



Universidade do Minho  
Escola de Engenharia

*Mestrado Integrado em Engenharia Eletrónica Industrial e  
Computadores*

2019/2020

Processamento Digital de Sinal

**DETEÇÃO E REMOÇÃO DE INTERVALOS DE SILÊNCIO NA  
FALA**



Rafael Andrade Silva, A74437

Docente:

**Professor Carlos Manuel Gregório Santos Lima**



## Índice

Índice de Figuras .....	3
1 Introdução .....	4
2 Fundamentos Teóricos .....	5
2.1 SNR e Ruído Branco .....	5
2.2 Processos Estocásticos .....	6
2.3 Outlier .....	6
2.4 Controlo de Shewhart e Distância Normalizada .....	7
2.5 Zero-crossing .....	8
2.6 Filtro de Wiener .....	9
3. Análise do Problema .....	15
4. Fluxograma .....	18
5. Implementação do código .....	20
5.1 Gravação do Áudio .....	20
5.2 Tamanho da Janela .....	20
5.3 Valor SNR .....	20
5.4 Média e Variância .....	21
5.5 Matriz para guardar a informação .....	21
5.6 Declaração/Definição de Variáveis .....	21
5.7 Vetores de ruído ou fala .....	22
5.8 SNR .....	22
6. Testes e Resultados da Implementação .....	23
6.1 Geração de ruído branco – 0 dB .....	23
6.2 Geração de ruído branco – 5 dB .....	24
6.3 Geração de ruído branco – 10 dB .....	24
6.4 Geração de ruído branco – 15 dB .....	25
6.5 Geração de ruído branco – 20 dB .....	25
6.6 Geração de ruído branco – 25 dB .....	26



6.7 Geração de ruído branco – 30 dB .....	26
6.8 Geração de ruído branco – 35 dB .....	27
6.9 Geração de ruído branco – 40 dB .....	27
7. Conclusão .....	29
8. Referências .....	30

## Índice de Figuras

Figura 1- Representação gráfica onde pode observar que uma das amostras se encontra bastante distanciada das demais, sendo por isso considerada outlier. ....	7
Figura 2- Exemplo de um control chart .....	7
Figura 3- O primeiro gráfico trata-se da forma de onda de um determinado som. O segundo gráfico apresenta-nos a taxa de zero-crossing desse mesmo som. ....	9
Figura 4- $N-1$ - ordem do filtro FIR de Wiener .....	13
Figura 5- Equação de Wiener-Hopf .....	14
Figura 6- Exemplo de aplicação do Filtro de Wiener .....	14
Figura 7- - Signal to noise ratio (SNR) .....	16
Figura 8- Esquematização da seleção de um segmento de fala ou ruído, a partir do tamanho da janela, do fator e do step, no algoritmo a desenvolver. ....	16
Figura 9- Código em Matlab para a criação do Som .....	20
Figura 15- Áudio com SNR de 0 dB .....	23
Figura 16- Áudio com SNR de 5 dB .....	24
Figura 17- Áudio com SNR de 10 dB .....	24
Figura 18- Áudio com SNR de 15 dB .....	25
Figura 19- Áudio com SNR de 20 dB .....	25
Figura 20- Áudio com SNR de 25 dB .....	26
Figura 21- Áudio com SNR de 30 dB .....	26
Figura 22- Áudio com SNR de 35 dB .....	27
Figura 23- Áudio com SNR de 40 dB .....	27



## 1 Introdução

Este trabalho foi realizado no âmbito da unidade curricular de Processamento Digital de Sinal, sob a orientação do professor Carlos Lima com o intuito de cobrir parcialmente a segunda parte da matéria da UC: Processamento de Sinal Estocástico. Propõe-se para este propósito a execução em Matlab de um módulo que permita fazer a deteção automática de pausas na fala.

No mundo digital é muito comum a gravação de áudio, nomeadamente da nossa fala. No entanto, num segmento de voz gravada existem segmentos de silêncio, que mesmo não contendo informação linguística relevante, contêm dados (de ruído branco) que será também enviado. Tal informação é desnecessária de ser processada e enviada, pelo que se concebeu um algoritmo que permite analisar todo um segmento de voz gravada e localizar os segmentos de silêncio para posterior eliminação desse conteúdo “ruidoso”.

Para se conseguir tal algoritmo, baseou-se nos gráficos de controlo de Shewhart, aplicando-se limites (threshold) variáveis para que se possam adaptar às condições do ruído.

Inicialmente efetuou-se um estudo sobre o melhor threshold a aplicar para cada relação sinal-ruído (SNR) diferente. Testando com valores de SNR a escalar de 0 dB até 40 dB com intervalos de 5 dB e aplicando então valores de threshold entre os 0.2 e 16, para se obter uma tabela que permite decidir qual o melhor threshold a utilizar consoante diferentes valores de SNR.

A partir desta tabela é então possível que o algoritmo consiga, através do cálculo automático do SNR, escolher o valor de threshold mais adequado para proceder à deteção e remoção de segmentos de silêncio na fala.

## 2 Fundamentos Teóricos

### 2.1 SNR e Ruído Branco

O SNR (*Signal to Noise Ratio*) é uma medida que compara os níveis de um sinal com o do ruído de fundo que este possa conter. Este pode ser definido em decibéis, em que 0 dB indica que o nível do ruído se equipara (sobrepõe-se) ao do sinal e quanto maior for, menor influência o ruído tem no sinal analisado. O SNR é definido como sendo:

$$SNR = \frac{P_{sinal}}{P_{ruído}} = \left( \frac{A_{sinal}}{A_{ruído}} \right)^2$$

Onde  $P$  é a potência média e  $A$  é a amplitude eficaz dos sinais.

Para o cálculo em dB, poder-se-á usar a mesma expressão aplicando-se a base dos decibéis, isto é:

$$SNR_{dB} = 10 \log_{10} \left[ \left( \frac{A_{sinal}}{A_{ruído}} \right)^2 \right]$$

Foi com base nesta expressão que se chegou ao cálculo da potência desejada (neste caso da amplitude desejada) do sinal de ruído gerado para testes (“Signal-to-noise ratio,” n.d.).

As características deste ruído são as típicas de um ruído branco, que não é mais que um sinal aleatório com igual intensidade em todas as frequências do espectro, traduzindo-se numa densidade espectral de potência constante (Mancini, Carter, & Texas Instruments Incorporated., 2009). Em tempo amostrado, este ruído, que será gerado com uma amplitude que cumpra o SNR desejado, será constituído por um vetor de pontos aleatórios com distribuição normal e comprimento igual ao de amostras do sinal.

## 2.2 Processos Estocásticos

Um processo estocástico é utilizado para analisar sinais em que não é possível determinar o seu conteúdo previamente. Como é possível saber o comportamento que apresentará, a partir daí pode-se descrever os sinais estocásticos através de um modelo probabilístico.

Podemos então considerar sinais estocásticos como sinais aleatórios, cujas amostras são independentes de todas as outras, pelo que sempre que repetimos as experiências e registamos o seu output, os valores das sucessivas amostras serão de um modo geral diferentes das amostras obtidas nas experiências anteriores.

No caso do ruído é possível saber os limites a partir dos quais um segmento do sinal é efetivamente ruído ou contém informação relevante, pois este apresenta as seguintes características:

- Distribuição normal de media igual a zero
- Variância constante
- Covariância nula
- Simetria
- É não correlacionado entre diferentes estantes.

## 2.3 Outlier

Em estatística, o **outlier (valor atípico)** acontece quando um determinado valor se encontra muito afastado em relação aos restantes. Sendo que a sua existência, implica, geralmente, dificuldades na interpretação dos resultados estatísticos aplicados às amostras.

Um dos métodos de identificação de outliers mais utilizado é o desvio-padrão. Sendo que neste método, será considerado outlier um determinado valor que se encontre a uma certa quantidade de desvios-padrões da média. Essa quantidade deverá variar consoante o tamanho da amostra.

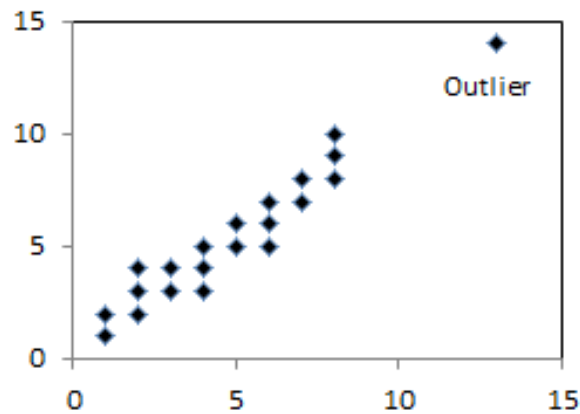


Figura 1- Representação gráfica onde pode observar que uma das amostras se encontra bastante distanciada das demais, sendo por isso considerada outlier.

## 2.4 Controlo de Shewhart e Distância Normalizada

Walter A. Shewhart desenvolveu os denominados *control charts*, que permitem verificar se um processo se encontra sob controlo ou não, através da estatística que os seus dados apresentem. Para se chegar a esses *control charts* são necessárias amostras, a média dessas amostras ( $\mu$ ), que indica o centro do gráfico, e o desvio padrão dessas amostras ( $\sigma$ ), que é depois usado para definir os limites de controlo, fixos por Shewhart em três vezes o desvio padrão calculado ( $3\sigma$ ) a partir da linha central do gráfico (Tague, 2004). Caso uma amostra ultrapassasse qualquer um desses limites, o processo seria considerado como não controlado. (“Control Charts - CQE Academy,” n.d.)

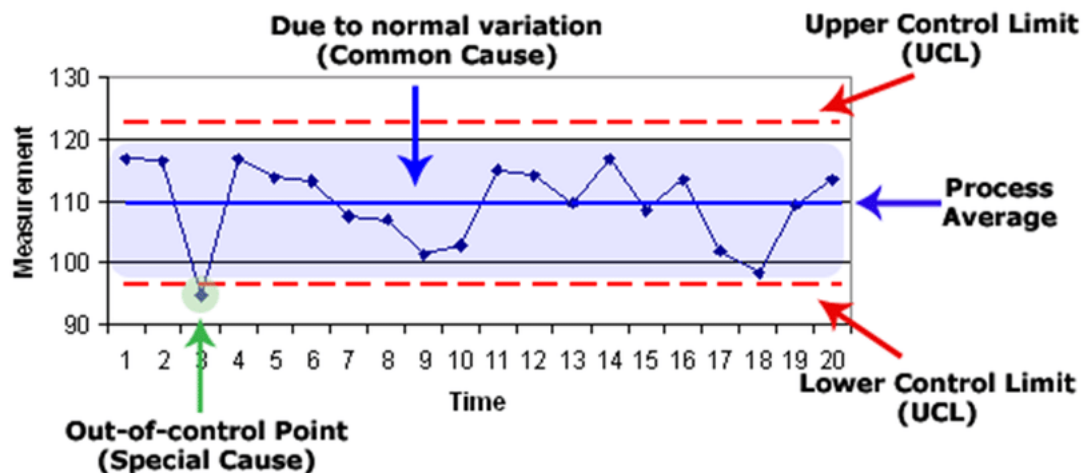


Figura 2- Exemplo de um control chart

Uma maneira intuitiva de verificar se uma amostra deve pertencer ou não a um certo conjunto onde surja inserida é através da distância normalizada, isto é:

$$\frac{|x - \mu|}{\sigma}$$

Sendo  $x$  o valor dessa amostra. Desta maneira entende-se que quanto mais perto o ponto estiver da média das amostras tendo em conta a dimensão do conjunto (desvio padrão), maior a probabilidade de essa amostra pertencer ao conjunto em estudo (Mahalanobis, 1936). É necessário então definir o limite que justifica a pertença ou não do ponto (*threshold*).

## 2.5 Zero-crossing

O detetor de cruzamento de zero procura lugares no Gaussiano de uma imagem em que o valor do Gaussiano passa pelo zero - isto é, pontos onde o Gaussiano muda de sinal. Esses pontos geralmente ocorrem em margens nas imagens - ou seja, pontos em que a intensidade da imagem muda rapidamente, mas também ocorrem em locais que não são tão fáceis de associar a margens. É melhor pensar no detetor de cruzamento zero como algum tipo de detetor de recurso, e não como um detetor de margem específico. Os zeros que se encontram em contornos fechados e, portanto, a saída do detetor da passagem de zero é geralmente uma imagem binária com linhas de espessura de pixel único mostrando as posições dos pontos de passagem de zero.

O núcleo do detetor de cruzamento zero é o filtro Gaussiano. Como descrito, as margens das imagens dão origem a zero cruzamentos na saída. Por exemplo, a Figura 3 mostra a resposta de um filtro a uma margem da etapa na imagem.



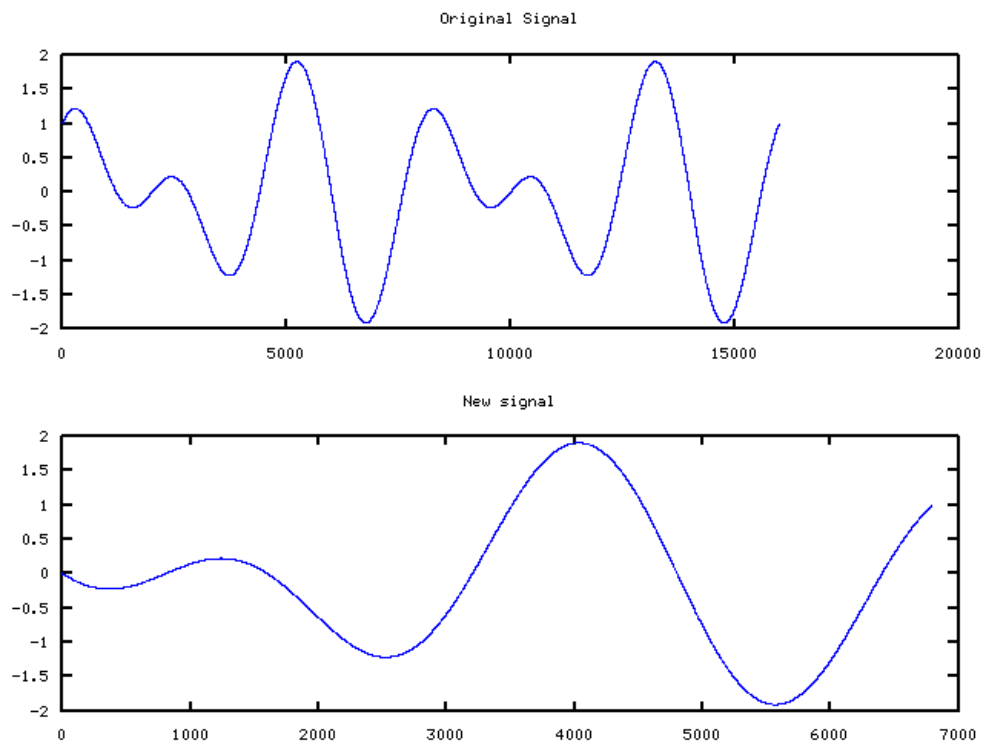


Figura 3- O primeiro gráfico trata-se da forma de onda de um determinado som. O segundo gráfico apresenta-nos a taxa de zero-crossing desse mesmo som.

No entanto, cruzamentos zero também ocorrem em qualquer lugar onde o gradiente de intensidade da imagem começa a aumentar ou começa a diminuir, e isso pode acontecer em locais que não são obviamente margens. Frequentemente, cruzamentos zero são encontrados em regiões de gradiente muito baixo, onde o gradiente de intensidade oscila para cima e para baixo em torno de zero.

## 2.6 Filtro de Wiener

No processamento de sinal, o **filtro de Wiener** é um filtro utilizado para produzir uma estimativa de um desejado alvo ou processo aleatório por (linear e invariante no tempo LTI) de filtragem de um processo de ruído observada, assumindo conhecido estacionário sinal e espectros de ruído, e ruído aditivo. O filtro de Wiener minimiza o erro quadrático médio entre o processo aleatório estimado e o processo desejado.

O objetivo do filtro de Wiener é a calcular uma estimativa estatística de um sinal desconhecido, usando um sinal relacionado como uma entrada e de filtragem que conhecido de sinais para produzir a estimativa como uma saída. Por exemplo, o sinal conhecido pode consistir de um sinal desconhecido de interesse que tem sido corrompido por aditivo de ruído. O filtro de Wiener pode ser usado para filtrar o ruído do sinal corrompido para fornecer uma estimativa do sinal subjacente de interesse. O filtro de Wiener é baseado numa estatística abordagem, e um relato mais estatística da teoria é dado no estimador mínimo de erro médio quadrático (MMSE) artigo.

Filtros determinísticos típicos são concebidos para uma desejada resposta de frequência. No entanto, o projeto do filtro Wiener tem uma abordagem diferente. Um é suposto para ter conhecimento das propriedades espectrais do sinal original e o ruído, e se procura o tempo linear invariante filtro cuja saída viria tão perto do sinal original possível. Wiener filtros são caracterizados pelo seguinte:

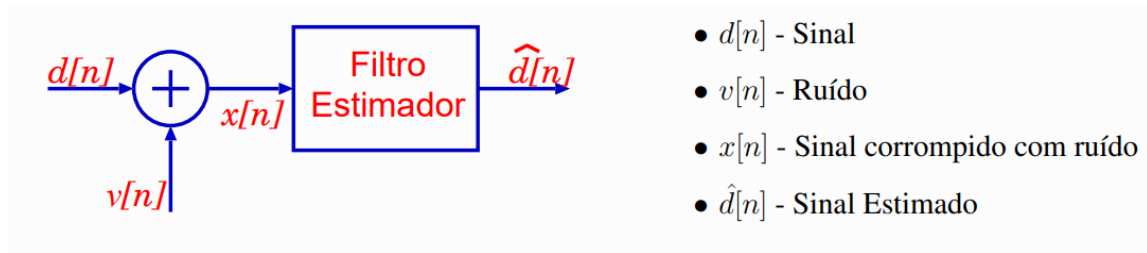
Pressuposto: sinal e (aditivo) de ruído são lineares estacionários processos estocásticos com características espectrais conhecidas ou conhecido autocorrelação e a correlação cruzada

Requisito: o filtro deve ser fisicamente realizável / causal (este requisito pode ser descartado, resultando em uma solução não-causal)

Critério de desempenho: erro médio quadrado mínimo (MMSE)

Este filtro é frequentemente utilizado no processo de desconvolução;

O problema de filtro Wiener tem soluções para três casos possíveis: um, onde um filtro não causal é aceitável (que necessitam de uma quantidade infinita de ambos dados passados e futuros), o caso em que um causal filtro é desejada (usando uma quantidade infinita de dados passado), e o finito resposta ao impulso caso (FIR), onde apenas os dados de entrada é utilizado (isto é. o resultado ou de saída não é alimentada de volta ao filtro como no caso IIR). O primeiro caso é simples de resolver, mas não é adequado para aplicações em tempo real. Principal realização do Wiener foi resolver o caso em que a exigência de causalidade está em vigor, e em um apêndice do livro de Wiener Levinson deu a solução FIR.



### Problemas a considerar:

- **Filtragem** - Estimar  $\hat{d}[n]$  quando o sinal está corrompido com ruído,  $x[n] = d[n] + v[n]$  com um filtro estimador causal, ou seja, considerando o valor presente e passados de  $x[n]$ .
- **Suavização** - O mesmo problema, mas considerando todos os dados possíveis, sendo permitido que o filtro estimador não causal.
- **Predição** - Sinal é estimado em  $n + k$  (futuro), usando dados observados até  $n$ .
- **Desconvolução** - Quando  $x[n] = d[n] * g[n] + v[n]$ , em que  $g[n]$  é a resposta impulsiva de um SLIT

### Solução não causal:

$$G(s) = \frac{S_{x,s}(s)}{S_x(s)} e^{\alpha s};$$

onde são densidades espectrais. Desde que é o ideal, então o mínimo erro médio quadrático equação reduz-se a  $S_g(t)$

$$E(e^2) = R_s(0) - \int_{-\infty}^{\infty} g(\tau) R_{x,s}(\tau + \alpha) d\tau,$$

e a solução é o inverso de dois lados transformada de Laplace da  $g(t)$   $G(s)$

## Solução de causalidade:

$$G(s) = \frac{H(s)}{S_x^+(s)},$$

Onde

- $H(s)$  consiste na parte de causalidade (isto é, que parte desta fracção com um tempo solução positiva sob a transformada inversa de Laplace):

$$\frac{S_{x,s}(s)}{S_x^-(s)} e^{\alpha s}$$

- $S_x^+(s)$  o componente causal da (isto é, a transformação inversa de Laplace de é não-zero apenas para)  $S_{x(s)} S_x^+(s) t \geq 0$
- $S_x^-(s)$  é o componente anti-causal de (ou seja, a transformação inversa de Laplace de é não-zero apenas para)  $S_{x(s)} S_x^-(s) t < 0$

## Filtro FIR de Wiener

$$\text{Filtro estimador} - w[n] \xleftrightarrow{Z} W(z)$$

O filtro FIR de Wiener produz uma estimativa do erro quadrático médio mínimo do processo  $d[n]$  filtrando o processo estatisticamente relacionado  $x[n]$ .

Assume-se que os processos  $x[n]$  e  $d[n]$  são estacionários em sentido restrito com:

- autocorreção de  $x[n]$  -  $R_x[k]$
- autocorreção de  $d[n]$  -  $R_d[k]$
- correlação cruzada -  $R_{dx}[k]$

$$W(z) = \sum_{k=0}^{N-1} w[k]z^{-k}$$

Figura 4- N- 1 - ordem do filtro FIR de Wiener

A estimativa do filtro à saída é dada pela convolução:

$$\hat{d}[n] = \sum_{l=0}^{N-1} w[l]x[n-l]$$

O filtro FIR de Wiener leva a coeficientes  $w[n]$  que minimiza o erro quadrático médio, ou seja:

$$\xi = E \left\{ |e[n]|^2 \right\} = E \left\{ |d[n] - \hat{d}[n]|^2 \right\} \Rightarrow \frac{\partial \xi}{\partial w[m]} = 0$$

A seguinte figura mostra as equações de Wiener-Hopf para o filtro FIR de Wiener:

*Equações de Winer-Hopf*  $\sum_{l=0}^{N-1} w[l]R_x[m-l] = R_{dx}[m], \quad m = 0, 1, \dots, N-1$

*Correlações*  $R_x = E \{ x[n]x^*[n-m] \}$   
 $R_{dx} = E \{ d[n]x^*[n-m] \}$

*Mínimo Erro*  $\xi_{min} = R_d(0) - \sum_{l=0}^{N-1} w[l]R_{dx}^*[l]$

**Em termos práticos onde se utiliza o filtro de Wiener?**

- Recuperação de sinais adquiridos em ambientes ruidosos.
- Enriquecimento da qualidade de imagem.

- Restauração de gravações antigas.
- Tratamento de sinais eletromiográficos.

Consideramos o ruído e o sinal não correlacionados. Logo  $R_{dv}[k] = E \{d[n]v^*[n - k]\} = 0$

Então:

$$R_{dx}[k] = E \{d[n]x^*[n - k]\} = E \{d[n]d^*[n - k]\} + E \{d[n]v^*[n - k]\} = R_d[k]$$

$$R_x[k] = E \{x[n+k]x^*[n]\} = E \{[d[n+k] + v[n+k]] [d[n] + v[n]]^*\} = R_d[k] + R_v[k]$$

Figura 5- Equação de Wiener-Hopf

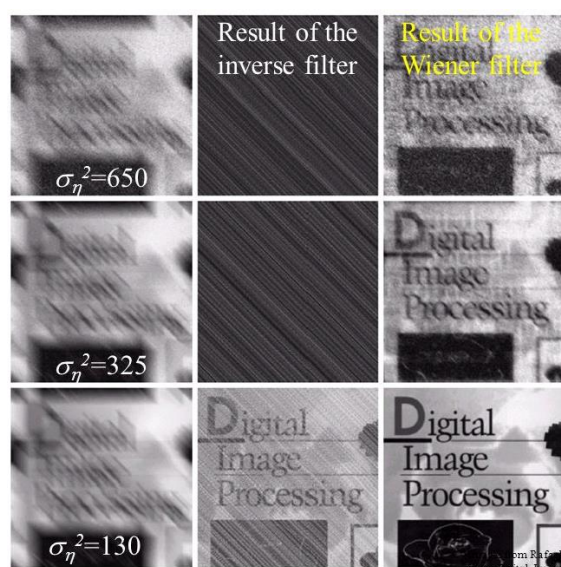


Figura 6- Exemplo de aplicação do Filtro de Wiener

### 3. Análise do Problema

Após leitura do enunciado proposto, ficou clarificado que o objetivo deste trabalho seria o desenvolvimento de um algoritmo capaz de detetar um ou vários intervalos de silêncio (end-points) num ficheiro áudio que resultou da gravação da fala humana. Estes intervalos de silêncio não possuem quaisquer informação linguística de carácter relevante, de modo que devem ser minimizados ou até eliminados do sinal.

O mesmo algoritmo deverá ainda ser capaz de detetar e analisar sons de elevada amplitude durante um curto período de tempo, como por exemplo um objeto a embater no chão ou mesmo um estalar de dedos, tendo em conta de que estes não contêm qualquer informação útil.

O algoritmo desenvolvido pode ser dividido nas seguintes etapas:

**Etapas 1:**

Gravação de um sinal áudio contendo a fala humana com recurso ao microfone do Matlab e armazenamento do mesmo num ficheiro com a extensão .wav;

**Etapas 2:**

Contaminação deste mesmo sinal áudio com ruído branco, ou seja, com média nula e potência associada ao valor de SNR (signal to noise ratio) e a soma do sinal de ruído branco criado ao sinal de áudio destinado a testes;

**Etapas 3:**

Criação de um algoritmo que permite o teste de vários valores de *threshold* para SNRs diferentes e resultado da relação entre SNR e *threshold*;

**Etapas 4:**

Adição ao algoritmo do resultado anterior obtido, de forma a automatizar o processo de adaptabilidade ao sinal de ruído com base no SNR.

O SNR (signal to noise ratio) consiste na razão da potência de um sinal e a potência do ruído sobreposto ao respetivo sinal. Esta razão é normalmente expressa em dB através da seguinte fórmula:

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \left( \frac{P_{\text{signal}}}{P_{\text{ruído}}} \right).$$

Figura 7 - - Signal to noise ratio (SNR)

O algoritmo de deteção de intervalos de silêncio numa fala começa por distinguir os segmentos de fala dos segmentos de ruído através da distribuição gaussiana. Inicialmente, é calculado o desvio padrão de cada amostra, e, se a seguinte relação se verificar:  $|X - \mu| > \sigma$  em que  $\sigma$  é o *outlier*, podemos aferir que se trata de um segmento de fala. Caso esta relação não se verifique, trata-se de um segmento com ruído. A partir da análise de cada amostra será armazenado num array o valor 1 caso seja um segmento de fala. Caso contrário, se o valor for 0, estamos perante um segmento de ruído.

A análise anteriormente referida deve ser registada num vetor para de seguida ser processada por uma parte do algoritmo que assenta na distinção entre segmentos de fala e de ruído. Esta distinção depende da parametrização do tamanho da janela, o fator de proporção entre valores de fala e de ruído na mesma janela de valores e o avanço do índice. Este método ainda tem a capacidade de eliminar segmentos de elevada amplitude num curto espaço de tempo que não contenham informação útil.

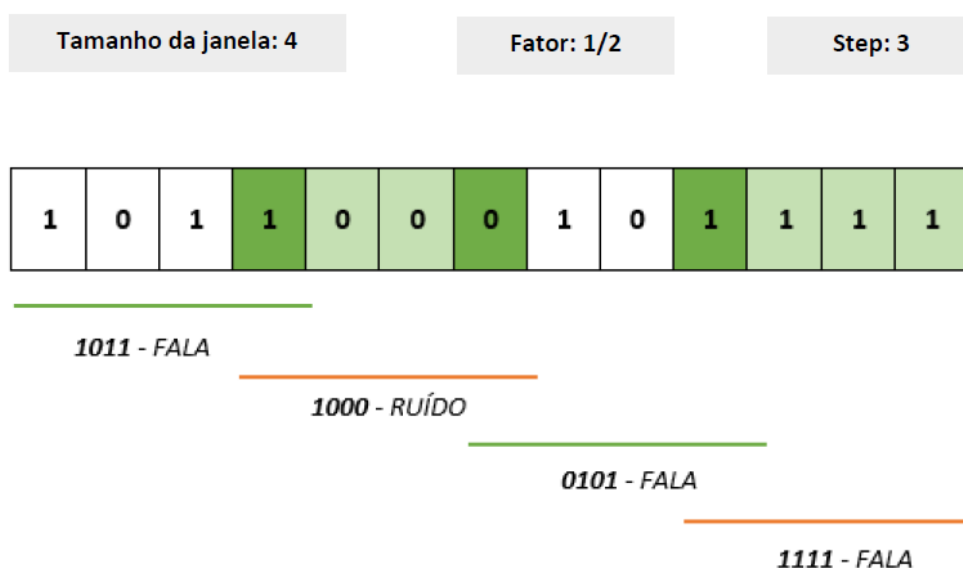


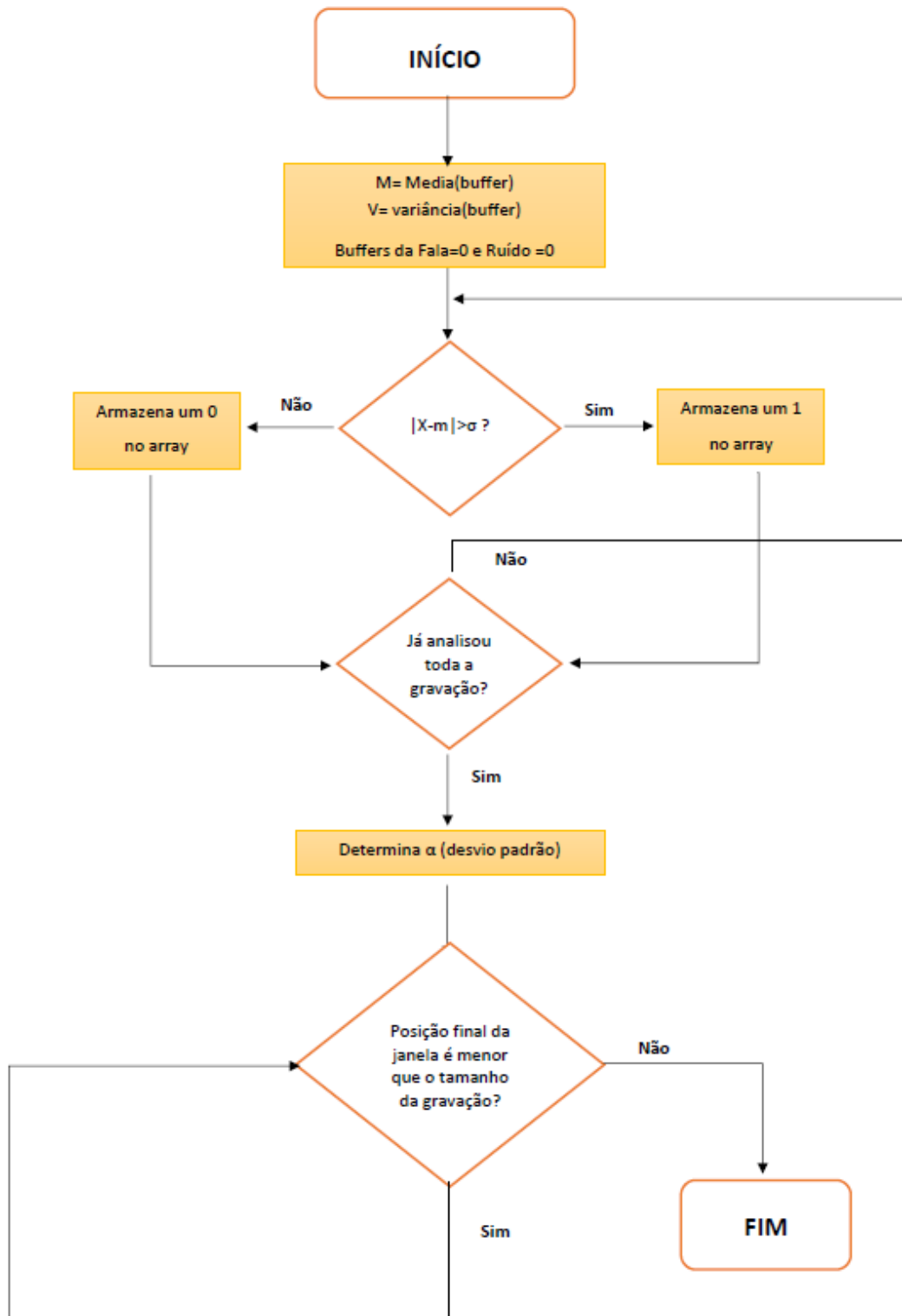
Figura 8- Esquematização da seleção de um segmento de fala ou ruído, a partir do tamanho da janela, do fator e do step, no algoritmo a desenvolver.

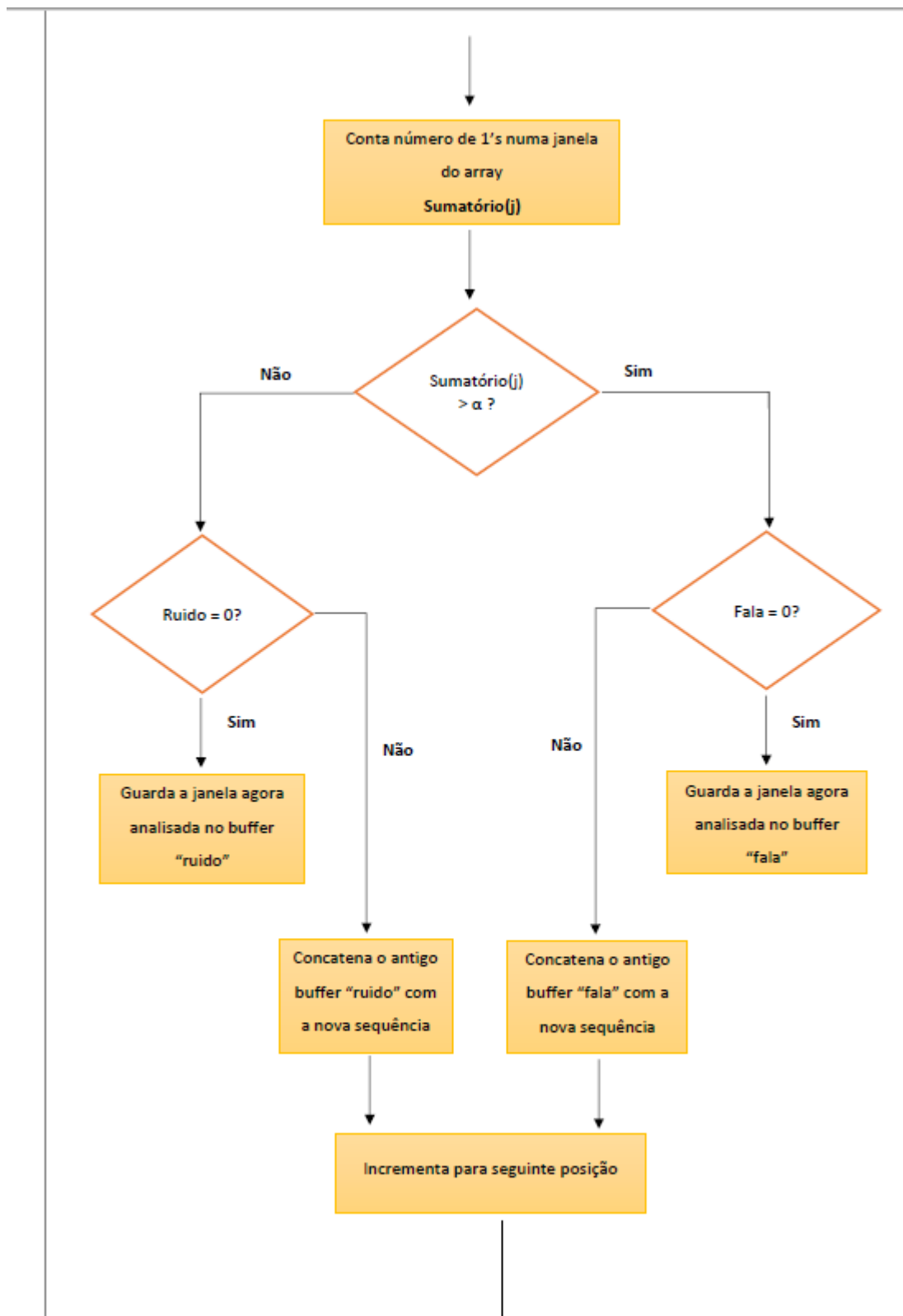




Logo, devido ao fator aplicado, serão armazenadas no buffer “fala”, apenas as sequências com 50% das amostras consideradas “fala” (isto é, com um valor de 1’s igual ou superior ao desejado), sendo que todas as restantes sequências guardadas no buffer correspondente ao ruído. Será, então, possível observar individualmente o segmento de ruído e o segmento de fala, permitindo assim perceber a fiabilidade do algoritmo desenvolvido, bem como as situações em que este apresenta suscetibilidades.

## 4. Fluxograma





## 5. Implementação do código

De forma a cumprir o especificado pelo problema realizaram-se os seguintes passos:

### 5.1 Gravação do Áudio

No início do programa foram executados os seguintes comandos de forma a gravar o áudio como se pode verificar na figura abaixo. Como resultado foi criado um ficheiro do tipo .wav, ficheiro este que consiste num sinal áudio de 5 segundos, o qual é transformada num array para posteriormente realizar o estudo do sinal.

```
%frequencia de amostragem
Fs=8000;

%Criação do Som
gravacao=audiorecorder; %criar o Objecto
disp('Inicio da Gravação.')
recordblocking(gravacao,5); %gravação do audio por 5 segundos
disp('Fim da Gravação.')
play(gravacao); %reprodução do Objecto(gravação)
y=getaudiodata(gravacao); %conversão do Objecto da gravação para um Array
audiowrite('som.wav',y,8000); %escrita de um ficheiro .wav com o Array
[y,Fs]=audioread('som.wav'); %leitura do ficheiro .wav
```

Figura 9- Código em Matlab para a criação do Som

### 5.2 Tamanho da Janela

Definição do tamanho da janela, fração de 1's por janela, para considerar fala ou ruído e número de segmentos de dados por segundo.

```
tamanho_janela = 150; %tamanho da janela a analisar
fator = 0.5; %fracção de 1s
```

### 5.3 Valor SNR

Definição do valor SNR em dB para o ruído branco.

```
asnr = awgn(y,20,'measured'); % atribuição dos valores db do ruído branco de 0dB-40dB
```

## 5.4 Média e Variância

Cálculo da média e da variância do sinal original.

```
media = mean(asnr(1:n_seg));           %calcula da media
variancia = var(asnr(1:n_seg));        %calcula da variancia
```

## 5.5 Matriz para guardar a informação

Alocação e preenchimento (percorrendo todo o vetor asnr) de uma matriz para guardar a informação se, a cada ponto, é ruído ou fala.

```
for a=1:length(asnr)                    %separa os sinais da fala, do ruído e guarda posições em aux
    if abs(asnr(a)-media)>sqrt(variancia)
        aux(a) = 1;                     %se for fala fica a 1 na posicao i
    else
        aux(a) = 0;                     %se for ruído fica a 0 na posicao i
    end
end
```

## 5.6 Declaração/Definição de Variáveis

1. Declaração e inicialização dos vetores de voz e ruído.
2. Definição do valor de 1's presentes na janela para ser considerado fala.
3. Definição e atribuição de valores das variáveis de controlo para gestão das janelas.

```
buffer_ruído = 0;                       % iniciar buffer ruído
buffer_fala = 0;                         % iniciar buffer fala
aux = 1:1:length(asnr);                 %cria matriz que guarda se é ruído ou fala
alpha = tamanho_janela * fator;         %fator de definição para verificar se é uma seq. de fala (com tantos 1's é fala)
```

## 5.7 Vetores de ruído ou fala

Preenchimento dos vetores de ruído ou fala, dependendo do número de 1's presentes na janela, comparado com o valor de  $\alpha$ .

```
início_janela = 1; %define o início da janela
fim_janela = tamanho_janela;

while((fim_janela < length(asnr))) %caso não tenha chegado ao fim de toda a gravação audio

    j = aux(início_janela:fim_janela); %guarda a seq. correspondente a uma janela

    if sum(j) > alpha %caso número de 1 for maior que o alpha - fala
        if ~buffer_fala %se ainda não houver o array da fala
            buffer_fala = asnr(início_janela:fim_janela);
        else %se os dois arrays que contêm fala num só
            buffer_fala = cat(1,buffer_fala,asnr(início_janela:fim_janela));
        end
    else %caso número de 1 for menor que o alpha - ruído
        if ~buffer_ruído %se ainda não houve o array ruído
            buffer_ruído = asnr(início_janela:fim_janela);
        else %se os dois arrays que contêm ruído num só
            buffer_ruído = cat(1,buffer_ruído,asnr(início_janela:fim_janela));
        end
    end

    início_janela = fim_janela;
    fim_janela = fim_janela + tamanho_janela;
end
```

## 5.8 SNR

Cálculo do signal-to-noise ratio.

```
snr = 10*log(mean(buffer_fala.^2)/mean(buffer_ruído(1:(Fs/5)).^2)); %cálculo do SNR
```

## 6. Testes e Resultados da Implementação

Para os diversos testes apresentados abaixo, foi gravado um som com a frase "Processamento Digital Sinal".

Para cada um dos testes é apresentada uma janela composta por três gráficos, gerada pelo matlab, que representa o sinal original, o ruído e o sinal com a informação vocal (para cada potência de sinal de ruído branco).

### 6.1 Geração de ruído branco – 0 dB

Uma vez que para este teste o SNR é de 0 dB, existe um efeito considerável do ruído no sinal original. À medida que o SNR for aumentando, o efeito do ruído sobre o sinal irá ser cada vez mais reduzido.

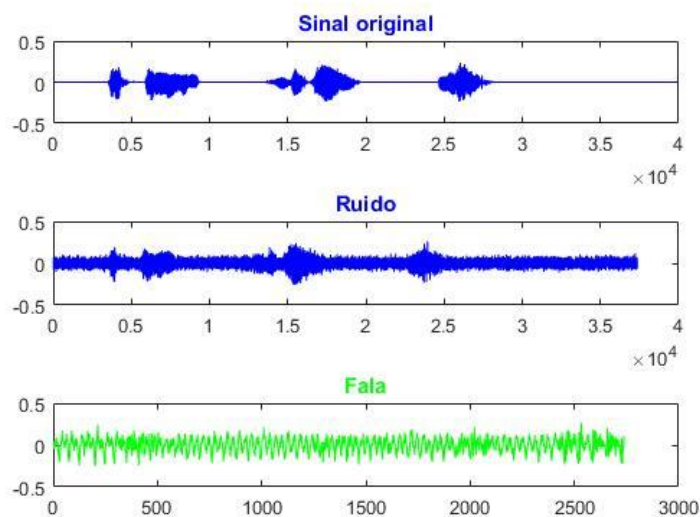


Figura 10- Áudio com SNR de 0 dB

## 6.2 Geração de ruído branco – 5 dB

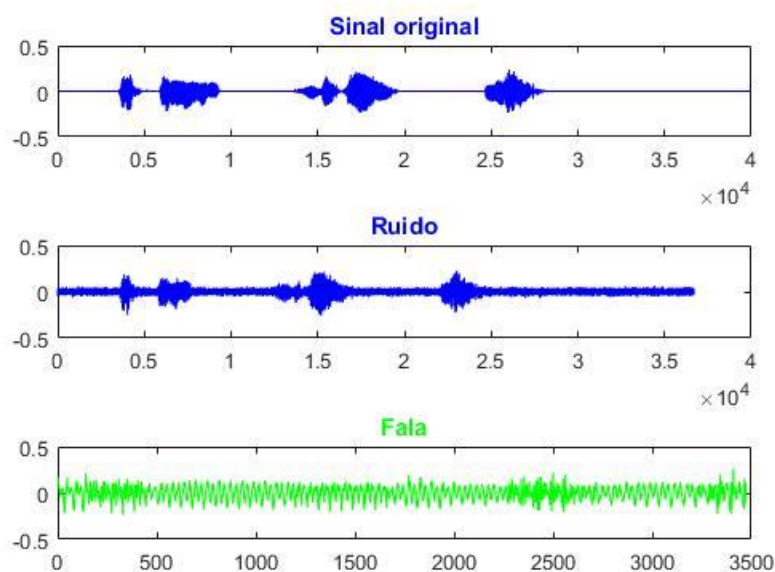


Figura 11- Áudio com SNR de 5 dB

## 6.3 Geração de ruído branco – 10 dB

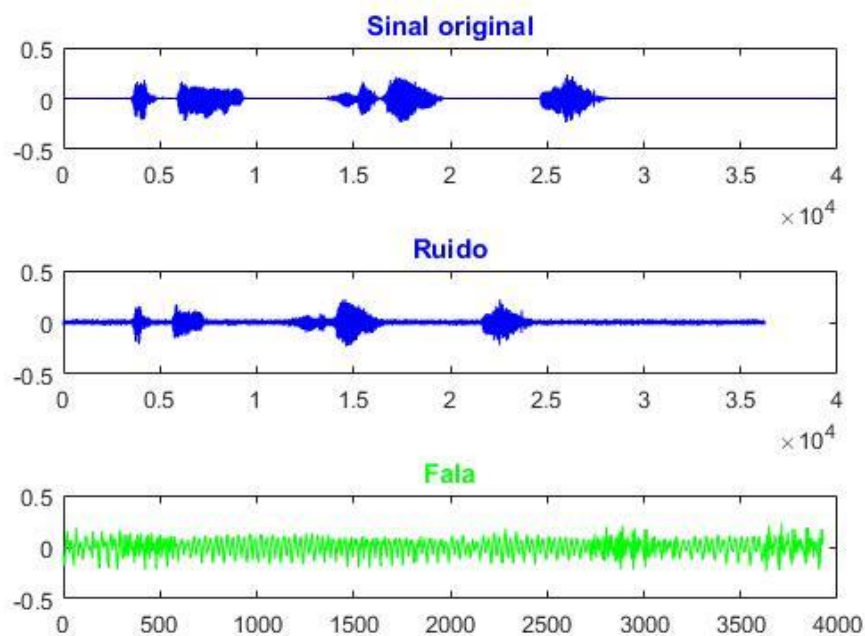


Figura 12- Áudio com SNR de 10 dB



#### 6.4 Geração de ruído branco – 15 dB

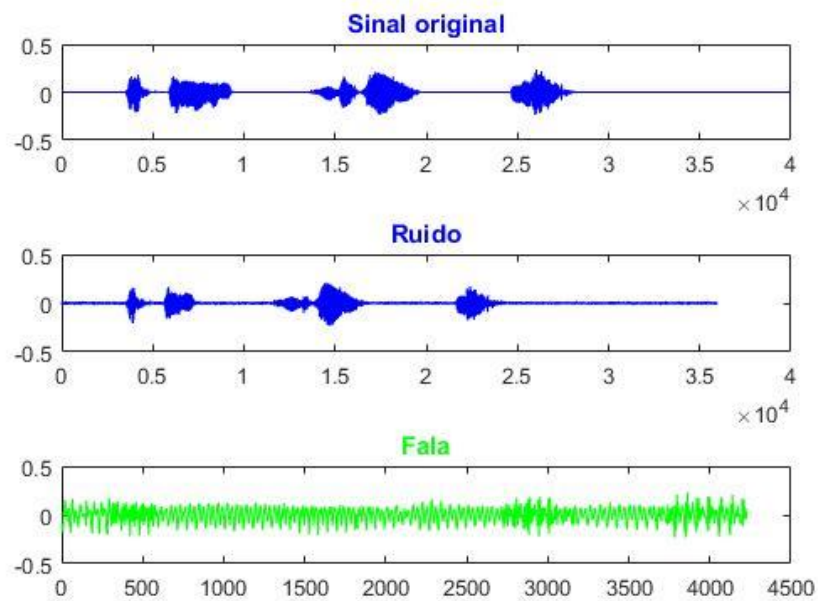


Figura 13- Áudio com SNR de 15 dB

#### 6.5 Geração de ruído branco – 20 dB

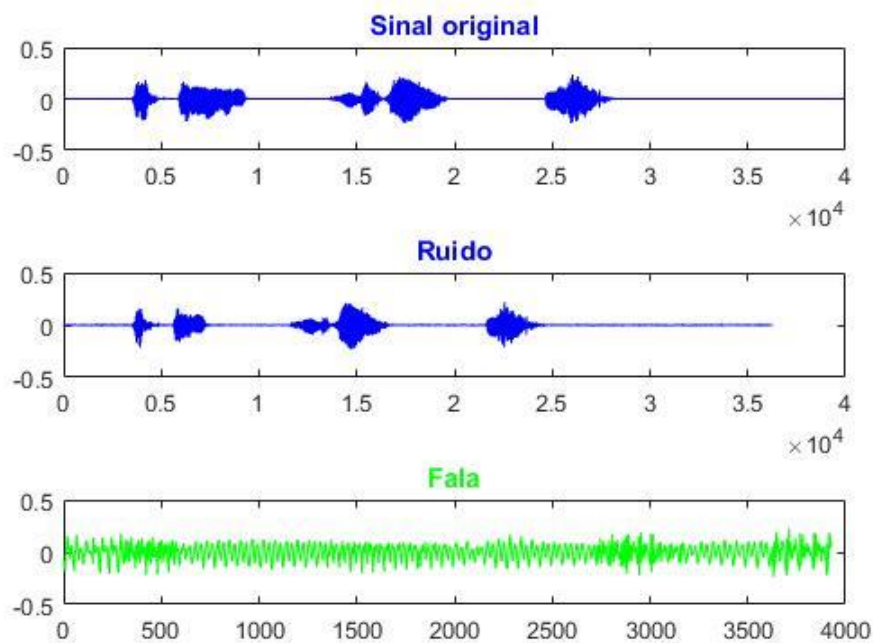


Figura 14- Áudio com SNR de 20 dB

## 6.6 Geração de ruído branco – 25 dB

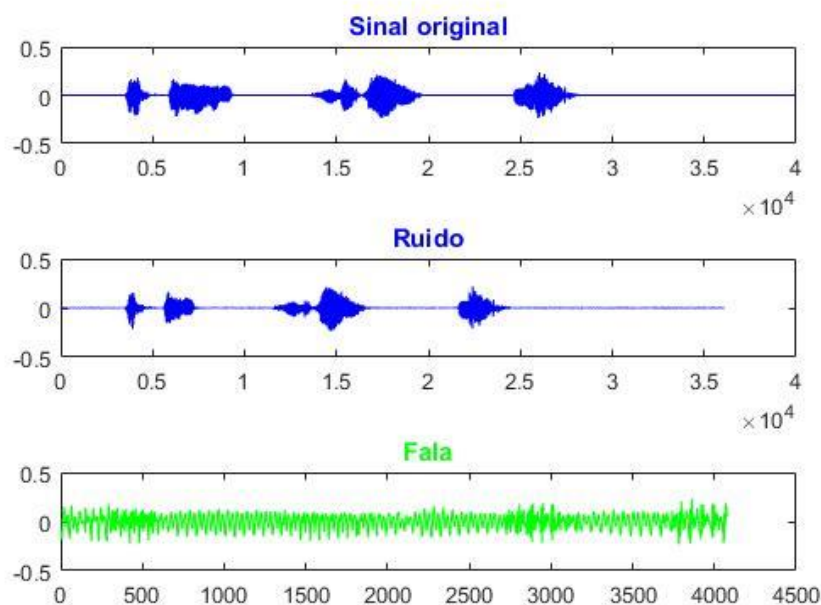


Figura 15- Áudio com SNR de 25 dB

## 6.7 Geração de ruído branco – 30 dB

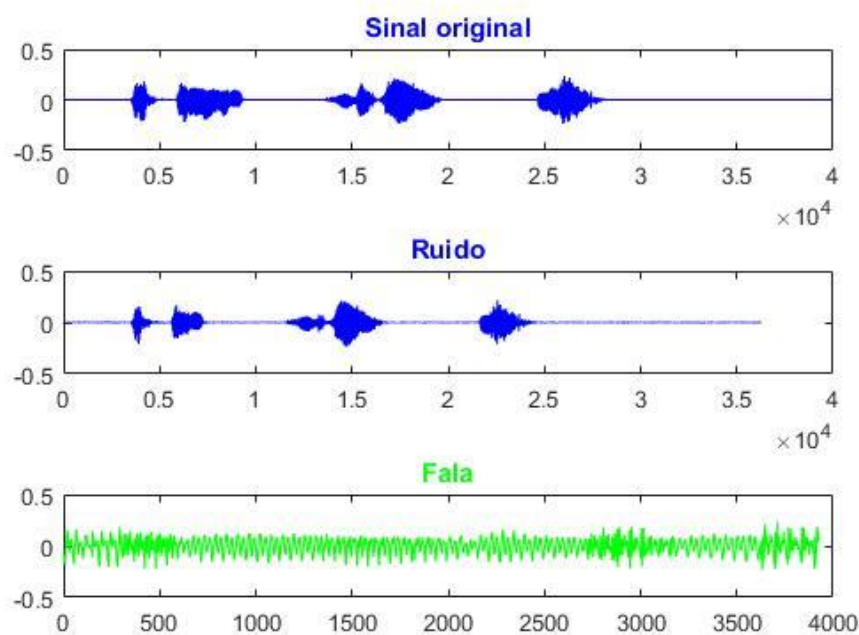


Figura 16- Áudio com SNR de 30 dB

## 6.8 Geração de ruído branco – 35 dB

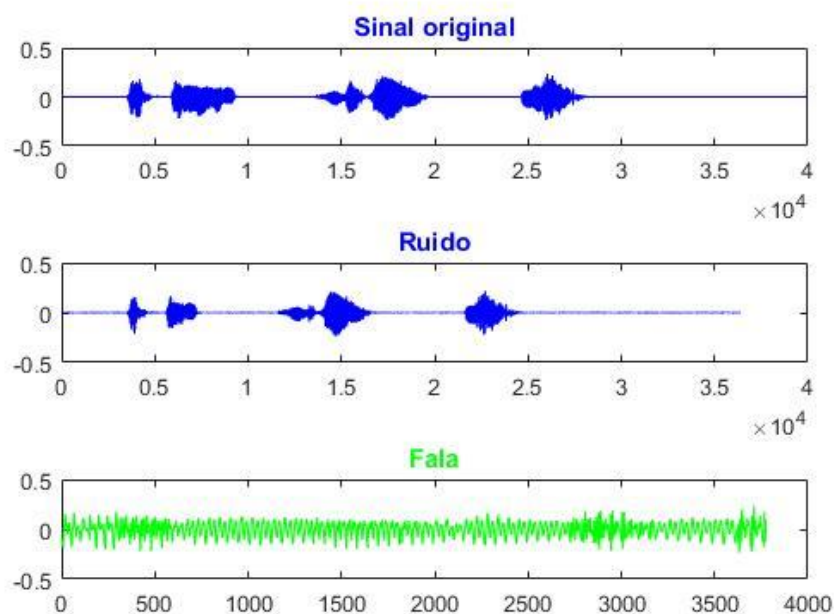


Figura 17- Áudio com SNR de 35 dB

## 6.9 Geração de ruído branco – 40 dB

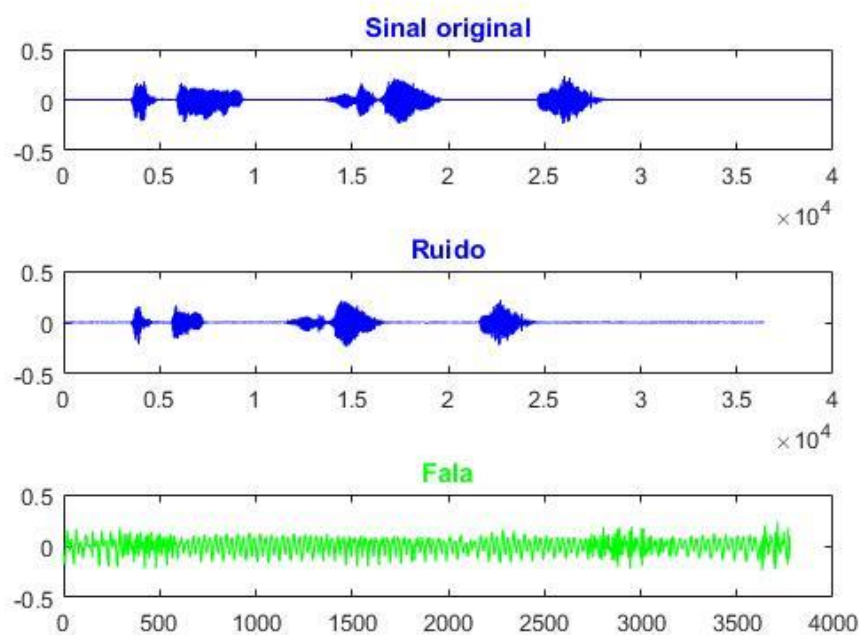


Figura 18- Áudio com SNR de 40 dB



A partir dos 40 dB, a influência do ruído é tão mínima, que já nem se nota diferenças com o sinal original, mais uma vez isso coloca o gráfico do sinal sem os silêncios o mais parecido possível aos restantes já obtidos, concluindo-se que para tal o melhor valor seria de 8 e sendo o referido gráfico o seguinte:

SNR (dB)	Threshold
0	0.75
10	0.85
20	1.5
30	2.8
SNR > 40	8

## 7. Conclusão

Com este trabalho foi possível verificar como um algoritmo de deteção de silêncios pode envolver um pouco de tudo, desde o cálculo do SNR até à fabricação de ruído, para que se consiga ajustar da melhor maneira o algoritmo, envolvendo componentes estatística e cálculo de probabilidades, de uma maneira intuitiva e simplificada, baseando-se inclusive em métodos de controlo como o de Shewhart e adaptando-se às implicações do problema.

Conclui-se que quanto maior o SNR, maior o threshold que se aplica no algoritmo, pois a influência do ruído no sinal passa a ser menor e não é necessário aplicar limites tão restritos na diferenciação de ruído e sinal.

De referir que o algoritmo usado não consegue detetar ruídos isolados que surjam descontextualizados, apenas serve para remoção do chamado background noise que esteja presente desde o início da gravação (uma vez que baseia a sua amostra de ruído nos primeiros pontos da gravação).

Em suma, foi possível verificar com a realização deste trabalho todos os conceitos aprendidos em contexto de sala de aula, sendo que, com a utilização do software Matlab, os conteúdos ficaram mais esclarecidos.



## 8. Referências

- Aulas e apontamentos de Processamento digital de sinal ano letivo 2019/2020;
- Control Charts - CQE Academy. (n.d.). Retrieved June 13, 2018, from <http://www.cqeacademy.com/cqe-body-of-knowledge/continuous-improvement/quality-control-tools/control-charts/>
- GT's Blog: Silence Removal and End Point Detection MATLAB Code. (n.d.). Retrieved June 11, 2018, from <https://ganeshtiwaridotcomdotnp.blogspot.com/2011/08/silence-removal-and-end-point-detection.html?m=1>
- Mahalanobis, P. C. (1936). On the generalised distância in statistics. *Proceedings of the National Institute of Sciences of India*, 2(1), 49–55.
- Mancini, R., Carter, B., & Texas Instruments Incorporated. (2009). *Op amps for everyone*. Newnes/Elsevier.
- Signal-to-noise ratio. (n.d.). *Scholarpedia.Org*. Retrieved from [http://www.scholarpedia.org/article/Signal-to-noise\\_ratio](http://www.scholarpedia.org/article/Signal-to-noise_ratio)
- Tague, N. R. (2004). Seven Basic Quality Tools. *The Quality Toolbox*, 15. Retrieved from <http://www.asq.org/learn-about-quality/seven-basic-quality-tools/overview/overview.html>