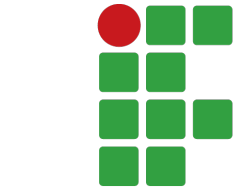


INSTITUTO FEDERAL
Santa Catarina

Câmpus
São José

# Modulador 16-QAM

Sistemas de Comunicação I

**Arthur Cadore Matuella Barcella**

29 de Julho de 2024

# Sumário

# 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:

```
%% Inicializando pacotes necessários:
clc; close all; clear all;
pkg load communications;

% Definindo o n° de símbolos QAM
M = 16;

% Definindo o fator de upsampling
n = 100;

% Definindo a taxa de bits de TX
Rb = 1e4;

% Definindo o período de bit
Tb = 1 / Rb;

% Definindo á frequência de amostragem
Fs = Rb * n;

% Definindo á Frequência de portadora
fc = Fs / 50;

% Definindo o Período de amostragem:
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 realizra 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

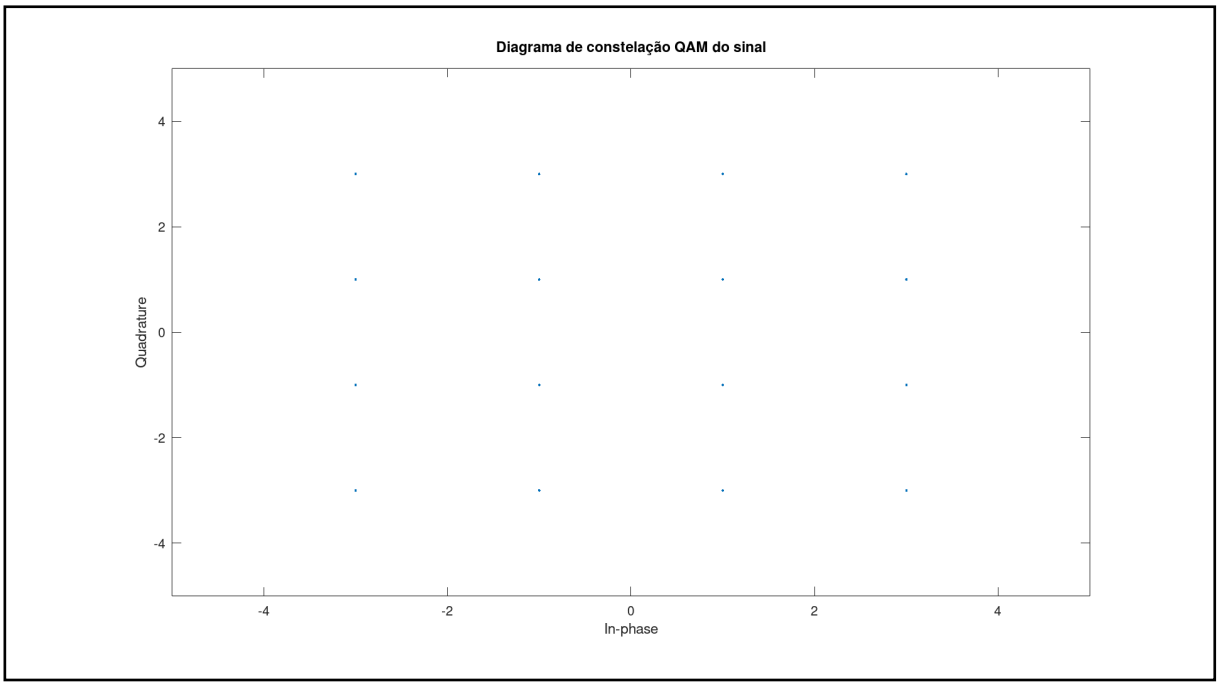



Diagrama de constelação QAM do sinal

### 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, oque 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`.

```
% Criando um filtro NRZ para realizar o upsample do sinal:
filtro_NRZ = ones(1, n);
info_r_real_up = upsample(info_r_real, n); % Upsampling
info_r_real_tx = filter(filtro_NRZ, 1, info_r_real_up); % Filtragem

% Realizando o plot no dominio do tempo:
figure;
subplot(221);
plot(t(1:length(info_r_real_tx)), info_r_real_tx, 'LineWidth', 2, 'Color',
'k');
title('Sinal de Informação (Componente Real)');
xlabel('Tempo (s)');
ylabel('Amplitude');
xlim([0 10 * Tb]);
ylim([-5 5]);

info_i_imag_up = upsample(info_i_imag, n); % Upsampling
info_i_imag_tx = filter(filtro_NRZ, 1, info_i_imag_up); % Filtragem

subplot(222);
```

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

### 2.1.4. Modulando o sinal para transmissão:

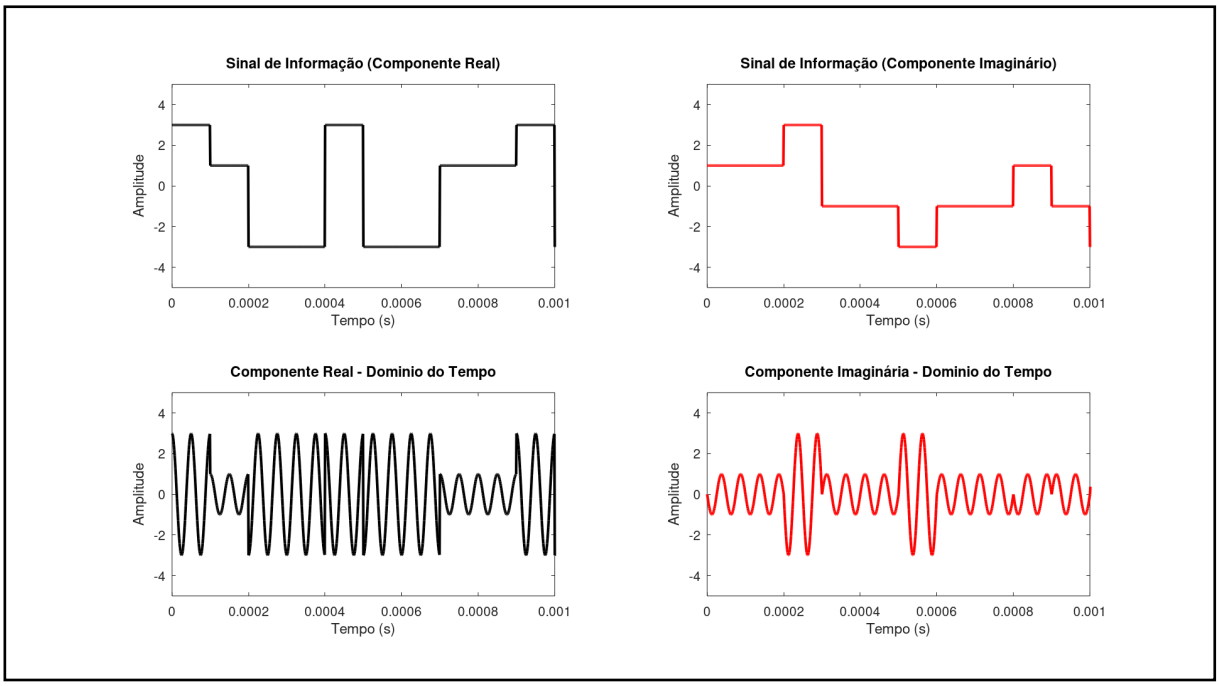
Em seguida, o sinal foi modulado para a transmissão, onde foi criado um sinal portadora cosseno e um sinal portadora seno, e o sinal de informação foi multiplicado por essas portadoras para realizar a modulação do sinal.

```
% Modulando para transmissão:

% Criando portadora Cosseno:
cos_carrier = cos(2 * pi * fc * t(1:length(info_r_real_tx)));
info_real_tx = info_r_real_tx .* cos_carrier;

% Criando portadora Seno:
sen_carrier = -sin(2 * pi * fc * t(1:length(info_i_imag_tx)));
info_imag_tx = info_i_imag_tx .* sen_carrier;

subplot(223);
plot(t(1:length(info_real_tx)), info_real_tx, 'LineWidth', 2, 'Color', 'k');
title('Componente Real - Dominio do Tempo');
xlabel('Tempo (s)');
ylabel('Amplitude');
xlim([0 10 * Tb]);
ylim([-5 5]);

subplot(224);
plot(t(1:length(info_imag_tx)), info_imag_tx, 'LineWidth', 2, 'Color', 'r');
title('Componente Imaginária - Dominio do Tempo');
xlabel('Tempo (s)');
ylabel('Amplitude');
xlim([0 10 * Tb]);
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 realizado a soma das componentes do sinal para obter o sinal de transmissão (Fase e Quadratura), oque 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, oque a torna o mais próximo possível de um piso de ruído térmico.

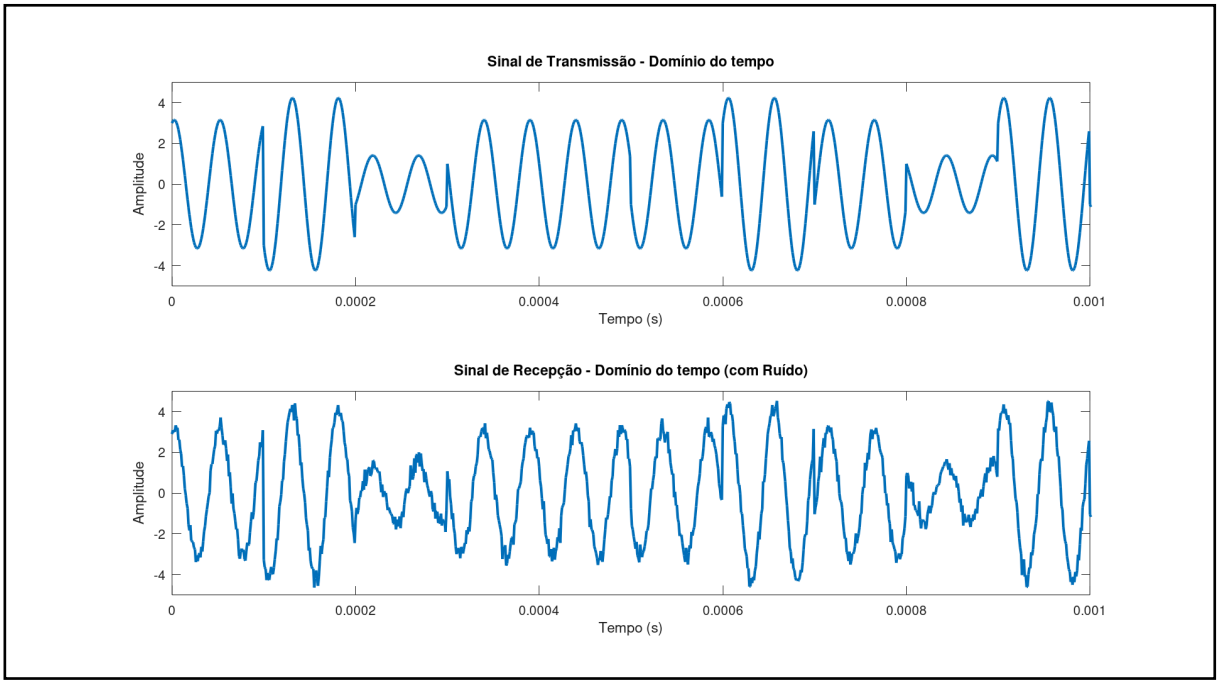
O objetivo de adicionar ruído ao sinal transmitido é simular o ambiente de transmissão real, onde o sinal é afetado por ruídos e interferências, e assim, é possível avaliar a qualidade do sinal recebido.

```
% Criando o sinal de transmissão:

sinal_tx = info_real_tx + info_imag_tx;

figure;
subplot(211);
plot(t(1:length(sinal_tx)), sinal_tx, 'LineWidth', 2);
title('Sinal de Transmissão - Domínio do tempo');
xlabel('Tempo (s)');
ylabel('Amplitude');
xlim([0 10 * Tb]);
ylim([-5 5]);

% Adicionando ruído ao sinal transmitido
```

```
sinal_recebido = awgn(sinal_tx, SNR);

subplot(212);
plot(t(1:length(sinal_recebido)), sinal_recebido, 'LineWidth', 2);
title('Sinal de Recepção - Domínio do tempo (com Ruído)');
xlabel('Tempo (s)');
ylabel('Amplitude');
xlim([0 10 * Tb]);
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.

```
% Demodulação do sinal de recepção:

% Ajuste do vetor de tempo:
t_rx = [0:Ts:(length(sinal_recebido) - 1) * Ts];

% Demodulando o sinal em fase e quadratura:
info_real_rx = sinal_recebido .* cos(2 * pi * fc * t_rx);
info_imag_rx = sinal_recebido .* (-sin(2 * pi * fc * t_rx));

```

```
figure;
subplot(221);
plot(t_rx(1:length(info_real_rx)), info_real_rx, 'LineWidth', 2, 'Color', 'k');
title('Componente Real - Demodulada');
xlabel('Tempo (s)');
ylabel('Amplitude');
xlim([0 10 * Tb]);

subplot(222);
plot(t_rx(1:length(info_imag_rx)), info_imag_rx, 'LineWidth', 2, 'Color', 'r');
title('Componente Imaginária - Demodulada');
xlabel('Tempo (s)');
ylabel('Amplitude');
xlim([0 10 * Tb]);
```

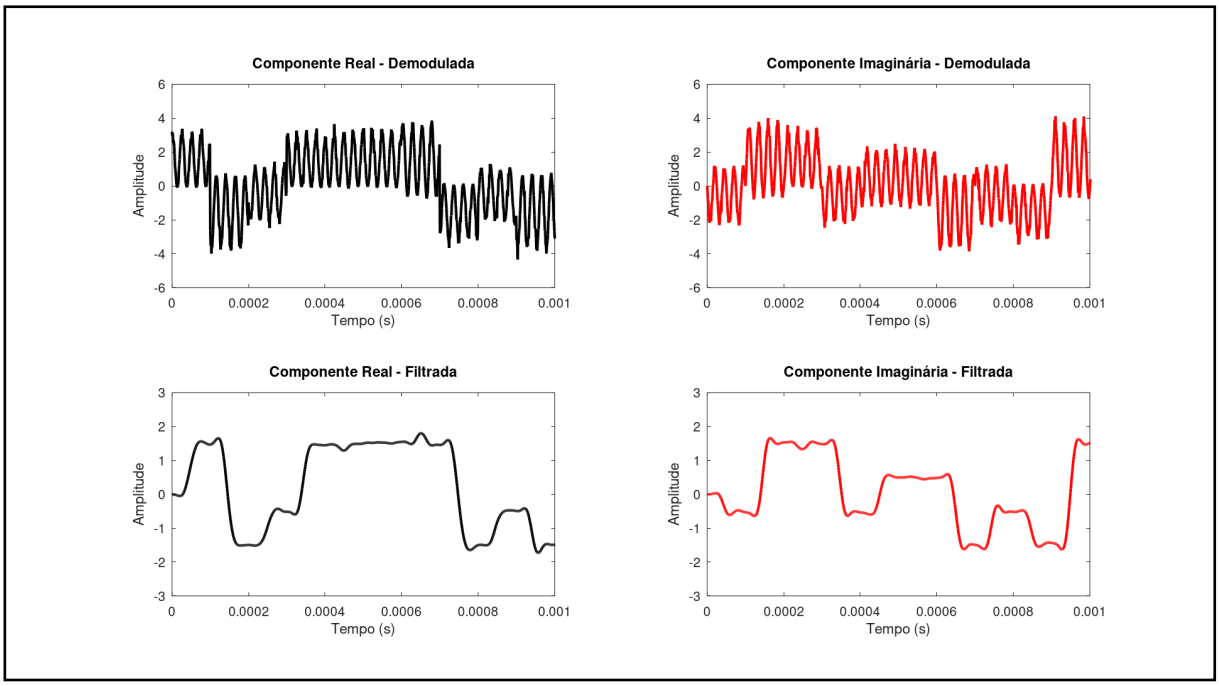
### 2.1.7. Filtrando o sinal demodulado:

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.

```
% Filtrando o sinal demodulado:

% Filtrando o sinal recebido em fase e quadratura:
info_real_rx_filtered = filter(filtro_passa_baixa, 1, info_real_rx);
info_imag_rx_filtered = filter(filtro_passa_baixa, 1, info_imag_rx);

subplot(223);
plot(t_rx(1:length(info_real_rx_filtered)),         info_real_rx_filtered, 'LineWidth', 2, 'Color', 'k');
title('Componente Real - Filtrada');
xlabel('Tempo (s)');
ylabel('Amplitude');
xlim([0 10 * Tb]);

subplot(224);
plot(t_rx(1:length(info_imag_rx_filtered)),         info_imag_rx_filtered, 'LineWidth', 2, 'Color', 'r');
title('Componente Imaginária - Filtrada');
xlabel('Tempo (s)');
ylabel('Amplitude');
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:

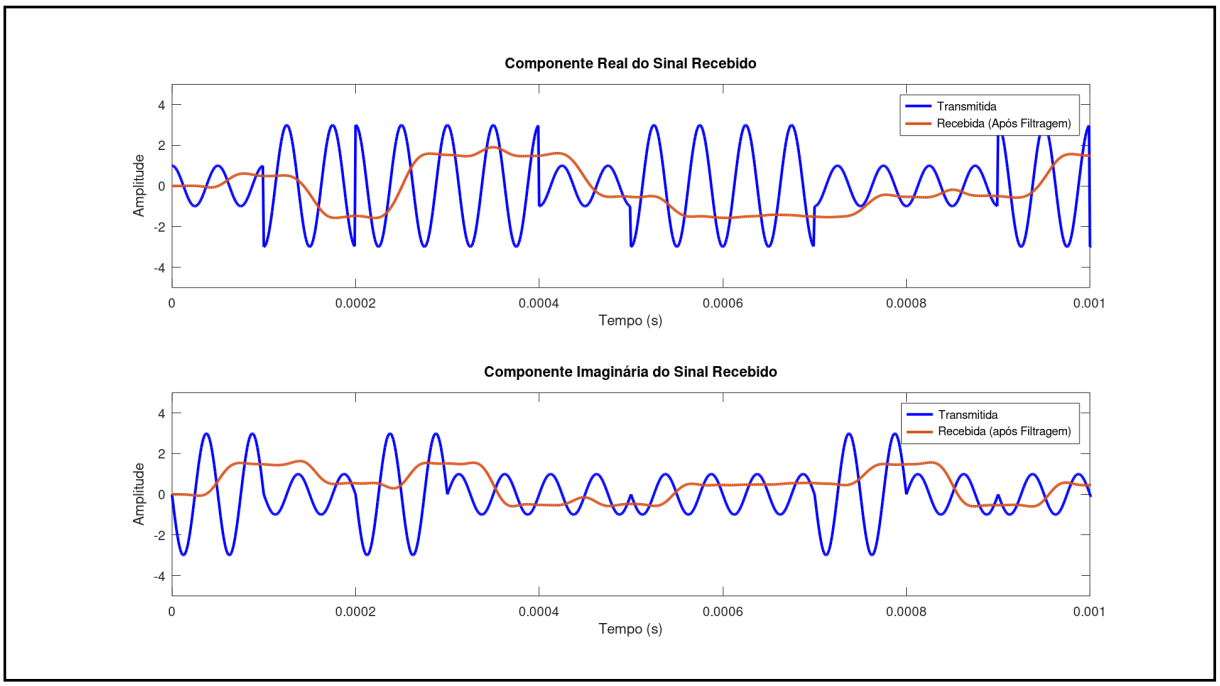
Uma vez com o sinal filtrado, podemos realizar o downsampling do sinal para retornar a taxa de amostragem original, onde o sinal foi decimado para a taxa de amostragem original, e o excesso de amostras foi removido.

```
% Realizando o downsampling do sinal:

% Remover o excesso de amostras:
info_real_rx_down = downsample(info_real_rx_filtered, n);
info_imag_rx_down = downsample(info_imag_rx_filtered, n);

info_real_rx_down = info_real_rx_down(ceil(n/2):end);
info_imag_rx_down = info_imag_rx_down(ceil(n/2):end);

% Reconstruindo o sinal QAM transmitido:
info_rx = info_real_rx_down + 1i * info_imag_rx_down;

figure;
subplot(211);
plot(t(1:length(info_real_tx)), info_real_tx, 'LineWidth', 2, 'Color',
'b');
hold on;
plot(t_rx(1:length(info_real_rx_filtered)), info_real_rx_filtered,
'LineWidth', 2);
title('Componente Real do Sinal Recebido');
xlabel('Tempo (s)');
ylabel('Amplitude');
legend('Transmitida', 'Recebida (Após Filtragem)');
xlim([0 10 * Tb]);
ylim([-5 5]);
```

```
24
25 subplot(212);
   plot(t(1:length(info_imag_tx)), info_imag_tx, 'LineWidth', 2, 'Color',
26 'b');
27 hold on;
   plot(t_rx(1:length(info_imag_rx_filtered)), info_imag_rx_filtered,
28 '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 possivel 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');
```

```
xlim([-5 5]);
ylim([-5 5]);

scatterplot(info_rx);
title('Diagrama de Constelação - Sinal RX');
xlim([-5 5]);
ylim([-5 5]);
```

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

Figure 6: Elaborada pelo Autor

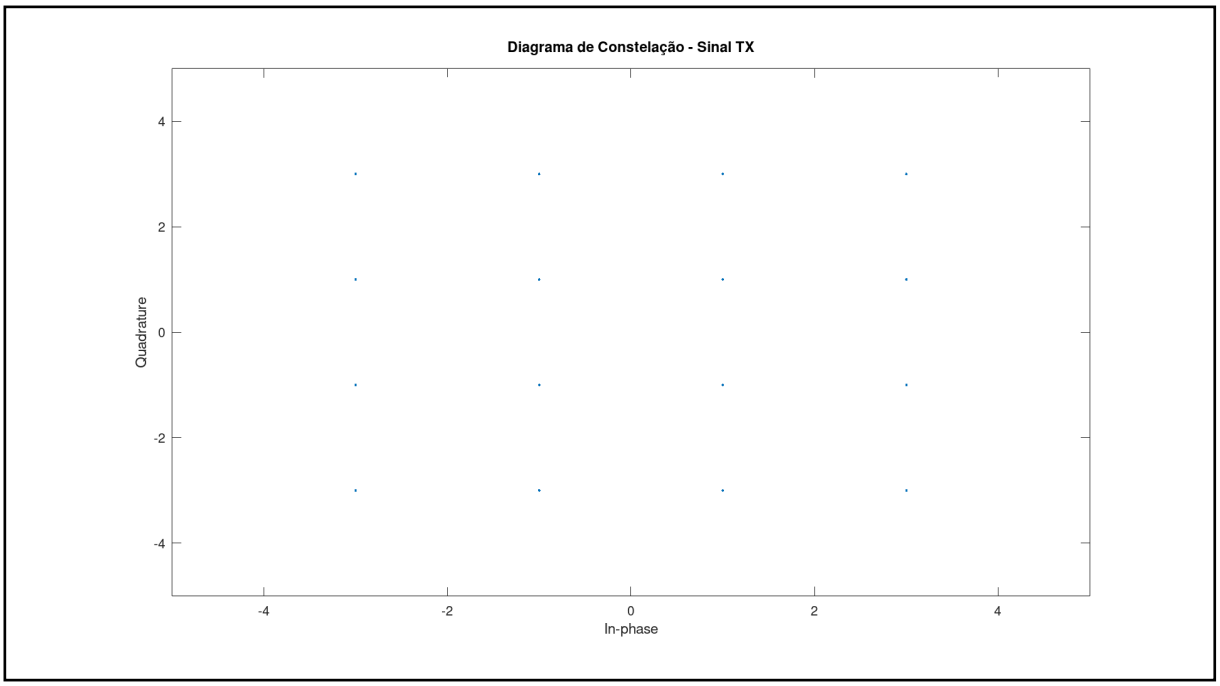



Diagrama de constelação QAM do sinal Transmitido

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

Figure 7: Elaborada pelo Autor

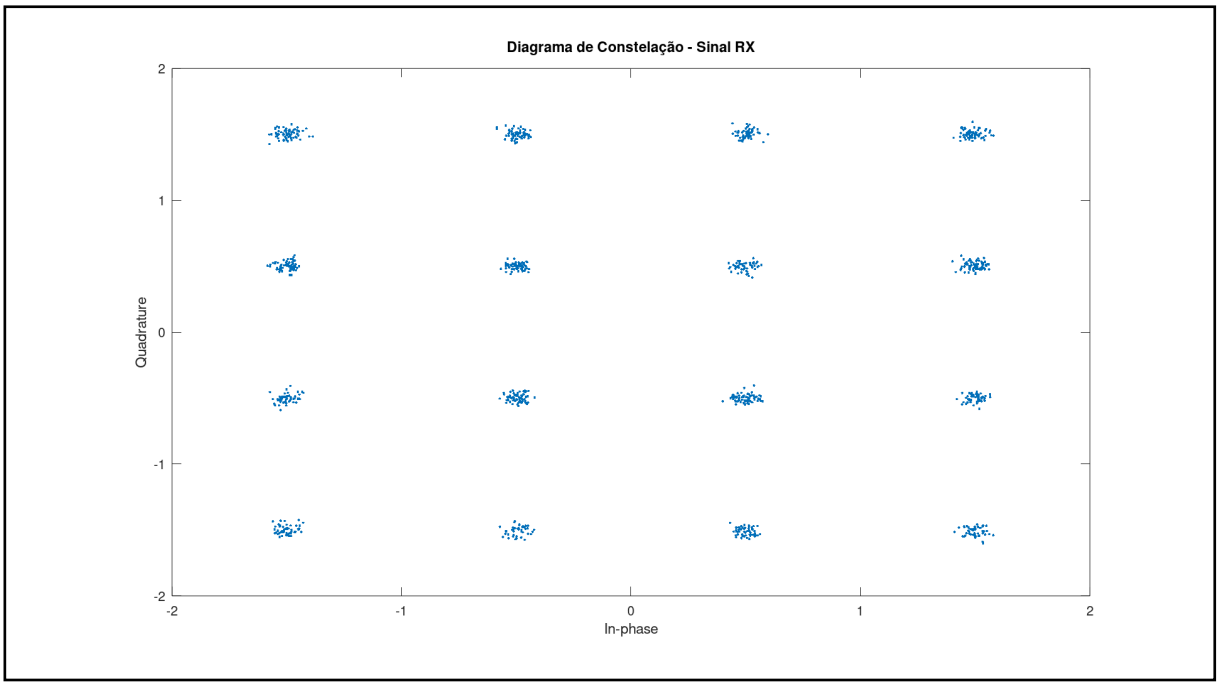



Diagrama de constelação QAM do sinal Recebido

## 2.2. Parte 2:

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

### 2.2.1. 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, foi definido os seguintes parâmetros:

```
clc; close all; clear all;
pkg load communications;

% Configuração de parâmetros
% Definindo o n° de símbolos QAM
M = 16;

% Definindo o fator de upsampling
n = 100;

% Definindo a taxa de bits de TX
Rb = 1e4;

% Definindo o período de bit
Tb = 1 / Rb;

% Definindo á frequência de amostragem
Fs = Rb * n;

```

```
% Definindo á Frequência de portadora
fc = Fs / 50;

% Definindo o Período de amostragem:
Ts = 1 / Fs;

% Definindo o SNR do sinal de transmissão:
SNR = 12;
```

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.

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

### 2.2.2. Modualando o sinal QAM:

Uma vez com os parâmetros definidos e o vetor de dados gerado, o primeiro passo foi realizra 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`.

```
% Modulação QAM:

info_mod = qammod(info, M);

% Modulando o sinal em QAM:
scatterplot(info_mod);
title('Diagrama de constelação QAM do sinal');
xlim([-5 5]);
ylim([-5 5]);
grid on;

% Criando o vetor de tempo com base no comprimento da informação:
t = [0:Ts:(length(info_mod) * 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 8: Elaborada pelo Autor

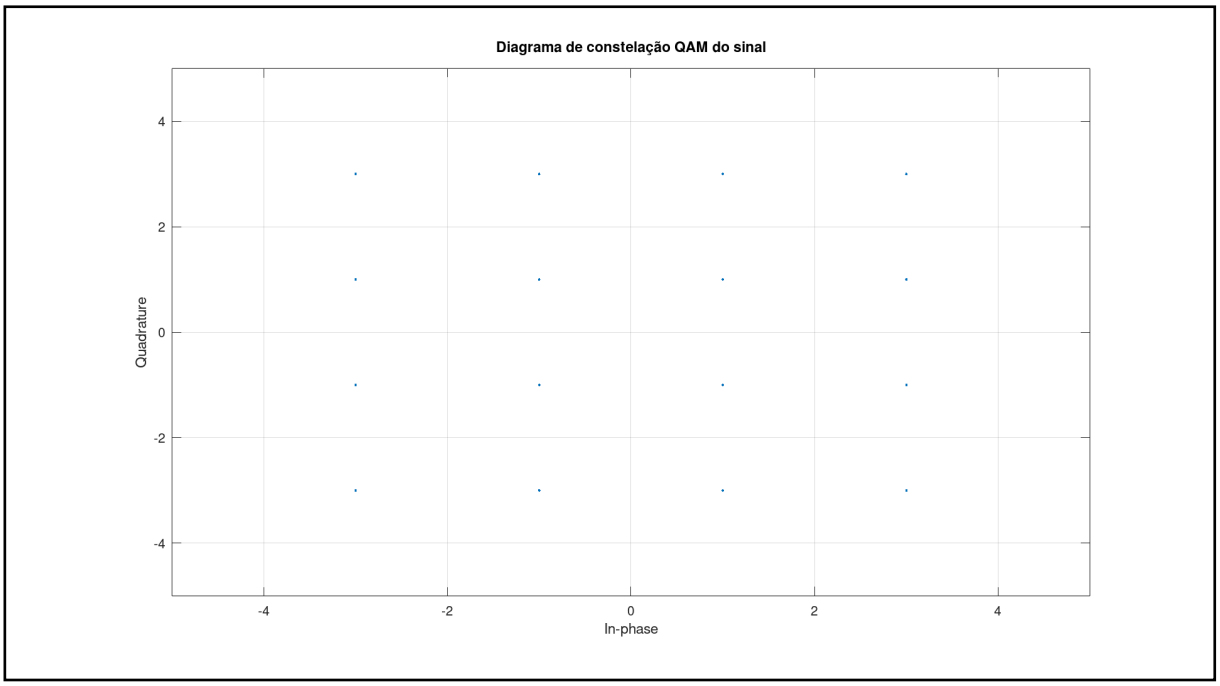



Diagrama de constelação QAM do sinal Recebido

### 2.2.3. Realizando o 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, oque melhora a qualidade do sinal e facilita a filtragem do sinal.

```
% Upsample do sinal:

% Upsampling do sinal modulado
info_mod_up = upsample(info_mod, n); % Upsampling
filtro_NRZ = ones(1, n); % Filtro NRZ
info_mod_tx = filter(filtro_NRZ, 1, info_mod_up); % Filtragem
```

### 2.2.4. Modulando o sinal para transmissão:

Na sequencia, com o sinal modulado e expandido, foi realizado a modulação do sinal para a transmissão, onde o sinal foi modulado utilizando a representação complexa, onde o sinal foi multiplicado pela portadora complexa.

```
% Modulando para transmissão:

% Modulação usando a representação complexa
portadora = exp(1j * 2 * pi * fc * t(1:length(info_mod_tx)));
sinal_transmitido = real(info_mod_tx .* portadora);

% Plotando o sinal transmitido
figure;
subplot(211);
plot(t(1:length(sinal_transmitido)), sinal_transmitido, 'LineWidth', 2);
```

```
title('Sinal Transmitido');
xlabel('Tempo (s)');
ylabel('Amplitude');
xlim([0 10 * Tb]);
ylim([-5 5]);

% Adicionando ruído ao sinal transmitido:
sinal_recebido = awgn(sinal_transmitido, SNR);

% Plotando o sinal recebido com ruído
subplot(212);
plot(t(1:length(sinal_recebido)), sinal_recebido, 'LineWidth', 2);
title('Sinal Recebido com Ruído');
xlabel('Tempo (s)');
ylabel('Amplitude');
xlim([0 10 * Tb]);
ylim([-5 5]);
```

Podemos ver no plot do sinal transmitido e do 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 9: Elaborada pelo Autor

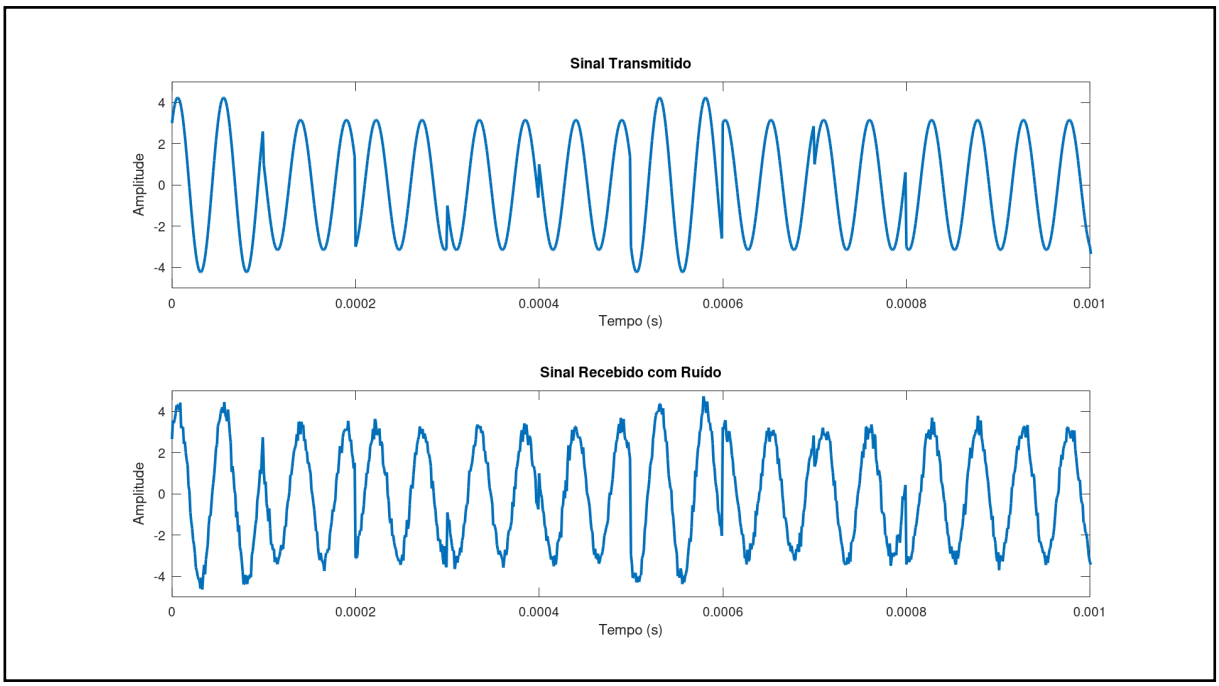



Diagrama de constelação QAM do sinal Recebido

### 2.2.5. Demodulando o sinal recebido:

Para realizar a demodulação do sinal recebido, foi utilizado a representação complexa, onde o sinal recebido foi multiplicado pela própria portadora complexa, retornando o sinal modulado para a banda base.

```
% 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:

Em seguida, o sinal 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 % 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:

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

```
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;
```

Com o sinal reduzido, podemos ver no dominio do tempo as componentes de fase e quadratura 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 10: Elaborada pelo Autor

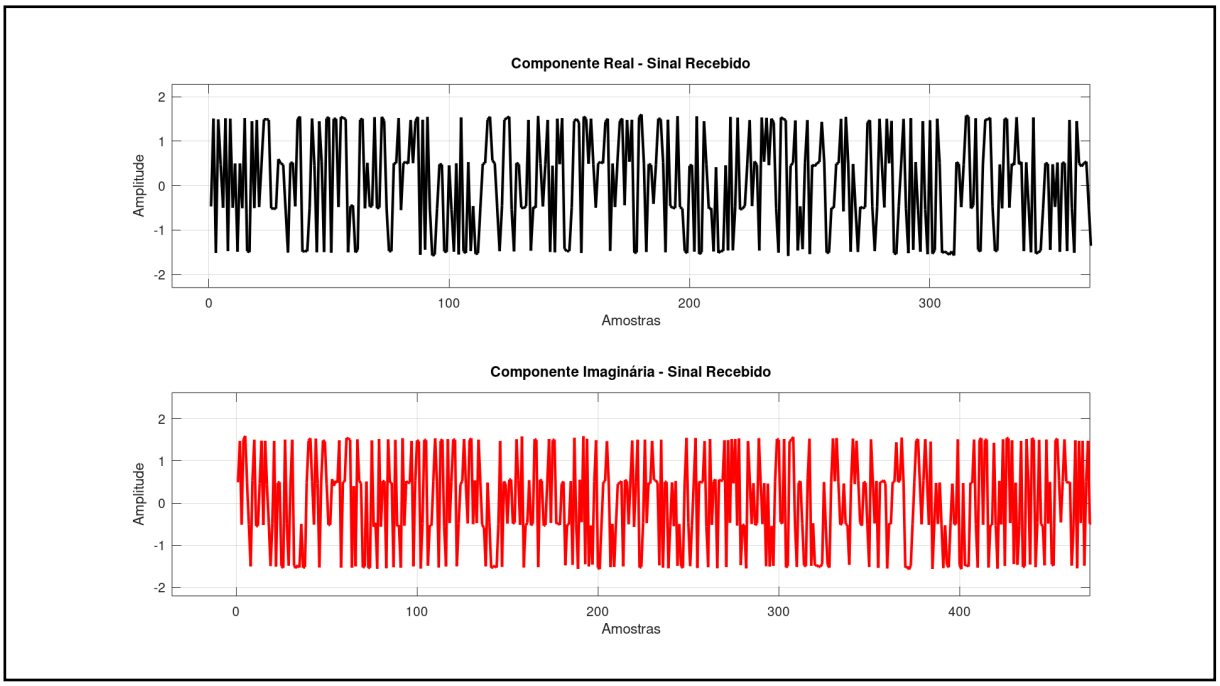



Diagrama de constelação QAM do sinal Recebido

### 2.2.8. Reconstruindo o sinal QAM Transmitido:

Com o sinal recebido já reconstruido, podemos realizar seu plot no diagrama de constelação, para verificar a diferença entre o sinal transmitido e o sinal recebido.

```
% Reconstruindo o sinal QAM Transmitido:

% Reconstrução do sinal QAM
info_rx = real(info_rx_down) + 1i * imag(info_rx_down);

% Plotando os diagramas de constelação
scatterplot(info_mod);
xlim([-5 5]);
ylim([-5 5]);
title('Diagrama de Constelação do Sinal Transmitido');
grid on;

scatterplot(info_rx);
xlim([-5 5]);
ylim([-5 5]);
title('Diagrama de Constelação do Sinal Recebido');
grid on;
```

Na figura abaixo, é possivel visualizar o sinal antes de ser transmitido, onde os pontos estão bem definidos e separados, oque indica que o sinal está bem modulado:

Figure 11: Elaborada pelo Autor

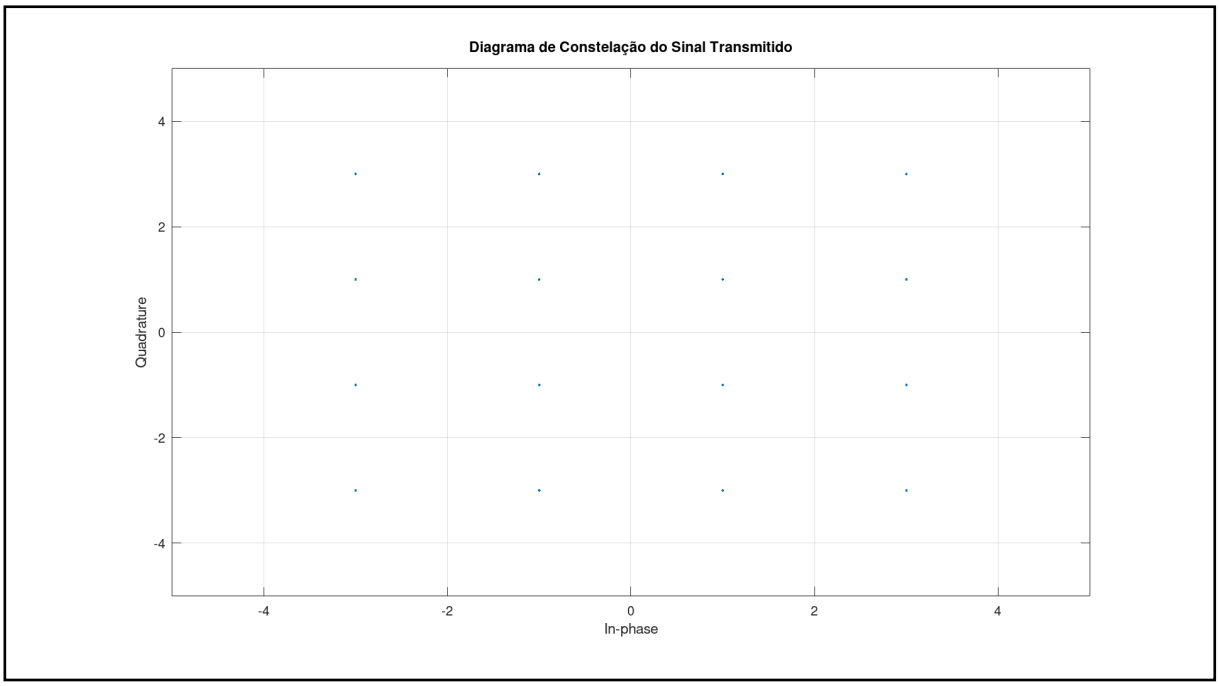



Diagrama de constelação QAM do sinal Recebido

Já na figura abaixo, podemos ver o sinal após a recepção, onde os pontos estão mais próximos e dispersos, oque indica que o sinal foi afetado pelo ruído e interferências, e a qualidade do sinal foi reduzida:

Figure 12: Elaborada pelo Autor

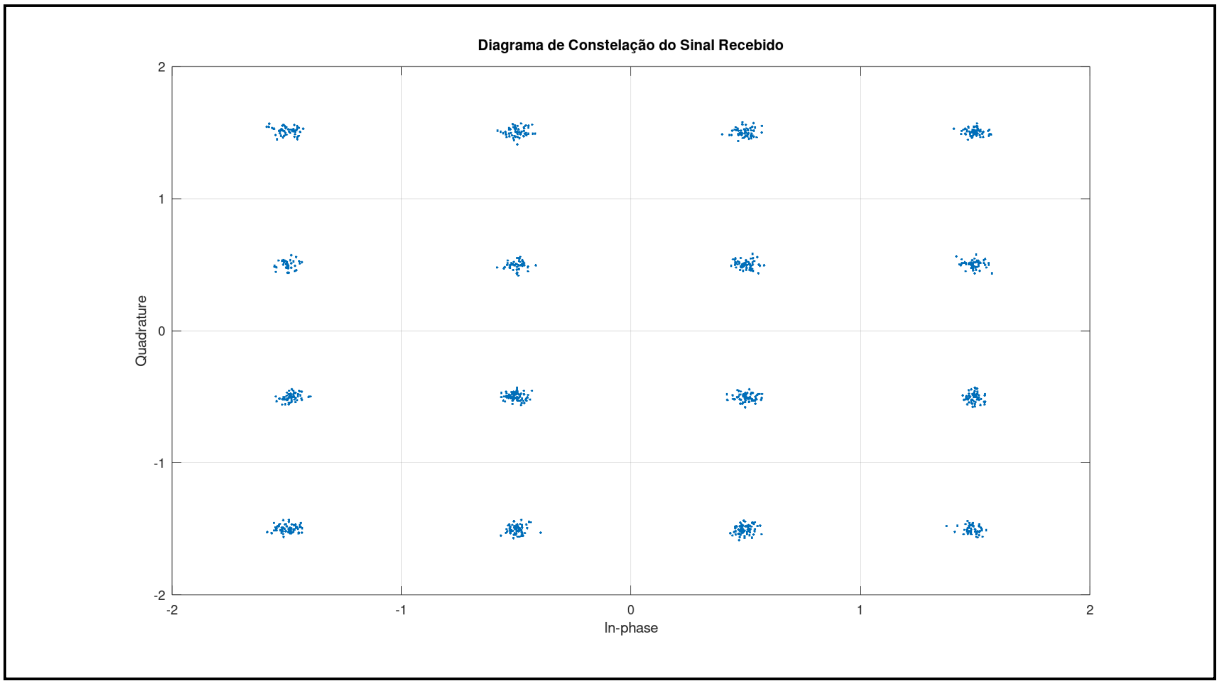



Diagrama de constelação QAM do sinal Recebido

### 2.2.9. Comparação das componentes real e imaginária:

Podemos também realizar a comparação do sinal no dominio do tempo nas componentes real e imaginária, onde é possível observar a diferença entre o sinal transmitido e o sinal recebido, e a presença do ruído no sinal recebido, isso pode ser visualizado através do script abaixo:

```
figure;
subplot(211);
plot(t(1:length(info_mod_tx)), real(info_mod_tx), 'LineWidth', 2, 'k');
hold on;
plot(t(1:length(info_rx_filtered)),  real(info_rx_filtered),  'LineWidth', 2, 'b');
title('Comparação da Componente Real');
xlabel('Tempo (s)');
ylabel('Amplitude');
legend('Transmitida', 'Recebida');
xlim([0 10 * Tb]);
ylim([-5 5]);
grid on;

subplot(212);
plot(t(1:length(info_mod_tx)), imag(info_mod_tx), 'LineWidth', 2, 'r');
hold on;
plot(t(1:length(info_rx_filtered)),  imag(info_rx_filtered),  'LineWidth', 2, 'b');
title('Comparação da Componente Imaginária');
xlabel('Tempo (s)');
ylabel('Amplitude');
legend('Transmitida', 'Recebida');
xlim([0 10 * Tb]);
ylim([-5 5]);
grid on;
```

A figura abaixo apresenta as diferenças entre as componentes real e imaginária 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 13: Elaborada pelo Autor

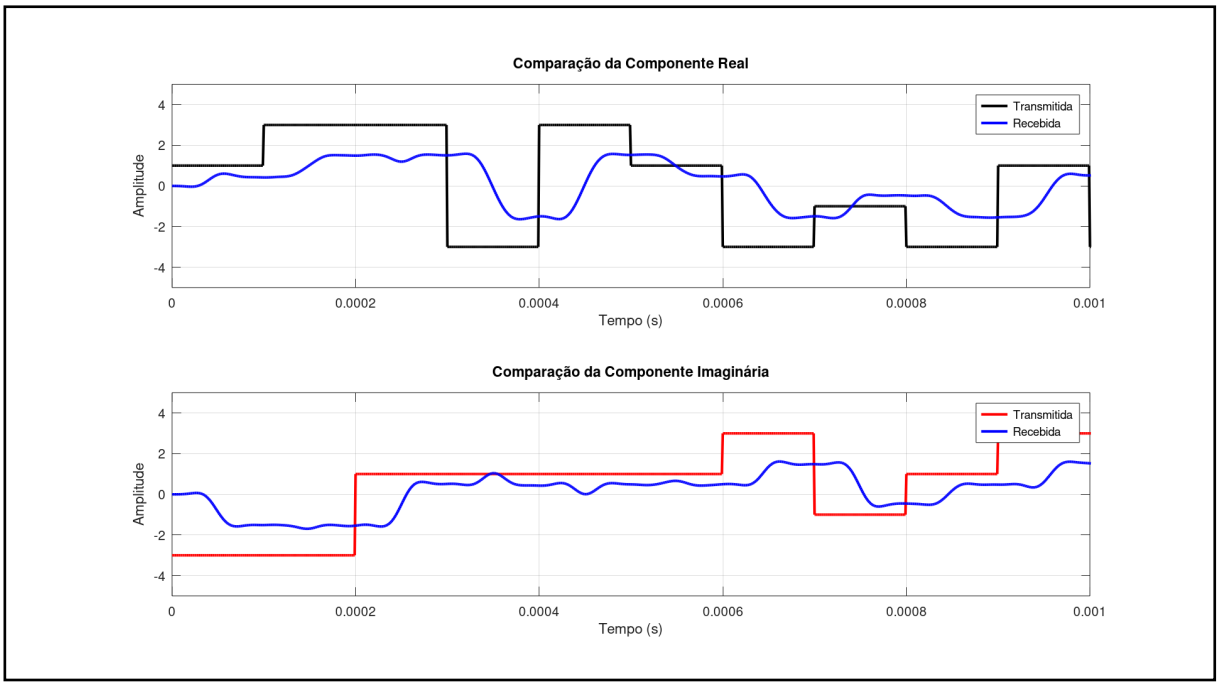



Diagrama de constelação QAM do sinal Recebido

# 3. Conclusão:

A partir dos conceitos vistos, do desenvolvimento e dos resultados obtidos, é possível concluir que a modulação QAM é uma técnica de modulação digital que permite transmitir dados de forma eficiente e robusta, onde é possível transmitir múltiplos bits por símbolo, oque aumenta a eficiência espectral do sinal.

Também é possivel concluir que com uma maior SNR do sinal de transmissão, é possível obter um sinal de melhor qualidade, onde o sinal recebido é mais próximo do sinal transmitido, e a qualidade do sinal é melhor.

E partir de um valor de SNR suficientemente alto, podemos aumentar a banda de transmissão do canal sem aumentar a taxa de erro de bit, oque permite transmitir mais dados em um mesmo canal de comunicação, além de não consumir mais energia na transmissão.

# 4. Referências Bibliográficas:

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