**Universidade do Minho – Escola de Engenharia**

**Mestrado Integrado em Engenharia de Eletrónica Industrial e Computadores**

***Redução de ruido numa gravação de voz***

Processamento Digital de Sinal

2017/2018

**Docente: Carlos Lima**

**Aluno: Carlos Manuel Lima Oliveira nº 74388**

# ***Índice***

[***Índice*** 2](#_Toc510444356)

[***Introdução*** 3](#_Toc510444357)

[***Fundamentos Teóricos*** 3](#_Toc510444358)

[***Algoritmo desenvolvido*** 3](#_Toc510444359)

[***Testes*** 6](#_Toc510444360)

[ ***1º Teste*** 6](#_Toc510444361)

[ ***2º Teste*** 7](#_Toc510444362)

[ ***3º Teste*** 8](#_Toc510444363)

[ ***4º Teste*** 9](#_Toc510444364)

[***Conclusão*** 10](#_Toc510444365)

# ***Introdução***

Este projeto foi desenvolvido no âmbito da unidade curricular de Processamento Digital de Sinal, sobre orientação do professor Carlos Lima e tem como objetivo implementar um algoritmo, em MATLAB, que permita eliminar o ruído de um sinal, através da análise do sinal de áudio original, otimizando, assim, o sinal, ficando apenas com a parte que interessa, a fala.

Num primeiro plano, será efetuado um breve resumo com a explicação de alguns conceitos teóricos fundamentais para uma melhor compreensão do problema proposto. E posteriormente, será apresentado o algoritmo desenvolvido, assim como a sua análise e demonstração através da realização de vários testes.

# ***Fundamentos Teóricos***

# ***Algoritmo desenvolvido***

Para a realização deste trabalho desenvolveu-se a função detetor\_fala() que percorre a variável que contém o sinal de áudio, passando os segmentos de amostras que são consideradas como fala para a variável fala e os intervalos de silêncio e ruído para a variável ruido. Caso existam segmentos de fala na variável ruido, é necessário ajustar os parâmetros que a função recebe. A seguir, encontra-se representado uma explicação detalhada desta função.

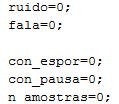
Os parâmetros que a função recebe são:

* fich\_audio: vetor que contém o sinal de áudio a analisar;
* noise\_len: número de amostras desde o início do vetor fich\_audio que correspondem a ruído;
* pause\_len: número de amostras permitido entre palavras num discurso, normalmente 50;
* espor\_noise\_len: número máximo de amostras que um ruido esporádico pode tomar.

A função retorna os argumentos:

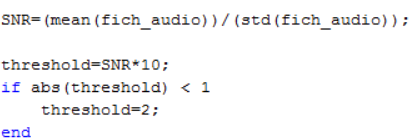
* fala: vetor que contém apenas os segmentos de fala;
* ruido: vetor que contém os segmentos de ruído.

No início da função inicializam-se os vetores de retorno e as variáveis necessárias a 0.



As variáveis con\_espor, con\_pausa e n\_amostras são usadas como contadores. A variável con\_espor conta o número de amostras para saber se o sinal obtido é esporádico ou não. A variável con\_pausa conta o número de amostras para saber se o sinal obtido é uma pausa entre as falas ou não. Por último, a variável n\_amostras é auxiliar para a contagem de amostras quando percorre-se o sinal pretendido.

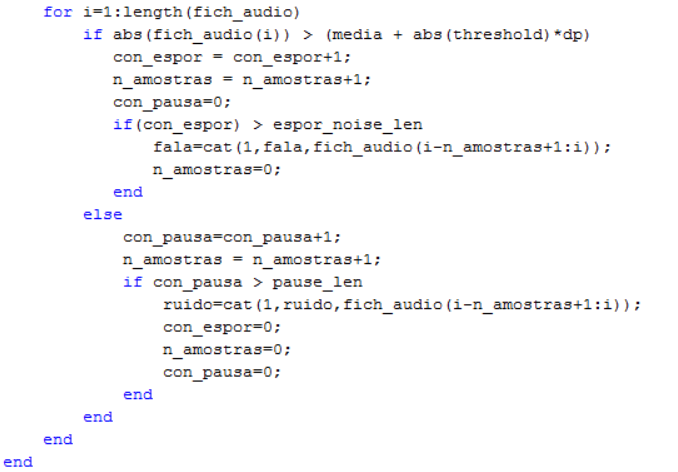
Posteriormente às inicializações, procedeu-se ao cálculo do SNR, relação sinal-ruído. Utilizamos este valor para estudarmos qual o melhor valor de threshold para SNR, sendo o threshold um limite, que indica qual é a probabilidade de encontrar ruido ou fala, assumindo uma distribuição gaussiana.



Assumindo que as primeiras amostras são compostas de ausência de fala, ou seja, apenas se gravou ruído, calcula-se o valor médio e o desvio-padrão deste “sub-sinal”,de modo a de seguida obtermos um modelo do ruído presente no sinal áudio. O próximo passo será então analisar o vetor com o sinal gravado e verificar se o valor será considerado ruido ou fala.



Após todas as inicializações e cálculos, passou-se para ao algoritmo, onde foi criado um ciclo que percorre o ficheiro de áudio e avalia-se se o valor do sinal trata-se de um valor considerado fala. Caso seja fala avalia-se se o contador do ruído esporádico é maior que o valor recebido como parâmetro, se for, conta-se como um segmento de fala e a informação é concatenada para o vetor fala, caso não seja é removido. O mesmo acontece para a pausa entre palavras, se o contador da pausa for maior que o valor passado por parâmetro, assume-se que é ruído e, neste caso, a informação é concatenada para o vetor ruido.



# ***Testes***

De seguida serão mostrados todos os teste e resultados obtidos usando a função detetor\_fala().

# ***1º Teste***

**Amostra**: “Ana Rodrigues”

Comandos a executar na linha de comandos do *Matlab*:

>> x=audiorecorder;

>> recordblocking(x,4);

>> play(x);

>> fich\_audio=getaudiodata(x);

>> sound(fich\_audio);

>> [fala,ruido]=detetor\_fala(fich\_audio,5000,50,250);

>> subplot(3,1,1); plot(fich\_audio);

>> title('Sinal original');

>> subplot(3,1,2); plot(fala);

>> title('Sinal "sem ruido"';)

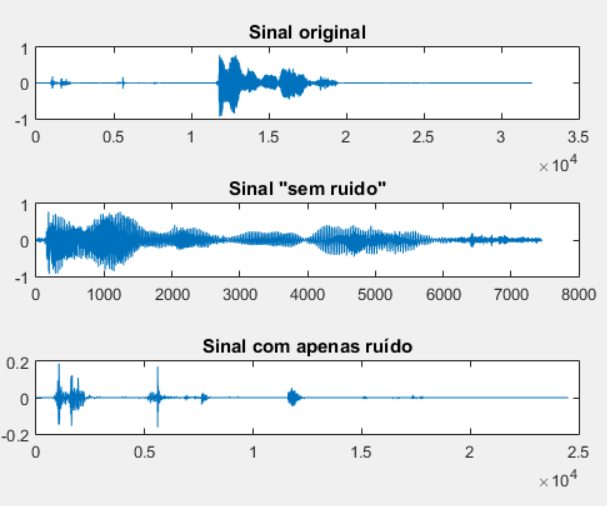
>> subplot(3,1,3); plot(ruido);

>> title('Sinal com apenas ruído');

>> sound(fala);

>> sound(ruido);

Neste teste, foi gravado, num ambiente sem ruido, a seguinte expressão “Ana Rodrigues”, podendo visualizar o sinal gravado, assim como, os sinais resultantes, no esquema abaixo representado.



Como podemos verificar, o resultado obtido com esta função é o pretendido, pois é eliminado algum ruído existente.

# ***2º Teste***

**Amostra**: “Ana” + pausa + “Rodrigues”

Comandos a executar na linha de comandos do *Matlab*:

>> x=audiorecorder;

>> recordblocking(x,4);

>> play(x);

>> fich\_audio=getaudiodata(x);

>> sound(fich\_audio);

>> [fala,ruido]=detetor\_fala(fich\_audio,5000,50,600);

>> subplot(3,1,1); plot(fich\_audio);

>> title('Sinal original');

>> subplot(3,1,2); plot(fala);

>> title('Sinal "sem ruido"';)

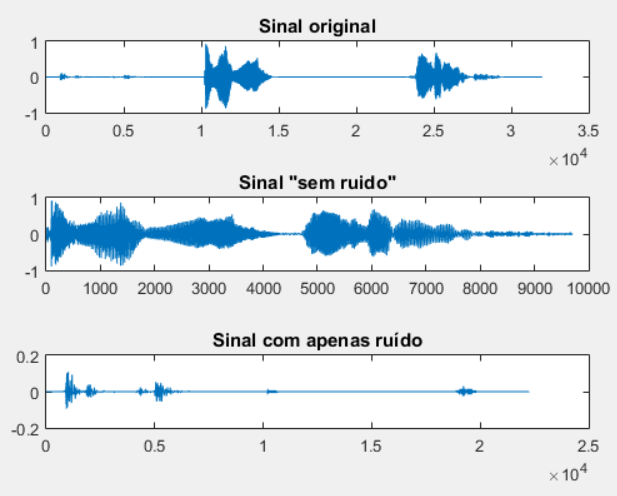
>> subplot(3,1,3); plot(ruido);

>> title('Sinal com apenas ruído');

>> sound(fala);

>> sound(ruido);

Neste teste, foi gravado, num ambiente sem ruido, a seguinte expressão “Ana Rodrigues” com uma pausa entre as palavras, de modo a perceber se a função reduz o espaço entre palavras para o valor passado por parâmetro. Pode-se visualizar o sinal gravado, assim como, os sinais resultantes, no esquema abaixo representado.



Como podemos verificar, o algoritmo está a funcionar pois é eliminada a pausa entre as palavras, assim como, algum ruído existente.

# ***3º Teste***

**Amostra**: “Ana Rodrigues” + objeto a cair

Comandos a executar na linha de comandos do *Matlab*:

>> x=audiorecorder;

>> recordblocking(x,4);

>> play(x);

>> fich\_audio=getaudiodata(x);

>> sound(fich\_audio);

>> [fala,ruido]=detetor\_fala(fich\_audio,5000,50,1150);

>> subplot(3,1,1); plot(fich\_audio);

>> title('Sinal original');

>> subplot(3,1,2); plot(fala);

>> title('Sinal "sem ruido"';)

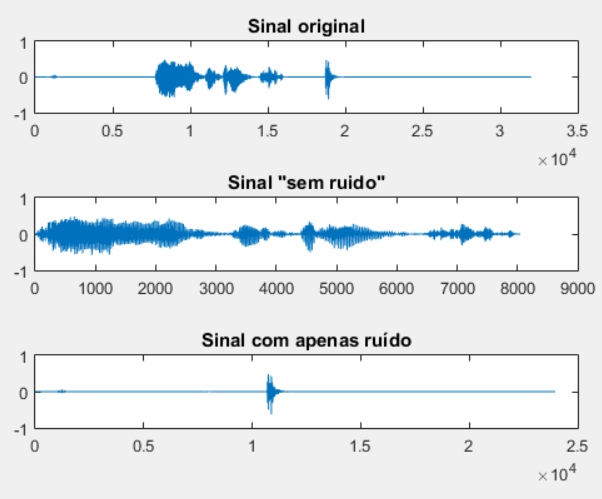
>> subplot(3,1,3); plot(ruido);

>> title('Sinal com apenas ruído');

>> sound(fala);

>> sound(ruido);

Neste teste, foi gravado a seguinte expressão “Ana Rodrigues” + ruido esporádico. Um ruído esporádico é um som ou barulho de curta duração com valores que se assemelham ao de uma palavra, como por exemplo um objeto a cair. No esquema abaixo é possível visualizar o sinal gravado, assim como, os sinais resultantes.



Como podemos verificar, o resultado obtido com esta função é o pretendido, pois o ruido foi eliminado.

# ***4º Teste***

**Amostra**: “Ana Rodrigues” + som do secador

Comandos a executar na linha de comandos do *Matlab*:

>> x=audiorecorder;

>> recordblocking(x,4);

>> play(x);

>> fich\_audio=getaudiodata(x);

>> sound(fich\_audio);

>> [fala,ruido]=detetor\_fala(fich\_audio,5000,50,1150);

>> subplot(3,1,1); plot(fich\_audio);

>> title('Sinal original');

>> subplot(3,1,2); plot(fala);

>> title('Sinal "sem ruido"';)

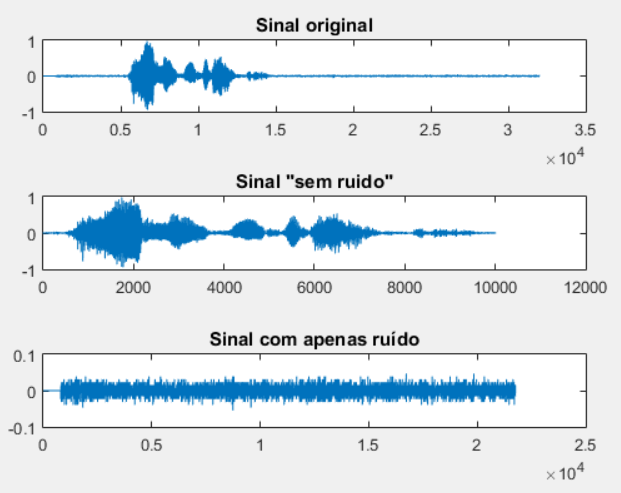
>> subplot(3,1,3); plot(ruido);

>> title('Sinal com apenas ruído');

>> sound(fala);

>> sound(ruido);

Neste teste, foi gravado a seguinte expressão “Ana Rodrigues” + ruido de fundo, sendo que foi utilizado um secador como ruido de fundo. No esquema abaixo é possível visualizar o sinal gravado, assim como, os sinais resultantes.



Como podemos verificar, o resultado obtido com esta função é o pretendido, embora alguma parte do ruído ainda aparecer no sinal “sem ruído”.

# ***Conclusão***

O trabalho realizado correu como previsto que no final foi atingido o objetivo 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.

Após a finalização de trabalho importa realçar também que a sua implementação permitiu adquirir conhecimentos acerca da ferramenta de desenvolvimento usada, *MATLAB.*