



UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA

ES879 – SISTEMAS DE AQUISIÇÃO DE DADOS

2º Semestre de 2023 – Trabalho Final  
Turma: A Prof. Tiago Henrique Machado

## Processamento Digital de Sinais Aplicado a um Sinal de Áudio

### Introdução

Neste trabalho, iremos aplicar os conceitos de projeto e aplicação de filtros digitais, análise em frequência a partir da Transformada de Fourier Discreta (DFT), decimação e interpolação para ajustar digitalmente a taxa de amostragem de um sinal, e uma aplicação de tratamento de eco em um sinal de áudio.

### Trabalhando com um sinal musical

A ideia agora é trabalhar com um arquivo de áudio referente a um trecho de alguma música entre as oferecidas na página do Google Classroom da disciplina.

Para carregar o arquivo de áudio no Matlab, basta usar o comando *audioread*:

```
[y, Fs] = audioread('creed_my_sacrifice.wav');
```

Este comando retorna o sinal de áudio,  $y$ , e a frequência de amostragem,  $F_s$ . Note que  $y$  corresponde a uma matriz com *num\_amostras* linhas e duas colunas, uma para cada canal de áudio. Apenas por simplicidade, some os dois canais para realizar o trabalho:

```
y = y(:, 1) + y(:, 2);
```

Em Python existem diversas bibliotecas que permitem a leitura e o processamento de arquivos de áudio. Dois exemplos populares e poderosos são *librosa* e *soundfile*. Outra possibilidade é usar o *SciPy*, que tem grandes chances de já fazer parte de sua instalação de Python. Para isso, use os comandos

```
import scipy.io as sio
```

```
Fs, y = sio.wavfile.read('creed_overcome.wav')
```

Talvez você receba um *Warning* ao fazer a leitura aqui. Isso está relacionado à forma como o arquivo foi gerado, e não tem impacto sobre o restante do experimento. Como em Matlab,  $y$  é um sinal estéreo, com dois canais. Para transformar em um sinal mono, execute o comando:

```
 $y_{soma}=y[:,0]+y[:,1]$ 
```

Após fazer este procedimento, vocês devem fazer o que se pede:

- a) Mostre o espectro de frequências do sinal de áudio e discuta seu conteúdo espectral.
- b) Sem realizar nenhum tipo de pré-filtragem no sinal, reduza a taxa de amostragem por um fator de  $M = 8$ . Apresente o espectro do sinal subamostrado e discuta as mudanças em relação ao espectro do sinal original.
- c) Ouça, então, o sinal de áudio original e o subamostrado. Comente as diferenças. Para isto, utilize o comando

**soundsc do Matlab:**

**soundsc(z,Fs),**

onde  $F_s$  denota a taxa de amostragem associada ao sinal  $z$ .

Para ouvir o áudio em Python, use os comandos

```
import IPython.display as ipd
```

```
ipd.Audio(z,rate=Fs)
```

Obs: Lembre-se que, após a decimação, a taxa de amostragem foi reduzida para  $F_s/M$ .

- d) Filtre, agora, o sinal de áudio original com um filtro passa-baixa. Você deve identificar valores apropriados para os parâmetros do filtro ( $k_1$ ,  $k_2$ ,  $\omega_p$  e  $\omega_r$ ). Apresente graficamente a resposta em frequência do filtro, o espectro do sinal filtrado e compare-o com o espectro do sinal original. Ouça, então, ambos os sinais e discuta as diferenças percebidas e os efeitos da filtragem.
- e) Reduza a taxa de amostragem do sinal filtrado pelo mesmo fator  $M = 8$  do item (b). Compare o espectro do sinal obtido com o sinal subamostrado do item (b) (ou seja,

compare o sinal original subamostrado e o sinal filtrado subamostrado). Escute ambos e discuta as diferenças.

- f) Volte ao sinal original, aumente a taxa de amostragem por um fator de  $L = 4$ . Apresente o espectro do sinal superamostrado e discuta as mudanças em relação ao espectro do sinal original. Ouça, então, o sinal de áudio original e o superamostrado. Comente as diferenças.

### Incluindo ECO em um sinal

A última parte do trabalho mostra uma aplicação curiosa que pode ser feita através do processamento digital de sinais. A ideia aqui é incluir Eco em um determinado sinal de áudio. Para isso, utilize o mesmo sinal de áudio fornecido. Este procedimento será feito utilizando um filtro conhecido como Filtro de Eco.

Um Filtro de Eco possui uma equação de diferenças no formato:

$$y[n] = x[n] + a x[n - D]$$

Sabendo disso, vocês devem fazer o que se pede:

- g) Mostre a função de transferência e a resposta ao impulso deste filtro.
- h) Escolha o número de amostras,  $D$ , que o sinal deve ser atrasado, de tal forma que o sinal seja atrasado em 0,5s, especifique  $a = 0,5$ . Trace a resposta em frequência do filtro e mostre o sinal após passar por este filtro.
- i) Faça o teste com valores de  $D$  cada vez menores e escute o sinal com eco até que não seja mais perceptível distinguir do sinal original. Calcule o tempo de atraso que este valor corresponde.

Outras informações importantes com relação a este trabalho:

- ✓ os trabalhos devem ser desenvolvidos em grupos de até duas pessoas. **Os grupos devem enviar o nome e o RA de cada integrante** por e-mail (n265296@dac.unicamp.br) até **sexta-feira (14/11)**;

- ✓ **os trabalhos devem ser entregues em uma versão escrita via Google Classroom até o dia 07/12/2023**, contendo o detalhamento de tudo que foi feito, apresentando gráficos, discussões e conclusões acerca do que é pedido. Além disso, em anexo, deve ser colocado o código em Matlab, Python ou software semelhante, utilizado para gerar os resultados;
- ✓ **os trabalhos também serão brevemente apresentados pelos grupos nos dias 07/12/2023 ou 12/12/2023** (apresentações de cerca de 15 minutos). As datas de apresentação de cada grupo serão definidas em sorteio realizado na aula de quinta-feira (16/11/2023);
- ✓ será valorizada a capacidade criativa dos grupos para resolver o problema e forma detalhada como o procedimento foi realizado e detalhado tanto na versão escrita como na apresentação;
- ✓ por mais que o enunciado seja o mesmo, as formas de se resolver e implementar são diversas, o que faz com que seja estatisticamente impossível que se tenham dois trabalhos iguais simplesmente por ‘coincidência’.