

Engenharia Electrónica Industrial e Computadores

Artur Manuel Pereira Faria

Optimização e redimensionamento de sinais de áudio usando o Matlab

Campus de Azurém, Guimarães, Portugal

Julho / 2013

Artur Manuel Pereira Faria

63358

Optimização e redimensionamento de sinais de áudio usando o Matlab

Unidade Curricular: Processamento Digital de Sinal

Coordenador: João Luís Marques Pereira Monteiro

Equipa docente: Carlos Manuel Gregório Santos Lima

Julho / 2013

Introdução

No final da unidade curricular de Processamento Digital Sinal, foi proposto um trabalho final suplementar, que terá um peso na avaliação final substancial, este trabalho tem como objectivo a optimização/redimensionamento de sinais de áudio provenientes da fala humana. A pergunta óbvia está no motivo/força que leva a este redimensionamento, ora o principal motivo é a optimização de recursos, isto é, é garantir que apenas informação/dados realmente uteis (fala) são transmitidos, ou seja, é impedir que dados inúteis/parasitas (lixo) sejam perpetuados, quer sejam ruídos de fundo ou silêncios. Aliás esta meta de eliminação de dados/informações inúteis é uma das grandes ambições por exemplo da área das comunicações. Para proceder a esta optimização terá de ser desenvolvido um algoritmo que permita em primeiro lugar detectar ruído, quer seja no início, meio ou fim de um dado sinal de áudio e em segundo lugar proceda à eliminação desse mesmo ruído, de forma que o sinal resultante seja apenas o sinal de áudio correspondente à fala humana, sem qualquer ruído/silêncio associado. O algoritmo a ser projectado deve inspirar-se substancialmente no protocolo de Shewart (“Shewart Protocol”) mas com threshold adaptado às condições do ruído.

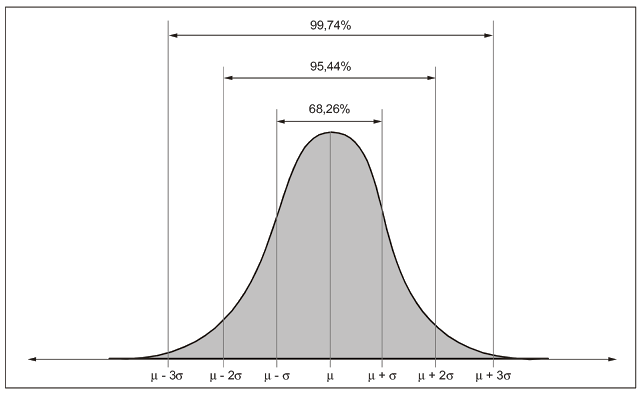
Para a realização deste trabalho recorre-se ou dever-se-á recorrer à utilização da plataforma/programa Matlab, quer para a projecção da(s) função(ões), quer para o teste da(s) mesma(s), sendo que os dados terão/deverão ser recolhidos e estudados/comentados.

Contextualização

* Contexto de Distribuição de Gauss / Gaussiana

A distribuição de Gauss, é talvez a mais conhecida e a mais importante de todas as distribuições de probabilidades, mas o que é ao certo uma distribuição de probabilidades?

É um modelo/construção que assenta na incerteza/grau de incerteza/variabilidade associada ao desempenho de uma dada variável monitorizada, ou seja, é o grau de incerteza associado a um dado fenómeno a ser estudado, sendo que acaba por depender sempre do comportamento de uma variável de monitorização/variação. No caso da distribuição de Gauss, esta caracteriza-se por uma curva de distribuição de probabilidades simétrica, e acaba directamente por assentar na própria lei dos grandes números de Bernoulli (uma aplicação interessante neste campo é por exemplo o lançamento de um dado, teoricamente apenas temos um sexto de probabilidade para cada face, mas isto só acontece para infinitos testes e não para reduzidos testes, ou seja, para mais testes temos menos erro), e na ideia desenvolvida a partir deste modelo por Abraham de Moivre, que é a ideia de que os erros de alto valor são muito pouco frequentes, isto é, é percentualmente mais comum a existência de erros de baixo ou muito baixo grau que alto grau, por isso a distribuição de Gauss assume um aspecto semelhante ao mostrado abaixo.

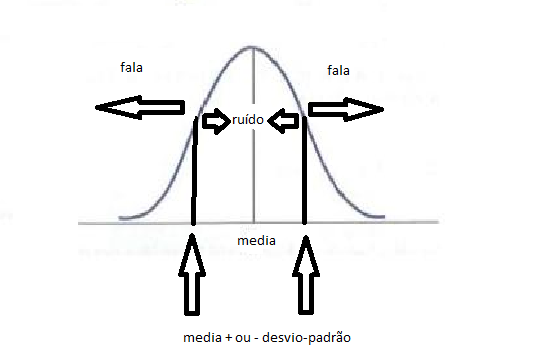


Constata-se que quanto mais alto o desvio/erro relativamente à média, menor é a probabilidade associada ao acontecimento. Note-se que a curva recai directamente na segmentação de probabilidades de uma dada variável e apenas obriga à necessidade de conhecimento de 2 factores muito importantes num estudo deste tipo, que são a média e desvio padrão.

* Mas qual o papel da distribuição de Gauss neste contexto?

É através do conceito de distribuição de Gauss / Gaussiana que se consegue chegar a um importante conceito/modelo, que é a um modelo de ruído, isto é, consegue-se de forma mais ou menos precisa construir uma espécie de distribuição de probabilidades a partir de dados relativos a um dado sinal, sinal este que contém fala e ruído.

Este modelo é útil pois é o que nos vai permitir de certa forma, diferenciarmos ruído/tempos mortos, de tempos de fala. Pensemos na média de um sinal, quando calculámos a média da primeira parte(ruído) de um sinal sinal x, com x = wavrecord(3\*8000,8000), esta vai dar um valor relativamente díspar por exemplo do valor máximo, pois grande parte do sinal vai ser ou puro ruido ou tempo morto(sem fala), ou seja a probabilidade de não termos fala é alta/mais alta. Recorrendo a uma simplificação do modelo/distribuição de Gauss, chegámos a algo do género do que é mostrado:



Ou seja, com µ=media e  σ=desvio padrão e X o valor da variável de monitorização:

P ( |X-µ| < σ) = 68,26 = 68.3 (mais ou menos) -> Probabilidade associada ao ruído

P ( |X-µ| > σ) = 100-68.26 = 31.74 = 31.7 (mais ou menos) ->Probabilidade associada à fala

Outros valores importantes

P ( |X-µ| < 2xσ) = 95.44 = 95.4 (mais ou menos) P ( |X-µ| >2xσ) = 100 - P ( |X-µ| < 2xσ)

P ( |X-µ| < 3xσ) = 99.74 = 99.7 (mais ou menos) P ( |X-µ| >3xσ) = 100 - P ( |X-µ| < 3xσ)

Projecção de possível implementação em Matlab:

Função 1:

Começando por uma função mais simples, que tem várias limitações associadas temos algo semelhante à função mostrada abaixo:

Em Matlab: Consultar os anexos (.m) para visualizar comentários relativos à implementação e funcionamento da mesma.

function [ y ] = filtra(x,nseg,amostras\_fala,dp\_multiplier)

med=mean(x(1:nseg));

v=var(x(1:nseg));

dp=sqrt(v);

j=0;

ciclo=1;

n\_amostras=0;

while(ciclo)

j=j+1;

if( abs(x(nseg+j)-med) > dp\_multiplier\*dp )

n\_amostras=n\_amostras+1;

if(n\_amostras==amostras\_fala)

y=x( (nseg + j - amostras\_fala) : length(x) );

ciclo=0;

end

else

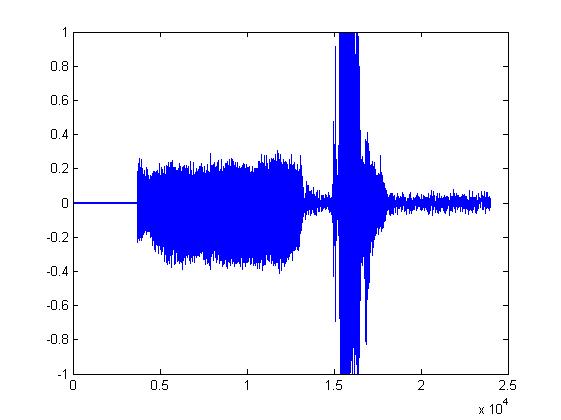
n\_amostras=0;

end

end

end

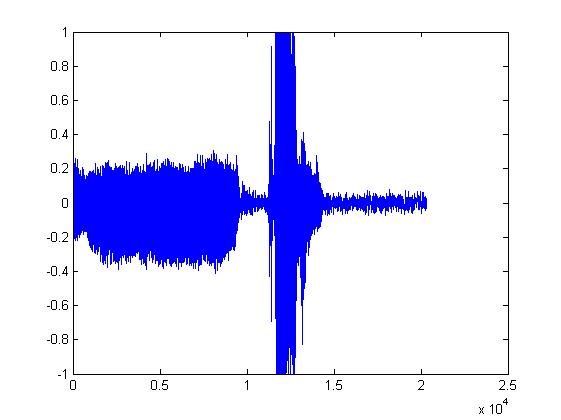
Notas breves: A função em questão acaba por ser um desenvolvimento da apresentada na aula de dúvidas, apenas possui um mecanismo acrescido de confirmação de situação de fala que é controlado pelo parâmetro amostras\_fala, este mecanismo foi necessário em alguns sinais para garantir que o modelo de ruído inicial era validado de uma forma controlada, pois em alguns casos chegava a quase não haver correcção com o método passado na aula, com este os resultados são satisfatórios.

 Testes da função em questão:

Teste 1:

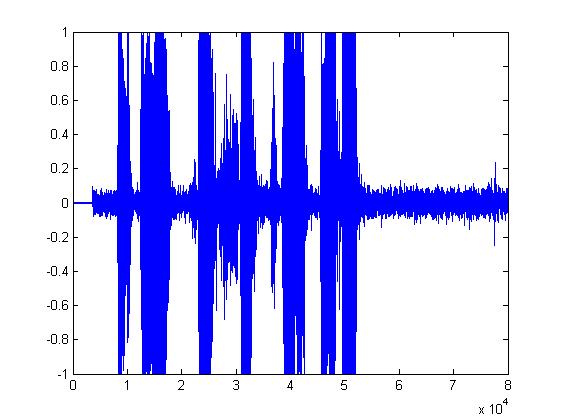
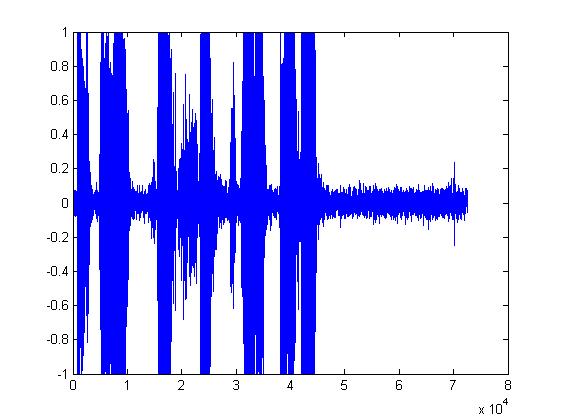
Palavra “Arca”:

Sinal original sem filtragem

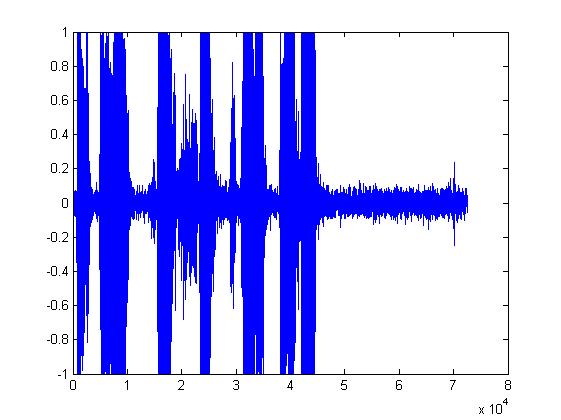
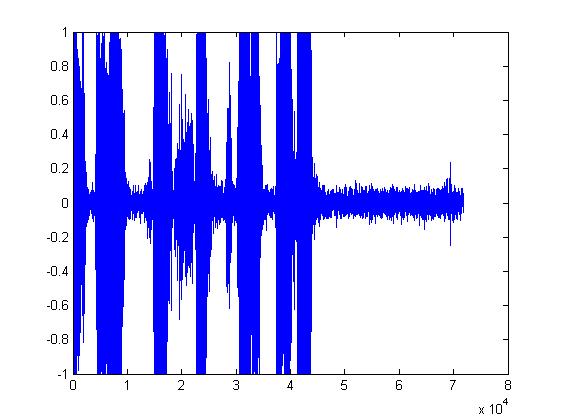


Sinal filtrado com dp\_multiplier=1 e amostras\_fala=10

Frase “O João foi à escola para estudar”:



Sinal sem filtragem Sinal com filtragem, com dp\_multiplier=1 e amostras\_fala=10



Sinal com filtragem, com Sinal com filtragem, com

dp\_multiplier=2 e amostras\_fala=10 dp\_multiplier=3 e amostras\_fala=10

Conclusões relativas à função:

[ y ] = filtra(x,nseg,amostras\_fala,dp\_multiplier)

A função usada é apenas um bom trunfo para se eliminar o ruído inicial, no entanto é preciso ter em conta, que nada mais faz além disso, isto é, apenas consegue eliminar ruído até detectar fala, o que de modo nenhum é o pretendido/desejado.

Note-se que é muito importante o intervalo escolhido do sinal que é definido como ruído (1 a nseg), pois deve ser tido em conta que não se deve pegar em sinal de fala útil e que a média e o desvio padrão deste intervalo são usados para a construção do modelo de ruido, usado para a identificação e eliminação de ruído no restante sinal. Para além deste parâmetro é de extrema importância o parâmetro amostra\_fala, pois assegura uma maior consistência na identificação de fala sem ter qualquer impacto negativo no sinal resultante.

Por fim dp\_multiplier é igualmente de extrema importância pois assegura também ele a construção do modelo de ruído, visto que, é este que permite a expansão ou compressão do intervalo de valores entre a média e a fronteira/limite do modelo actual de ruído.

Função 2:

Projecção de uma função mais completa, que é capaz de eliminar ruído e tempos mortos num sinal de fala completo, isto é, não tem qualquer problema de interrupção de execução, pois apenas acaba a execução depois de analisar todo o sinal, fazendo com que seja capaz de eliminar tudo o que não é fala.

function [buff,noisebuff]

=filtra\_melhorada(x,nseg,wlen,step,frac,conta\_amostras\_fala,dp\_multiplier)

m = mean(x(1:nseg));

v = var(x(1:nseg));

dp = sqrt(v);

alfa = frac \* wlen;

conta=0;

for i=1 : length(x)

a(i)=0;

if ( abs(x(i) - m) > dp\_multiplier\*dp )

conta=conta+1;

if(conta==conta\_amostras\_fala)

while(conta~=0)

a(i-conta)=1;

conta=conta-1;

end

end

else

conta=0;

end

end

buff=0;

noisebuff=0;

n=0;

while( ( n \* step + wlen ) < length(x) )

w = a( ( n \* step + 1) : ( n \* step + wlen ) );

if(sum(w) >= alfa)

if(~buff)

buff = x( ( n \* step + 1) : ( n \* step + wlen) );

else

buff = cat(1, buff , x( ( n \* step +1) : ( n \* step + wlen) ) );

end

else

if(~noisebuff)

noisebuff=x( ( n \* step + 1) : ( n \* step + wlen) );

else

noisebuff = cat(1, noisebuff , x( ( n \* step +1) : ( n \* step + wlen) )

end

end

n=n+1;

end

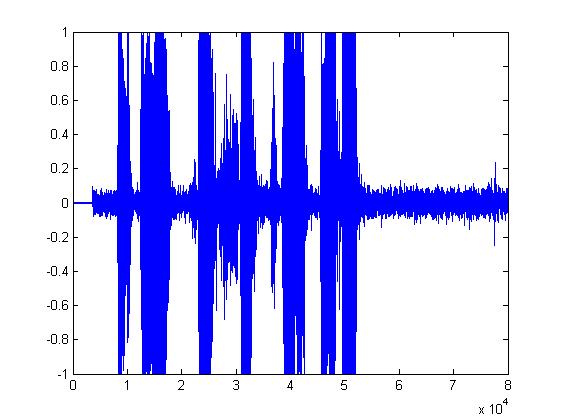
end

Notas breves: A função em questão acaba por ser também ela um desenvolvimento da apresentada na aula de dúvidas, apenas possui um mecanismo acrescido de confirmação de situação de fala que é controlado pelo parâmetro conta\_amostras\_fala, este mecanismo embora não seja estritamente necessário neste caso, corresponde a mais um meio de configuração/ajuste de filtragem/análise. Note-se que esta função necessita de uma estrutura auxiliar que funciona como guia para análise e correcção do sinal recebido, sendo que esta estrutura acaba por apenas guardar 0 ´s e 1´s, que servem para identificar situações de fala ou ruido através da comparação com o parâmetro alfa;

Testes da função em questão:

function [buff,noisebuff]

=filtra\_melhorada(x,nseg,wlen,step,frac,conta\_amostras\_fala,dp\_multiplier)



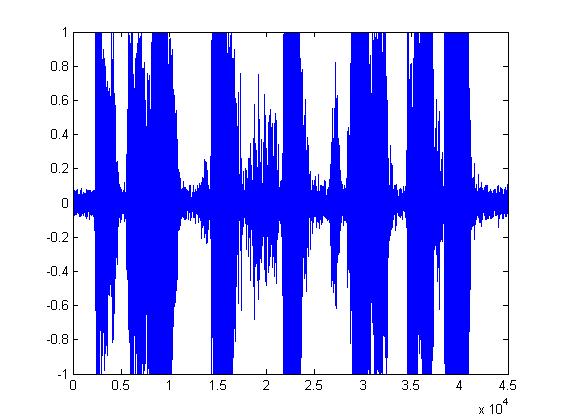
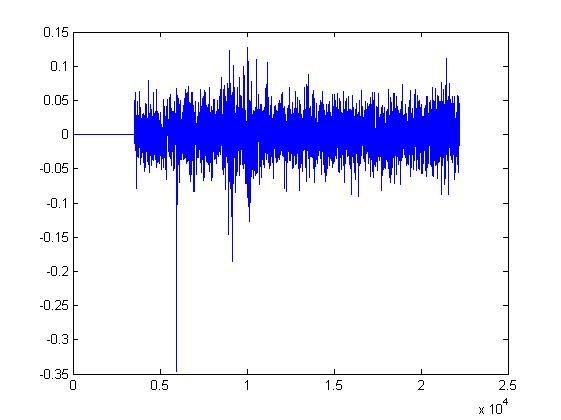
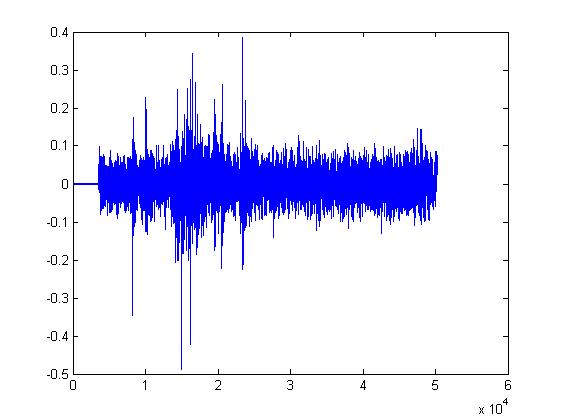
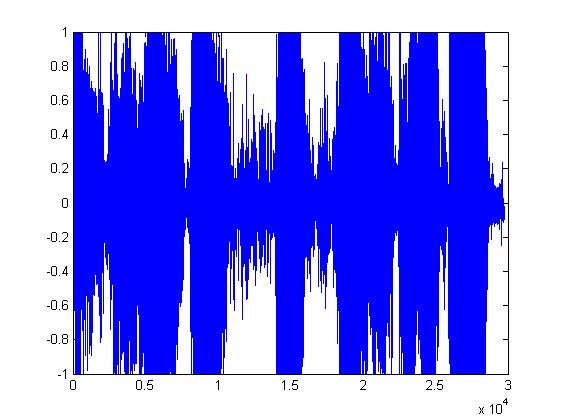
Frase/Palavra de teste: “O João foi à escola para estudar”:

Resultados com wlen=25, step=25, frac=0.5, conta\_amostras\_fala=1:

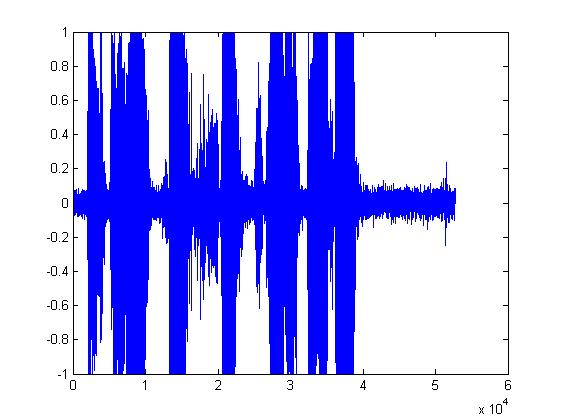
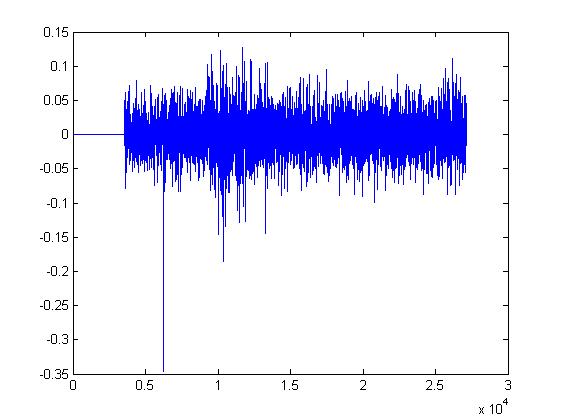
Organização:

lado direito -sinal sem ruído lado esquerdo-ruído

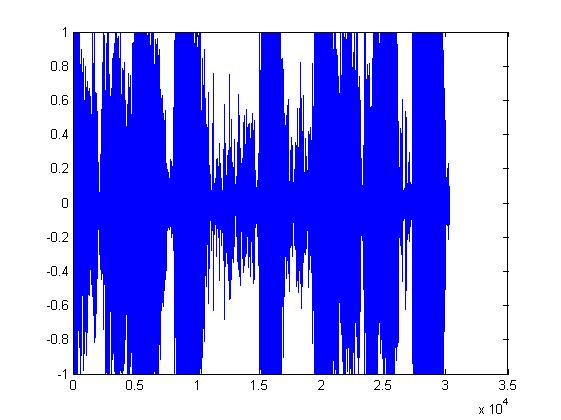
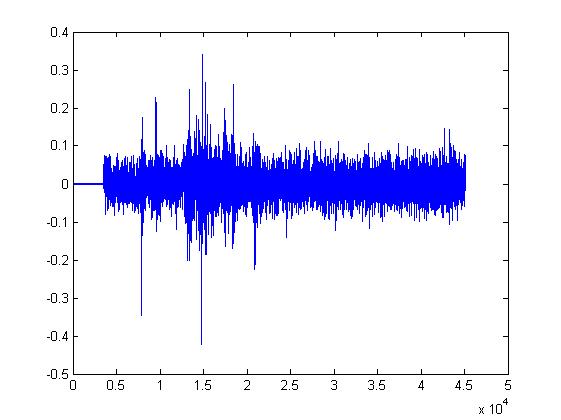
Ooooooo

*  Com dp\_multiplier=1:
*  Com dp\_multiplier=3 (+probabilidade para eliminar ruido (modelo com mais expansão/largura)), note-se que esta expansão pode ser prejudicial, pois pode-se estar a eliminar fala útil, coisa que acontece com relativa facilidade;

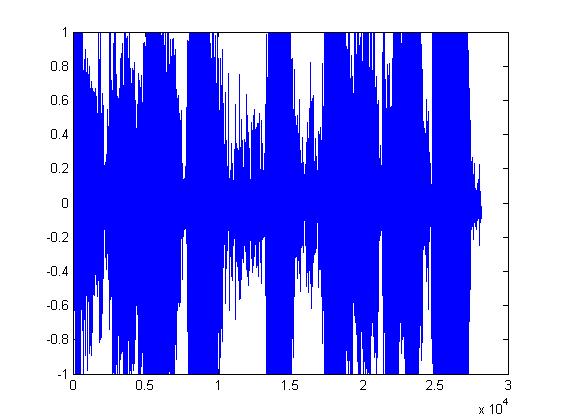
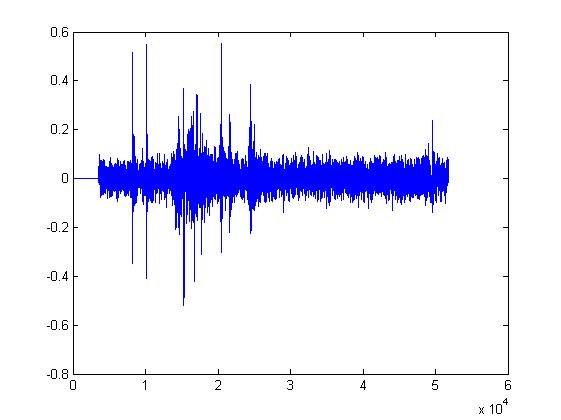
Resultados com wlen=25, step=25, frac=0.6, conta\_amostras\_fala=1:

*  Com dp\_multiplier=1:

* Com dp\_multiplier=2:

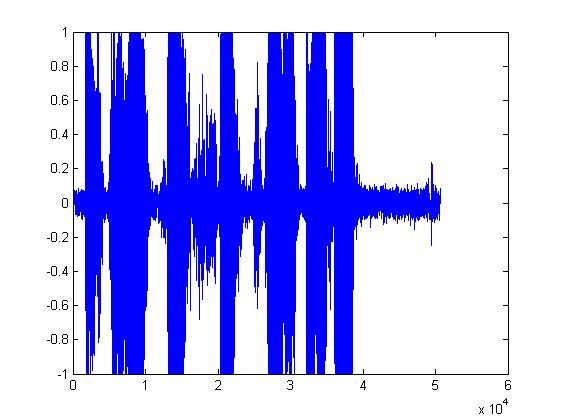
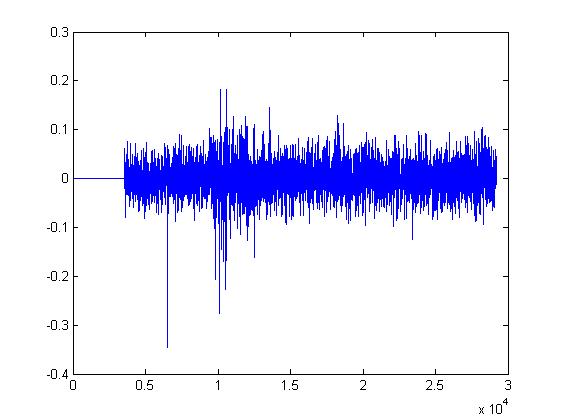


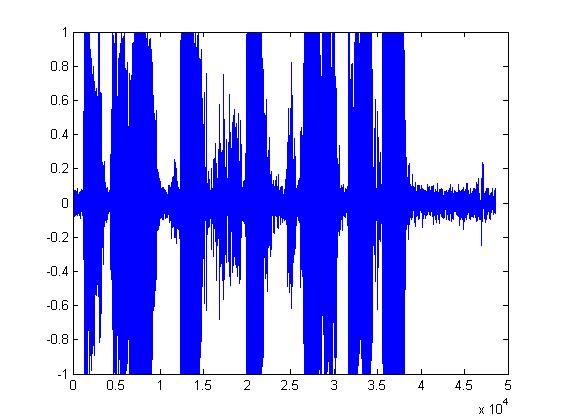
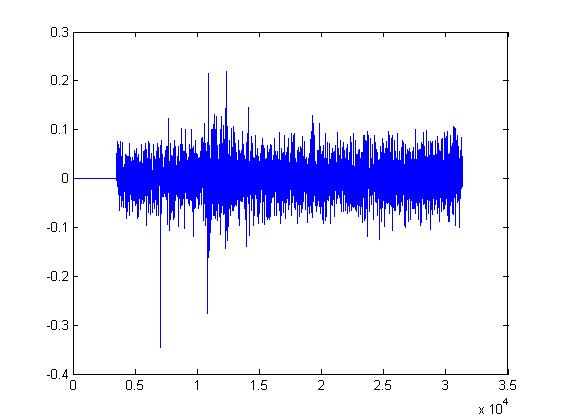
* Com dp\_multiplier=3:



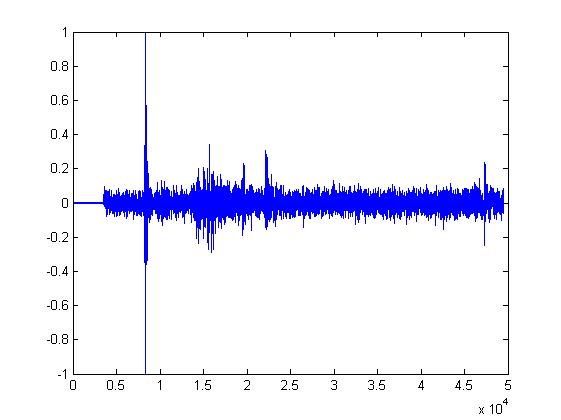
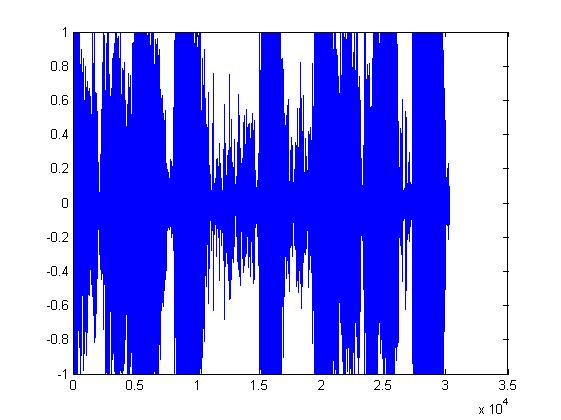
Analise com variação tamanho da janela e dp\_multiplier também variável para frac=0.6 :

* Com dp\_multiplier=1:

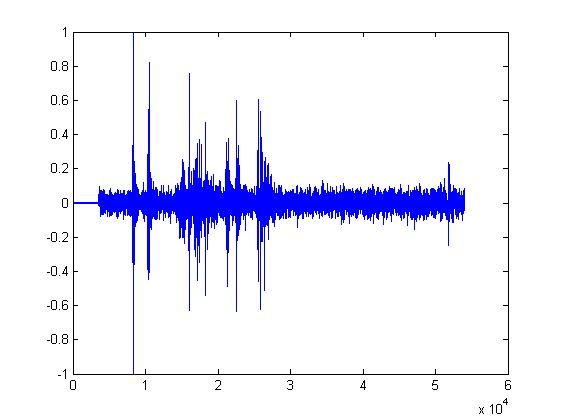
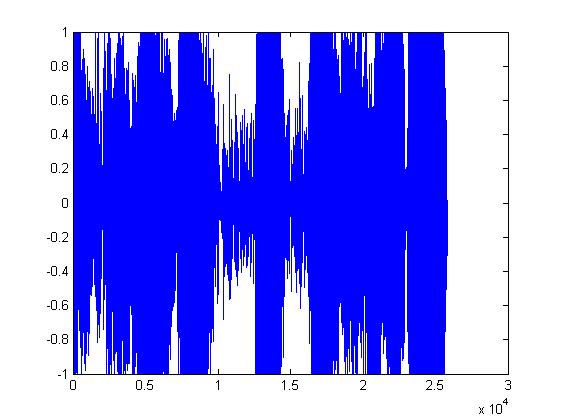
Resultados com wlen=50, step=50, frac=0.6, conta\_amostras\_fala=1:

Resultados com wlen=150, step=150, frac=0.6, conta\_amostras\_fala=1:

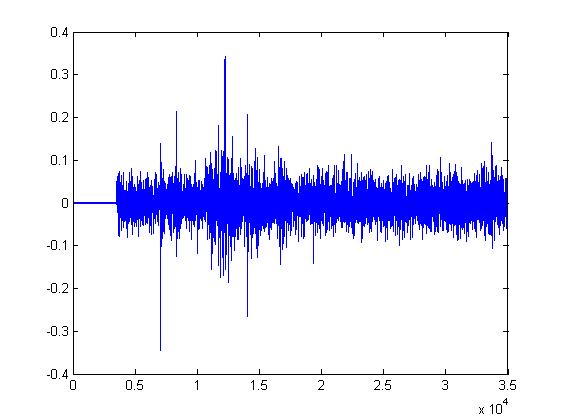
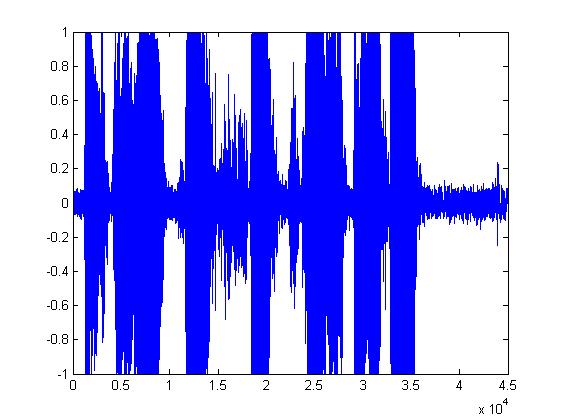
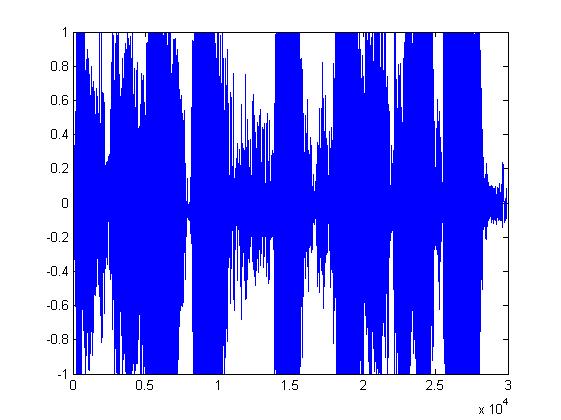
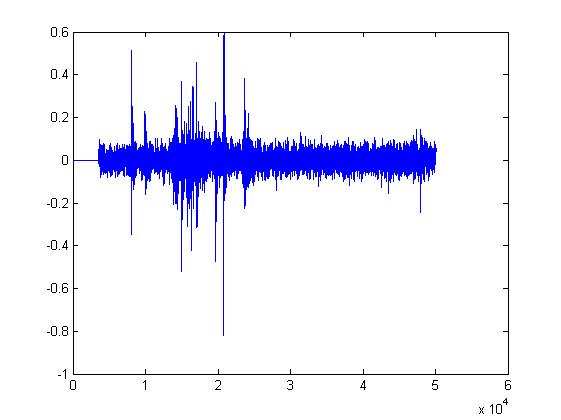
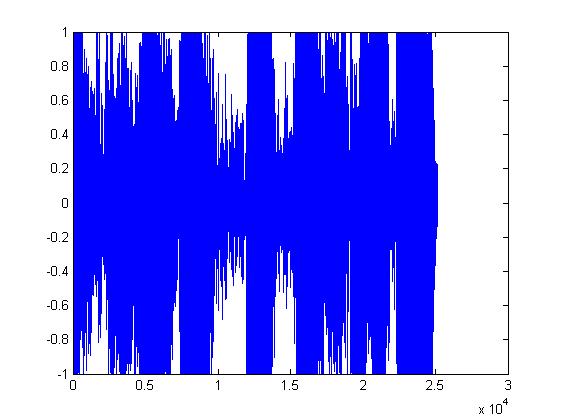
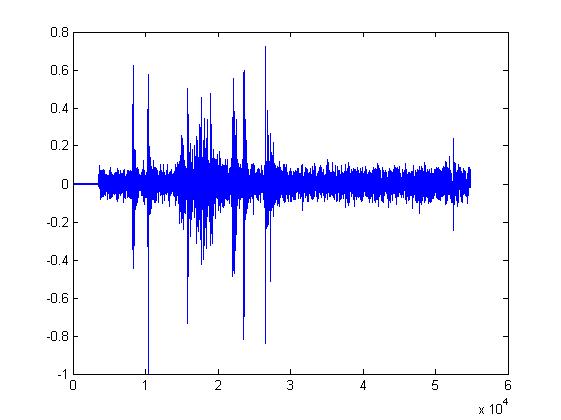
* Com dp\_multiplier=2

Resultados com wlen=300, step=300, frac=0.6, conta\_amostras\_fala=1:

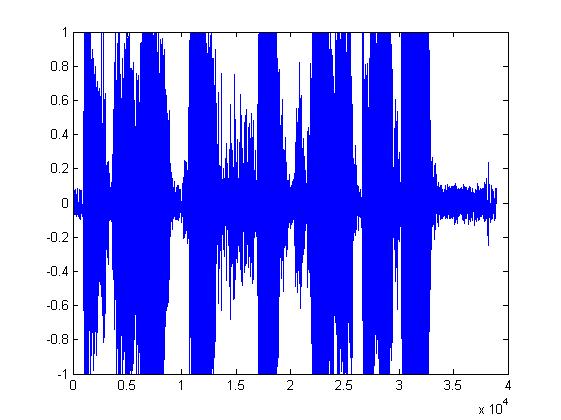
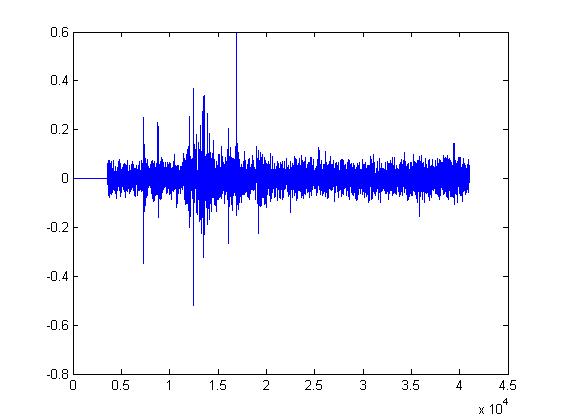
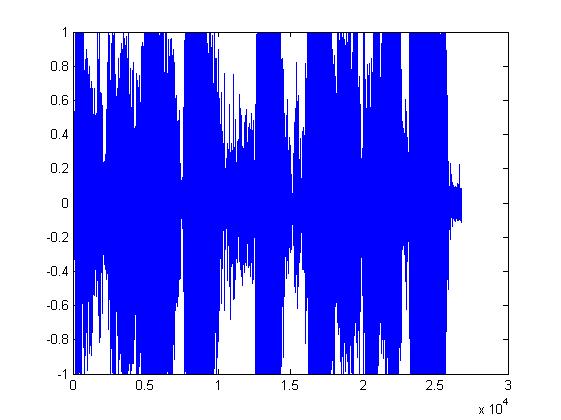
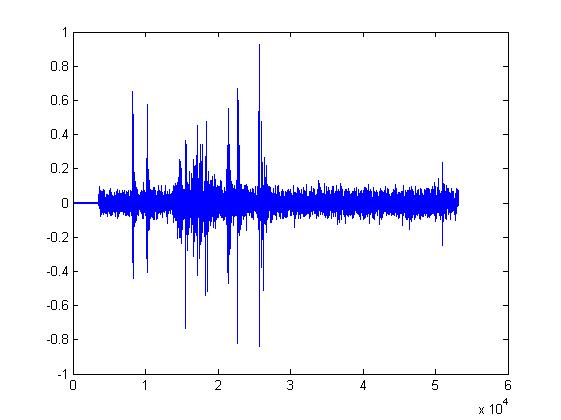
* Com dp\_multiplier=3

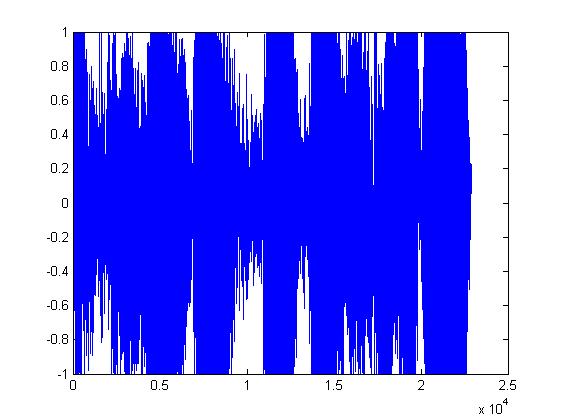
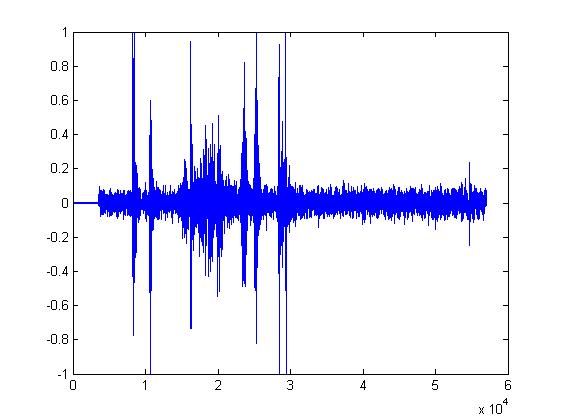
Resultados com wlen=300, step=300, frac=0.6, conta\_amostras\_fala=1:

Resultados com wlen=25, step=25, frac=0.7, conta\_amostras\_fala=1:

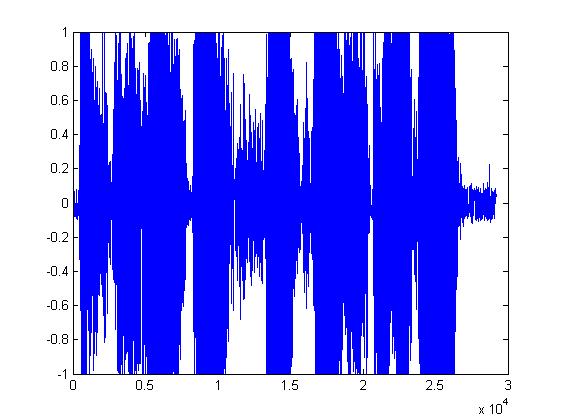
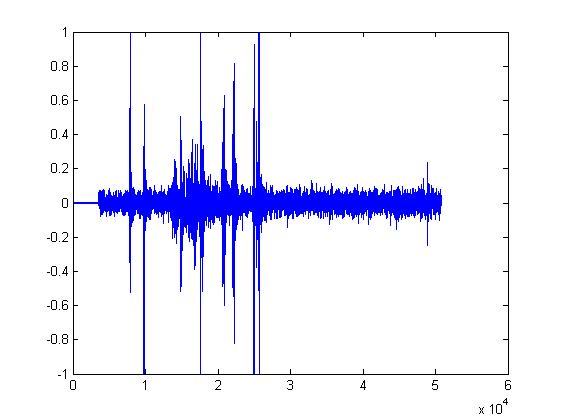
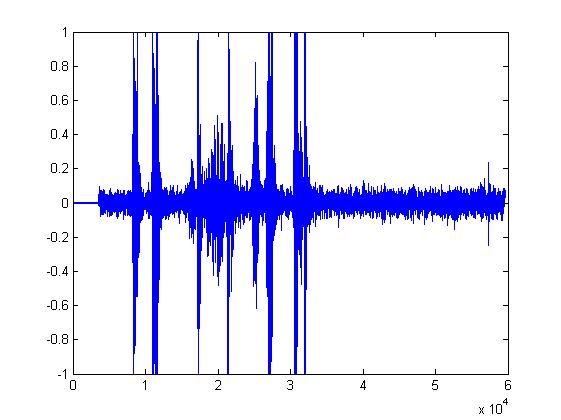
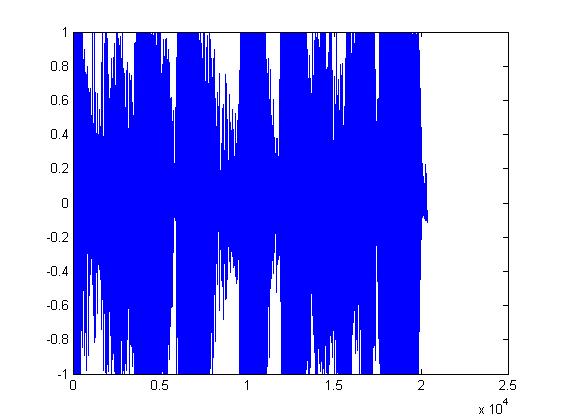
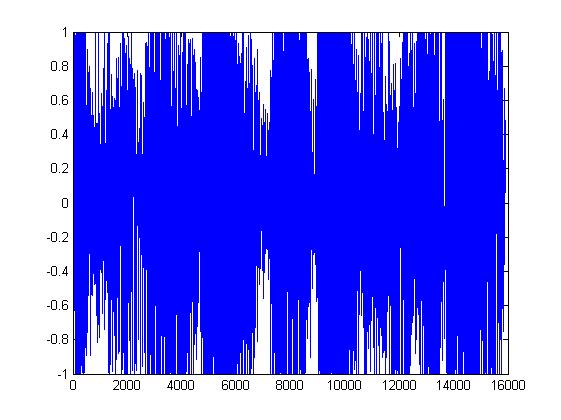
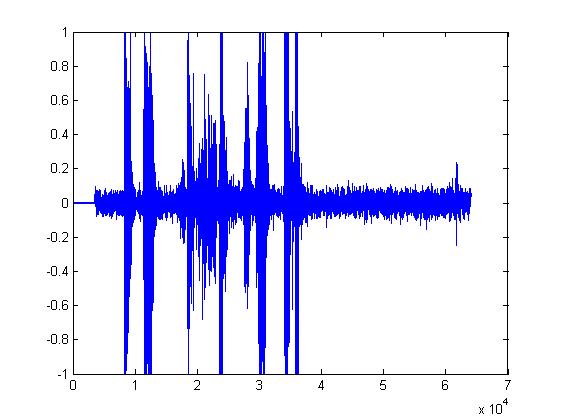
* Com dp\_multiplier=1
* Com dp\_multiplier=2
* Com dp\_multiplier=3

Resultados com wlen=25, step=25, frac=0.8, conta\_amostras\_fala=1:

* Com dp\_multiplier=1
* Com dp\_multiplier=2

* Com dp\_multiplier=3

Resultados com wlen=25, step=25, frac=0.9, conta\_amostras\_fala=1:

* Com dp\_multiplier=1
*  Com dp\_multiplier=2
* Com dp\_multiplier=3

Conclusões relativas à função:

É uma função que cumpre satisfatoriamente o exigido, no entanto é preciso ter em conta a parametrização que se aplica, pois, podemos ter resultados bastante díspares e até inesperados. Note-se que o campo conta\_amostras\_fala é colocado a 1, pois não houve necessidade de melhorar/adaptar a análise do sinal (para criação do array de análise). Temos ainda o parâmetro wlen que assume por opção apenas o valor que também é passado ao step, isto quer dizer que os pontos/amostras de uma janela são analisados uma única vez. Por último dp\_multiplier é também muito importante pois permite optimizar a análise em alguns sinais, no entanto devemos ter cuidado, uma vez que com esta tentativa de optimização há sempre o risco de perder componentes de fala, coisa que acontece com relativa facilidade.

Conclusões acerca da Optimização de sinais:

Pela análise de todos os testes acima expostos, bem como outros a ser enviados em anexos, as conclusões são algumas, embora se tenha de realçar o grau de incerteza de algumas, essencialmente no que toca à parametrização da função, visto que aqui, pisa-se um pouco uma linha ténue entre o desejado e o que é realmente conseguido.

A primeira conclusão reflecte que existe a necessidade de por vezes, de palavra para palavra, variarmos o tipo de modelo de ruído a ser construído, uma vez que, por vezes (a maioria), é de melhor tom escolher um modelo de ruído mais centralizado (com desvio-padrão x 1) pois pela variação dos restantes parâmetros, se consegue colmatar as deficiências de alguns sinais, no entanto por vezes para maior optimização consegue-se melhores resultados com um modelo mais disperso (k x desvio-padrão).

Qual o problema de um modelo mais disperso?

Essencialmente é que, é muito fácil perder componentes de fala válidas, isto é, é fácil perder dados válidos, sendo que isso é algo a evitar pois em qualquer área e em qualquer tipo de trabalho de optimização, o que se quer eliminar é o desnecessário e não o útil, no entanto em alguns casos, poder-se-á dizer que estes modelos mais dispersos são úteis, tudo depende da qualidade/fiabilidade final pretendida.

Outro aspecto refere-se ao tamanho da janela escolhido, na apreciação que foi feita, na maioria das vezes é de bom tom escolher uma janela em torno dos 150 a 300, pois embora possa parecer menos preciso, nos testes realizados, revelou-se uma forma facilitada de se obter bons resultados, pois o ruído era bastante bem detectado, tal como seria de esperar e para além disso não há informação útil(fala) a ser posta em risco e o mais importante é que dependendo da aplicação e da restante parametrização, o esforço computacional pode ser muito menor, e muitas vezes isto é algo a ter em conta.

Para além disso é importante reflectir no caso da função 2 em certos parâmetros:

1º - wlen e nseg – neste caso à escolha foi dar o mesmo valor por simplificação e também para diminuição do esforço computacional, embora para maior rigor(à partida), step devia ser inferior(tipico metade de wlen), para haver uma avaliação de amostras mais rigorosa;

2º - O método de avaliação escolhido – não é o ideal, mas é certamente bastante intuitivo e simples, e cumpre de uma forma satisfatória o que foi definido, embora por exemplo um método do tipo recursivo pudesse ser também bastante bom (em teoria).

3º - A falta de uma verificação mais rigorosa por uma ferramenta já desenhada, torna o trabalho de experimentação/audição bastante incerto, pois o ideal para uma pessoa, pode não o ser para outra, isto porque o diferencial de qualidade que se pode atingir com esta função é relativamente alargado, daí ser importante a análise individual e a consequente parametrização realizadas.

De um modo geral com a optimização realizada, caso a parametrização seja bem realizada os resultados são satisfatórios e até quase desejados em alguns casos.

Conclusões Gerais:

Apesar de muito pouco tempo de realização do trabalho, pois foram apenas poucos dias, visto haver trabalhos também eles pendentes a outras disciplinas, bem como houveram exames de recurso, mostrou-se um trabalho muito interessante e útil, embora a vontade fosse que o estudo fosse mais aprofundado e reflectido, este foi o estudo possível.

Mostrou-se um trabalho interessante pelo próprio conceito em geral, o de optimização de um sinal (eliminação de ruído), bem como pelo trabalho de programação, bastante semelhante por exemplo à linguagem C, embora com algumas diferenças notórias, e por último pela maneira eficaz com que este trabalho se ligou à última matéria abordada na disciplina, que por si só teria sido pouco compreendida e até não trabalhada.

Desta forma penso que o saldo é positivo, embora reforce a ideia de que o grau de exigência de execução do trabalho, fosse difícil de ser respeitado, devido à falta de tempo existente, e por isso os resultados não são devidamente fundamentados com mais testes e avaliações.