

**Universidade do Minho**

**PROCESSAMENTO DIGITAL DE SINAL  
2012/2013**

# **Separação do Ruído à fala**

**Ricardo Amorim**

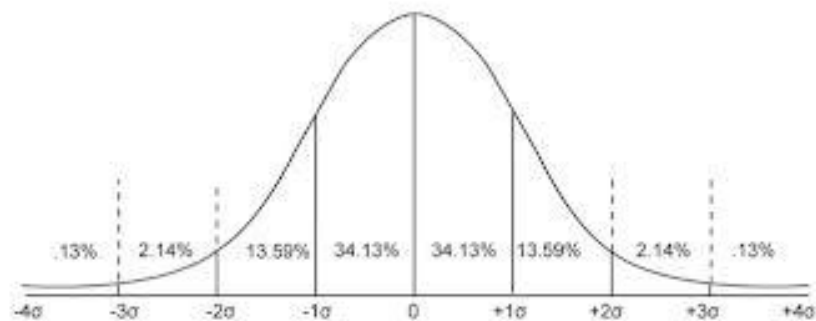
**61998**

**MIEEIC**

## Introdução

Com este trabalho pretende-se, recorrendo ao MatLab, elaborar um algoritmo que permita partir de um sinal de audio original e retirar apenas a parte da fala. Isto é, a uma sinal separam-se as partes de ruído ficando apenas com a parte que interessa, a fala.

Para o efeito recorreu-se a métodos estatísticos pelo que será feita uma breve explicação.



### Ilustração 1

#### Curva de distribuição normal

O método estatístico utilizado refere-se à curva de distribuição normal. Utilizando os conceitos de média e variância é possível fazer estimativas.

Relacionando isto com o trabalho propriamente dito, pretende-se estabelecer um método de forma a poder ser perceptível se uma parte do sinal é ou não ruído. Para isso, deve-se analisar uma parte do sinal que se tenha a certeza ser ruído, calcula-se o seu valor médio e variância. Utilizando as propriedades da distribuição normal assume-se que se:

$$V_{\text{médio-variância}} < V_{\text{medido}} < V_{\text{médio+variância}}$$

Então, trata-se de ruído.

## Procedimento

Para fazer o pretendido começa-se por gravar o sinal utilizando o comando do Matlab

```
x=wavrecord[N,freq]
```

N- numero de amostras

freq –frequencia de amostragem.

Uma vez tendo o sinal a analisar é necessário fazer então o processo que faz o cálculo do ruído e separe esta componente da fala.

Como já foi referido calcula-se o valor médio e variância e depois compara-se as amostras para verificar se podem ou não ser consideradas ruído. Ainda assim, podem surgir erros como por exemplo considerar ruído quando é fala e vice-versa. No caso de surgir um barulho de fundo que pode ser considerado erradamente como fala.

Para resolver estes problemas a análise feita em vez de ser amostra a amostra é em segmentos. Faz-se a contabilização do número de amostras dentro dos valores da curva e fora. Se houver mais amostras dentro é ruído, senão é fala.

Neste trabalho serão analisadas três situações

- Silêncio – Fala
- Fala – Silêncio – Fala
- Silêncio – Ruído - Fala

## Script Matlab

```
function [noisebuff, buff] = mruido(x, nseg, wlen, fr)

m=mean(x(1:nseg));
v=var(x(1:nseg));

alfa=fr*wlen;
buff=0;
noisebuff=0;
a = 1:1:length(x);

if(rem(wlen,2)==1)
    wlen=wlen+1;
end

for i=1: length(x)
    if(abs(x(i)-m)>sqrt(v))
        a(i)=1;
    else
        a(i)=0;
    end
end

initw=1;
endw=initw+wlen;
prev_noise=0;

while(endw<length(x))
    w=a(initw:endw);
    if(sum(w)>alfa)
        if(prev_noise==1)
            buff=cat(1, buff, x(initw:endw));
        else
            buff=cat(1, buff, x(initw+wlen/2:endw));
        end
    else
        if(prev_noise==1)
            noisebuff=cat(1, noisebuff, x(initw+wlen/2:endw));
        else
            noisebuff=cat(1, noisebuff, x(initw:endw));
        end
    end

    initw=initw+wlen/2;
    endw=endw+wlen/2;

    if(endw>length(x))
        endw=length(x);
        initw=endw-initw;
    end
end

subplot(3,1,1), plot(x);
subplot(3,1,2), plot(buff);
subplot(3,1,3), plot(noisebuff);
```

## Silencio Fala

```
[noisebuff, buff] = mruido(x, 500, 500, 1/2);
```

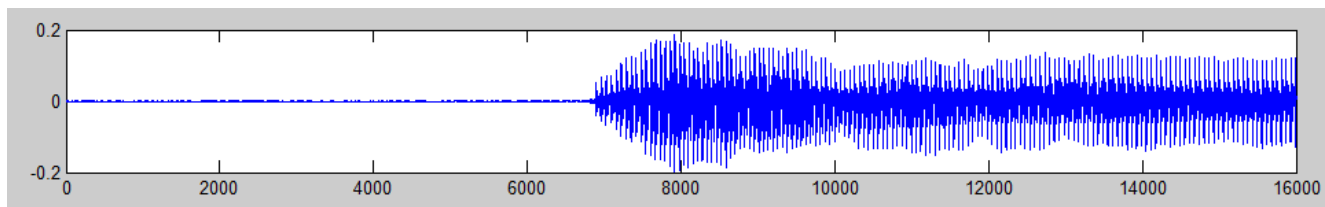


Ilustração 2  
Sinal original

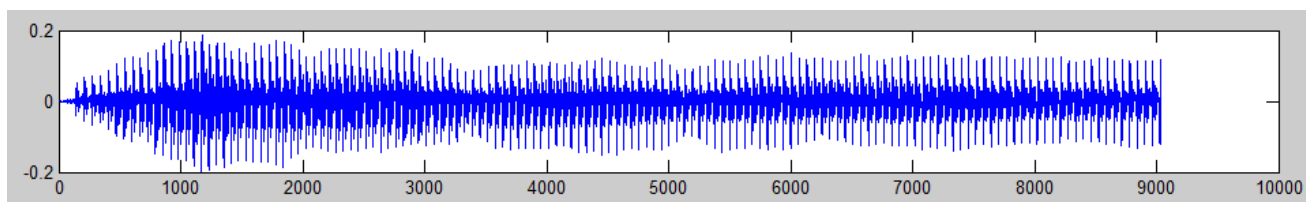


Ilustração 3  
Componente Fala

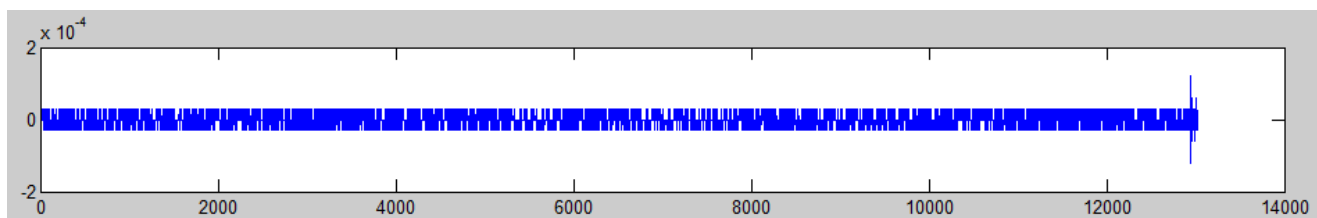


Ilustração 4  
Componente Ruído

## Silêncio -> Fala -> Silêncio -> Fala

```
[noisebuff, buff] = mruido(y, 1500, 200, 4/5);
```

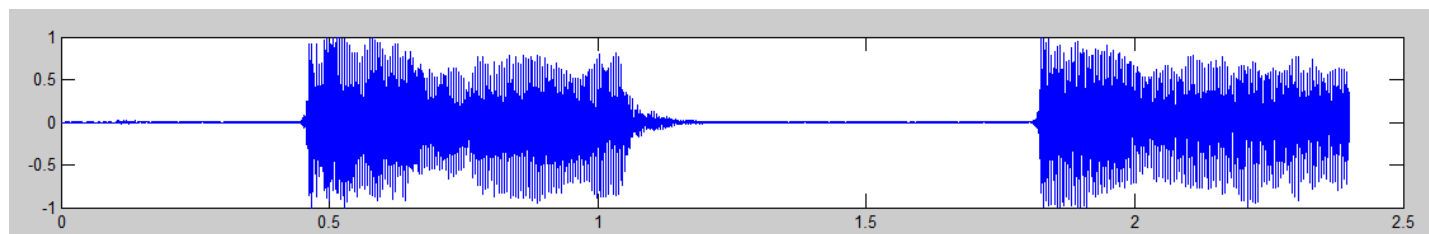


Ilustração 7  
Sinal Original

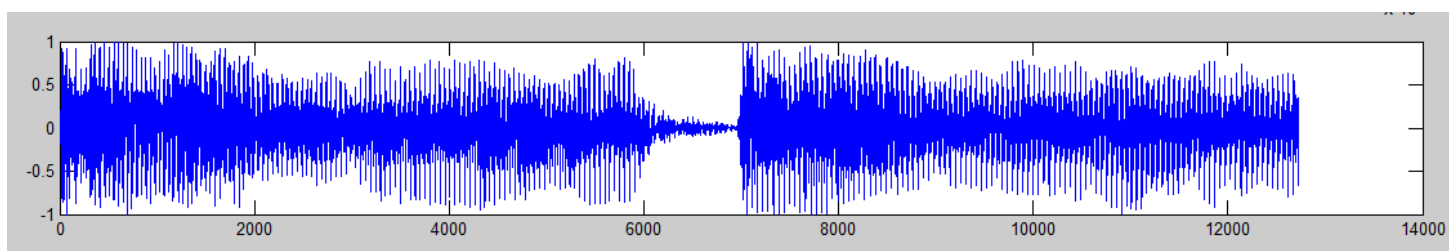


Ilustração 6  
Componente Fala

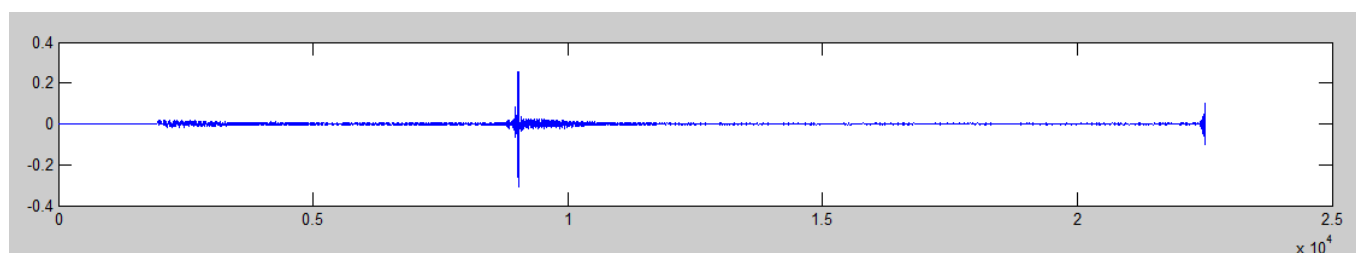


Ilustração 5  
Componente Ruído

## Silêncio, barulho, fala.

```
[noisebuff, buff] = mruido(x, 2500, 1000, 4/5);
```

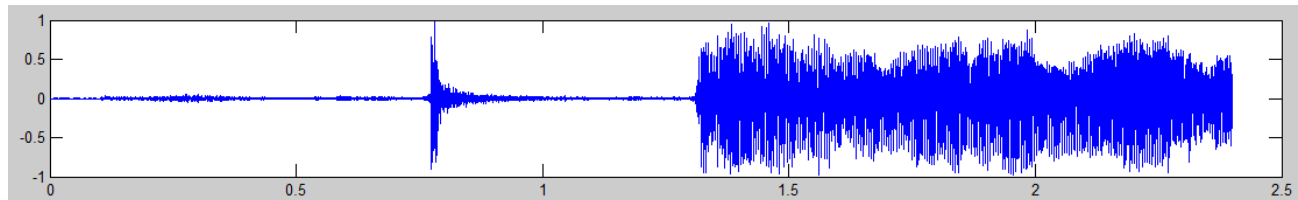


Ilustração 8  
sinal Original

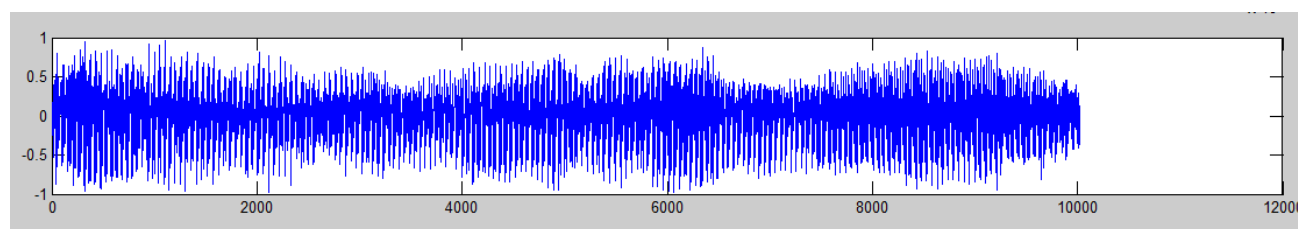


Ilustração 9  
Componente Fala

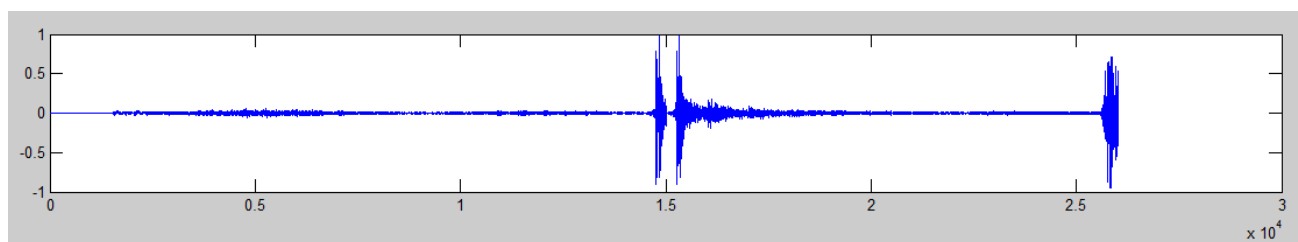


Ilustração 10  
Componente Ruído

## Conclusão

O trabalho realizado correu bem visto que no final foi atingido o objectivo traçado. Com recurso a simples processos estatísticos e a uma função do Matlab, foi possível retirar as partes indesejadas a um sinal de áudio de forma e reduzir ao seu tamanho e ficar apenas com o que interessa ao utilizador.

Logo à partida surge como vantagem neste processo, o facto de poupar memória visto que foi truncado o sinal. Para além disso, é muito mais agradável de trabalhar com um sinal que tem apenas o necessário e não ser preciso andar a procura do conteúdo que verdadeiramente interessa.