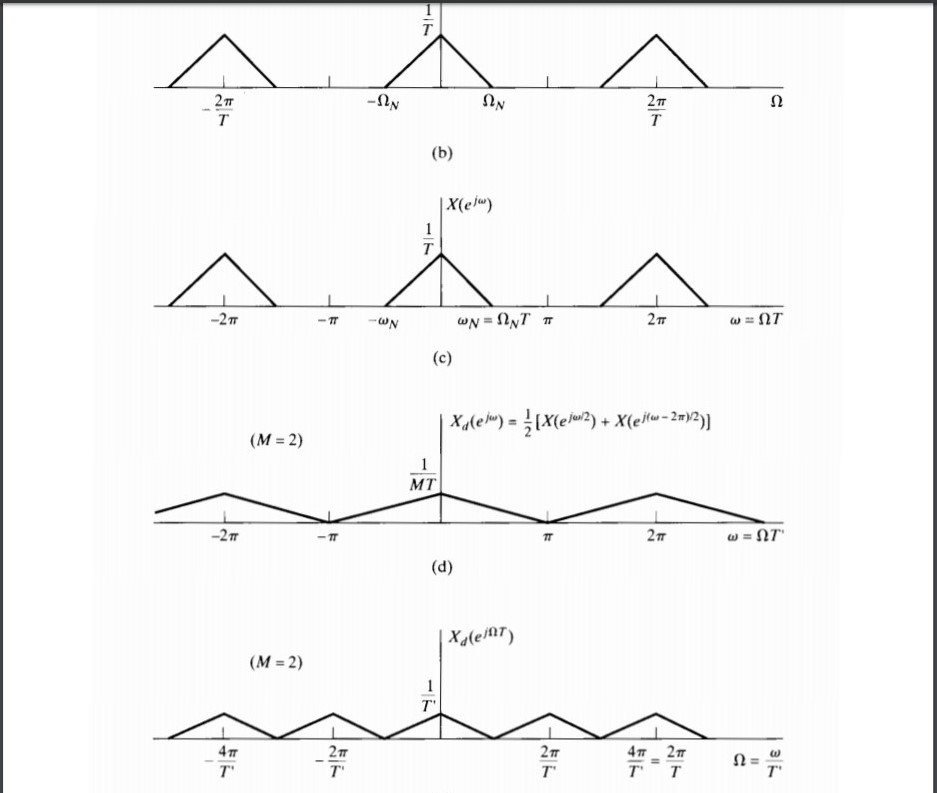


Figura 8- Downsampling

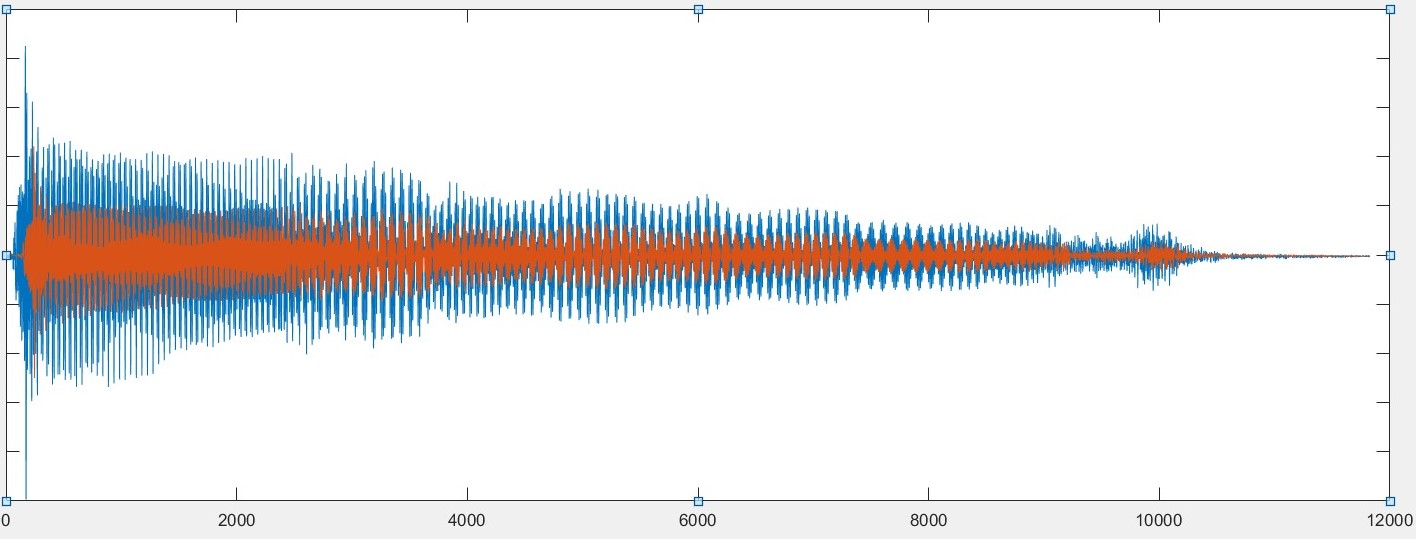
Resultados Obtidos

Figura 9-Exemplo de ficheiro .wav filtrado com a janela de Kaiser

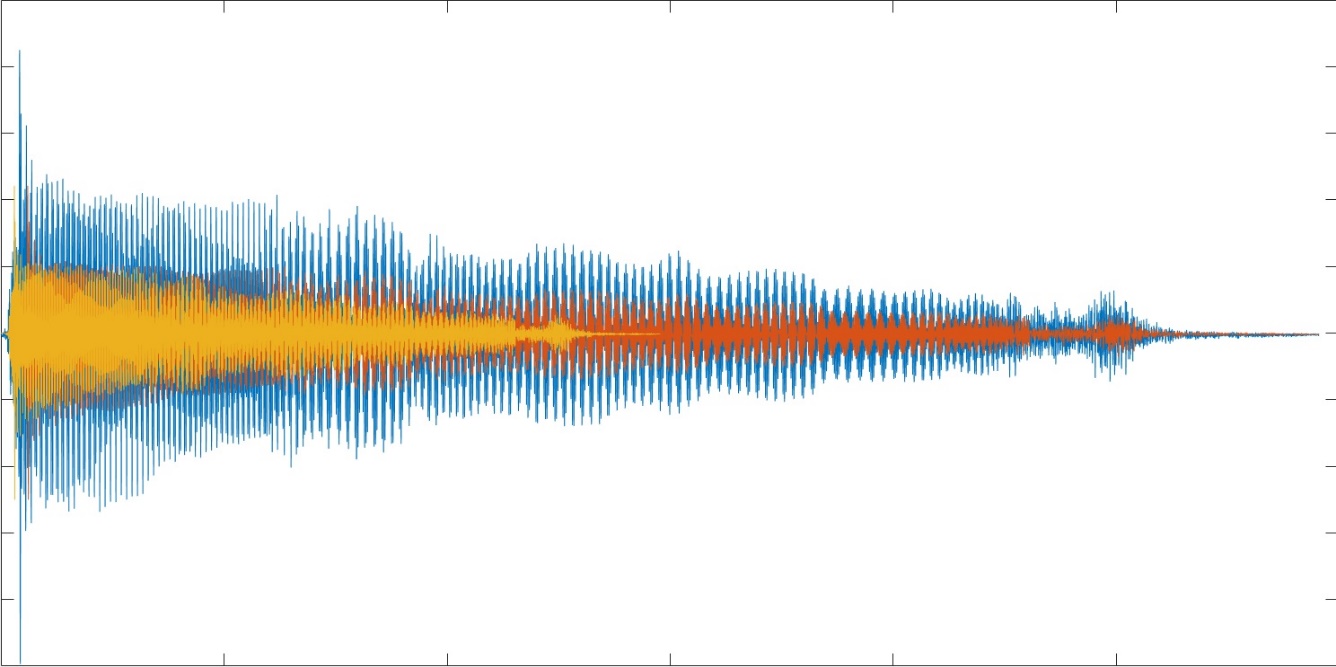


Figura 10- Exemplo de ficheiro .wav com janela de Kaiser e respetiva decimação com fator 2

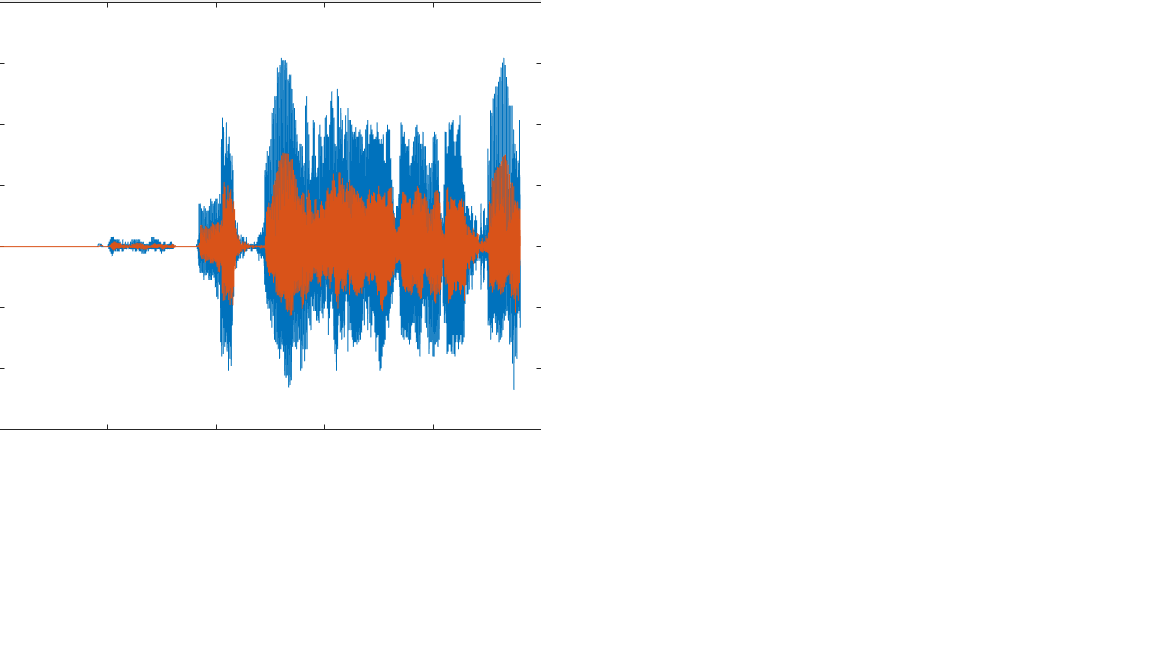


Figura - Excerto de voz gravada e filtrada com janela de Kaiser

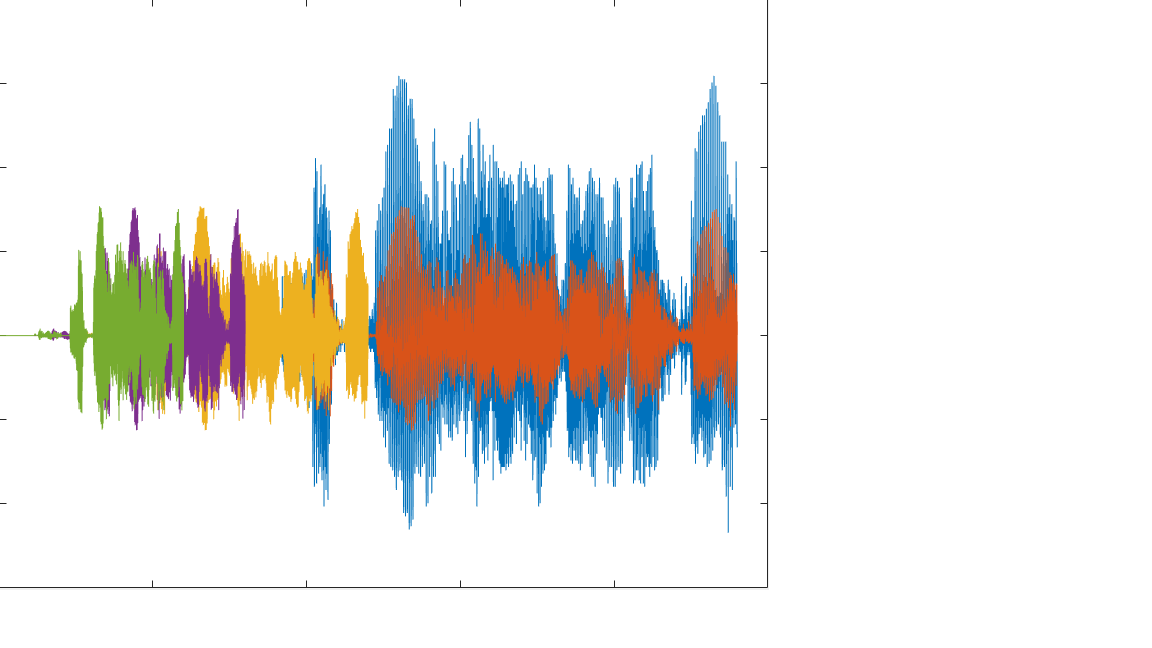


Figura - Excerto de voz gravada, filtrada com janela de Kaiser e decimada 3 vezes

Conclusão

Após a realização deste trabalho é possível afirmar que os conhecimentos ao nível de filtros e processamentos de sinais digitais foram expandidos, tanto a nível prático como a nível teórico.

Foi possível, além dos conhecimentos em processamento de sinal, interiorizar diversos conceitos relacionados com o Matlab, nomeadamente as funções dos filtros bem como o significado e a influência que as diversas variáveis de entrada têm para com o resultado da saída da função, neste caso, o respetivo filtro.