



**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: .....	5
5. Conclusões .....	7
6. Referências .....	7

# 1. Introdução

O objetivo deste relatório é apresentar o desenvolvimento de um sistema de modulação e demodulação em frequência (FM) para sinais de áudio. O sistema foi desenvolvido através de linguagem MATLAB (Octave), e tem como objetivo principal a compreensão do processo de modulação e demodulação em frequência, bem como a análise dos sinais modulados e demodulados.

Neste relatório será apresentado a fundamentação teórica do processo de modulação e demodulação em frequência, bem como a análise dos sinais modulados e demodulados, os scripts MATLAB utilizados e os resultados obtidos.

Desta forma, poderemos compreender o processo de modulação e demodulação FM, bem como a análise dos sinais modulados e demodulados, e a importância deste processo para a transmissão de sinais de áudio em sistemas de comunicação.

## 2. Fundamentação teórica

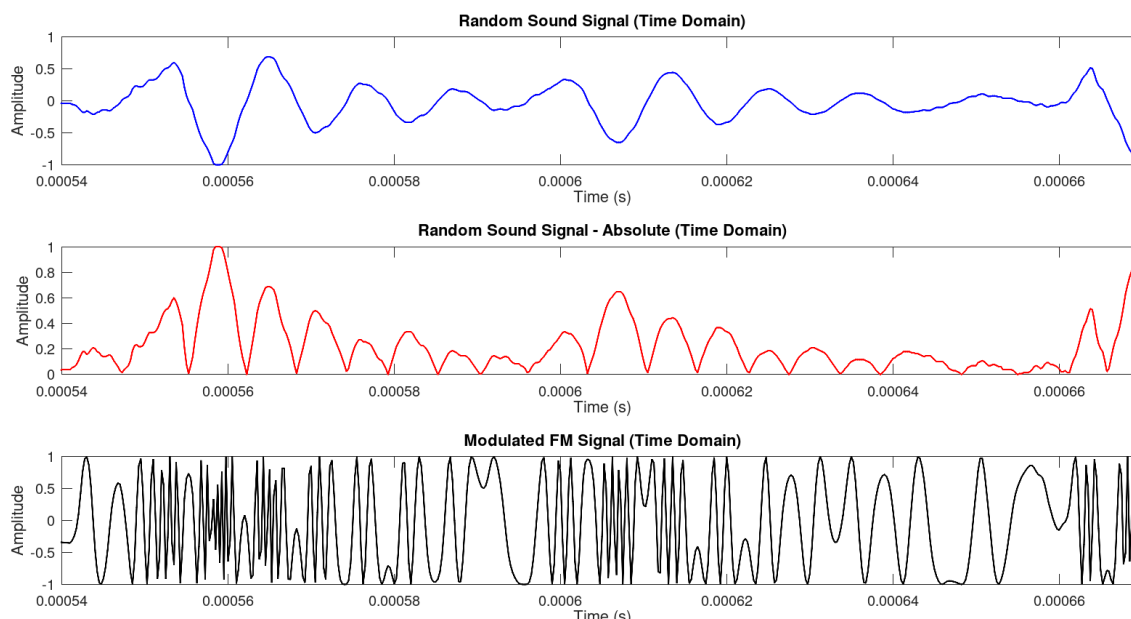
Seção II - Conceitos teóricos utilizados no relatório

## 3. Análise dos resultados

Inicialmente, foi feita a importação de um sinal de áudio para ser utilizado como modulante da portadora em frequência, para transmissão em FM.

A figura abaixo mostra o plot do sinal no domínio do tempo, bem como seu respectivo plot do sinal no domínio da frequência.

Figure 1: Elaborada pelo Autor



Sinal de entrada no domínio do tempo

Uma vez com o sinal de entrada definido, a modulação em frequência foi realizada através da integração do argumento de fase da portadora a partir do sinal da modulante, conforme o script abaixo:

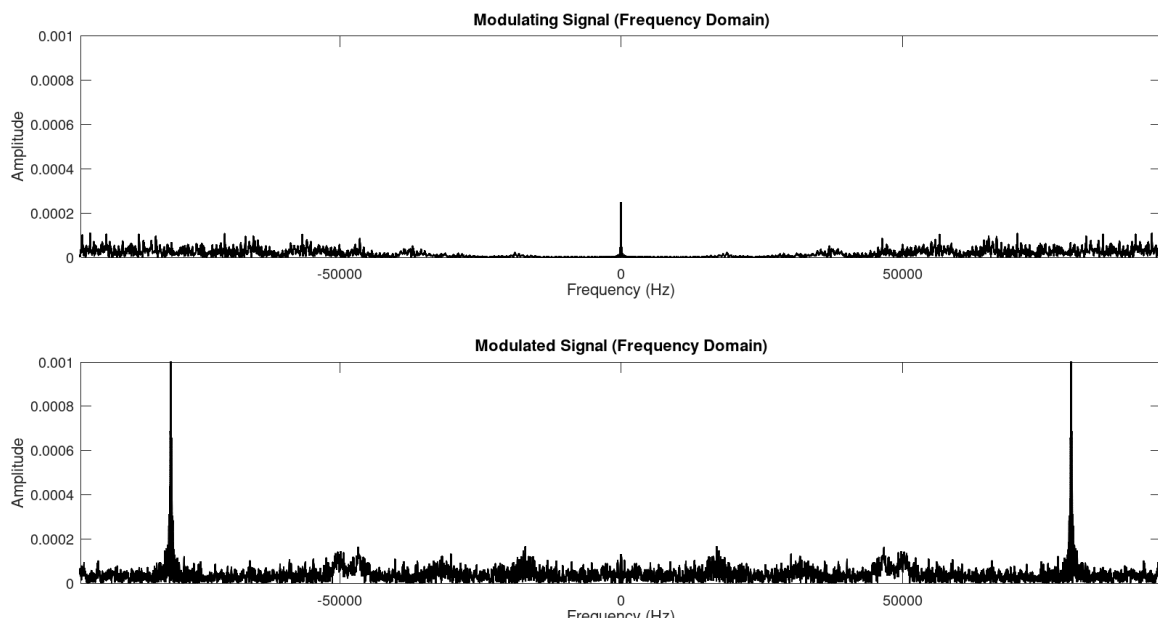
```
1 % Creating the FM modulated signal:
2 phase_argument = 2*pi*k_f*cumsum(modulating_signal)*(Ts);
3 modulated_signal = A_carrier * cos(2*pi*f_carrier*t + phase_argument);
```

Onde na figura acima os parâmetros são:

- `modulating_signal` é o sinal de áudio importado.
- `k_f` é a sensibilidade do modulador para variação de frequência.
- `Ts` é o período de amostragem do sinal.
- `A_carrier` é a amplitude da portadora
- `f_carrier` é a frequência da portadora.
- `t` é o vetor de tempo do sinal modulado (utilizado para realizar a modulação em FM).
- `phase_argument` é o argumento de fase da portadora do sinal, gerado a partir da integração do sinal modulante.
- `modulated_signal` é o sinal modulado em FM.

Uma vez com o sinal modulado em FM, podemos compreender o formato do sinal modulado no domínio do tempo e da frequência, conforme a figura abaixo:

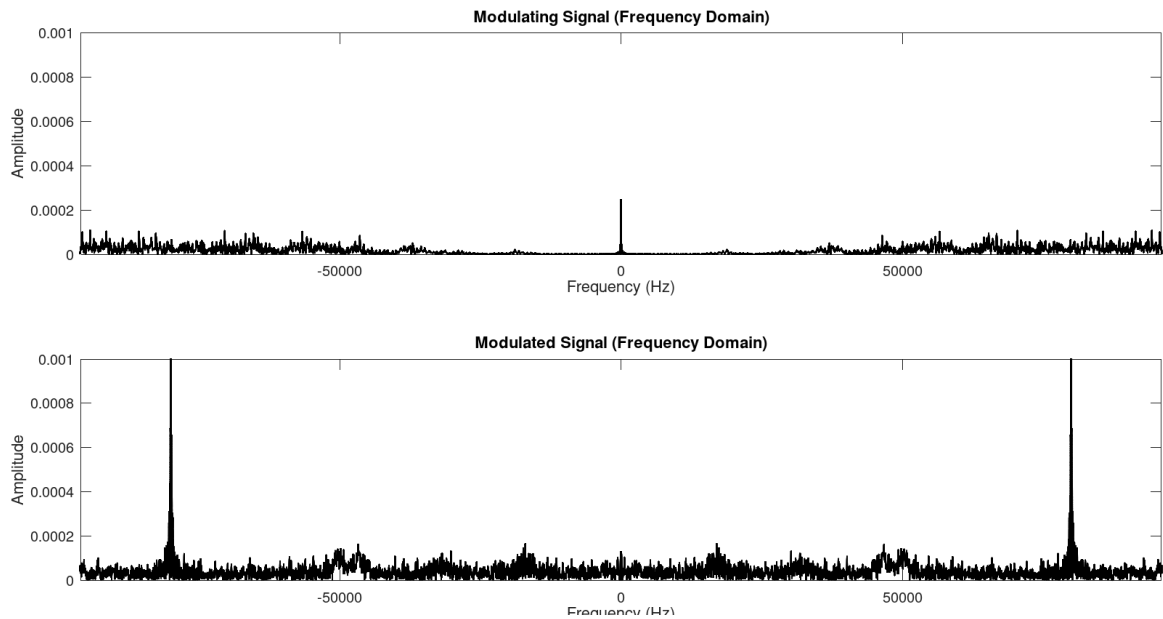
Figure 2: Elaborada pelo Autor



Sinal modulado em FM no domínio do tempo e da frequência

Uma vez com o sinal modulado, e multiplexado, podemos transmiti-lo pelo meio físico sem que haja interferência entre cada portadora (idealmente). O sinal no meio físico é ilustrado abaixo em azul.

Figure 3: Elaborada pelo Autor



Sinal modulado e “transmitido” no meio físico

Na recepção do sinal, precisamos realizar sua demodulação para ter novamente o sinal de áudio original. Para isso, utilizamos um demodulador FM, que é basicamente um circuito que realiza a derivação do sinal modulado, conforme o script abaixo:

## 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
17 the frequency of the modulating.
18 b = d_f/f_modulating_max;
19
20 % Defining the period and frequency of sampling:
21 fs = 50*f_carrier;
22 Ts = 1/fs;
23 T = 1/f_modulating_max;
```

```

24
25 % Defining the sinal period.
26 t_inicial = 0;
27 t_final = 2;
28
29 % "t" vector, correspondig to the time period of analysis, on time domain.
30 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 bibliograficas