



**INSTITUTO  
FEDERAL**

Santa Catarina

---

Câmpus  
São José

# **Modulação de Fase e Quadratura (IQ)**

Sistemas de Comunicação I

**Arthur Cadore Matuella Barcella**

07 de Abril de 2024

# Sumário

<b>1. Introdução .....</b>	<b>3</b>
<b>2. Fundamentação teórica .....</b>	<b>3</b>
2.1. Principais Conceitos .....	3
2.2. Resumo dos Itens abordados (Material de Referência) .....	4
2.2.1. Item 5.1 (Real and Complex Signals — it's all Sines and Cosines) .....	4
2.2.2. Item 5.4 (Quadrature Modulation and Demodulation (QAM) .....	4
2.2.3. Item 5.5 (Quadrature Amplitude Modulation using Complex Notation) .....	4
2.2.4. Item 5.6 (Quadrature Amplitude Demodulation using Complex Notation) ...	4
2.2.5. Item 5.7 (Spectral Representation for Complex Demodulation) .....	4
2.2.6. Item 5.8 (Frequency Offset Error and Correction at the Receiver) .....	4
2.2.7. Item 5.9 (Frequency Correction using a Complex Exponential) .....	5
2.2.8. Item 5.10 (RTL-SDR Quadrature / Complex Architecture) .....	5
<b>3. Análise dos resultados .....</b>	<b>5</b>
<b>4. Scripts e Codigos .....</b>	<b>8</b>
4.1. Etapa 1 - Import do sinal e definições: .....	8
4.2. Etapa 2 - Modulação IQ dos sinais: .....	9
4.3. Etapa 3 - Demodulação dos sinais: .....	10
4.4. Etapa 4 - Filtragem: .....	11
4.5. Etapa 5 - Comparando com o sinal original: .....	12
4.6. Etapa 6 - Calculando a densidade espectral de potência: .....	12
<b>5. Conclusões .....</b>	<b>13</b>
<b>6. Referências .....</b>	<b>13</b>

# 1. Introdução

O objetivo deste relatório, é apresentar os conceitos teóricos para a transmissão de dados utilizando modulação de fase e quadratura (IQ) através de dois diferentes sinais modulados por senos e cossenos.

Neste relatório, será apresentado a fundamentação teórica, os conceitos teóricos utilizados, a análise dos resultados, os scripts e códigos utilizados, as conclusões e as referências bibliográficas, de maneira em que serão utilizados dois sinais modulantes distintos, o primeiro irá modular um sinal senoidal (que será a portadora) e o segundo irá modular um sinal cossenoidal.

Desta forma, poderemos verificar a eficiência da modulação IQ na transmissão de dados, utilizando a mesma região do espectro eletromagnético para transmitir dois sinais distintos sem interferência (idealmente).

## 2. Fundamentação teórica

### 2.1. Principais Conceitos

Os principais conceitos teóricos abordados neste relatório são:

- **Modulação (IQ):** A modulação IQ é uma técnica de modulação digital que utiliza dois sinais modulados por senos e cossenos de igual fase para transmitir dados idealmente sem interferência. A modulação IQ é amplamente utilizada em sistemas de comunicação modernos, como o 4G e o 5G, devido à sua eficiência espectral e capacidade de transmitir dados de forma confiável.
- **Ortogonalidade:** A ortogonalidade é um conceito fundamental na modulação IQ, que garante que os sinais modulados por senos e cossenos sejam independentes e não interfiram uns nos outros. Isso permite que os dados sejam transmitidos de forma eficiente e confiável, mesmo em ambientes ruidosos.
- **Sinal Portador:** A portadora é o que permite a transmissão do sinal, suas características são ideais para o meio onde a onda irá se propagar, a portadora é influenciada pela modulante de maneira em que o receptor consiga extrair a informação transmitida.
- **Sinal Modulante:** A modulante é a informação que será transmitida, ela é a responsável por alterar as características da portadora, de maneira que o receptor consiga extrair a informação transmitida.
- **Processo de Modulação/Demodulação:** O processo de modulação consiste em alterar (normalmente através de multiplicação) as características da portadora de acordo com a modulante, de maneira em que o receptor consiga extrair a informação transmitida. Nos processos de modulação vistos até o momento, realizamos através de mutiplicação (ou no caso da FM, através de integração do argumento de fase). Já o processo de demodulação consiste em, uma vez com o sinal modulado, transmitido, e recebido no destinatário, extrair a informação do sinal portador, de maneira em que o receptor consiga interpretar a informação transmitida

e utiliza-la a nível de aplicação (como na rádio FM por exemplo, onde a música é recebida interpretada e reproduzida no carro).

## **2.2. Resumo dos Itens abordados (Material de Referência)**

Além dos conceitos base apresentados acima, também está abaixo um resumo dos principais conceitos teóricos das sessões 5.1, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9 e 5.10 do livro “Software Defined Radio Using MATLAB & Simulink and the RTL-SDR”:

### **2.2.1. Item 5.1 (Real and Complex Signals — it’s all Sines and Cosines)**

Objetivo: Explorar a representação de sinais reais e complexos na engenharia de sinais, utilizando a fórmula de Euler para simplificar matematicamente a manipulação de sinais em termos de exponenciais complexas. Conteúdo Principal: Demonstra que, embora operemos com sinais reais no mundo físico (como tensões induzidas por campos eletromagnéticos), a representação complexa dos sinais facilita a análise matemática. A seção se aprofunda na identidade de Euler e na conversão de sinais sinusoidais para sua forma exponencial complexa, permitindo uma manipulação matemática mais simples.

### **2.2.2. Item 5.4 (Quadrature Modulation and Demodulation (QAM))**

Objetivo: Introduzir o conceito de modulação em quadratura e demonstrar sua implementação no RTL-SDR. Conteúdo Principal: Explica a motivação por trás do uso da modulação em quadratura para melhorar a eficiência espectral. Utiliza sinais de exemplo para ilustrar a modulação e demodulação em quadratura, destacando o processo de recuperação de sinais de banda base a partir do sinal recebido.

### **2.2.3. Item 5.5 (Quadrature Amplitude Modulation using Complex Notation)**

Objetivo: Descrever a modulação de amplitude em quadratura (QAM) usando notação complexa. Conteúdo Principal: Detalha como a notação complexa pode simplificar a representação e manipulação de sinais QAM, apresentando um exemplo prático de como os sinais podem ser modulados para transmissão.

### **2.2.4. Item 5.6 (Quadrature Amplitude Demodulation using Complex Notation)**

Objetivo: Explorar o processo de demodulação de amplitude em quadratura usando notação complexa. Conteúdo Principal: Apresenta o método de demodulação QAM, evidenciando como a notação complexa facilita o processo de extração de sinais de banda base do sinal recebido.

### **2.2.5. Item 5.7 (Spectral Representation for Complex Demodulation)**

Objetivo: Examinar a representação espectral de sinais complexos e sua aplicação na demodulação. Conteúdo Principal: Demonstra a transformação espectral de sinais modulados em frequência para sua baseband complexa, detalhando o impacto da modulação complexa nas propriedades espectrais dos sinais.

### **2.2.6. Item 5.8 (Frequency Offset Error and Correction at the Receiver)**

Objetivo: Discutir o erro de deslocamento de frequência no receptor e como corrigi-lo. Conteúdo Principal: Explica o conceito de erro de deslocamento de frequência, suas causas potenciais e métodos para correção, especialmente em relação ao uso do RTL-SDR.

### 2.2.7. Item 5.9 (Frequency Correction using a Complex Exponential)

Objetivo: Descrever um método de correção de frequência usando uma exponencial complexa.

Conteúdo Principal: Apresenta uma técnica para ajustar a frequência de sinais recebidos por meio da multiplicação por uma exponencial complexa, abordando o contexto de sua aplicação prática.

### 2.2.8. Item 5.10 (RTL-SDR Quadrature / Complex Architecture)

Objetivo: Conectar os conceitos de modulação complexa e demodulação com a arquitetura do RTL-SDR. Conteúdo Principal: Explica como os princípios de modulação e demodulação em quadratura são implementados na arquitetura do RTL-SDR, preparando o terreno para aplicações práticas discutidas nos capítulos seguintes.

## 3. Análise dos resultados

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

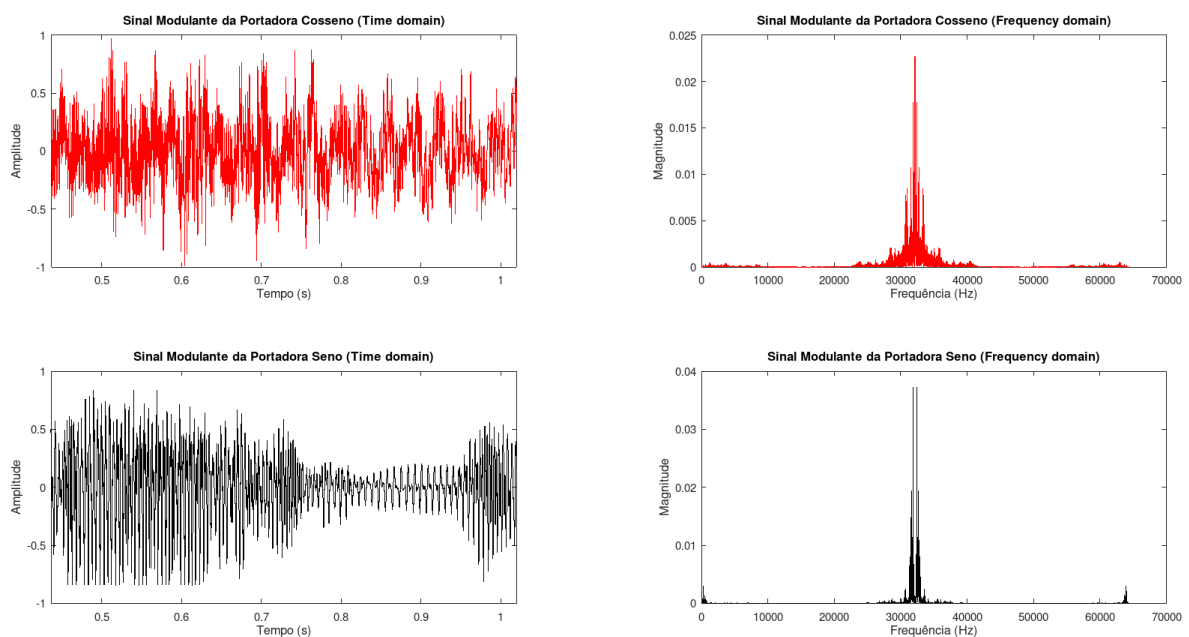


Figura 1: Sinal de Áudio aplicado como modulante

Figura elaborada pelo autor

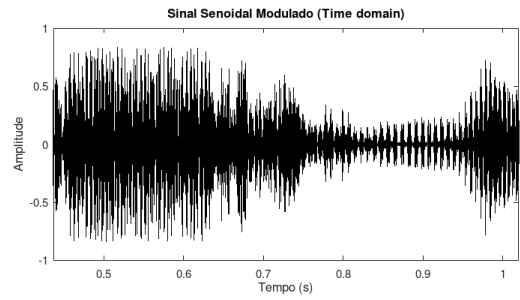
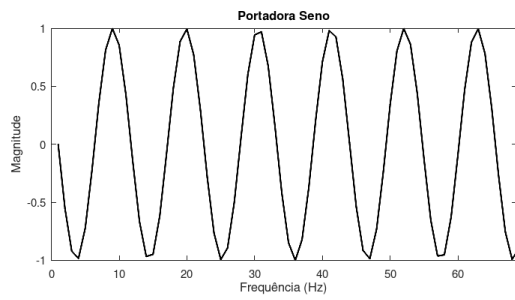
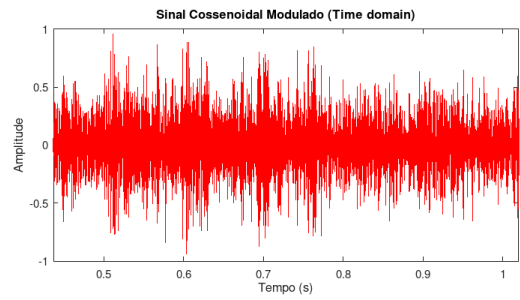
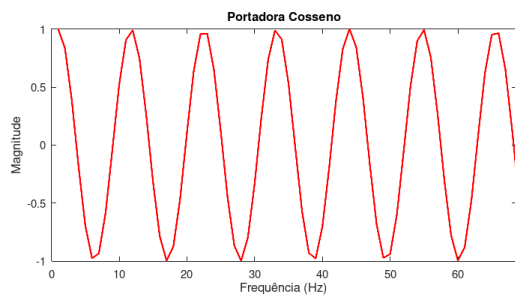


Figura 2: Sinal de Áudio aplicado como modulante  
Figura elaborada pelo autor

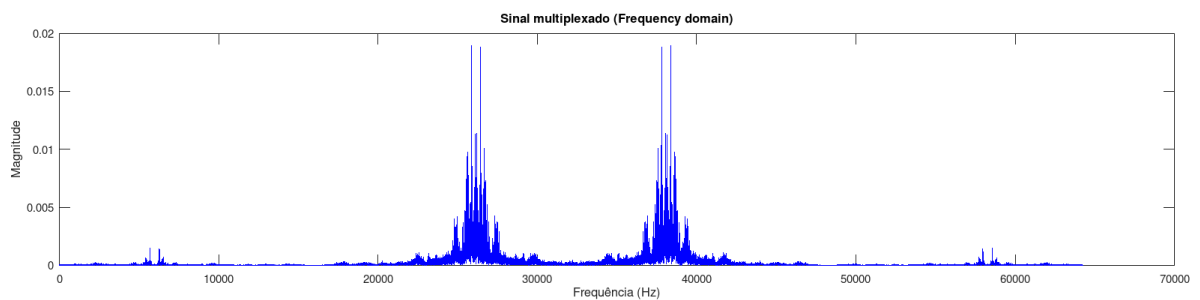
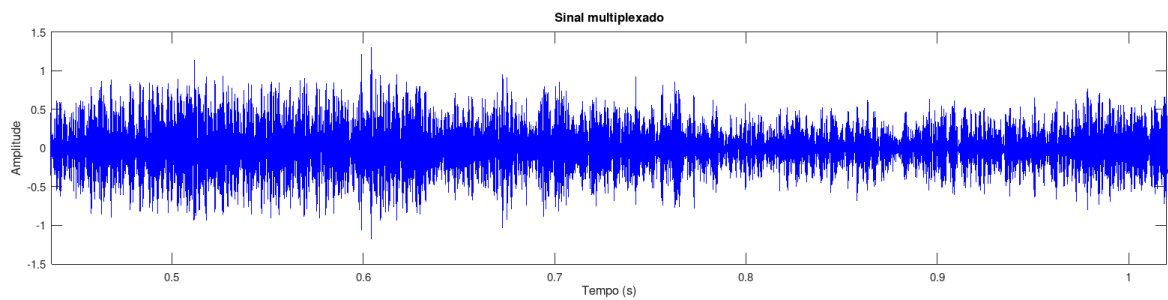


Figura 3: Sinal de Áudio aplicado como modulante  
Figura elaborada pelo autor

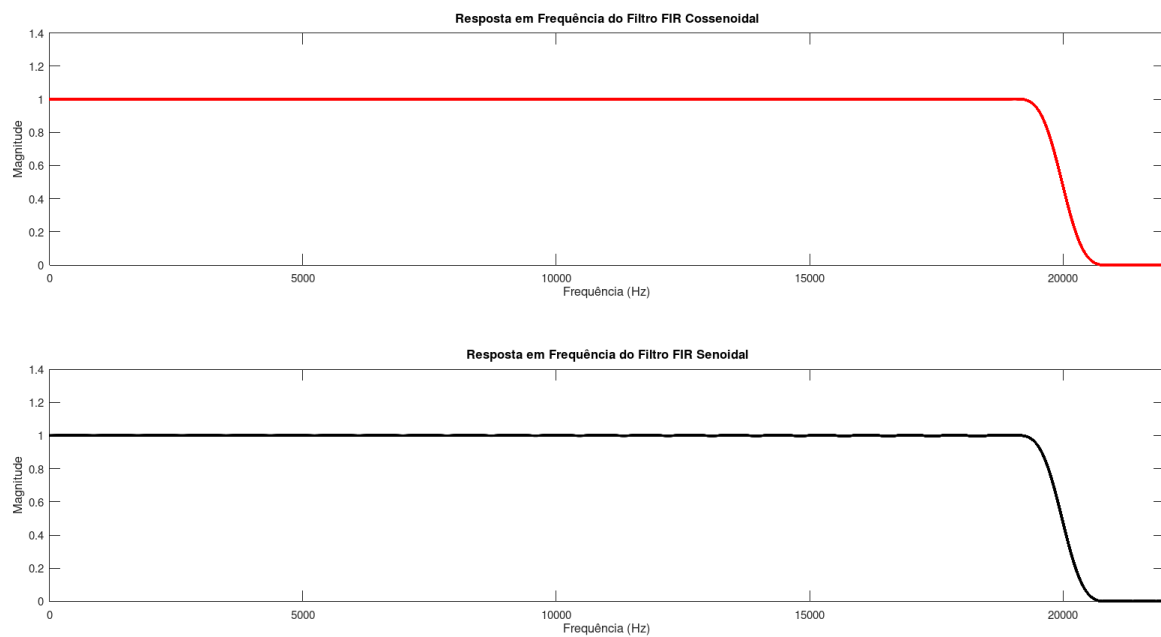


Figura 4: Sinal de Áudio aplicado como modulante  
Figura elaborada pelo autor

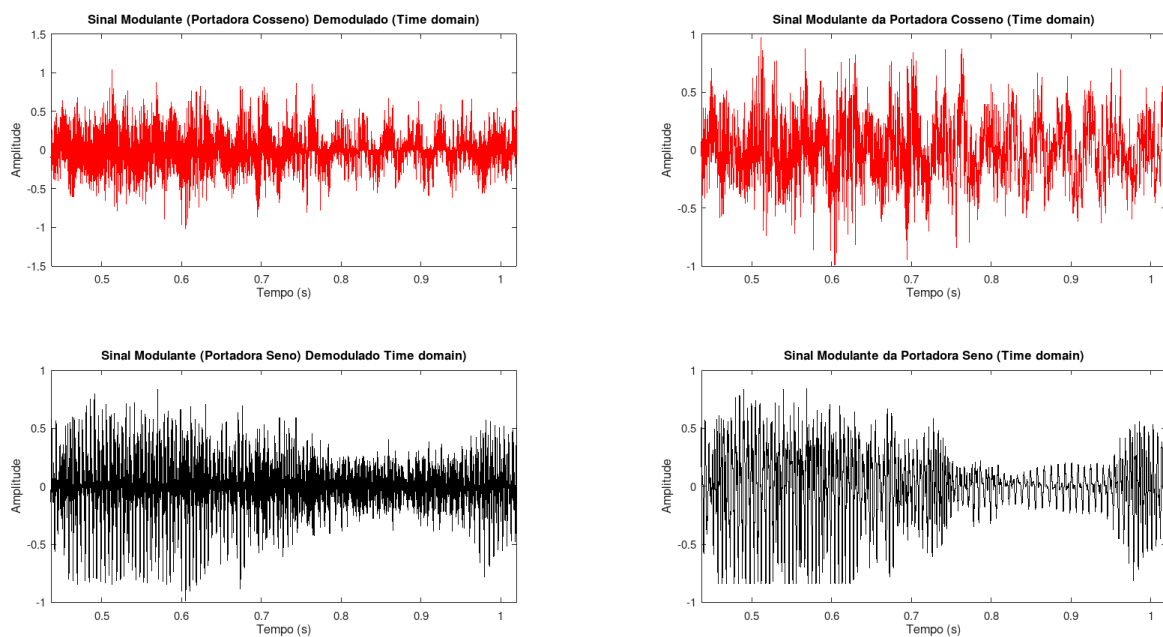


Figura 5: Sinal de Áudio aplicado como modulante  
Figura elaborada pelo autor

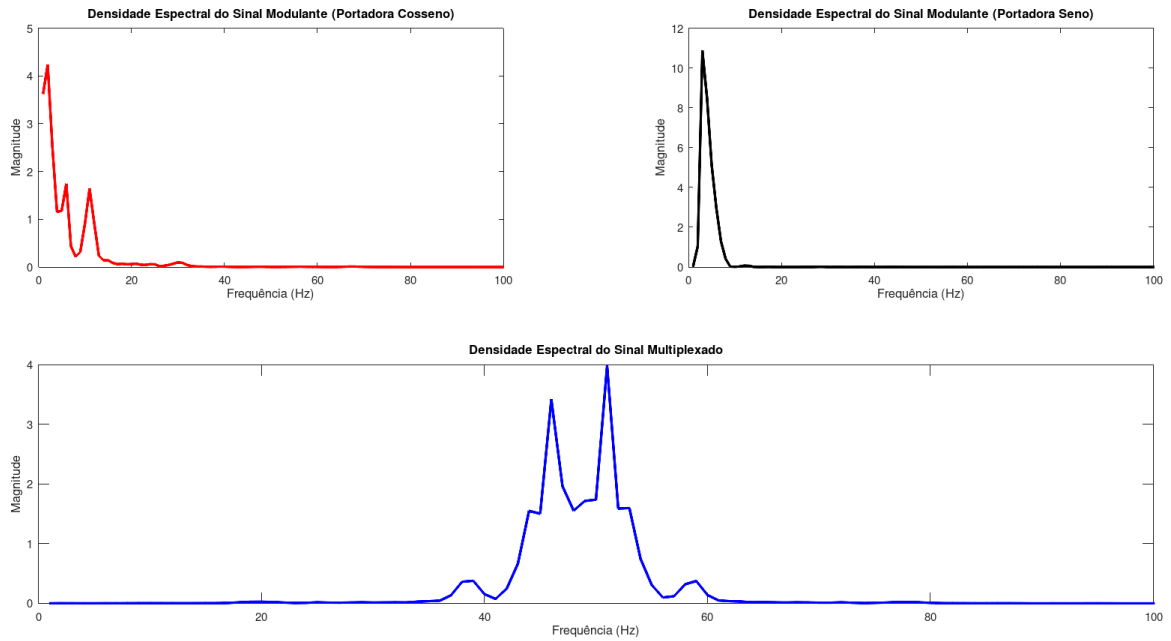


Figura 6: Sinal de Áudio aplicado como modulante  
Figura elaborada pelo autor

## 4. Scripts e Codigos

Abaixo estão os scripts e códigos utilizados para a realização deste relatório:

### 4.1. Etapa 1 - Import do sinal e definições:

```

1 % IQ transmission of two diferent audio signals.
2 % IQ - In-Phase and Quadrature Modulation
3
4 clc; clear all; close all
5 pkg load signal
6
7 % Definição dos parâmetros da portadora do sinal IQ:
8 carrier_amplitude = 1;
9 carrier_frequency = 40000;
10
11 % Coletando os sinais para transmissão:
12 [short_signal, Fs] = audioread('short-signal.wav');
13 [long_signal, Fs2] = audioread('long-signal.wav');
14
15 % Fazendo a transposição linha/coluna do sinal de entrada
16 short_signal = transpose(short_signal);
17 long_signal = transpose(long_signal);
18
19 % pegando a duração da transmissão a partir do tamanho do menor sinal;
20 duracao = length(short_signal)/Fs;
21
22 % calculando vetor de t no dominio do tempo;
23 Ts = 1/Fs;
24 t=[0:Ts:duracao-Ts];
25
26 % Igualando o comprimento dos sinais ao vetor de tempo
27 signal_cos = short_signal(1:length(t));

```



```

28 signal_sin = long_signal(1:length(t));
29
30 % calculando o passo no dominio da frequência;
31 f_step = 1/duracao;
32
33 % vetor "f" correspondente ao periodo de análise (dominio da frequência);
34 f = [-Fs/2:f_step:Ff/2];
35 f = [1:length(signal_cos)];
36
37 % calculando a FFT do sinal de entrada (que será utilizado no cosseno):
38 signal_cos_F = fft(signal_cos)/length(signal_cos);
39 signal_cos_F = fftshift(signal_cos_F);
40
41 % calculando a FFT do sinal de entrada (que será utilizado no seno):
42 signal_sin_F = fft(signal_sin)/length(signal_sin);
43 signal_sin_F = fftshift(signal_sin_F);
44
45 % Plot dos sinais de entrada (dominio do tempo e frequência):
46 figure(1)
47 subplot(221)
48 plot(t,signal_cos,'r')
49 xlim([(duracao*0.3) (duracao*0.7)])
50 title('Sinal Modulante da Portadora Cosseno (Time domain)')
51 xlabel('Tempo (s)')
52 ylabel('Amplitude')
53
54 subplot(223)
55 plot(t,signal_sin,'k')
56 xlim([(duracao*0.3) (duracao*0.7)])
57 title('Sinal Modulante da Portadora Seno (Time domain)')
58 xlabel('Tempo (s)')
59 ylabel('Amplitude')
60
61 subplot(222)
62 plot(f,abs(signal_cos_F), 'r')
63 title('Sinal Modulante da Portadora Cosseno (Frequency domain)')
64 xlabel('Frequência (Hz)')
65 ylabel('Magnitude')
66
67 subplot(224)
68 plot(f,abs(signal_sin_F), 'k')
69 title('Sinal Modulante da Portadora Seno (Frequency domain)')
70 xlabel('Frequência (Hz)')
71 ylabel('Magnitude')

```

## 4.2. Etapa 2 - Modulação IQ dos sinais:

```

1 % Criando dos sinais de portadora para transmissão ortogonal (um com seno
  e outro com cosseno):
2 carrier_cos = carrier_amplitude*cos(2*pi*carrier_frequency*t);
3 carrier_sin = carrier_amplitude*sin(2*pi*carrier_frequency*t);
4
5 % Realizando a modulação AM do sinal de áudio na portadora correspondente
  a cada sinal:
6 modulated_cos = signal_cos .* carrier_cos;
7 modulated_sin = signal_sin .* carrier_sin;
8

```

```

9 % Realizando a multiplexação do sinal (a partir do princípio de
  ortogonalidade):
10 multiplexed_signal = modulated_cos + modulated_sin;
11
12 % Calculando a FFT do sinal para amostrar seu estado no domínio da frequência:
13 multiplexed_signal_F = fft(multiplexed_signal)/length(multiplexed_signal);
14 multiplexed_signal_F = fftshift(multiplexed_signal_F);
15
16 figure(2)
17 subplot(221)
18 plot(f,carrier_cos,'r', 'LineWidth', 2)
19 xlim([0 100*f_step])
20 title('Portadora Cosseno')
21 xlabel('Frequência (Hz)')
22 ylabel('Magnitude')
23
24 subplot(223)
25 plot(f,carrier_sin,'k', 'LineWidth', 2)
26 xlim([0 100*f_step])
27 title('Portadora Seno')
28 xlabel('Frequência (Hz)')
29 ylabel('Magnitude')
30
31 subplot(222)
32 plot(t,modulated_cos,'r')
33 xlim([(duracao*0.3) (duracao*0.7)])
34 title('Sinal Cossenoidal Modulado (Time domain)')
35 xlabel('Tempo (s)')
36 ylabel('Amplitude')
37
38 subplot(224)
39 plot(t,modulated_sin,'k')
40 xlim([(duracao*0.3) (duracao*0.7)])
41 title('Sinal Senoidal Modulado (Time domain)')
42 xlabel('Tempo (s)')
43 ylabel('Amplitude')
44
45 % Verificando o sinal multiplexado:
46
47 figure(3)
48 subplot(211)
49 plot(t,multiplexed_signal,'b')
50 xlim([(duracao*0.3) (duracao*0.7)])
51 title('Sinal multiplexado')
52 xlabel('Tempo (s)')
53 ylabel('Amplitude')
54
55 subplot(212)
56 plot(f,abs(multiplexed_signal_F), 'b')
57 title('Sinal multiplexado (Frequency domain)')
58 xlabel('Frequência (Hz)')
59 ylabel('Magnitude')

```

### 4.3. Etapa 3 - Demodulação dos sinais:

```

1 % Realizando a demodulação do sinal no receptor:
2

```

```

3 demodulated_cos = multiplexed_signal .* carrier_cos;
4 demodulated_sin = multiplexed_signal .* carrier_sin;
5
6 % Ordem do filtro FIR
7 filtro_ordem = 100;
8
9 % Frequência de corte do filtro FIR
10 % Como trata-se de um sinal de áudio, a frequência de corte pode ser fixada
    em 20kHz
11 frequencia_corte = 20000;
12
13 % Coeficientes do filtro FIR para cada sinal demodulado
14 coeficientes_filtro = fir1(filtro_ordem, frequencia_corte/(Fs/2));
15
16 % Resposta em frequência do filtro FIR para cada sinal demodulado
17 [H_cos, f_cos] = freqz(coeficientes_filtro, 1, length(t), Fs);
18 [H_sin, f_sin] = freqz(coeficientes_filtro, 1, length(t), Fs);
19
20 % Plot da resposta em frequência dos filtros:
21 figure(5)
22 subplot(211)
23 plot(f_cos, abs(H_cos), 'r', 'LineWidth', 3)
24 xlim([0 frequencia_corte*1.1])
25 title('Resposta em Frequência do Filtro FIR Cossenoidal')
26 xlabel('Frequência (Hz)')
27 ylabel('Magnitude')
28
29 subplot(212)
30 plot(f_sin, abs(H_sin), 'k', 'LineWidth', 3)
31 xlim([0 frequencia_corte*1.1])
32 title('Resposta em Frequência do Filtro FIR Senoidal')
33 xlabel('Frequência (Hz)')
34 ylabel('Magnitude')

```

#### 4.4. Etapa 4 - Filtragem:

```

1 % Filtragem dos sinais demodulados
2 demodulated_cos_filtered = filter(coeficientes_filtro, 1, demodulated_cos);
3 demodulated_sin_filtered = filter(coeficientes_filtro, 1, demodulated_sin);
4
5 % Calculando a FFT dos sinais demodulados para amostrar seu estado no dominio
    da frequência:
6 demodulated_sin_F = fft(demodulated_sin_filtered)/
    length(demodulated_sin_filtered);
7 demodulated_sin_F = fftshift(demodulated_sin_F);
8
9 demodulated_cos_F = fft(demodulated_cos_filtered)/
    length(demodulated_cos_filtered);
10 demodulated_cos_F = fftshift(demodulated_cos_F);
11
12 % Plot dos sinais demodulados
13 figure(4)
14 subplot(221)
15 plot(t, demodulated_cos_filtered, 'r')
16 xlim([(duracao*0.3) (duracao*0.7)])
17 title('Sinal Modulante (Portadora Cosseno) Demodulado (Time domain)')
18 xlabel('Tempo (s)')

```

```

19 ylabel('Amplitude')
20
21 subplot(223)
22 plot(t, demodulated_sin_filtered, 'k')
23 xlim([(duracao*0.3) (duracao*0.7)])
24 title('Sinal Modulante (Portadora Seno) Demodulado Time domain')
25 xlabel('Tempo (s)')
26 ylabel('Amplitude')
27
28 subplot(222)
29 plot(f,abs(demodulated_cos_F), 'r')
30 title('Sinal Modulante (Portadora Cosseno) Demodulado (Frequency domain)')
31 xlabel('Frequência (Hz)')
32 ylabel('Magnitude')
33
34 subplot(224)
35 plot(f,abs(demodulated_sin_F), 'k')
36 title('Sinal Modulante (Portadora Seno) Demodulado (Frequency domain)')
37 xlabel('Frequência (Hz)')
38 ylabel('Magnitude')

```

#### 4.5. Etapa 5 - Comparando com o sinal original:

```

1 % Comparando sinal transmitido com sinal recebido:
2
3 figure(4)
4 subplot(221)
5 plot(t, demodulated_cos_filtered, 'r')
6 xlim([(duracao*0.3) (duracao*0.7)])
7 title('Sinal Modulante (Portadora Cosseno) Demodulado (Time domain)')
8 xlabel('Tempo (s)')
9 ylabel('Amplitude')
10
11 subplot(223)
12 plot(t, demodulated_sin_filtered, 'k')
13 xlim([(duracao*0.3) (duracao*0.7)])
14 title('Sinal Modulante (Portadora Seno) Demodulado Time domain')
15 xlabel('Tempo (s)')
16 ylabel('Amplitude')
17
18 subplot(222)
19 plot(t,signal_cos,'r')
20 xlim([(duracao*0.3) (duracao*0.7)])
21 title('Sinal Modulante da Portadora Cosseno (Time domain)')
22 xlabel('Tempo (s)')
23 ylabel('Amplitude')
24
25 subplot(224)
26 plot(t,signal_sin,'k')
27 xlim([(duracao*0.3) (duracao*0.7)])
28 title('Sinal Modulante da Portadora Seno (Time domain)')
29 xlabel('Tempo (s)')
30 ylabel('Amplitude')

```

#### 4.6. Etapa 6 - Calculando a densidade espectral de potência:

```

1 % Calculando a densidade espectral dos sinais:
2 figure(7)
3 subplot(221)
4 plot(pwelch(signal_cos), 'r', 'LineWidth', 3);
5 xlim([0 200])
6 title('Densidade Espectral do Sinal Modulante (Portadora Cosseno)')
7 xlabel('Frequência (Hz)')
8 ylabel('Magnitude')
9 xlim([0 100])
10
11 subplot(222)
12 plot(pwelch(signal_sin), 'k', 'LineWidth', 3);
13 xlim([0 200])
14 title('Densidade Espectral do Sinal Modulante (Portadora Seno)')
15 xlabel('Frequência (Hz)')
16 ylabel('Magnitude')
17 xlim([0 100])
18
19 subplot(2, 2, [3 4])
20 plot(pwelch(multiplexed_signal), 'b', 'LineWidth', 3);
21 xlim([0 100])
22 title('Densidade Espectral do Sinal Multiplexado')
23 xlabel('Frequência (Hz)')
24 ylabel('Magnitude')

```

## 5. Conclusões

Seção V - Conclusões

## 6. Referências

Para o desenvolvimento deste relatório, foi utilizado o seguinte material de referência:

- Software Defined Radio Using MATLAB & Simulink and the RTL-SDR, de Robert W. Stewart