PSI 3531 - Processamento de Sinais Aplicado

Experiência 2

Codificação de sinais de voz por análise preditiva

Alunos: Gabriel Moraes da Cruz e Talita Ferreira Abdulmassih

Números USP: 10335020 e 9838272

Professor: Vítor H. Nascimento

16 de maio de 2022

Este relatório segue as questões propostas nas partes experimentais do documento *Codificação de sinais de voz por análise preditiva*. O código desenvolvido em Matlab para a resolução do exercício é apresentado no final deste documento.

Primeira Parte Experimental

1. Para um trecho do sinal de voz de 240 amostras, os parâmetros de um modelo LPC com 10 coeficientes foram calculados. Na tabela abaixo, tem-se os valores dos coeficientes a_1, a_2,..., a_{10}.

a_1	-0.9688
a_2	-0.2256
a_3	0.1966
a_4	0.4494
a_5	-0.0673
a_6	-0.5598
a_7	0.1325
a_8	0.5139
a_9	0.0094
a_{10}	-0.2692

Tabela 1: Valores obtidos para os parâmetros de um modelo LPC com 10 coeficientes.

2. Considerando que o sinal de 240 amostras é surdo, tem-se que o ganho G é dado por:

$$G = \sqrt{\xi_n} = \sqrt{0.0160} = 0.1266$$

Na Figura 1, é possível observar a resposta em frequência do filtro H(z) calculado anteriormente com ganho G.

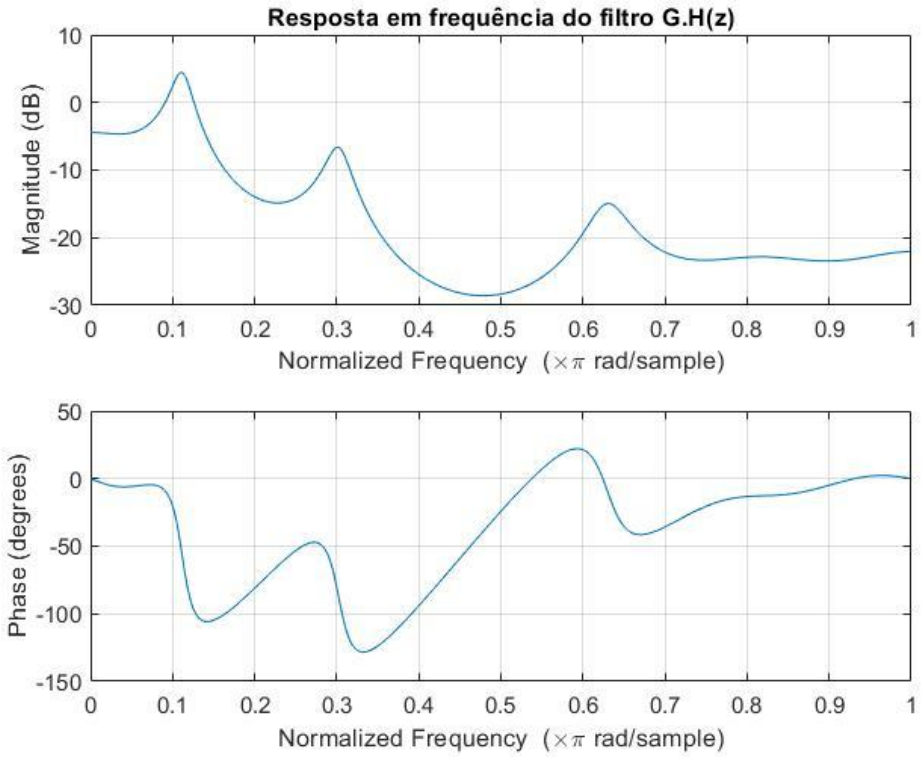

Figura 1: Resposta em frequência do filtro obtido.

3. Uma estimativa da transformada do trecho de sinal de voz e a resposta em frequência do filtro G.H(z) são exibidos na Figura 2. Nela, é possível observar que o modelo LPC obtido acompanha a envoltória da transformada do sinal, o que valida o resultado obtido.

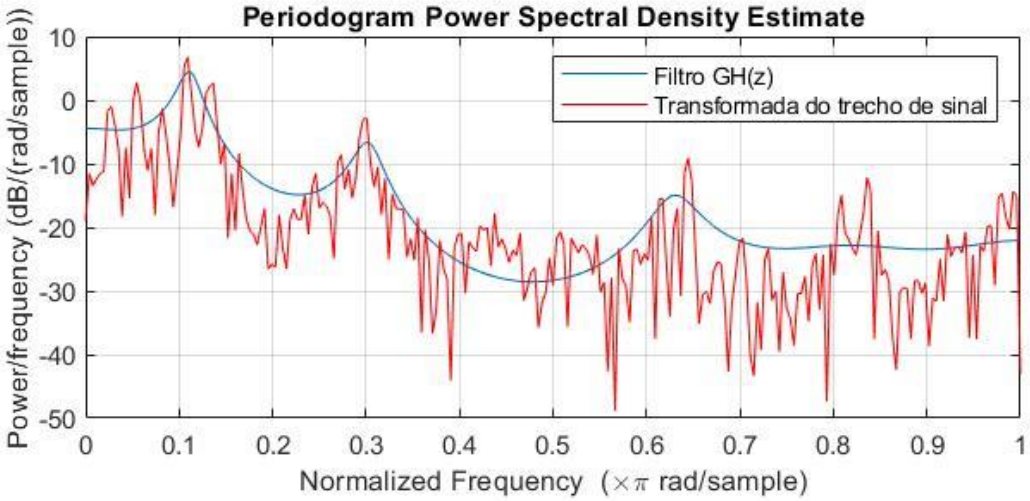

Figura 2: Módulo da resposta em frequência do filtro G.H(z) em dB e estimativa da transformada do trecho de sinal de voz.

4. Na Figura 3, é possível observar que, quanto mais aumentamos a ordem do modelo LPC, mais a resposta em frequência do filtro obtido se aproxima da estimativa da transformada do trecho de sinal.

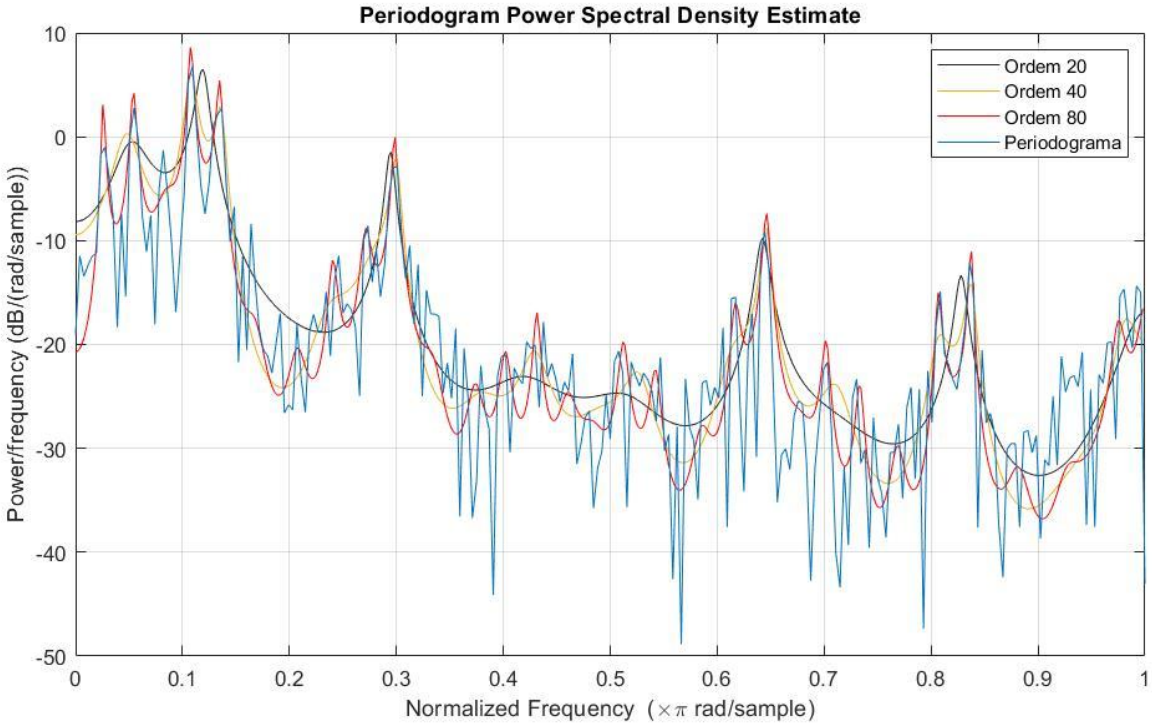

Figura 3: Comparação entre o periodograma do trecho de sinal de voz e a resposta em frequência do filtro GH(z) obtido para diferentes ordens do modelo LPC.

5. Na Figura 4, é possível observar um dos gráficos obtidos com o uso da função *yaapt.m*.

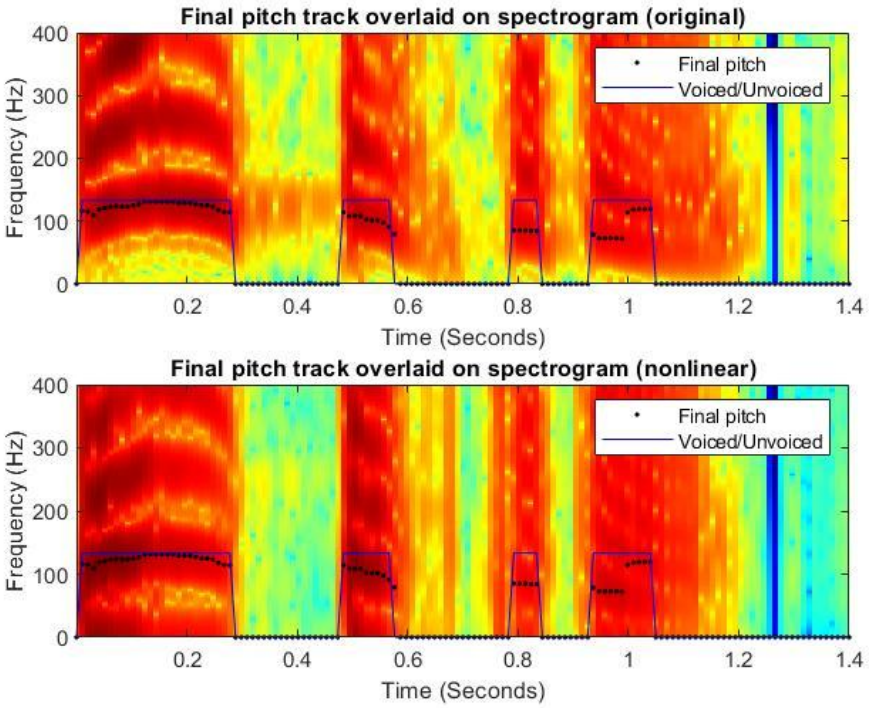

Figura 4: Espectrograma do sinal de voz, valores estimados de *pitch* e detector de sinal sonoro ou surdo.

6. Observa-se nas Figuras 5 e 6 a comparação entre o sinal original e os sinais de saída, no caso em que os coeficientes não estão e estão quantizados, respectivamente.

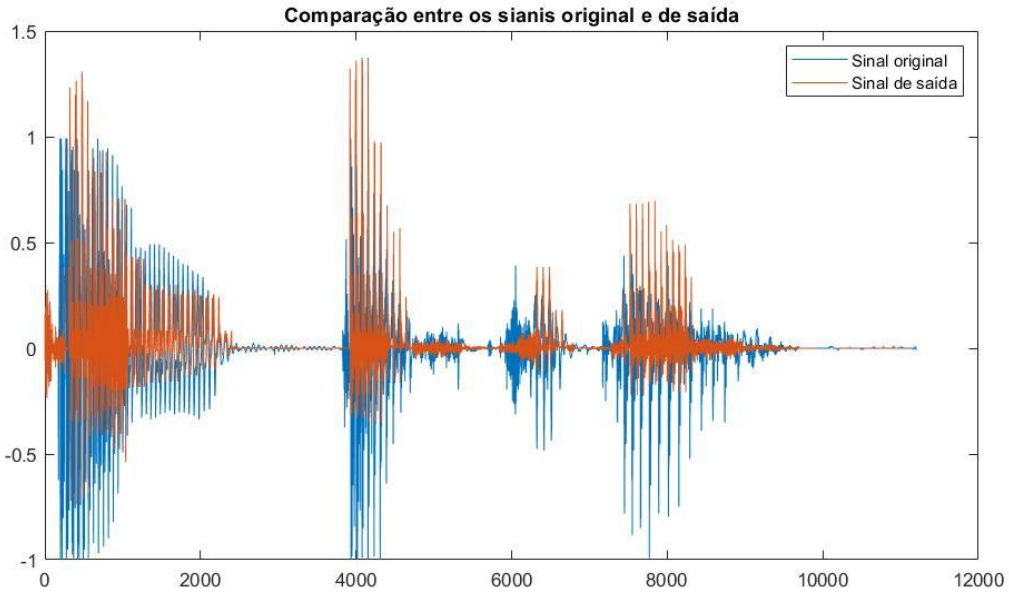

Figura 5: Comparação entre a entrada e a saída do programa para o caso em que os coeficientes não são quantizados.

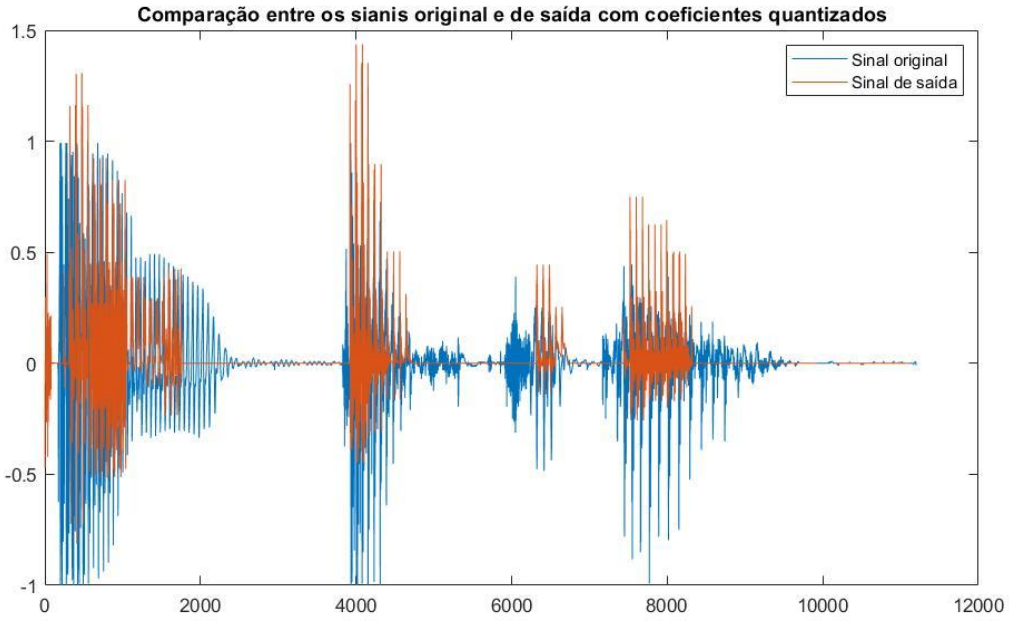

Figura 6: Comparação entre a entrada e a saída do programa para o caso em que os coeficientes são quantizados.

Quando o codificador é quantizado nas condições do item *g*, tem-se 7 bits para cada um dos 10 coeficientes do modelo LPC a cada 30 ms, e 10 bits de ganho e período de *pitch* a cada 10 ms. Nesse caso, calcula-se a taxa de bits por segundo do codificador como:

$$\frac{70}{0{,}03} + \frac{10}{0{,}01} = 3\,333,3\ bits/s$$

Portanto, a taxa conseguida nessas condições para o codificador é de, aproximadamente, 3,3 kbits/s.

Parte Experimental: Análise por síntese

Observa-se nas Figuras 7 e 8 a comparação entre o sinal original e os sinais de saída, no caso em que os coeficientes não estão e estão quantizados, respectivamente.

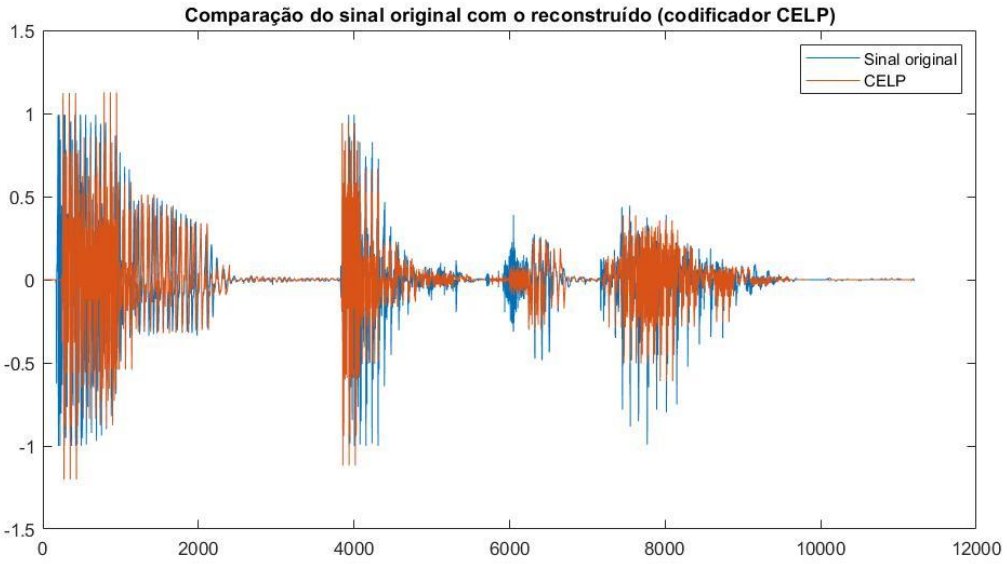

Figura 7: Comparação entre a entrada e a saída do programa para o caso em que os coeficientes não são quantizados e utiliza-se o codificador CELP.

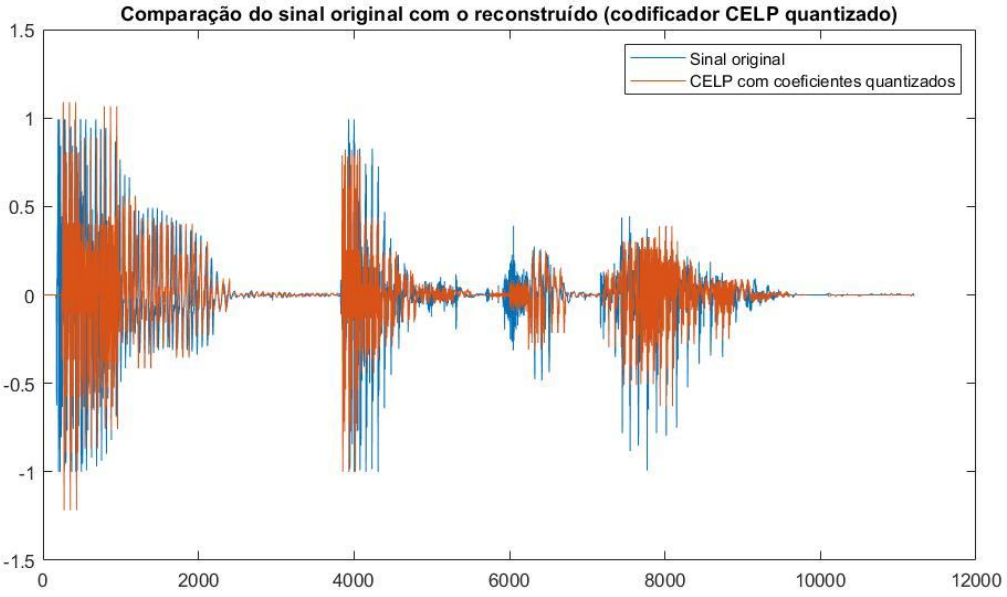

Figura 8: Comparação entre a entrada e a saída do programa para o caso em que os coeficientes são quantizados e utiliza-se o codificador CELP.

Quando o codificador é quantizado nas condições do item *i*, tem-se 7 bits para cada um dos 10 coeficientes do modelo LPC a cada 30 ms, e 10 bits armazenam dois valores de ganho a cada 10 ms. Além disso, são necessários 9 bits para transmitir cada um dos dois índices a cada 10 ms. Nesse caso, calcula-se a taxa de bits por segundo do codificador como:

$$\frac{70}{0{,}03} + \frac{10}{0{,}01} + \frac{18}{0{,}01} = 5\,133{,}3\ bits/s$$

Portanto, a taxa conseguida nessas condições para o codificador é de, aproximadamente, 5,1 kbits/s.

Ao comparar os resultados obtidos nas Figuras 7 e 8, nota-se que os ganhos obtidos para o codificador CELP quantizado resultam em amplitudes mais próximas às do sinal original. No momento de escolher os ganhos do livro-código é melhor, portanto, usar os coeficientes quantizados.

Códigos desenvolvidos em Matlab

Primeira parte experimental

```
close all;
clear all;
clc;

%% Parte experimental
% Lendo o sinal
[sinal, fs] = audioread('antarctica.wav'); %wavread

%% 1
% Cortando sinal e obtendo os parâmetros do modelo LPC com 10 coeficientes
sinal_cortado = sinal(200:439);

[ak, sig10]= lpc(sinal_cortado.*hamming(240), 10);

%% 2
% Calculando o ganho G
qsi = sig10;
G = sqrt(qsi);

%GH = G*H;
figure(5);
```

```
freqz(G,ak, 512);
% title('Resposta em frequência do filtro G.H(z)');

%% 3
% Calculando o periodograma
hold on;
periodogram(sinal_cortado,[],512);

legend('Filtro GH(z)', 'Transformada do trecho de sinal');

%% 4
% Aumentando a ordem do modelo LPC
figure(6);

% n = 20
[ak20, sig20]= lpc(sinal_cortado.*hamming(240), 20);
freqz(sqrt(sig20), ak20, 512);

%n = 40
hold on;
[ak40, sig40]= lpc(sinal_cortado.*hamming(240), 40);
freqz(sqrt(sig40), ak40, 512);

%n = 80
hold on;
[ak80, sig80]= lpc(sinal_cortado.*hamming(240), 80);
freqz(sqrt(sig80), ak80, 512);

hold on;
periodogram(sinal_cortado,[],512);

legend('Ordem 20', 'Ordem 40', 'Ordem 80', 'Periodograma');
```

```
%% 5
% plotando o Yaapt

% Figuras 1, 2, 3 e 4
pitch = yaapt(sinal, fs, 1, [], 1, 1);
pitch = [0 pitch];

%% 6
% Testando o codificador

% Calculando os parâmetros do modelo LPC
aks = zeros(11,46);
sigs = zeros(46,1);
Gs = zeros(138,1);
gerado = zeros(138*80,1);

% Gera coefiientes LPC para cada 30ms
for i = 1:46
   [aks(:,i), sigs(i)] = lpc(sinal(((i-1)*240 + 1):(i*240)).*hamming(240), 10);
end

% Calcula o valor de G para cada 10ms
for i = 1:138
   if (pitch(i) ~= 0) % Sinal sonoro
      Gs(i) = sqrt((8000/pitch(i))*sigs(ceil(i/3)));
   else % Sinal surdo
      Gs(i) = sqrt(sigs(ceil(i/3)));
   end
end

for i = 1:138
```

```
    if (pitch(i) ~= 0) % Sinal sonoro
        gerado((80*(i-1))+1:round(8000/pitch(i)):80*i) = 1;
        gerado((80*(i-1))+1:80*i) = filter(Gs(i), aks(:,ceil(i/3)), gerado((80*(i-1))+1:80*i));
    else % Sinal surdo
        aleatorio = randn(80,1);
        gerado(80*(i-1)+1:80*i) = Gs(i)*aleatorio*sqrt(var(aleatorio));
    end
end

exc_sinal_cortado = randn(80,1);
saida_sinal_cortado = G*exc_sinal_cortado*sqrt(var(exc_sinal_cortado));

gerado = [saida_sinal_cortado; gerado];

%Gerando gráfico

figure(7);
plot(sinal);
hold on;
plot(gerado);
legend(["Sinal original", "Sinal de saída"]);
title("Comparação entre os sianis original e de saída");

%% G

% Quantizando os coeficientes

Ba = 7;
Bg = 5;

Q_aks = zeros(11,46);
```

```
for i = 1:46
   Q_aks(:,i) = quantize3(aks(:,i), Ba);
end

Q_Gs = quantize3(Gs, Bg);
Q_pitch = quantize3(pitch, Bg);

Q_gerado = zeros(138*80,1);

% Repetino a codificação

for i = 1:138
   if (pitch(i) ~= 0) % Sinal sonoro
      Q_gerado((80*(i-1))+1:round(8000/Q_pitch(i)):80*i) = 1;
                        Q_gerado((80*(i-1))+1:80*i)    =    filter(Q_Gs(i),    Q_aks(:,ceil(i/3)),
Q_gerado((80*(i-1))+1:80*i));
   else % Sinal surdo
      aleatorio = randn(80,1);
      Q_gerado(80*(i-1)+1:80*i) = Q_Gs(i)*aleatorio*sqrt(var(aleatorio));
   end
end

exc_sinal_cortado = randn(80,1);
Q_saida_sinal_cortado = quantize3(G, Bg)*exc_sinal_cortado*sqrt(var(exc_sinal_cortado));

Q_gerado = [Q_saida_sinal_cortado; Q_gerado];

figure(8);
plot(sinal);
hold on;
plot(Q_gerado);
legend(["Sinal original", "Sinal de saída"]);
```

```
title("Comparação entre os sianis original e de saída com coeficientes quantizados");


%% RESULTADO
%Para ouvir use:
% sound(gerado, fs)
% sounda(Q_gerado, fs)

%gerando arquivos .wav
audiowrite("antarctice_Qgerado.wav", Q_gerado, fs);
audiowrite("antarctica_gerado.wav", gerado, fs);
```

Parte experimental: Análise por síntese

```
close all;
clear;
clc;

% Definições
L = 240;
N = 80;
Q = 512;
K = 2;
gama = 1;
[sinal, fs] = audioread('antarctica.wav');

%% 1
% Cortando sinal em trechos de 240 amostras (30 ms)
% trechos = zeros(240,46);
%
% for i = 1:46
%         %trechos(:,46) = sinal(240*(i-1)+1:240*i);
```

```
% end

%% 2
% Base com 512 funções aleatórias
fnc_base = randn(N, Q);

%% 3
% Condição inicial do filtro de trato vocal
zs = zeros(1, 10);

%% 4

gerado = zeros(size(sinal));

for i = 1:138
    % A
    % Determinando os coeficientes LPC do quadro atual
    trecho = sinal(240*(ceil(i/3)-1)+1:240*ceil(i/3));
    [aq, sig]= lpc(trecho.*hamming(240), 10);

    subquadro = trecho(81:160);

    % C
    % Filtrando as 512 sequ"Encias da base pelo filtro
    fnc_base_filt = zeros(N, Q);
    fnc_base_filt = filter(1, aq, fnc_base);

    % D
    [y0, zs] = filter(1, aq, zeros(N,1));

    % E
    e0 = subquadro - y0;
```

```
    % F
    [ganhos, indices] = find_Nbest_components(e0, fnc_base_filt, K);

    % G
    d = fnc_base_filt(:,indices)*ganhos;

    % H
    [gerado_trecho, zs] = filter(1, aq, d, zs);

    gerado(80*(i-1)+1:80*i) = d;

end

figure(1);
plot(sinal);
hold on
plot(gerado);
legend("Sinal original", "CELP");
title("Comparação do sinal original com o reconstruído (codificador CELP)");

audiowrite("antarctida_CELP.wav", gerado, fs);

% I
Q_gerado = zeros(size(sinal));

for i = 1:138
    % Determinando os coeficientes LPC do quadro atual
    trecho = sinal(240*(ceil(i/3)-1)+1:240*ceil(i/3));
    [aq, sig]= lpc(trecho.*hamming(240), 10);

    % Quantizando os coeficientes do preditor com 7 bits
```

```
    Q_aq = quantize3(aq, 7);

    subquadro = trecho(81:160);

    % Filtrando as 512 sequ"Encias da base pelo filtro
    fnc_base_filt = zeros(N, Q);
    fnc_base_filt = filter(1, Q_aq, fnc_base);

    [y0, zs] = filter(1, Q_aq, zeros(N,1));

    e0 = subquadro - y0;

    [ganhos, indices] = find_Nbest_components(e0, fnc_base_filt, K);
    % Quantizando os ganhos e os inidices com 5 bits cada
    Q_ganhos = quantize3(ganhos, 5);

    % G
    d = fnc_base_filt(:,indices)*Q_ganhos;

    % H
    [gerado_trecho, zs] = filter(1, Q_aq, d, zs);
    Q_gerado(80*(i-1)+1:80*i) = d;

end

figure(2);
plot(sinal);
hold on
plot(Q_gerado);
legend("Sinal original", "CELP com coeficientes quantizados");
title("Comparação do sinal original com o reconstruído (codificador CELP quantizado)");
audiowrite("antarctida_CELP_quantizado.wav", Q_gerado, fs);
```