



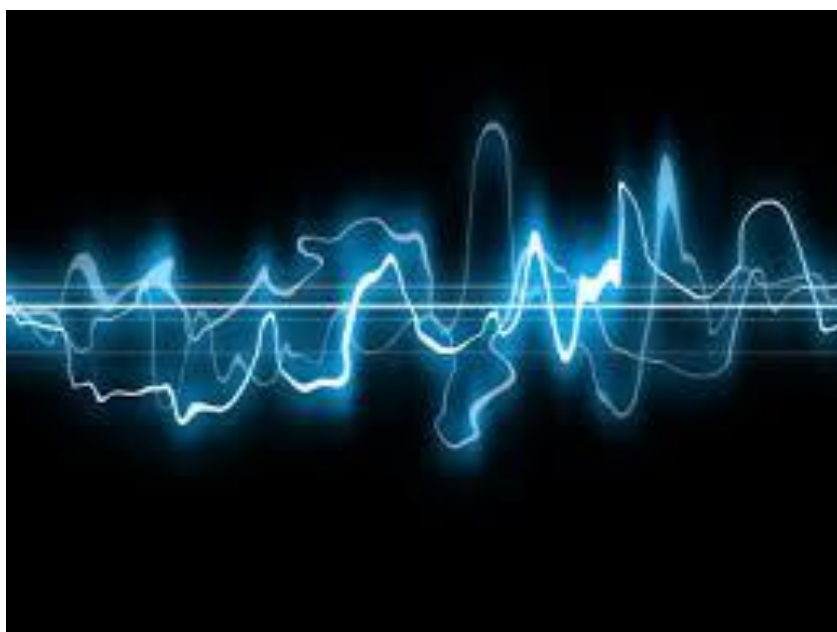
Universidade do Minho

Processamento Digital de Sinal

Trabalho Nº2

2012/2013

Separação do ruído da fala



Trabalho Realizado Por: César Monteiro A61989

Índice

Introdução.....	3
1. Ruído Branco.....	3
2.Modelo do Ruído.....	4
3.Processamento de Sinal.....	5
4.Problemas.....	5
5.Função.....	5
6.Resultados Obtidos.....	7
6.1Discussão dos Resultados.....	15
Bibliografia.....	14
Conclusão.....	16

Introdução

O objectivo deste trabalho será analisar um ficheiro de áudio e separar o ruído da fala.

Pretende-se criar uma função em Matlab capaz de resolver o problema supra citado.

Através de processos estocásticos é possível calcular o modelo de ruído (variância, media). Tendo estes parâmetros é fácil saber se a amostra faz parte do ruído ou da fala, através da curva de Gauss, verifica-se se a amostra se encontra dentro ou fora do “outlier” se estiver fora é fala se estiver dentro é ruído.

Na componente prática deste trabalho realizei experimentações, variando a janela, o número de amostras e simulando algumas formas de ruído.

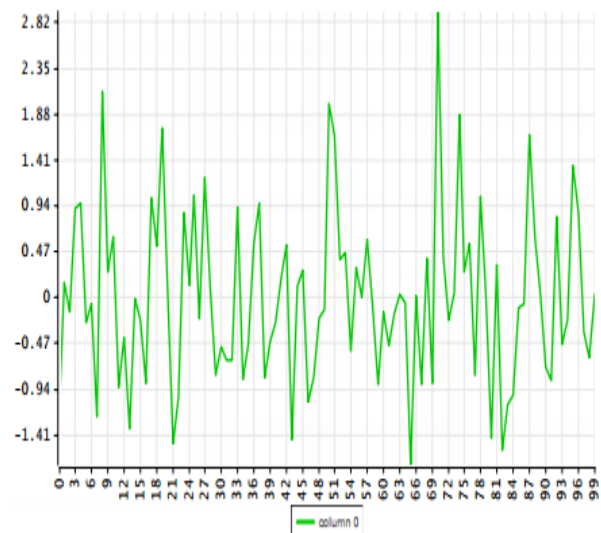
1. Ruído Branco

O processo Ruído Branco: é um processo estocástico onde todas as variáveis aleatórias seguem uma distribuição normal de média zero, variância constante e as covariâncias são nulas.

O valor esperado é constante:

- A variância é constante
- Simétrica
- Não correlacionados entre instantes

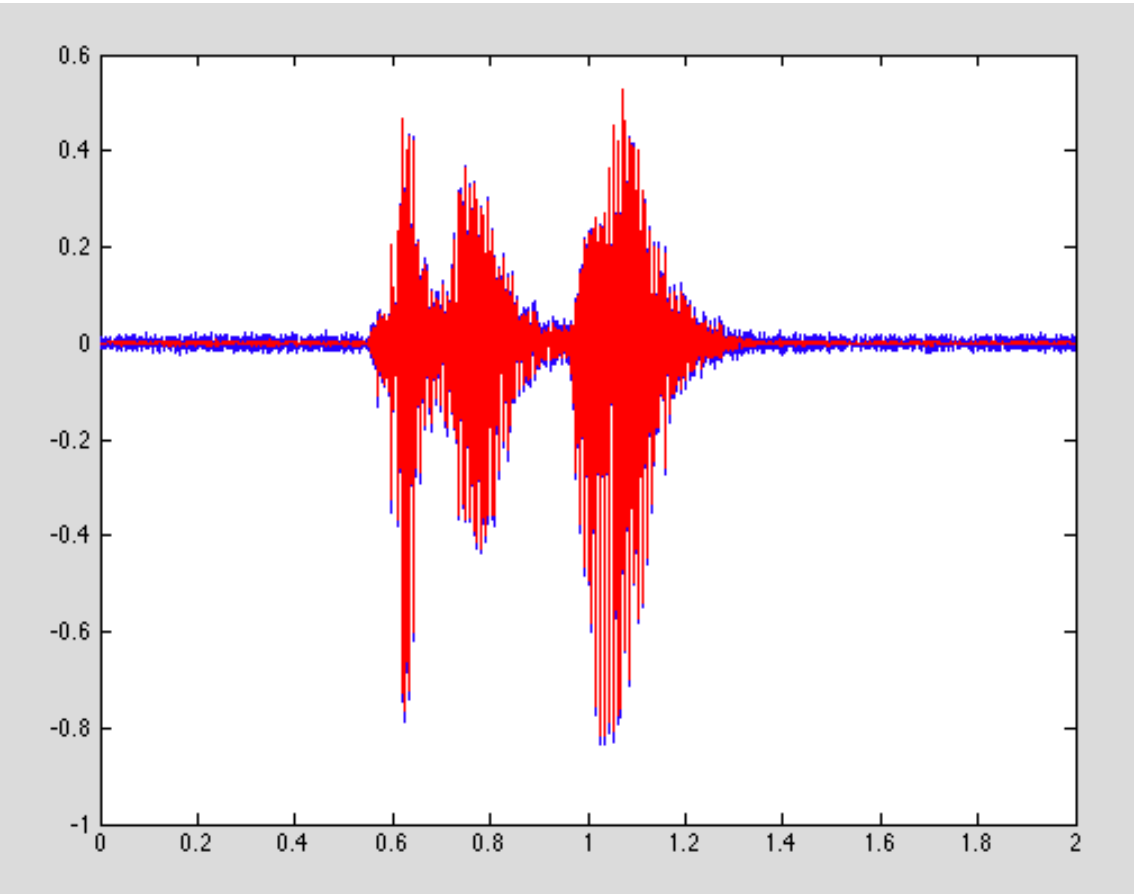
diferentes



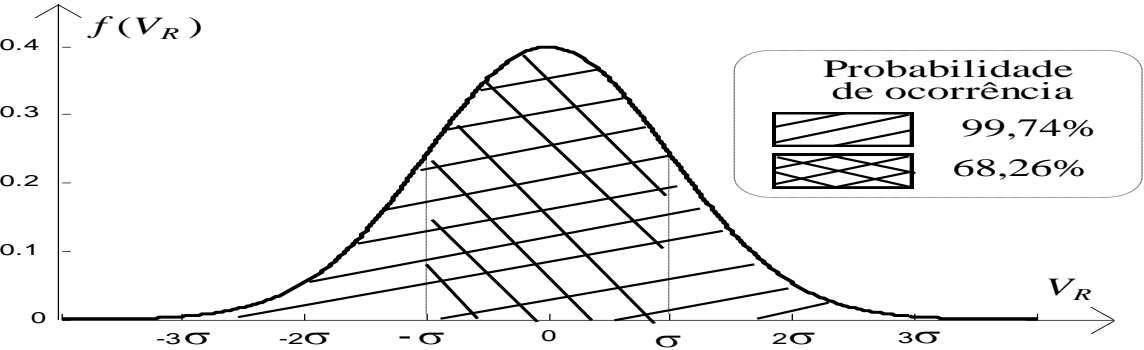
2. Modelo do ruído

Através do modelo do ruído seremos capazes de separar o ruído da fala, através do cálculo da variância e da média do ruído, geralmente estes parâmetros são retirados e calculados sobre os primeiros índices da amostra.

Ex



Com o modelo do ruído, ao fim de calculados os parâmetros irá ser possível separar o ruído da fala. Se a amostra der dentro do “outlier” é ruído, se der fora é fala.



3. Processamento de sinal

Primeiro irá proceder-se à aquisição do sinal através da função Wavrecord().

Depois para poder realizar o pretendido foi criada um função que irá calcular a média e a variância do ruído(amostras iniciais), seguidamente a função irá analisar o sinal de áudio e separar o ruído da fala. Criando dois vectores novos, um para a fala e outro para o ruído. Segue-se o cálculo do SNR .

4. Problemas

- Os pontos do sinal que correspondem à fala também podem estar dentro dos parâmetros do Modelo do Ruído;
- Pode existir um barulho de fundo durante um curto período de tempo e é confundido com fala.

Para solucionar estes problemas analisa-se o sinal, segmento a segmento e verifica-se se nesse tempo houve mais amostras cujos valores estavam dentro ou fora dos parâmetros do Modelo do Ruído.

Se houver mais amostras dentro é ruído, senão é fala.

5. Função:

```
function [noisebuff, buff] = mruido(x, nseg, wlen, fr)
//nseg-nº de amostras do segmento; wlen - tamanho da janela;
//fracção de '1's (amostras no Outlier)
m=mean(x(1:nseg)); //média
v=var(x(1:nseg)); //variância
alfa=fr*wlen;
buff=0; //buffer de fala
noisebuff=0; //buffer de ruído
a = 1:1:length(x);
if(rem(wlen,2)==1)
wlen=wlen+1;
end
for i=1: length(x) // sinal todo
if(abs(x(i)-m)>sqrt(v)) //se a amostra estiver no outlier
a(i)=1;
else //se a amostra estiver dentro dos limites
a(i)=0;
end
end
initw=1;
endw=initw+wlen;
prev_noise=0;
while(endw<length(x))
w=a(initw:endw);
if(sum(w)>alfa) //se houver mais amostras no outlier do que dentro
if(prev_noise==1) //se anteriormente foi ruído
buff=cat(1, buff, x(initw:endw));
else //se anteriormente foi fala
buff=cat(1, buff, x(initw+wlen/2:endw));
end
else //se houver mais amostras dentro do que no outlier
if(prev_noise==1) //se anteriormente foi ruído
noisebuff=cat(1, noisebuff, x(initw+wlen/2:endw));
```

```

else //se anteriormente foi fala
noisebuff=cat(1, noisebuff, x(initw:endw));
end
end
initw=initw+wlen/2;
endw=endw+wlen/2;
if(endw>length(x)) //se a janela passou dos limites do sinal
endw=length(x);
initw=endw-initw;
end
end

subplot(3,1,1), plot(x);
subplot(3,1,2), plot(buff);
subplot(3,1,3), plot(noisebuff);

clean=mean(buff.^2);

noise=mean(noisebuff.^2);

SNR=20log(clean/noise)

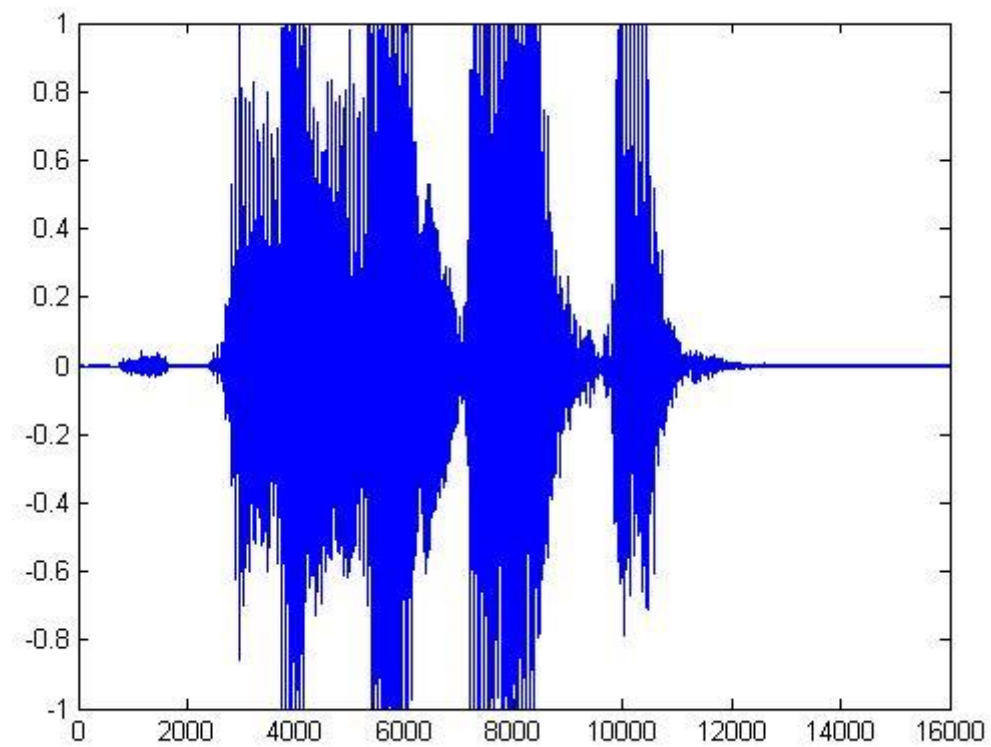
```

6. Resultados Obtidos

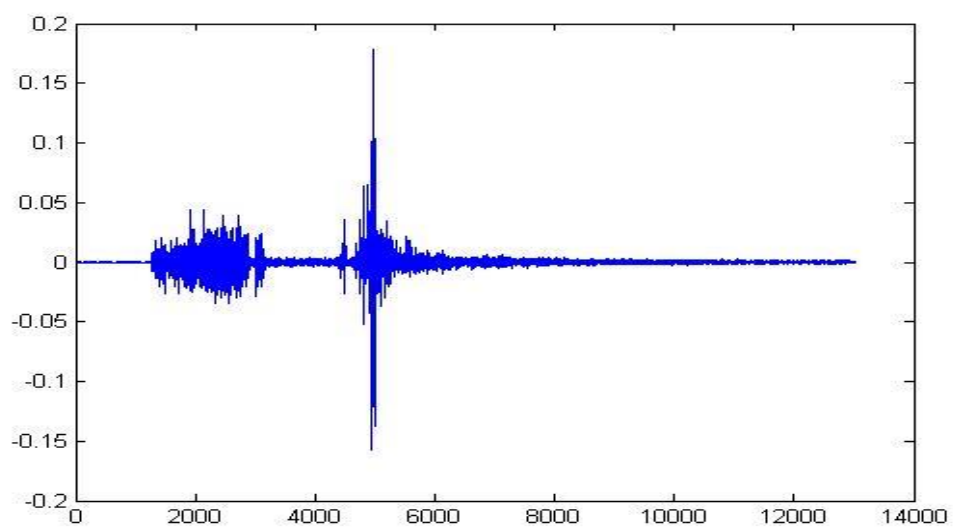
1ª Amostra

```
[noisebuff, buff] = mruido(x, 1500, 500, 2/3);
```

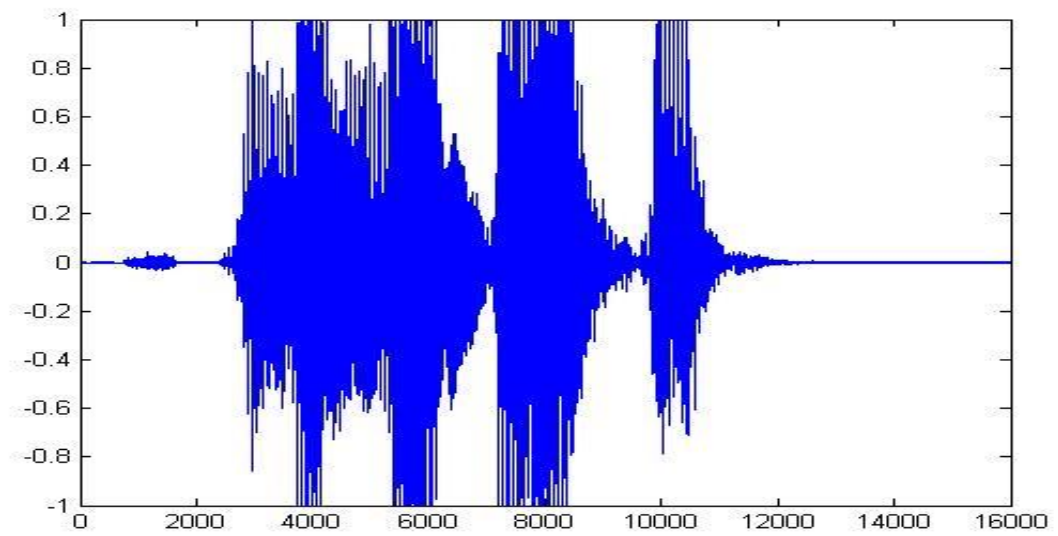
Sinal Recolhido



Ruído retirado do sinal



Sinal sem ruído



SNR=66.69 DB

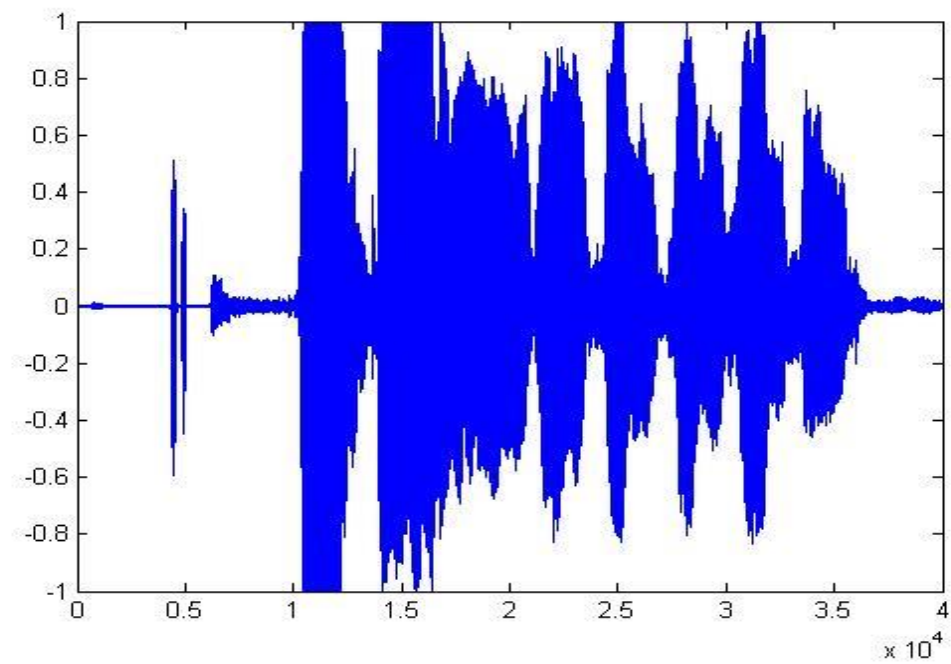
Com a amostra anterior, alterou se o número de uns para 9/10 e conseguimos um SNR de 80 DB

2ª Amostra

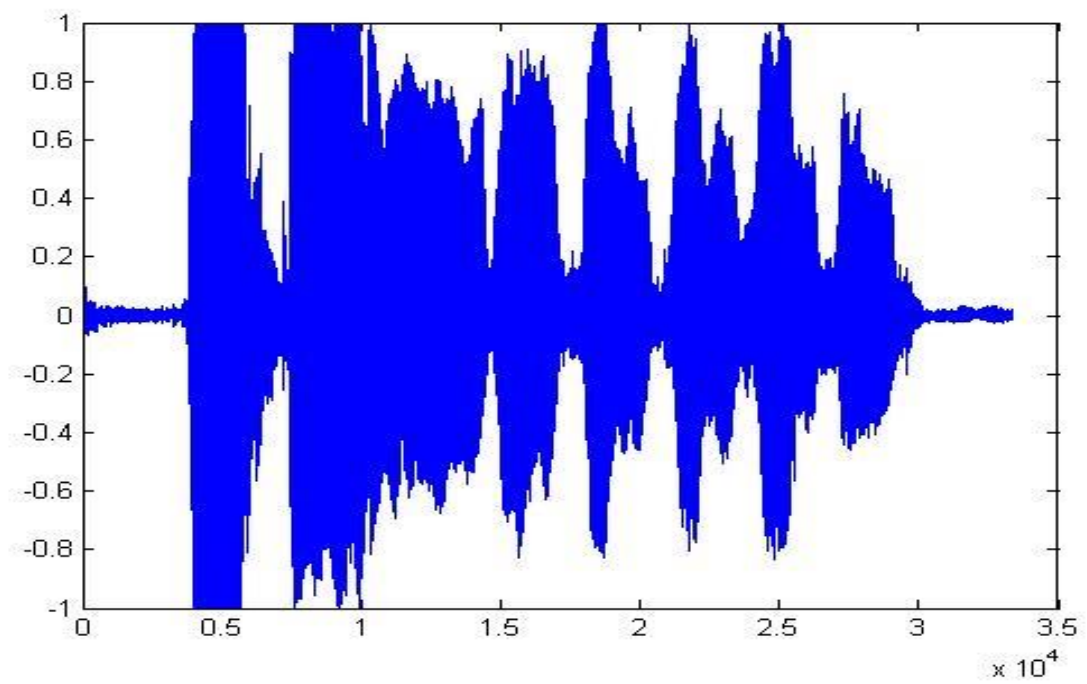
Cair de objecto->Fala

```
[noisebuff, buff] = mruido(x, 1500, 500, 2/3);
```

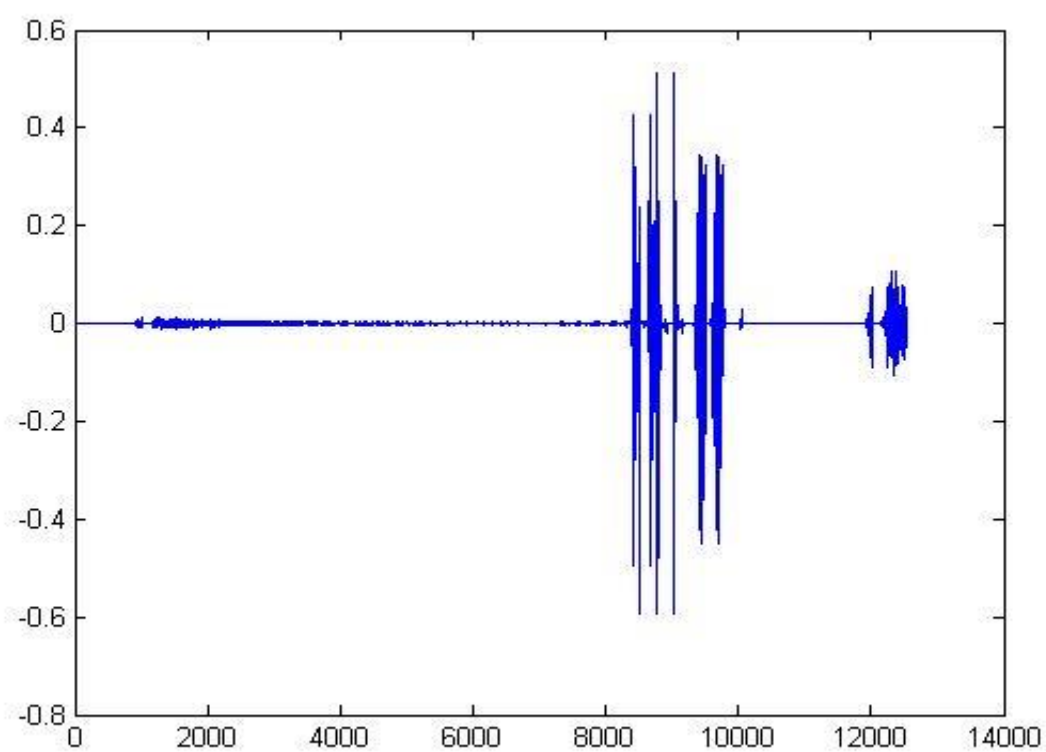
Sinal Recolhido



Sinal Limpo



Ruido do Sinal

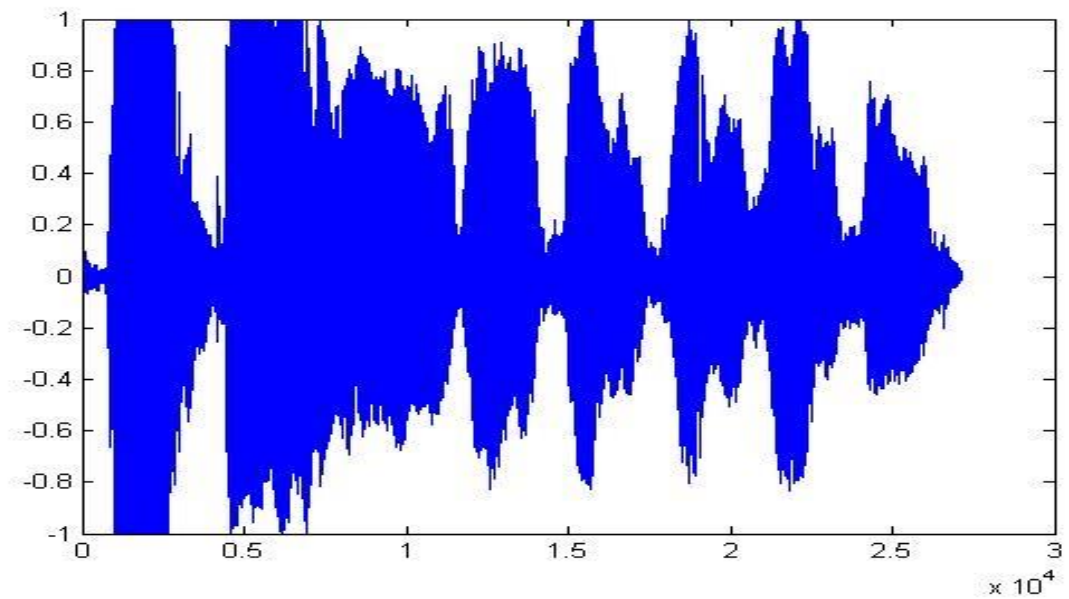


SNR=78 DB

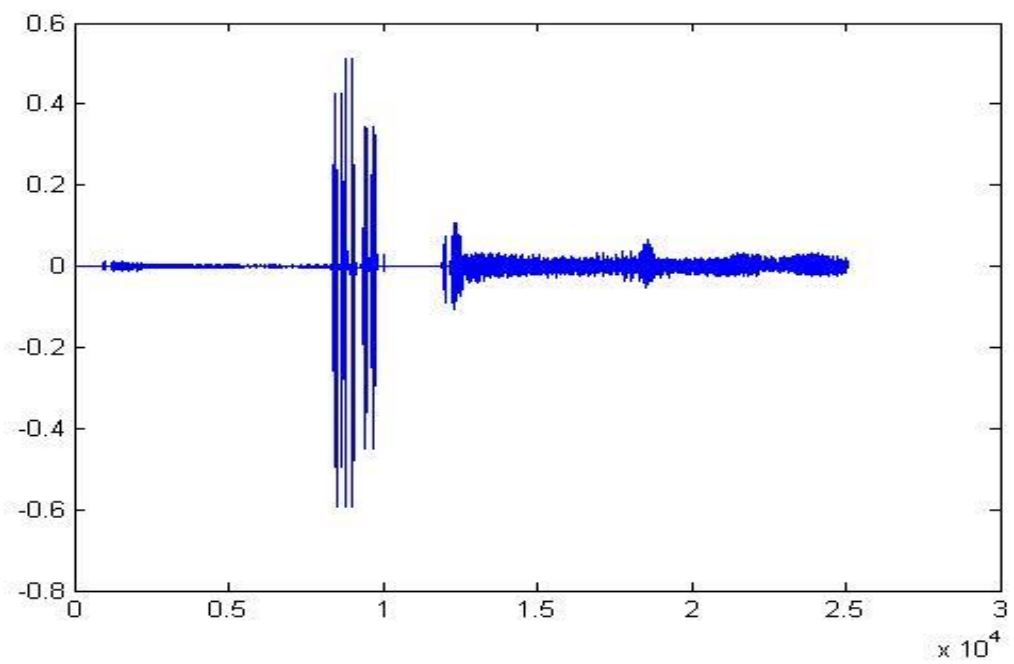
Aumentando o número de segmentos e o número de 1 de cada segmento para o mesmo sinal

```
[noisebuff, buff] = mruido(x, 2000, 500, 9/10);
```

Sinal Limpo



Ruido

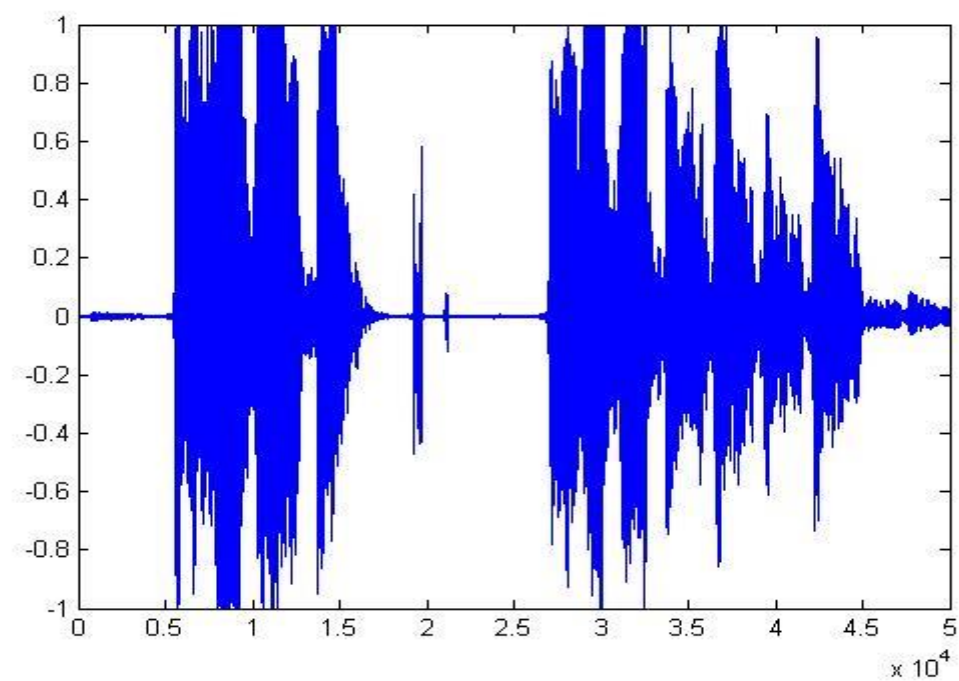


SNR=95 DB

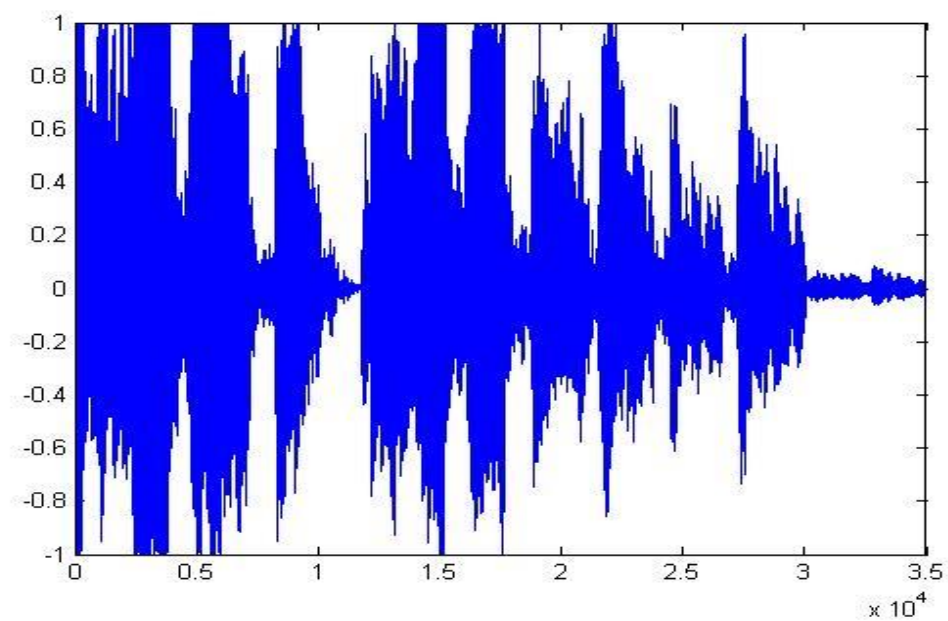
3ª Amostra

Fala -> Cair de objecto-> Fala

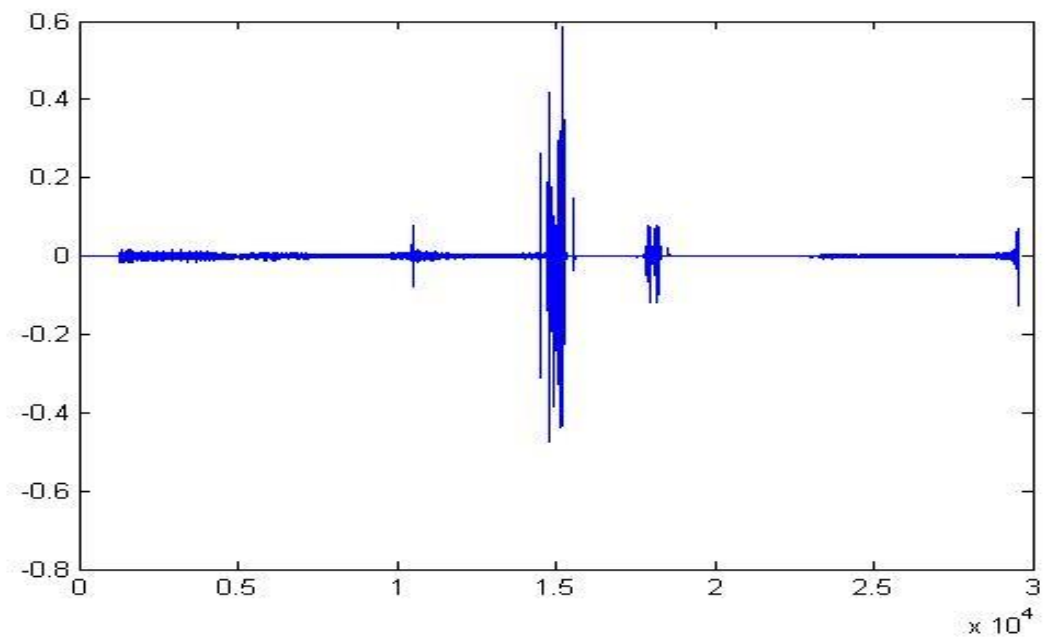
Sinal



Sinal Limpo



Ruído

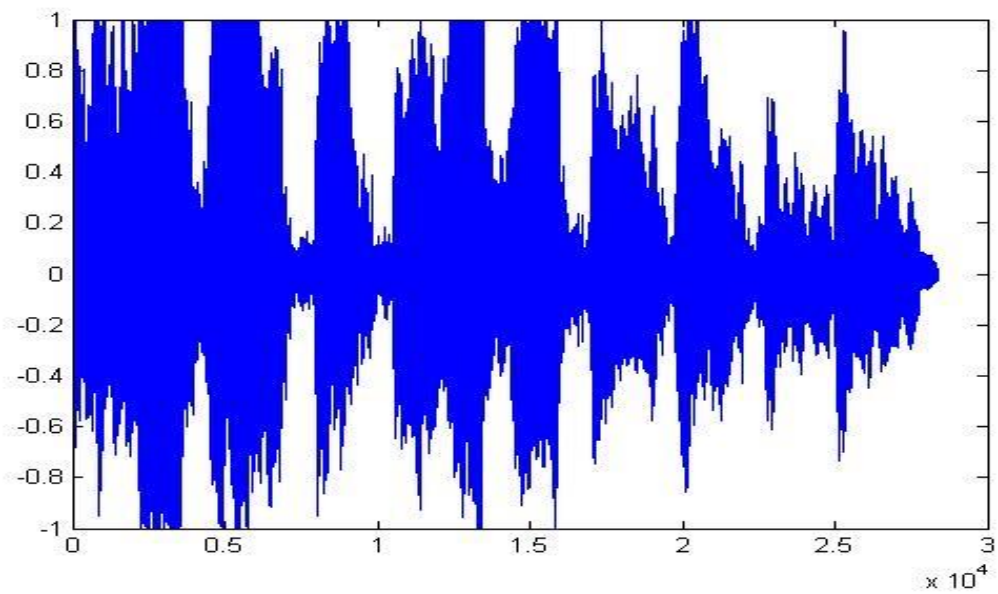


SNR=60 DB

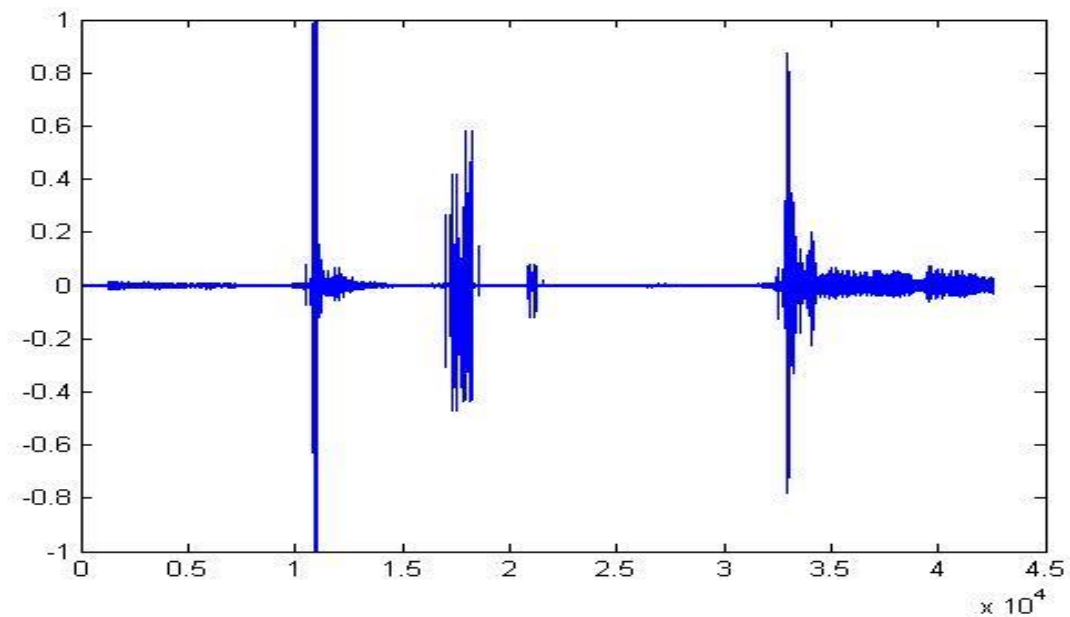
Aumentando o número de 1 da janela

```
[noisebuff, buff] = mruido(x, 2000, 500, 95/100);
```

Sinal Limpo



Ruído



SNR=76 DB

Resultados Obtidos

1ª Amostra

Na 1ª amostragem pode-se verificar que foi possível separar o ruído da fala mas mesmo assim verifica-se algum ruído no sinal filtrado. Ao aumentar a percentagem de '1s' que cada segmento deve ter verificou-se a diminuição do ruído no vector do sinal limpo.

Verificou-se ainda um aumento do SNR quando aumentamos a percentagem de '1s'.

2ª Amostra

Na 2ª amostragem, conseguiu-se separar a componente da queda do objecto da fala, mas mesmo assim, o sinal de fala continha algum ruído. Procedeu-se ao aumento da percentagem de '1s' e conseguiu-se diminuir o ruído no sinal de fala.

Verificou-se ainda um aumento do SNR quando aumentamos a percentagem de '1s'.

3ª Amostra

Na 3ª amostragem simulou-se a fala, de seguida a queda de um objecto e outra vez a fala.

Foi possível separar a fala da componente da queda do objecto e de quase todo o ruído. Procedeu-se ao aumento da percentagem de '1s' e conseguiu-se eliminar a maior parte do ruído da fala.

Verificou-se ainda um aumento do SNR quando aumentamos a percentagem de '1s'.

6.1 Discussão dos resultados obtidos

Foi possível separar a fala do ruído, com base nos testes feitos é possível também desprezar outras formas de ruído (barulho provocado por queda de objectos) com êxito.

Com base nas 3 amostragens é possível afirmar que quanto maior a percentagem de '1s' de cada segmento melhor será o sinal de fala (irá conter menos ruído).

Ao aumentar a percentagem '1s' iremos aumentar o SNR.

Conclusão

É possível concluir que se pode separar de um sinal o ruído e fala, através do modelo do ruído.

Com o aumento da percentagem de '1s' é possível concluir, que irá ter uma grande influência para que o sinal da fala contenha o menor ruído possível. Sempre que aumentava-mos a percentagem de '1s' aumentava-mos o SNR, podendo assim afirmar que a percentagem de '1s' tem influência no SNR.

Por isso conclui-se que alterando os parâmetros da função é possível melhorar o resultado final.

Através deste método é possível enviar sinais com menor tamanho, pois o ruído não ajuda em nada o processamento do mesmo, conseguindo assim diminuir o tamanho do sinal, e envia-lo e processa-lo mais rápido.

O uso do Matlab facilitou esta demonstração, tendo assim conseguido chegar a estas conclusões. Revelou-se uma ferramenta importante na área de processamento de sinal.

Bibliografia

Aulas de Processamento digital de sinal

http://en.wikipedia.org/wiki/Signal-to-noise_ratio

http://pt.wikipedia.org/wiki/Ru%C3%ADdo_branco