



**INSTITUTO  
FEDERAL**

Santa Catarina

---

Câmpus  
São José

# **Modulação e Demodulação em Frequência (FM)**

Sistemas de Comunicação I

**Arthur Cadore Matuella Barcella**

09 de Abril de 2024

# Sumário

1. Introdução .....	3
2. Fundamentação teórica .....	3
3. Análise dos resultados .....	3
4. Scripts e Códigos Utilizados: .....	3
5. Conclusões .....	4
6. Referências .....	4

# 1. Introdução

Seção I - Descrição do que será desenvolvido/abordado no relatório

## 2. Fundamentação teórica

Seção II - Conceitos teóricos utilizados no relatório

## 3. Análise dos resultados

Seção III - Apresentação e comentários dos gráficos/figuras das etapas de desenvolvimento do relatório

## 4. Scripts e Códigos Utilizados:

Seção IV - Scripts e Codigos

```
1 close all; clear all; clc;
2
3 % Defining the signals amplitude.
4 A_modulating = 1;
5 A_carrier = 1;
6
7 % Defining the signals frequency
8 f_modulating_max = 20000;
9 f_carrier = 80000;
10
11 % modulator sensibility for frequency variation (Hz/volts)
12 k_f = 2000000;
13
14 % Delta variable, corresponding to max frequency variation.
15 d_f = k_f*A_modulating;
16
17 % Beta variable, corresponding to percentage of frequency variation about
18 % the frequency of the modulating.
19 b = d_f/f_modulating_max;
20
21 % Defining the period and frequency of sampling:
22 fs = 50*f_carrier;
23 Ts = 1/fs;
24 T = 1/f_modulating_max;
25
26 % Defining the sinal period.
27 t_inicial = 0;
28 t_final = 2;
29
30 % "t" vector, corresponding to the time period of analysis, on time domain.
31 t = [t_inicial:Ts:t_final];
```

```
1 % modulating_singal = A_modulating *cos(2*pi*f_modulating_max*t);
2 [modulating_signal, Hs] = audioread('general-signal.wav');
```

```

3 modulating_signal = transpose(modulating_signal);
4
5 % Calculate the number of zeros to be added
6 num_zeros = length(t) - length(modulating_signal);
7
8 % Add the zeros to the end of the modulating_signal vector
9 modulating_signal = [modulating_signal, zeros(1, num_zeros)];
10
11 % Transpose the modulated signal if necessary
12 modulated_signal = transpose(modulating_signal);
13
14 % Creating the FM modulated signal:
15 phase_argument = 2*pi*k_f*cumsum(modulating_signal)*(Ts);
16 modulated_signal = A_carrier * cos(2*pi*f_carrier*t + phase_argument);
17
18 % Plot signals on time domain:
19 figure(1)
20 subplot(311)
21 plot(t, (modulating_signal), 'b', 'LineWidth', 2)
22 xlim([0.00054 0.00067])
23 xlabel('Time (s)')
24 ylabel('Amplitude')
25
26 subplot(312)
27 plot(t, abs(modulating_signal), 'r', 'LineWidth', 2)
28 xlim([0.00054 0.00067])
29 xlabel('Time (s)')
30 ylabel('Amplitude')
31
32 subplot(313)
33 plot(t, modulated_signal, 'k', 'LineWidth', 2)
34 xlim([0.00054 0.00067])
35 xlabel('Time (s)')
36 ylabel('Amplitude')

```

```

1 % calculating the step of the frequency vector "f" (frequency domain);
2 f_step = 1/t_final;
3
4 % creating the frequency vector "f" (frequency domain);
5 f = [-fs/2:f_step:fs/2];
6
7 % calculating the FFT of the modulated signal;
8 modulated_f = fft(modulated_signal)/length(modulated_signal);
9 modulated_f = fftshift(modulated_f);

```

## 5. Conclusões

Seção V - Conclusões

## 6. Referências

Seção VI - Referências bibliográficas