

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO SUPERIOR DE BACHARELADO EM ENGENHARIA DE
TELECOMUNICAÇÕES

Emanuel Savegnago Maziero

**TRABALHO FINAL DE PROJETO DE FILTROS DIGITAIS:
EQUALIZAÇÃO DE CANAL EM CENÁRIOS DE MULTIPATH COM
FILTROS ADAPTATIVOS**

Santa Maria, RS
2023

SUMÁRIO

1	INTRODUÇÃO	3
1.1	DESCRIÇÃO DO FENÔMENO E DO PROBLEMA	3
1.2	ABORDAGEM ESCOLHIDA PARA SOLUCIONAR O PROBLEMA	4
2	DESENVOLVIMENTO	5
2.1	SISTEMA DE TELECOMUNICAÇÕES SIMPLIFICADO	5
2.1.1	Bloco de Transmissão	5
2.1.2	Canal	6
2.1.3	Bloco de Recepção	7
2.1.4	Análise de performance desse sistema	8
2.2	MODELO DE CANAL COM ECO	10
2.3	FILTRO ADAPTATIVO LEAST MEAN SQUARES (LMS)	13
2.4	EQUALIZAÇÃO DE UM CANAL COM ECO USANDO UM FILTRO LMS....	14
2.4.1	L = 10, 5% dos símbolos para treinamento	15
2.4.2	L = 10, 10% dos símbolos para treinamento	16
2.4.3	L = 20, 10% dos símbolos para treinamento	18
2.4.4	Análise e discussão dos resultados da equalização	19
3	CONCLUSÃO	21
	REFERÊNCIAS BIBLIOGRÁFICAS	22
	APÊNDICE A – CÓDIGOS DO MATLAB	23
A.1	SISTEMA COM CANAL IDEAL	23
A.2	FUNÇÃO DE MODULAÇÃO QPSK	24
A.3	FUNÇÃO DE DEMODULAÇÃO QPSK	26
A.4	SISTEMA COM CANAL EM CENÁRIO COM MULTIPATH	27
A.5	SISTEMA COM CANAL EM CENÁRIO COM MULTIPATH E EQUALIZAÇÃO LMS	29
A.6	FUNÇÃO DO EQUALIZADOR LMS	33
A.7	FUNÇÃO DO EQUALIZADOR NLMS	34

1 INTRODUÇÃO

1.1 Descrição do Fenômeno e do Problema

O eco é um fenômeno geralmente relacionado ao som, onde uma onda sonora que parte de uma fonte e reflete em uma superfície, retornando ao seu ponto de origem, sendo assim percebido como a repetição do som original de forma atenuada. A Figura 1.1 mostra um exemplo desse fenômeno.

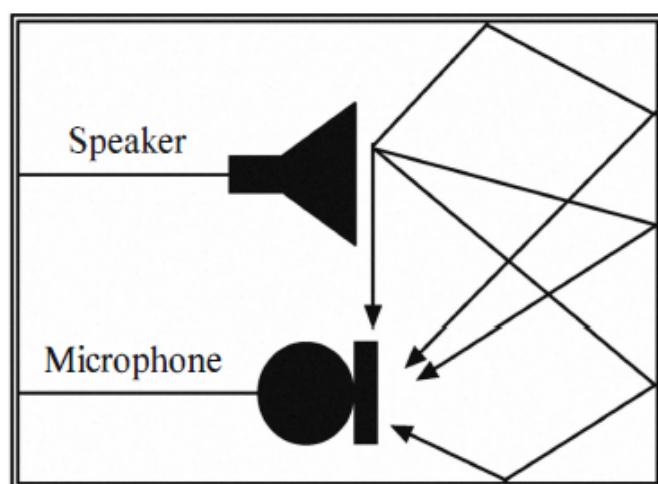


Figura 1.1 – Eco acústico. Fonte: [1]

Nessa Figura, o áudio de um alto-falante (*speaker*) que foi refletido várias vezes em diferentes direções é captado em um microfone, que por sua vez vai amplificar versões atenuadas desse áudio uma atrás da outra, gerando a repetição inconveniente e indesejada de um som.

A outra definição de eco é quando um sinal (onda eletromagnética que carrega informação) que foi transmitido sofre com o fenômeno dos múltiplos caminhos (*multipath*), o qual diz respeito ao fato de que essas ondas podem chegar a um receptor por múltiplos caminhos a partir de reflexões em diferentes superfícies, como metal e água. A Figura 1.2 exemplifica como o sinal de uma antena transmissora chega em uma antena receptora de um telefone de formas diferentes.

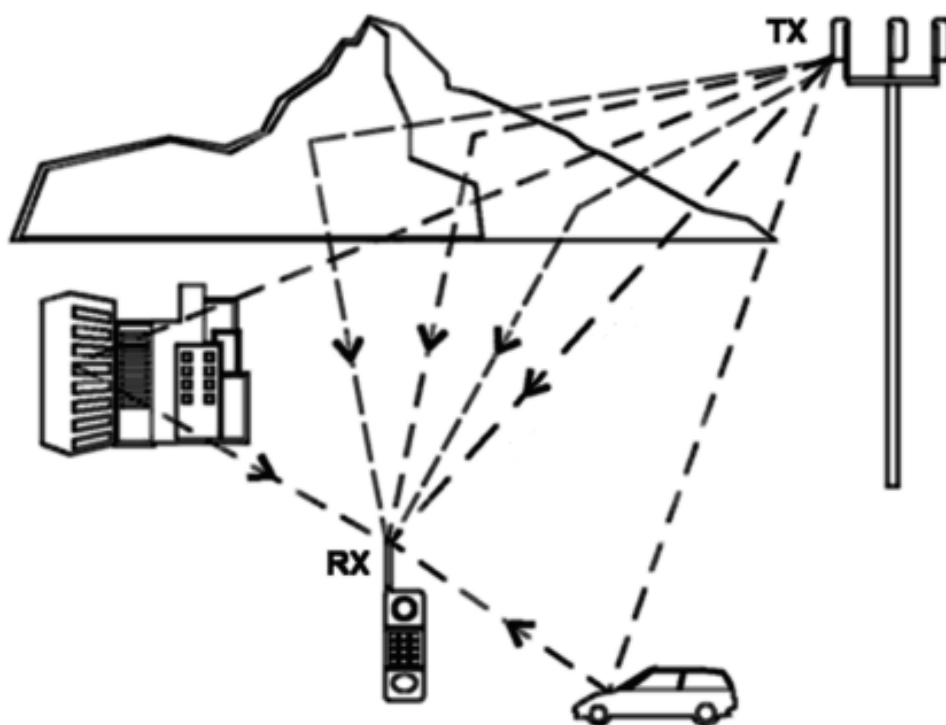


Figura 1.2 – Transmissão em um Cenário com *multipath*. Fonte: [2]

Nessa Figura, onde TX é o transmissor e RX o receptor, os vários caminhos formados pelos sinais vão chegar em tempos diferentes e com amplitudes diferentes nesse RX, ou seja, terão atraso e atenuação. Isso é prejudicial ao sistema pois esses ecos causarão interferências e tornarão a recuperação do sinal mais difícil. Nesse caso a interferência é Interferência Intersimbólica (ISI), que é quando um símbolo se sobrepõe ao outro e dificulta a recuperação do sinal.

1.2 Abordagem Escolhida para Solucionar o Problema

Para ilustrar o efeito desse fenômeno na performance de um sistema de telecomunicações, esse sistema será simulado em MATLAB de forma simplificada e os resultados serão analisados. Isso será feito em três etapas: inicialmente, transmissão e recepção considerando um canal ideal; depois, transmissão e recepção considerando canais em cenários de *multipath*; e por fim, os mesmos cenários de *multipath* mas agora com o uso de uma técnica de filtragem adaptativa chamada equalização por Least Mean Squares (LMS) na recepção para tentar minimizar esse problema.

2 DESENVOLVIMENTO

2.1 Sistema de Telecomunicações Simplificado

O sistema de telecomunicações simplificado que será implementado nesse trabalho terá o ruído representado na forma de ruído gaussiano branco aditivo e a relação entre a potência do sinal e a potência do ruído (Signal-Noise Ratio, SNR) será testada na faixa de -10 até 20dB. O número de bits padrão para as simulações é $1 \cdot 10^5$, a *SymbolRate* é 16MHz, a frequência da portadora f_c é 32MHz e a frequência de amostragem f_s é 256MHz. Para analisar o sistema em termos de Bit Error Rate (BER), a *Ratio of bit energy to noise power spectral density* (E_b/N_0), uma medida normalizada da SNR que é chamada de SNR por bit será utilizada já que essa é a convenção nesse tipo de análise.

2.1.1 Bloco de Transmissão

O sistema de telecomunicações implementado, na parte da transmissão, será da forma mostrada pela Figura 2.1.

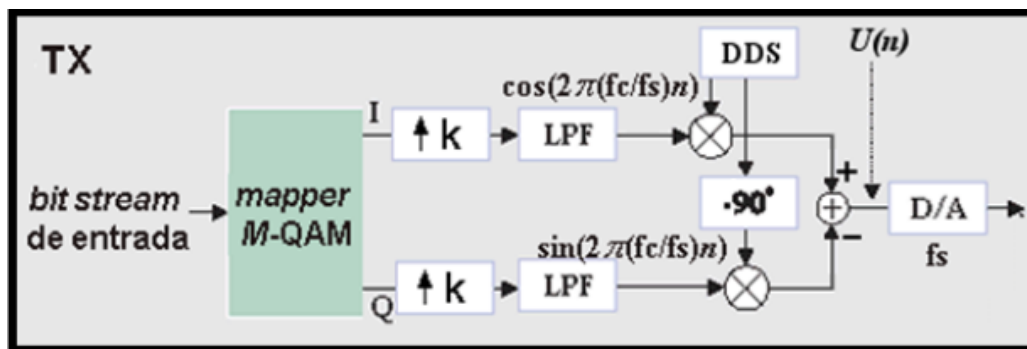


Figura 2.1 – Bloco de transmissão. Fonte: [3]

Nesse bloco, o processo inicia com uma sequência de dados em forma de bits que serão aplicados a um mapper Quadrature Phase-Shifting Key (QPSK), que mapeará os bits em um símbolo IQ correspondente, que no caso da modulação QPSK, é um símbolo de dois bits. Nessa modulação, apenas a fase da portadora é alterada e sua amplitude se mantém constante. Isso é mostrado na Figura 2.2.

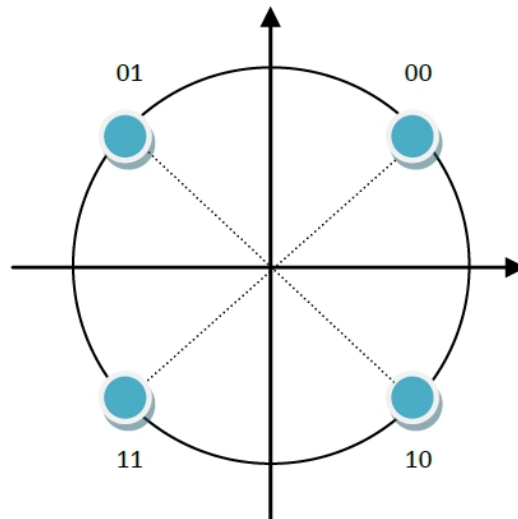


Figura 2.2 – Constelação QPSK. Fonte: [4]

Depois disso, a parte real I e a parte imaginária Q serão aplicadas em um superamostrador (*upsampler*) que irá aumentar a taxa de amostragem pelo fator k ($k = \frac{\text{Freq. de amostragem}}{\text{Symbol Rate}}$) de modo a adicionar $k-1$ zeros após cada amostra, para então ser filtrado por um filtro passa-baixas gaussiano (*lowpass filter, LPF*) que é também chamado de *shaping filter*, já que ao eliminar as altas frequências ele irá suavizar mudanças abruptas e então representar aquele bit não só por um "impulso", mas sim por algo que chega nesse pico de forma progressiva, já que vai seguir a distribuição gaussiana. Nesse caso, o filtro será um Finite Impulse Response (FIR), caracterizado por uma resposta ao impulso finita, estabilidade garantida e fase linear.

A penúltima etapa é a heterodinação, mostrada na Figura 2.3. que faz a translação do sinal de banda-base (centrado na frequência zero) para a banda-passante, cuja frequência central é a frequência da portadora. Após isso, os sinais das duas partes são combinados gerando o sinal que irá ser convertido para analógico, amplificado e transmitido.

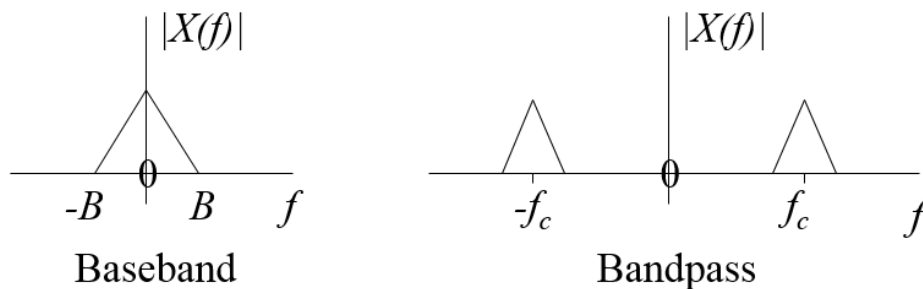


Figura 2.3 – Heterodinação. Fonte: [5]

2.1.2 Canal

Completados os processos do bloco de transmissão, o sinal será transmitido por um canal, um meio físico que irá causar a perda de potência do sinal, causada pela perda por percurso, onde a onda eletromagnética perde energia quanto mais distância percorre e pela atenuação dos *multipaths* e distorção na fase, que ocorre também pelos *multipaths*, já que o sinal refletido em superfícies vai chegar ao receptor com uma fase diferente. A

representação matemática desses efeitos a partir dos atrasos de chegada de cada sinal no receptor e o modo que ela será aplicada no sistema serão abordados em capítulo posterior.

2.1.3 Bloco de Recepção

Na parte da recepção, a Figura 2.4 representará o bloco que define os processos.

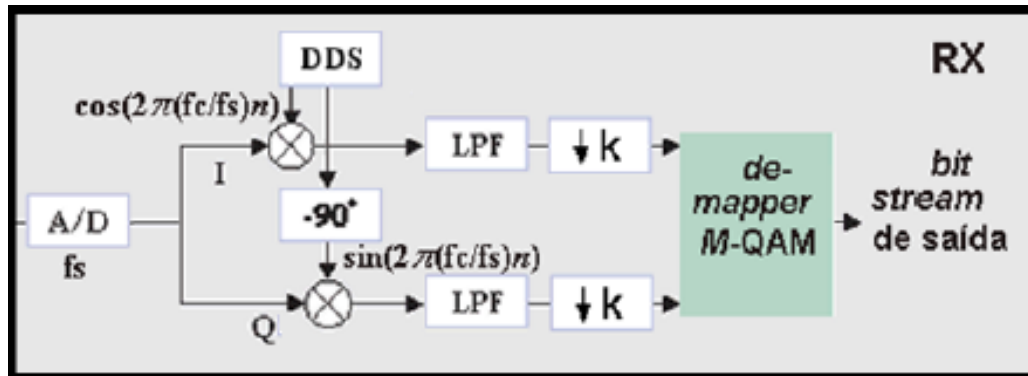


Figura 2.4 – Bloco do receptor. Fonte: [3]

Quando o sinal chega no receptor, ele é convertido para digital e então entra novamente a heterodinação, mas dessa vez fazendo a downconversão, que irá transladar o sinal de banda passante para banda base. A Figura 2.5 mostra o resultado desse processo.

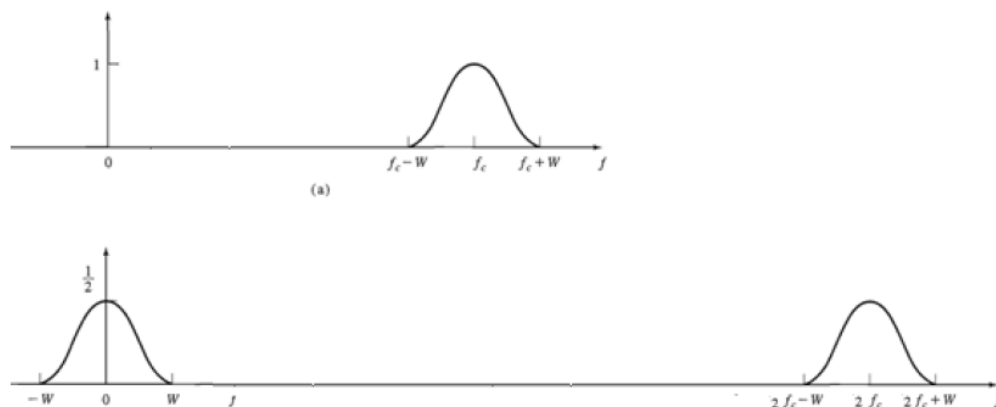


Figura 2.5 – Downconversão. Fonte: [6]

Esse processo gera réplicas indesejadas do sinal que serão eliminadas pelo mesmo LPF gaussiano. Assim, a subamostragem (*downsampling*) entra e retira uma amostra a cada duração de símbolo (definida pelo fator k), amostra essa que deve ser retirada no instante ótimo de amostragem para maximizar a SNR. Por fim, o *demapper* recebe cada uma dessas amostras e desmapeia os símbolos IQ recebidos de forma a transformá-los em bits novamente. Esse processo, no sistema considerado, é feito dividindo a constelação em quatro regiões de decisão de acordo com a Figura 2.6, onde o símbolo que cair em uma das regiões será desmapeado como os bits correspondentes daquela região.



Figura 2.6 – Demapper. Fonte: [6]

Assim, após todos esses processos, os bits recebidos são comparados com os bits enviados e então será possível analisar quantos bits foram recebidos de forma errada, permitindo o cálculo da BER em relação às diferentes SNRs.

2.1.4 Análise de performance desse sistema

Uma simulação com os parâmetros descritos no início da seção "Sistema de Telecomunicações Simplificado" foi feita e os resultados são mostrados nas cinco Figuras abaixo.

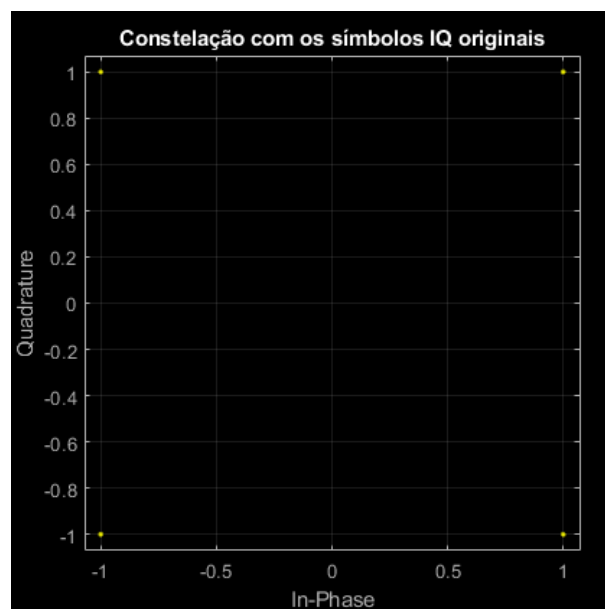


Figura 2.7 – Constelação QPSK. Fonte: Autor

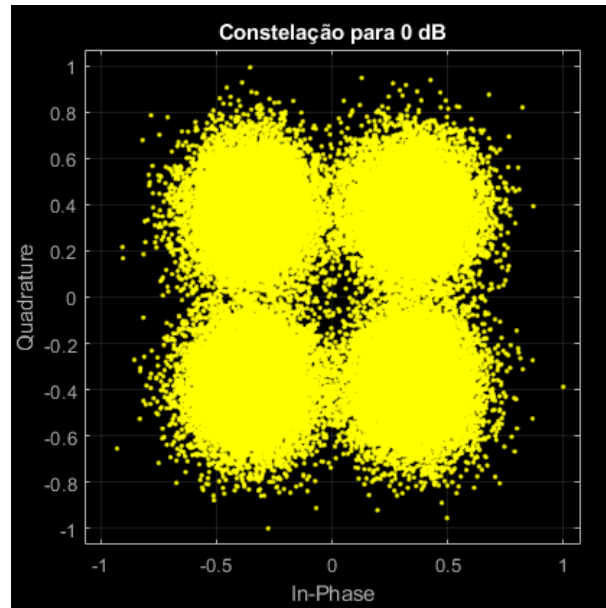


Figura 2.8 – Constelação após demodulação, SNR = 0dB. Fonte: Autor

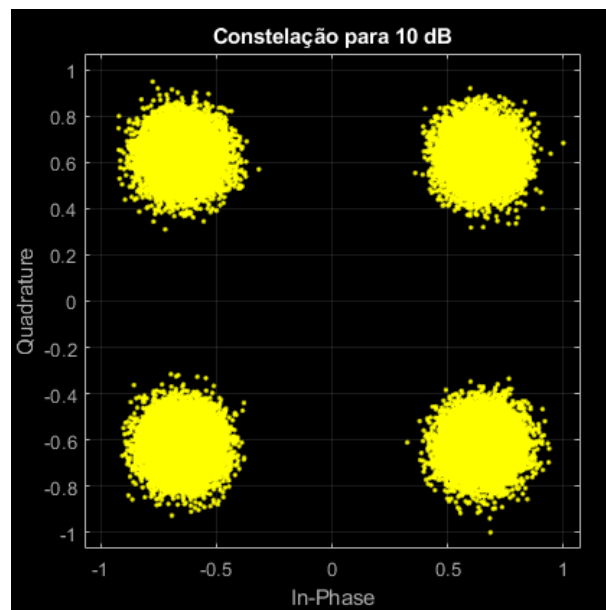


Figura 2.9 – Constelação após demodulação, SNR = 10dB. Fonte: Autor

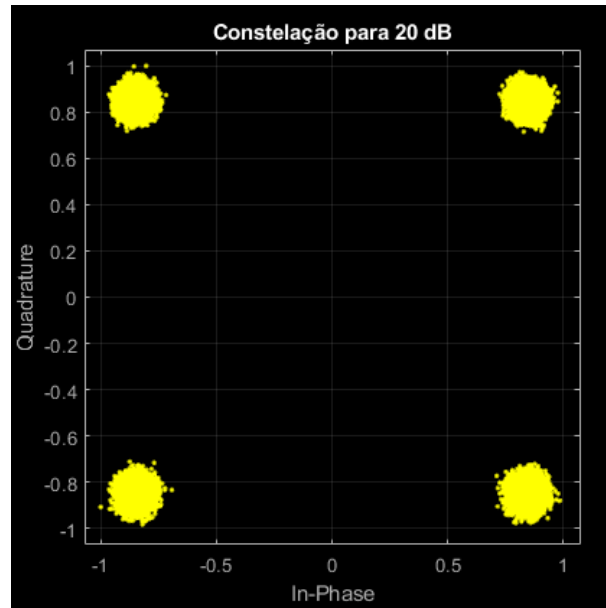


Figura 2.10 – Constelação após demodulação, SNR = 20dB. Fonte: Autor

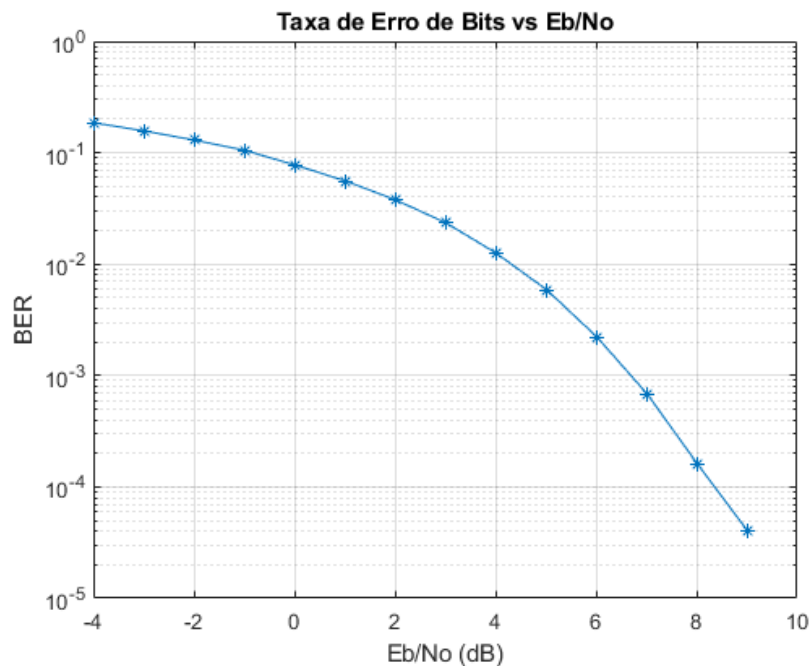


Figura 2.11 – BERxEbNo. Fonte: Autor

As imagens referentes a constelações mostram resultados esperados, quanto maior a SNR, mais próximos estarão os símbolos do seu ponto original, na forma de um foco. A Figura 2.11 mostra a curva de BERxEbNo e nela é possível ver que num sistema com canal ideal, depois do ponto de aproximadamente 9dB não são detectados mais erros.

2.2 Modelo de Canal com Eco

Para modelar como o cenário de *multipath* num canal *wireless*, se transmite um dos símbolos IQ com a antena transmissora convenientemente posicionada para se registrar a

magnitude, fase e atraso dos ecos desse símbolo transmitido de modo a gerar uma tabela como a Tabela 2.1 chamada de *delay profile*.

Tabela 2.1 – Delay profile. Fonte: [2], [7]

Brazil A	ℓ	0	2	36	49	94	95
	h_ℓ	1.00	0.20	0.15	0.18	0.21	0.15
Brazil B	ℓ	0	5	56	70	152	203
	h_ℓ	1.00	0.25	0.63	0.45	0.18	0.08
Brazil C	ℓ	0	1	7	24	37	45
	h_ℓ	0.72	1.00	0.65	0.99	0.75	0.86
Brazil D	ℓ	2	10	36	49	94	95
	h_ℓ	0.99	0.65	0.74	0.86	1.00	0.72
Brazil E	ℓ	0	16	32	-	-	-
	h_ℓ	1.00	1.00	1.00	-	-	-

Essa tabela consiste na análise de cinco transmissões teste de sinal de TV Digital feitas, de 1999 a 2002, pela Universidade Presbiteriana do Mackenzie em diferentes cenários na cidade São Paulo. Desse modo, estamos determinando a resposta ao impulso do canal para cada transmissão, onde os canais vão de A até E, ℓ é a amostra e h_ℓ é a magnitude do eco com a fase abstraída por padrão [2], [7].

O canal Brazil A representa uma transmissão externa típica, com alta potência. O Brazil B representa transmissões regionais com potência menor. O Brazil C representa a transmissão para dispositivos móveis e tem potência mais fraca ainda. O Brazil D representa a transmissão interna. Por fim, o Brazil E é um canal mais usado para testes e experimentos.

Mas essa acima, na verdade, é a resposta ao impulso e não exatamente o *delay profile*. Para obtê-lo precisamos definir o intervalo de duração do símbolo $T = 1/SR = 1/16MHz = 0.0625\mu s$ e multiplicar esse valor pelo número da amostra. É importante ressaltar que essa tabela depende do *SymbolRate* caso isso não esteja claro, já que a consequência disso são diferentes respostas para diferentes *SymbolRates*.

Assim, é possível analisar também em termos de BER, o quanto o *multipath* afeta o sistema. As duas Figuras abaixo mostram de que forma o eco aparece na constelação e a curva BERxEbNo, respectivamente.

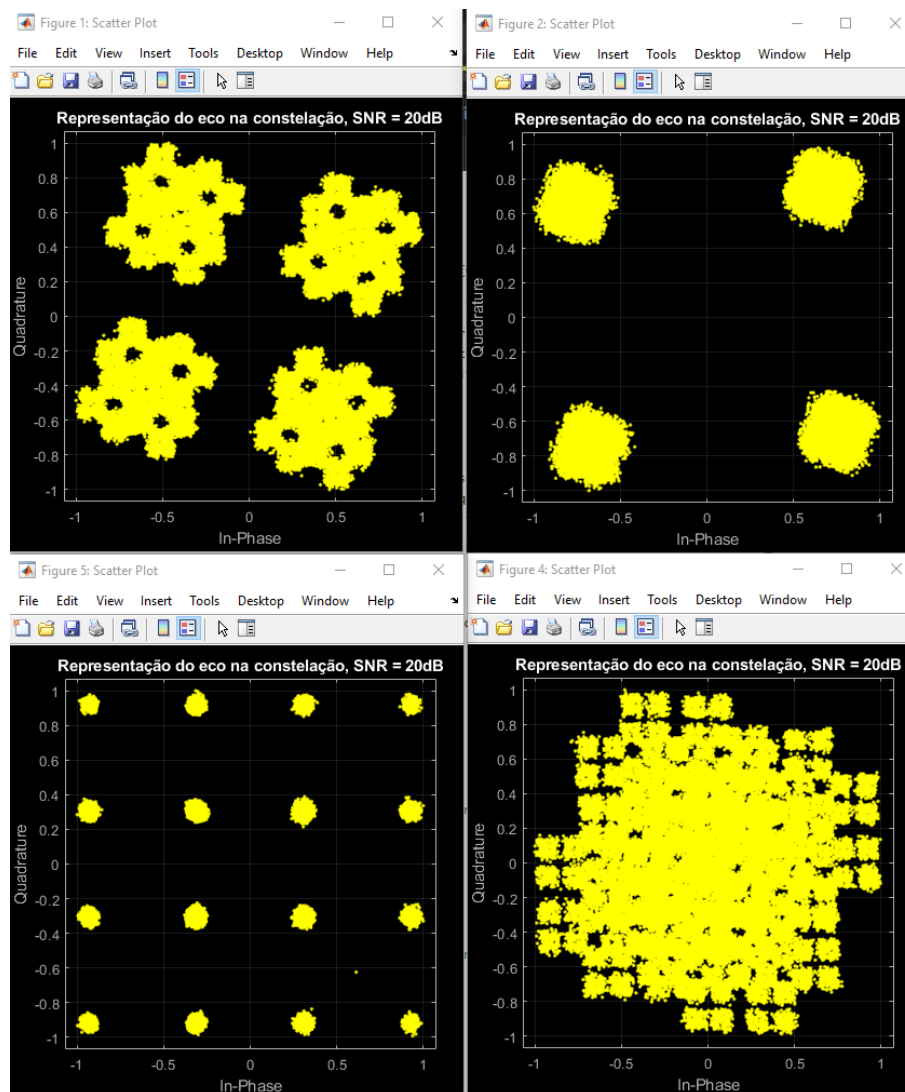


Figura 2.12 – Constelações com eco após demodulação, SNR = 20dB. Fonte: Autor

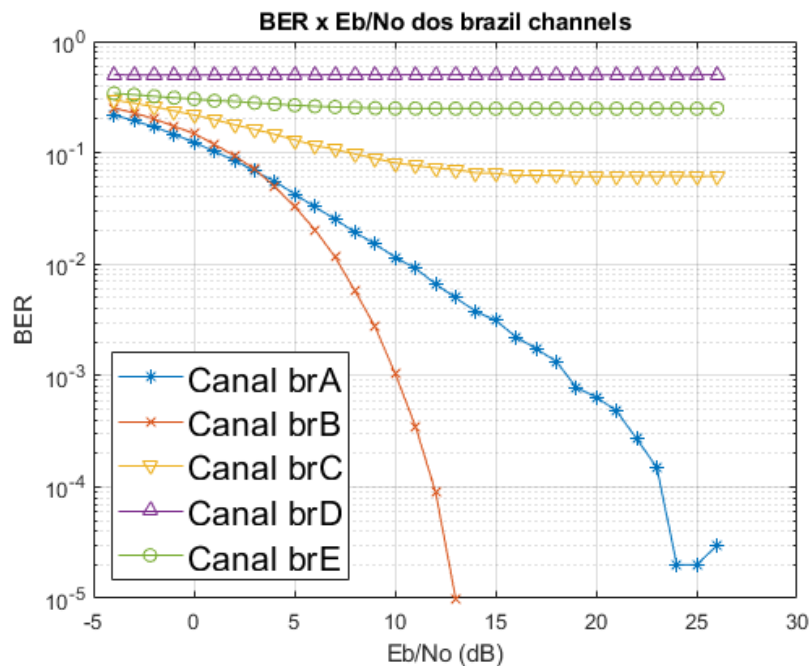


Figura 2.13 – BERxEbNo. Fonte: Autor

A Figura 2.12 mostra como a constelação demonstra o eco causado pelos *multipaths* após o *demapping*, em uma SNR de 20dB, para 4 dos canais. 'Figure 1' representa canal brA, 'Figure 2' representa canal brB e assim por diante. O canal brC não foi mostrado por razões de melhor visualização dos diagramas na mesma imagem e sua falta não prejudica a análise.

A Figura 2.13 mostra a BERxEbNo para cada uma das respostas ao impulso dos canais e assim é possível analisar que os canais brC, brD e brE sofrem com o eco ao ponto de que a BER apresenta resultados ruins e indesejados, enquanto os canais brA e brB conseguiram alcançar uma BER boa, mas somente quando a SNR era maior, apesar de que o canal brA demonstrou uma performance pior e inconstante no final. Podemos olhar novamente a 2.11, que alcança um resultado de BER equivalente no ponto de aproximadamente 9dB enquanto aqui os pontos foram aproximadamente 13 e 24dB para brB e brA, respectivamente. Assim, podemos ver que a consideração do canal no sistema faz com que sua performance piore, e a presença de um canal com muito eco pode fazer com que a performance seja insuficiente para uma transmissão confiável.

Então, ao comparar as duas Figuras acima, percebe-se que quanto mais "bagunçada" estiver a constelação, pior será o resultado em termos de performance.

2.3 Filtro Adaptativo Least Mean Squares (LMS)

A filtragem adaptativa é uma técnica que usa um algoritmo recursivo que a partir de algumas condições iniciais, atualiza seus coeficientes buscando se adaptar a um ambiente estacionário e converge para uma solução ótima. Se o ambiente não for estacionário, o algoritmo é capaz de rastrear variações desse ambiente.

Um exemplo clássico de filtro adaptativo é o LMS, um filtro adaptativo com aprendizado supervisionado baseado no método do Gradiente Descendente Estocástico [8]. De forma generalizada, o LMS pode ser representado pelas Equações a seguir.

$$y(n) = \hat{w}^T(n)\hat{r}(n) \quad (2.1)$$

$$e(n) = d(n) - y(n) \quad (2.2)$$

$$\hat{w}(n+1) = \hat{w}(n) + \mu e(n)\hat{r}^*(n) \quad (2.3)$$

onde $r(n)$ é o vetor de entrada (regressor), $d(n)$ é a resposta desejada, $\hat{w}(n)$ é a estimativa do vetor de coeficientes do filtro, $e(n)$ é o erro e μ é o passo de aprendizado.

Para o contexto da equalização LMS, o equalizador vai ajustar sua função de transferência buscando reduzir a ISI. A partir de uma sequência conhecida tanto no transmissor quanto no receptor, o equalizador passará por uma etapa de treinamento supervisionado no início da comunicação para que o algoritmo alcance a convergência, ou seja, um erro baixo (função de custo minimizada) [9].

2.4 Equalização de um Canal com Eco Usando um Filtro LMS

Utilizando o código da seção 5 do Apêndice A desse trabalho, um sistema com os mesmos parâmetros da simulação com cenários de *multipath* foi utilizado, mas agora com a adição do equalizador LMS e NLMS buscando melhorar os resultados dos canais que antes não obtiveram um bom resultado em termos de BER x EbNo. Os parâmetros do filtro são:

- Ordem $L = 10$ e 20 ;
- Passo adaptativo $\mu = 0.1$ que é multiplicado por 0.1 a cada 30% das amostras;
- 5% e 10% dos símbolos destinados para o treinamento supervisionado
- Curva de erro utilizando o *Mean Squared Error (MSE)*.

2.4.1 $L = 10$, 5% dos símbolos para treinamento

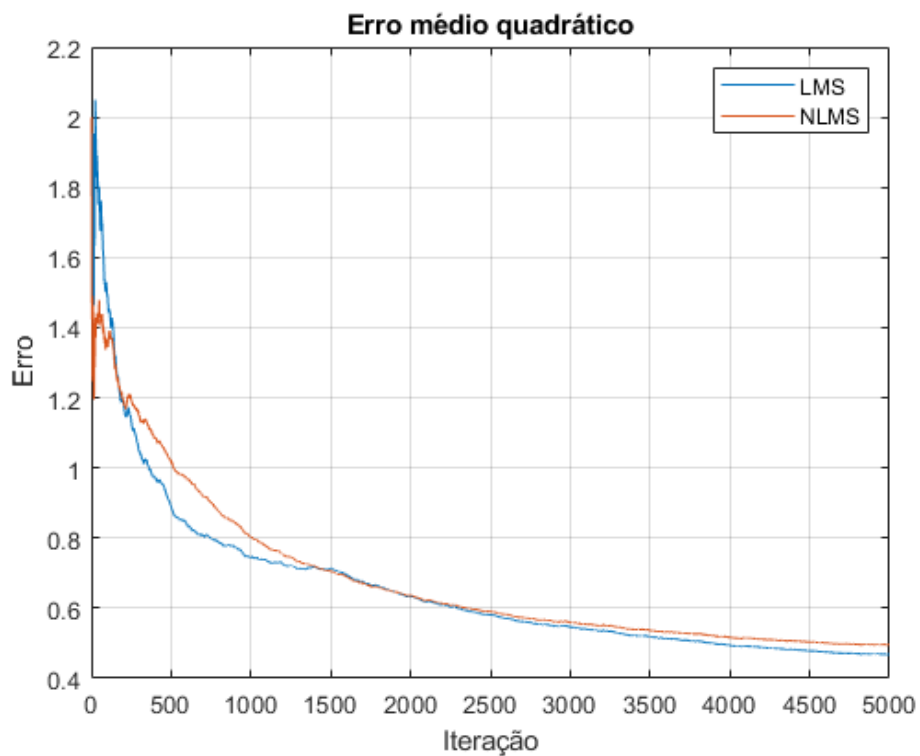


Figura 2.14 – MSE dos filtros LMS e NLMS. Fonte: Autor

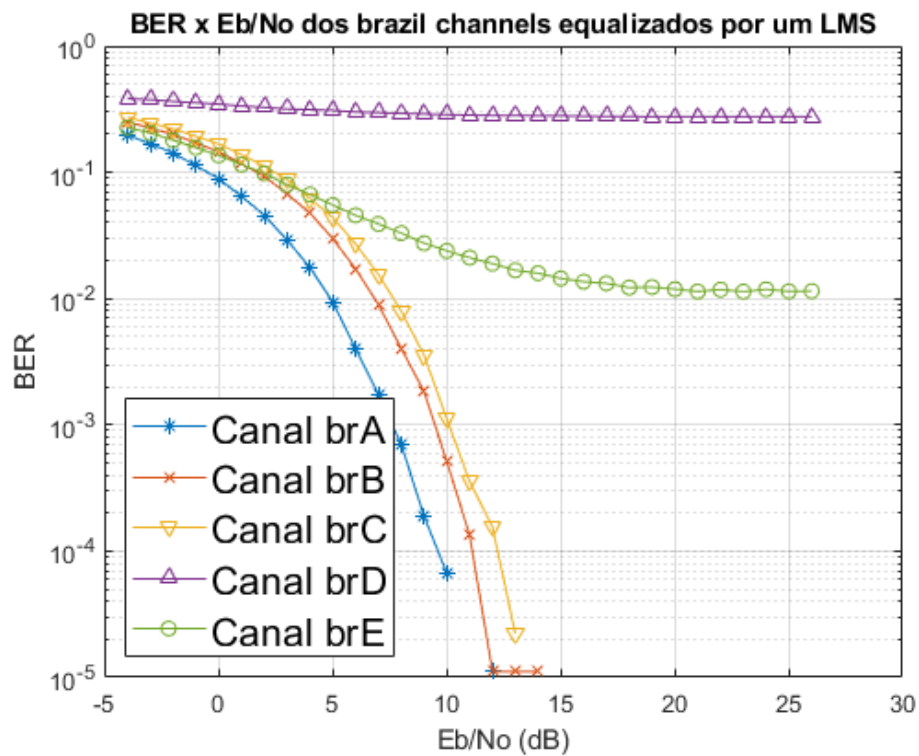


Figura 2.15 – BERx E_b/N_0 do LMS para $L = 10$ e 5% de símbolos de treinamento. Fonte: Autor

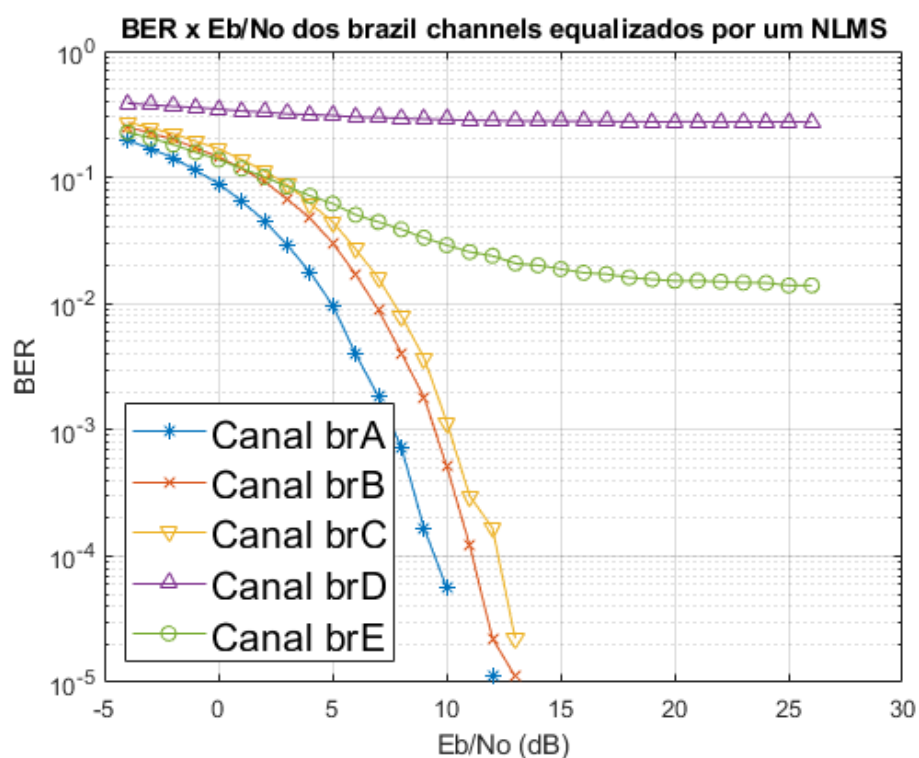


Figura 2.16 – BERx E_b/N_0 do LMS para $L = 10$ e 5% de símbolos de treinamento. Fonte: Autor

2.4.2 $L = 10$, 10% dos símbolos para treinamento

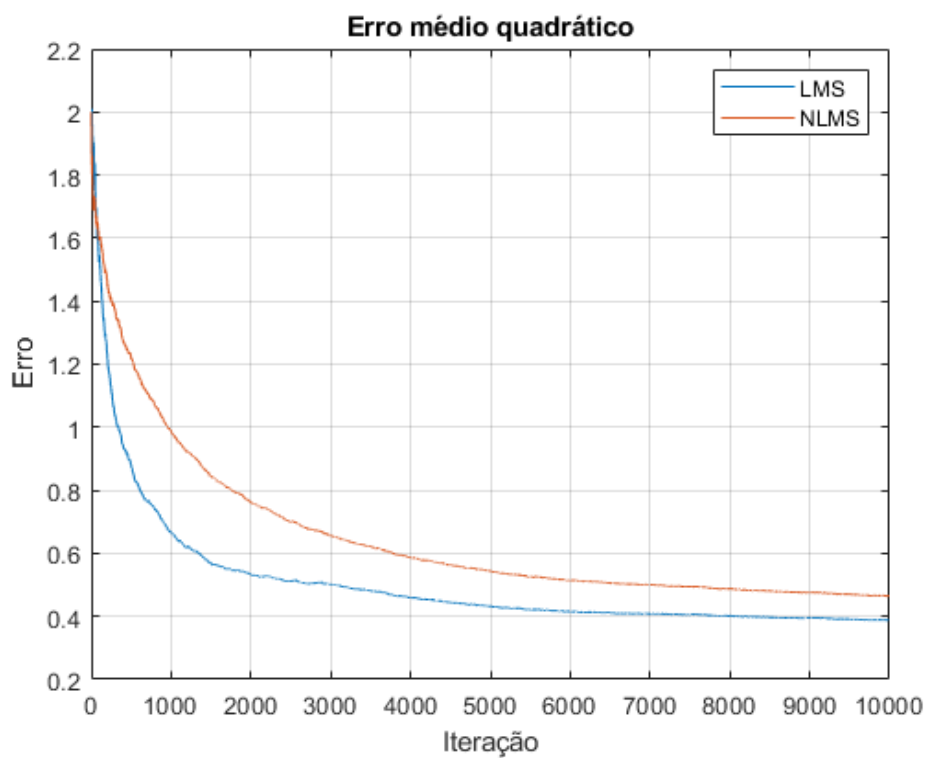


Figura 2.17 – MSE dos filtros LMS e NLMS. Fonte: Autor

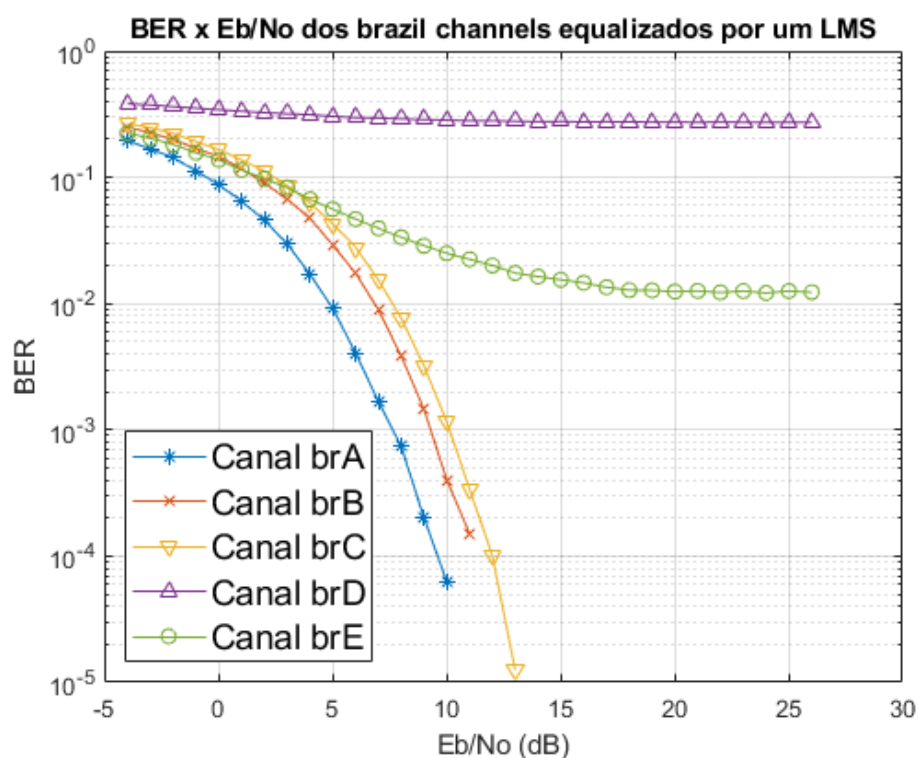


Figura 2.18 – BERx E_b/N_0 do LMS para $L = 10$ e 10% de símbolos de treinamento. Fonte: Autor

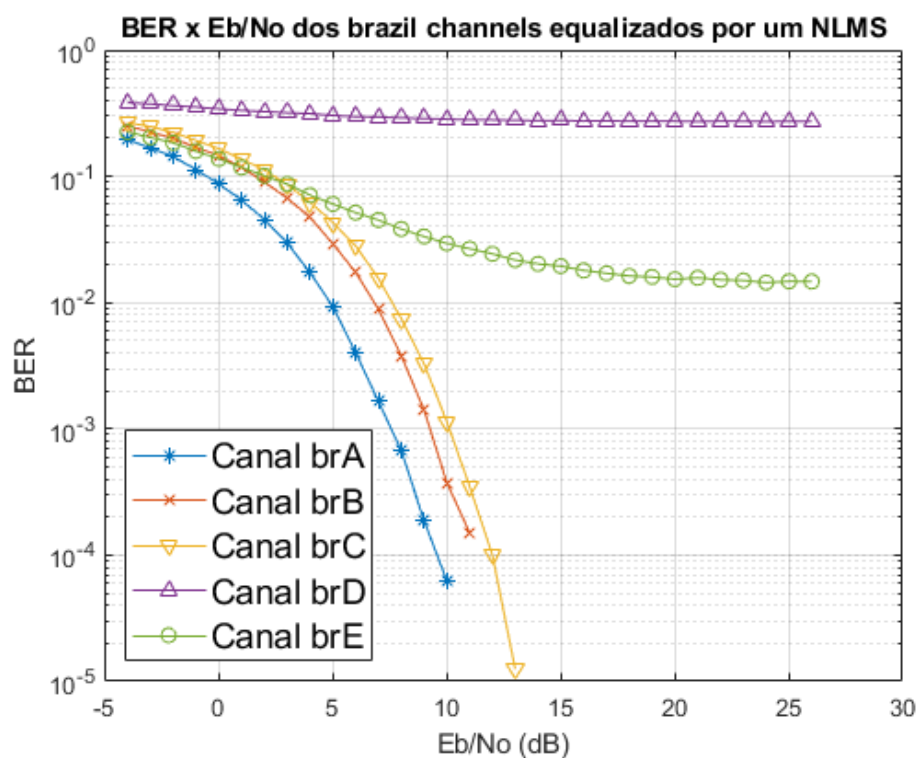


Figura 2.19 – BERx E_b/N_0 do NLMS para $L = 10$ e 10% de símbolos de treinamento. Fonte: Autor

2.4.3 $L = 20$, 10% dos símbolos para treinamento

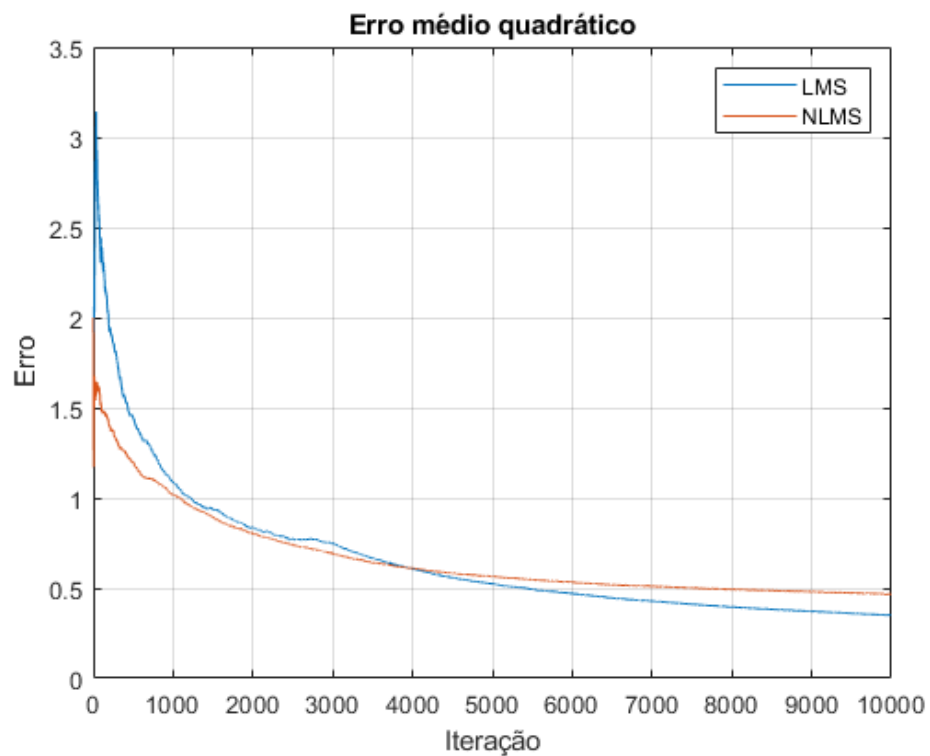


Figura 2.20 – MSE dos filtros LMS e NLMS. Fonte: Autor

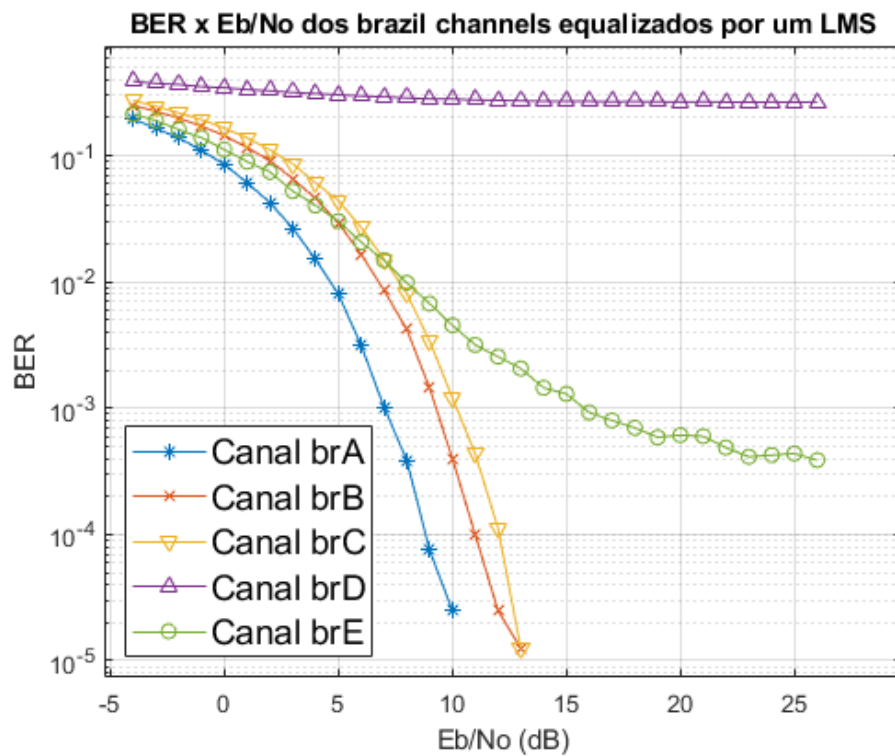


Figura 2.21 – BERx E_b/N_0 do LMS para $L = 10$ e 10% de símbolos de treinamento. Fonte: Autor

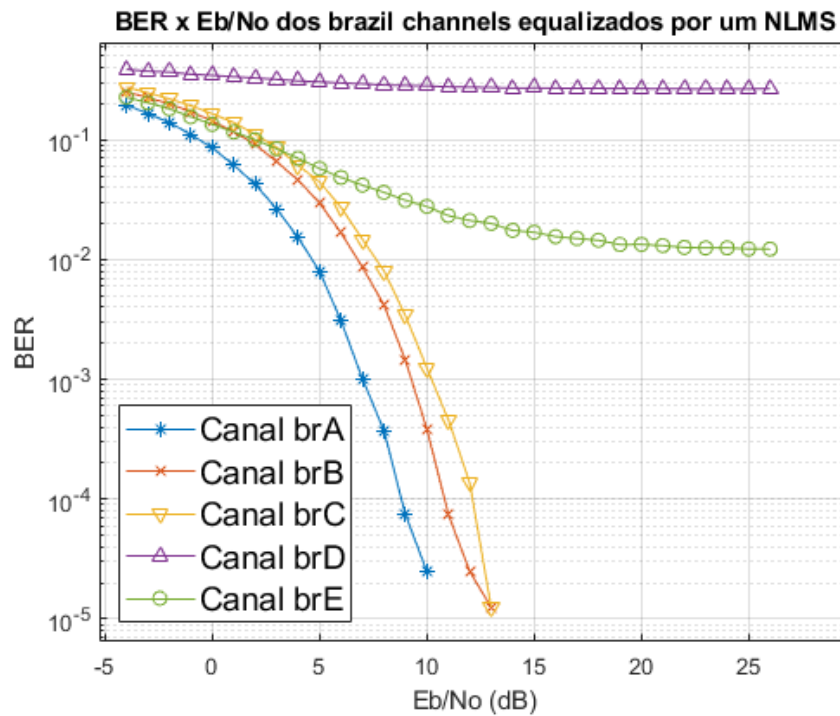


Figura 2.22 – BERx E_b/N_0 do NLMS para $L = 10$ e 10% de símbolos de treinamento. Fonte: Autor

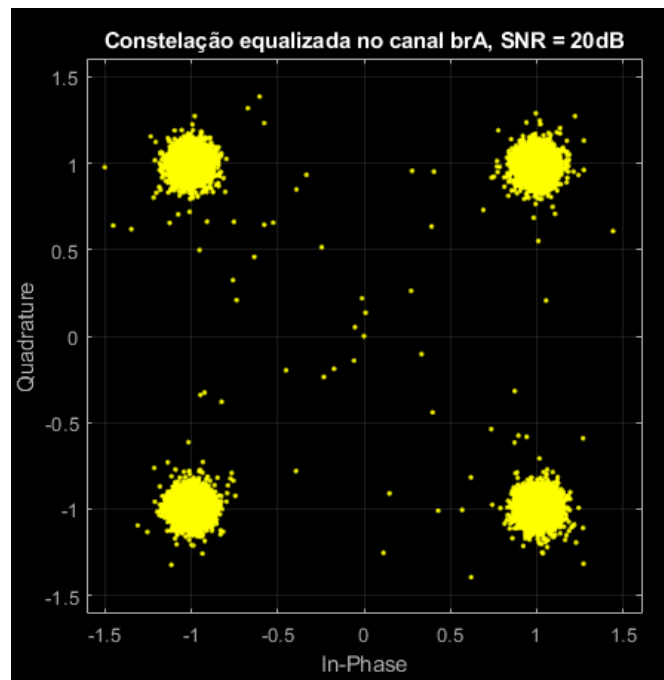


Figura 2.23 – Constelação equalizada, Canal brA, SNR = 20dB. Fonte: Autor

2.4.4 Análise e discussão dos resultados da equalização

Primeiramente, é mostrado na Figura 2.23 como a equalização mudou a constelação em um caso que a mesma foi um sucesso.

Depois, analisando as curvas de erro, nota-se que todas alcançaram valores bem próximos e que o NLMS convergiu mais lentamente que o LMS. Segundo [8], se fosse encontrando um passo de aprendizado ótimo, o NLMS convergiria mais rápido, coisa que não ocorreu.

Quanto a ordem do filtro, os resultados significativos e relevantes obtidos foram com $L=10$ ou 20 . A ordem em $L=10$ se mostrou suficiente para equalizar os canais brA, brB e brC, mas não os canais brD e brE. No caso do brA e brB, não esperada grande dificuldade, já que mesmo antes da equalização eles demonstravam resultados aceitáveis. O brC era menos problemático que brD e brE e também foi equalizado com sucesso. A mudança na porcentagem de símbolos de treinamento não teve um impacto tão grande, mas foi possível perceber que 10% gera resultados menos inconstantes e então foi mantido para $L=20$. Com $L=20$, os resultados anteriores para brA, brB, brC e brD se mantiveram, mas brE conseguiu uma melhora um pouco expressiva, apesar de ainda distante do ideal, considerando os canais que conseguiram ser equalizados.

Sobre outras mudanças de parâmetros: o aumento de L além de 20 resultou na incapacidade do filtro de convergir, gerando resultados inconsistentes; o aumento da porcentagem de símbolos no treino não mostrou mudanças significativas; o passo de aprendizado foi testado de 0.1 até 0.0001, sendo que em 0.1 e 0.0001 ele tinha problemas de aprendizado ruim e convergência lenta, respectivamente, ficando então como padrão o passo adaptativo.

O que se conclui disso é que o filtro LMS não foi suficiente para um problema de equalização em uma situação mais complexa, que exigiria outras abordagens também mais complexas. Apesar de ser uma técnica clássica e poderosa, o equalizador LMS encontra limitações.

3 CONCLUSÃO

Para a implementação desse sistema simplificado, um passo necessário para testar algo como a equalização de um canal com *multipath*, foi feito um estudo sobre o eco e seu efeito na recepção de um sinal numa comunicação sem fio. Depois disso, foi necessária uma revisão sobre comunicação digital e por fim o estudo pelo qual foi motivado o trabalho, o filtro adaptativo LMS.

Após esses estudos e revisões, considera-se que os resultados obtidos foram bem satisfatórios, até mesmo pela barreira encontrada no final da análise, quando o filtro LMS não foi capaz de equalizar um canal mais complexo, mostrando que para superar esse problema, seria necessária ainda mais pesquisa na busca de outras abordagens.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] E. Hari Krishna, M. Raghuram, K. Venu Madhav and K. Ashoka Reddy, "Acoustic echo cancellation using a computationally efficient transform domain LMS adaptive filter," 10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010), Kuala Lumpur, Malaysia, 2010, pp. 409-412, doi: 10.1109/ISSPA.2010.5605458.
- [2] http://www.fccdecastro.com.br/pdf/T2_Aulas16a20_27052020.pdf
- [3] Relatório da Disciplina Laboratório de Comunicações Digitais. "Experiência 05: Modulação M-QAM". Candice Muller.
- [4] https://shopdelta.eu/obrazki_art/dpsk_img3_d.jpg
- [5] <http://anengineerzdiary.blogspot.com/2014/01/baseband-transmission.html>
- [6] Relatório da Disciplina Laboratório de Comunicações Digitais. "Experiência 06: Demodulador M-QAM". Candice Muller.
- [7] Daniel L. Hatae, Marcio Eisencraft, Cristiano Akamine, Carlos E. S. Dantas, Magno T. M. Silva, and Maria D. Miranda, "OFDM Systems for Brazilian Digital Television channels" Mackenzie Presbyterian University Sao Paulo, Brazil.
- [8] S. Haykin, Adaptive Filter Theory, 5th ed., Pearson, London, 2013.
- [9] Relatório da Disciplina Laboratório de Comunicações Digitais. "Experiência 10: Equalização de Canal – Equalizador LMS". Candice Muller.

APÊNDICE A – CÓDIGOS DO MATLAB

A.1 Sistema com Canal Ideal

```
% SISTEMA DE COMUNICAO COM MODULAO QPSK PARA ANLISE
% DE EQUALIZAO COM FILTROS ADAPTATIVOS EM UM CANAL COM ECO DEVIDO AO
% CENRIO DE MULTIPATH USANDO OS DELAY PROFILES DOS BRAZIL CHANNELS
clear;
close all;
clc;

bs = randi([0 1], 1, 1e5); % Gerao aleatoria de bits
sr = 16e6;                % Taxa de smbolos
fc = 32e6;                % Frequncia da portadora
fs = 256e6;               % Frequncia de amostragem
SNR = -10:1:20;            % Relaes sinal-ruído a serem testadas
SNRs = length(SNR);       % Nmero de SNRs
BER = zeros(1,SNRs);      % Varivel da taxa de erro de bit
EbNo = zeros(1,SNRs);     % Varivel da relao EbNo

% Funo que faz a modulao e o processo necessrio para a transmissao do
% sinal
[Iog, Qog, Mod] = qpsk_mod(bs, sr, fs, fc);

% Varivel auxiliar
c = 1;

% Loop que varia a SNR de cada transmissao
for n=1:SNRs
    % Adio de ruído gaussiano branco
    nMod = awgn(Mod,SNR(n),'measured');

    % Incio do processo de demodulao
    [I, Q] = qpsk_demod(sr, fs, fc, nMod);

    % Demodulao QPSK por regio de deciso
    nn = 1;
    for k=1:length(I)
        if I(k) >= 0 && Q(k) >= 0
            demap(nn:nn+1) = [0 0];
        elseif I(k) < 0 && Q(k) >= 0
            demap(nn:nn+1) = [0 1];
        elseif I(k) < 0 && Q(k) < 0
            demap(nn:nn+1) = [1 1];
        elseif I(k) >= 0 && Q(k) < 0
            demap(nn:nn+1) = [1 0];
        end
        nn = nn + 2;
    end
end
```

```

% Taxa de Erro de Bit
BER(c) = (sum(xor(bs, demap)))/length(bs);

% Ratio of symbol energy to noise power spectral density (EsNo)
EsNo = 10*log10(0.5*(1/sr)/(1/fs)) + SNR(c);
% Ratio of bit energy to noise power spectral density (EbNo)
EbNo(c) = EsNo - 10*log10(2);

% Plot de constelao para alguns SNRs
if rem(c,10) == 0
scatterplot(I+1i*Q)
title(" Constelao para " + SNR(n+1) + " dB")
grid on
end

% Incremento de varivel auxiliar
c = c + 1;
end

figure
semilogy(EbNo,BER,'-*')
xlabel('Eb/No (dB)')
ylabel('BER')
title('Taxa de Erro de Bits vs Eb/No')
grid on

```

A.2 Função de modulação QPSK

```

% MODULAO QPSK
function [Iog, Qog, un] = qpsk_mod(bs, sr, fs, fc)

%% BITS -> SMBOLOS IQ
% QPSK usa 2 bits por smbolo
Iog = zeros(1,length(bs)/2);
Qog = zeros(1,length(bs)/2);
nn = 1;
% Mapeia bits em smbolos IQ
for n=1:2:length(bs)
    if bs(n:n+1) == [0 0]
        Iog(nn) = +1;
        Qog(nn) = +1;
    elseif bs(n:n+1) == [0 1]
        Iog(nn) = -1;
        Qog(nn) = +1;
    elseif bs(n:n+1) == [1 0]
        Iog(nn) = +1;
        Qog(nn) = -1;
    elseif bs(n:n+1) == [1 1]

```



```

        Iog(nn) = -1;
        Qog(nn) = -1;
    end
    nn = nn +1;
end

% scatterplot(Iog+1i*Qog)
% title(' Constelao  com os smbolos IQ originais')
% grid on

%% Upsampling
% Aumenta a taxa de amostragem dos sinais IQ
k = fs/sr; % fator de upsampling

upI = [];
upQ = [];
% Adiciona k-1 zeros a cada smbolo
for n=1:length(Iog)
    upI = [upI Iog(n) zeros(1,k-1)];
    upQ = [upQ Qog(n) zeros(1,k-1)];
end

% figure
% subplot(2,1,1)
% stem(upI)
% title('Smbolos I upsampled no modulador')
% xlabel('Amostra')
% ylabel('Amplitude')
% grid on
% subplot(2,1,2)
% stem(upQ)
% title('Smbolos Q upsampled no modulador')
% xlabel('Amostra')
% ylabel('Amplitude')
% grid on

%% LPF (FIR Gaussiano)
% Nesse passo, o filtro FIR LPF Gaussiano busca suavizar transies
% abruptas

% Coeficientes do filtro - funo de transferencia
hn = [0.0030 0.0118 0.0390 0.1051 0.2367 0.4445 0.6972 0.9138 1.0000 0.9138
      0.6972 0.4445 0.2367 0.1051 0.0390 0.0118];

% Aplica o filtro
lpfI = filter(hn,1,upI);
lpfQ = filter(hn,1,upQ);

% figure
% subplot(2,1,1)
% stem(lpfI)

```

```

% title('Smbolos I filtrados no modulador')
% xlabel('Amostra')
% ylabel('Amplitude')
% grid on
% subplot(2,1,2)
% stem(lpfQ)
% title('Smbolos Q filtrados no modulador')
% xlabel('Amostra')
% ylabel('Amplitude')
% grid on

%% Heterodinao : banda-base -> banda passante
% Faz a translaao da banda-base para banda passante (cuja frequencia
% central a da portadora)
pb = 0:length(lpfI)-1;
XI = cos(2*pi*fc*pb/fs);
XQ = sin(2*pi*fc*pb/fs);
ui = lpfI.*XI;
uq = lpfQ.*XQ;
un = ui - uq;

% figure
% scatter(ui,uq)
% title('Sinal em banda passante')
% grid on

end

```

A.3 Função de demodulação QPSK

```

% DEMODULAO QPSK (sem a parte do demapper)
function [I, Q] = qpsk_demod(sr, fs, fc, rx)

%% Downconversion
% Retorna o sinal para a banda-base
bb = 0:length(rx)-1;
XI = cos(2*pi*fc*bb/fs);
XQ = sin(2*pi*fc*bb/fs);
uir = rx.*XI;
uqr = rx.*-XQ;

%% LPF (FIR Gaussiano)
% Aqui, o filtro responsvel por eliminar as rplicas indesejveis
% geradas pela heterodinao, mantendo apenas o espectro centrado em zero
hn = [0.0030 0.0118 0.0390 0.1051 0.2367 0.4445 0.6972 0.9138 ...
      1.0000 0.9138 0.6972 0.4445 0.2367 0.1051 0.0390 0.0118];

k = fs/sr;
lpfI = conv(uir,hn,'same');

```

```

lpfQ = conv(uqr,hn,'same');

% figure
% subplot(2,1,1)
% stem(lpfI)
% title('Smbolos I filtrados no demodulador')
% grid on
% subplot(2,1,2)
% stem(lpfQ)
% title('Smbolos Q filtrados no demodulador')
% grid on

%% Downsampling
% Para fazer o downsampling, busca-se o melhor instante de amostragem e faz
% um chaveamento mantendo somente esses instantes

% Por inspeção da variável, o instante típico de amostragem é 9, 25, 41, ...
ost = 9;
for n=0:(length(lpfI)/k)-1
    downI(n+1) = lpfI((n*k)+ost);
    downQ(n+1) = lpfQ((n*k)+ost);
end

% figure
% subplot (2,1,1)
% stem(downI)
% title ('Smbolos I aps downsampling');
% grid on
% subplot (2,1,2)
% stem(downQ)
% title ('Smbolos Q aps downsampling');
% grid on

%% Automatic Gain Control
% Normalização [-1,1] a partir de constante considerando ruído e
% multipercursos
I = downI/max(abs(downI));
Q = downQ/max(abs(downQ));

% figure
% scatter(I,Q)
% grid on

end

```

A.4 Sistema com Canal em Cenário com Multipath

```

% SISTEMA DE COMUNICAÇÃO COM MODULAÇÃO QPSK PARA ANÁLISE
% DE EQUALIZAÇÃO COM FILTROS ADAPTATIVOS EM UM CANAL COM ECO DEVIDO AO

```

```

% CENRIO DE MULTIPATH
clear;
close all;
clc;

bs = randi([0 1], 1, 1e5); % Gerao aleatoria de bits
sr = 16e6;                % Taxa de smbolos
fc = 32e6;                % Frequncia da portadora
fs = 256e6;               % Frequncia de amostragem
SNR = -10:1:20;           % Relaes sinal-rudo a serem testadas
SNRs = length(SNR);       % Nmero de SNRs
BER = zeros(1,5);         % Varivel da taxa de erro de bit
EbNo = zeros(1,5);        % Varivel da relao EbN0

% Delay profiles do BRAZIL CHANNELS, canais reais com multipath
brA = [1 0 0.2 zeros(1,36-2-1) 0.15 zeros(1,49-36-1) 0.18 zeros(1,94-49-1) 0.21
0.15];
brB = [1 0 0 0 0 0.25 zeros(1,56-5-1) 0.63 zeros(1,70-56-1) 0.45
zeros(1,152-70-1) 0.18 zeros(1,203-152-1) 0.08];
brC = [0.72 1 0 0 0 0 0.65 zeros(1,24-7-1) 0.99 zeros(1,37-24-1) 0.75
zeros(1,45-37-1) 0.86];
brD = [0 0 0.99 zeros(1,10-2-1) 0.65 zeros(1,36-10-1) 0.74 zeros(1,49-36-1)
0.86 zeros(1,94-49-1) 1 0.72];
brE = [1 zeros(1,16-1) 1 zeros(1,32-16-1) 1];
dp = cell(5,1);
dp{1,1} = brA;
dp{2,1} = brB;
dp{3,1} = brC;
dp{4,1} = brD;
dp{5,1} = brE;

% Funo que faz a modulao e o processo necessrio para a transmissao do
% sinal
[Iog, Qog, Mod] = qpsk_mod(bs, sr, fs, fc);

% Varivel auxiliar
c = 1;

% Loop que usa cada um dos canais a uma SNR n
for n=1:SNRs
    % Adio de rudo gaussiano branco
    nMod = awgn(Mod,SNR(n),'measured');

    for dpn=1:size(dp,1)
        % Sinal recebido distorcido pelo canal
        rx = filter(dp{dpn,1},1,nMod);

        % Incio do processo de demodulao
        [I, Q] = qpsk_demod(sr, fs, fc, rx);

        % Demodulao QPSK por regio de deciso

```

```

nn = 1;
for k=1:length(I)
    if I(k) >= 0 && Q(k) >= 0
        demap(nn:nn+1) = [0 0];
    elseif I(k) < 0 && Q(k) >= 0
        demap(nn:nn+1) = [0 1];
    elseif I(k) < 0 && Q(k) < 0
        demap(nn:nn+1) = [1 1];
    elseif I(k) >= 0 && Q(k) < 0
        demap(nn:nn+1) = [1 0];
    end
    nn = nn + 2;
end

% Taxa de Erro de Bit
BER(c,dpn) = (sum(xor(bs, demap)))/length(bs);

% Ratio of symbol energy to noise power spectral density (EsNo)
EsNo = 10*log10(0.5*(1/sr)/(1/fs)) + SNR(c);
% Ratio of bit energy to noise power spectral density (EbNo)
EbNo(c,dpn) = EsNo - 10*log10(2);

if SNR(n) == 20
    scatterplot(I+1i*Q)
    title('Representao do eco na constelao , SNR = 20dB')
    grid on
end
% Incremento de varivel auxiliar
end
c = c + 1;
end

figure
semilogy(EbNo(:,1),BER(:,1),'-*')
hold on
semilogy(EbNo(:,2),BER(:,2),'-x')
semilogy(EbNo(:,3),BER(:,3),'-v')
semilogy(EbNo(:,4),BER(:,4),'-^')
semilogy(EbNo(:,5),BER(:,5),'-o')
xlabel('Eb/No (dB)')
ylabel('BER')
legend('Canal brA','Canal brB','Canal brC','Canal brD','Canal
brE','Location','southwest','FontSize',16)
title('BER x Eb/No dos brazil channels')
grid on

```

A.5 Sistem com Canal em Cenário com Multipath e Equalização LMS

```

% SISTEMA DE COMUNICAO COM MODULAO QPSK PARA ANLISE

```

```

% DE EQUALIZAO COM FILTROS ADAPTATIVOS EM UM CANAL COM ECO DEVIDO AO
% CENRIO DE MULTIPATH
clear;
close all;
clc;
tic
bs = randi([0 1], 1, 1e5); % Gerao aleatria de bits
sr = 16e6;                % Taxa de smbolos
fc = 32e6;                % Frequncia da portadora
fs = 256e6;               % Frequncia de amostragem
SNR = -10:1:20;           % Relaes sinal-ruído a serem testadas
SNRs = length(SNR);      % Nmero de SNRs
BER = zeros(1,5);        % Varivel da taxa de erro de bit
BERn = zeros(1,5);       % Varivel da taxa de erro de bit
EbNo = zeros(1,5);       % Varivel da relao EbNo

% Delay profiles do BRAZIL CHANNELS, canais reais com multipath
brA = [1 0 0.2 zeros(1,36-2-1) 0.15 zeros(1,49-36-1) 0.18 zeros(1,94-49-1) 0.21
0.15];
brB = [1 0 0 0 0 0.25 zeros(1,56-5-1) 0.63 zeros(1,70-56-1) 0.45
zeros(1,152-70-1) 0.18 zeros(1,203-152-1) 0.08];
brC = [0.72 1 0 0 0 0 0 0.65 zeros(1,24-7-1) 0.99 zeros(1,37-24-1) 0.75
zeros(1,45-37-1) 0.86];
brD = [0 0 0.99 zeros(1,10-2-1) 0.65 zeros(1,36-10-1) 0.74 zeros(1,49-36-1)
0.86 zeros(1,94-49-1) 1 0.72];
brE = [1 zeros(1,16-1) 1 zeros(1,32-16-1) 1];
dp = cell(5,1);
dp{1,1} = brA;
dp{2,1} = brB;
dp{3,1} = brC;
dp{4,1} = brD;
dp{5,1} = brE;

% Funo que faz a modulao e o processo necessrio para a transmissao do
% sinal
[log, qog, mod] = qpsk_mod(bs, sr, fs, fc);

% Varivel auxiliar
c = 1;

% Loop que usa cada um dos canais a uma SNR n
for n=1:SNRs
    % Adio de ruído gaussiano branco
    nMod = awgn(mod,SNR(n),'measured');

    for dpn=1:size(dp,1)
        % Sinal recebido distorcido pelo canal
        rx = filter(dp{dpn,1},1,nMod);

        % Incio do processo de demodulao
        [I, Q] = qpsk_demod(sr, fs, fc, rx);
    end
end

```

```

% Equalizao do canal com um filtro LMS
L = 20; % Ordem do filtro
N = length(bs)*0.1; % Nmero de bits para o treinamento
step = 0.1; % Passo de aprendizagem
[y, mse] = lms_filter(Iog, Qog, I, Q, step, L, N);
[yn, nmse] = nlms_filter(Iog, Qog, I, Q, step, L, N);

%% Demapp com o sinal filtrado pelo LMS
% Demapper - pega os smbolos IQ recebidos e divide a constelao em
% regies de deciso para tentar recuperar os bits transmitidos
nn = 1;
for k=1:length(y)
    if real(y(k)) >= 0 && imag(y(k)) >= 0
        demap(nn:nn+1) = [0 0];
    elseif real(y(k)) < 0 && imag(y(k)) >= 0
        demap(nn:nn+1) = [0 1];
    elseif real(y(k)) < 0 && imag(y(k)) < 0
        demap(nn:nn+1) = [1 1];
    elseif real(y(k)) >= 0 && imag(y(k)) < 0
        demap(nn:nn+1) = [1 0];
    end
    nn = nn + 2;
end

% Taxa de Erro de Bit
BER(c,dpn) = sum(xor(bs(N*2:end), demap(N*2:end)))/length(bs(N*2:end));

%% Demapp com o sinal filtrado pelo NLMS
% Demapper - pega os smbolos IQ recebidos e divide a constelao em
% regies de deciso para tentar recuperar os bits transmitidos
nn = 1;
for k=1:length(yn)
    if real(yn(k)) >= 0 && imag(yn(k)) >= 0
        demapn(nn:nn+1) = [0 0];
    elseif real(yn(k)) < 0 && imag(yn(k)) >= 0
        demapn(nn:nn+1) = [0 1];
    elseif real(yn(k)) < 0 && imag(yn(k)) < 0
        demapn(nn:nn+1) = [1 1];
    elseif real(yn(k)) >= 0 && imag(yn(k)) < 0
        demapn(nn:nn+1) = [1 0];
    end
    nn = nn + 2;
end

% Taxa de Erro de Bit
BERn(c,dpn) = sum(xor(bs(N*2:end), demapn(N*2:end)))/length(bs(N*2:end));

%% Razo energia de smbolo por densidade espectral de potncia de rudo e
% Razo energia de bit por densidade espectral de potncia de rudo
% Ratio of symbol energy to noise power spectral density (EsN0)

```

```

EsNo = 10*log10(0.5*(1/sr)/(1/fs)) + SNR(c);
% Ratio of bit energy to noise power spectral density (EbN0)
EbNo(c,dpn) = EsNo - 10*log10(2);

if dpn == 1
if SNR(n) == 20
scatterplot(y)
title(' Representao do eco na constelao , SNR = 20dB')
grid on
end
end

end
% Incremento de varivel auxiliar
c = c + 1;
end

figure
semilogy(EbNo(:,1),BER(:,1),'-*')
hold on
semilogy(EbNo(:,2),BER(:,2),'-x')
semilogy(EbNo(:,3),BER(:,3),'-v')
semilogy(EbNo(:,4),BER(:,4),'-^')
semilogy(EbNo(:,5),BER(:,5),'-o')
xlabel('Eb/No (dB)')
ylabel('BER')
legend('Canal brA','Canal brB','Canal brC','Canal brD','Canal
brE','Location','southwest','FontSize',16)
title('BER x Eb/No dos brazil channels equalizados por um LMS')
grid on

figure
semilogy(EbNo(:,1),BERn(:,1),'-*')
hold on
semilogy(EbNo(:,2),BERn(:,2),'-x')
semilogy(EbNo(:,3),BERn(:,3),'-v')
semilogy(EbNo(:,4),BERn(:,4),'-^')
semilogy(EbNo(:,5),BERn(:,5),'-o')
xlabel('Eb/No (dB)')
ylabel('BER')
legend('Canal brA','Canal brB','Canal brC','Canal brD','Canal
brE','Location','southwest','FontSize',16)
title('BER x Eb/No dos brazil channels equalizados por um NLMS')
grid on

figure
plot((mse))
hold on
plot((nmse))
hold off
title('Erro mdio quadratico ')

```



```

xlabel(' Iterao ')
ylabel('Erro')
legend('LMS','NLMS')
grid on
toc

```

A.6 Função do Equalizador LMS

```

% FILTRO ADAPATATIVO LMS
function [y, mse] = lms(Iog, Qog, I, Q, step, L, N) %#ok<FNDEF>
w = zeros(1,L); % Vetor de coeficientes do filtro
rk = zeros(1,L); % Vetor regressor
rxIQ = I + 1i*Q; % Smbolos IQ no receptor
d = Iog + 1i*Qog; % Smbolos IQ originais

% Implementao do filtro LMS utilizando o vetor regressor rk para fazer
% uma fila FIFO e e percorrer todos vetor aplicando o treinamento
% supervisionado obtendo uma estimativa e um MSE
% Adaptao do passo de aprendizagem
opt = 0.3*N;
for n=1:N
    rk(2:L) = rk(1:L-1);
    rk(1) = rxIQ(n);
    y(n) = rk*w.';
    e(n) = d(n) - y(n);
    w = w + step*e(n)*conj(rk);
    mse(n) = (sum((abs(e)).^2))/n;
    if n == opt
        step = step*0.1;
        opt = opt*2;
    end
end
% Aplicao da estimativa ( validao )
for n=(N+1):length(d)
    rk(2:L) = rk(1:L-1);
    rk(1) = rxIQ(n);
    y(n) = rk*w.';
end

% figure
% plot(mse)
% title('Erro mdio quadrático ')
% xlabel(' Iterao ')
% ylabel('Erro')
% grid on

end

```

A.7 Função do Equalizador NLMS

```
% FILTRO ADAPATATIVO NLMS
function [y, mse] = lms(Iog, Qog, I, Q, step, L, N) %#ok<FNDEF>
w = zeros(1,L); % Vetor de coeficientes do filtro
rk = zeros(1,L); % Vetor regressor
rxIQ = I + 1i*Q; % Smbolos IQ no receptor
d = Iog + 1i*Qog; % Smbolos IQ originais

% Implementao do filtro LMS utilizando o vetor regressor rk para fazer
% uma fila FIFO e e percorrer todos vetor aplicando o treinamento
% supervisionado obtendo uma estimativa e um MSE
% Adaptao do passo de aprendizagem
opt = 0.3*N;
for n=1:N
    rk(2:L) = rk(1:L-1);
    rk(1) = rxIQ(n);
    y(n) = rk*w.';
    e(n) = d(n) - y(n);
    w = w + step*e(n)*conj(rk)/norm(rk)^2;
    mse(n) = (sum((abs(e)).^2))/n;
    if n == opt
        step = step*0.1;
        opt = opt*2;
    end
end
% rk = abs(rk).^2;
% Aplicao da estimativa ( validao )
for n=(N+1):length(d)
    rk(2:L) = rk(1:L-1);
    rk(1) = rxIQ(n);
    y(n) = rk*w.';
    % y(n) = rk*w.'*norm(rk(1))^2;
end

end
```
