



**INSTITUTO
FEDERAL**

Santa Catarina

Câmpus
São José

Modulador 16-QAM

Sistemas de Comunicação I

Arthur Cadore Matuella Barcella

30 de Junho de 2024

Sumário

1. Introdução:	3
2. Desenvolvimento e Resultados:	3
2.1. Parte 1:	3
2.1.1. Definindo parâmetros de execução:	3
2.1.2. Realizando a modulação QAM:	4
2.1.3. Upsampling do sinal:	5
2.1.4. Modulando o sinal para transmissão:	6
2.1.5. Criando o sinal de transmissão:	7
2.1.6. Demodulando o sinal recebido:	8
2.1.7. Filtrando o sinal demodulado:	9
2.1.8. Realizando o downsampling do sinal:	10
2.1.9. Plotando o sinal QAM Transmitido e Recebido:	11
2.2. Parte 2:	13
2.2.1. Definindo parâmetros de execução:	13
2.2.2. Modulando o sinal QAM:	14
2.2.3. Realizando o Upsampling do sinal:	14
2.2.4. Modulando o sinal para transmissão:	14
2.2.5. Demodulando o sinal recebido:	15
2.2.6. Filtrando o sinal demodulado:	16
2.2.7. Realizando o downsampling do sinal:	16
2.2.8. Reconstruindo o sinal QAM Transmitido:	17
2.2.9. Comparação das componentes real e imaginária:	18
3. Conclusão:	19
4. Referências Bibliográficas:	19

1. Introdução:

O objetivo deste relatório é realizar a transmissão e recepção de um sinal digital utilizando a modulação QAM (Quadrature Amplitude Modulation), o sinal será composto por um vetor de dados aleatórios com 1000 elementos, onde cada elemento obtido através dos dados aleatórios representa um diferente símbolo QAM. O sinal será modulado e transmitido, e posteriormente será recebido e demodulado, sendo possível comparar o sinal transmitido com o sinal recebido.

2. Desenvolvimento e Resultados:

O desenvolvimento do relatório foi dividido em duas partes, na primeira parte foi realizado a transmissão e recepção do sinal QAM utilizando a modulação QAM com a representação do sinal em fase e quadratura, e na segunda parte foi realizado a transmissão e recepção do sinal QAM utilizando a modulação QAM com a representação complexa.

2.1. Parte 1:

Para o desenvolvimento da primeira parte, foi estruturado um script em octave para realizar a transmissão e recepção do sinal QAM utilizando a modulação QAM com a representação do sinal em fase e quadratura.

2.1.1. Definindo parâmetros de execução:

A primeira etapa do desenvolvimento, é a definição das variáveis que serão utilizadas nos processos de modulação e demodulação do sinal QAM. Desta forma, defini os seguintes parâmetros:

```
1 %% Inicializando pacotes necessários:
2 clc; close all; clear all;
3 pkg load communications;
4
5 % Definindo o n° de símbolos QAM
6 M = 16;
7
8 % Definindo o fator de upsampling
9 n = 100;
10
11 % Definindo a taxa de bits de TX
12 Rb = 1e4;
13
14 % Definindo o período de bit
15 Tb = 1 / Rb;
16
17 % Definindo a frequência de amostragem
18 Fs = Rb * n;
19
20 % Definindo a Frequência de portadora
21 fc = Fs / 50;
22
23 % Definindo o Período de amostragem:
24 Ts = 1 / Fs;
```

```

25
26 % Definindo o SNR do sinal de transmissão:
27 SNR = 12;
28
29 % Definindo o filtro FIR passa-baixa para a recepção:
30 filtro_passa_baixa = fir1(100, fc/(Fs/2));

```

Em seguida, foi estruturado também o vetor de dados que será utilizado para a modulação do sinal QAM, para isso, foi gerado um vetor de dados aleatórios com 1000 elementos.

```

1 % Criando o vetor de dados:
2 Vector_length = 1000;
3 info = randi([0 M-1], 1, Vector_length);

```

2.1.2. Realizando a modulação QAM:

Uma vez com os parâmetros definidos e o vetor de dados gerado, o primeiro passo foi realizar a modulação do sinal QAM, onde o sinal foi modulado utilizando a função `qammod` do pacote de comunicações do octave. Em seguida, os dados gerados pela função de modulação QAM podem ser visualizados através de um diagrama de constelação, utilizando a função `scatterplot`.

```

1 % Modulação QAM:
2
3 % Modulando o sinal em QAM:
4 info_mod = qammod(info, M);
5
6 % Fazendo o plot do sinal modulado:
7 scatterplot(info_mod);
8 title('Diagrama de constelação QAM do sinal');
9 xlim([-5 5]);
10 ylim([-5 5]);
11 info_r_real = real(info_mod);
12 info_i_imag = imag(info_mod);
13
14 % Criando o vetor de tempo com base no comprimento da informação:
15 t = [0:Ts:(length(info_r_real) * Tb - Ts)];

```

Com base no diagrama de constelação gerado, é possível visualizar a representação dos símbolos QAM no plano complexo, onde cada símbolo é representado por um ponto no plano complexo, sendo a parte real do sinal representada no eixo x e a parte imaginária do sinal representada no eixo y:

Figure 1: Elaborada pelo Autor

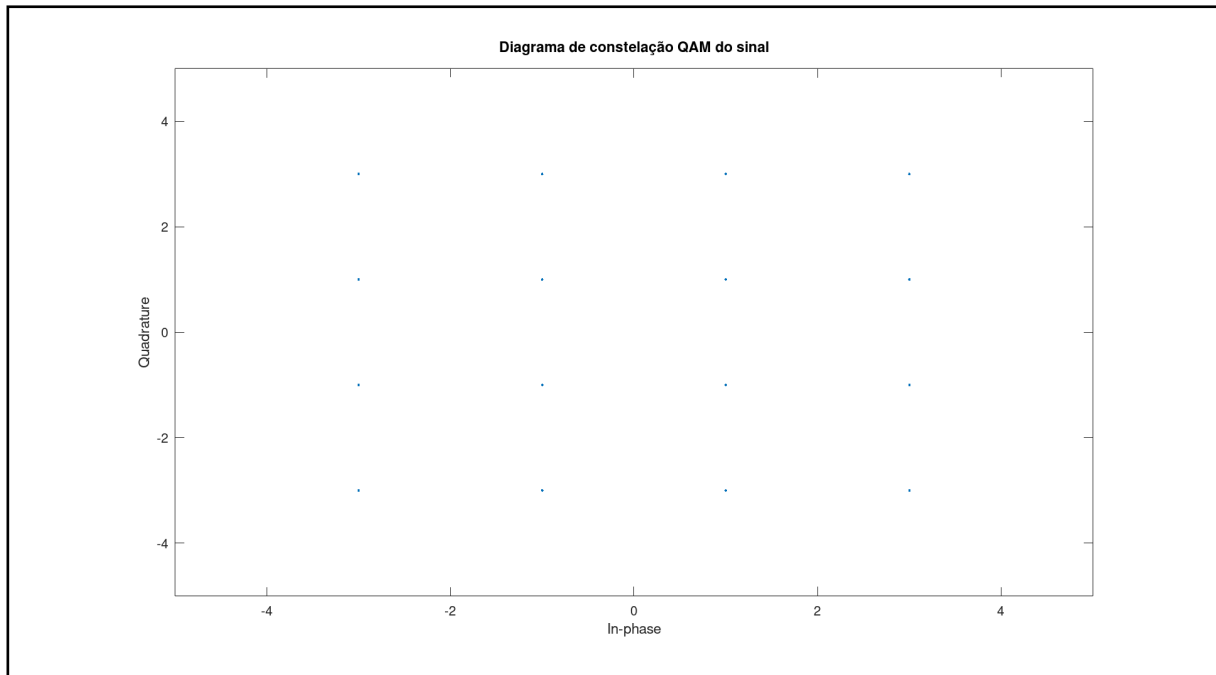


Diagrama de constelação QAM do sinal

2.1.3. Upsampling do sinal:

Em seguida, foi realizado o processo de upsampling do sinal, onde o sinal modulado foi expandido para a taxa de amostragem desejada. O objetivo desse processo é aumentar a taxa de amostragem do sinal, o que melhora a qualidade do sinal e facilita a filtragem do sinal.

Para isso, foi utilizado a função `upsample` do octave, que adiciona zeros entre as amostras. Em seguida, foi criado um filtro NRZ para realizar o upsample (valor positivo) do sinal, e o sinal foi filtrado utilizando a função `filter`.

```
1 % Criando um filtro NRZ para realizar o upsample do sinal:
2 filtro_NRZ = ones(1, n);
3 info_r_real_up = upsample(info_r_real, n); % Upsampling
4 info_r_real_tx = filter(filtro_NRZ, 1, info_r_real_up); % Filtragem
5
6 % Realizando o plot no dominio do tempo:
7 figure;
8 subplot(221);
9 plot(t(1:length(info_r_real_tx)), info_r_real_tx, 'LineWidth', 2, 'Color',
10 'k');
11 title('Sinal de Informação (Componente Real)');
12 xlabel('Tempo (s)');
13 ylabel('Amplitude');
14 xlim([0 10 * Tb]);
15 ylim([-5 5]);
16
17 info_i_imag_up = upsample(info_i_imag, n); % Upsampling
18 info_i_imag_tx = filter(filtro_NRZ, 1, info_i_imag_up); % Filtragem
19 subplot(222);
```

```

20 plot(t(1:length(info_i_imag_tx)), info_i_imag_tx, 'LineWidth', 2, 'Color',
    'r');
21 title('Sinal de Informação (Componente Imaginário)');
22 xlabel('Tempo (s)');
23 ylabel('Amplitude');
24 xlim([0 10 * Tb]);
25 ylim([-5 5]);

```

2.1.4. Modulando o sinal para transmissão:

Em seguida, o sinal foi modulado para a transmissão, onde foi criado um sinal portadora cosseno e um sinal portadora seno, e o sinal de informação foi multiplicado por essas portadoras para realizar a modulação do sinal.

```

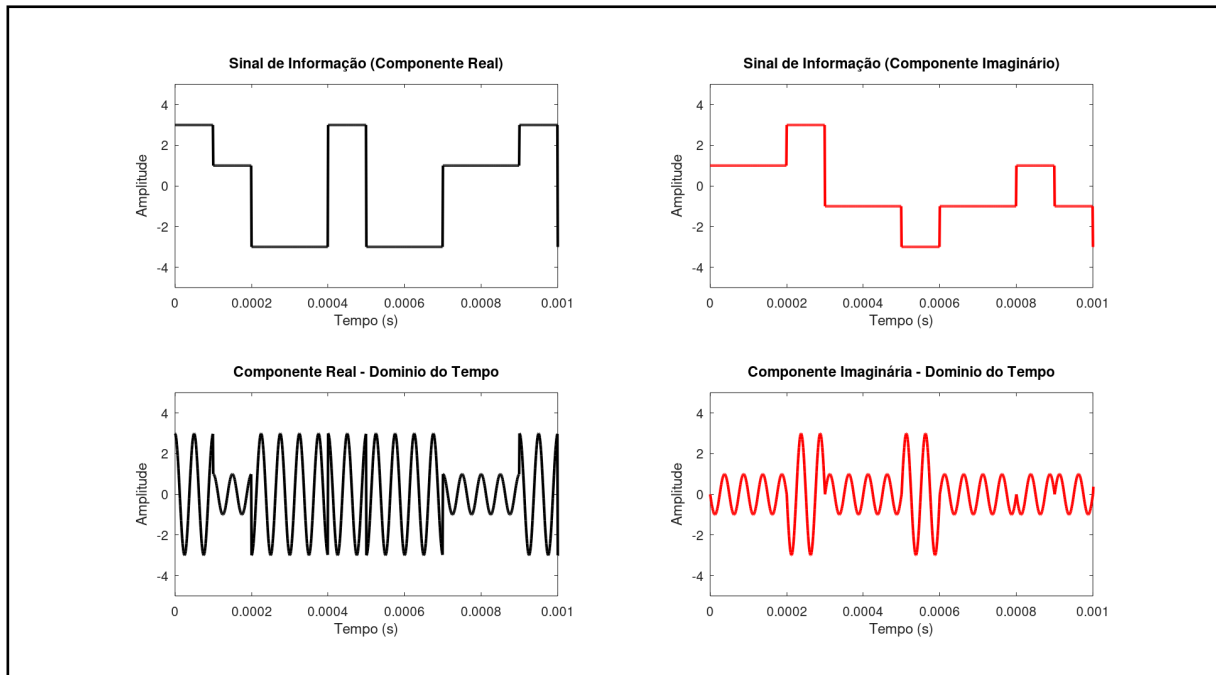
1 % Modulando para transmissão:
2
3 % Criando portadora Cosseno:
4 cos_carrier = cos(2 * pi * fc * t(1:length(info_r_real_tx)));
5 info_real_tx = info_r_real_tx .* cos_carrier;
6
7 % Criando portadora Seno:
8 sen_carrier = -sin(2 * pi * fc * t(1:length(info_i_imag_tx)));
9 info_imag_tx = info_i_imag_tx .* sen_carrier;
10
11 subplot(223);
12 plot(t(1:length(info_real_tx)), info_real_tx, 'LineWidth', 2, 'Color',
    'k');
13 title('Componente Real - Dominio do Tempo');
14 xlabel('Tempo (s)');
15 ylabel('Amplitude');
16 xlim([0 10 * Tb]);
17 ylim([-5 5]);
18
19 subplot(224);
20 plot(t(1:length(info_imag_tx)), info_imag_tx, 'LineWidth', 2, 'Color',
    'r');
21 title('Componente Imaginária - Dominio do Tempo');
22 xlabel('Tempo (s)');
23 ylabel('Amplitude');
24 xlim([0 10 * Tb]);
25 ylim([-5 5]);

```

Desta forma, foi possível visualizar o sinal de informação antes da modulação, note que após a expansão das amostras, existem diversos zeros e uns contínuos, o que permite uma visualização “quadrada” do sinal.

Também podemos ver o sinal após a modulação, onde o sinal foi multiplicado pela portadora cosseno e seno, e o sinal de informação foi modulado para a transmissão.

Figure 2: Elaborada pelo Autor



Componentes do sinal de transmissão

2.1.5. Criando o sinal de transmissão:

Uma vez com as componentes do sinal já moduladas, foi realizada a soma das componentes do sinal para obter o sinal de transmissão (Fase e Quadratura), o que resulta no sinal de transmissão QAM.

Em seguida, foi adicionado ruído ao sinal transmitido, utilizando a função `awgn` do octave, onde foi adicionado um ruído com a relação sinal ruído (SNR) de 12 dB. Foi utilizada a função `awgn` pois essa função adiciona ruído gaussiano branco ao sinal transmitido, o que a torna o mais próximo possível de um piso de ruído térmico.

O objetivo de adicionar ruído ao sinal transmitido é simular o ambiente de transmissão real, onde o sinal é afetado por ruídos e interferências, e assim, é possível avaliar a qualidade do sinal recebido.

```
1 % Criando o sinal de transmissão:
2
3 sinal_tx = info_real_tx + info_imag_tx;
4
5 figure;
6 subplot(211);
7 plot(t(1:length(sinal_tx)), sinal_tx, 'LineWidth', 2);
8 title('Sinal de Transmissão - Domínio do tempo');
9 xlabel('Tempo (s)');
10 ylabel('Amplitude');
11 xlim([0 10 * Tb]);
12 ylim([-5 5]);
13
14 % Adicionando ruído ao sinal transmitido
```

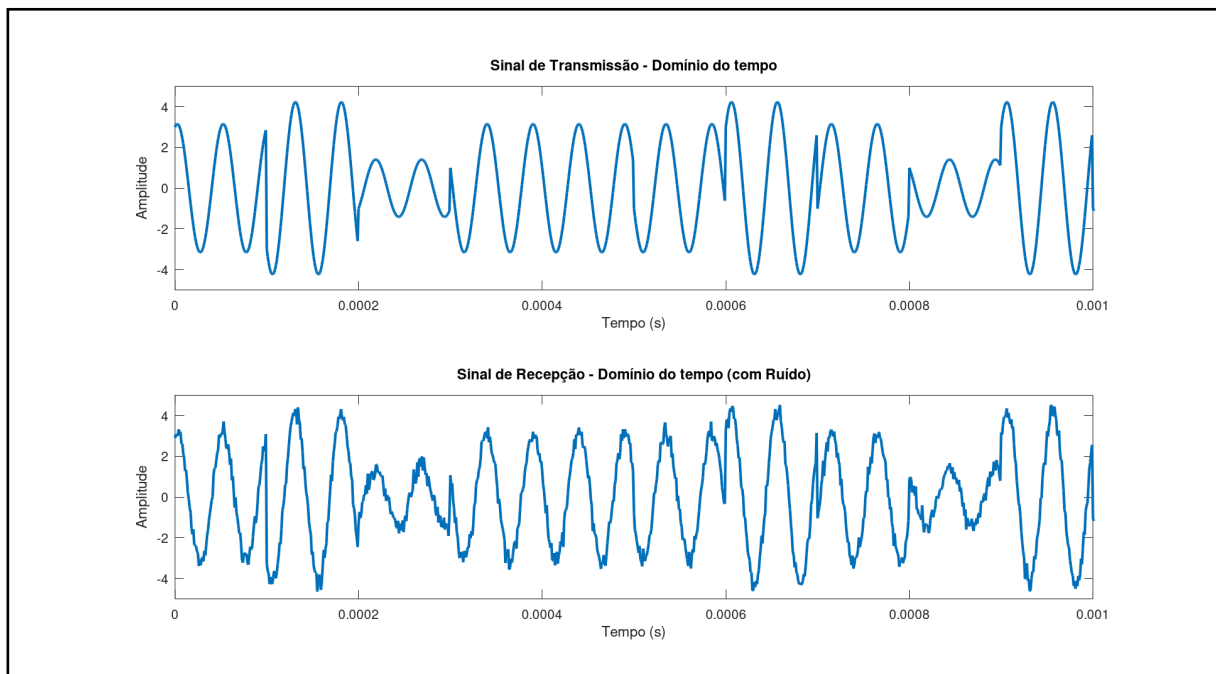
```

15 sinal_recebido = awgn(sinal_tx, SNR);
16
17 subplot(2,1,2);
18 plot(t(1:length(sinal_recebido)), sinal_recebido, 'LineWidth', 2);
19 title('Sinal de Recepção - Domínio do tempo (com Ruído)');
20 xlabel('Tempo (s)');
21 ylabel('Amplitude');
22 xlim([0 10 * Tb]);
23 ylim([-5 5]);

```

Desta forma, foi possível visualizar o sinal de transmissão e o sinal recebido com ruído, onde é possível observar a diferença entre o sinal transmitido e o sinal recebido, e a presença do ruído no sinal recebido.

Figure 3: Elaborada pelo Autor



Sinal de Transmissão (Sem e com ruído)

2.1.6. Demodulando o sinal recebido:

Na recepção do sinal, o sinal recebido foi demodulado utilizando a representação do sinal em fase e quadratura, onde o sinal recebido foi multiplicado pela portadora cosseno e seno, retornando o sinal para a banda base.

```

1 % Demodulação do sinal de recepção:
2
3 % Ajuste do vetor de tempo:
4 t_rx = [0:Ts:(length(sinal_recebido) - 1) * Ts];
5
6 % Demodulando o sinal em fase e quadratura:
7 info_real_rx = sinal_recebido .* cos(2 * pi * fc * t_rx);
8 info_imag_rx = sinal_recebido .* (-sin(2 * pi * fc * t_rx));
9

```



```

10 figure;
11 subplot(221);
12 plot(t_rx(1:length(info_real_rx)), info_real_rx, 'LineWidth', 2, 'Color',
13      'k');
14 title('Componente Real - Demodulada');
15 xlabel('Tempo (s)');
16 ylabel('Amplitude');
17 xlim([0 10 * Tb]);
18 subplot(222);
19 plot(t_rx(1:length(info_imag_rx)), info_imag_rx, 'LineWidth', 2, 'Color',
20      'r');
21 title('Componente Imaginária - Demodulada');
22 xlabel('Tempo (s)');
23 ylabel('Amplitude');
24 xlim([0 10 * Tb]);

```

2.1.7. Filtrando o sinal demodulado:

Com o sinal já demodulado, foi filtrado utilizando um filtro passa-baixa para recuperar o sinal original, e o sinal foi filtrado utilizando a função `filter` do octave com base nos parâmetros definidos anteriormente.

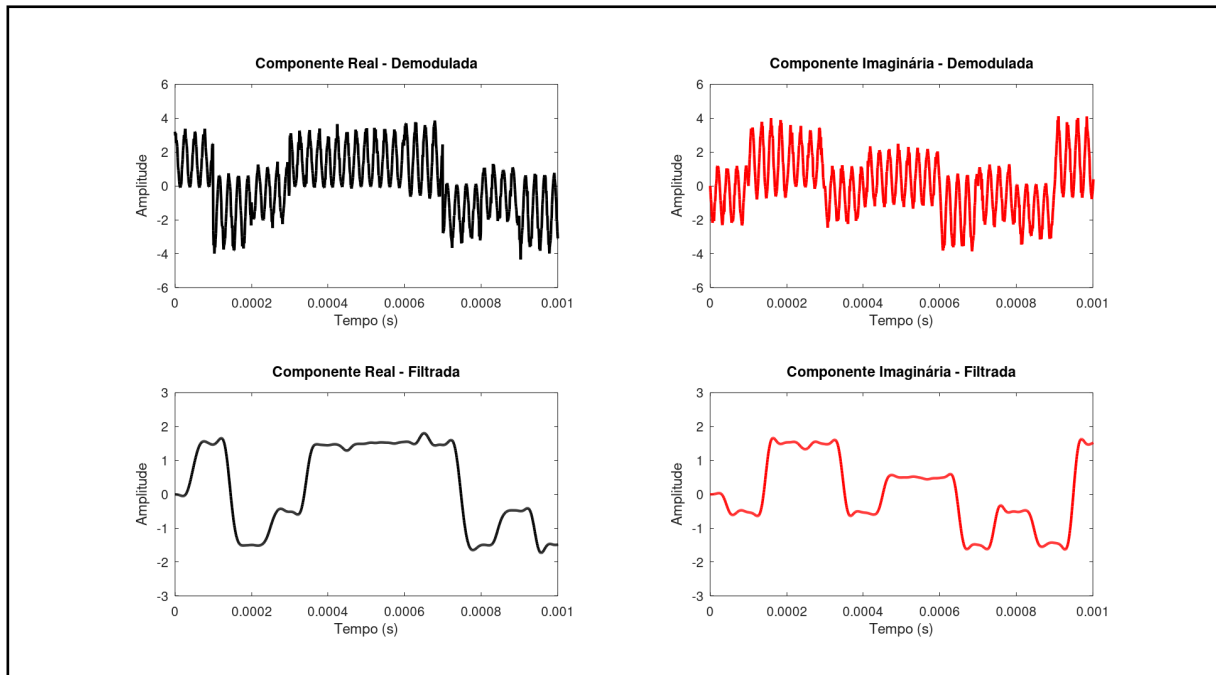
```

1 % Filtrando o sinal demodulado:
2
3 % Filtrando o sinal recebido em fase e quadratura:
4 info_real_rx_filtered = filter(filtro_passa_baixa, 1, info_real_rx);
5 info_imag_rx_filtered = filter(filtro_passa_baixa, 1, info_imag_rx);
6
7 subplot(223);
8 plot(t_rx(1:length(info_real_rx_filtered)), info_real_rx_filtered,
9      'LineWidth', 2, 'Color', 'k');
10 title('Componente Real - Filtrada');
11 xlabel('Tempo (s)');
12 ylabel('Amplitude');
13 xlim([0 10 * Tb]);
14 subplot(224);
15 plot(t_rx(1:length(info_imag_rx_filtered)), info_imag_rx_filtered,
16      'LineWidth', 2, 'Color', 'r');
17 title('Componente Imaginária - Filtrada');
18 xlabel('Tempo (s)');
19 ylabel('Amplitude');
20 xlim([0 10 * Tb]);

```

Desta forma, foi possível visualizar as componentes do sinal demoduladas, onde é possível observar a diferença entre o sinal transmitido e o sinal recebido, e a presença do ruído no sinal recebido:

Figure 4: Elaborada pelo Autor



Componentes do sinal Demoduladas e Filtradas

2.1.8. Realizando o downsampling do sinal:

Uma vez com o sinal filtrado, podemos realizar o downsampling do sinal para retornar a taxa de amostragem original, onde o sinal foi decimado para a taxa de amostragem original, e o excesso de amostras foi removido.

```

1  % Realizando o downsampling do sinal:
2
3  % Remover o excesso de amostras:
4  info_real_rx_down = downsample(info_real_rx_filtered, n);
5  info_imag_rx_down = downsample(info_imag_rx_filtered, n);
6
7  info_real_rx_down = info_real_rx_down(ceil(n/2):end);
8  info_imag_rx_down = info_imag_rx_down(ceil(n/2):end);
9
10 % Reconstruindo o sinal QAM transmitido:
11 info_rx = info_real_rx_down + 1i * info_imag_rx_down;
12
13 figure;
14 subplot(211);
15 plot(t(1:length(info_real_tx)), info_real_tx, 'LineWidth', 2, 'Color',
16      'b');
17 hold on;
18 plot(t_rx(1:length(info_real_rx_filtered)), info_real_rx_filtered,
19      'LineWidth', 2);
20 title('Componente Real do Sinal Recebido');
21 xlabel('Tempo (s)');
22 ylabel('Amplitude');
23 legend('Transmitida', 'Recebida (Após Filtragem)');
24 xlim([0 10 * Tb]);
25 ylim([-5 5]);

```

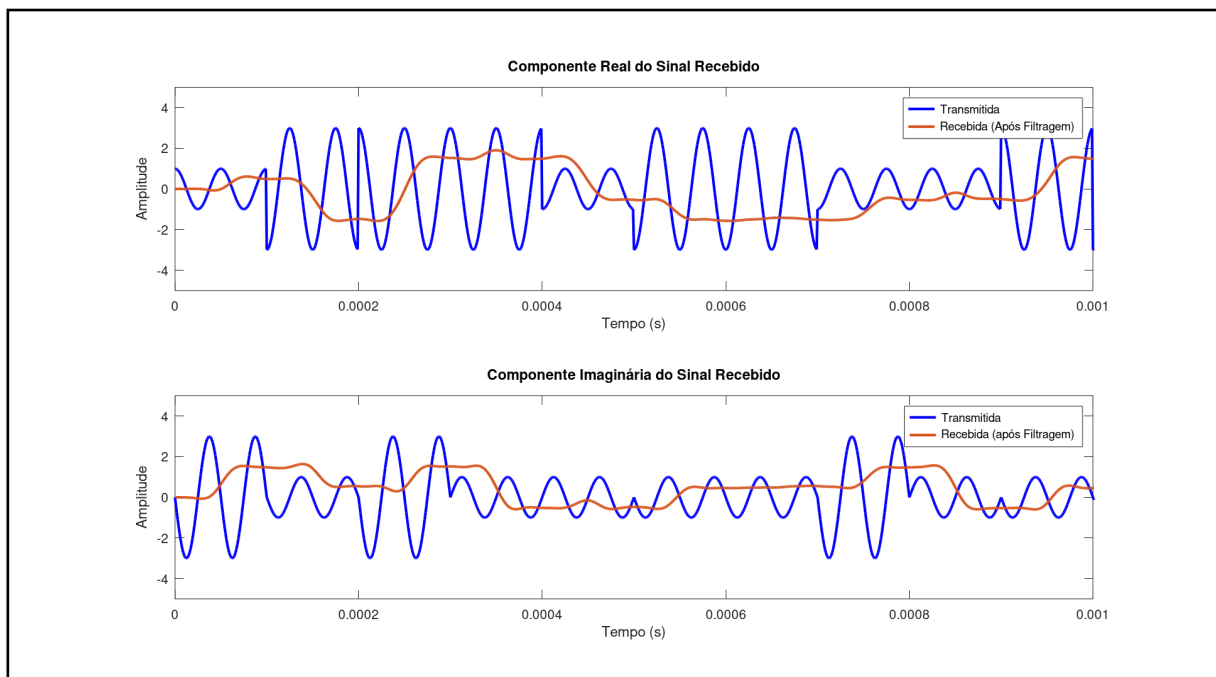
```

24
25 subplot(212);
26 plot(t(1:length(info_imag_tx)), info_imag_tx, 'LineWidth', 2, 'Color',
    'b');
27 hold on;
28 plot(t_rx(1:length(info_imag_rx_filtered)), info_imag_rx_filtered,
    'LineWidth', 2);
29 title('Componente Imaginária do Sinal Recebido');
30 xlabel('Tempo (s)');
31 ylabel('Amplitude');
32 legend('Transmitida', 'Recebida (após Filtragem)');
33 xlim([0 10 * Tb]);
34 ylim([-5 5]);

```

Desta forma, foi possível visualizar a comparação entre as componentes do sinal transmitido e do sinal recebido, onde é possível observar a diferença entre o sinal transmitido e o sinal recebido, e a presença do ruído no sinal recebido:

Figure 5: Elaborada pelo Autor



Comparando sinal de TX com sinal de RX

2.1.9. Plotando o sinal QAM Transmitido e Recebido:

Com os sinais antes da transmissão e após a recepção, é possível reconstruir o sinal QAM transmitido e verificar o sinal QAM recebido.

Assim, podemos plotar os diagramas de constelação dos sinais transmitidos e recebidos e realizar um comparativo entre ambos.

```

1 scatterplot(info_mod);
2 title('Diagrama de Constelação - Sinal TX');

```

```

3 xlim([-5 5]);
4 ylim([-5 5]);
5
6 scatterplot(info_rx);
7 title('Diagrama de Constelação - Sinal RX');
8 xlim([-5 5]);
9 ylim([-5 5]);

```

Abaixo está o diagrama 16-QAM antes da transmissão, note que os pontos estão bem definidos e separados, o que indica que o sinal está bem modulado.

Figure 6: Elaborada pelo Autor

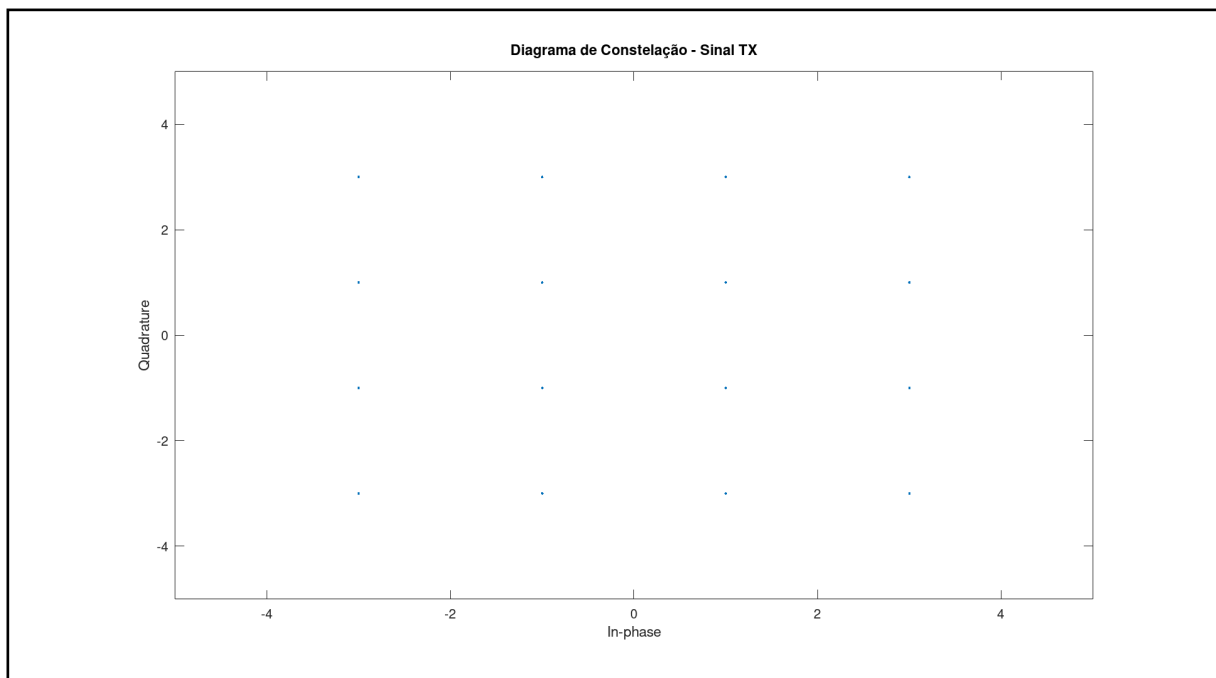


Diagrama de constelação QAM do sinal Transmitido

Abaixo está o diagrama 16-QAM após a recepção, note que os pontos estão mais próximos e dispersos, o que indica que o sinal foi afetado pelo ruído e interferências, e a qualidade do sinal foi reduzida.

Figure 7: Elaborada pelo Autor

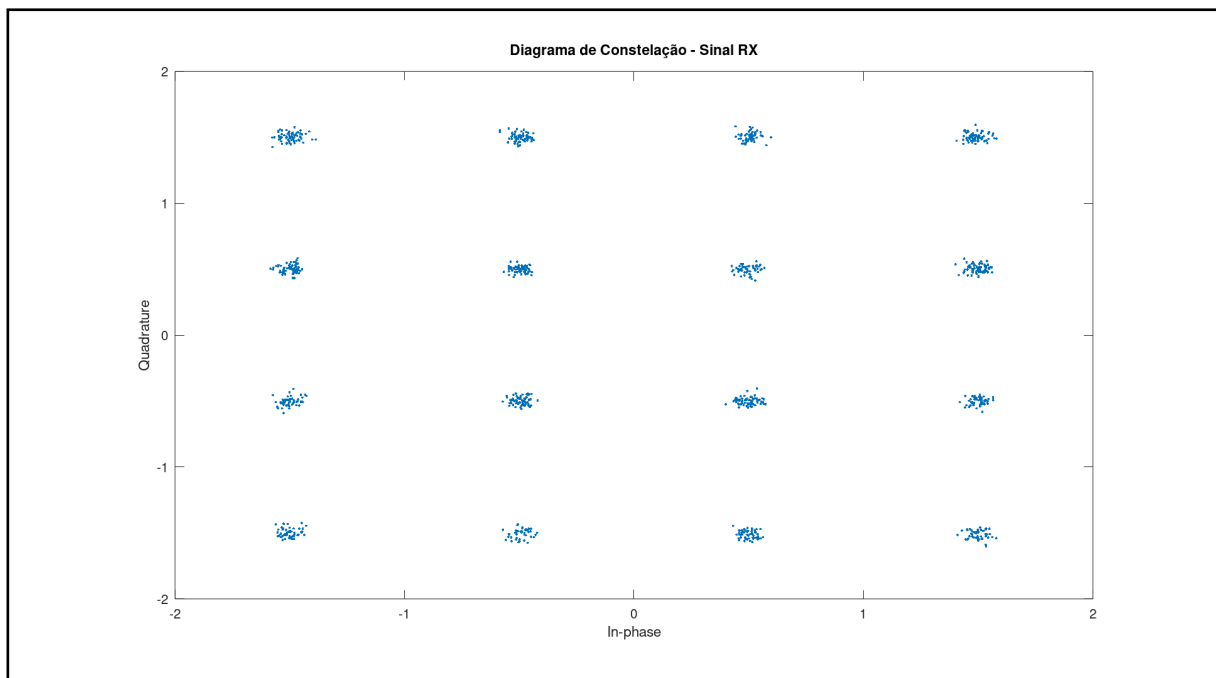


Diagrama de constelação QAM do sinal Recebido

2.2. Parte 2:

2.2.1. Definindo parâmetros de execução:

```
1  clc; close all; clear all;
2  pkg load communications;
3
4  % Configuração de parâmetros
5  % Definindo o n° de símbolos QAM
6  M = 16;
7
8  % Definindo o fator de upsampling
9  n = 100;
10
11 % Definindo a taxa de bits de TX
12 Rb = 1e4;
13
14 % Definindo o período de bit
15 Tb = 1 / Rb;
16
17 % Definindo a frequência de amostragem
18 Fs = Rb * n;
19
20 % Definindo a Frequência de portadora
21 fc = Fs / 50;
22
23 % Definindo o Período de amostragem:
24 Ts = 1 / Fs;
25
26 % Definindo o SNR do sinal de transmissão:
27 SNR = 12;
28
29 % Criando o vetor de dados:
```

```

30 Vector_length = 1000;
31 info = randi([0 M-1], 1, Vector_length);

```

2.2.2. Modulando o sinal QAM:

```

1 % Modulação QAM:
2
3 info_mod = qammod(info, M);
4
5 % Modulando o sinal em QAM:
6 scatterplot(info_mod);
7 title('Diagrama de constelação QAM do sinal');
8 xlim([-5 5]);
9 ylim([-5 5]);
10 grid on;
11
12 % Criando o vetor de tempo com base no comprimento da informação:
13 t = [0:Ts:(length(info_mod) * Tb - Ts)];

```

Figure 8: Elaborada pelo Autor

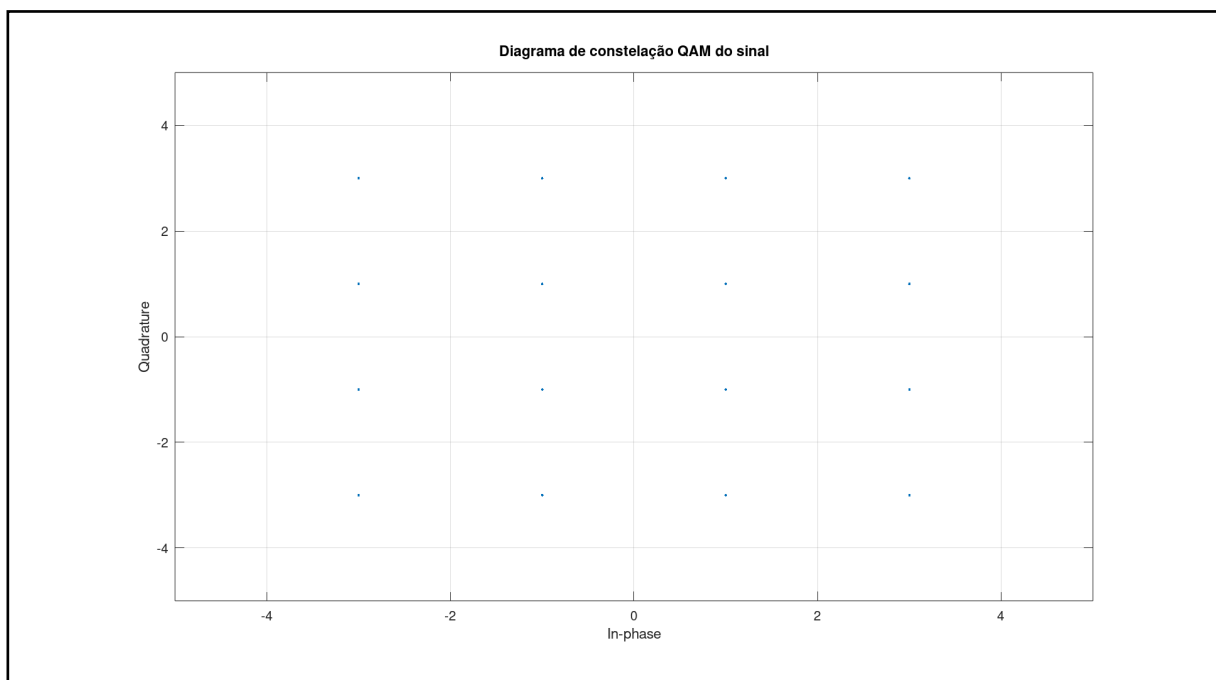


Diagrama de constelação QAM do sinal Recebido

2.2.3. Realizando o Upsampling do sinal:

```

1 % Upsample do sinal:
2
3 % Upsampling do sinal modulado
4 info_mod_up = upsample(info_mod, n); % Upsampling
5 filtro_NRZ = ones(1, n); % Filtro NRZ
6 info_mod_tx = filter(filtro_NRZ, 1, info_mod_up); % Filtragem

```

2.2.4. Modulando o sinal para transmissão:

```

1 % Modulando para transmissão:
2
3 % Modulação usando a representação complexa
4 portadora = exp(1j * 2 * pi * fc * t(1:length(info_mod_tx)));
5 sinal_transmitido = real(info_mod_tx .* portadora);
6
7 % Plotando o sinal transmitido
8 figure;
9 subplot(211);
10 plot(t(1:length(sinal_transmitido)), sinal_transmitido, 'LineWidth', 2);
11 title('Sinal Transmitido');
12 xlabel('Tempo (s)');
13 ylabel('Amplitude');
14 xlim([0 10 * Tb]);
15 ylim([-5 5]);
16
17 % Adicionando ruído ao sinal transmitido:
18 sinal_recebido = awgn(sinal_transmitido, SNR);
19
20 % Plotando o sinal recebido com ruído
21 subplot(212);
22 plot(t(1:length(sinal_recebido)), sinal_recebido, 'LineWidth', 2);
23 title('Sinal Recebido com Ruído');
24 xlabel('Tempo (s)');
25 ylabel('Amplitude');
26 xlim([0 10 * Tb]);
27 ylim([-5 5]);

```

Figure 9: Elaborada pelo Autor

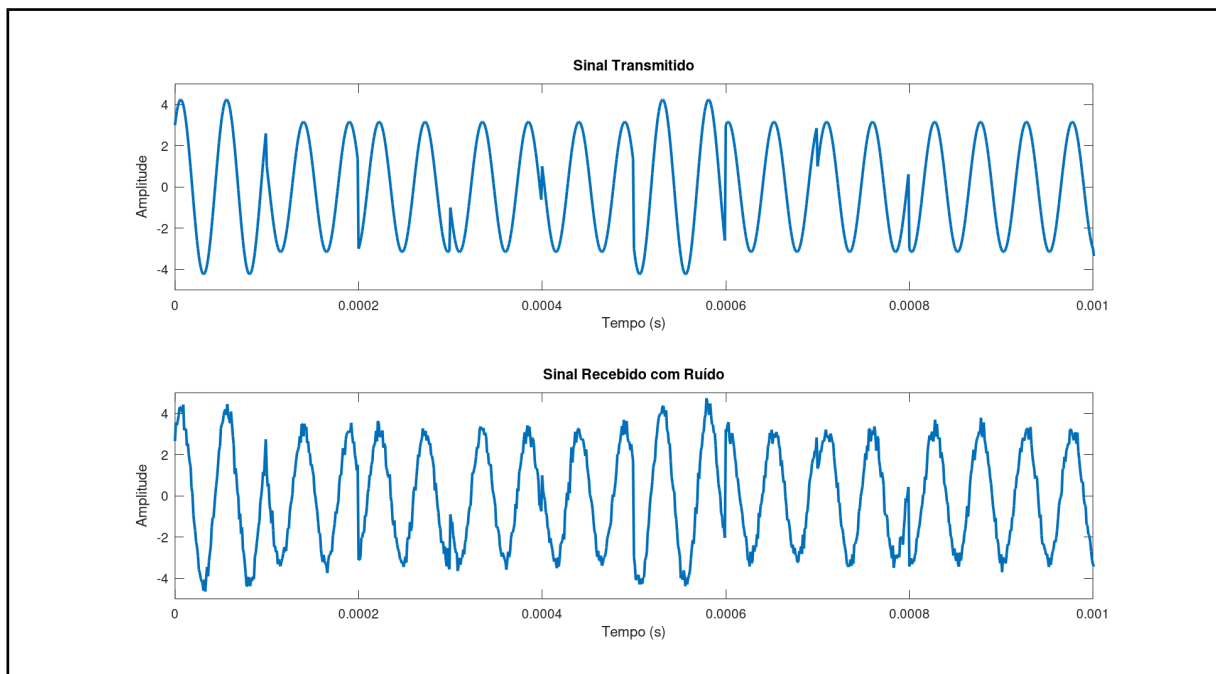


Diagrama de constelação QAM do sinal Recebido

2.2.5. Demodulando o sinal recebido:

```

1 % Demodulação do sinal de recepção:

```

```

2
3 % Demodulação usando a representação complexa
4 portadora_rx = exp(-1j * (2 * pi * fc * t(1:length(sinal_recebido))));
5 sinal_demodulado = sinal_recebido .* portadora_rx;

```

2.2.6. Filtrando o sinal demodulado:

```

1 % Filtrando o sinal demodulado:
2
3 % Filtragem passa-baixa para recuperar o sinal original
4 filtro_passa_baixa = fir1(100, fc/(Fs/2));
5 info_rx_filtered = filter(filtro_passa_baixa, 1, sinal_demodulado);

```

2.2.7. Realizando o downsampling do sinal:

```

1 % Realizando o downsampling do sinal:
2
3 % Downsampling para retornar à taxa de amostragem original
4 info_rx_down = downsample(info_rx_filtered, n);
5
6 % Remover o excesso de amostras devido ao filtro
7 info_rx_down = info_rx_down(ceil(n/2):end);
8
9 % Plotando as componentes real e imaginária do sinal recuperado
10 figure;
11 subplot(211);
12 plot(real(info_rx_down), 'LineWidth', 2, 'Color', 'k');
13 title('Componente Real - Sinal Recebido');
14 xlabel('Amostras');
15 ylabel('Amplitude');
16 grid on;
17
18 subplot(212);
19 plot(imag(info_rx_down), 'LineWidth', 2, 'Color', 'r');
20 title('Componente Imaginária - Sinal Recebido');
21 xlabel('Amostras');
22 ylabel('Amplitude');
23 grid on;

```


Figure 10: Elaborada pelo Autor

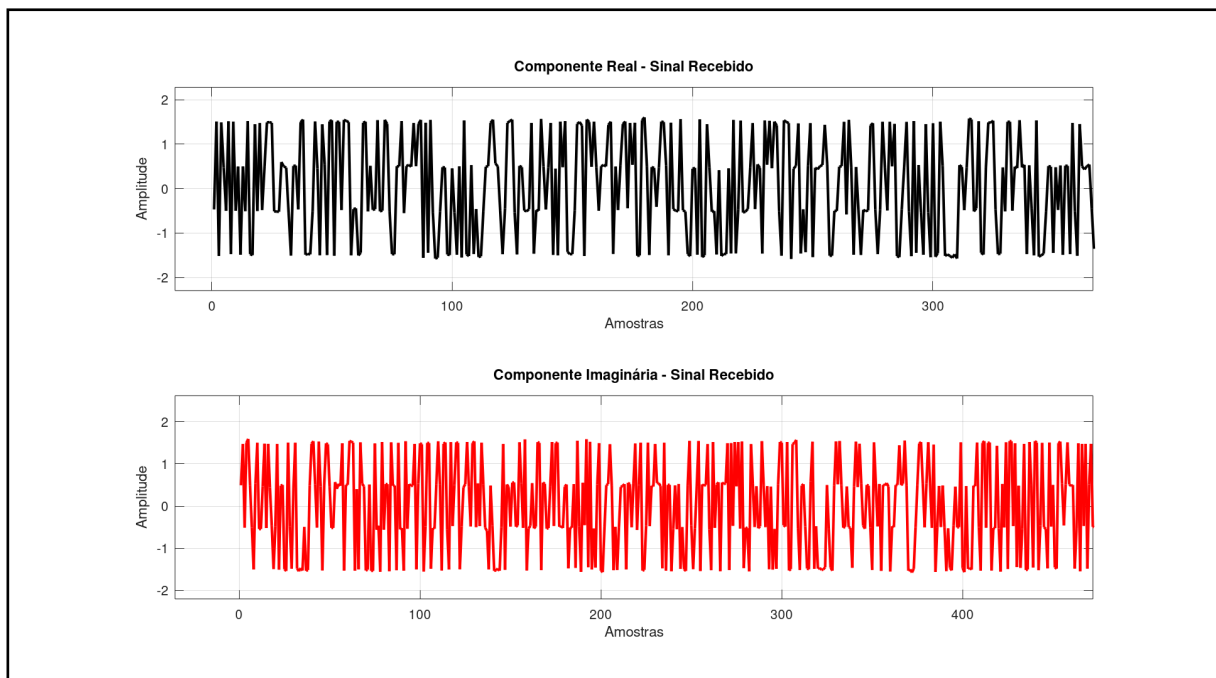


Diagrama de constelação QAM do sinal Recebido

2.2.8. Reconstruindo o sinal QAM Transmitido:

```
1 % Reconstruindo o sinal QAM Transmitido:
2
3 % Reconstrução do sinal QAM
4 info_rx = real(info_rx_down) + 1i * imag(info_rx_down);
5
6 % Plotando os diagramas de constelação
7 scatterplot(info_mod);
8 xlim([-5 5]);
9 ylim([-5 5]);
10 title('Diagrama de Constelação do Sinal Transmitido');
11 grid on;
12
13 scatterplot(info_rx);
14 xlim([-5 5]);
15 ylim([-5 5]);
16 title('Diagrama de Constelação do Sinal Recebido');
17 grid on;
```

Figure 11: Elaborada pelo Autor

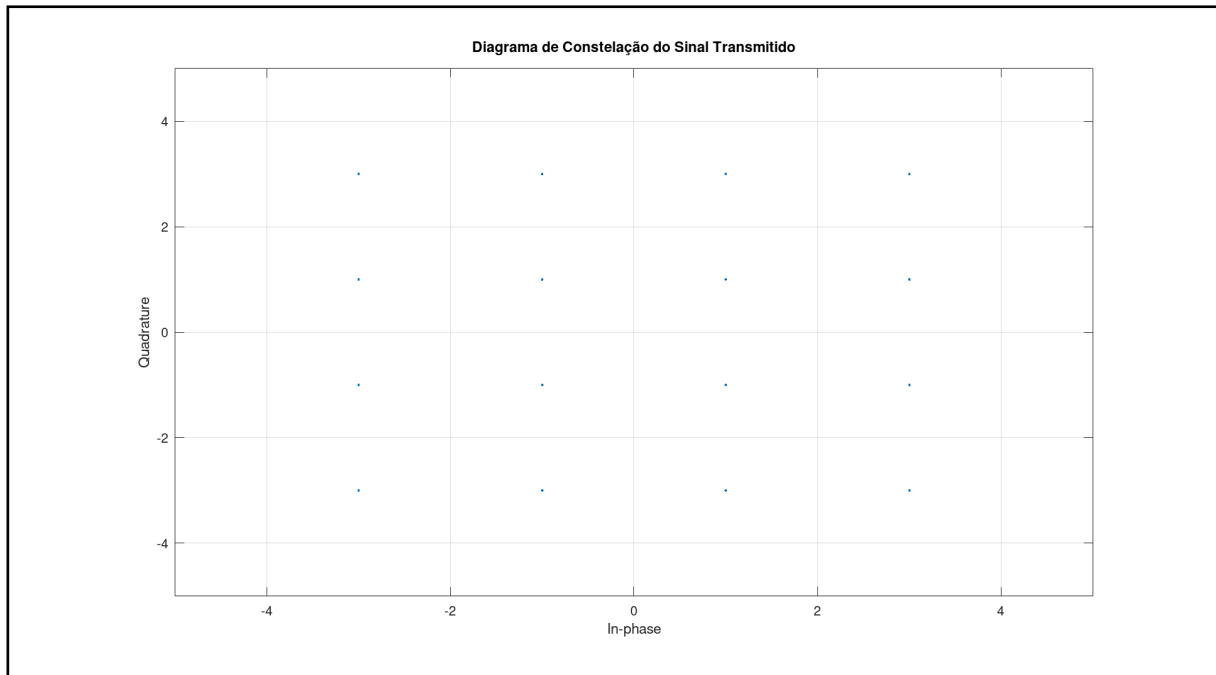


Diagrama de constelação QAM do sinal Recebido

Figure 12: Elaborada pelo Autor

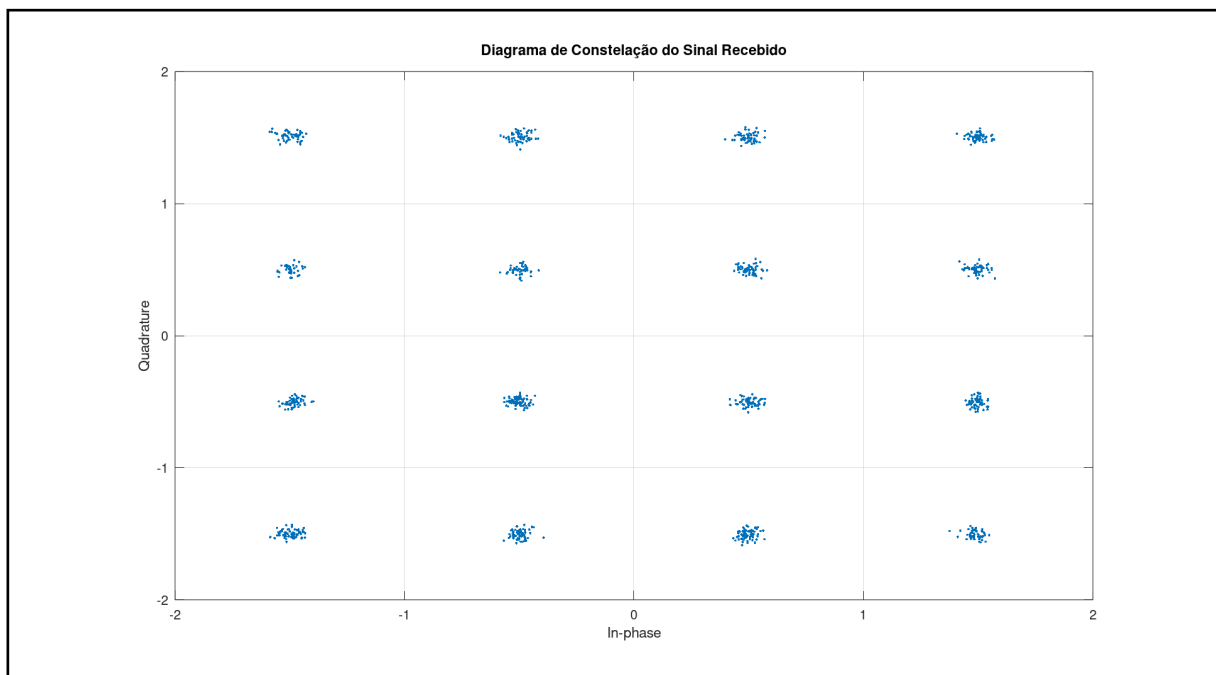


Diagrama de constelação QAM do sinal Recebido

2.2.9. Comparação das componentes real e imaginária:

```
1 figure;  
2 subplot(211);  
3 plot(t(1:length(info_mod_tx)), real(info_mod_tx), 'LineWidth', 2, 'k');  
4 hold on;
```

```

5 plot(t(1:length(info_rx_filtered)), real(info_rx_filtered), 'LineWidth',
6 2, 'b');
7 title('Comparação da Componente Real');
8 xlabel('Tempo (s)');
9 ylabel('Amplitude');
10 legend('Transmitida', 'Recebida');
11 xlim([0 10 * Tb]);
12 ylim([-5 5]);
13 grid on;
14 subplot(212);
15 plot(t(1:length(info_mod_tx)), imag(info_mod_tx), 'LineWidth', 2, 'r');
16 hold on;
17 plot(t(1:length(info_rx_filtered)), imag(info_rx_filtered), 'LineWidth',
18 2, 'b');
19 title('Comparação da Componente Imaginária');
20 xlabel('Tempo (s)');
21 ylabel('Amplitude');
22 legend('Transmitida', 'Recebida');
23 xlim([0 10 * Tb]);
24 ylim([-5 5]);
25 grid on;

```

3. Conclusão:

4. Referências Bibliográficas: