Relatório Projeto 2

Comunicações Móveis

Autoria	**Matrícula**
Pedro Henrique Dornelas Almeida	242110048

Programa de Pós Graduação em Engenharia Elétrica
Universidade de Brasília

24 de novembro de 2025

1 Simulação

O projeto foi desenvolvido em Matlab e está dividido em quatro sessões. São elas: (i) Simulação da probabilidade de interrupção em um canal Rayleigh; (ii) Simulação da probabilidade de interrupção em canais Rice; (iii) Simulação da probabilidade de erro de símbolo em canais Rayleigh e canais AWGN; e, (iv) Funções auxiliares. As três sessões de simulações seguem a seguinte organização:

1. Definição de Variáveis: faz-se a definição de variáveis que serão utilizadas ao longo da sessão;

2. Simulação: faz-se toda a simulação e lógicas necessárias para implementar as fórmulas e geração de variáveis aleatórias, tornando possível implementar as métricas de desempenho a serem avaliadas. Aqui se faz o uso de funções auxiliares para simular os canais de acordo com suas distribuições e gerar as métricas necessárias;

3. Geração de Figuras: a lógica deste trecho foi feita para gerar as figuras padronizadas e de forma automática, não necessitando de ajustes posteriores.

As funções Auxiliares,por sua vez, estão no trecho final do código e são utilizadas durante o código. Em geral são funções criadas para facilitar o entendimento e encapsular uma lógica específica. Pode ser reaproveitável em vários locais, o que facilita a reprodução, caso necessário.

A simulação foi realizada em um único arquivo e segue a ordem descrita no guia do projeto e nos slides apresentados em sala.

Para rodar a simulação, basta executar o arquivo *simulation.m*. Disponível em https://github.com/pedrodornelas/ComMoveis-Metrics.

2 Probabilidade de Interrupção

A probabilidade de interrupção é definida como a probabilidade de haver uma interrupção da comunicação caso a relação sinal ruído caia abaixo de um limiar, geralmente definido pelo operador da rede. Esta métrica é essencialmente importante pois pode ajudar a prever em quais condições de canal a comunicação será mais provável de falhar.

2.1 Canal Rayleigh

Utilizando de um canal Rayleigh foram geradas simulações e curvas téoricas para a probabilidade de interrupção. A simulação está descrita a seguir.

2.1.1 Definição de Variáveis

No trecho abaixo faz-se as definições de variáveis que serão utilizadas durante a simulação.

```
%% Rayleigh Channel Simulation

points = 100;
N = 15;  % simulation points
samples = 1e7 ;

gamma_th_db = linspace(-30, 30, points);   % para ficar igualmente distribuido no gráfico em db
gamma_th = 10 .^ (gamma_th_db ./ 10);      % conversao para linear

gamma_th_sim_db = linspace(-30, 30, N);        % para ficar igualmente distribuido no gráfico em db
gamma_th_sim = 10 .^ (gamma_th_sim_db ./ 10);  % conversao para linear

gamma_bar_db = [-20, 0, 20];
gamma_bar = 10 .^ (gamma_bar_db ./ 10);
```

Figura 1: Definição de Variáveis

Vale ressaltar que a mesma lógica será utilizada na definição de variáveis das simulações restantes. Para a implementação da curva teórica serão utilizado 100 pontos para termos uma curva contínua e suave. Para a simulação utilizaremos 15 pontos, o que já é suficiente para verificar a aderência dos pontos simulados às curvas teóricas.

2.1.2 Simulação

Para facilitar a implementação e para encapsular a lógica do processamento da simulação foram feitas funções genéricas para gerar as curvas teóricas e os pontos simulados. A seguir está o trecho do código que realiza as curvas pedidas no relatório.

```
% Processing Outage Probability
pout = zeros(points, length(gamma_bar));
pout_sim = zeros(N, length(gamma_bar));
for i = 1 : length(gamma_bar)
    pout(:, i) = pout_rayleigh(gamma_th, gamma_bar(i));
    [pout_sim(:, i)] = pout_sim_rayleigh(gamma_th_sim, gamma_bar(i), samples);
end
```

Figura 2: Processamento da Simulação

Note que são utilizadas duas funções: (i) `pout_rayleigh`, que implementa a curva teórica; e (ii) `pout_sim_rayleigh`, que gera os pontos simulados para a probabilidade de interrupção. Estas funções encapsulam a lógica da implementação para que consigamos utilizar várias vezes e facilite na geração das curvas implementadas neste trabalho.

Abaixo temos a função `pout_rayleigh` que gera a curva teórica para a probabilidade de interrupção conforme sua fórmula encontrada.

```
%% Auxiliar Functions
function [pout] = pout_rayleigh(gamma_th, gamma_bar)
    pout = 1 - exp(-(gamma_th ./ gamma_bar));
end
```

Figura 3: Probabilidade de Interrupção Teórica - Rayleigh

A seguir vemos a implementação da simulação de Monte Carlo para gerar os pontos com base em variáveis aleatórias por meio da função `pout_sim_rayleigh`. Veja

que aqui normalizamos as variáveis aleatórias para mantermos a energia unitária, o que facilita para validar os resultados e sobrepor às curvas teóricas. O mesmo princípio será utilizado nas simulações posteriores, em especial a simulação da SEP, onde mantemos a energia dos símbolos unitária e variamos a SNR média variando a potência do ruído, isto será de extrema importância para realizarmos o processo de demodulação mais facilmente.

```
173  function [pout] = pout_sim_rayleigh(gamma_th, gamma_bar, N)
174      % pout = ones(1, length(gamma_th));
175      for i = 1 : length(gamma_th)
176          % normalized channel
177          h = (1/sqrt(2)) * (randn(N, 1) + 1j*randn(N, 1));
178          beta = abs(h);
179          % gamma_s
180          gamma_s = gamma_bar .* (beta .^ 2);
181          % P(gamma_s < gamma_th)
182          pout(i) = mean(gamma_s < gamma_th(i));
183      end
184  end
```

Figura 4: Probabilidade de Interrupção Simulada - Rayleigh

É importante entender como a simulação acima foi feita pois as demais simulações também utilizarão o mesmo princípio. Para realizar uma simulação de Monte Carlo faz-se o uso de variáveis aleatórias, como dito no trecho anterior, porém, precisa seguir alguns princípios para que tenha sucesso.

Primeiro, é necessário garantir um número de amostras suficientemente grande para que a lei dos grandes números torne a aproximação bem próxima à realidade. Segundo, para os canais de comunicações sem fio que se utilizam da modelagem estocástica para tentar prever o comportamento do canal a simulação deve se basear na geração de variáveis aleatórias seguindo a premissa anterior. Aqui, estamos utilizando um canal que segue a distribuição Rayleigh, e é possível descrever a sua envoltória da seguinte forma:

$$\beta(t) = |h(t)| = \sqrt{x(t)^2 + y(t)^2},$$

onde $x(t)$ e $y(t)$ são variáveis gaussianas independentes $x, y \sim \mathcal{N}(0, \sigma^2)$. Para o modelo normalizado (potência média unitária) tem-se $\sigma^2 = 1/2$. Vale ressaltar que um canal com distribuição Rayleigh não modela linha de visada, portanto, a envoltória acima é puramente sem linha de visada.

Tendo isto, é possível então gerar a envoltória de forma estatística, ou seja, gerando as variáveis aleatórias com distribuição Gaussiana, em seguida, descobre-se as amostras da envoltória do canal, transformamos na SNR do canal ($\Gamma = \beta^2$), com Γ sendo a variável aleatória que representa a SNR do canal. O próximo passo é encontrar a SNR instantânea por símbolo ($\gamma_s = \bar{\gamma}\beta^2$), e por fim, calculamos quantas amostras ficaram com a SNR abaixo do limiar γ_{th}.

2.1.3 Resultados

Como resultado das expressões implementadas e do processamento realizado foi possível gerar a figura a seguir das curvas da probabilidade de interrupção para três cenários de SNR médias ($\bar{\gamma}$), variando o limiar γ_{th} (dB) para observar os diferentes cenários.

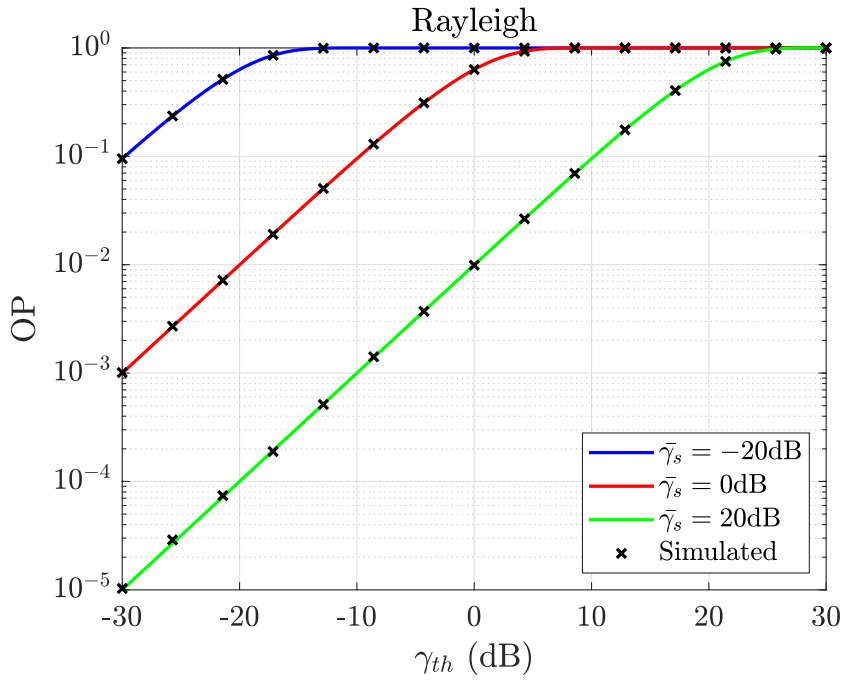

Figura 5: Curvas de Probabilidade de Interrupção - Rayleigh

É importante notar da figura acima que houve uma boa convergência dos dados simulados e as curvas teóricas, observada pela sobreposição dos símbolos simulados com as curvas teóricas. Este fato é justificado pelo fato de que foram utilizados 10^7 amostras na simulação e a menor probabilidade de interrupção observada no intervalo foi de aproximadamente 10^{-5}.

2.2 Canal Rice

Neste momento foi realizada uma implementação da probabilidade de interrupção utilizando um canal Rice para gerar as amostras simuladas e gerar as curvas teóricas.

2.2.1 Definição de Variáveis

No trecho abaixo faz-se as definições de variáveis que serão utilizadas durante a simulação. Aqui faz-se necessário gerar três figuras com fatores de Rice diferentes (K_r), para isto, a variável `kr` tem 3 valores, e serão os valores utilizados para gerar os resultados.

```
%% Rice Channel Simulation

points = 100;
N = 15;  % simulation points

gamma_th_db = linspace(-30, 30, points);   % para ficar igualmente distribuido no gráfico em db
gamma_th = 10 .^ (gamma_th_db ./ 10);      % conversao para linear

gamma_th_sim_db = linspace(-30, 30, N);         % para ficar igualmente distribuido no gráfico em db
gamma_th_sim = 10 .^ (gamma_th_sim_db ./ 10);   % conversao para linear

gamma_bar_db = [-20, 0, 20];
gamma_bar = 10 .^ (gamma_bar_db ./ 10);

kr = [0.1, 1, 10];
```

Figura 6: Definição de Variáveis

2.2.2 Simulação

A simulação foi feita neste cenário da mesma forma da anterior, com funções para facilitar e encapsular a lógica de processamento, tornando mais fácil reproduzir a simulação e gerar canais do tipo Rice. Para isto, a simulação foi implementada da seguinte forma:

```
for j = 1 : length(kr)
    % Processing Outage Probability
    pout = zeros(points, length(gamma_bar));
    pout_sim = zeros(N, length(gamma_bar));
    for i = 1 : length(gamma_bar)
        pout(:, i) = pout_rice(gamma_th, gamma_bar(i), kr(j));
        [pout_sim(:, i)] = pout_sim_rice(gamma_th_sim, gamma_bar(i), samples, kr(j));
    end
end
```

Figura 7: Processamento da Simulação

A função `pout_rice` implementa a curva teórica da probabilidade de interrupção em um canal rice com fator `kr`:

```
function [pout] = pout_rice(gamma_th, gamma_bar, kr)
    pout = 1 - marcumq(sqrt(2*kr) , sqrt(2 * ((kr + 1) ./ gamma_bar) .* gamma_th));
end
```

Figura 8: Probabilidade de Interrupção Teórica - Rice

Em seguida, temos a função `pout_sim_rice`, que implementa a probabilidade de interrupção simulada, ou seja, por meio de geração de variáveis aleatórias em um canal Rice:

```
function [pout] = pout_sim_rice(gamma_th, gamma_bar, N, kr)
    % pout = ones(1, length(gamma_th));
    for i = 1 : length(gamma_th)
        % line of sight component
        h_los = sqrt(kr/(kr+1));
        % non-line of sight component
        h_nlos = sqrt(1/(kr+1)) * (randn(N, 1) + 1j*randn(N, 1)) / sqrt(2);
        % normalized channel
        h = h_los + h_nlos;
        beta = abs(h);
        % gamma_s
        gamma_s = gamma_bar .* (beta .^ 2);
        % P(gamma_s < gamma_th)
        pout(i) = mean(gamma_s < gamma_th(i));
    end
end
```

Figura 9: Probabilidade de Interrupção Simulada - Rice

Vale ressaltar que os princípios seguidos para a implementação desta função foram os mesmos da simulação feita para o canal Rayleigh, porém, no caso do canal Rice, temos algumas diferenças. Podemos observar que um canal do tipo Rice considera em seu modelo a linha de visada, portanto, o sistema a implementar será diferente pois além das componentes sem linha de visada, agora temos também as componentes com linha de visada.

Neste modelo a linha de visada é considerada determinística, então, basta consideramos a proporção do fator de rice para gerar a componente com linha de visada. Por sua vez, a componente sem linha de visada é aleatória e complexa. Portanto, o canal fica com o seguinte modelo:

$$h = \sqrt{\frac{K}{K+1}} + \sqrt{\frac{1}{K+1}}\, h_{\text{NLOS}}, \qquad h_{\text{NLOS}} \sim \mathcal{N}(0,1),$$

com h_{NLOS} sendo exatamente h de um canal Rayleigh, como mostrado na sessão anterior. Repare nas normalizações feitas para manter $E[|h|^2] = 1$, foram elas a normalização do fator de Rice e também das variáveis aleatórias da h_{NLOS}.

2.2.3 Resultados

Abaixo estão as figuras geradas com diferentes fatores de Rice mostraram uma boa aderência entre curvas simuladas e teóricas, validando o modelo e as expressões implementadas.

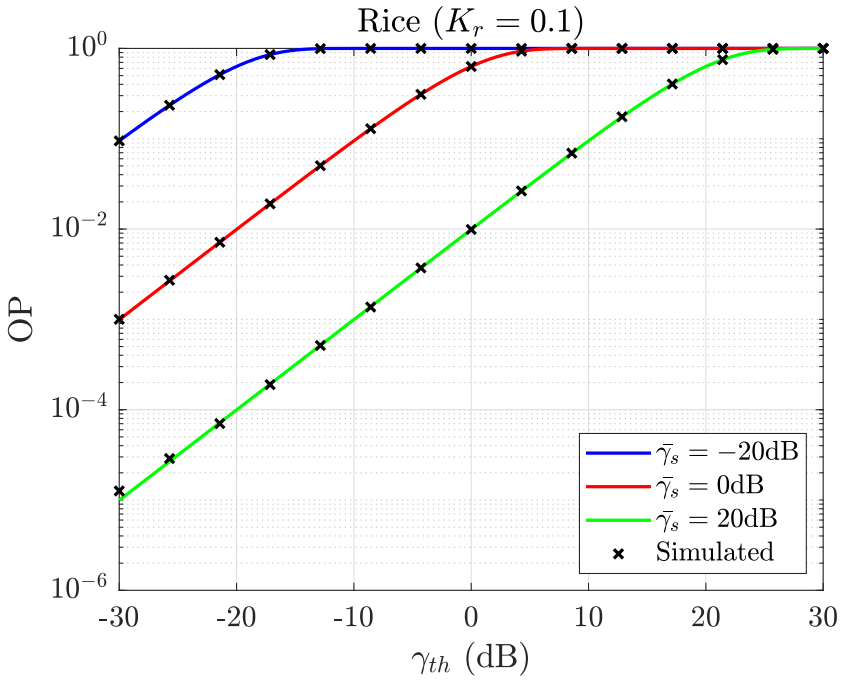

Figura 10: Curvas de Probabilidade de Interrupção - Rice $K_r = 1$

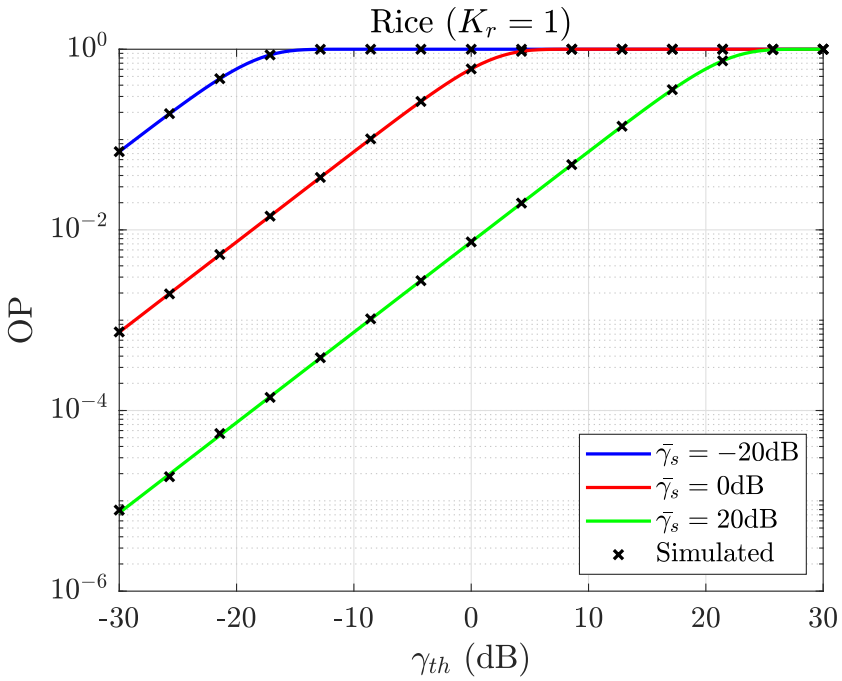

Figura 11: Curvas de Probabilidade de Interrupção - Rice $K_r = 1$

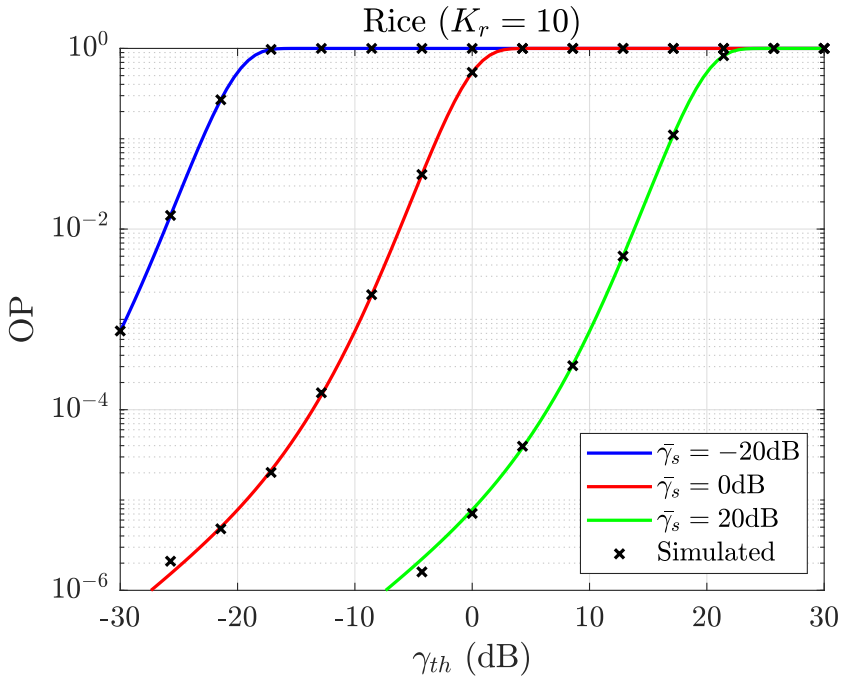

Figura 12: Curvas de Probabilidade de Interrupção - Rice $K_r = 10$

Nota-se da última figura que a aderência dos pontos simulados em baixos valores de probabilidade ficaram comprometidos, isto se deve ao fato de que a simulação estava utilizando 10^7 amostras, o que não contém amostras suficientes para estimar bem o comportamento em baixos valores de probabilidade de erro, que está próximo a 10^{-6}.

Vale enaltecer também que ao aumentar o fator de Rice K_r é possível ver que as curvas de probabilidade de interrupção ficam mais determinísticas e há um melhor desempenho devido à maior contribuição da componente de visada direta.

3 Probabilidade de Erro de Símbolo

Nesta sessão será apresentada a análise da probabilidade de erro de símbolo em um canal Rayleigh e, a fim de comparação, também foram geradas curvas para um canal AWGN. Foram utilizados aqui modelos M-QAM de diferentes ordens para observamos o efeito.

3.1 Definição de Variáveis

Para esta simulação foram utilizadas as seguintes variáveis:

```
113   %% SEP Analisys
114   M = [4, 16, 64];
115
116   gamma_bar_db = linspace(-10, 30, points);
117   gamma_bar = 10 .^ (gamma_bar_db ./ 10);
118
119   gamma_bar_sim_db = linspace(-10, 30, N);
120   gamma_bar_sim = 10 .^ (gamma_bar_sim_db ./ 10);
121
122   sep = zeros(points, length(M));
123   sep_awgn = zeros(points, length(M));
```

Figura 13: Definição de Variáveis

3.2 Simulação

Seguindo a mesma lógica das simulações anteriores, foram feitas funções para encapsular a lógica de processamento e tornar o programa mais fácil de ser compreendido e reutilizado. O processamento foi feito da seguinte forma:

```
125 for i = 1 : length(M)
126     [sep(:, i), sep_awgn(:, i)] = sep_rayleigh(gamma_bar, M(i));
127     [sep_sim(:, i), sep_sim_awgn(:, i)] = sep_sim_rayleigh(gamma_bar_sim, samples, M(i));
128 end
```

Figura 14: Processamento da Simulação

A função `sep_rayleigh` gera as curvas teóricas da probabilidade de erro de símbolo tanto para o canal Rayleigh quanto para o canal AWGN:

```
203 function [sep, sep_awgn] = sep_rayleigh(gamma_bar, M)
204     % sep
205     cm = sqrt( (1.5 .* gamma_bar) ./ (M - 1 + (1.5 .* gamma_bar)) );
206     cte = (sqrt(M) - 1) / sqrt(M);
207     sep = (2 .* cte .* (1 - cm)) - ((cte .^ 2) .* (1 - ((4/pi) .* cm .* atan(1./cm))));
208     % sep awgn
209     q = qfunc(sqrt( (3 ./ (M - 1)) .* gamma_bar ));
210     A = 4 * (1 - 1/sqrt(M));
211     sep_awgn = A .* q - (A.^2 ./ 4) .* (q.^2);
212 end
```

Figura 15: Probabilidade de Erro de Símbolo - Teórica

Por sua vez, a função `sep_sim_rayleigh` implementa a simulação da probabilidade de erro de símbolo para compararmos com as curvas teóricas. Esta simulação, por sua vez, tem algumas peculiaridades, precisamos simular um canal com uma transmissão, ou seja, fazer a modulação em símbolos, simular o canal Rayleigh, adicionar ruído, equalizar o sinal recebido e então demodular o sinal com todos os efeitos corretamente atribuídos à transmissão dos dados.

```
function [sep, sep_awgn] = sep_sim_rayleigh(gamma_bar, N, M)
    % bits
    bits = randi([0, M-1], N, 1);
    % conversion to QAM symbols
    symbols = qammod(bits, M, 'UnitAveragePower', true);
    for i = 1 : length(gamma_bar)
        % normalized channel
        h = (1/sqrt(2)) * (randn(N, 1) + 1j*randn(N, 1));
        beta = abs(h);
        % normalized noise power, Es = 1
        sigma = 1 ./ gamma_bar(i);
        % awgn noise
        noise = sqrt(sigma/2) .* (randn(N, 1) + 1j*randn(N, 1));
        % received signal
        received = (beta .* symbols) + noise;
        % awgn
        received_awgn = symbols + noise;
        % equalization
        equalized = received ./ beta;
        % demodulation
        received_bits = qamdemod(equalized, M, 'UnitAveragePower', true);
        % awgn demodulation
        received_bits_awgn = qamdemod(received_awgn, M, 'UnitAveragePower', true);
        % SEP
        sep(i) = mean(bits ~= received_bits);
        % SEP awgn
        sep_awgn(i) = mean(bits ~= received_bits_awgn);
    end
end
```

Figura 16: Probabilidade de Erro de Símbolo - Simulada

Como visto acima, o canal AWGN é mais fácil de implementar, pois só é necessário adicionar o ruído ao sinal recebido. Após seguir o passo a passo para gerar o sinal recebido demodulado, pode-se então fazer a média dos bits recebidos erroneamente para de determinar a probabilidade de erro de símbolo, tanto para o caso Rayleigh quanto para o caso AWGN.

3.3 Resultados

Com as simulações realizadas, foi possível observar o seguinte:

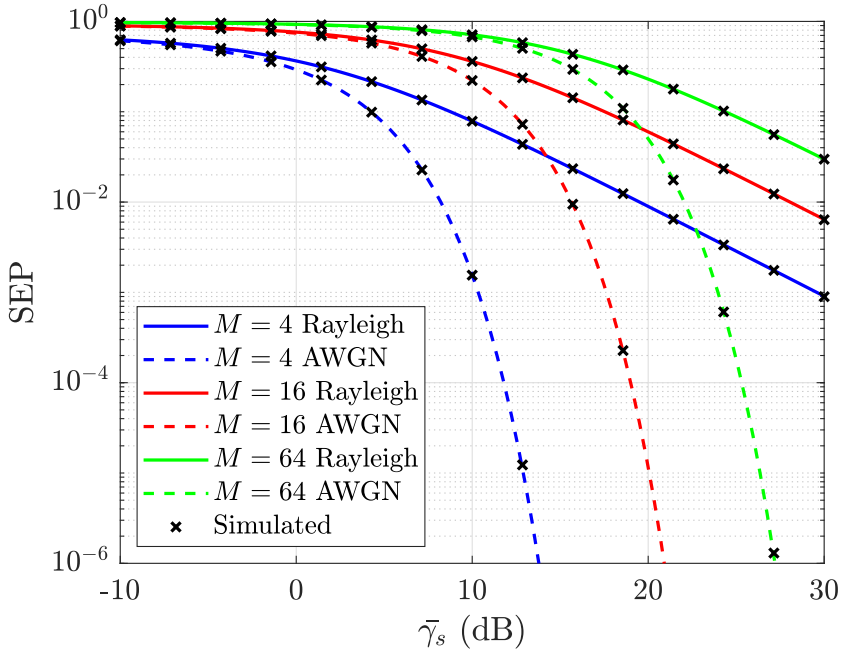

Figura 17: Curvas de Probabilidade de Erro de Símbolo - Rayleigh e AWGN

Note que há bastante diferença do desempenho utilizando um canal Rayleigh e um canal AWGN, mostrando que o canal com desvanecimento Rayleigh tem um desempenho bem pior que um canal AWGN. Isto é de esperar pois o desvanecimento é bem danoso ao canal, e o canal Rayleigh ainda considera um canal sem linha de visada, o que torna o canal mais aleatório, causando uma piora no quadro geral deste canal.

Note que ao aumentar o esquema de modulação há uma queda no desempenho, isto é de se esperar pois o receptor terá mais dificuldade em estimar o símbolo recebido, apesar de que ao aumentar a ordem da modulação ganhamos na capacidade do canal. Isto deve ser considerado para ser elaborado um *trade-off*, ou seja, caso tenhamos boas condições de canal, e de SNR, pode-se aumentar a modulação para obter uma melhor taxa, enquanto que para baixas SNRs, ou condições ruins de canal, é melhor priorizar uma menor taxa mas com o índice de acerto um pouco maior para garantir a comunicação.

4 Conclusão

Por último, foi possível observar em todas as simulações uma aderência aceitável, com uma boa sobreposição das curvas teóricas e simuladas. Foram observadas duas métricas extremamente importantes para qualquer sistema de comunicação que se deseja implementar, vale ressaltar que existem outras bem importantes também, como a capacidade ergótica do canal, dentre várias outras. Foi possível também aprender e simular canais estatísticos por meio de simulações de Monte Carlo, que são imprescindíveis em comunicações móveis e na área de telecomunicações.