

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO MINHO



## **Remoção de ruídos e silêncios entre palavras**

**Marcos André Ferreira de Medeiros Pereira**

Trabalho prático nº 2

Carlos Manuel Gregório Santos Lima

Julho de 2013

# Índice

Introdução	3
1. O Problema	4
2. Resolução	5
3. Resultado Final	7
Conclusão	8
Referências	9
Anexos	10

# Introdução

“Remoção de ruídos e silêncios entre palavras” foi o trabalho proposto pelo professor Carlos Manuel Gregório Santos Lima da Unidade Curricular (UC) Processamento Digital de Sinal.

Para a elaboração deste trabalho prático, recorri a material de apoio relativo às aulas do docente da UC – seus slides e apontamentos tirados nas mesmas -, bem como páginas, tutoriais e filmes da Internet.

Começarei por apresentar o problema e explicar o seu objectivo.

De seguida, irei demonstrar qual a resolução escolhida por mim.

Seguidamente, passarei para o resultado final.

Para melhor explicação deste trabalho, apresentarei gráficos durante o mesmo e os códigos estarão em *Anexos*.

Terminarei o trabalho com a conclusão, reflexão crítica, referências e anexos para conclusão do mesmo.

# 1. O Problema

Este trabalho teve como objectivo, além de nos inculir a programação em *Matlab* e habituarmo-nos a diferentes ambientes daquilo a que estamos habituados a programar, remover – via *Matlab* – silêncios e ruídos de uma frase gravada por nós. Teve também o propósito de nos mostrar uma componente prática da UC.

## 2. Resolução

Recorrendo ao Matlab e com auxilio às aulas resolvi o problema proposto.

Para gravar um som recorri ao *audiorecorder*<sup>1</sup>. Para poder visualizar a forma de onda, fiz um *plot(fala)* cujo o resultado é mostrado na seguinte figura.

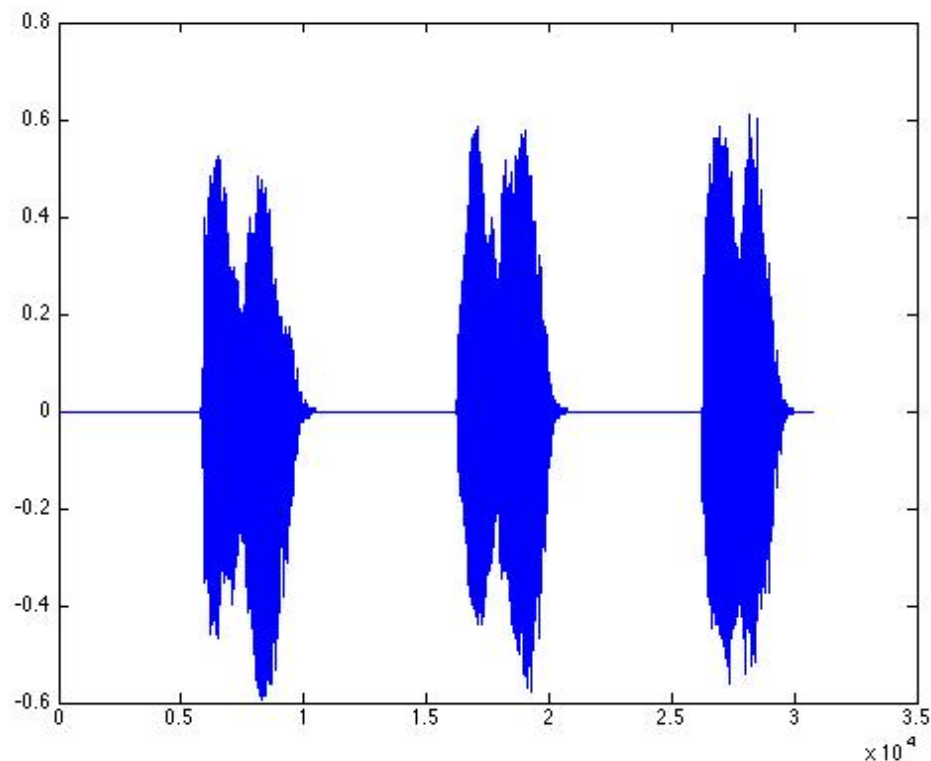


Figura 1: *plot(fala)* de "Olá olá olá"

A média, a variância e o *Threshold*<sup>2</sup> (através dos comandos *mean*<sup>3</sup> e *var*<sup>4</sup>, respectivamente) são importantes de calcular; o primeira dá-nos o valor para onde os dados se encontram mais concentrados numa distribuição e o segundo indica o quão longe, em geral, os valores se encontram do valor esperado.

Aprendemos que, para um modelo de ruído, a zona que está fora do *Threshold* – que é equivalente a *média + 3\*desviopadrão* – é chamada de *outlier* (figura 2). É esta zona que queremos silenciar. O código correspondente é dado através da seguinte condição:

```

if (som(i) > média + 3*sqrt(variância) | - (média + 3*sqrt(variância)))
%significa que é outlier
    i = 0;
end
    i = i + 1;

```

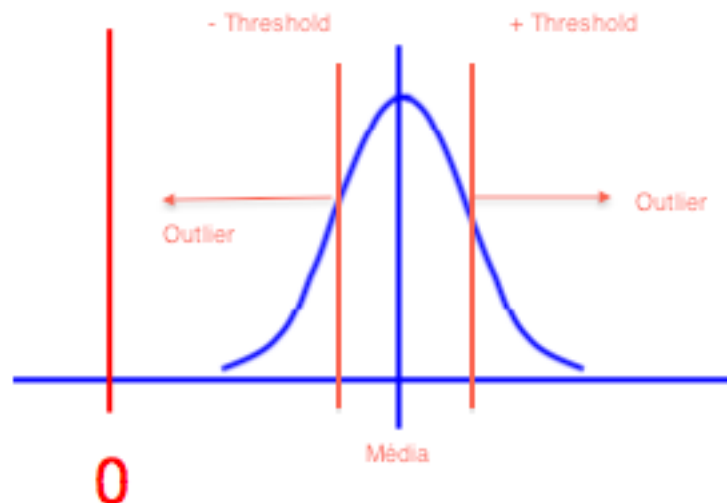


Figura 2: Modelo de Ruído

Como não restringi um tempo de fala, calculei um valor máximo de janela:

```

maioertamanho = length(fala) - 30; % calcula janela com maior
    tamanho,          % descontando 30 amostras
apontador_ANT = 0;      % apontador para o vector de
                        fala(i)
apontador_novo = 1;      % apontador para o vector
    resultante de % fala(i) sem os silencias

```

É num momento de fala que várias instruções têm de ser efetuadas (ver anexos)<sup>5</sup>.

### 3. Resultado Final

Após ser feito um *plot* do sinal final (figura 3), repara-se que o resultado final cumpriu com os requisitos do problema proposto na aula.

Notasse que os ruídos e silêncios entre as palavras foram removidos.

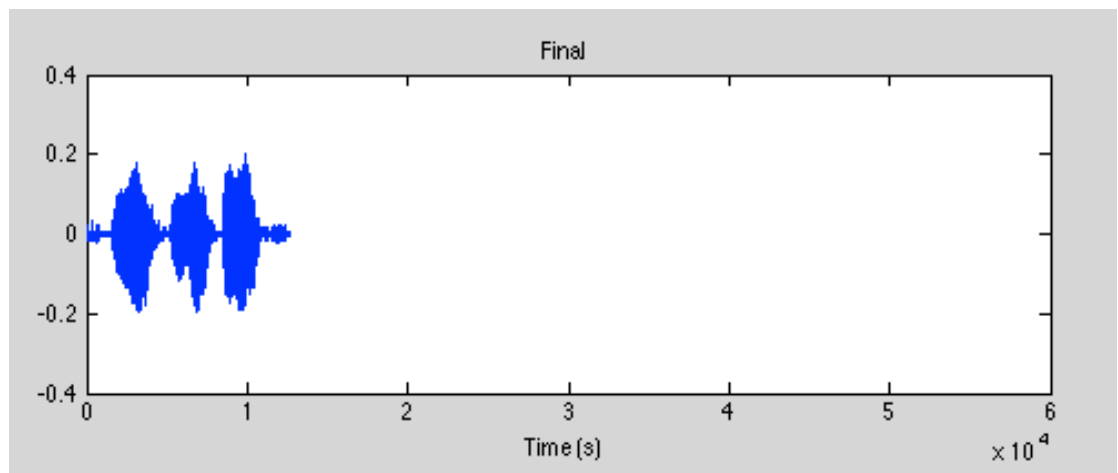


Figure 3: "Olá Olá Olá" sem ruídos e silêncios

# Conclusão

Iniciei este trabalho com a demonstração do objectivo deste problema.

Depois, mostrei a resolução de como o fiz, seus códigos e gráficos.

Por fim, mostrei o resultado final.

Terminarei o trabalho com uma reflexão crítica do mesmo, respectivas referências e anexos, onde estarão algumas definições que achei importante realçar durante o trabalho, bem como algumas funções e código *Matlab* usado.

Sobre este trabalho, considere-o bastante importante e uma excelente ideia. Penso que, visto que a UC complementar a este semestre – Processamento de Sinal - tem menos matéria, deveria ser uma boa aposta a programação em *Matlab*. A minha experiência com este trabalho é a prova de que uma componente prática a esta UC seria uma boa aposta, mas uma vez que não existe penso que deveríamos ter um pequeno curso em *Matlab*.



# Referências

Slides e apontamentos retirados das aulas da Unidade Curricular Processamento Digital de Sinal;

<http://pt.wikipedia.org/wiki/M%C3%A9dia>

<http://pt.wikipedia.org/wiki/Vari%C3%A2ncia>

[http://pt.wikipedia.org/wiki/Ru%C3%ADdo\\_branco\\_\(estat%C3%ADstica\)](http://pt.wikipedia.org/wiki/Ru%C3%ADdo_branco_(estat%C3%ADstica))

[http://pt.wikipedia.org/wiki/Distribui%C3%A7%C3%A3o\\_normal](http://pt.wikipedia.org/wiki/Distribui%C3%A7%C3%A3o_normal)

[http://pt.wikipedia.org/wiki/Fun%C3%A7%C3%A3o\\_de\\_Gauss](http://pt.wikipedia.org/wiki/Fun%C3%A7%C3%A3o_de_Gauss)

<http://www.mathworks.com/matlabcentral/answers/28513>

<http://www.mathworks.com/help/matlab/ref/audioread.html>

<http://www.mathworks.com/help/matlab/ref/sound.html>

<http://www.mathworks.com/help/matlab/ref/audiorecorder.ht>

<http://www.mathworks.com/help/matlab/ref/audiorecorder.html>

# Anexos

## Audiorecorder<sup>1</sup>

```
som = audiorecorder ();  
disp('Start Speaking.');
```

record(som);

```
stop(som);  
disp('Finish.');
```

fala = getaudiodata(som);  
plot(fala);

## Threshold<sup>2</sup>, mean<sup>3</sup> e var<sup>4</sup>

```
media = mean(lugar);           % calculo da média  
variancia = std(lugar);        % calculo da variância  
threshold = media + 3*variancia; % calculo do Threshold
```

## Código<sup>5</sup>

```
while maiortamanho > 0 % enquanto tiver amostras para ler de fala(i)
    threshold_mais = 0;
    threshold_menos = 0;

    for i=1 : 1 : 30
        if(abs(janela_amostras(i)) > abs(threshold)) % confirma se o sinal esta acima
%ou abaixo do threshold
            threshold_mais = threshold_mais + 1;
        else
            threshold_menos = threshold_menos + 1;
        end
    end

    if(threshold_menos ~= 30)
        Sinal_resultante(apontador_novo) = janela_amostras(1); % caso seja, copia
% para o vector sinal_resultante o primeiro valor do vector de 30 amostras
        apontador_novo = apontador_novo + 1; % incrementa o apontador
% do vector Sinal_resultante
    end

    apontador_ANT = apontador_ANT + 1; % incrementa do
%apontador fala()

    for i = 1 : 1 :30
        janela_amostras(i) = fala(i + apontador_ANT); % desloca a janela
%uma unidade para a direita
    end

    maiortamanho = maiortamanho - 1 ; % decrementa uma unidade ao
%tamanho da janela
end
```