Universidade do Minho - Escola de Engenharia

Mestrado Integrado em Engenharia Eletrónica Industrial e Computadores

Algoritmo para a deteção de silêncios na fala

Processamento Digital de Sinal

2014/2015

Discente: Sara Filipa Martins de Sá 68523

Docente: Professor Carlos Lima

Índice

Introdução

Este projeto foi realizado no âmbito da unidade curricular de Processamento Digital de Sinal, sob a orientação do professor Carlos Lima com o intuito de permitir a deteção de silêncios na fala, usando o modelo gaussiano para modelação do background e o conceito de outlier para deteção da fala.

A dinâmica da fala será incorporada no modelo do ruído para não descartar amostras de sinal de baixa amplitude que ocorrem na vizinhança dos zero-crossing.

Assim, para melhor compreensão do trabalho realizado, este relatório conterá, num primeiro plano, a explicação e consolidação de alguns conceitos teóricos, bem como uma análise do problema, seguido da sua implementação e um conjunto de demonstrações do seu funcionamento.

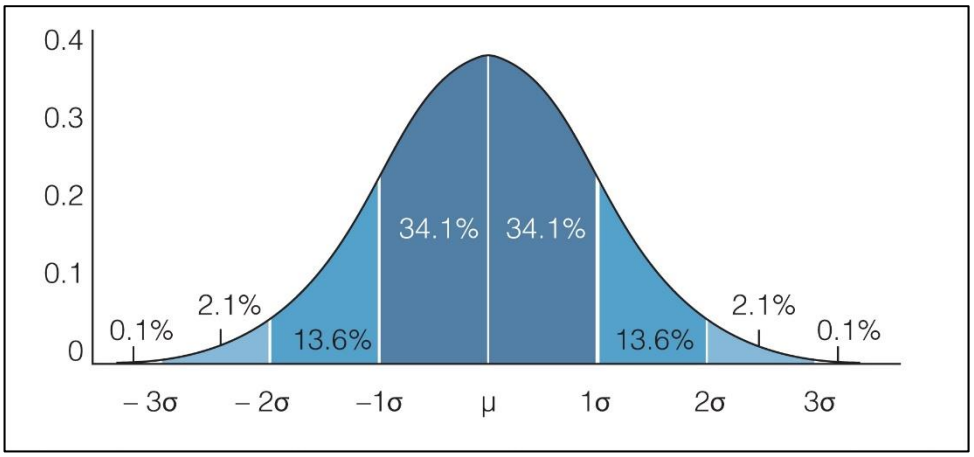
Conceitos teóricos fundamentais

- **Modelo gaussiano**

Uma distribuição normal ou gaussiana consiste num conjunto significativo de amostras de eventos aleatórios, cuja ocorrência individual não obedece a regras ou padrões que permitam fazer previsões acertadas, mas que, no entanto, a partir da análise do seu conjunto permitem concluir que os eventos tendem a ficar próximos de uma determinada posição, que representa a sua média matemática, μ.

Desta forma, o modelo gaussiano é inteiramente descrito pelos parâmetros: **média** (μ), valor para o qual mais se concentram os dados de uma distribuição e **desvio-padrão** (σ), que representa o valor de dispersão em relação à média. Assim um baixo desvio-padrão indica que os dados se encontram próximos da média.

De salientar ainda a **variância** (σ^2), quadrado do desvio-padrão, que indica o "quão longe" os valores de encontram, em geral, do valor esperado.



$$f(x,\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}, -\infty < x < \infty, \sigma > 0.$$

Figura 1: *Representação da gráfica do integral e expressão matemática da função densidade de probabilidade de uma distribuição normal, em que μ e σ, representam, respetivamente, a média e o desvio padrão. Podendo observar-se que 68% dos valores se encontram a uma distância da média inferior a um desvio padrão.*

- **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.

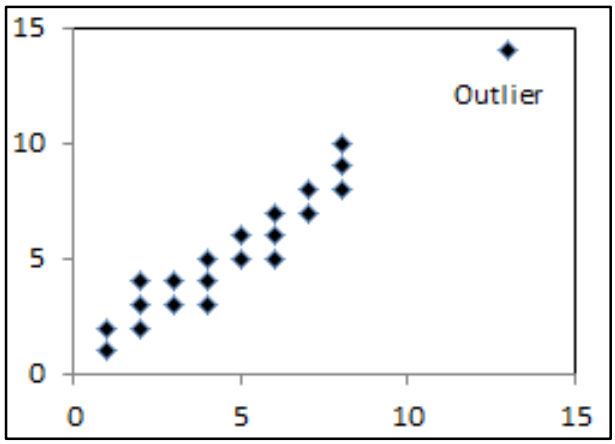


Figura 2: *Representação gráfica onde pode observar que uma das amostras se encontra bastante distanciada das demais, sendo por isso considerada outlier.*

- **Zero-crossing**

Para estimar a frequência fundamental da fala, por vezes, é utilizada a taxa de zero-crossing (ZCR), que consiste em vários pontos, nos quais ocorre uma alteração da forma de onda (de valores positivos para negativos). Estes pontos são facilmente identificados através da passagem da função no eixo das abcissas.

As principais características da taxa de zero-crossing são:

- A ZCR de sons ambientais ou sem voz é mais elevada do que a dos sons vozeados, que apresentam períodos fundamentais observáveis;
- É difícil distinguir através da ZCR sons não vozeados de sons ambientais.

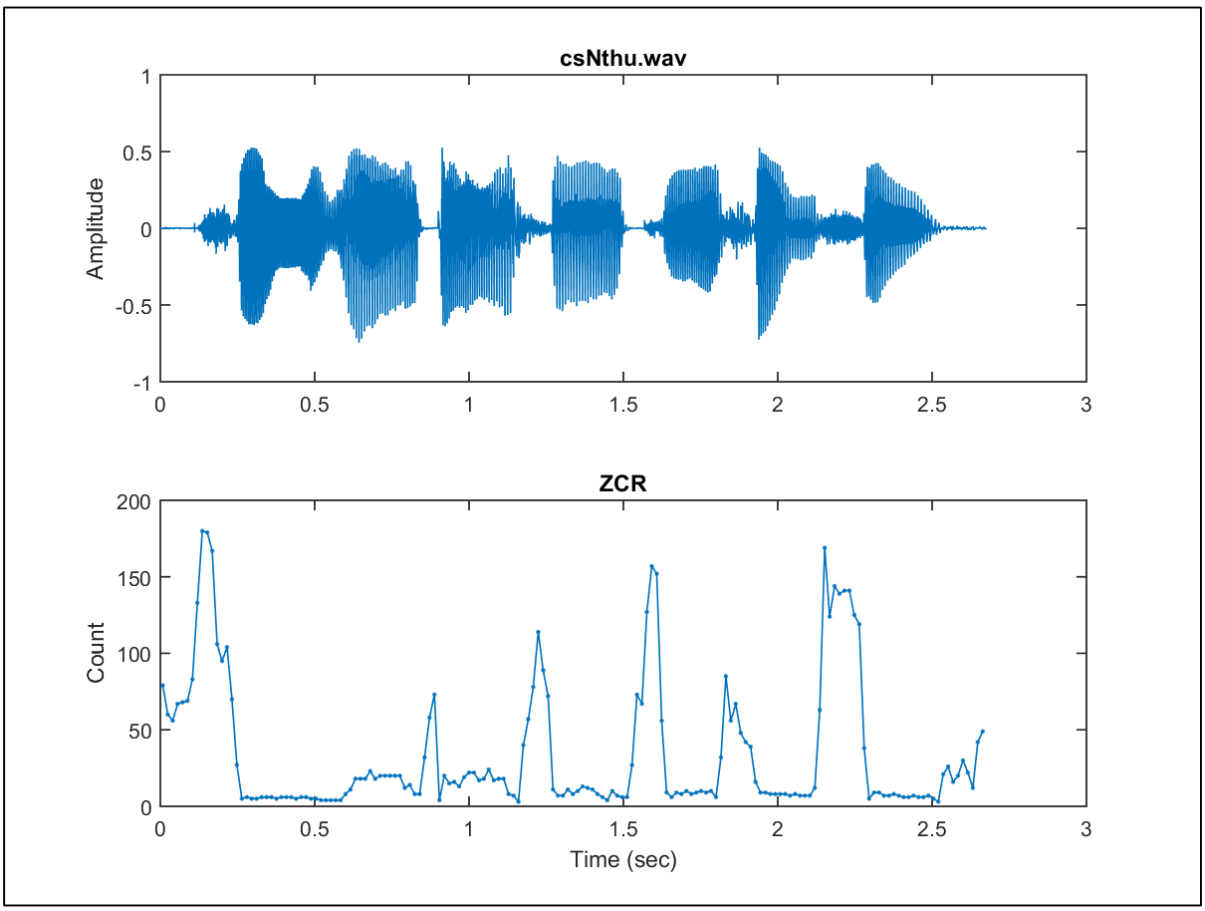


Figura 3: *O primeiro gráfico trata-se da forma de onda de um determinado som. O segundo gráfico apresenta-nos a taxa de zero-crossing desse mesmo som.*

- **Ruído Branco e Modelo do ruído**

O ruído branco é um tipo de ruído produzido pela combinação simultânea de sons de todas as frequências. Em estatística, este aplica-se a uma sequência de erros aleatórios, sempre que esta tiver média e variância constantes e não apresentar autocorrelação. O valor da média, por conveniência, será zero, podendo no entanto especificar-se um outro valor. Assim, o ruído branco será temporalmente homogéneo, estacionário e não dependente de instantes anteriores.

Através do Modelo do Ruído, pelo cálculo da média e da variância de um sinal, conseguimos retirar a componente de ruído existente ficando, assim, apenas com o sinal "limpo".

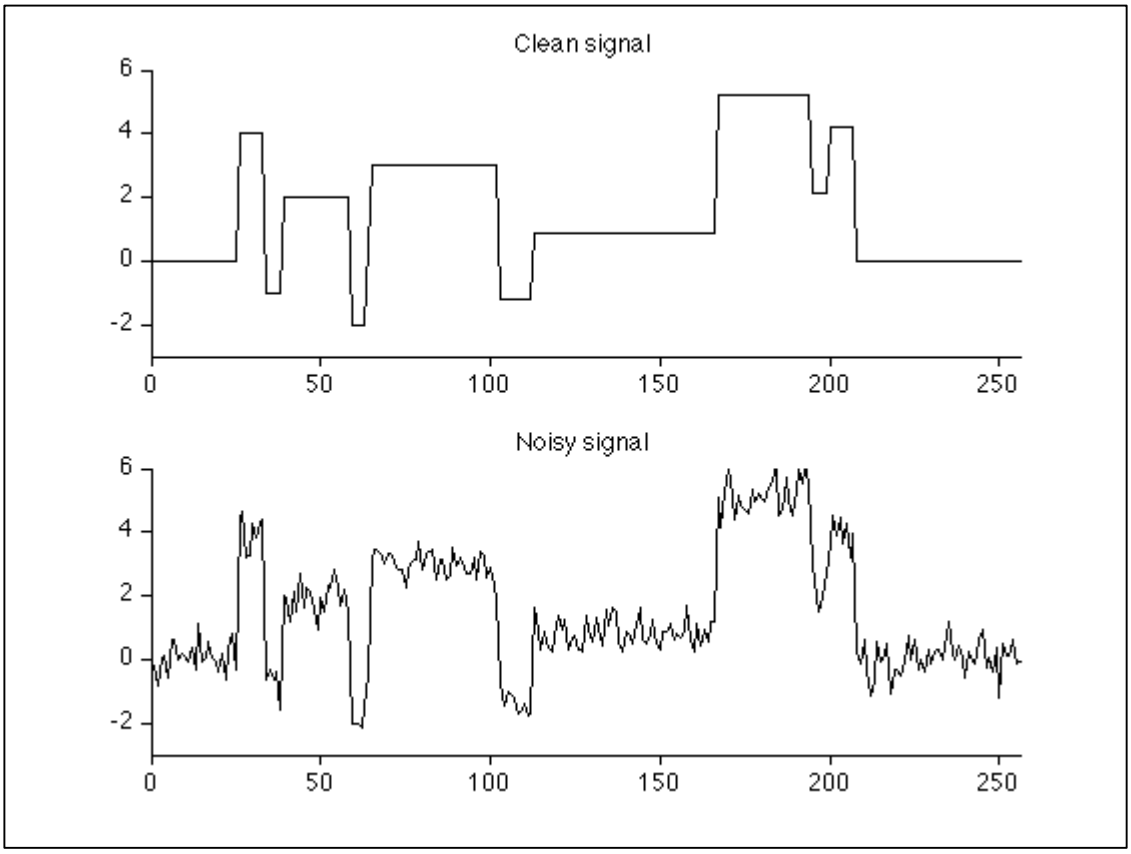


Figura 4: *No primeiro gráfico pode-se observar a forma de onda do sinal "limpo", sendo que no segundo o sinal se encontra afetado pelo ruído.*

Análise do problema proposto

A partir do enunciado proposto é necessário ter em atenção que o algoritmo concedido deve salvaguarda toda a informação contida na gravação, não podendo, assim, serem eliminadas quaisquer amostras.

Desta forma, o algoritmo a desenvolver deverá abordar o conceito de janela, o que permitirá analisar um determinado conjunto de amostras, evitando que estas sejam analisadas individualmente.

Assim, tornar-se-á necessário estabelecer quando é que uma sequência tem como correspondência um segmento de fala ou ruído. Tal será possível através do cálculo do desvio-padrão da amostra em questão, assim se **| X - μ | > σ** (outlier), estaremos perante um segmento de fala, caso contrário tratar-se-á de um segmento de ruído. A partir desta análise, será paralelamente armazenado num array **'a'** o valor 1, no caso da fala e o valor 0, no caso do ruído.

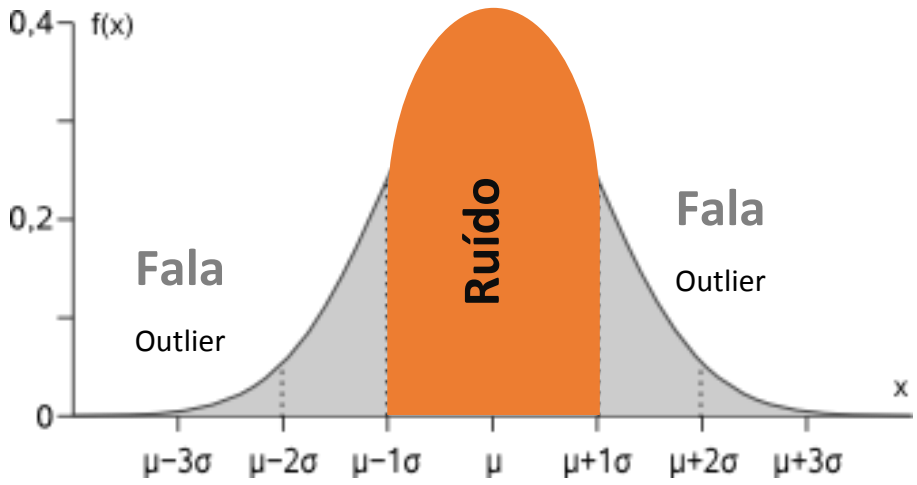


Figura 5: Distribuição normal do algoritmo a desenvolver.

Posteriormente através da contabilização do número de 1's num determinado step do array **'a'** e comparando, esse valor, com o resultado da operação matemática *tamanho_da_janela * fator*, a janela será percorrida, assim por exemplo:

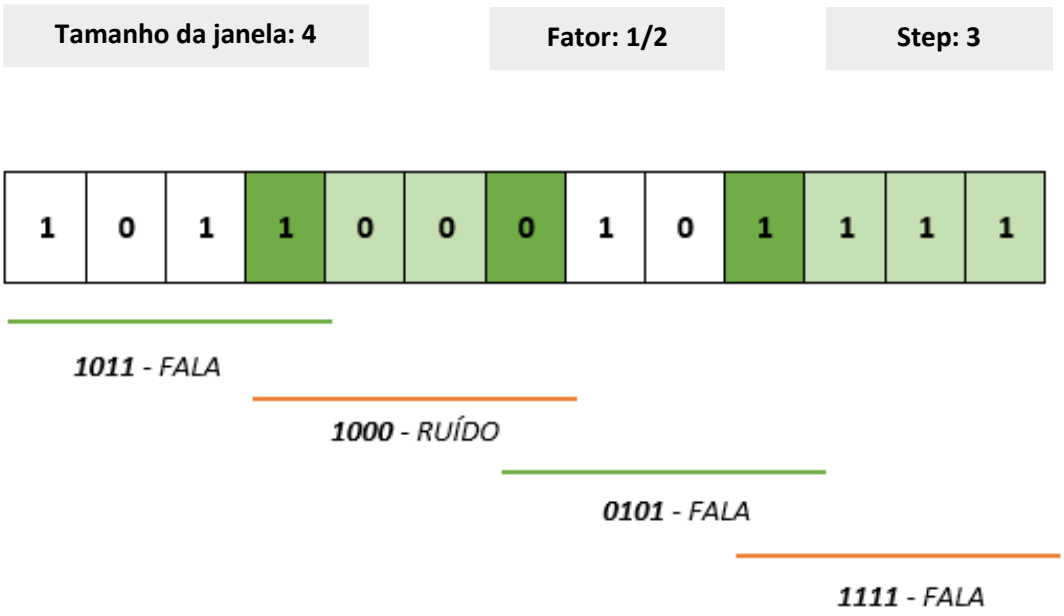
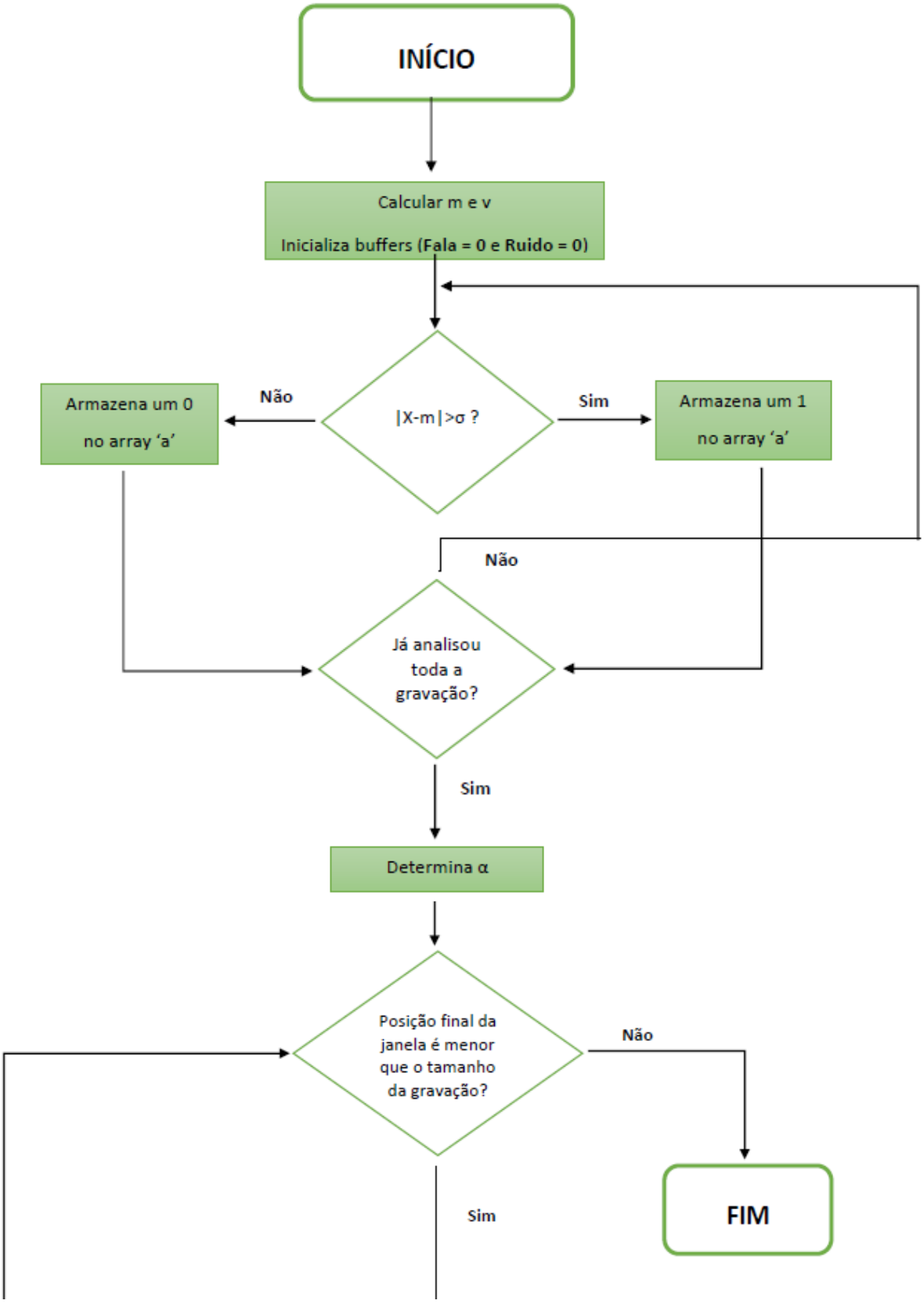


Figura 6: *Esquematização da seleção de um segmento de fala ou ruído, a partir do tamanho da janela, do fator e do step, no algoritmo a desenvolver.*

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.

- **Diagrama de Blocos**



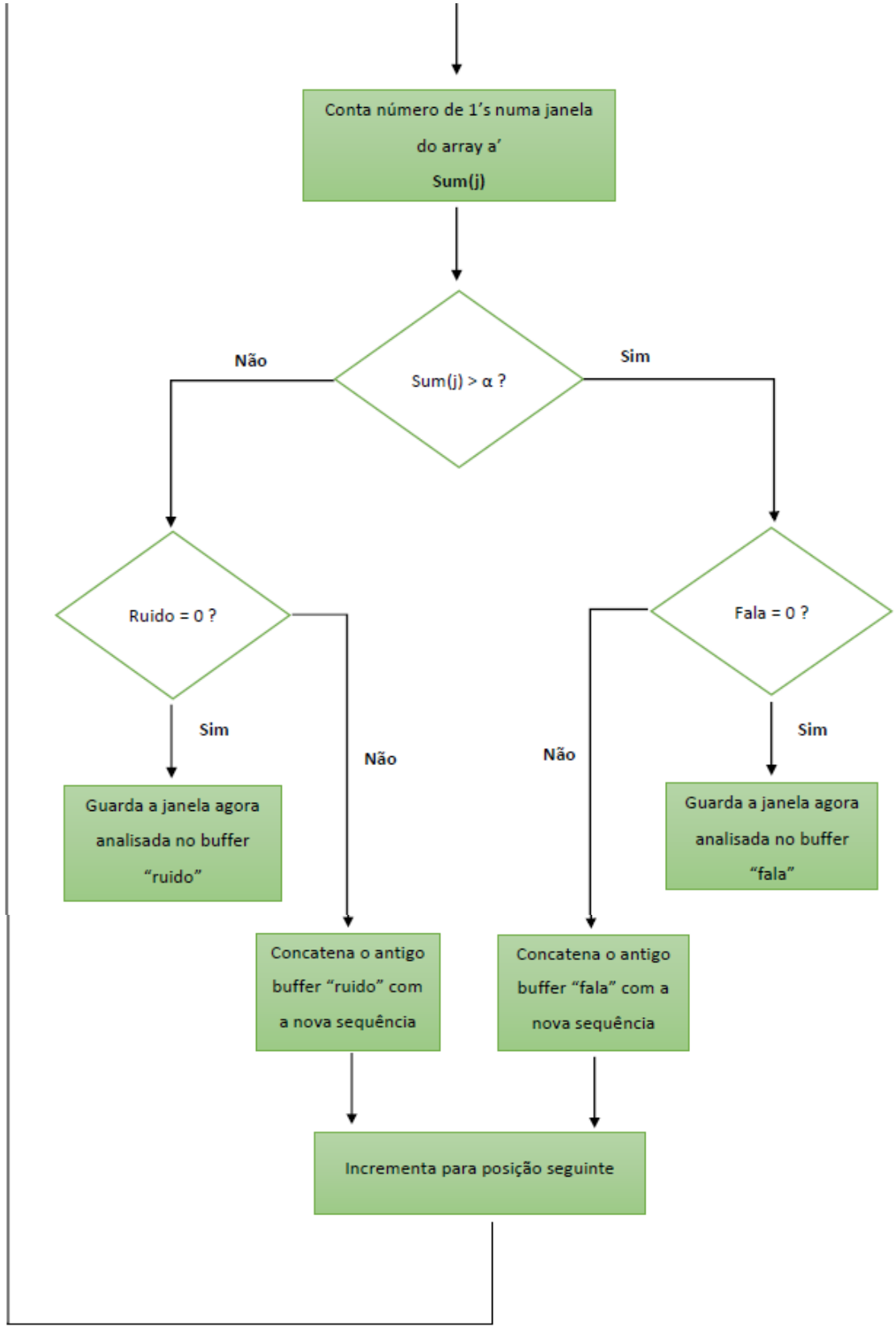


Figura 7: *Diagrama de blocos do algoritmo desenvolvido.*

Desenvolvimento em Matlab

Assim, tendo em conta a análise do enunciado anteriormente efetuada, o algoritmo desenvolvido foi o seguinte:

```
% n_seg = nº de amostras do segmento;
% fator = fracção de 1's (outlier de amostras);
% step = avanço na janela;
% janela = tamanho da janela a analisar

function [ruido, fala] = ruido_fala(x, n_seg, fator, step, tam_janela);

m = mean(x(1:n_seg));                                   %cálculo da média
v = var(x(1:n_seg));                                    %cálculo da variância
% inicialização dos buffers
ruido = 0;
fala = 0;


a = 1:1:length(x);

for i=1:length(x)                                       %sinal original
    if abs(x(i)-m)>sqrt(v)
        a(i) = 1;                                       % |X - m| > σ (outlier)
                                                        -- fala
    else
        a(i) = 0;                                       % |X - m| <= σ -- ruido
    end
end


alpha = tam_janela * fator;                             %fator de definição se é
                                                        uma sequência de
                                                        fala(com tantos 1's é
                                                        fala)
n = 1;                                                  %define o inicio da
                                                        janela


while(((n-1)*step + tam_janela) < length(x))            %se ainda não se chegou
                                                        ao fim de toda a
                                                        gravação audio

    j = a((n-1)*step + 1 : (n-1)*step + tam_janela);%armazena a sequência
                                                        correspondente a uma só
                                                        janela

    if sum(j) > alpha                                   %se o número de 1's for
                                                        superior a alpha -- fala
        if ~fala                                        %se o buffer da fala
                                                        ainda for nulo
            fala = x((n-1)*step + 1 : (n-1)*step + tam_janela);
```

```
        else                                            %une os dois buffers que
                                                        contêm fala, num só
                                                        buffer
            fala = cat(1,fala,x((n-1)*step + 1 : (n-1)*step + tam_janela));
        end


    else                                                %se o número de 1's for
                                                        inferior a alpha -
                                                        ruído
        if ~ruido                                       %se o buffer do ruido
                                                        ainda for nulo
           ruido = x((n-1)*step + 1 : (n-1)*step + tam_janela);
        else                                            %une os dois buffers que
                                                        contêm ruído, num só
                                                        buffer
           ruido = cat(1,ruido,x((n-1)*step + 1 : (n-1)*step + tam_janela));
        end
    end

        n=n+1;                                          %incrementa para a
                                                        posição seguinte
end;
```

Figura 8: *Implementação do algoritmo desenvolvido.*

- **APLICAÇÕES DO ALGORITMO DESENVOLVIDO**

- Exemplo 1

Amostra: "Processamento Digital de Sinal"

Comandos a seres executados na linha de comandos do Matlab:

```
>> x=wavrecord(4*8000,8000);
>> sound(x);
>> [ruido,fala]=ruido_fala(x,2000,2/3,400,500);
>> sound(fala);
>> sound(ruido);
>> subplot(3,1,1), plot(x);
>> subplot(3,1,2), plot(ruido);
>> subplot(3,1,3), plot(fala);
```

Resultados obtidos:

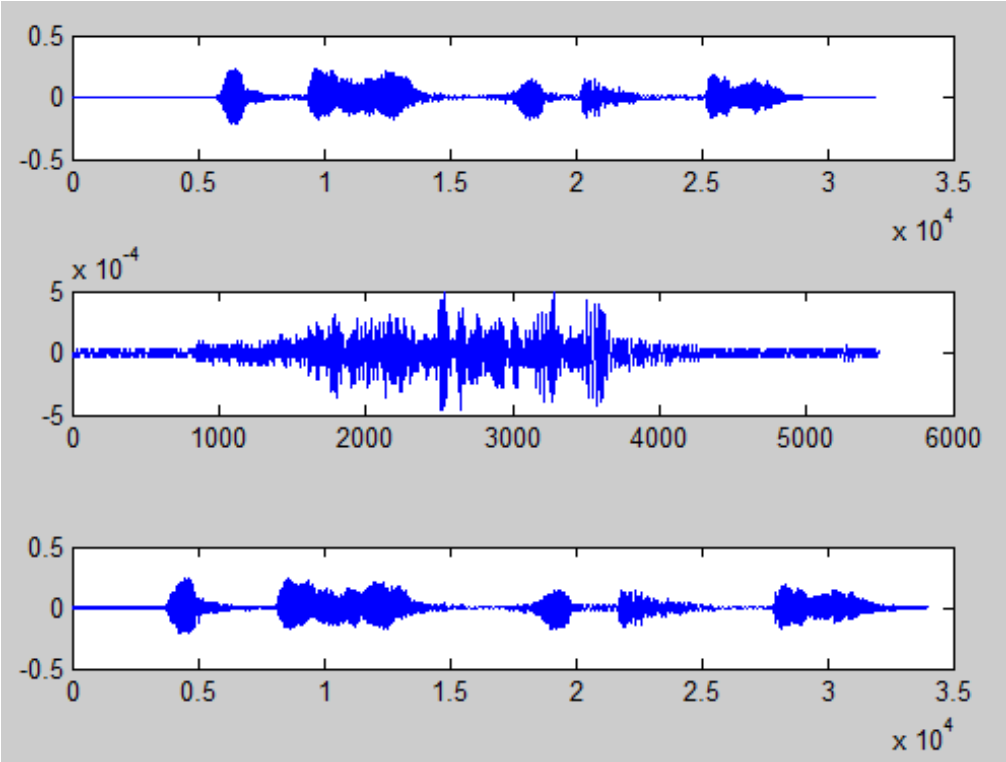


Figura 9: *Gráfico 1 – Representação gráfica da gravação efetuada. Gráfico 2 – Representação gráfica obtida no buffer " ruído". Gráfico 3 – Representação gráfica obtida no buffer " fala".*

Considerando uma janela de 500 amostras, com um avanço de 400 e um fator de 2/3, isto é, serão armazenadas no buffer "fala", apenas as sequências com 66,67% das amostras consideradas "fala" e testando o algoritmo a partir deste exemplo, em específico, foram alcançados os resultados pretendidos. No buffer "fala" apenas podemos ouvir a frase "Processamento Digital de Sinal", enquanto no buffer "ruído" não ficou armazenada qualquer informação relevante.

- Exemplo 2

Amostra: "Processamento Digital de Sinal" + objeto a cair

Comandos a seres executados na linha de comandos do Matlab:

```
>> x=wavrecord(4*8000,8000);
>> sound(x);
>> [ruido,fala]=ruido_fala(x,2000,2/3,400,500);
>> sound(fala)
>> sound(ruido);
>> subplot(3,1,1), plot(x);
>> subplot(3,1,2), plot(ruido);
>> subplot(3,1,3), plot(fala);
```

Resultados obtidos:

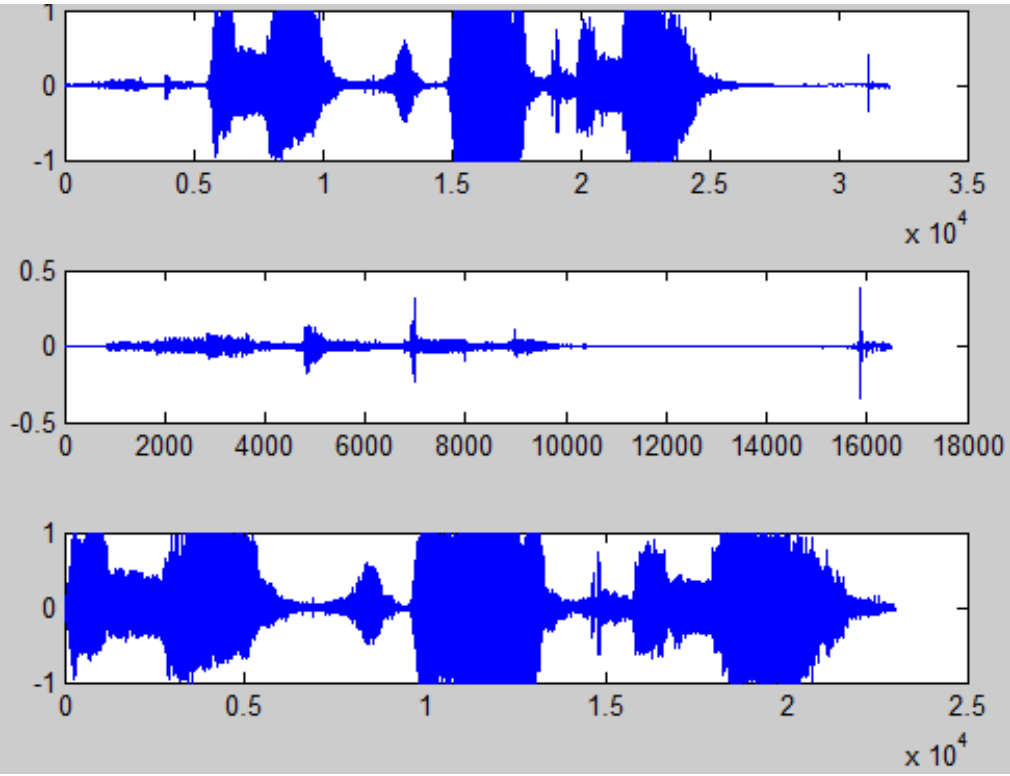


Figura 10: *Gráfico 1 – Representação gráfica da gravação efetuada. Gráfico 2 – Representação gráfica obtida no buffer " ruído". Gráfico 3 – Representação gráfica obtida no buffer " fala".*

Tendo em conta, os mesmos dados inseridos na função *ruido_sinal*, que o exemplo anterior, neste caso, é possível concluir que o algoritmo já apresenta algumas suscetibilidades. Estas estão relacionadas essencialmente com o facto de ser perdida alguma informação no caso do "P" e do "D", das palavras "Processamento" e "Digital", respetivamente. Sendo, no entanto, de realçar que o ruído provocado pelo objeto a cair é armazenado no buffer "ruido", tal como era desejado.

- Exemplo 3

Amostra: "Processamento" + pausa + "Digital de Sinal"

Comandos a seres executados na linha de comandos do Matlab:

```
>> x=wavrecord(4*8000,8000);
>> sound(x);
>> [ruido,fala]=ruido_fala(x,2000,2/3,400,500);
>> sound(fala)
>> sound(ruido);
>> subplot(3,1,1), plot(x);
>> subplot(3,1,2), plot(ruido);
>> subplot(3,1,3), plot(fala);
```

Resultados obtidos:

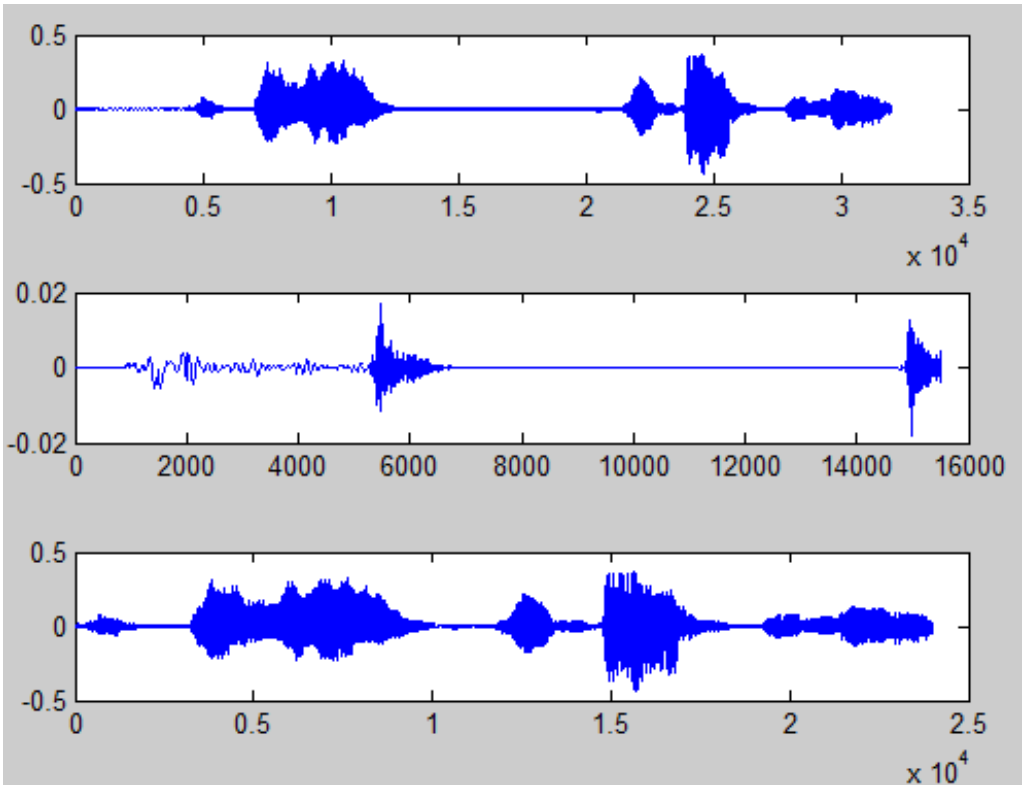


Figura 11: Gráfico 1 – Representação gráfica da gravação efetuada. Gráfico 2 – Representação gráfica obtida no buffer " ruído". Gráfico 3 – Representação gráfica obtida no buffer " fala".

Uitilizando mais uma vez, os mesmos parâmetros para função *ruido_fala*, podemos através deste exemplo concluir que, efetivamente existem algumas perdas, quando algumas letras, neste caso o "P" e o "S", são pronunciadas com uma maior intensidade. Nestes casos, o algoritmo desenvolvido, considera a sua amplitude um falso positivo, resultando assim, numa carência de informação relevante. Sendo também, essencial, realçar que a pausa entre as palavras "Processamento" e "Digital" é corretamente armazenada no buffer "ruido".

- ## Exemplo 4

Amostra: "Processamento" + pausa + "Digital" + objeto a cair + "de Sinal"

Comandos a seres executados na linha de comandos do Matlab:

```
>> x=wavrecord(6*8000,8000);
>> sound(x);
>> [ruido,fala]=ruido_fala(x,4000,3/4,200,400);
>> sound(fala)
>> sound(ruido);
>> subplot(3,1,1), plot(x);
>> subplot(3,1,2), plot(ruido);
>> subplot(3,1,3), plot(fala);
```

Resultados obtidos:

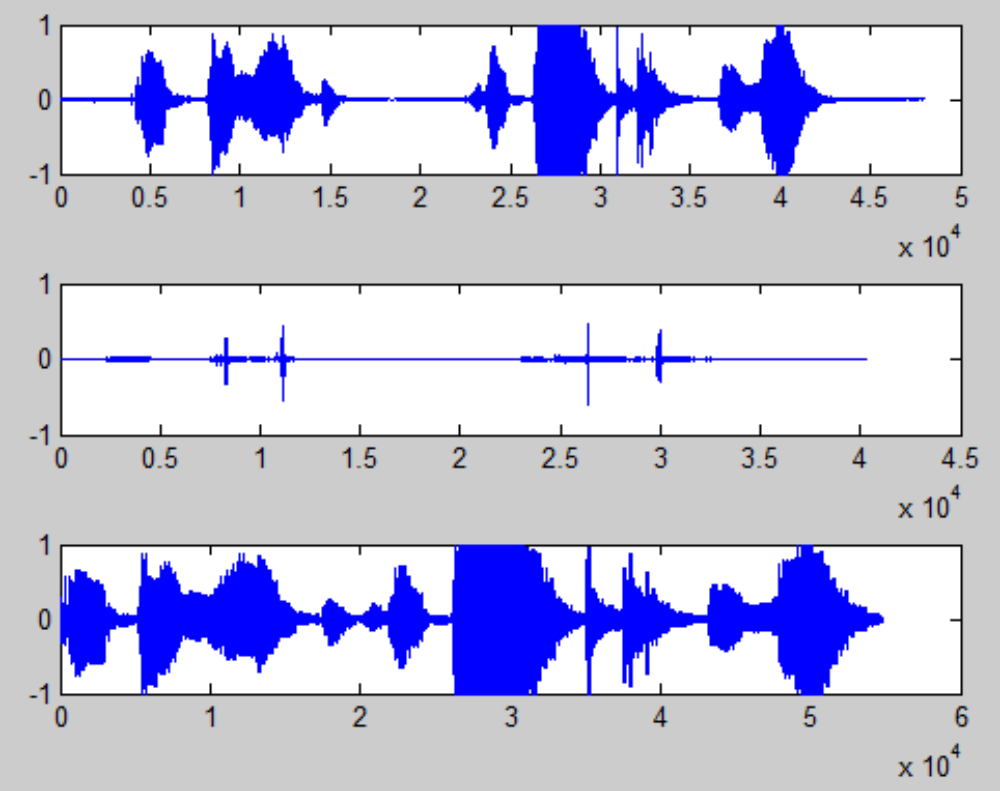


Figura 12: *Gráfico 1 – Representação gráfica da gravação efetuada. Gráfico 2 – Representação gráfica obtida no buffer " ruído". Gráfico 3 – Representação gráfica obtida no buffer " fala".*

Considerando, agora, uma janela de 400 amostras, com um avanço de 200 e um fator de 3/4, isto é, serão armazenadas no buffer "fala", apenas as sequências com 75% das amostras consideradas "fala" e testando o algoritmo a partir deste exemplo, foi possível detetar mais algumas das suas fragilidades. Neste caso, em específico, o som do objeto a cair não se encontra na sua totalidade no buffer "ruido", tendo aliás, maior parte da sua informação armazenada no buffer destinado a sons vozeados. Além do mais, é possível observar, novamente, algumas perdas de informação em algumas letras da frase pronunciada. Sendo de realçar, como facto positivo, a pausa ser armazenada corretamente no buffer "ruido".

Em relação à alteração dos parâmetros inseridos na função *ruido_fala*, pode-se observar que as formas de onda obtidas nos buffers apresentam uma maior duração, dado que o valor da janela é menor, bem como o step, sendo, por isso, necessário um maior número de ciclos para analisar toda a gravação. Sendo também notório que ao aplicar um fator mais elevado, alguns sons vozeados serão mais suscetíveis de serem considerados ruído.

- Exemplo 5

Amostra: "Processamento Digital de Sinal" + som da televisão

Comandos a seres executados na linha de comandos do Matlab:

```
>> x=wavrecord(4*8000,8000);
>> sound(x);
>> [ruido,fala]=ruido_fala(x,2000,2/3,400,500);
>> sound(fala);
>> sound(ruido);
>> subplot(3,1,1), plot(x);
>> subplot(3,1,2), plot(ruido);
>> subplot(3,1,3), plot(fala);
```

Resultados obtidos:

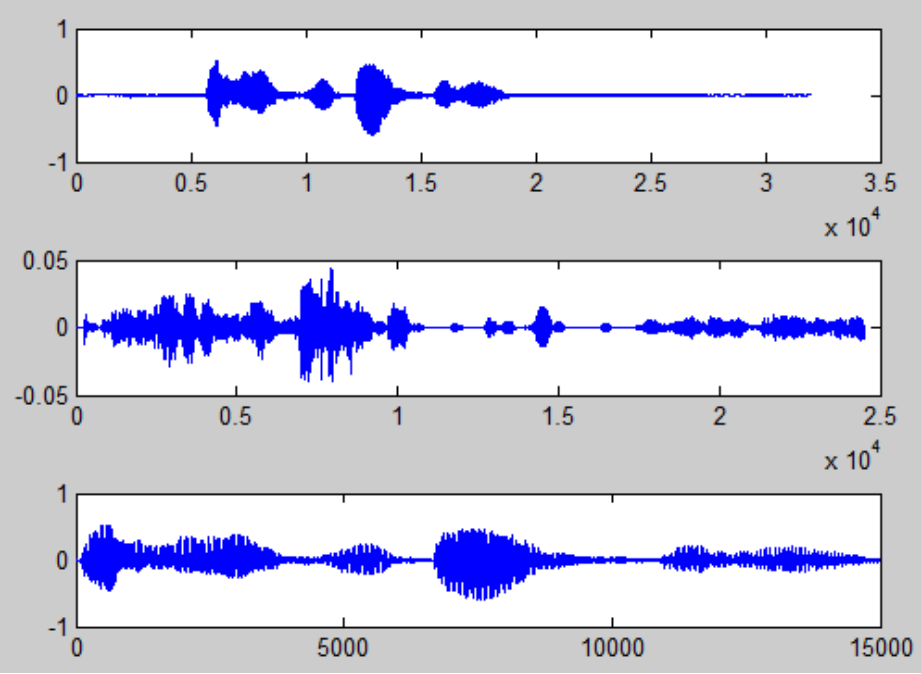


Figura 13: *Gráfico 1 – Representação gráfica da gravação efetuada. Gráfico 2 – Representação gráfica obtida no buffer " ruído". Gráfico 3 – Representação gráfica obtida no buffer " fala".*

Considerando, mais uma vez, uma janela de 500 amostras, com um avanço de 400 e um fator de 2/3, isto é, serão armazenadas no buffer "fala", apenas as sequências com 66,67% das amostras consideradas "fala", neste caso específico conseguimos os resultados pretendidos. No buffer "fala" apenas podemos ouvir a frase "Processamento Digital de Sinal", estando então armazenado no buffer "ruído" o som da televisão.

Conclusão

Através da realização deste trabalho foi possível consolidar, de uma forma geral, todos os conhecimentos, em particular da ferramenta de desenvolvimento (*Matlab*), abordados nas aulas da unidade curricular de Processamento Digital de Sinal.

Assim, após a realização de todos os testes ao algoritmo desenvolvido, pode ser observado de uma melhor forma o seu funcionamento, consoante a gravação efetuada, previamente. Podendo-se, então, retirar as seguintes conclusões:

- No caso do fator, se este for baixo, será armazenado no buffer "fala" todos os sons vozeados, no entanto também o ruído e os silêncios. Caso seja alto, poderá ocorrer a perda de alguma informação relevante;
- O step, por sua vez, influencia a duração das formas de onda dos buffers, dado que serão necessários um maior número de ciclos para analisar toda a gravação. Este deve ser conjugado corretamente com o tamanho da janela, de forma, a permitir uma melhor análise do som registado.

Bibliografia

- https://pt.wikipedia.org;
- http://stackoverflow.com;
- http://www.mathworks.com.