

Mateus Matos Rizzi

# **Método de Programação Matemática Para Equalizadores**

Santo André

2019



Mateus Matos Rizzi

## **Método de Programação Matemática Para Equalizadores**

Trabalho de graduação apresentado à Universidade Federal do ABC de Santo André como requisito parcial para a obtenção de grau de bacharel em Engenharia da Informação

Universidade Federal do ABC – UFABC

Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas da UFABC – CECS

Programa de Graduação

Orientador: Prof. Dr. André Kazuo Takahata

Santo André

2019

Mateus Matos Rizzi

Método de Programação Matemática Para Equalizadores/ Mateus Matos Rizzi. – Santo André, 2019-

48 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. André Kazuo Takahata

Trabalho de graduação (Graduação) – Universidade Federal do ABC – UFABC  
Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas da UFABC – CECS  
Programa de Graduação, 2019.

1. Programação Matemática. 2. Programação Linear. 3. Equalização não Supervisionada. I. Andre Kazuo Takahata. II. Universidade Federal do abc. III. Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas da UFABC IV. Método de Programação Matemática para Equalização Não supervisionada

Mateus Matos Rizzi

## **Método de Programação Matemática Para Equalizadores**

Trabalho de graduação apresentado à Universidade Federal do ABC de Santo André como requisito parcial para a obtenção de grau de bacharel em Engenharia da Informação

Trabalho aprovado. Santo André, 12 de Abril de 2019:

---

**Prof. Dr. André Kazuo Takahata**  
Orientador

---

**Prof. Dr. Ricardo Suyama**

---

**Prof. Dra. Aline de Oliveira Neves**

Santo André  
2019



# Agradecimentos

Agradeço em destaque aos meus pais, por sempre me darem apoio e proporcionar um ambiente onde eu pudesse crescer e me desenvolver como pessoa. Permitir um ambiente acolhedor em casa quando eu precisasse e acreditar em mim, mesmo quando eu não acreditasse.

Agradeço a minha família em geral, por sempre acreditarem em mim.

Agradeço a minha namorada por ser compreensiva e acolhedora durante a jornada que foi realizar esse trabalho.

Serei eternamente grato ao meu orientador Dr. prof. Andre Kazuo por ter paciência e me ajudar a desenvolver este trabalho ao qual apresento hoje. Sem a sua compreensão em me permitir errar, entender meu tempo de aprendizagem e principalmente, por me acordar na reta final para que eu pudesse terminar o trabalho.



*“Não importa o quanto você bate,  
mas o quanto aguenta apanhar e continuar.  
O quanto pode suportar e seguir em frente.  
É assim que se ganha.  
(Silvester Stallone -Rocky Balboa V)*



# Resumo

Em sistemas de comunicações, os sinais são transmitidos através de canais de comunicação que, idealmente, deveriam transportar os dados sem causar distorção alguma. Porém, em sistemas reais, existem limitações que interferem neste processo causando degradação nas informações transmitidas e podendo comprometer sua recepção. Tais limitações ocorrem devido à presença de ruído aditivo, e principalmente por interferência intersimbólica. A equalização de canal é uma das técnicas existentes que reduz os efeitos da interferência intersimbólica, dando maior confiabilidade e robustez aos sistemas de comunicações.

Este trabalho tem como objetivo validar o comportamento de algoritmos adaptativos cegos, assim denominados por não utilizarem sequências de treinamento, usando uma abordagem baseada em programação matemática, e métodos com equalizadores supervisionados, que utilizam sinais de treinamento para equalizar o sinal, e buscam encontrar compensador para retirar a degradação no sinal do canal e recuperar o sinal original.

Os métodos de equalização que foram escolhidos a serem implementados neste trabalho foram o filtro de Wiener, que é um método para equalização supervisionada, e o método de Ding, que é um método de equalização não supervisionada.

Os testes foram executados no software Matlab, utilizando um pacote de resolução de programação linear chamado Sedumi. Os resultados obtidos mostraram que método de programação matemática equaliza bem sinais tanto de fase mínima e fase não mínima, porém, utiliza muito custo computacional.

**Palavras-chave:** Programação Matemática. Programação Linear. Equalizador Não supervisionado.



# Abstract

In communication systems, signals are transmitted through communication channels, and ideally they should be transmitted in such a way that they do not suffer any distortion. However, in real systems, limitations interfere in the process causing degradation on the transmitted information. That limitation occurs mainly due to the presence of additive noise and mostly by intersymbolic interference. The channel equalizer is one of the techniques that can mitigate the effects of intersymbolic interference, giving more reliability and safety in communication systems.

This work has the objective of testing the behavior of a blind adaptive algorithm, which does not use a training sequence, using solutions based on linear programming for supervised equalizer, that use practice signs for equalize the channel, searching a pattern for the noising problem.

The filter chosen to be reproduced in this job are filter of Wiener, that is a supervised equalizer, and Ding method, that is blind equalizer.

Tests are executed in Matlab software, using the resolution of linear programming calling Sedumi.

**Keywords:** Equalization . Linear Programming . Blind Equalizer.



# Lista de ilustrações

Figura 1 – Transmissão de um equalizador supervisionado (NEVES et al., 2001) . . . . .	24
Figura 2 – Modulação 4-QAM . . . . .	37
Figura 3 – Modulação 16-QAM . . . . .	38
Figura 4 – Sinal depois do canal de fase mínima . . . . .	38
Figura 5 – Sinal depois do canal de fase não mínima . . . . .	39
Figura 6 – Filtro de Wiener em um canal de fase mínima em modulação 4-QAM . . . . .	39
Figura 7 – Filtro de Wiener em um canal de fase mínima em modulação 16-QAM . . . . .	40
Figura 8 – Filtro de Wiener em um canal de fase não mínima em modulação 4-QAM . . . . .	40
Figura 9 – Filtro de Wiener em um canal de fase não mínima em modulação 16-QAM . . . . .	41
Figura 10 – Programação Linear em uma modulação 4-QAM em canal de fase mínima . . . . .	41
Figura 11 – Programação Linear em uma modulação 16-QAM em canal de fase mínima . . . . .	42
Figura 12 – Programação Linear em uma modulação 4-QAM e canal de fase não mínima . . . . .	42
Figura 13 – Programação Linear em uma modulação 16-QAM e canal de fase não mínima . . . . .	43



# Lista de abreviaturas e siglas

IIS	Interferência Intersimbólica
MCT	Ministério de ciência e tecnologia
FIR	Finite impulse response
IIR	Infinite impulse response
EQM	Erro quadrático médio
ZF	Zero forcing
EOS	Estatística de ordem superior
MMSE	Sinal de erro
CMA	Algoritmo de módulos de constante
QAM	Quadrature Amplitude Modulation



# Lista de símbolos

$\Theta$	Equalizador
$\theta$	Coefficiente do Equalizador
$H$	Canal
$A$	Mensagem transmitida
$a$	Coefficientes da mensagem transmitida
$X$	Mensagem recebida
$x$	Coefficiente da Mensagem recebida
$v$	Delay
$y$	Mensagem recuperada
$d$	Atraso do sistema
$E$	Operador esperança
$A$	Matriz de input no Sedumi
$B$	Matriz de input no Sedumi
$c$	vetor de input no Sedumi
$R$	Número de restrições
$N$	Tamanho do número de amostras usadas para equalizar
$R_u$	Autocorrelação
$p_u$	Matriz de erro Filtro de Wiener



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>21</b>
<b>2</b>	<b>EQUALIZADORES</b>	<b>23</b>
<b>2.1</b>	<b>Equalizadores Supervisionados</b>	<b>24</b>
2.1.1	Filtro de Wiener	25
<b>2.2</b>	<b>Equalizadores Não Supervisionados</b>	<b>25</b>
2.2.1	Método de Ding	27
<b>2.3</b>	<b>Sedumi</b>	<b>29</b>
<b>3</b>	<b>METODOLOGIA</b>	<b>31</b>
<b>3.1</b>	<b>Modelagem do Filtro de Wiener</b>	<b>31</b>
<b>3.2</b>	<b>Modelagem do Método de Ding</b>	<b>31</b>
3.2.1	Implementação no Sedumi	32
<b>4</b>	<b>RESULTADOS</b>	<b>37</b>
<b>4.1</b>	<b>Filtro de Wiener</b>	<b>39</b>
<b>4.2</b>	<b>Método de Ding</b>	<b>41</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>45</b>
	<b>REFERÊNCIAS</b>	<b>47</b>



# 1 Introdução

Estar conectado em qualquer lugar do globo terrestre com a mesma qualidade de velocidade de recepção e transmissão se tornou uma questão de disponibilidade de serviço para companhias de tecnologia. Segundo levantamento do ministério de ciência e tecnologia (MCT), em 2008, foram investidos no Brasil 12,365 bilhões de reais em tecnologia da informação e comunicação, em comparação com 2018, onde foram investidos 105 bilhões de reais em tecnologia da informação e comunicação, um investimento dez vezes maior em um intervalo de 10 anos. Esse aumento significativo no investimento se dá pela corrida tecnológica em buscar transmissões cada vez mais robustas que usem cada vez menos energia, que transmitam o máximo de informação possível em uma largura de banda menor (BRASIL, 2018 (acessado 02 de Abril, 2019)).

O processo de transmissão de informação pode ser dividido em etapas de codificação do sinal, transmissão e recebimento da informação. A etapa de codificação busca transmitir a informação com o menor tamanho possível e de forma mais segura, como por exemplo, colocando criptografia. Na parte de transmissão, os desafios mais importantes estão dentre eles, a questão de segurança da informação, latência e custo operacional. A questão do recebimento da informação se preocupa na reconstrução do sinal recebido sem que haja perdas.

Em transmissão, a segurança da informação com criptografias cada vez mais robustas e com chaves pareadas tem recebidos cada vez mais atenção e investimento. Novas modulações são propostas a cada ano, levando a probabilidade de erro na escala de um bit a cada milhão. Empresa líder de mercado em transmissões Broadcast Novelsat, lançou em 2017 a modulação *NS4*, sendo o que se tem de mais moderno no mercado broadcast, trazendo um *rolloff* de 2% em comparação aos *rolloff* de 5% da geração anterior *NS3* (NOVELSAT, 2018 (acessado 02 de Abril, 2019)).

Mas inerente dos mecanismos de transmissão que sejam criados, a interferência intersimbólica é encontrada não importando o método escolhido de transmissão. Essas distorções no sinal transmitido tem inúmeras fontes de causa, seja pela dispersão de energia provenientes de reflexões no solo, em obstáculos devido ao multi-percurso que o sinal transcorre e desvanecimento do sinal. Outras complicações, como a degradação e enfraquecimento do sinal devido a chuva e o efeito Doppler podem acarretar na perda da disponibilidade do sinal (HAYKIN, 2001). Uma das distorções mais comuns que ocorrem devido à interferência no canal são as interferências inter simbólica (IIS), que consiste na extensão de um símbolo transmitido sobre um intervalo de tempo maior do que sua duração original, fazendo com que interfira no próximo símbolo transmitido. Isso ocorre

devido a efeitos de banda limitada na resposta em frequência do canal (NEVES et al., 2001).

Para recuperar o sinal depois que ele sofre distorções lineares, utilizar equalização do canal é um método muito eficiente, pois compensa as distorções sofridas pelo sinal (HAYKIN, 1994). Mas muitas das aplicações têm o canal variável ao longo do tempo, requerendo ajustes na equalização do sinal recebido, esse mecanismo é chamado de equalizador adaptativo. O componente utilizado para realização desses métodos são os equalizadores (WIDROW, 1985).

Existem dois tipos de equalizadores. Equalizadores supervisionados utilizam sinais de treinamentos conhecidos pelo transmissor e receptor para que seja medido o tipo de deformação que o sinal sofre, possibilitando a determinação do equalizador e assim, recupere o sinal. Para isso pode-se utilizar inúmeras técnicas tais como o filtro de Wiener (HAYKIN, 2008).

Mas dependendo da aplicação, não é possível a comunicação prévia entre receptor e transmissor e o transmissor não tem nenhum sinal de treinamento. Para isso se utilizam equalizadores não supervisionados, que apenas com a modulação usada e informações estatística da distribuição de símbolos, recupera o sinal enviado (HAYKIN, 2008). Esse tipo de equalização é o alvo de estudo desse trabalho .

Para realizar uma equalização não supervisionada, pode-se utilizar métodos como algoritmo de Ding, Shalvi Weinstein, CMA e Goddard. (DING; LUO, 2000) propõe um método de programação linear para encontrar o melhor coeficiente do equalizador para uma transmissão que não se conhece a informação e a modulação.

Esse trabalho propõe um estudo mais aprofundado de equalizadores, explicando com mais detalhes as principais técnicas e aplicações. Para isso, foram implementados um método de equalização supervisionada, o Filtro de *Wiener*, e um método de equalização não supervisionado , algoritmo de *Ding*.

## 2 Equalizadores

As aplicações mais estudadas na literatura de filtros adaptativos são a equalização de canal e identificação de sistemas (WIDROW, 1985).

A proposta de identificação de sistemas tem o objetivo de obter uma aproximação da resposta ao impulso de um sistema. A obtenção de modelos de uma canal de comunicação de telefonia móvel ou de sistemas geológicos, como a reflexão sísmica, são exemplos deste tipo de aplicação (HAYKIN, 1994).

Equalização de canal com uso de filtros lineares, por sua vez, é implementado por um processo de convolução. Tal processo pode ser definido como um método de compensação, como a reversão dos efeitos causados pela convolução entre a resposta impulsiva de um determinado sistema e um sinal portador de informação que passa através do sistema. Aplicando na questão de um canal de comunicação sem fio, a convolução da resposta do canal e o sinal de interesse transmitido termina por gerar o efeito IIS. Este tipo de interferência costuma ocorrer devido ao efeito de multipercurso presente neste tipo de canal. Tal efeito, ocorre quando diversas versões do mesmo sinal chegam ao receptor em instantes distintos, permitindo que ocorra a sobreposição no tempo de versões atuais e atrasadas de um mesmo símbolo transmitido. Além disso, equalização de canal consiste em inserir um filtro inverso em série em relação a um determinado sistema, cuja função é a de recuperar a informação contida no sinal. Por causa dessa característica, é uma aplicação também conhecida com filtragem inversa (MANN; WALKER, 2002).

Considerando a sequência  $a_t$  como sinal a ser transmitido e desconhecido do ponto de vista do receptor, e filtrado por um canal desconhecido com resposta ao impulso  $h_k$ , obtêm a mensagem recebida no equalizador ,  $x_t$ , dada por:

$$x_t = \sum_{k=0}^{\infty} h_k a_{t-k} + z_t \quad (2.1)$$

A equação (2.1) contém as informações do sinal original mais ruído aditivo, denotado por  $z_t$  . O objetivo final da tarefa de desconvolução é restaurar, ou remover os efeitos causados por este processo de distorção do sinal de origem sofrida no canal. O processo de desconvolução se dá pela filtragem inversa deste sinal recebido, com um vetor de coeficientes que idealmente equivalem a inversa da resposta ao impulso do canal. Por simplicidade, o processo de desconvolução é aproximado pelo modelo de resposta de impulso finito (FIR) (HAYKIN, 1994):

$$y = \sum_{k=0}^L \theta_k^* x_{n-k} \quad (2.2)$$

Sendo  $N$  o tamanho do equalizador,  $\theta_k$  um coeficiente do equalizador. A tarefa da desconvolução causada pelo equalizador  $\Theta$  é obter um:

$$y_n = \alpha x_{n-v} \quad (2.3)$$

Sendo  $\alpha$  um ganho desconhecido e  $d$  um valor inteiro referente ao atraso do sistema. Em algumas situações, o sinal original é recuperado na saída do sistema de desconvolução, exceto talvez por um ganho e atraso arbitrários.

## 2.1 Equalizadores Supervisionados

Alguns algoritmos utilizados para equalização de canal, requerem um sinal de treinamento. Um sinal de treinamento é uma sequência transmitida pela fonte emissora e conhecida pelo receptor. A disponibilidade deste sinal no receptor permite ao algoritmo utilizá-lo como referência para realizar a equalização. Este sinal de referência, ou sequência de treinamento, pode ser transmitido de maneira constante (HAYKIN, 1994).

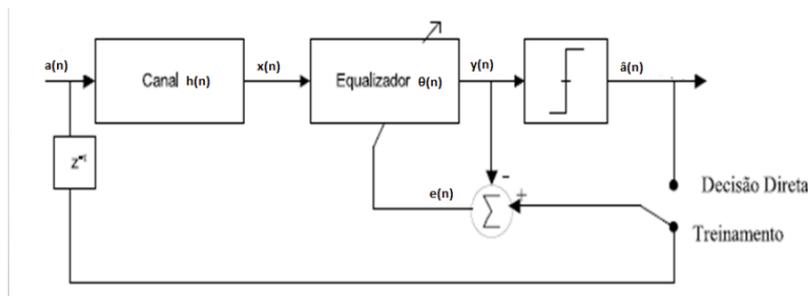


Figura 1 – Transmissão de um equalizador supervisionado (NEVES et al., 2001)

O dispositivo de decisão tem por objetivo remover alguma IIS residual que ainda exista no sinal de saída do equalizador. Se o canal tiver sido equalizado corretamente e o diagrama de olho neste ponto estiver aberto, o decisor conseguirá recuperar a sequência transmitida sem erros. Basicamente, este dispositivo procura qual o símbolo da constelação do sinal transmitido mais próximo da atual saída  $y(n)$  (NEVES et al., 2001). Métodos que utilizam sinais de referência são amplamente estudados na literatura, tais como Algoritmo de gradiente determinístico, algoritmo LMS e filtro de *Wiener*, sendo esse último, estudado mais detalhadamente e aplicado nesse trabalho.

Transmitir sequências de treinamento implica na redução da banda de transmissão de dados efetivos, por isso seu uso não é indicado em sistemas de comunicações que dispõe

de banda de transmissão limitada. Devido a este problema, algoritmos cegos podem ser usados para realizar a equalização de canal. Estes algoritmos trabalham de forma auto ajustável utilizando as próprias estatísticas do sinal recebido, sem que haja necessidade de transmissão de seqüências de treinamento (DENG; DOROSLOVACKI, 2006).

### 2.1.1 Filtro de Wiener

Os filtros de Wiener buscam minimizar o erro quadrático médio (EQM) entre a saída do equalizador  $y(n)$  e a resposta desejada  $d(n)$ . Possui uma função custo como mostrada na equação (2.4), sendo  $E[.]$  um operador esperança,  $t$  um atraso constante,  $\theta$  o equalizador e  $x(n)$  a saída do canal .

$$J_{eqm} = E[(e(n))^2] = E[(d(n) - y(n))^2] \quad (2.4)$$

Para encontrar os valores dos coeficientes do equalizador,  $\theta_i$ ,  $i = 0, \dots, L$ , que minimizam  $J_{eqm}$ , temos que derivar  $J_{eqm}$  em função de  $\theta(n)$ , se atentando, que quando  $\theta(n)$  for um número complexo, tem o formato de  $\theta_i = a_i + jb_i$  e tem que derivar em função de  $a_i$  e  $b_i$ . Assim, se divide o problema em duas partes, onde se procura minimizar tanto a parte real e a imaginária, mostrada em (2.5) (NEVES et al., 2001).

$$\theta = R_{xx}^{-1} p_{dx} \quad (2.5)$$

Definindo  $R_{xx}$  e  $p_{dx}$ , temos:

$$R_{xx} = E[\mathbf{x}^*(n)\mathbf{x}^T(n)] \quad (2.6)$$

$$p_{dx} = E[\mathbf{d}(n)\mathbf{x}^*(n)] \quad (2.7)$$

Para o cálculo de  $R_{xx}$ , definido em (2.6) é feito a autocorrelação dos valores recebidos com ele mesmo e depois calculado a sua inversa. Já o calculo de  $p_{dx}$ , definido em (2.7) é calculado a autocorrelação do valor esperado com o valor recebido.

Esse procedimento é chamado de filtro de *Wiener* (HAYKIN, 1994) e com ele é possível determinar os parâmetros ótimos para  $\theta$ .

## 2.2 Equalizadores Não Supervisionados

Equalizadores não supervisionados ou cegos buscam encontrar parâmetros baseados na observação da saída do canal, apenas sabendo da distribuição e propriedades estatísticas da informação da entrada (KENNEDY; DING, 1992). Essa técnica é utilizada em situações

onde ter uma sequência de informação de treinamento seja impossível, o custo de operação seja inviável ou que a limitação de banda seja crucial. Em comunicações do tipo *broadcast* a desconvolução cega pode ser utilizada para a remoção de ISI sem que haja comprometimento da banda disponível neste canal para a transmissão de sequências de treinamento. Em aplicações de teleconferência, a desconvolução cega pode ser utilizada para remoção dos efeitos de reverberação em sinais de voz, onde não há sequências de treinamento disponíveis.

Por não possuir mais o sinal de treinamento, o receptor se torna autoadaptativo sem que seja necessário parar o fluxo de envio de informação ou consumir banda com informação de treinamento. Além disso, permite que a informação seja recuperada em caso de falha do sistema ou variação do canal, permitindo que seja adaptado conforme o canal muda (KENNEDY; DING, 1992).

O primeiro processo de equalização cega foi proposto por (SATO, 1975), que desenvolveu um algoritmo de equalização mais robusto em comparação aos processos de equalização existentes na época e que não obtinham resultados satisfatórios.

A condição de convergência corresponde em assegurar que a condição *Open eye*, medida de análise de desempenho de transmissão que mede a precisão da recepção, seja satisfeita quando dados binários são transmitidos através do canal, que resulta na blindagem dos efeitos da interferência inter-simbólica na decisão do equalizador. Mesmo que Sato tenha conseguido alcançar tal condição, em alguns casos não necessariamente é possível equalizar o canal, tendo que recorrer a outros métodos de equalização (HAYKIN, 1994). O algoritmo de Sato pode ser visto como uma aproximação estocástica do algoritmo do Gradiente Estocástico para minimizar a função de custo.

Motivado pelo trabalho de Sato, (BENVENISTE; GOURSAT; RUGET, 1980) observou que, se o canal tiver um número finitos de polos (estável), a função distribuição dos dados de saída do canal tende a aumentar a sua porcentagem de acerto e assim equalizar os dados cujo comportamento são distribuições gaussianas.

(DONOHO, 1981) mostrou que é possível estabelecer uma ordem parcial, chamada Gaussianidade, em variáveis aleatórias e suas combinações lineares, concluindo que equalização cega deve simplesmente fazer com que a Gaussianidade decresça.

Os estudos realizados por (SATO, 1975) (BENVENISTE; GOURSAT; RUGET, 1980) e (DONOHO, 1981) definiram algumas características para equalizadores cegos que foram levantadas em (KENNEDY; DING, 1992) :

- Estatísticas de segunda ordem do sinal recebido apenas provêm informações de magnitude do canal e são insuficientes para equalização cega de um canal de fase não mínima.
- Canais de fase não mínima, a variação de fase do sinal ocorre de maneira ampla,

pois não necessariamente todos os zeros encontram-se dentro do círculo unitário do plano  $Z$ .

- Canais de fase não mínima não podem ser identificados a partir de sua saída quando o sinal de entrada for gaussiano, pois neste caso, estarão presente somente as estatísticas de segunda ordem.
- A inversa de um canal de fase não mínima é instável e sua expansão anticausal truncada pode ser atrasada por um delay, de maneira a permitir uma aproximação causal aos equalizadores do tipo zero forcing (ZF). equalizadores sem definição no ponto  $z=0$ .
- Equalizadores ZF não podem ser aplicados em canais cuja resposta contenha zeros fora do círculo unitário.

Devido à ausência da sequência de treinamento, exploram-se várias informações sobre a sequência de entrada e também da sequência de saída, com o intuito de melhorar a qualidade do equalizador (DING; LI, 2001).

No algoritmo de programação matemática aplicada em equalizadores autoidadada, a equalização ocorre com o tamanho da mensagem transmitida, sendo útil em casos onde há limitação de largura de banda ou com canais com muitos zeros (GODARD, 1980). (KENNEDY; DING, 1992) apresenta uma solução linear para equalizadores cegos com função de caráter convexo, onde a abordagem foca na eliminação da IIS sem identificação de ganho e fase. A abordagem desse método é a busca da otimização de uma função custo convexa através de programação linear, onde se procura a otimização da função, encontrando assim, o equalizador do canal.

### 2.2.1 Método de Ding

Baseado nos equalizadores por taxa de transmissão de (GODARD, 1980), que utiliza algoritmo de módulos constante (CMA) e algoritmo de Shalvi-Weinstein (SHALVI; WEINSTEIN, 1990), (DING; LUO, 2000) propôs um equalizador não supervisionado usando programação linear com uma função convexa. Essa formulação não identifica diretamente o ganho, gerando uma saída de equalizador com uma escala diferente do canal de entrada. Além disso, essa abordagem é independente de defasagem de fase, e para tratar desvio de fase, é necessário adicionar uma compensação após a equalização.

Inicialmente, em (KENNEDY; DING, 1992) foi proposto um equalizador que visava a minimização do custo da função de transmissão através da aproximação e estimação iterativa de dados. Isso resultou em um algoritmo lento, deixando inviável a sua aplicação em implementações práticas. (DING; LI, 2001) é proposto uma formulação dessa função para ser resolvida em programação linear para minimizar a função custo.

(DING; LI, 2001) utilizou Quadrature Amplitude Modulation(QAM) para equalizadores cegos na modelagem do problema abordado por programação linear. A modulação QAM é uma representação banda base, onde os símbolos originais  $a_k$  assumem valores com sinais complexos, conhecido como constelações, e garantindo que todas as possibilidades de símbolos finitos tem as mesmas chances de ocorrer. Esse sinal com valores complexos  $a_k$  são transmitidos em um canal não ideal, linear, causal e estável. Podemos escrever esse canal conforme a fórmula abaixo.

$$H(z^{-1}) = \sum_{i=-\infty}^{\infty} h_i z^{-i} \quad h_i \in C \quad (2.8)$$

Onde  $C$  significa o conjunto de números complexos e  $h_i$  representa a resposta impulsiva do canal. A saída do canal sofre com IIS e remoção de distorção linear se torna o objetivo de equalizador de canal, conforme mostra a equação (2.9).

$$x_k = \sum_{i=-\infty}^{\infty} h_i a_{k-i} \quad (2.9)$$

O equalizador linear é um filtro linear  $\theta(z^{-1})$  que é aplicado na saída do canal  $x_k$  com o objetivo de eliminar a IIS. Inicialmente, tomamos o filtro como estável e potencialmente não causal(duplamente infinito), servindo para poder lidar com canais de fase não mínima.

$$\theta(z^{-1}) = \sum_{i=0}^{\infty} \theta(i) z^{-i} \quad (2.10)$$

Como idealmente, um equalizador necessita ser IIR para equalizar  $x_k$  e obter o sinal transmitido  $a_k$ , os coeficientes do equalizador precisam ser definidos tanto negativamente e positivamente.

O sinal de saída do equalizador,  $y$ , pode ser escrito como a autocorrelação dos coeficientes do equalizador e a saída do canal.

$$y_k = \sum_{i=-\infty}^{\infty} \theta_k x_{k-i} \quad (2.11)$$

Se multiplicarmos o canal com equalizador, espera-se obter uma recepção sem ruído, como representado na equação (2.14).Onde podemos tolerar um inteiro delay fixo, que identifica o resultado do equalizador como canal inverso .

$$H(z^{-1})\theta(z^{-1}) = z^{-v} \quad (2.12)$$

Em equalização cega, o sinal original é desconhecido para o receptor. A única informação que o receptor possui são as propriedades probabilística ou estatística do

alfabeto de símbolos, a constelação tem propriedades simétricas, o que permite concluir que qualquer rotação  $m\pi/2$  não causa mudança estatística na saída do canal.

(DING; LI, 2001) chega na função custo (2.15), que se busca minimizar:

$$J(\theta) = \max|y_k^R| + \max|y_k^I| = \sum_{i=-\infty}^{\infty} [\theta_i^R x_{k-i}^R + \theta_i^I x_{k-i}^I] \quad (2.13)$$

Onde  $y_k^R$  é a parte real do sinal de saída do equalizador e  $y_k^I$  é a parte imaginária do sinal de saída do equalizador,  $x_{k-i}^I$  é a parte imaginária do sinal de saída do canal e  $x_{k-i}^R$  é a parte real. O mesmo se aplica para  $\theta_i^R$  e  $\theta_i^I$ , que representam a parte real e imaginária do equalizador.

Mas sem nenhuma restrição em  $\theta$ , a função obtém o valor global trivial de  $\theta = 0$ , levando a saída do equalizador a ser também  $y_k = 0$ . Logo, é necessário colocar uma restrição na função custo, para que  $J(\theta)$  assumam resultados úteis, então usamos a restrição linear (2.16).

$$\theta_0^R + \theta_0^I = 1 \quad (2.14)$$

Essa restrição foi proposta em (KENNEDY; DING, 1992), para evitar uma solução com apenas zeros e não comprometer a habilidade do equalizador remover a IIS e a distorção de fase. Com a restrição proposta, a função mantém a linearidade, a convexidade e a convergência em um resultado global assegurado.

## 2.3 Sedumi

O sedumi é uma biblioteca de extensão do Matlab que resolve problemas de programação linear. Criado por Jos F. Sturm, falecido em 2003, atualmente o laboratório de otimização avançada da universidade de McMaster continua desenvolvendo e mantendo o software. O sedumi utiliza o método de *dual cone* para resolver problemas lineares.

A técnica de dual cone é baseada em uma estratégia de se obter limitantes inferiores ( caso seja um problema de minimização). Esta estratégia é muito comum em otimização e utiliza o conceito de relaxação, ou seja, se eu tenho um conjunto  $R$  que contém  $S$ ,  $R$  é uma relaxação de  $S$ , então o mínimo de uma função  $f$  em  $R$  deve ser menor ou igual ao mínimo de  $f$  em  $S$ . Isso porque, se existe um  $x^0 \in S$  tal que  $f(x^0) = \text{mínimo}\{f(x), x \in S\}$  e como  $x^0 \in R$ , então também pode ser um mínimo de  $f$  em  $R$  ou um outro ponto de  $R$  pode ser ainda melhor. Ou seja, diminui o problema em grupos menores, cujo ótimo local desse grupo pode ser um ótimo local ou ótimo global do grupo maior (ARENALES et al., 2015).

O sedumi utiliza a forma matricial do problema, portanto, é necessário a tradução dos problemas lineares para a forma matricial:

$$\min c^t x \quad (2.15)$$

$$\text{Sujeito a } Ax = b$$

Onde o vetor  $c$ , tem dimensão  $1 \times N$ , onde  $N$  representa a quantidade de variáveis a ser minimizada do problema. O vetor tem o formato  $\vec{c} = \{c_0 \ c_1 \ \dots \ c_N\}$ , que representa a equação a ser minimizada. Caso queira maximizar uma função, é necessário inverter o sinal dos valores de  $c$  (STURM, 1999). Já a matriz  $A$  e vetor  $\vec{b}$  representa todas as restrições na forma de equação linear,  $A$  representa todos os valores que cada elemento de  $x$  assume nas restrições. Portanto, tem a dimensão  $R \times N$ , sendo  $R$  o número de restrições do problema. Já o vetor  $\vec{b}$  tem a dimensão  $R \times 1$ , pois representa a igualdade ou inequação das restrições.

Para trabalhar com inequações no sedumi, é necessário transformar a inequação na forma de uma restrição de igualdade. Para isso utiliza-se mais duas variáveis de folga, variáveis nesse caso para transformar inequação em equação, quando é uma inequação negativa a primeira variável será  $-1$  e quando for positiva a primeira variável será  $+1$  (STURM, 1999).

Em problemas lineares, é necessário definir a condição de existência das variáveis, o modo *default* do sedumi, vem todas as variáveis como maior ou igual a zero com condição de existência, mas em muitos problemas, é necessário uma customização dessa condição. O sedumi apresenta a variável  $k$ , que altera a configurações das condições de existência. A variável  $k.f$  configura o número de variáveis livres, variáveis que podem ser positivo ou negativas e  $k.l$  define o número de não negativos (STURM, 1999).

O sedumi seta as variáveis de forma sequencial, se está trabalhando um vetor de variáveis  $\vec{c}$  com 10 elementos, se é configurado  $k.f$  igual a 4 e  $k.l$  igual a 3, o sedumi vai entender que  $x_0$  a  $x_3$  são livres,  $x_4$  a  $x_6$  são variáveis que só podem ter valores negativos e  $x_7$  em diante só assumem valores maior ou igual a zero.

Configurado as variáveis  $(A, b, c, k)$ , é necessário criar um vetor para receber os resultados do sedumi, e chamar a função na forma  $\text{sedumi}(A, B, c, k)$ , ou caso queira ver as iterações que o programa faz, necessário chamar o programa com dois vetores na forma  $[x, y, info] = \text{sedumi}(A, B, C, k)$  (LABIT; PEAUCELLE; HENRION, 2002).

## 3 Metodologia

### 3.1 Modelagem do Filtro de Wiener

A implementação do filtro de Wiener foi feita seguindo as equações definidas em (2.5), (2.6) e (2.7). Primeiro foi definido o vetor  $d(n)$ , que recebe a parcela do sinal  $a(n)$  que foi transmitido para ser usado como sinal de treinamento. Depois disso, configura  $R_{xx}$  como a autocorrelação do sinal de saída do canal  $x(n)$  com ele mesmo e  $p_{dx}$  como a autocorrelação do vetor  $d(n)$  com o vetor  $x(n)$ . Construído  $R_{xx}$  e  $p_{dx}$ , realiza a equação (2.5), multiplicado a inversa de  $R_{xx}$  por  $p_{dx}$ , obtendo assim o valor ótimo de  $\theta$ .

Obtido o equalizador  $\theta$ , faz-se a filtragem inversa do equalizador com o sinal de saída do canal  $x(n)$ , obtendo como resultado final, os dados equalizados.

Para obtenção da equalização em canais de fase não-mínima, apenas a construção mostrada anteriormente não é o suficiente, é necessário atrasar o sinal de treinamento para que o equalizador ótimo seja suficiente para equalizar o sinal, logo  $d(n)$  tem que ser  $d(n + v)$ , sendo  $v$  um atraso no sinal de treinamento.

### 3.2 Modelagem do Método de Ding

Antes da implementação da função linear, foram examinadas algumas limitações teóricas e outras considerações:

- Convexidade é uma propriedade interessante na formulação para equalizadores, pois garante a convergência global do algoritmo.
- A identidade em módulo e fase dos símbolos da constelação, garante que a defasagem de fase obtida na equalização seja de  $k\pi/2$ . Garantindo que a formulação recupere a fase também.

Para a implementação do algoritmo proposto em (DING; LI, 2001), é necessário escolher a dimensão do equalizador, o tamanho da parcela da mensagem que será usada na equalização. Como visto no capítulo anterior, o equalizador cego ideal precisa ser um IIR, duplamente infinito, mas na prática, pela limitação do tamanho da mensagem e limitação de implementação, se restringe o tamanho usado a  $2N + 1$ , onde  $N$  é o tamanho do equalizador  $\theta = (\theta_{-N}, \theta_{-N+1}, \dots, \theta_0, \dots, \theta_N)$ , que significa que o tamanho do equalizador será duas vezes o tamanho da amostra da mensagem que será utilizada, deixando o vetor  $\theta$  com um comportamento de um filtro IIR.

Estaremos trabalhando com modulação 4QAM , sendo necessário utilizar as partes reais e imaginárias separadas dos símbolos, ficando a função a ser minimizada igual a:

$$\min \max |y_k^R| + \max |y_k^I| \quad (3.1)$$

$$\text{Sujeito a } \theta_0^R + \theta_0^I = 1$$

Introduzindo a variável auxiliar  $t_2$  proposto em (DING; LI, 2001), é possível reformular a equação acima como o seguinte problema linear:

$$\min t_2 \quad (3.2)$$

$$\text{Sujeito a } \theta_0^R + \theta_0^I = 1$$

$$-t_2 \leq \sum_{i=-N}^N [x_{k-i}^R \theta_i^I - x_{k-i}^I \theta_i^R] \leq t_2$$

$$k = -K, \dots, -1, 0, 1, \dots, +K$$

Nota-se a aparição de um somatório que varia  $i$  de  $-N$  a  $+N$ , sendo  $k$  a variação dos coeficiente do equalizador  $\theta$ . Para cada valor que  $k$  define, aumenta a mensagem  $x_t$  utilizada para equalizar o  $\theta$ . Como definido antes, o equalizador ideal é duplamente infinito, um IIR, e para que seja possível sua implementação, é necessário transformar ele para um FIR sem que perca as suas característica IIR. Para que isso ocorra, é necessário implementar o equalizador com coeficientes de variável livres, podendo tomar valores positivos e negativos. O tamanho do equalizador, idealmente teria que ser infinito, mas computacionalmente, não é possível implementar isso, então a variável  $N$  é definida para construir o tamanho do equalizador e a variável  $K$  a variação da mensagem para equalizar.

### 3.2.1 Implementação no Sedumi

Como visto no capítulo 2.3, o Sedumi resolve os problemas lineares na forma matricial, resolvendo a formulação de minimização com um vetor  $c$  que recebe a função a ser minimizada, e recebe as restrições na forma  $Ax = b$ , para isso é necessário a adaptação do problema definido em (2.15) na forma matricial.

Como o caso trabalhado utiliza variáveis reais e imaginárias, é necessário separar as variáveis em parte real e outra em parte imaginária. Depois de encontrar os coeficientes  $\theta$  do equalizador, é necessário juntar as partes reais e imaginárias para realizar a filtragem inversa e restaurar o sinal enviado. Dessa forma, o Sedumi não estará dando os coeficientes do equalizador de forma direta, pois é necessário juntar as componentes reais e imaginárias antes disso.

Trabalhando com a variável auxiliar, é possível trabalhar com a função (3.2), que utiliza a variável auxiliar  $t_2$  como função a ser minimizada. Além disso, as restrições contêm inequações, sendo necessária a criação de uma variável auxiliar pra cada inequação que seja utilizada, impactando diretamente no tamanho de  $\vec{c}$  e da matriz  $A$ , pois para cada restrição com inequação utilizada, vai aumentar uma variável de  $x$ . No algoritmo de Ding, definido em (3.2), utilizam-se duas inequações para cada  $k$ , então, o número de restrições ( $R$ ) que utiliza inequações acaba sendo  $R - 1$ , pois a restrição que evita valores triviais é uma equação.

As demais variáveis são definidas pelo tamanho do equalizador ( $N$ ) que vai ser utilizada, multiplicada por quatro, por causa da separação das variáveis nos seus coeficientes reais e imaginários, pela variável zero e pelo comportamento IIR que o equalizador tem que seguir, é necessário a expansão simétrica dos valores em positivo e negativo.

Levando em conta que para cada inequação na restrição vai ser necessário uma variável livre, além da variável auxiliar  $t_2$  e o tamanho da amostra utilizado vezes quatro mais a variável zero. O vetor de variáveis tem o formato:

$$\mathbf{x}^t = \left( \theta_{-N}^R \quad \theta_{-N}^I \quad \dots \quad \theta_0^R \quad \theta_0^I \quad \dots \quad \theta_N^R \quad \theta_N^I \quad t_2 \quad s_1 \quad s_2 \quad \dots \quad s_{R-1} \right)$$

Para definir a matriz  $A$ , é necessário criar uma matriz com dimensão  $[R, (2N + 3 + (R - 1))]$ , o número de colunas se dá pelo tamanho do equalizador que será utilizado, definido positivamente e negativamente, além dele definido na componente em  $\theta_0$  e o número de linhas se dá pelo número de restrições, que é dado pelo número de  $K$  usados a restrição não trivial.

Além disso, existe a variação de  $k$ , que define o tamanho da mensagem usada na equalização e seguindo a formulação (3.2), é necessário uma restrição para cada valor de  $k$ , sendo que  $k$  varia tanto para valores positivos e negativos. Por exemplo, caso  $k$  seja 2, necessitará de 5 restrições, pois  $k$  vai assumir valores iguais a  $k = -2, -1, 0, 1, 2$ . Logo, como são duas inequações, são 2 restrições para passar no Sedumi para cada valor definido por  $k$ . O vetor  $b$  tem dimensão  $[R, 1]$  e recebe uma matriz com apenas valores de cada restrição.

Como o problema necessita de variáveis livres, é necessário configurar  $k$ . O problema necessita que todas as variáveis sejam livres, então é setado  $k.f$  com  $2N + 3$ , e como está sendo trabalhado com inequação, é setado  $k.l$  com  $R - 1$ , pois a restrição  $\theta_0^R + \theta_0^I = 1$  é uma igualdade. Como o Sedumi trabalha em pilha, a restrição de igualdade então deverá ser representada na última linha da matriz  $A$ , para que o programa entenda que é uma igualdade.

Para saber qual parte da mensagem transmitida utilizar na equalização e saber qual o tamanho máximo que pode ser utilizado para equalizar, é necessário o controle do

tamanho da Mensagem (M), que precisa ser menor ou igual ao tamanho do equalizador mais a mensagem equalizada.

Supondo um exemplo, com uma mensagem transmitida com modulação 4-QAM com 10 amostras, cada amostra original é descrita como  $a_k$ . Esse sinal é passado por um canal de fase mínima (H) e vai ser utilizado o equalizador de Ding para definir o melhor equalizador para recuperar o sinal. O equalizador será definido por  $\Theta$ , cujo os coeficiente são  $\theta_i$ . O sinal que chega no equalizador é definido como  $x_i = \sum_{i=0}^N h_i a_i$ , sendo que  $i$  varia de 0 a 9.

Para esse equalizador, vai ser utilizado um tamanho 3, isso quer dizer,  $N=1$ , logo  $\Theta$  terá os coeficientes  $\Theta = (\theta_0^R, \theta_0^I, \theta_1^R, \theta_1^I, \theta_2^R, \theta_2^I)$ , que serão ajustados pelo método. Para definir os valores possíveis de  $N$ , é necessário que  $N + k < 10$ , para esse exemplo, será variado a mensagem em  $k=1$ , isso quer dizer que  $k = -1, 0, 1$ , sendo o tamanho da mensagem usada (M) igual a 7, respeitando a restrição de ser menor de 10. Portanto, para escolher quais amostras da mensagem equalizar, o ponto que representar  $\theta_o$  mais 3 não pode superar o tamanho da mensagem, que é 10. O pivo escolhido para equalizar nesse problema será a amostra  $x_5$  como  $\theta_0$ .

Para definir os parâmetros do método de programação linear que será passado para o Sedumi resolver o problema linear, tem que definir  $\vec{c}$ ,  $\vec{b}$  e a matriz  $\mathbf{A}$ .

Para o vetor  $\vec{c}$ , recebe nesse exemplo tamanho 1x13, o vetor  $\vec{c}$  é representado com cada valor de  $\theta$  e as variáveis de folga, porém, como visto na formula (2.15), apenas é definido na variável  $t_2$ :

$$\mathbf{c}^t = ( 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 )$$

Para a matriz  $\mathbf{A}$ , definida com tamanho igual  $R \times (4k+5+(R-1))$ , duas para cada valor de  $k$ , e uma pela restrição que impede valores nulos. A primeira linha de restrição representa  $k=-1$ , sendo que o primeiro valor da somatória da restrição, sendo que  $x_{k-i}$ ,  $i$  assume do primeiro valor da somatória, que é -1, então, o primeiro valor é  $x_0$ . A matriz  $\mathbf{A}$  leva em conta a equação descrita em (3.3), onde existe uma somatória de  $-N$  a  $N$ .

Substituindo pelos valores das amostras, tem a forma de:

$$\mathbf{A} = \begin{pmatrix} x_5^R & x_5^I & x_4^R & x_4^I & x_3^R & x_3^I & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ x_5^R & x_5^I & x_4^R & x_4^I & x_3^R & x_3^I & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ x_6^R & x_6^I & x_5^R & x_5^I & x_4^R & x_4^I & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ x_6^R & x_6^I & x_5^R & x_5^I & x_4^R & x_4^I & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ x_7^R & x_7^I & x_6^R & x_6^I & x_5^R & x_5^I & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ x_7^R & x_7^I & x_6^R & x_6^I & x_5^R & x_5^I & -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & x_5^R & x_5^I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

O vetor  $\mathbf{b}$  recebe os valores de cada restrição, portanto tem uma dimensão de  $[7,1]$ , com o formato abaixo:

$$\mathbf{b} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Com essas matrizes configuradas, para rodar o `sedumi`, é necessário passar no formato `sedumi(A,b,c,k)`. Os resultados serão no formato  $\theta_i^R$  e  $\theta_i^I$ , que devem ser juntados em  $\theta_i = \theta_i^R + i\theta_i^I$  antes de fazer a filtragem inversa de  $\Theta$  com a mensagem recebida do canal  $x$  e obter a mensagem recuperada  $y$ .



## 4 Resultados

Os resultados foram rodados em um computador da marca Dell XPS 13, com processadores intel i7 de 1,8 e 1,99 Ghz e memória RAM de 8Gb. Foi utilizado a versão R2014b do Matlab e versão 5.0 do Sedumi. Para essas simulações não foram usadas ruído branco.

Foram implementados dois equalizadores, um equalizador supervisionado, filtro de *Wiener*, e um equalizador não supervisionado, Programação Linear.

Para validação dos equalizadores, foram usadas as modulações 4QAM e 16QAM. Também foram usados dois tipos de canais, o canal de fase mínima  $H(z) = e^{j\pi/4}/(0.7 - z^{-1})$  e o canal de fase não mínima  $H(z) = (e^{j\pi/4}(0.7 - z^{-1}))/((1 - 0,7z^{-1}))$ .

Foram utilizadas mensagens de 1000 amostras geradas aleatoriamente, tendo a mesma chance de ter cada um dos símbolos. Nas figuras 2 e 3, é mostrado as modulações 4QAM e 16QAM geradas no Matlab.

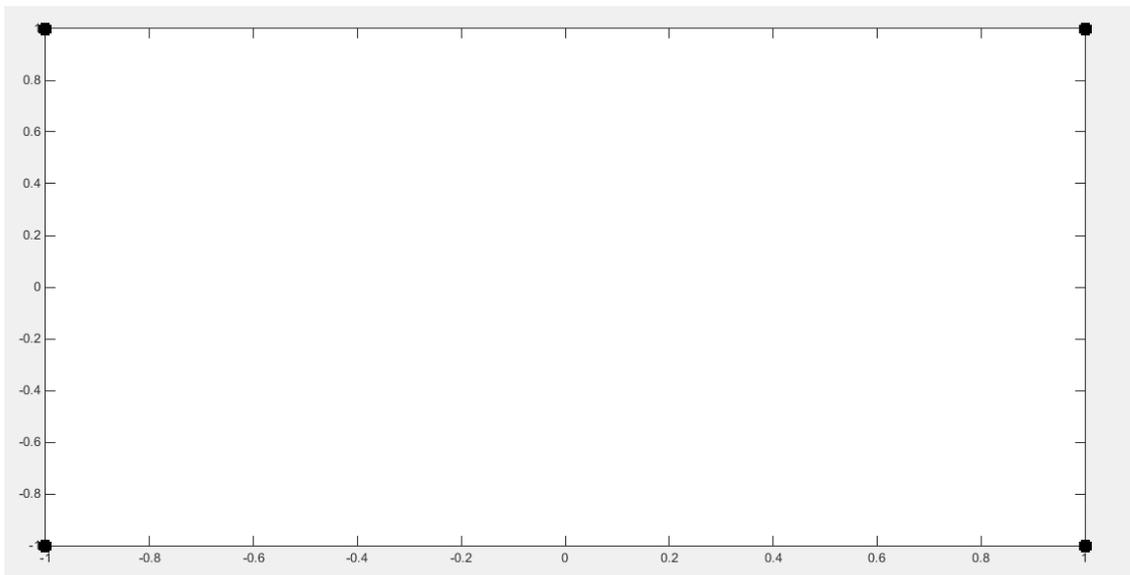


Figura 2 – Modulação 4-QAM

As figuras 4 e 5, mostram a mensagem 4QAM na saída do canal de fase mínima e pelo canal de fase não mínima, respectivamente.

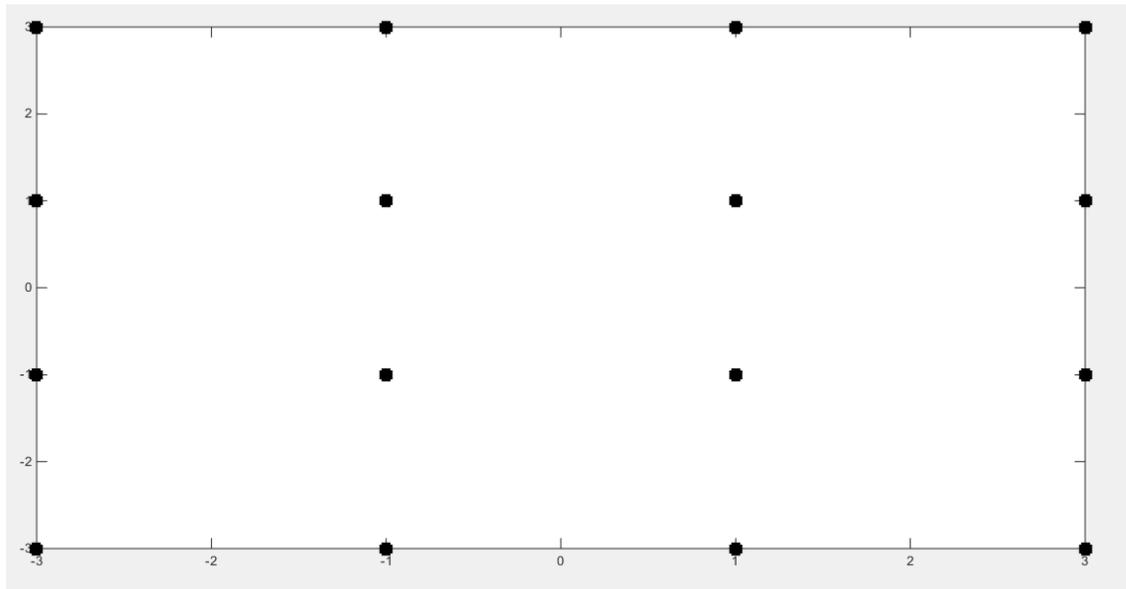


Figura 3 – Modulação 16-QAM

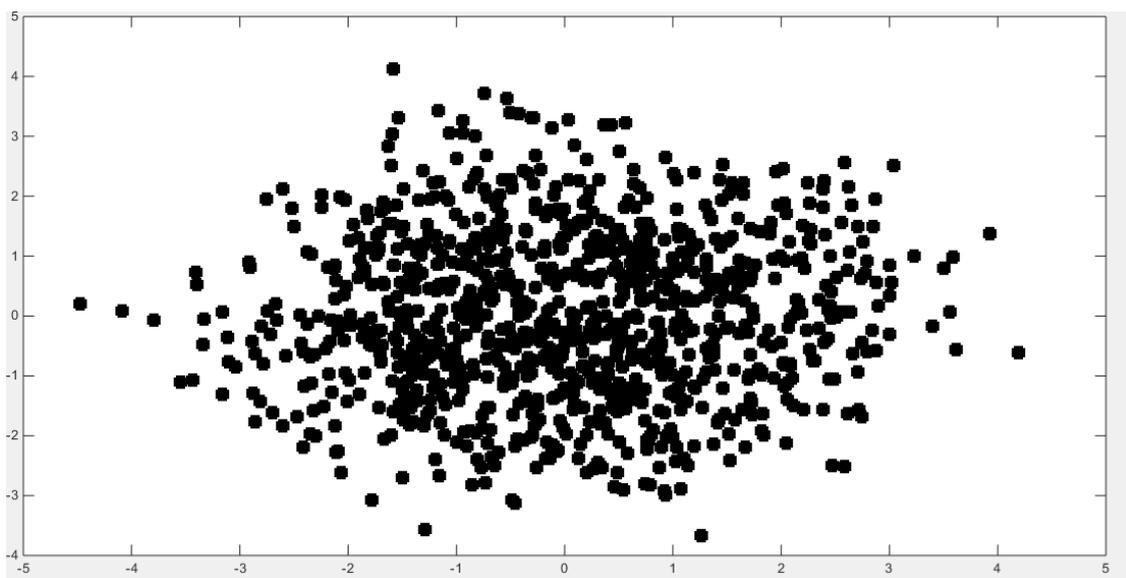


Figura 4 – Sinal depois do canal de fase mínima

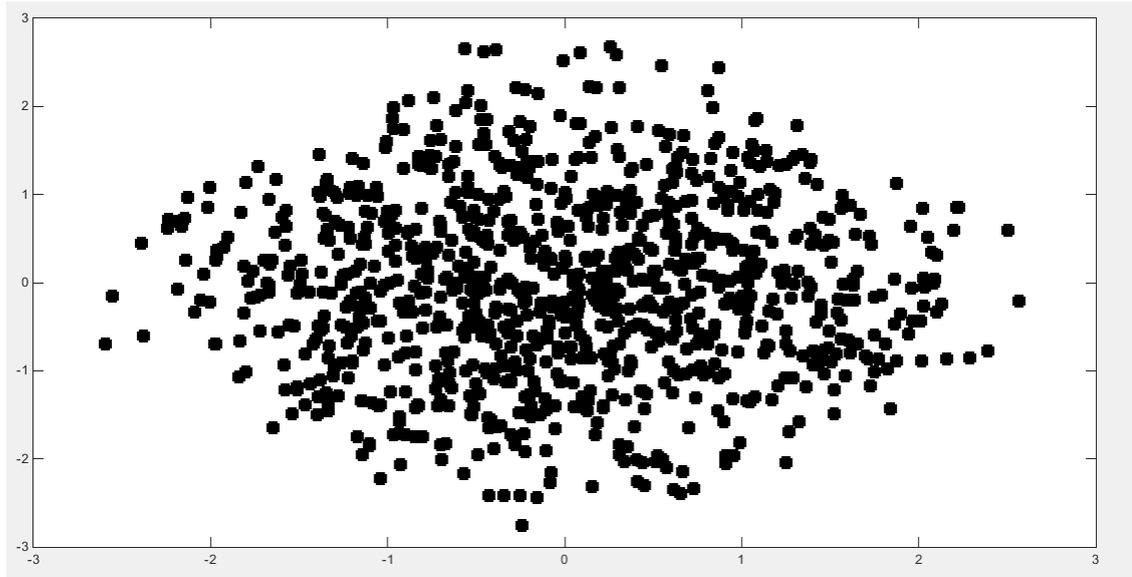


Figura 5 – Sinal depois do canal de fase não mínima

## 4.1 Filtro de Wiener

Passando pelo filtro de Wiener, os sinais equalizados para canal de fase mínima foram:

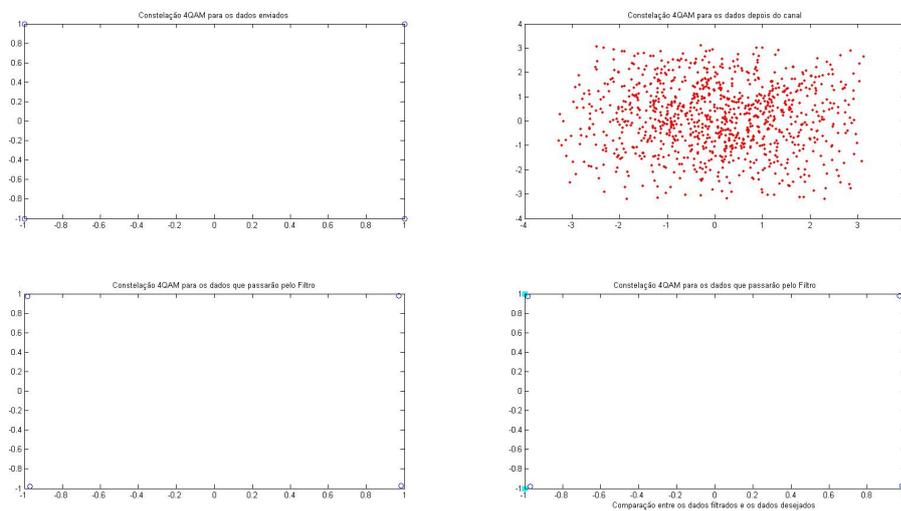


Figura 6 – Filtro de Wiener em um canal de fase mínima em modulação 4-QAM

Na figura 6, vemos todos os processos da transmissão para uma modulação 4-QAM passando em um canal de fase mínima, o resultado do equalizador e a comparação dele com o resultado original. Vemos que o filtro de Wiener que foi aplicado conseguiu equalizar de forma eficiente o proposto.

Vemos na figura 7, o filtro de Wiener aplicado em uma modulação 16-QAM passando

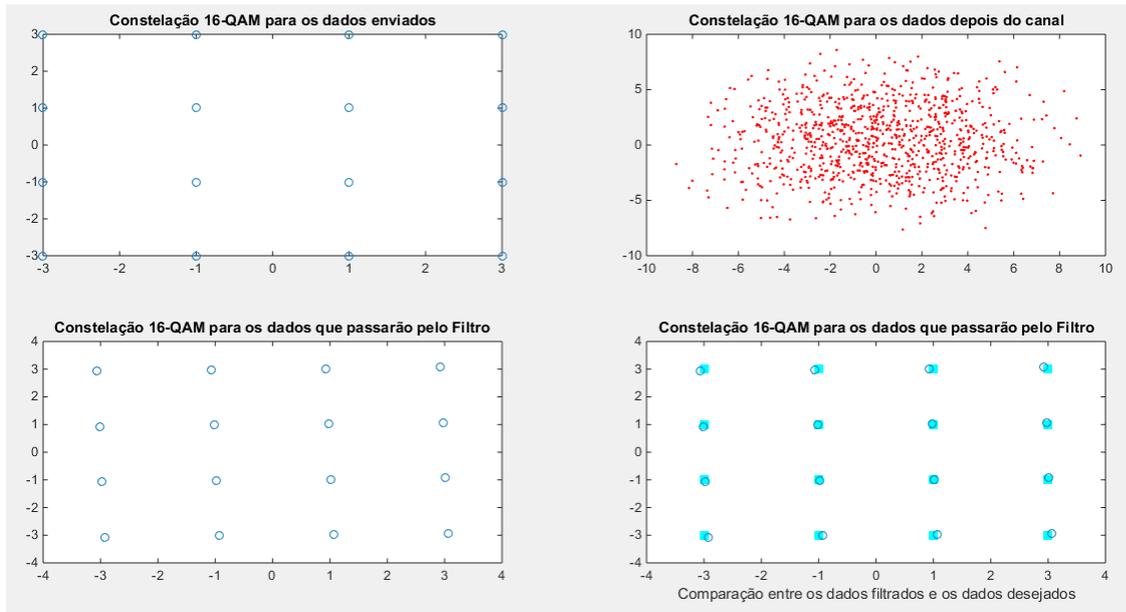


Figura 7 – Filtro de Wiener em um canal de fase mínima em modulação 16-QAM

em um canal de fase mínima, novamente consegue equalizar de forma eficiente o sinal.

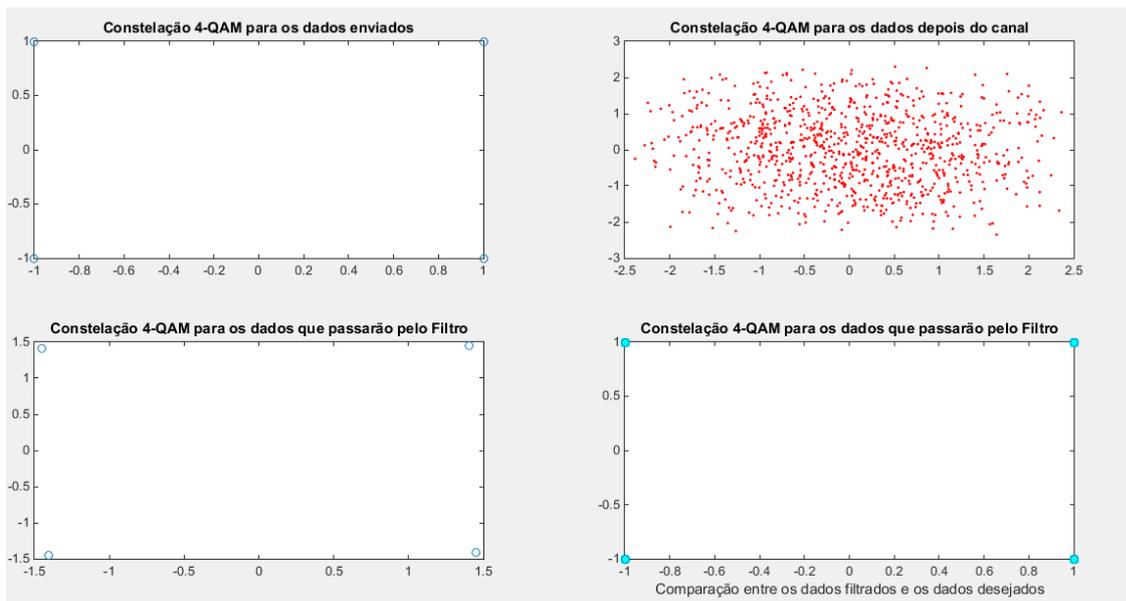


Figura 8 – Filtro de Wiener em um canal de fase não mínima em modulação 4-QAM

Na figura 8, temos a resposta do filtro de Wiener para uma modulação 4-QAM em um canal de fase não mínima. Vemos que o equalizador apesar de recuperar a fase, não consegue recuperar de forma exata a magnitude do sinal. Sendo necessário o atraso do sinal de treinamento em 2 símbolos.

Já na figura 9, apesar de recuperar o sinal de forma eficiente, o filtro de Wiener conseguiu obter a magnitude da mensagem original. Sendo necessário o atraso do sinal de treinamento.

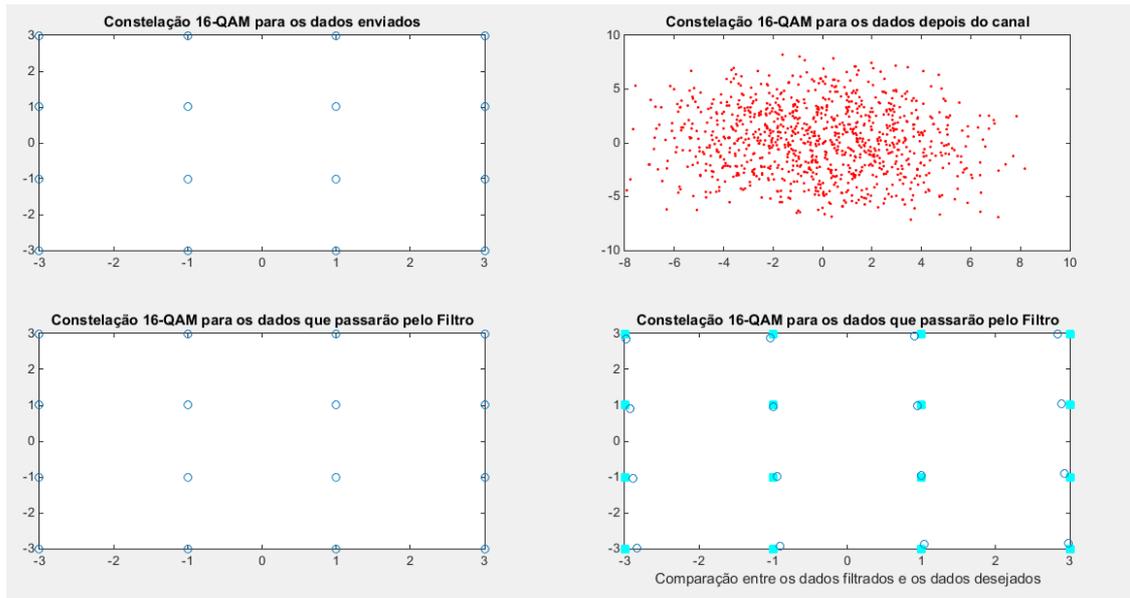


Figura 9 – Filtro de Wiener em um canal de fase não mínima em modulação 16-QAM

## 4.2 Método de Ding

Para cada caso, foi necessário o ajuste das variáveis de tamanho do equalizador necessário para equalizar e a quantidade de amostras necessárias da mensagem transmitida. Passando pelo equalizador do método de Ding, os sinais equalizados que foram encontrados para canal de fase mínima foram:

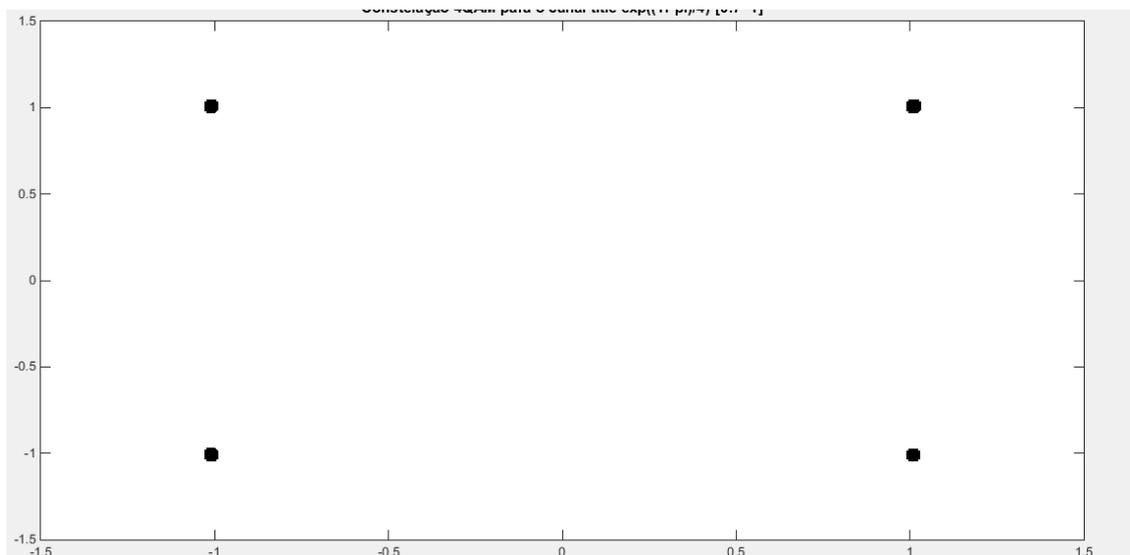


Figura 10 – Programação Linear em uma modulação 4-QAM em canal de fase mínima

Na figura 10, como podemos ver, o método de programação linear é capaz de obter a equalização de forma eficaz, apesar de não ter recuperado a magnitude do sinal transmitido, foi utilizado um  $k = 1$  e  $N = 17$  para conseguir equalizar.

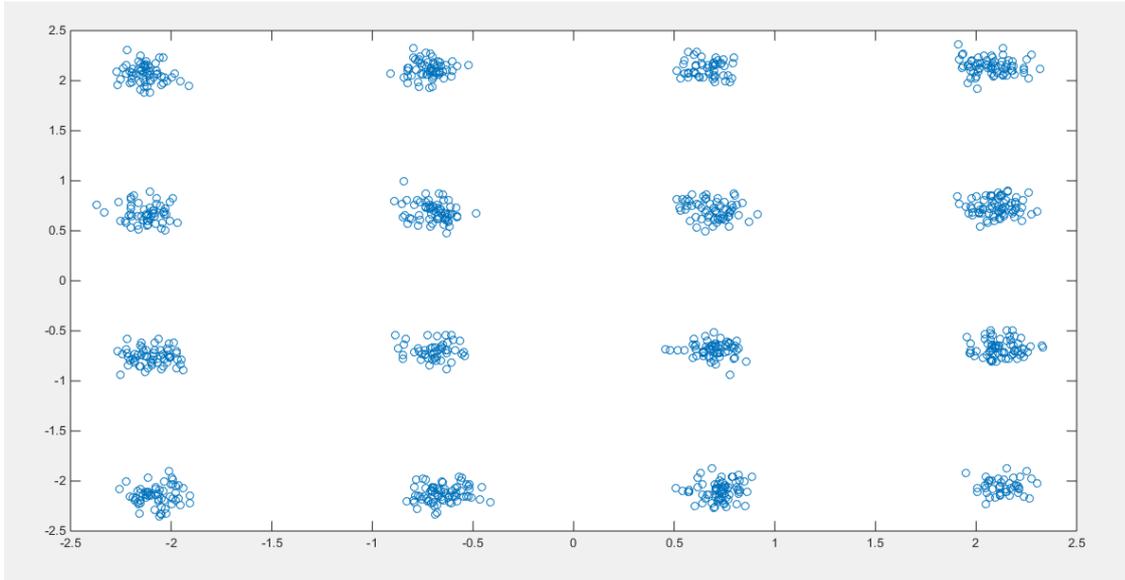


Figura 11 – Programação Linear em uma modulação 16-QAM em canal de fase mínima

Já na figura 11, temos o equalizador em um 16-QAM, podemos notar que ele é capaz de equalizar a mensagem, apesar de não ser tão preciso como o filtro de Wiener ao passar pelo mesmo canal, foi utilizado um  $k=1$  e  $N=60$ .

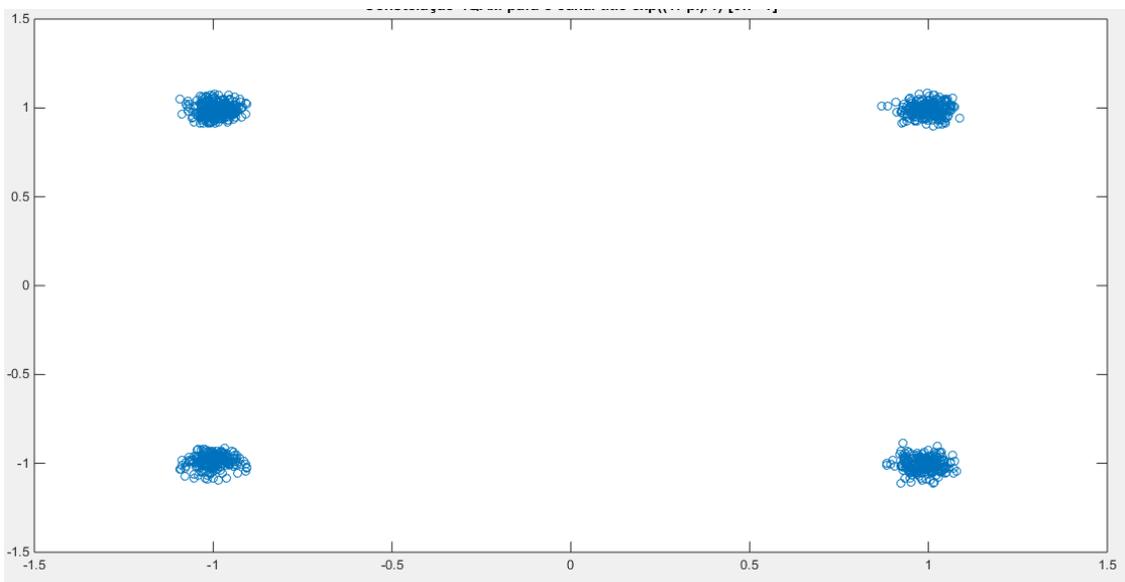


Figura 12 – Programação Linear em uma modulação 4-QAM e canal de fase não mínima

A figura 12 e 13, apresenta o método de programação linear em um canal de fase não mínima, foi necessária a implementação de uma compensação de fase de  $\pi/2$ , pois o algoritmo não prevê correção de fase. Para conseguir realizar a equalização, foram realizados inúmeros testes com varias simulações e um equalizador com  $k=9$  e  $N=400$  obteve o melhor resultado. Em comparação com os resultados obtidos no trabalho de (DING; LI, 2001), onde foi necessário 600 amostras e tamanho do equalizador de 17 para

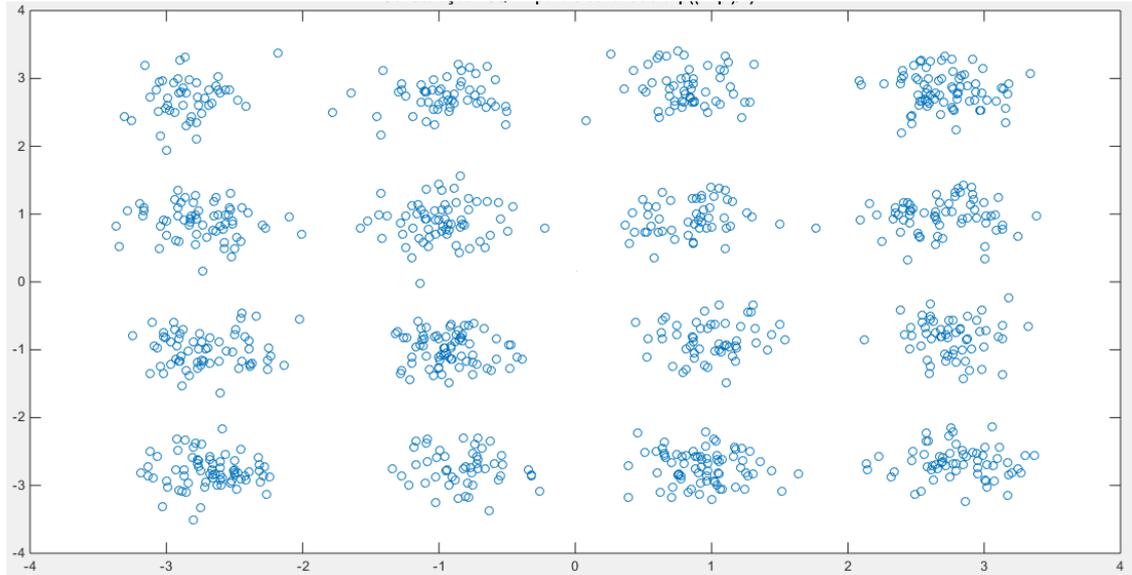


Figura 13 – Programação Linear em uma modulação 16-QAM e canal de fase não mínima

realizar a equalização do sinal quando é modulação 4-QAM e necessário 2000 amostras e equalizador de tamanho 21 para equalizar o sinal transmitido em fase não mínima para 16-QAM. O resultado apresentado nesse projeto apresentou resultados semelhantes ao apresentado no trabalho fonte. Os resultados obtidos foram fixados em 1000 amostras, devido ao custo computacional necessário para realizar a programação linear, foi visto que até 1000 amostras, o computador utilizada apresentava uma boa performance ao utilizar o método.



## 5 Conclusão

O objetivo do presente trabalho foi abordar o tema de equalizadores, debater um pouco sobre a importância do tema, definir os mais importantes métodos existentes na literatura e validar de forma prática, dois métodos diferentes de equalizar um sinal.

No presente estudo foram abordados dois algoritmos diferentes para equalização de canais de comunicação, um que opera de forma cega e outro que opera de forma supervisionada. Um algoritmo baseado em um método de programação linear para resolver equalização não supervisionada e um algoritmo que utiliza filtro de Wiener como equalização supervisionada.

Os algoritmos foram simulados computacionalmente, em ambiente controlado, com dois exemplos de possíveis cenários de aplicação, um canal mais simples, fase mínima, e um canal de fase não mínima. De acordo com as revisões teóricas para a aplicação de equalização de sistemas supervisionados e não supervisionados, e as devidas fundamentações teóricas de cada algoritmo através do capítulo 2, e os resultados obtidos através das simulações simuladas que foram apresentadas no capítulo 3. Verificou-se através dos resultados que todos os algoritmos testados cumprem o objetivo de equalização de canal, e diferenciam-se pelo seu desempenho.

O primeiro algoritmo, filtro de Wiener, apresentou exatidão e precisão ao equalizar canais de fase mínima, mas ao equalizar canais de fase não mínima, apresentou diferenças do resultado esperado, precisou-se então, da inserção de atrasos nos valores esperados para obter a equalização esperada.

O segundo algoritmo, utilizando equalizador cego, o algoritmo apresentou desempenho semelhante ao apresentado no estudo de (DING; LI, 2001). Para futuros trabalhos, considerando canais de maior complexidade e modulações de ordem mais alta que exigem maior precisão dos algoritmos, surgem novas vertentes para a sequência de estudos relacionados a estes algoritmos. A complexidade computacional é uma característica importante a ser abordada, principalmente o algoritmo de programação matemática, que exige um processamento mais robusto.



# Referências

- ARENALES, M. et al. *Pesquisa operacional: para cursos de engenharia*. [S.l.]: Elsevier Brasil, 2015. Citado na página 29.
- BENVENISTE, A.; GOURSAT, M.; RUGET, G. Robust identification of a nonminimum phase system: Blind adjustment of a linear equalizer in data communications. *IEEE Transactions on Automatic Control*, IEEE, v. 25, n. 3, p. 385–399, 1980. Citado na página 26.
- BRASIL, A. *Investimentos no setor de tecnologia aumentam 4,5% em 2017 no Brasil*. 2018 (acessado 02 de Abril, 2019). <<http://agenciabrasil.ebc.com.br/economia/noticia/2018-03/investimentos-no-setor-de-tecnologia-aumentam-45-em-2017-no-brasil>>. Citado na página 21.
- DENG, H.; DOROSLOVACKI, M. Proportionate adaptive algorithms for network echo cancellation. *IEEE Transactions on Signal Processing*, IEEE, v. 54, n. 5, p. 1794–1803, 2006. Citado na página 25.
- DING, Z.; LI, Y. *Blind equalization and identification*. [S.l.]: CRC press, 2001. Citado 7 vezes nas páginas 27, 28, 29, 31, 32, 42 e 45.
- DING, Z.; LUO, Z.-Q. A fast linear programming algorithm for blind equalization. *IEEE Transactions on Communications*, IEEE, v. 48, n. 9, p. 1432–1436, 2000. Citado 2 vezes nas páginas 22 e 27.
- DONOHO, D. On minimum entropy deconvolution. In: *Applied time series analysis II*. [S.l.]: Elsevier, 1981. p. 565–608. Citado na página 26.
- GODARD, D. Self-recovering equalization and carrier tracking in two-dimensional data communication systems. *IEEE transactions on communications*, IEEE, v. 28, n. 11, p. 1867–1875, 1980. Citado na página 27.
- HAYKIN, S. *Sistemas de comunicação*. 4. ed. Nova York: John Wiley Sons, 2001. Acesso em: 21 ago 2001. Citado na página 21.
- HAYKIN, S. S. *Blind deconvolution*. [S.l.]: Prentice Hall, 1994. Citado 5 vezes nas páginas 22, 23, 24, 25 e 26.
- HAYKIN, S. S. *Adaptive filter theory*. [S.l.]: Pearson Education India, 2008. Citado na página 22.
- KENNEDY, R. A.; DING, Z. Blind adaptive equalizers for quadrature amplitude modulated communication systems based on convex cost functions. *Optical Engineering*, International Society for Optics and Photonics, v. 31, n. 6, p. 1189–1200, 1992. Citado 4 vezes nas páginas 25, 26, 27 e 29.
- LABIT, Y.; PEAUCELLE, D.; HENRION, D. Sedumi interface 1.02: a tool for solving lmi problems with sedumi. In: IEEE. *Proceedings. IEEE International Symposium on Computer Aided Control System Design*. [S.l.], 2002. p. 272–277. Citado na página 30.

MANN, J. A.; WALKER, W. A constrained adaptive beamformer for medical ultrasound: Initial results. In: IEEE. *Ultrasonics Symposium, 2002. Proceedings. 2002 IEEE*. [S.l.], 2002. v. 2, p. 1807–1810. Citado na página 23.

NEVES, A. d. O. et al. Uma abordagem unificada para algoritmos de equalização autodidata. [sn], 2001. Citado 4 vezes nas páginas 13, 22, 24 e 25.

NOVELSAT. *NS4*. 2018 (acessado 02 de Abril, 2019). <<http://novelsat.com/novelsat-ns4/>>. Citado na página 21.

SATO, Y. A method of self-recovering equalization for multilevel amplitude-modulation systems. *IEEE Transactions on communications*, IEEE, v. 23, n. 6, p. 679–682, 1975. Citado na página 26.

SHALVI, O.; WEINSTEIN, E. New criteria for blind deconvolution of nonminimum phase systems (channels). *IEEE Transactions on information theory*, IEEE, v. 36, n. 2, p. 312–321, 1990. Citado na página 27.

STURM, J. F. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization methods and software*, Taylor & Francis, v. 11, n. 1-4, p. 625–653, 1999. Citado na página 30.

WIDROW, B. *S. D. Stems, Adaptive Signal Processing*. [S.l.]: Englewood Cliffs, NJ: Prentice-Hall, 1985. Citado 2 vezes nas páginas 22 e 23.