

# CAMADA FÍSICA DA COMPUTAÇÃO

ENGENHARIA DA COMPUTAÇÃO - Rodrigo Carareto – 0#07E3/02

## PROJETO 8 – Modulação AM

### Objetivos do projeto

Nesse projeto você irá construir dois softwares com as seguintes funcionalidades:

#### Software 1

1. Faça a leitura de um arquivo de áudio previamente gravado com uma taxa de amostragem de 44100Hz.
2. Normalize esse sinal.
3. Filtre as altas frequências desse sinal (frequências acima de 4000 Hz).
4. Codifique esse sinal de áudio em AM (portadora de 14000 Hz).
5. Construa o gráfico nos domínios do **tempo** e da **frequência** (Fourier) para os seguintes sinais:
  - a. Sinal de áudio original.
  - b. Sinal de áudio normalizado.
  - c. Sinal de áudio filtrado.
  - d. Sinal de áudio modulado.
6. Execute o áudio do sinal modulado.

#### Software 2

7. Grave o áudio executado pelo software 1 com a mesma taxa de amostragem (44100).
8. Demodule o sinal.
9. Execute o áudio do sinal demodulado.
10. Mostre o gráfico no domínio do **tempo** e da **frequência** (Fourier) do sinal captado e do sinal demodulado.

### Ajudas

#### Para importar o sinal:

Importar um arquivo .wav: Você poderá usar a biblioteca soundfile, que contém uma função `.read(...)`. Após importar o arquivo, você deverá extrair o vetor com as amplitudes e então normaliza-lo (valores entre 0 e 1. Para melhores resultados, você poderá tratar o sinal lido aplicando um filtro passa baixa (mostrado abaixo) utilizando-se a classe `signal` (`from scipy import signal`).

A execução do áudio pode ser feita com a função `play` da biblioteca `sounddevice`

#### Filtro passa baixa:

Você precisará encontrar alguma função ou classe que sirva como filtro passa-baixa para o sinal (exclui as componentes em alta frequência, restando apenas as de baixa. Por isso o nome “filtro passa-baixa”). Nessa função de filtragem, você deverá ter a possibilidade de determinar a partir de qual frequência o filtro atuará (frequência de corte “cut-off frequency”)

*#exemplo de filtragem do sinal yAudioNormalizado*

*# <https://scipy.github.io/old-wiki/pages/Cookbook/FIRFilter.html>*

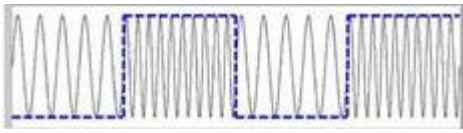
*nyq\_rate = samplerate/2*

*width = 5.0/nyq\_rate*

*ripple\_db = 60.0 #dB*

*N, beta = signal.kaiserord(ripple\_db, width)*

*cutoff\_hz = 4000.0*



# CAMADA FÍSICA DA COMPUTAÇÃO

ENGENHARIA DA COMPUTAÇÃO - Rodrigo Carareto – 0#07E3/02

```
taps = signal.firwin(N, cutoff_hz/nyq_rate, window=('kaiser', beta))
```

```
yFiltrado = signal.lfilter(taps, 1.0, yAudioNormalizado)
```

## Modulando e demodulando

1. A modulação do sinal poderá ser feita com a multiplicação entre a portadora de amplitude 1 e o sinal importado e normalizado.
2. A demodulação deverá ser feita novamente multiplicando-se o sinal modulado pela portadora, com um filtro passa-baixa na frequência de corte do sinal importado posteriormente. O módulo do sinal poderá ser obtido com a multiplicação do sinal de áudio e a portadora.