

Sinais e Sistemas

Trabalhando com áudio - Parte 2

Rafael Corsi
rafael.corsi@maua.br

27 de março de 2015

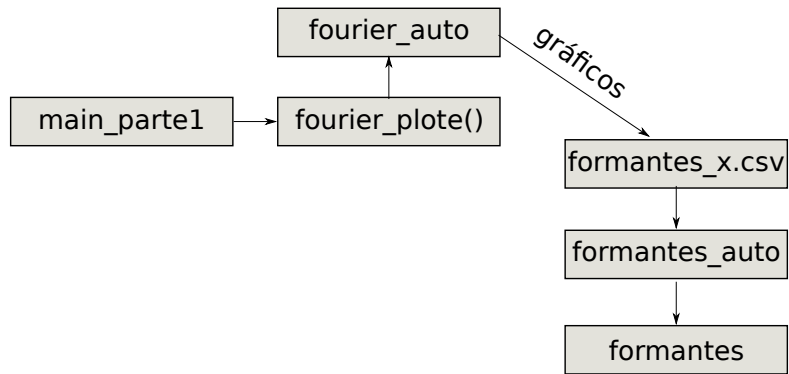
1 Objetivo

Automatizar via software Matlab o cálculo de Fourier aplicada a arquivos de áudios (vogais) e plotar um gráfico com todas as formantes levantadas.

1.1 Arquivos e estrutura

Ao final do projeto, deveremos ter os seguintes arquivos (sugestão):

- *main_Parte1_R.m* : Script Matlab da primeira parte do projeto;
- *fourier_auto.m* : Script Matlab automatizado para gerar a análise espectral dos áudios;
- *fourier_plote.m* : Função Matlab que recebe como parâmetro o áudio (vetor) e plota o fourier do sinal (baseado em main.m);
- *formante_X.csv* : Arquivo tabulado com as informações das formantes (F1, F2) de cada áudio, X = A ou E ou ...;
- *formante_import.m* : Função que importa os dados tabulado para o workspace;
- *formante_auto.m* : Script Matlab para plotar as formantes num único gráfico.
- /Audios : Pasta contendo as vogais gravadas



1.1.1 Áudios

Os arquivos devem estar numa subpasta do Matlab, no seguinte formato : **VGIN.mp3**, onde :

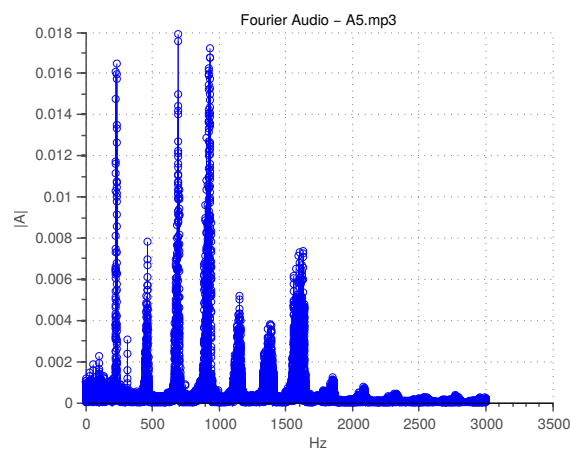
- **V** : Vogal (ex. A,E,I,O,U)
- **G** : Grupo (ex. 1,2,3,4, ...)
- **I** : Indivíduo (ex. 1,2,3,4, ...)
- **N** : Gravação (ex. 1,2,3,4,5,...)

Exemplo : *A122.mp3* (Gravação da vogal A, pelo grupo 1, indivíduo 2, gravação 2).

2 Matlab

2.1 fourier_plote.m

Crie com base em *main_parte1.m* uma função que receba as variáveis: y, Fs, nome_arquivo e gere um gráfico em frequência desse sinal.



Dicas:

1. Para criar uma função basta ir em: \rightarrow nem \rightarrow function.

```
function [ ] = formante_plot(y, Fs, arquivo )
```

```
end
```

2. No Matlab o nome da função deve ser o mesmo do arquivo (sem o .m)
3. Coloque no título do gráfico o nome do arquivo plotado, para isso use a função *title(nome)*, onde nome é uma string.

2.2 **fourier_auto.m**

Script que varre a pasta áudio carregando em cada interação um novo áudio e executando a função *fourier_plote.m*. Deve gerar um gráfico para cada áudio.

Dicas:

1. Crie um vetor com as possíveis vogais e use esse vetor para criar o arquivo de áudio:
vogais = ['A' ; 'B' ; 'C' ; 'D' ; 'E']; Com isso podemos acessar qualquer vogal:

```
>> vogais(3)
>> C
```

2. a função *num2str(x)* converte um número para uma string.
3. para concatenar strings basta colocar entre [] , exe:

```
>> nome = ['Aureliano' 'Buendia']
nome =
AurelianoBuendia
```

4. Crie uma variável para armazenar o nome do arquivo :

```
>> k = 2; i = 1; j = 1;
>> nome = [vogais(i) j k '.mp3']
>> B11.mp3
```

5. Faça agora uma varredura em todos os áudios e para cada áudio importado chame a função: *fourier_plote.m*.
6. Execute o *audioread* dentro do *try/cache* para evitar que um arquivo inexistente pare a execução do código:

```
try
    Codigo que pode conter erro
end
```

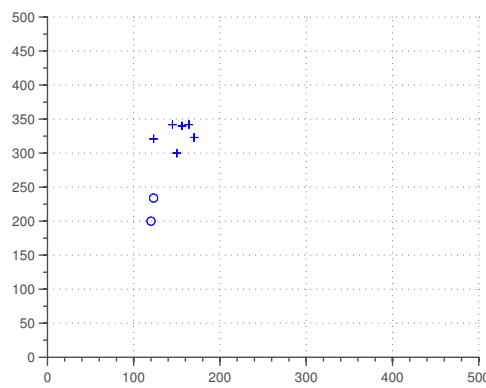
2.3 formantes_X.csv

Crie um arquivo para cada vogal formantes_X.csv com as formantes de cada áudio, o arquivo deve ser da seguinte maneira (pode usar o Excel e exportar como .csv):

Vogal;	Grupo;	Individuo;	Gravacao	F1;	F2;
A;	1;	1;	1	120;	200;
A;	1;	2;	2	123;	234;

2.4 formante_auto.m

Deve importar os dados dos arquivos formantes_X.csv e plotar em um gráfico F1 vs F2 o valor das formantes.



Dicas:

1. Para importar automaticamente arquivos .csv, basta abrir o arquivo via Matlab, formatar a importação e criar uma função que execute isso automaticamente (generate function).

2. A função `formante_import(Arquivo)` já faz essa leitura:

```
[Va, Ga, Ia, F1a, F2a] = formante_import('formantes_A.csv');
```

2.5 Fourier Matlab

A seguir, exemplo de como gerar e plotar o Fourier no Matlab:

figure

```
Fs = Fs;           % Sampling frequency
Ts = 1/Fs;         % Sample time
L = size(y,1);     % Length of signal
```

```

NFFT = 2nextpow2(L);           % Next power of 2 from length of y
Y = fft(y,NFFT)/L;             % Calcula FFT
f = Fs/2*linspace(0,1,NFFT/2+1); % Ajusta frequencia

xf = find(f >= 3000,1);         % encontra index da freq. >= 3k Hz

stem(f(1:xf),2*abs(Y(1:xf)))    % plota resultado
title(['Fourier_Audio_-' nome]);
xlabel('Hz'); ylabel('|A|');

```