Universidade Técnica do Atlântico
Licenciatura em Engenharia Informática de Telecomunicações
Sistemas e Sinais

TRABALHO PRATICO PROCESSAMENTO DE SINAIS DE SOM

Aluno: Ramilton Monteiro Conceição
Professor orientador: Emanuel Ribeiro

Maio
2023

Universidade Técnica do Atlântico
LEIT

Relatório

Primeiro Relatório de Projeto de laboratório do Curso LEIT da Universidade Técnica do Atlântico, como requisito parcial para .

Aluno: Ramilton Monteiro Conceição

Professor orientador: Emanuel Ribeiro

Maio
2023

Conteúdo

1 Resumo

Neste trabalho, vamos estudar alguns conceitos e técnicas de processamento de sinais de som, como a conversão de sinais analógicos em digitais, a compressão de sinais de áudio e a análise espectral de sinais de voz. O objetivo é entender como os sons podem ser representados, manipulados e transmitidos por computadores, usando métodos matemáticos e algoritmos.

Para realizar este trabalho, vamos usar o software MATLAB, que permite criar, visualizar e processar sinais de som de forma fácil e eficiente. Vamos também usar alguns arquivos de áudio em formato WAV, que contêm gravações de voz e música. Vamos aplicar diferentes operações sobre esses arquivos, como filtragem e transformada de Fourier.

Em cada parte, vamos seguir os passos indicados no roteiro do trabalho, que contém as instruções detalhadas e as questões a serem respondidas. Vamos também usar os arquivos fornecidos pelo professor, que contêm as funções e os dados necessários para realizar o trabalho.

2 Apresentação

Este trabalho prático tem como objetivo introduzir os conceitos básicos de processamento de sinais de som usando o matlab. Você deverá realizar as seguintes tarefas:

- Ler e reproduzir um arquivo de áudio no formato wav; - Geração de tom (cossenoide);

- Introdução de eco;

- Introdução de eco;

- Analisar o conteúdo espectral dos sinais de som usando a transformada de Fourier:

-Introdução de eco;

-Redução do ruído no sinal senoidal com SNR = 5 dB;

-Energia e taxa de cruzamentos por zero (TCZ);

-Função de autocorrelação.

3 Introdução

O processamento de sinais de som é uma área da engenharia que visa analisar, modificar e sintetizar sons a partir de diferentes fontes. Neste trabalho prático, pretende-se fornecer uma introdução aos conceitos básicos do processamento de sinais de som em matlab, uma ferramenta poderosa e versátil para manipular dados numéricos. O objetivo deste trabalho é acarretar uma compreensão dos princípios fundamentais do processamento de sinais de som, tais como amostragem, quantização, transformada de Fourier, filtragem e modulação. Além disso, supõe-se que o aluno tenha conhecimentos prévios de álgebra linear e cálculo diferencial e integral. O trabalho consiste em tratar diversos problemas práticos envolvendo sinais de som, tais como geração de tons puros, mistura de sinais, remoção de ruído e compressão de áudio. Para cada problema, o aluno deverá investigar as soluções possíveis, considerar as vantagens e desvantagens de cada uma e apontar os resultados obtidos.

4 Geração de tom (cossenoide)

```
% Funcao do COSSENOIDE
% A funcao cocenoide e uma fun   o que
% retorna o coceno de um angulo em graus.

fs = 44100; % frequencia da amostragem, Hz
f1 = 100; % frequencia fundamental, Hz
T = 2; % dura   o , s
N = floor(fs/f1); % pontos por per odo
n = 0:floor(T*fs); % indice das amostras
t = n/fs; % Escala de tempo, s
C = 5 %conforme mudar o valor de constante C
%aunmeta o numero de itens no gr fico

% Numero componentes para Serie
P = 2; % Numero de periodos para visualizacao
ind = 1:P*N; % Indice eixo x para visualizacao
figure(1),
s = geracos(1/2, 0, 0, fs, T); % n vel DC
plot(t(ind), s(ind));
hold on,
for i = 1:C, % Soma dos C componentes a s
s = s + geracos(1/pi/i, i*f1, pi/2, fs, 2);
plot(t(ind), s(ind));
end
plot(t(ind), s(ind), 'k', 'LineWidth', 2);
msg = 'Composicao dente-de-serra (sawtooth)';
msg = sprintf('%s com %d componentes', msg, C);
title(msg); xlabel('t, s');
grid, hold off,
```

```
%GERACOS
function tom = geracos(A, f1, ph, fs, T)
%N = floor(fs/f1); % pontos por per odo
n = 0:floor(T*fs); %  ndice  das amostras
tom = A * cos(2*pi*f1*n/fs + ph);
```

Este código gera uma onda de dente de serra usando uma série de Fourier. Uma série de Fourier é uma forma de representar uma função periódica como uma soma de funções seno e cosseno. O código define a frequência de amostragem, a frequência fundamental e a duração da onda. Em seguida, calcula o número de pontos por período e gera uma escala de tempo. A constante C determina o número de componentes na série de Fourier

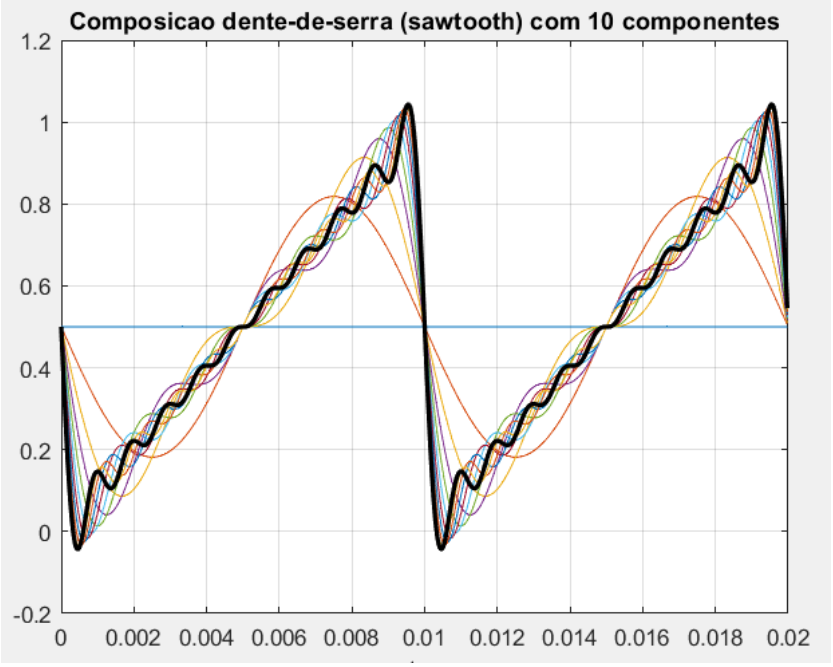


Figura 1: Gráfico de 10 Componentes

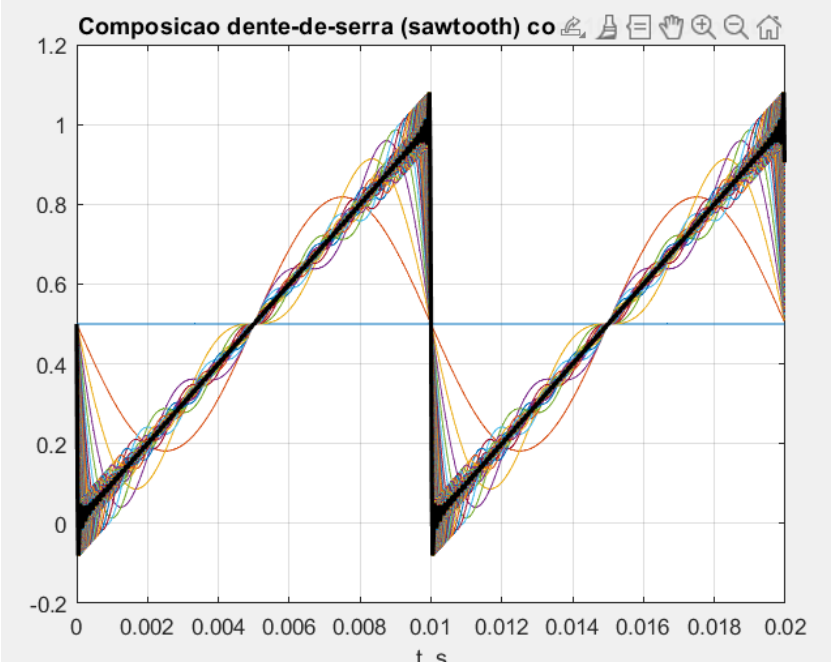


Figura 2: Gráfico de 100 Componentes

5 Introdução de eco

```
%Funcao Introduzir ECO
[y, fs] = audioread('seis_roms.m4a');
y=y(:,1);%impedir de formar erros
T = 800; % echo em ms
N = T * fs/1000;
A = 1; % a sua amplitude do echo
pulse = [A zeros(1, N)];
trem = pulse
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:14,
    pulse = .8*pulse;
    trem = [trem pulse];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s = conv(trem,y);
sound(s, fs);
```

O som será reproduzido com um eco maneira que é feita uma alteração nas variáveis acima no código. Abaixo há um texto explicando o que é o eco.

Eco é um fenômeno acústico que ocorre quando uma onda sonora é refletida por uma superfície e retorna ao ponto de origem com um atraso perceptível. A introdução de eco num som pode ter diferentes efeitos, dependendo da intensidade, duração e frequência do eco. Por exemplo, um eco suave e curto pode criar uma sensação de reverberação, que é usada para dar mais profundidade e ambiência a uma gravação musical. Já um eco forte e longo pode causar interferência e distorção no som original, prejudicando a qualidade e a inteligibilidade da mensagem sonora. Portanto, é importante controlar a quantidade e o tipo de eco que se deseja introduzir num som, usando técnicas adequadas de captação, processamento e reprodução do áudio.

6 Inversão no tempo

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%esse c digo vai fazer a invers o do som

    [y, fs] = audioread('seis_roms.m4a');

    % revers o do audio
    w = y(end:-1:1);

    % plot audio waveform
    t = (0:length(w)-1)/fs;
    plot(t, w);
    xlabel('Time (s)');
    ylabel('Amplitude');
    title('Reversed Audio');
```

A inversão do tempo é uma técnica matemática que consiste em reverter a ordem dos elementos de uma sequência ou de uma função. No matlab, existem várias formas de realizar a inversão do tempo de um vetor ou de um sinal.

A inversão do tempo pode ser útil para analisar propriedades de sinais e sistemas, como simetria, causalidade e estabilidade.

7 Áudio estéreo

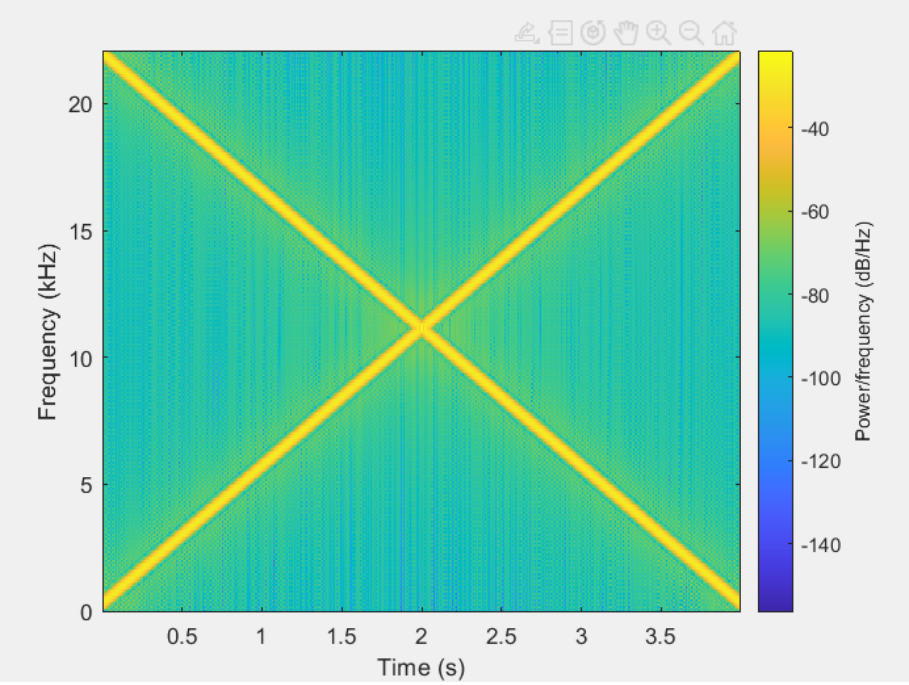
```
fs = 44100; % freq amostragem, Hz
s = A * sin (2*pi*f1*t); % amostras da senoide
sound(s, fs); % reproducao no alto-falante
figure(1)
sinal1 = genlin(4, 300, 22000, fs);
soundsc(sinal1, fs);
spectrogram(sinal1, hamming(256), 128, 512, fs, 'yaxis');
%
    ----------------------------------------------------------------

figure(2)
sinal2 = genlin(4, 22000, 300, fs);
soundsc(sinal2, fs);
spectrogram(sinal2, hamming(256), 128, 512, fs, 'yaxis');
%
    ----------------------------------------------------------------

figure(3)
sinal3 =sinal1 + sinal2;
soundsc(sinal3, fs);
spectrogram(sinal3, hamming(256), 128, 512, fs, 'yaxis');
%
    ----------------------------------------------------------------

[y, fs] = audioread('sinal_som.wav');
spectrogram(y, hamming(256), 128, 512, fs, 'yaxis');
function sinal = genlin(td, fi, ff, fs)
t = 0:1/(td*fs):1;
if fi <= ff,
freqlin = fi + (ff-fi)*t;
else
freqlin = fi - (fi-ff)*t;
end
sinal = freqgen(freqlin, fs); % funcao definida no proximo
    slide
end
% Geracao de tom de frequencia variavel dada por fr
function [sinal] = freqgen(fr, fs)
fr = fr*2*pi/fs; % normalizacao para rad/s
phi = cumsum(fr); % soma acumulada
sinal = sin(phi);
end
function sinal = genexp(td, c, fi, ff, fs)
x = 0:1/(td*fs):1; m = exp(c); % m constante de warping
M = exp(m);
if fi <= ff,
```

```
freqexp = fi + (exp(m*x)-1)/(M-1)*(ff-fi);
else
freqexp = ff + (exp(m*x)-1)/(M-1)*(fi-ff);
freqexp = freqexp(end:-1:1);
end
sinal = freqgen(freqexp, fs);
end
```



Espectogram dos Sinal do som gravado

8 Redução do ruído no sinal senoidal com SNR = 5 dB

A redução de ruído em um sinal senoidal com SNR = 5 dB pode ser realizada usando uma variedade de métodos. Uma abordagem comum é usar um filtro notch. Um filtro notch é um tipo de filtro projetado para atenuar uma frequência específica ou faixa de frequências. No caso de redução de ruído, um filtro notch pode ser usado para atenuar a frequência do ruído. Isso ajudará a melhorar a relação sinal-ruído (SNR) do sinal.

Outra abordagem para redução de ruído é usar um filtro Wiener. Um filtro de Wiener é um tipo de filtro projetado para minimizar o erro quadrático médio entre o sinal de entrada e o sinal de saída. No caso de redução de ruído, um filtro de Wiener pode ser usado para minimizar o erro quadrático médio entre o sinal original e o sinal com o ruído removido.

Finalmente, a redução de ruído também pode ser realizada usando uma variedade de outros métodos, como filtragem adaptativa, processamento de sinal no domínio do tempo e processamento de sinal no domínio da frequência. A melhor abordagem para redução de ruído dependerá das características específicas do sinal e do ruído.

9 Filtragem

```
% Carregue o sinal
[y, fs] = audioread('seis_roms.m4a');

% Defina as especifica  es do filtro
cutoff_freqs = [500, 1000, 2000]; % Hz
orders = [100, 200, 400];

% Aplique diferentes filtros com especifica  es diferentes
for i = 1:length(cutoff_freqs)
% Projete o filtro FIR
b = fir1(orders(i), cutoff_freqs(i)/(fs/2));

% Filtrar o sinal
y_filtered = filter(b, 1, y);
plot( y_filtered );
end
```

Filtragem é uma técnica de processamento de imagens que permite modificar ou realçar certas características de uma imagem. No MATLAB, existem várias funções que podem ser usadas para aplicar filtros a imagens, como filter, fspecial, imfilter e medfilt2. Cada função tem seus próprios parâmetros e tipos de filtros disponíveis, como passa-baixa, passa-alta, média, mediana, gaussiano, laplaciano, etc.

Para usar essas funções, é preciso primeiro converter a imagem para o domínio da frequência usando a transformada de Fourier (fft2). Em seguida, é preciso definir os coeficientes do filtro desejado e multiplicá-los pela imagem transformada. Por fim, é preciso converter a imagem filtrada de volta para o domínio espacial usando a transformada inversa de Fourier (ifft2).

10 Função de autocorrelação

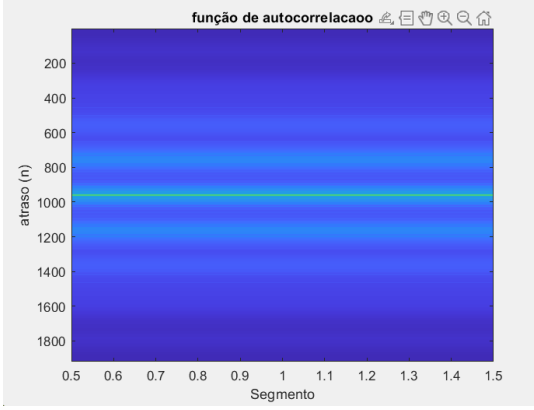
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Autocorrelacao

[y, fs] = audioread('seis_roms.m4a');
M = fix(fs * 0.02);
P = fix(fs * 0.01);
L = length(y);
N = ceil((L - M) / P) + 1;
y = y - mean(y);
R = zeros(N, M);
% Calcula a funcao de autocorrelacao
for i = 1:N
    % Define os indices da janela atual
    start = 1 + (i-1)*P;
    stop = start + M - 1;

    % Extrai a janela atual
    x = y(start:stop);

    % Calcula a autocorrelacao da janela atual
    R = xcorr(x, x, 'biased');
end

% Plota a funcao de autocorrelacao
imagesc(R')
xlabel('Segmento')
ylabel('atraso (n)')
title('fun   o de autocorrelacaoo')
```



11 Conclusão

Processamento do sinal de voz é uma área da engenharia elétrica que estuda as técnicas de análise, síntese, codificação e reconhecimento de sinais sonoros produzidos pela fala humana. O objetivo é extrair informações relevantes do sinal de voz, como o conteúdo semântico, a identidade do locutor, o estado emocional, etc., ou transformar o sinal de voz em outro formato mais adequado para uma determinada aplicação, como transmissão, armazenamento ou síntese.

O processamento do sinal de voz envolve diversas etapas, como a aquisição do sinal sonoro por meio de um microfone ou outro dispositivo, a conversão do sinal analógico em digital por meio de amostragem e quantização, a aplicação de algoritmos de processamento no domínio do tempo ou da frequência, a utilização de modelos matemáticos e estatísticos para representar e classificar o sinal de voz, e a geração de um novo sinal de voz ou de uma saída textual ou gráfica.

Algumas das principais aplicações do processamento do sinal de voz são:

- Codificação de voz: consiste em reduzir a taxa de bits necessária para representar o sinal de voz sem comprometer a qualidade perceptual. A codificação de voz é importante para otimizar o uso da largura de banda em sistemas de comunicação, como telefonia e internet.

- Síntese de voz: consiste em gerar um sinal de voz artificial a partir de um texto ou de outro tipo de entrada. A síntese de voz é útil para sistemas que precisam produzir fala sintética, como assistentes virtuais, leitores de tela, jogos, etc.

- Reconhecimento de voz: consiste em identificar as palavras ou frases pronunciadas por um locutor em um sinal de voz. O reconhecimento de voz é essencial para sistemas que precisam interagir com o usuário por meio da fala, como comandos por voz, tradução automática, transcrição, etc.

- Identificação do locutor: consiste em determinar a identidade do locutor que produziu um sinal de voz. A identificação do locutor pode ser usada para fins de segurança, autenticação, personalização, etc. - Análise da emoção: consiste em inferir o estado emocional do locutor a partir do sinal de voz. A análise da emoção pode ser aplicada para melhorar a interação humano-computador, monitorar a saúde mental, avaliar o desempenho, etc.

Em conclusão, o processamento do sinal de voz é uma área multidisciplinar que combina conhecimentos de engenharia elétrica, ciência da computação, matemática, física, linguística e psicologia. O processamento do sinal de voz tem diversas aplicações práticas que podem melhorar a qualidade e a eficiência dos sistemas que envolvem comunicação por fala.

Bibliografia

Introdução à Análise e ao Processamento de Sinais Usando o MATLAB, Sampaio, Rubens and Cataldo, Edson and Riquelme, Roberto, 1998, Gráfica do INPE X. Huang, A. Acero and H.W. Hon. Spoken Language Processing:

A Guide to Theory, Algorithm and System Development. Prentice Hall; 2001 P. Taylor. Text-to-Speech Synthesis. Cambridge University Press; 2009 I.J.

Tashev. Sound Capture and Processing. John Wiley; 2009