



Universidade do Minho
Escola de Engenharia

Mestrado integrado em Engenharia Eletrónica e
Computadores

Trabalho de processamento digital de sinal

Nome: Rui Filipe Oliveira Costa

Número: 80632

Docente: Carlos Manuel Gregório Santos Lima



Universidade do Minho
Escola de Engenharia

Mestrado integrado em Engenharia Eletrónica e Computadores

Índice:

1. Introdução
2. Parte Teórica
3. Resultados
4. Conclusão
5. Anexos



Universidade do Minho
Escola de Engenharia

Mestrado integrado em Engenharia Eletrónica e Computadores

1. Introdução:

O trabalho proposto foi implementar um algoritmo para deteção dos intervalos de silêncio na fala. Este algoritmo é útil pois em comunicações os intervalos de silêncio não são enviados devido a não conterem informação linguística. Para isso, pretende-se usar o conhecido “shewhart protocol” mas com threshold adaptativo às condições de ruído.

Este mecanismo requer:

- 1) Estudo do melhor valor de threshold para cada relação sinal-ruído. Pretende-se que faça este estudo para SNR de 0 a 50 dB com intervalos de 10 dB. Este estudo requer síntese de ruído branco, soma ao sinal e detecção dos segmentos contendo apenas ruído usando vários valores do threshold, valores entre 0.2 e 5, que devem ser colocados numa tabela. Trata-se de um estudo heurístico que deve ser justificado em relatório.
- 2) Cálculo automático da SNR e aplicação dos valores calculados no ponto 1).



2. Parte teórica:

O ruído consiste numa alteração das características do sinal, em que essa alteração pode ser devido ao efeito de um outro sinal exterior ao sistema de transmissão ou gerado pelo próprio. A natureza deste sinal é aleatório, o que significa que não é possível prever o seu comportamento, por esse motivo deve ser tratado recorrendo a métodos estatísticos. O efeito que o ruído no sinal, pode ser aditivo, em que o ruído se soma ao sinal, ou multiplicativo, em que o ruído é multiplicado pelo sinal.

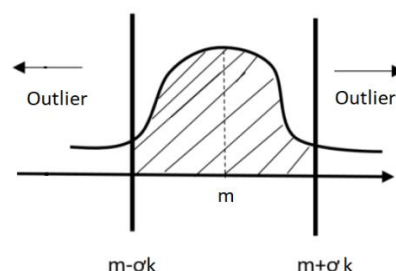
O ruído branco é um sinal aleatório com igual intensidade em diferentes frequências, o que lhe dá uma densidade espectral de potência constante. Em tempo discreto as suas amostras são dadas como variáveis aleatórias com distribuição normal e de média zero.

Nos sistemas de transmissão analógicos, a qualidade do sinal recebido mede-se através da relação entre a potência do sinal e a potência do ruído (SNR – Signal to Noise Ratio). Essa relação é definida pela divisão da potência de um sinal pela potência do ruído sobreposto a esse mesmo sinal. Resumidamente, o SNR mede a fidelidade de reprodução dos sinais, comparando o nível de um sinal desejado com o nível de ruído, quanto maior for o SNR, menor será o efeito do ruído. Para calcularmos o SNR:

$$\text{SNR} = \frac{P_{\text{signal}}}{P_{\text{noise}}} = \left(\frac{A_{\text{signal}}}{A_{\text{noise}}} \right)^2$$
$$\text{SNR}_{\text{dB}} = 10 \log_{10} \left[\left(\frac{A_{\text{signal}}}{A_{\text{noise}}} \right)^2 \right] = 20 \log_{10} \left(\frac{A_{\text{signal}}}{A_{\text{noise}}} \right)$$

Sendo que P representa a potência e o A valor médio da amplitude. Estes valores têm de ser medidos no mesmo ou em pontos equivalentes do sistema e dentro da mesma largura de banda.

Walter Shewhart desenvolveu um método, o qual permite verificar se um processo se encontra sob controlo ou não, através da estatística que os seus dados apresentem. Para tal são necessárias amostras, a média dessas amostras (m), que indica o centro do gráfico, e o desvio padrão dessas amostras (σ), o qual é depois multiplicado pelo valor de threshold (k) e que é depois usado para definir os limites de controlo.





Para calcular a média e variância:

$$m_x(n) = E[X_n] = \int_{-\infty}^{+\infty} x f_{X_n}(x) dx$$

$$\sigma_x^2(n) = E[(X_n - m_x(n))^2] = \int_{-\infty}^{+\infty} (x - m_x(n))^2 f_{X_n}(x) dx$$

Se os valores estiverem incluídos no intervalo $[m-\sigma, m+\sigma]$ podemos dizer que o sinal é ruído, se os valores estiverem fora dos parâmetros, afirmámos que o sinal é de fala.

- Ruído: $P(|x-m| < \sigma)$
- Fala: $P(|x-m| > \sigma)$

No entanto podem existir partes da fala dentro dos parâmetros do ruído ou existirem períodos de pausa no meio, sendo estes barulhos confundidos com o sinal da fala. Para resolver este problema, temos de analisar um conjunto de amostras em vez de uma só amostra.

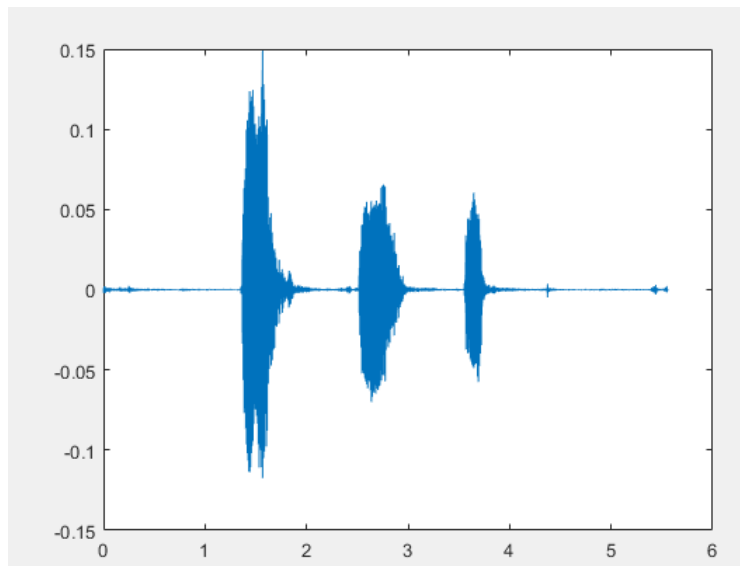


Universidade do Minho
Escola de Engenharia

Mestrado integrado em Engenharia Eletrónica e Computadores

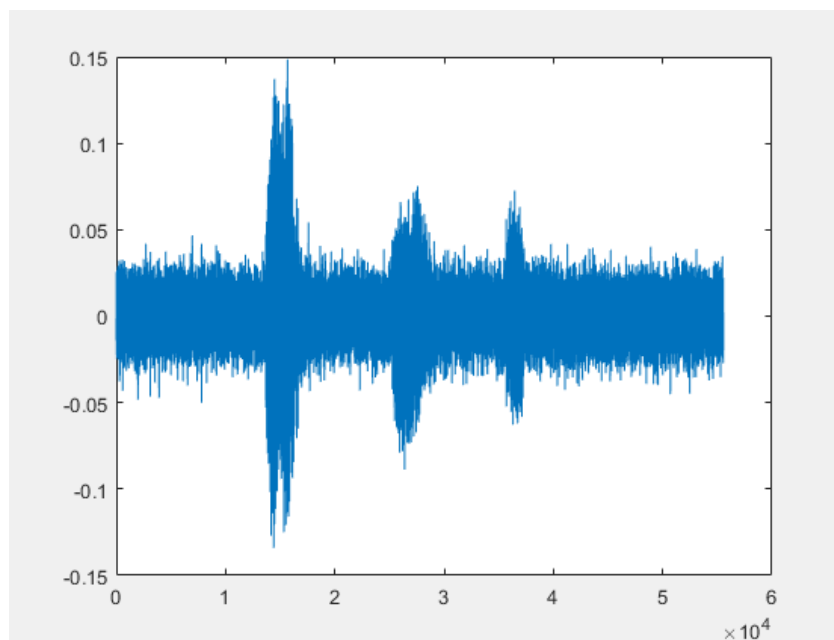
Resultados:

Inicialmente, grava-se um som pausado durante 5 segundos. A gravação foi feita num programa auxiliar (audacity), com uma frequência de amostragem de 11025 Hz e foi utilizado um canal de 16 bits. O sinal está representado a seguir:



De seguida, de forma a encontrar os melhores valores do threshold, é necessário contaminar o sinal de entrada com um ruído de amplitude tal que provoque um SNR pretendido.

Para um SNR de 0dB, o sinal resultante depois da soma do ruído foi o seguinte:

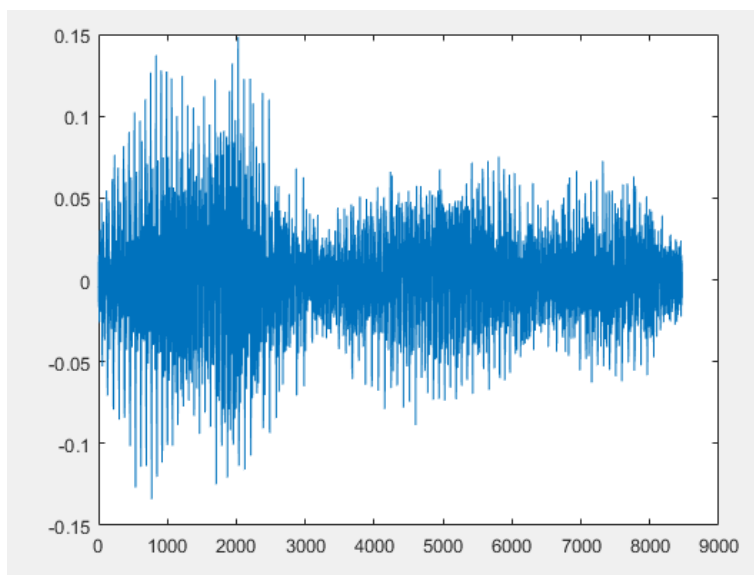




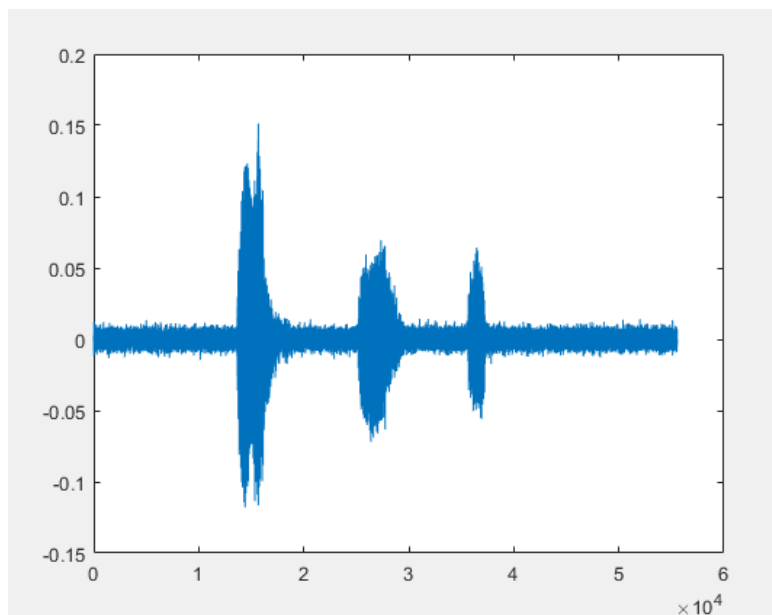
Universidade do Minho
Escola de Engenharia

Mestrado integrado em Engenharia Eletrónica e Computadores

Começando com um threshold de 0.2 e aumentado de 0.1 em 0.1, determinou-se que o melhor valor para obter um sinal sem pausas seria o 0.85 obtendo-se o seguinte sinal:



Para um SNR de 10dB, o sinal resultante fica:

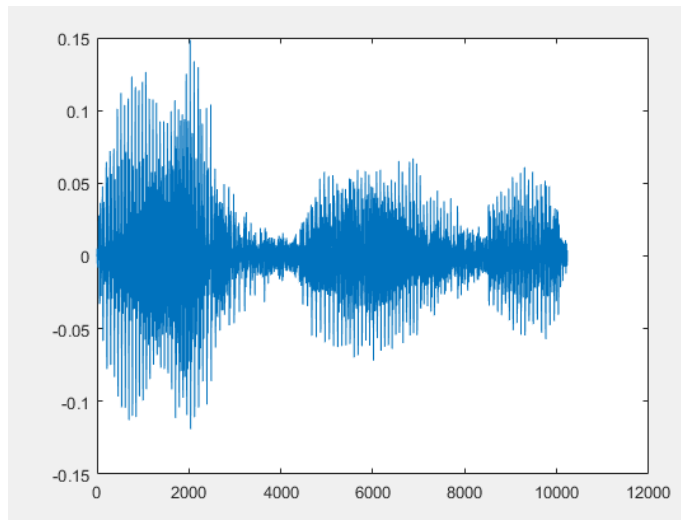




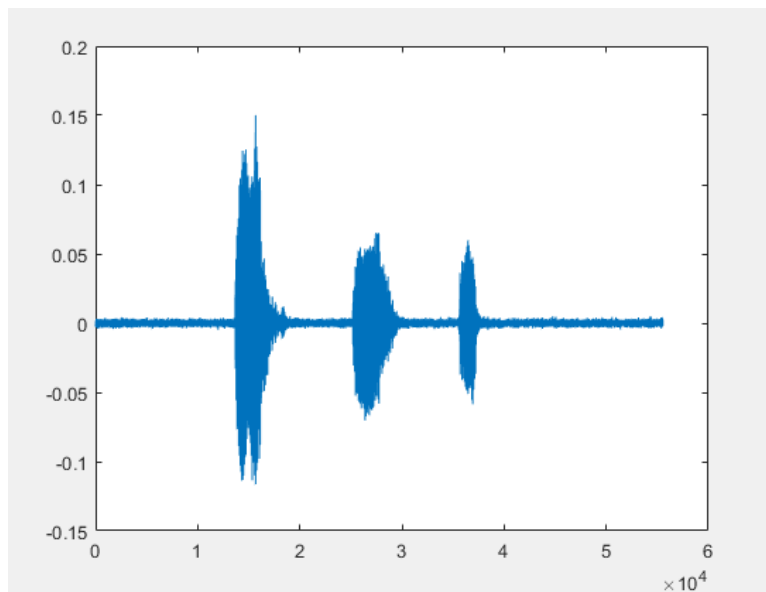
Universidade do Minho
Escola de Engenharia

Mestrado integrado em Engenharia Eletrónica e Computadores

Para um SNR de 10 dB, o valor de threshold foi de 1.1



Para um SNR de 20 dB, o sinal inicial contaminado pelo ruído fica o seguinte:

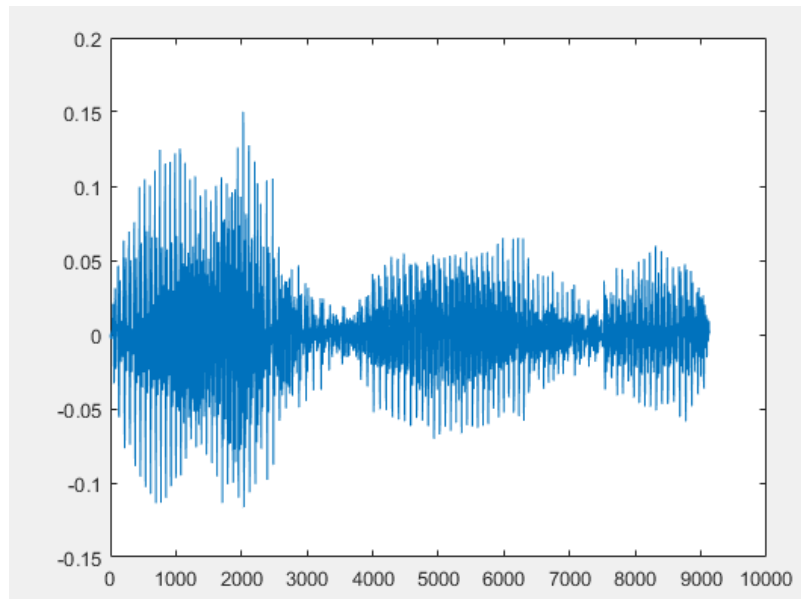




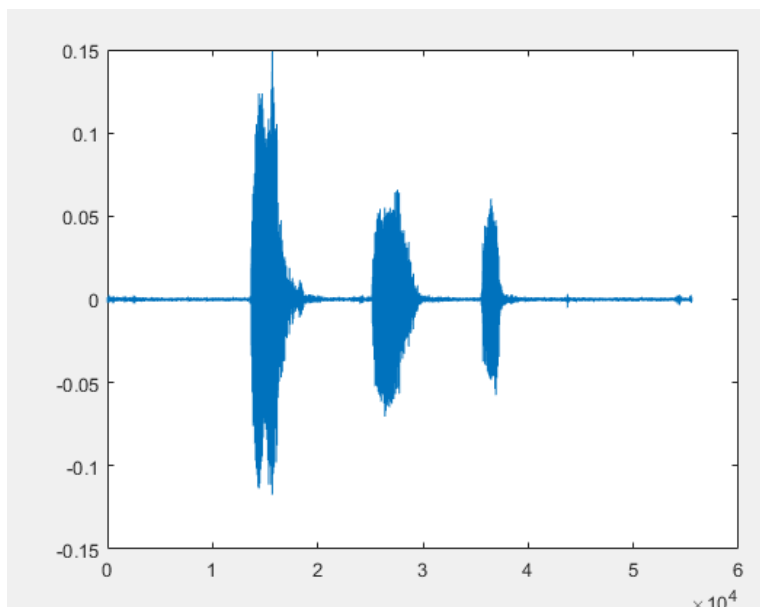
Universidade do Minho
Escola de Engenharia

Mestrado integrado em Engenharia Eletrónica e Computadores

Com o valor de SNR de 20 dB, fomos à procura do melhor valor de threshold e o valor que obtivemos foi de 4, resultando no seguinte sinal:



Para um SNR de 30 dB, o sinal resultante da contaminação do ruído foi:

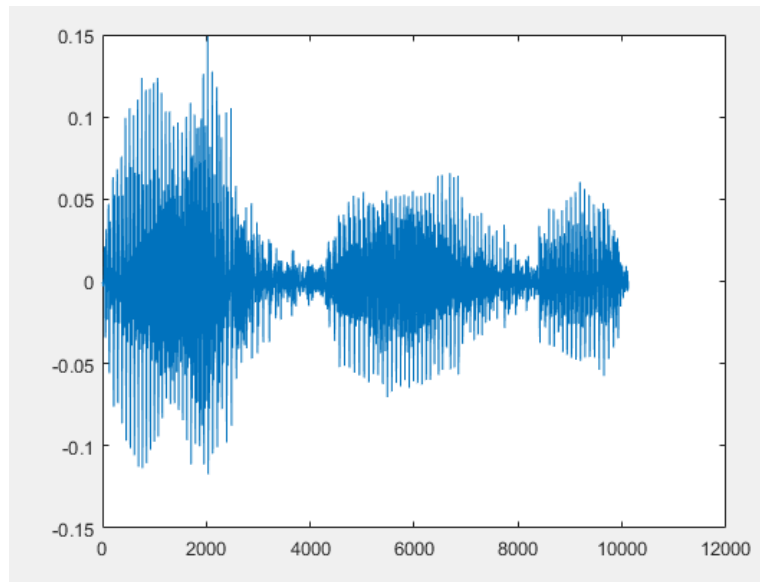




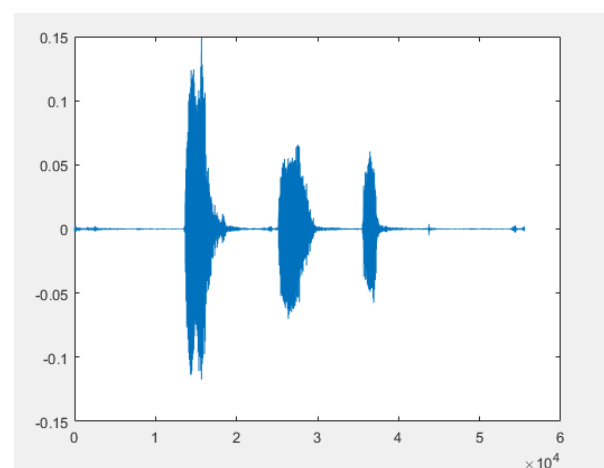
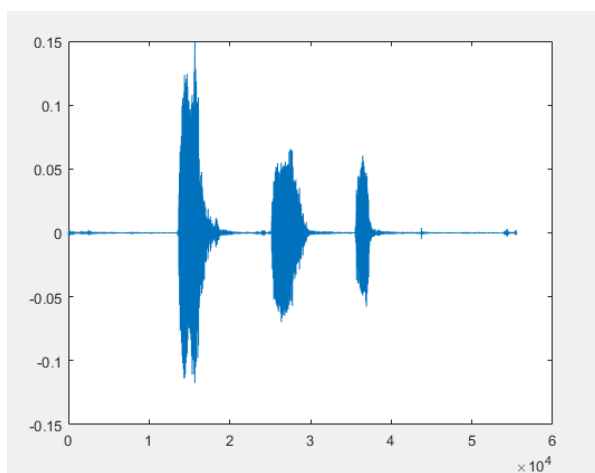
Universidade do Minho
Escola de Engenharia

Mestrado integrado em Engenharia Eletrónica e Computadores

O melhor valor de threshold para um SNR, após várias tentivas concluiu-se que era de 6:



Para valores de SNR de 40 dB e 50 db, podemos verificar que o sinal resultante da ação do ruído são muito similares, devido aos valores altos de SNR o que significa que a soma do ruído já não é tão sentida.

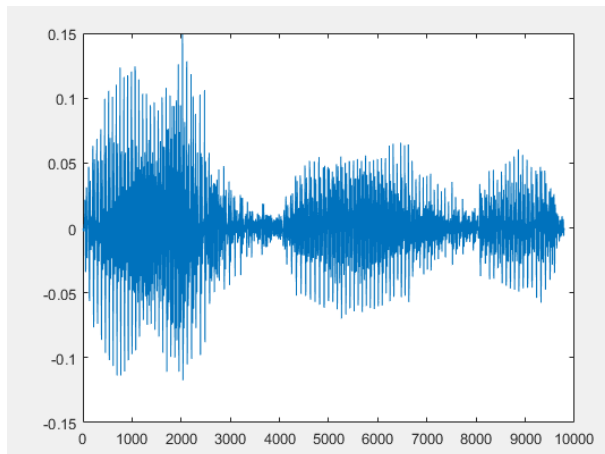




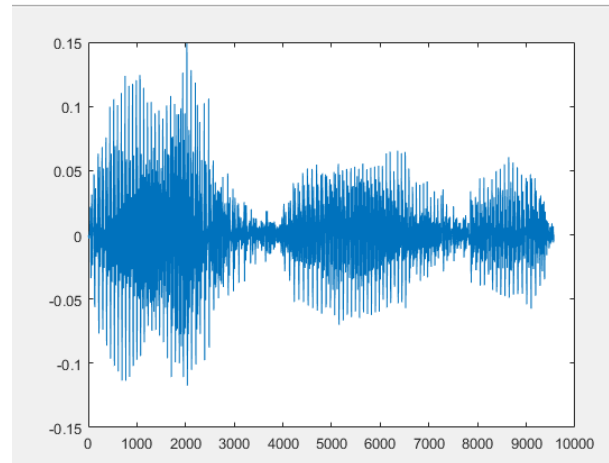
Mestrado integrado em Engenharia Eletrónica e Computadores

Universidade do Minho
Escola de Engenharia

SNR de 40dB, threshold de 8:



SNR de 50dB, threshold de 9:



Após fazermos os testes anteriores, foi possível elaborar a seguinte tabela:

SNR(dB)	Threshold
0 dB	0.85
10 dB	1.1
20 dB	4
30 dB	6
40 dB	8
50 dB	9

A partir da tabela anterior, é agora possível realizar um algoritmo que calcule o melhor valor de threshold para valores de SNR diferentes, removendo assim da melhor forma os silêncios presentes no sinal.



Universidade do Minho
Escola de Engenharia

Mestrado integrado em Engenharia Eletrónica e Computadores

Conclusão:

Este trabalho foi possível verificar como funciona um algoritmo de deteção de silêncios, pode envolver o cálculo do SNR até à produção de um ruído para conseguirmos ajustar o algoritmo da melhor maneira, envolvendo componentes de estatística, baseando-se em métodos de controlo como o de Shewhart e adaptando-se às propriedades do nosso sinal.

Conclui-se que quanto maior o SNR, maior o threshold que se aplica ao algoritmos, pois a influência do ruído passa a ser menos e não é preciso aplicar limites tão restritos para a diferenciação entre sinal e ruído.



Mestrado integrado em Engenharia Electrónica e Computadores

Universidade do Minho
Escola de Engenharia

Anexos:

```
function sinal=contaminar(recorded,SNR)

%%contamina o sinal recorder com um ruído de média 0 e potência que
%%corresponde ao SNR desejado

S=mean(recorded.^2); %potência do sinal

noise=randn(1,length(recorded)); %ruído branco com distribuição normal
noise=noise-mean(noise); %ruído branco com média 0
noise=noise * sqrt(S/10^(SNR/10)); %ruído com potência pretendida - vem da fórmula do SNR
sinal=recorded+noise'; %contaminação do sinal recebido

function [sinal_sem_ruído,ruído]=retirar_ruído(sinal,threshold,Fs)

amostrasPorFrame = floor(Fs/100);
bgSampleCount = floor(Fs/5); %a primeira quinta parte do sinal é ruído

%calcula a média e o desvio padrão do ruído
bgSample=[];
for i=1:1:bgSampleCount
    bgSample = [bgSample sinal(i)];
end
media=mean(bgSample);
desvio=std(bgSample);

%identificar partes vozeadas para cada valor
for i=1:1:length(sinal)
    if(abs(sinal(i)-media)/desvio > threshold)%se o valor da amostra menos a média do ruído, a dividir pelo desvio
        voiced(i)=1; %guarda em voiced indicação de quais amostras no sinal são vozes
    else
        voiced(i)=0;
    end
end

%identificar as partes vozeadas por frame
amostrasUteis=length(sinal)-mod(length(sinal),amostrasPorFrame);
%quantos frames tem de percorrer
frameCount = amostrasUteis/amostrasPorFrame;
voicedFrameCount=0;
for i=1:1:frameCount %percorrer todos os frames
    cVoiced=0;
    cUnvoiced=0;
    for j=i*amostrasPorFrame - amostrasPorFrame + 1:1:(i*amostrasPorFrame) %percorrer todos os frames
        if (voiced(j)==1)
            %conta as amostras que são vozeadas
            cVoiced=(cVoiced+1);
        else
            %conta as amostras que não são vozeadas
            cUnvoiced = cUnvoiced + 1;
        end
    end
    %a frame contém mais amostras vozeadas que não vozeadas
    if (cVoiced > cUnvoiced)
        %conta as frames vozeadas
        voicedFrameCount = voicedFrameCount+1;
        voicedUnvoiced(i)=1; %indica que é uma frame vozeada
    else
        voicedUnvoiced(i)=0; %indica que é uma frame não vozeada
    end
end
```



Mestrado integrado em Engenharia Electrónica e Computadores

Universidade do Minho
Escola de Engenharia

```
        voicedUnvoiced(i)=0; %indica que e uma frame nao vozeada
    end
end

sinal_sem_ruido=[];
ruído=[];
for i=1:1:frameCount
    for j=i*amostrasPorFrame-amostrasPorFrame+1:1:(i*amostrasPorFrame)
        if(voicedUnvoiced(i)==1) %se e uma frame vozeada
            sinal_sem_ruido=[sinal_sem_ruido sinal(j)];
        else
            ruído=[ruído sinal(j)];
        end
    end
end

function threshold=melhor_threshold(sinal,Fs)

%Calculo do valor de threshold
potencia_sinal=mean(sinal.^2); %potencia do sinal
potencia_ruído=mean(sinal(1:(Fs/5)).^2); %potencia do ruído, considerando que a primeira quinta parte do sinal e

SNR=10*log10(potencia_sinal/potencia_ruído); %calcula do SNR

if (SNR >= 0 && SNR < 10)
    threshold = SNR*((1.1-0.85)/10)+0.85;
elseif (SNR >= 10 && SNR < 20)
    threshold = SNR*((4-1.1)/10)+(1.1-((4-1.1)/10)*10);
elseif (SNR >= 20 && SNR < 30)
    threshold = SNR*((6-4)/10)+(4-((6-4)/10)*20);
elseif (SNR >= 30 && SNR < 40)
    threshold = SNR*((8-6)/10)+(6-((8-6)/10)*30);
elseif (SNR >=40 && SNR < 50)
    threshold = SNR*((9-8)/10)+(8-((9-8)/10)*40);
elseif (SNR >= 50)
    threshold = 9;
end
```



Mestrado integrado em Engenharia Electrónica e Computadores

Universidade do Minho
Escola de Engenharia

```
function executar()  
[sinal, Fs] = audioread('pds.wav');  
  
%%Informacoes sobre o audio%%  
info = audiointro('pds.wav');  
disp(info);  
  
%%display do som gravado  
disp('Audio Inicial com Ruído')  
sound(sinal,Fs);  
pause(4);  
  
%-----Teste-----%  
%%sinal=contaminar(sinal,10);  
%%threshold=1.1;  
  
%calcula do threshold do audio  
threshold = melhor_threshold(sinal,Fs)  
  
%remocao do silencio  
[sinal_sem_ruído,ruído]=retirar_ruído(sinal,threshold,Fs);  
figure  
subplot(3,1,1);
```

```
subplot(3,1,1);  
plot(sinal);  
title('Sinal Inicial com ruído');  
subplot(3,1,2);  
plot(sinal_sem_ruído);  
title('Sinal sem Ruído');  
subplot(3,1,3)  
plot(ruído);  
title('Ruído');  
  
figure  
plot(sinal);  
figure  
plot(sinal_sem_ruído)  
  
disp('Audio Final sem Ruído')  
  
pause(2);  
sound(sinal_sem_ruído,Fs);
```