Universidade do Minho
Escola de Engenharia

2018/2019

Mestrado Integrado em Engenharia Eletrónica Industrial e Computadores
2018/2019

Deteção dos intervalos de silêncio na fala

Relatório do Segundo Trabalho de Processamento Digital de Sinal

Sérgio Baixo - A76513

Docente:

Professor Carlos Lima

Universidade do Minho
Escola de Engenharia

Índice:

Introdução

Este trabalho tem como objetivo detetar e eliminar pausas entre falas de um ficheiro de áudio cujo conteúdo resulta da gravação de uma voz com recurso ao Matlab. Este processo é conhecido como Speech Activity Detection e tem como finalidade detetar a presença ou a ausência de voz humana. Este processo possui ainda a capacidade para eliminar sons de elevada amplitude durante curtos períodos de tempo. O referido processo permite eliminar segmentos que não contêm qualquer informação(apenas ruído), de forma a evitar processamento de dados desnecessário e ainda, a transmissão desnecessária de conteúdo áudio que não contém quaisquer informação.

Este projeto foi desenvolvido com recurso ao Matlab e conta ainda com fundamentos de nível teóricos, que permitem consolidar ainda mais, os conhecimentos abordados ao longo das aulas bem como dos conhecimentos resultantes da realização deste trabalho.

Ruído Branco

O ruído branco caracteriza-se por ser um sinal estocástico causado pela combinação de sons de toda a gama de frequências. Em termos estatísticos, trata-se de uma sequência de erros aleatórios enquanto esta tiver variância e média constantes e não apresentar autocorrelação. Neste caso em particular, a média tomará o valor de 0.

A modelação do ruído pode ser efetuada através do cálculo da respetiva média e variância. Isto permite-nos diminuir os efeitos do mesmo ruído num sinal previamente contaminado, obtendo assim um sinal mais "suave".

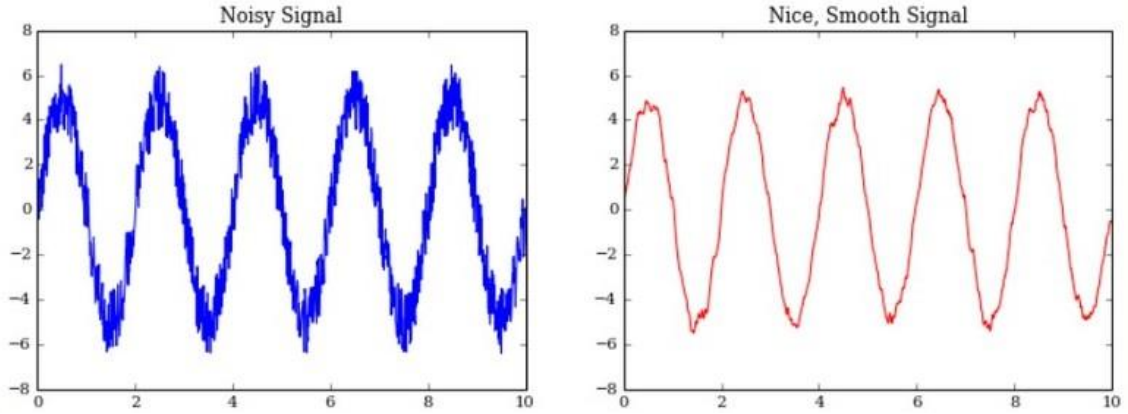

Figura 1-Sinal contaminado com ruído(á esquerda) e sinal suavizado com ruído reduzido(á direita)

Modelo Gaussiano

Uma distribuição gaussiana consiste num conjunto significativo de amostras de eventos aleatórios, cuja ocorrência individual não obedece a regras ou padrões que nos permitam tirar conclusões ou fazer previsões acertadas, mas que, a partir da análise das suas amostras, nos permitem concluir que os eventos tendem a ficar próximos de uma determinada posição, que representa a sua média matemática. Posto isto, o modelo gaussiano pode ser descrito pelos seguintes parâmetros:

-Média(μ)- valor para o qual os dados de uma distribuição estão mais centralizados;

-Desvio padrão(σ)- representa o valor de dispersão relativamente á média, ou seja quanto maior for este desvio padrão maior será o afastamento em relação á média;

-Variância(σ^2)- representa a distância á qual os valor obtidos se encontram dos valores esperados.

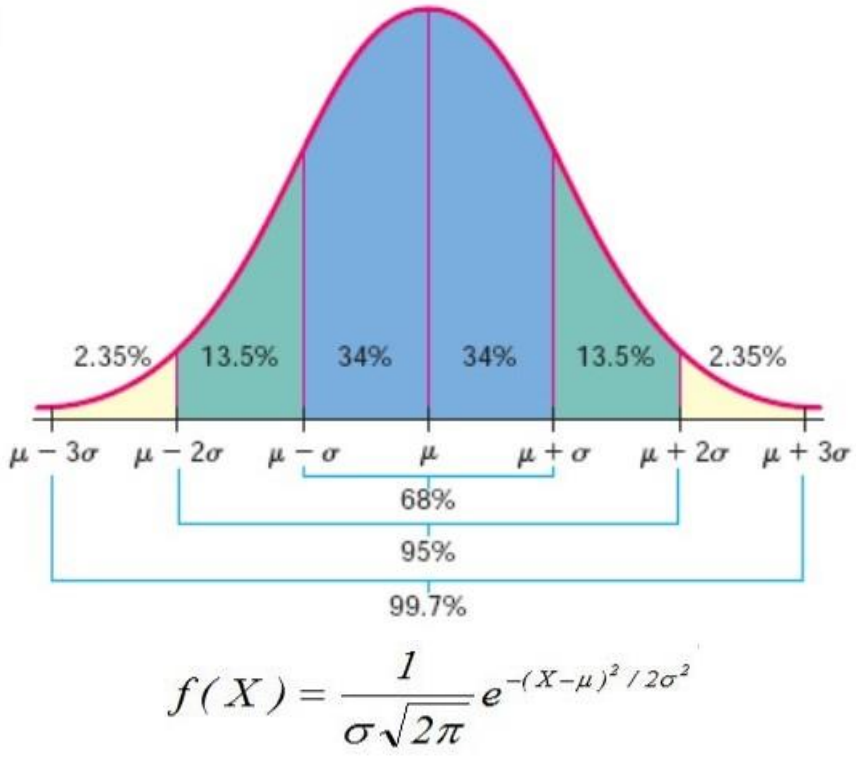

Figura 2- Distribuição Gaussiana

Através da distribuição gaussiana é possível modelar o sinal de ruído branco, assim como detetar *outliers*.

Um *outlier* ocorre sempre que um determinado valor se encontra muito afastado em relação aos restantes valores. 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.

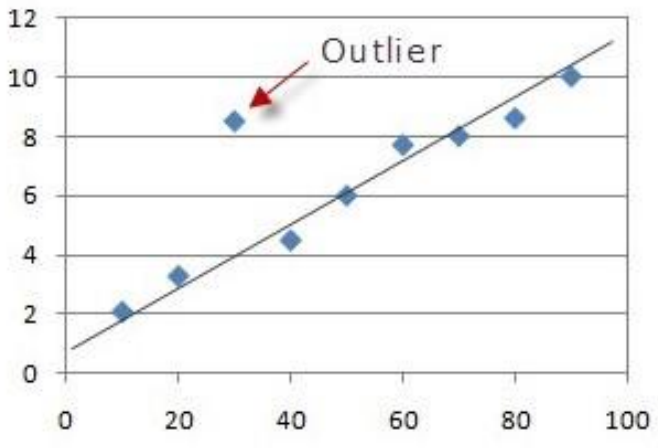

Figura 3- Outlier

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:

Etapa 1:

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

Etapa 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;

Etapa 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;*

Etapa 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:

$$\mathrm{SNR_{dB}} = 10\log_{10}\left(\frac{P_{\mathrm{sinal}}}{P_{\mathrm{ruido}}}\right).$$

Figura 4- 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 σ 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.

1	0	1	1	1	0	0	0	1	0	1	0
Fala				**Ruído**				**Fala**			

Figura 5- - Ilustração do método de diferenciação entre ruído e fala

Acima podemos observar uma ilustração do método de diferenciação entre ruído e fala. O array referido anteriormente é analisado entre segmentos, cujo tamanho deve ser definido através da variável “size_of_window”. Cada segmento é considerado de ruído ou fala mediante a razão “fraction_of_ones”. Se o número de uns for igual ou superior ao número de zeros, significa que estamos perante uma situação de fala, caso contrário será considerado ruído. A variável “índex_jump” define o salto do número de amostras em cada iteração da janela.

Desta forma verificamos que com os parâmetros “size_of_window =4”, “fraction_of_ones>=1/2” e “índex_jump=3”, a parte verde e laranja do array acima desenhado representam segmentos de fala, enquanto que a parte azul diz respeito a um segmento de ruído.

Implementação do problema em Matlab

O código desenvolvido em Matlab apresenta-se dividido em 3 partes: a gravação do sinal de áudio a analisar, bem como o seu armazenamento num ficheiro áudio .wav, a contaminação desse mesmo sinal com ruído branco e por fim a parte que diz respeito á deteção de pausas na fala.

As primeiras linhas de código do script referem-se ás variáveis de entrada usadas ao longo do programa:

```
%%%%%%VARIÁVEIS DE ENTRADA
    time_talk = 6
    fs = 8000
    nbits = 16
    nchannels = 1
    Length = 7;
    audio_file = 'C:\Users\sergi\OneDrive\Ambiente de Trabalho\Universidade\PDS\som8k.wav'
    SNR = 50
    threshold = 0;
    fraction_of_ones = 0.35;
    index_jump = 1400;
    size_of_window = 1500;

%%%%%%%%%%%%%%%%
```

Figura 6- Variáveis de entrada

A parte do código que se segue diz respeito ao segmento de gravação de áudio:

```
%%%%%%GRAVAÇÃO DO SINAL ÁUDIO

record = audiorecorder(fs,nbits, nchannels)      %objeto para a gravação de áudio
message1 = ['Start speaking, you have ', num2str(time_talk), ' seconds']
disp(message1)
recordblocking(record, time_talk)        %grava o áudio durante 'time_talk' segundos
f = getaudiodata(record)
disp('End of Recording.')
audiowrite(audio_file,f,fs)      %escrita dos dados da gravação num ficheiro .wav

plot(f)
figure(2)


%%%%%%%%%%%%%%%%
```

Figura 7- Gravação do segmento de áudio

Em seguida encontra-se o algoritmo que se encarrega da parte de contaminar o sinal de fala previamente gravado:

É de notar que a escolha do SNR terá impacto na variável de saída 'contaminated', quanto menos fosse o SNR, maior a perceção de ruído presente no sinal.

```
%%%%%A contaminar o sinal com ruido%%%
disp('A contaminar o sinal com o SNR desejado...')
    f_energy = power(f,2)              %cálculo da energia do sinal de áudio
    f_energy = mean(f_energy)              %cálculo da energia do sinal de áudio
    wn = randn(1,length(f))               %sinal de ruído do tamanho do sinal de áudio f
    wn = wn - mean(wn)                          %média 0
    wn = wn * sqrt(f_energy / (10^(SNR / 10)))
    contaminated = f + wn'

    plot(contaminated)
    figure(3)


%%%%%%%%%%%%%%%%
```

Figura 8- Contaminação do sinal

Segue-se abaixo a última parte do programa, que diz respeito ao algoritmo de deteção de pausas na fala:

```
%%%%%A detetar pausas no sinal%%%
 disp('Erasing all the pauses from speech. . .');
    if (SNR > 0 && SNR < 15) threshold = 0.45;
    elseif (SNR > 15 && SNR < 25) threshold = 0.5;      %o valor do treshold varia consoante o SNR escolhido
    elseif (SNR > 25 && SNR < 35) threshold = 0.6;
    elseif (SNR > 35 && SNR < 45) threshold = 0.75;
    else threshold = 0.8;
    end
    ruido = 0;
    fala = 0;
    cont_mean = mean(contaminated);        %cálculo da média do sinal
    cont_desv = std(contaminated);         %cálculo do desvio padrão do sinal de áudio
    cont_array = 1:1:length(contaminated);     %array com o tamanho do sinal áudio onde são guardados os zeros e uns
    for i=1:length(contaminated)       %percorrer o sinal de áudio e colocar zeros e uns em cont_array
        if(abs(contaminated(i) - cont_mean) / cont_desv > threshold)              %se é um outlier (fala)
            cont_array(i) = 1;                                                     %coloca 1 no array
        else                                                                  %se é ruído
            cont_array(i) = 0;                                                     %coloca 0 no array
        end
    end
    n_ones = size_of_window * fraction_of_ones;             %quantidade de 1's para ser validado como fala
    n = 1;
    while(((n-1)*index_jump + size_of_window) < length(contaminated))               %se a gravação ainda não acabou
        j = cont_array((n-1)*index_jump + 1 : (n-1)*index_jump + size_of_window);   %armazena a informação de 1 janela
        if sum(j) > n_ones                                                    %se for fala
            if ~fala                                                          %se o buffer da fala for nulo
                fala = contaminated((n-1)*index_jump + 1 : (n-1)*index_jump + size_of_window);
            else                                                              %concatena os buffers que contêm fala
               fala = cat(1,fala,contaminated((n-1)*index_jump + 1 : (n-1)*index_jump + size_of_window));
            end
        else                                                                  %se for ruído
            if ~ruido                                                         %se o buffer do ruido for nulo
                ruido = contaminated((n-1)*index_jump + 1 : (n-1)*index_jump + size_of_window);
            else                                                              %concatena os buffers que contêm ruído
                ruido = cat(1,ruido,contaminated((n-1)*index_jump + 1 : (n-1)*index_jump + size_of_window));
            end
        end
        n=n+1;
    end

%%%%%%%%%%%%%%%%
```

Figura 9- Deteção de pausas

SNR	Threshold
0 < SNR < 15	0.45
15 < SNR < 25	0.5
25< SNR < 35	0.6
35 < SNR < 45	0.75
SNR > 50	0.8

Tabela 1- Relação entre SNR e threshold

Resultados obtidos

Através dos seguintes gráficos podemos observar que, quanto menor for o valor do SNR, mais difícil se torna a deteção de pausas na fala e sobretudo mais difícil será a remoção de ruídos de maior amplitude tal como um objeto a cair. Isto acontece pois torna-se mais difícil para o algoritmo distinguir a diferença entre fala e silêncio, devido ao ruído ganhar proporções muito elevadas e se aproximar da amplitude da fala. Os testes abaixo foram efetuados dizendo as letras "A B C" seguidas e com uma pausa de aproximadamente 500ms entre elas. Os parâmetros selecionados foram:

'fraction_of_ones' = 0.18;
'index_jump' = 400;
'size_of_window' = 500;

O SNR varia entre 0dB e 50dB. Para efeitos de simplificação apenas serão colocados os resultados para SNR=8 e SNR=50.

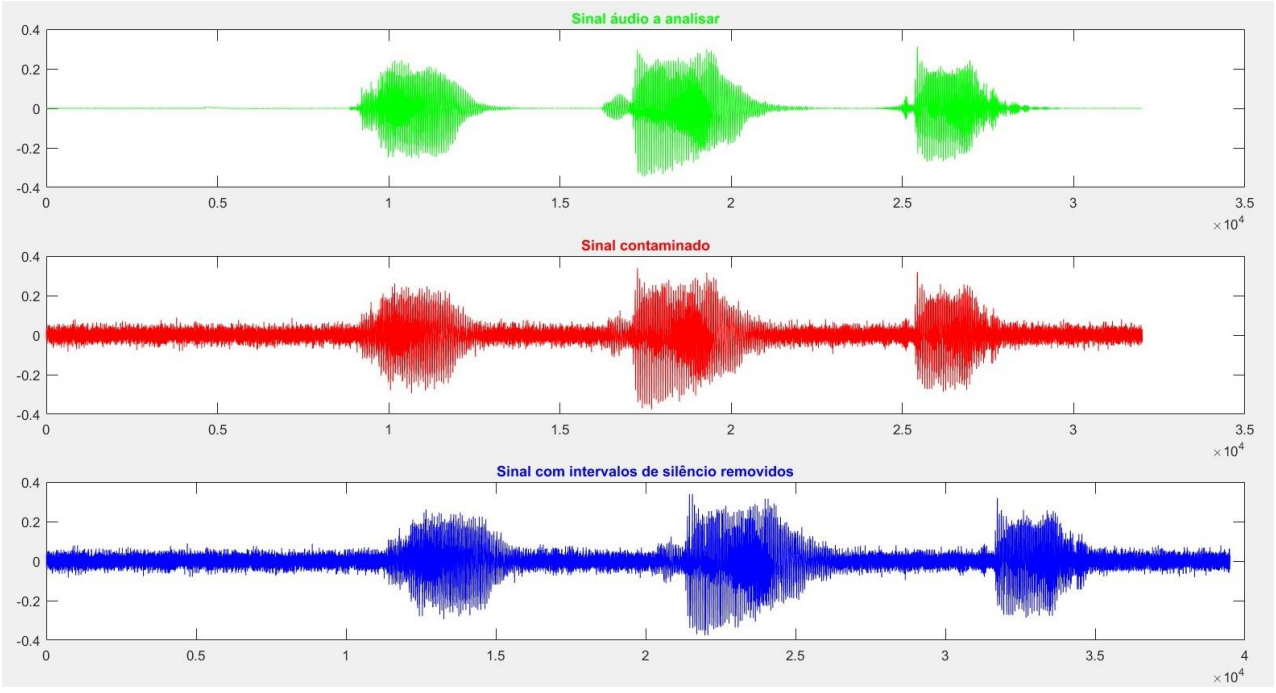

Figura 10- Remoção de pausas para SNR = 8

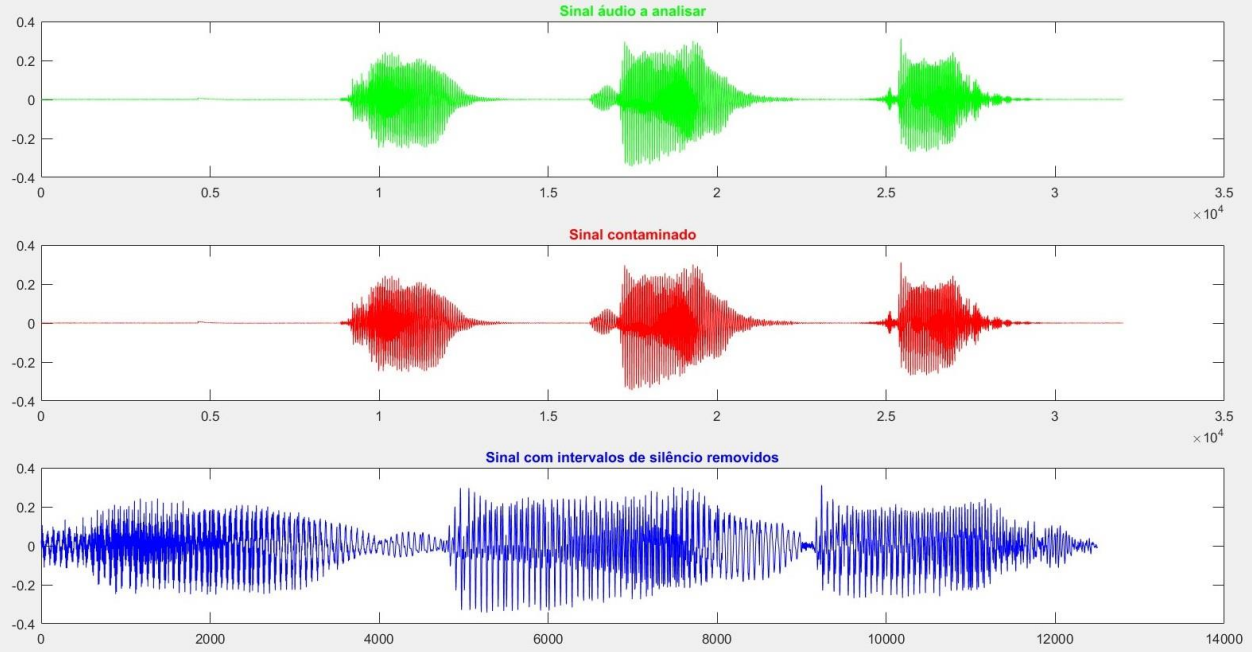

Figura 11- Remoção de pausas para SNR = 50

Analisando as duas figuras, torna-se evidente que o algoritmo é eficiente para valores de SNR elevados. Contudo, á medida que diminuímos o valor de SNR, cada vez mais segmentos de pausa ficam armazenados no sinal de fala. Isto deve-se á maior dificuldade de distinção dos segmentos de ruído dos de fala pois a amplitude do ruído está cada vez mais próxima da amplitude do sinal de fala.

Porém, convém ressaltar que o ajuste dos parâmetros 'fraction_of_ones', 'index_jump' e 'size_of_window' permitem a eliminação de sinais de curta duração e amplitude relativamente maiores. De forma a demonstrar este acontecimento, foram alterados os parâmetros acima mencionados para os valores de '0.30', '1500' e '1600' respetivamente e foi gravado uma nova amostra da fala ligeiramente diferente da amostra acima, desta vez proferindo as letras "A B C" com um toque numa mesa de madeira entre as letras e ainda adicionando um tempo de atraso de 1.5 segundos entre o início da gravação e a fala da letra A, de modo a demonstrar mais uma vez a remoção de silêncios entre falas.

Primeiramente o valor de SNR selecionado foi de 10dB. Obteve-se o seguinte resultado:

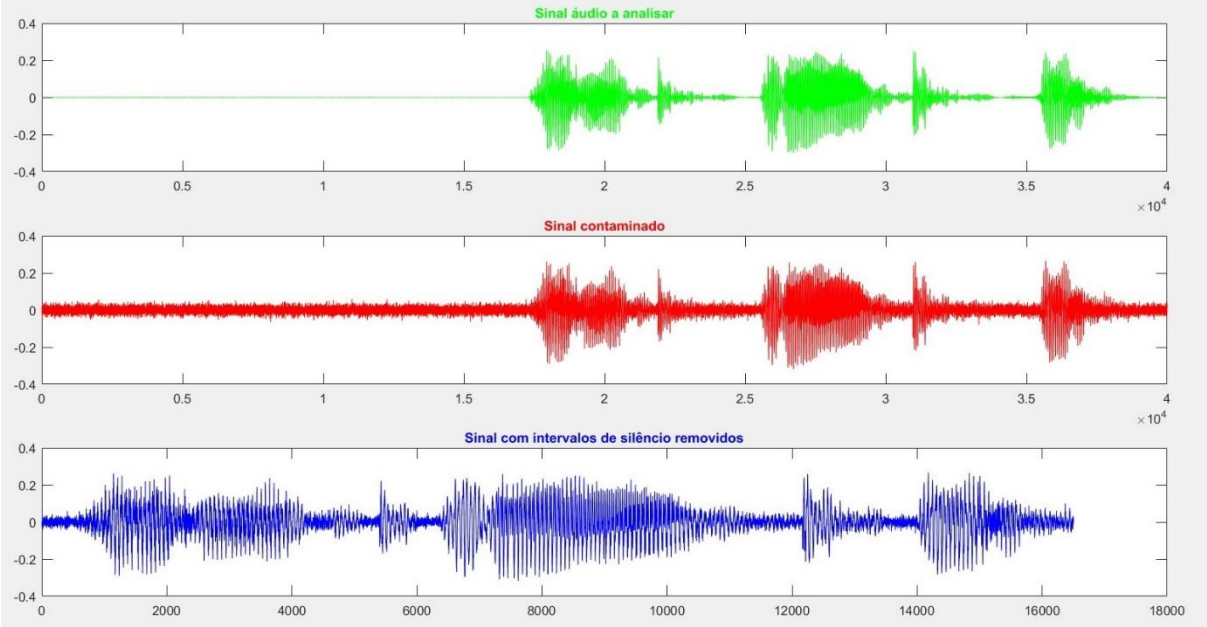

Figura 12- Remoção de silêncios com toques na mesa para SNR = 10

De seguida foi realizado o mesmo estudo porém, utilizando um valor de SNR = 50dB.

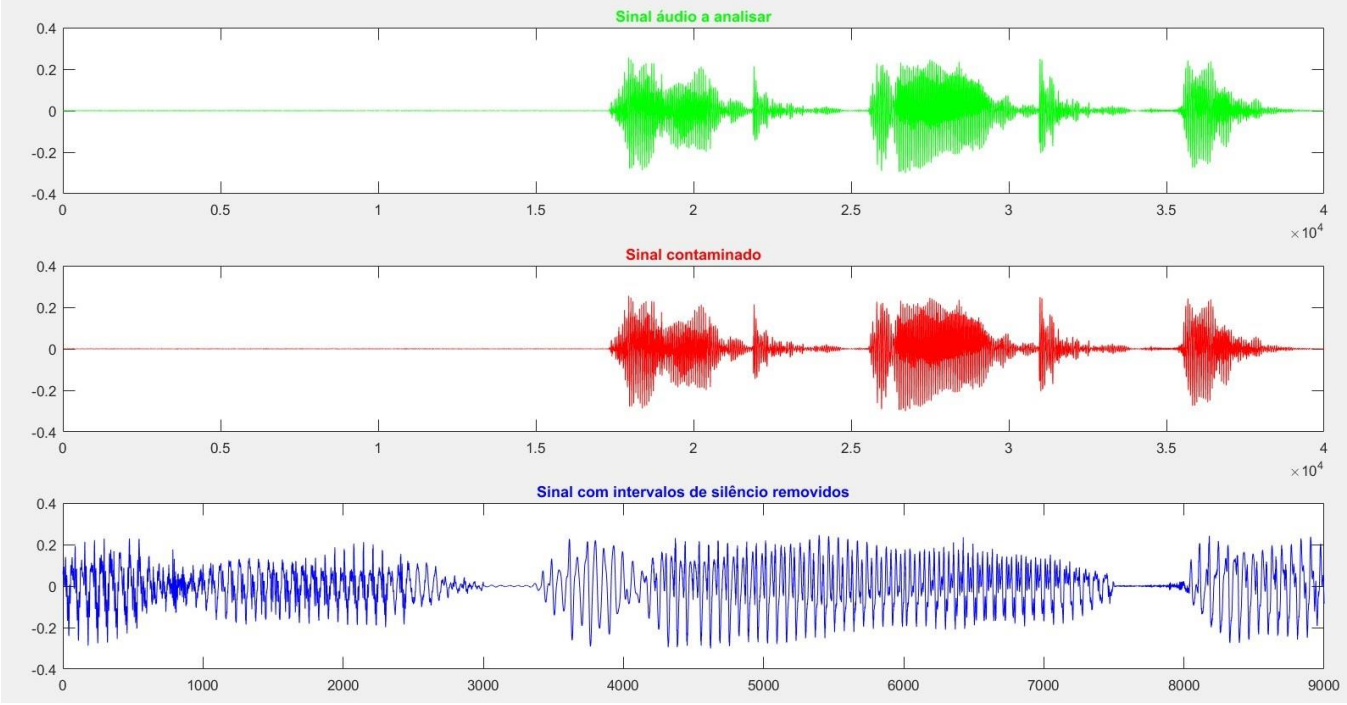

Figura 13- Remoção de silêncios com toques na mesa para SNR =50

Tendo em conta os resultados obtidos através da análise dos gráficos e do áudio resultante da aplicação do algoritmo de remoção dos intervalos de silêncio, conclui-se que á medida que o valor de SNR diminui, menor será a capacidade do algoritmo de remover os sons curtos de maior amplitude, neste caso concreto, os toques na mesa entre as letras. De tal modo que, para SNR = 50, os toques na mesa são completamente removidos, bem como os intervalos de silêncio designados como ruído.

Conclusão

Através da realização deste trabalho foi possível, acima de tudo, consolidar os diversos conhecimentos lecionados nas aulas teóricas de Processamento Digital de Sinal aplicando os mesmos á resolução de um problema do dia a dia relacionado transmissão de informação quer seja de carácter relevante ou não.

É de salientar inclusive os conceitos adquiridos ao longo do desenvolvimento do algoritmo através do uso do Matlab, ferramenta que se revelou ser bastante útil na execução e análise das diversas partes do algoritmo utilizado.