

Otimização Convexa Aplicada à
Redução de Ruído Impulsivo em
Sistemas de Comunicação Acústica
sob a Água

Lucas Silvestre Chaves

Dezembro/2014

**Otimização Convexa Aplicada à
Redução de Ruído Impulsivo em
Sistemas de Comunicação Acústica sob
a Água**

LUCAS SILVESTRE CHAVES

Dissertação apresentada ao Instituto Nacional de Telecomunicações, como parte dos requisitos para obtenção do Título de Mestre em Telecomunicações.

ORIENTADOR: Prof. Dr. Dayan Adionel Guimarães.

Chaves, Lucas Silvestre

C512o

Otimização convexa aplicada à redução de ruído impulsivo em sistemas de comunicação acústica sob a água. / Lucas Silvestre Chaves. – Santa Rita do Sapucaí, 2015.

71p.

Orientador: Prof. Dr. Dayan Adionel Guimarães.

Dissertação de Mestrado – Engenharia de Telecomunicações – Instituto Nacional de Telecomunicações – INATEL.

Inclui bibliografia e anexo.

1. Redução de ruído 2. Otimização convexa 3. Comunicação acústica subaquática 4. Ruído impulsivo 5. Ruído dos camarões de estalo 6. Engenharia de Telecomunicações. I. Guimarães, Dayan Adionel. III. Instituto Nacional de Telecomunicações – INATEL. IV. Título.

CDU 621.39

FOLHA DE APROVAÇÃO

Dissertação defendida e aprovada em ____/____/____,
pela comissão julgadora:

Prof. Dr. Dayan Adionel Guimarães
INATEL

Prof. Dr. Carlos Alberto Ynoguti
INATEL

Prof. Dr. Cássio Guimarães Lopes
USP

Coordenador do Curso de Mestrado
Prof. Dr. José Marcos Câmara Brito

*Sem a certeza do amanhã, o que
nos resta é a certeza de que vale a
pena.*

Lucas Silvestre Chaves

Agradecimentos

Agradeço a Deus, por ter me atendido sempre que precisei, me confortando nas horas mais difíceis e me guiando durante esta caminhada.

Aos meus pais, Elisabete e Raul, pelo amor, dedicação e pelos ensinamentos de vida, através de suas palavras e de seus atos.

Aos meus irmãos, Raul e Felipe, pela união, amizade, incentivo e apoio.

A Patrícia, pelo amor, companheirismo, amizade e por fazer parte de minha vida.

Aos meus amigos e colegas do Inatel, pelas ajudas, apoios e descontrações.

Aos meus amigos e colegas de estudo, Guilherme, Lucas, Mário e Zan, por toda a ajuda concebida durante o desenvolvimento deste trabalho.

Ao Prof. Dr. Dayan Adionel Guimarães, pelo comprometimento, atenção, responsabilidade, orientação e amizade.

Ao Prof. Dr. Rausley Adriano Amaral de Souza, pela atenção, pela ajuda e amizade.

Ao Prof. Dr. José Antônio Justino Ribeiro, pelo exemplo de profissional e de pessoa, por ser um grande amigo e incentivador.

A todos os outros professores do Inatel, que por aulas, ajudas ou conversas, contribuíram com minha formação pessoal e profissional.

A secretária Gisele e funcionários do Inatel em geral, pelos excelentes serviços prestados à Instituição, contribuindo para um estudo mais eficiente e produtivo.

Lucas Silvestre Chaves

Índice

Lista de Figuras	xii
Lista de Tabelas	xiii
Lista de Siglas	xv
Lista de Símbolos	xvii
1 Introdução	1
1.1 Contextualização	1
1.1.1 Ruído	1
1.1.2 Redução de ruído	2
1.1.3 A otimização convexa aplicada à redução de ruídos	3
1.1.4 Redução de ruído em sistemas de comunicação digital	4
1.1.5 Redução de ruído não gaussiano em sistemas UWAC	6
1.2 Contribuições e publicações do autor	8
2 Reconstrução de sinais por otimização convexa	9
2.1 Problemas de otimização convexa	10
2.2 O CVX e o conjunto de regras DCP	14
2.3 Técnicas de reconstrução de sinais via otimização convexa	19
3 Otimização convexa aplicada à redução de ruído impulsivo em sistemas de comunicação sob a água	27
3.1 Sistemas de comunicação sob a água	27
3.2 A técnica COD aplicada a um sistema UWAC	30
3.2.1 O ruído composto	30
3.2.2 Modelo de Sistema UWAC	35
3.2.3 Validação de modelo e formas de onda dos sinais envolvidos na técnica COD	45

4	Resultados numéricos	51
4.1	Configuração da simulação	51
4.1.1	Escolha de M e δ	53
4.2	Desempenho sem redução de ruído	56
4.3	Desempenhos das técnicas END e END modificada	57
4.4	Desempenho da técnica COD	64
5	Conclusões e oportunidades para novas pesquisas	67
5.1	Conclusões	67
5.2	Oportunidades para novas pesquisas	68
A	Simulação do sistema no MATLAB	72

Lista de Figuras

1.1	<i>Sistema de comunicação digital em banda passante com modulação BPSK coerente com detector ótimo para canal AWGN.</i>	5
2.1	<i>Curva de compromisso ótima entre os objetivos $\ \mathbf{Ax} - \mathbf{b}\ _2^2$ e $\ \mathbf{x}\ _2^2$.</i>	13
2.2	<i>Função de penalização quadrática para $r \leq 2$.</i>	21
2.3	<i>Funções de penalização quadrática e de Huber com $M = 0.5$.</i>	23
2.4	<i>Soluções resultantes de problemas de regressão linear.</i>	23
2.5	<i>Fluxograma da solução do problema de otimização da técnica COD.</i>	25
2.6	<i>Exemplos de: a) vetor \mathbf{s}, b) vetor \mathbf{y} e c) vetor $\hat{\mathbf{s}}$.</i>	26
3.1	<i>PSD do ruído ambiente (de [14], reproduzida com permissão).</i>	32
3.2	<i>Melhor ajuste $S\alpha S$ (de [14], reproduzida com permissão).</i>	34
3.3	<i>(a) PDFs $S\alpha S$ com $c = 0, 1178$. (b) forma de onda do ruído composto.</i>	35
3.4	<i>Modelo de sistema adotado para a análise dos desempenhos.</i>	39
3.5	<i>Exemplo de forma de onda do vetor de bits \mathbf{b}.</i>	39
3.6	<i>Exemplo de forma de onda do vetor de símbolos \mathbf{d}.</i>	40
3.7	<i>Exemplo de forma de onda do vetor de símbolos desvanecidos.</i>	40
3.8	<i>Exemplo de forma de onda do sinal amostrado e filtrado.</i>	41
3.9	<i>Exemplo de forma de onda do sinal transmitido \mathbf{s}.</i>	41
3.10	<i>Exemplo de forma de onda do ruído \mathbf{r} sem impulsividade.</i>	42
3.11	<i>Exemplo de forma de onda do sinal \mathbf{y}.</i>	42
3.12	<i>Exemplo de forma de onda do sinal convertido para a banda base.</i>	43
3.13	<i>Exemplo de forma de onda do sinal recebido filtrado.</i>	43
3.14	<i>Exemplo de forma de onda do vetor de símbolos estimados $\hat{\mathbf{d}}$.</i>	44
3.15	<i>Exemplo de forma de onda do vetor de bits estimados $\hat{\mathbf{b}}$.</i>	44
3.16	<i>Validação do modelo adotado.</i>	46
3.17	<i>Diagrama de olho obtido na saída do filtro de recepção.</i>	47
3.18	<i>Formas de onda para $E_b/N_0 = 7$ dB: (a) \mathbf{s}, (b) \mathbf{y} e (c) $\hat{\mathbf{s}}$.</i>	47
3.19	<i>Formas de onda para $E_b/N_0 = 21$ dB: (a) \mathbf{s}, (b) \mathbf{y} e (c) $\hat{\mathbf{s}}$.</i>	48
3.20	<i>Formas de onda para $E_b/N_0 = 35$ dB: (a) \mathbf{s}, (b) \mathbf{y} e (c) $\hat{\mathbf{s}}$.</i>	48

4.1	<i>Curvas $MSE \times \delta$ para $E_b/N_0 = 7$ dB.</i>	54
4.2	<i>Curvas $MSE \times \delta$ para $E_b/N_0 = 21$ dB.</i>	54
4.3	<i>Curvas $MSE \times \delta$ para $E_b/N_0 = 35$ dB.</i>	55
4.4	<i>Desempenho do sistema para $1,5 \leq \alpha \leq 2$, sem redução de ruído.</i>	56
4.5	<i>Desempenho do sistema com redução de ruído pela técnica END para $F = 3; 2; 1,5; 1$ e $\alpha = 1,5$.</i>	59
4.6	<i>Desempenho do sistema com redução de ruído pela técnica END para $F = 3; 2; 1,5; 1$ e $\alpha = 1,7$.</i>	60
4.7	<i>Desempenho do sistema com redução de ruído pela técnica END para $F = 3; 2; 1,5; 1$ e $\alpha = 1,9$.</i>	60
4.8	<i>Desempenho do sistema com redução de ruído pela técnica END modificada para $F = 3; 2; 1,5; 1$ e $\alpha = 1,5$ com o cálculo de ρ para $i = 200 \dots 1300$.</i>	62
4.9	<i>Desempenho do sistema com redução de ruído pela técnica END modificada para $F = 3; 2; 1,5; 1$ e $\alpha = 1,7$ com o cálculo de ρ para $i = 200 \dots 1300$.</i>	63
4.10	<i>Desempenho do sistema com redução de ruído pela técnica END modificada para $F = 3; 2; 1,5; 1$ e $\alpha = 1,9$ com o cálculo de ρ para $i = 200 \dots 1300$.</i>	63
4.11	<i>Desempenho das técnicas COD com $M = 0,01$ e $\delta = 1$, END modificada com $F = 1$ e ρ com $i = 200 \dots 1300$ e END com $F = 3$ para $\alpha = 1,5$.</i>	65
4.12	<i>Desempenho das técnicas COD com $M = 0,01$ e $\delta = 1$, END modificada com $F = 1$ e ρ com $i = 200 \dots 1300$ e END com $F = 3$ para $\alpha = 1,7$.</i>	66
4.13	<i>Desempenho das técnicas COD com $M = 0,01$ e $\delta = 1$, END modificada com $F = 1$ e ρ com $i = 200 \dots 1300$ e END com $F = 3$ para $\alpha = 1,9$.</i>	66

Lista de Tabelas

3.1	<i>Largura de banda disponível para diferentes alcances em canais UWAC.</i>	28
3.2	<i>Ganhos de MSE proporcionados pela redução do ruído composto. . .</i>	49
4.1	<i>Valores práticos e simulados dos parâmetros do sistema adotado. . .</i>	52

Lista de Siglas

UWAC	<i>Underwater acoustic communications</i>
MSE	<i>Mean squared error</i>
NLP	<i>Nonlinear programming</i>
LS	<i>Least squares</i>
LP	<i>Linear programming</i>
QP	<i>Quadratic programming</i>
DCP	<i>Disciplined convex programming</i>
BPSK	<i>Binary phase shift keying</i>
AWGN	<i>Additive white Gaussian noise</i>
ISI	<i>Intersymbol Interference</i>
SNR	<i>Signal-to-noise ratio</i>
BER	<i>Bit error rate</i>
AWNGN	<i>Additive white non-Gaussian noise</i>
UC	<i>Up converter</i>
DC	<i>Down converter</i>
S α S	<i>Symmetric α stable</i>
COD	<i>Convex optimization denoising</i>
PSD	<i>Power spectral density</i>
END	<i>Empirical nonlinear-device denoising</i>
SDP	<i>Semidefinite programming</i>
LMI	<i>Linear matricial inequality</i>
MFSK	<i>Frequency-shift keying</i>
MPSK	<i>Phase-shift keying</i>
DPSK	<i>Differential phase-shift keying</i>
QAM	<i>Quadrature amplitude modulation</i>
DSSS	<i>Direct-sequence spread-spectrum</i>
DS-CDMA	<i>Direct-sequence code-division multiple-access</i>
OFDM	<i>Orthogonal frequency-division multiplexing</i>
MIMO	<i>Multiple-input multiple-output</i>
DFE	<i>Decision feedback equalizers</i>
E_b/N_0	<i>Average energy per bit / noise power spectral density</i>
PDF	<i>Probability density function</i>
RRC	<i>Root-raised-cosine</i>
IS	<i>Inverse sinc</i>

Lista de Símbolos

\mathbf{s}	Vetor de sinal não contaminado por ruído
n	Número de amostras dos vetores envolvidos no processo de redução
\mathbf{r}	Vetor de ruído aditivo
\mathbf{y}	Vetor de sinal contaminado por ruído
$\hat{\mathbf{s}}$	Vetor de sinal estimado
\mathbf{x}	Variável (ou vetor de variáveis) do problema de otimização
f_0	Função objetivo do problema de otimização
f_i	Função associada à i -ésima restrição de desigualdade do problema de otimização
m	Número de funções de desigualdade do problema de otimização
h_i	Função associada à i -ésima restrição de igualdade do problema de otimização
p	Número de funções de igualdade do problema de otimização
\mathbb{R}^n	Espaço dos vetores de n elementos reais
\mathbb{R}	Espaço dos números reais
$f : \mathbb{A} \rightarrow \mathbb{B}$	Domínio e imagem de f pertencentes aos conjuntos \mathbb{A} e \mathbb{B} respectivamente
\mathbf{x}^*	Ponto ótimo ou solução do problema de otimização
v^*	Valor ótimo do problema de otimização
\mathbf{b}	Vetor de bits transmitidos
\mathbf{d}	Vetor de símbolos transmitidos
$\hat{\mathbf{b}}$	Vetor de bits estimados
$\hat{\mathbf{d}}$	Vetor de símbolos estimados
$s(t)$	Sinal transmitido
$h(t)$	Resposta ao impulso do canal de comunicação
$r(t)$	Ruído térmico
T	Duração do símbolo
M	Fator de forma da função de <i>Huber</i>
	Tal que
\mathbb{S}_{++}^n	Espaço das matrizes definidas positivas de ordem n
\mathbb{S}_+^n	Espaço das matrizes semi definidas positivas de ordem n
\mathbb{H}^n	Espaço das matrizes Hermitianas de ordem n
\mathbb{R}_{++}	Espaço dos números reais positivos
$\mathbb{R}^{m \times n}$	Espaço das matrizes de elementos reais e ordem $m \times n$

X_{oti}	Conjunto dos pontos ótimos de uma função
$\ \cdot\ _2$	Norma euclidiana
\in	Pertence
$\mathbf{q} \preceq \mathbf{w}$	$q_i \leq w_i$ para $i = 1, \dots, n$ em que $\mathbf{q}, \mathbf{w} \in \mathbb{R}^n$
\cap	Intersecção
\subseteq	Contido propriamente
$\text{dom}f$	Domínio de f
z	Define o conjunto subnível de uma função convexa
\mathbb{S}^n	Espaço das matrizes simétricas de ordem n
$\mathbf{P} \succeq 0$	A matriz \mathbf{P} é semi definida positiva
$\mathbf{P} \preceq 0$	A matriz \mathbf{P} é semi definida negativa
D	Domínio de um problema de otimização convexa
C	Conjunto de pontos factíveis de um problema de otimização convexa
F_1, F_2	Objetivos do problema de otimização bicritério escalarizado
λ_1	Parâmetro de escalarização positivo, correspondente ao peso dado a F_1
λ_2	Parâmetro de escalarização positivo, correspondente ao peso dado a F_2
λ	$\lambda = (\lambda_1, \lambda_2) \in \mathbb{R}_{++}^2$ é o vetor de parâmetros de escalarização
μ	Peso relativo ou importância relativa dada por $\mu = \lambda_2/\lambda_1$
δ	Peso relativo ou parâmetro de escalarização do problema de reconstrução de sinais
ϕ_{l2}	Função pênalti quadrática
ϕ_{suav}	Função de regularização ou objetivo de suavização
ϕ_{hub}	Função pênalti de <i>Huber</i>
ϕ_{quad}	Objetivo de suavização quadrática
ϕ_{tv}	Objetivo de suavização por variação total
α	Expoente característico da distribuição $S\alpha S$
β	Parâmetro de simetria da distribuição $S\alpha S$
γ	Parâmetro de dispersão da distribuição $S\alpha S$
a	Parâmetro de localização da distribuição $S\alpha S$
c	Parâmetro de escala da distribuição $S\alpha S$
P	Potência média do sinal recebido
R_b	Taxa de bits
BW	Largura de faixa
f_a	Frequência de amostragem
f_p	Frequência da portadora
t	Limiar de corte do dispositivo não linear
σ	Desvio padrão da distribuição Gaussiana
η	Fator de desvanecimento
ρ	Potência média de séries temporais das amostras recebidas em cada quadro
F	Fator de calibração

Resumo

Em águas marítimas quentes e rasas é comum a presença de um ruído acústico de natureza impulsiva gerado por camarões de estalo. Este ruído degrada o desempenho de sistemas de comunicação acústica subaquática se os detectores de tais sistemas são projetados apenas para canais com ruído Gaussiano puro. Com objetivo de aumentar a robustez desses sistemas frente ao ruído de camarão de estalo, propõe-se nesta dissertação uma técnica de redução de ruído impulsivo baseada em otimização convexa. O sistema de comunicação subaquática foi simulado no Matlab, sendo que a técnica de redução de ruído proposta foi tratada pelo CVX, que é um sistema de modelagem e solução de problemas de otimização convexa desenvolvido para o Matlab. Técnicas de redução de ruído impulsivo via detecção não linear também foram analisadas para fins de comparação. Os resultados mostram uma significativa melhoria no desempenho do sistema com a redução do ruído realizada pela técnica proposta para todos os cenários considerados, sem a necessidade de estimação ou adaptação de qualquer parâmetro. Em comparação com a redução de ruído via detecção não linear, a técnica proposta mostrou-se mais eficaz para altos valores de relação sinal-ruído em todos os cenários considerados.

Palavras-chave: redução de ruído; otimização convexa; CVX; reconstrução de sinais; detecção não linear; comunicação acústica subaquática; ruído impulsivo; ruído dos camarões de estalo;

Abstract

In warm shallow waters it is common the presence of an impulsive-like acoustic noise generated by snapping shrimps. This noise degrades the performance of underwater acoustic communication systems if the detectors of these systems are designed for the pure Gaussian noise channel. Aiming at improving the robustness of such systems against the snapping shrimp noise, this dissertation proposes a noise reduction technique based on convex optimization. The underwater acoustic communication system was simulated using the Matlab software tool, and the proposed noise reduction technique was treated by the CVX, which is a system for modeling and solving convex optimization problems, developed to work under the Matlab platform. Techniques for noise reduction based on nonlinear detectors were also considered for comparison purposes. The results unveil a significant performance improvement brought by the proposed technique in all considered scenarios, with no need for adaptation or knowledge of any parameter. When compared with the nonlinear detector, the proposed noise reduction has unveiled more efficacy in the high signal-to-noise regime, again for all considered scenarios.

Keywords: denoising; convex optimization; CVX; signal reconstruction; nonlinear detection; underwater acoustic communication; impulsive noise; snapping shrimp noise;

Capítulo 1

Introdução

1.1 Contextualização

Neste capítulo, contextualiza-se inicialmente o problema de redução de ruído de forma genérica e o problema da redução do ruído em sistemas de comunicação acústica subaquática. Em seguida, registram-se as contribuições da pesquisa e as correspondentes publicações.

1.1.1 Ruído

Ruído pode ser definido como qualquer sinal não desejado que interfere em um sinal de informação durante a comunicação, medida ou processamento [1]. Existem diversos tipos de ruídos em diferentes níveis e em quase todos os meios. Durante uma simples conversa entre duas pessoas na rua, podem existir ruídos acústicos que dificultam a comunicação como, por exemplo, a conversa de outras pessoas e o som das buzinas e dos motores de carros passando. O processo de conversão de um sinal analógico para um sinal digital tem como consequência a adição de um ruído de processamento no sinal digital, introduzido durante o processo de quantização e conhecido como ruído de quantização. Em imagens digitais é comum a presença de ruído impulsivo, conhecido como ruído sal e pimenta [2]. A interferência deste ruído ocorre em apenas alguns pixels isolados na imagem original, tendo como consequência a aparição de pontos pretos e brancos dispersos na imagem corrompida, o que justifica o nome dado. Em um sistema de telefonia móvel celular digital podem existir diferentes tipos de ruídos, tais como o ruído térmico, o ruído impulsivo, o ruído de rádio frequên-

cia eletromagnético e o ruído de processamento. Em sistemas UWAC (*underwater acoustic communications*), além do ruído térmico e do ruído de processamento podem existir ruídos acústicos gerados por diferentes fontes como a quebra das ondas, chuva, vida marinha, embarcações e, em águas quentes e rasas, camarões de estalo (do inglês *snapping shrimp*) gerando ruído impulsivo [3–6]. A presença dos ruídos em sistemas de comunicação corrompem o sinal transmitido, dificultando a recuperação da informação. Como consequência, tem-se uma queda na qualidade da comunicação.

1.1.2 Redução de ruído

Para atenuar a influência dos ruídos aditivos em um sinal corrompido é preciso elaborar uma estratégia de redução de ruídos (do inglês *noise reduction* ou *denoising*) que não distorça o sinal de informação, ou distorça o mínimo possível. Os estudos das estratégias de redução de ruído dizem respeito a um importante campo de aplicação da área de processamento de sinais.

Matematicamente, seja s um vetor contendo n amostras de um sinal não corrompido, r um vetor contendo n amostras de um ruído aditivo e y o vetor de n amostras corrompidas dado por

$$y = s + r. \quad (1.1)$$

O objetivo da redução de ruído é estimar o vetor s a partir do vetor y com base em características conhecidas do vetor r . O vetor estimado de s é denotado por \hat{s} .

Para medir o desempenho de uma técnica de redução de ruído, pode ser utilizada a medida conhecida como erro quadrático médio (MSE, *mean squared error*). O MSE é a média dos quadrados dos erros entre as amostras do sinal estimado (\hat{s}) e as amostras do sinal alvo (s), e é dado por

$$e_{\text{MSE}} = \frac{1}{n} \sum_{i=1}^n (\hat{s}_i - s_i)^2. \quad (1.2)$$

Quanto menor o valor de e_{MSE} , melhor é o desempenho da técnica analisada. Um valor de e_{MSE} igual a zero implicaria em uma estimação perfeita, em que o vetor estimado \hat{s} seria exatamente igual ao vetor s .

Várias técnicas de redução de ruído foram desenvolvidas para diferentes aplicações, mas pesquisadores continuam concentrando esforços na criação de novas estratégias, objetivando obter desempenhos cada vez melhores. Um artigo de 2010 no campo de

processamento de imagens motiva a busca de estratégias de redução de ruído mais eficazes. Ele possui o título *Is Denoising Dead?* [7] e reporta um estudo com o objetivo de responder seu próprio título: a redução de ruído está morta? A motivação desse estudo foi o fato de estratégias de redução de ruído propostas sob diferentes abordagens, na época, terem apresentado desempenhos comparáveis com as melhores técnicas de redução existentes até então. O resultado do estudo mostrou que a área de redução de ruído não está "morta". Pelo contrário, ainda há muito o que melhorar.

A partir de estudos realizados sobre otimização convexa [8], sub-área da Matemática Aplicada, desenvolveu-se uma técnica de redução de ruído através da modelagem de um problema de otimização convexa.

1.1.3 A otimização convexa aplicada à redução de ruídos

Um problema no qual se busca maximizar ou minimizar uma função real (função objetivo) de variáveis reais ou inteiras, sujeito a restrições sobre essas variáveis, é um problema de otimização matemática [8]. A formulação geral padrão dessa classe de problemas é

$$\begin{aligned} & \text{minimize} && f_0(\mathbf{x}) \\ & \text{sujeito a} && f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \quad , \\ & && h_i(\mathbf{x}) = 0, \quad i = 1, \dots, p \end{aligned} \quad (1.3)$$

sendo $\mathbf{x} = (x_1, x_2, \dots, x_n)$ a variável (ou vetor de variáveis) do problema de otimização, $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ a função objetivo, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ com $i = 1, 2, \dots, m$ as funções associadas às restrições de desigualdade e $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$ com $i = 1, 2, \dots, p$ as funções associadas às restrições de igualdade. O vetor \mathbf{x}^* é chamado de ponto ótimo ou solução do problema de otimização, uma vez que, dentre os vetores que satisfazem as restrições, é com ele que a função objetivo apresenta menor valor, ou seja, para qualquer \mathbf{z} tal que $f_1(\mathbf{z}) \leq 0, \dots, f_m(\mathbf{z}) \leq 0$ e $h_1(\mathbf{z}) = 0, \dots, h_p(\mathbf{z}) = 0$ tem-se $f_0(\mathbf{z}) \geq f_0(\mathbf{x}^*)$. O valor ótimo da função correspondente ao ponto ótimo \mathbf{x}^* é dado por $v^* = f_0(\mathbf{x}^*)$. Um problema de otimização convexa corresponde ao Problema (1.3) com a função objetivo e as funções restrições todas convexas [8]. A otimização convexa é uma sub classe da programação não linear (NLP, *nonlinear programming*) que unifica e generaliza o método dos mínimos quadrados (LS, *least squares*), a programação linear (LP, *linear programming*) e a programação quadrática (QP, *quadratic programming*) [9].

Apesar de suas bases teóricas serem anteriores, a otimização convexa experimentou grande desenvolvimento a partir das últimas décadas do Século XX, com o desenvol-

vimento de algoritmos baseados no conceito de pontos interiores. A generalização desses algoritmos propiciou o surgimento de métodos confiáveis e eficientes para a solução de uma grande variedade de problemas de otimização, utilizando as interessantes propriedades dos conjuntos convexos.

A disseminação da otimização convexa foi por muito tempo dificultada por exigir do usuário um grande conhecimento sobre análise convexa e sobre algoritmos numéricos. A partir da metodologia de modelagem proposta por Michael Grant, Stephen Boyd, e Yinyu Ye em [9, 10], designada por programação convexa disciplinada (DCP, *disciplined convex programming*), essa dificuldade foi consideravelmente reduzida. A DCP estabelece um conjunto de regras que devem ser seguidas na construção e solução de um problema de otimização. O problema assim construído pode ser automaticamente convertido para uma forma solucionável através da adaptação de métodos numéricos comuns da área de otimização. Desta forma, o usuário pode concentrar todo seu esforço apenas na modelagem do problema, sem a necessidade de se preocupar com os detalhes matemáticos relacionados à sua solução.

A técnica aqui proposta é designada por COD (*convex optimization denoising*) é uma estratégia de redução de ruídos baseada na modelagem de um problema de otimização convexa sem restrições, tal como

$$\text{minimize } f_0(\mathbf{x}; \mathbf{y}, \delta), \quad (1.4)$$

em que $\mathbf{x} \in \mathbb{R}^n$ é a variável de otimização, $\mathbf{y} \in \mathbb{R}^n$ é o parâmetro de entrada correspondente ao vetor de sinal corrompido, dado por $\mathbf{y} = \mathbf{s} + \mathbf{r}$ (1.1), e $\delta \in \mathbb{R}_{++}$ é chamado de parâmetro de escalarização. No contexto de redução de ruído, o ponto ótimo \mathbf{x}^* corresponde ao sinal estimado $\hat{\mathbf{s}}$. Diferentes valores de δ implicam em diferentes problemas de otimização convexa, podendo resultar em diferentes valores de $\hat{\mathbf{s}}$ para um mesmo \mathbf{y} . O método adotado para a escolha de δ foi encontrar aquele que apresente o menor MSE entre o sinal estimado $\hat{\mathbf{s}}$ e o sinal livre de ruído \mathbf{s} para um dado vetor \mathbf{y} .

1.1.4 Redução de ruído em sistemas de comunicação digital

Todo sistema de comunicação digital é constituído por uma unidade de transmissão e uma unidade de recepção, que são conectadas por um canal de comunicação. Um possível sistema de comunicação digital em banda passante, com modulação BPSK (*binary phase shift keying*) coerente e projetado para canal AWGN (*additive white gaussian noise*) [11] é mostrado na Figura 1.1. Neste sistema, o sinal BPSK filtrado

transmitido $s(t)$ sofre influência do canal de propagação com resposta ao impulso $h(t)$ e é corrompido pelo ruído térmico $r(t)$. A etapa de filtragem na entrada do receptor, realizada por um correlator ou um filtro casado, reduz a energia do ruído térmico presente no sinal recebido. Na frequência, esta parcela de energia eliminada corresponde à banda do ruído fora da faixa de passagem do filtro. Se a banda de passagem do canal é menor do que a banda do sinal, ocorre uma distorção na frequência e uma dispersão temporal no sinal, podendo causar a interferência inter simbólica (ISI, *intersymbol interference*) [11]. Quanto menor a banda do canal em relação a banda do sinal, maior será o nível de ISI causado. Após a filtragem de recepção, considerando-se o perfeito casamento entre os filtros, obtém-se a maximização da relação sinal ruído (SNR, *signal-to-noise ratio*) no instante de decisão. Desta forma, a amostragem e retenção deve ser feita a cada múltiplo de T (tempo de símbolo) para garantir que o processo de decisão opere com máxima SNR.

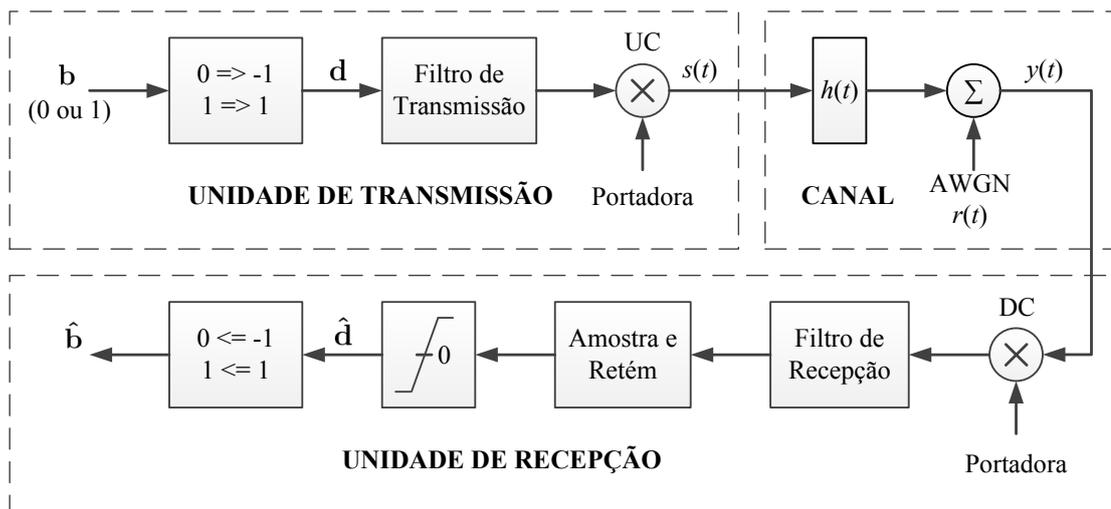


Figura 1.1: Sistema de comunicação digital em banda passante com modulação BPSK coerente com detector ótimo para canal AWGN.

Substituindo o ruído AWGN do sistema da Figura 1.1 por um ruído aditivo branco não Gaussiano (AWNGN, *additive white non-gaussian noise*), verifica-se uma relevante piora de desempenho. Isto ocorre porque o receptor que era ótimo para o caso do ruído AWGN passou a ser sub-ótimo para o caso do ruído AWNGN. Para diminuir a degradação do desempenho proporcionada pelo ruído AWNGN é necessário realizar uma redução de ruído adicional. Como a redução de ruído proporcionada pela etapa de filtragem ocorre fora da banda do sinal transmitido, esta redução adicional deve ser realizada dentro da banda do sinal. Esta nova etapa de redução pode ser incluída antes do filtro de recepção. A estratégia de redução de ruído será o quanto melhor quanto maior for a redução realizada com a menor distorção do sinal desejado. Desta forma,

quanto melhor for a estratégia adotada, maior será a melhora do desempenho do sistema. A técnica de redução de ruído proposta nesta dissertação é aplicada na redução do ruído não gaussiano presente em sistemas UWAC.

1.1.5 Redução de ruído não gaussiano em sistemas UWAC

A alta atenuação de ondas eletromagnéticas, especialmente pela água do mar, é um dos principais fatores que limitam a utilização de ondas de rádio para comunicações subaquáticas. Por outro lado, estas atenuações são significativamente menores para ondas acústicas, motivando o uso de sinais acústicos em sistemas de comunicação sob a água [12–14].

Sistemas de comunicação acústica subaquática, ou sistemas UWAC, são usados em muitas aplicações tais como navegação sonora e detecção de eco (sonar), comunicação entre mergulhadores, comunicação entre mergulhadores e embarcações, comunicação entre uma embarcação ou uma base da superfície com um submarino ou outros veículos subaquáticos de exploração marítima, comunicações entre submarinos, em telemetria e em sistemas de controle subaquático. Aplicações relacionadas a redes de sensores subaquáticos são umas das mais recentes [15–18]. A principal característica dos sistemas UWAC é o uso de ondas de portadoras acústicas, com frequências geralmente acima de 5 kHz. O limite superior depende do alcance. Por exemplo, para um alcance de 10 km este limite superior está em volta de 10 kHz. Para um alcance de 100 m, sinais até 100 kHz podem ser utilizados. Alguns modernos sistemas UWAC de alta velocidade para comunicações de curto alcance utilizam portadoras acústicas com frequências ainda maiores. As potências de transmissão geralmente são da ordem de 30 W ou menos. As taxas de transmissão alcançadas são de até poucos kb/s em uma distância de cerca de 10 km.

Os danos à comunicação encontrados em canais acústicos subaquáticos são ainda mais severos do que aqueles encontrados em canais de rádio sem fio, representando limitações desafiadoras para os projetos de sistemas UWAC. Os principais fatores limitantes são o atraso de propagação, a atenuação e o ruído ambiente, todos de forma aleatória.

O canal UWAC é sujeito a uma variabilidade tridimensional do índice de refração. Esta é uma das principais causas de múltiplos percursos e sombreamento. A propagação por múltiplos percursos também é causada por reflexões na superfície e no fundo do mar ou rio. Pequenos tempos de coerência combinados com grandes atrasos de

propagação também impõem grandes limitações ao projeto do sistema. Por exemplo, em um alcance de 1 km o sinal pode levar até 0,66 segundos para atingir o receptor. Por outro lado, as características do canal podem mudar drasticamente dentro de 0,5 segundo ou menos. Os atrasos entre percursos podem chegar a até 60 ms, causando severo desvanecimento seletivo em frequência. O efeito Doppler e o espalhamento Doppler também aparecem e são causados devido à movimentação das ondas de superfície e ondas internas e devido à movimentação relativa entre o receptor e o transmissor. Além disso, a velocidade do som varia dentro da água, principalmente com a profundidade, dependendo também das características aquáticas e das condições do clima.

Os principais mecanismos de perda de sinal dentro de canais UWAC são a distribuição da energia com a distância, similar ao que ocorre em sistemas de radiocomunicações, absorção e espalhamento. A absorção é a conversão de energia acústica em calor e é um fenômeno dependente das propriedades químicas da água e da frequência do sinal. O espalhamento ocorre na superfície e no fundo do mar ou rio, mas também sobre bolhas de ar e sobre pequenos volumes de água com diferentes temperaturas. Além disso, uma vez que a SNR depende do alcance da comunicação e da frequência do sinal, maiores alcances significam menores larguras de banda disponíveis para transmissão de dados.

Além do ruído térmico gerado no receptor, existem ainda os ruídos acústicos típicos encontrados em ambientes marítimos onde operam sistemas UWAC, gerados por quebras de ondas, chuva, vida marinha e embarcações. Entre estes ruídos, especificamente em águas quentes e rasas, é comum a existência de um ruído com característica impulsiva. Este ruído é resultado dos estalos provocados pelo fechamento do quelópode mais desenvolvido dos camarões de estalo [3–6].

O ruído composto pela adição do ruído térmico ao ruído ambiente impulsivo degrada o desempenho de sistemas UWAC com detectores ótimos para canais AWGN. Nesta dissertação, descreve-se e utiliza-se a técnica COD como primeira etapa da recepção de um sistema UWAC com detector ótimo para canal AWGN, com objetivo de reduzir o ruído composto presente no canal e melhorar o desempenho do sistema.

1.2 Contribuições e publicações do autor

Este trabalho apresenta um novo método de redução de ruídos impulsivos utilizando otimização convexa, designado por técnica COD, em que a redução é realizada sem a necessidade de se estimar qualquer parâmetro. O método proposto foi aplicado em um modelo de sistema baseado em sistemas UWAC reais localizados em águas marítimas quentes e rasas, com finalidade de combater o ruído composto resultante da adição do ruído térmico com o ruído impulsivo gerado por camarões de estalo. Além disso, foram também analisadas técnicas de redução de ruído impulsivo via dispositivos não lineares, denominadas END (*empirical nonlinear-device denoising*) e END modificada. Esta última também é uma contribuição do trabalho. Foram geradas curvas de desempenhos da técnica COD e das técnicas END e END modificada para os ruídos compostos com diferentes níveis de impulsividade. Os resultados mostraram uma significativa melhoria no desempenho.

Os seguintes artigos foram elaborados como consequência dos trabalhos de pesquisa relacionados ao tema desta dissertação:

D. A. Guimarães, L. S. Chaves, and R. A. A. Souza, "Snapping Shrimp Noise Reduction using Convex Optimization for Underwater Acoustic Communication in Warm Shallow Water", *International Telecommunications Symposium (ITS)*, São Paulo, Brazil, 17 – 20 August 2014.

L. S. Chaves, G. H. F. Floriano, and D. A. Guimarães, "Um Tutorial Sobre a Aplicação do CVX na Solução de Problemas de Otimização Convexa", *IEEE Latin America Transactions*. Submetido em fevereiro de 2014.

Capítulo 2

Reconstrução de sinais por otimização convexa

Este capítulo inicia-se com a apresentação da forma padrão de um problema de otimização convexa e de conceitos básicos sobre estes problemas. Na sequência descreve-se o problema de otimização convexa do tipo bicritério escalarizado sem restrições, que é o tipo de problema utilizado no desenvolvimento da técnica COD. Em seguida é apresentado o sistema de modelagem e resolução de problemas de otimização convexa, conhecido como CVX, e o conjunto de regras da programação convexa disciplinada. Na sequência são apresentados os códigos no CVX referentes aos problemas de otimização convexa abordados anteriormente. Por fim são descritas algumas técnicas de reconstrução de sinais via otimização convexa, a técnica COD e a sua modelagem no CVX.

Mais detalhes sobre a otimização convexa tais como os conjuntos e as funções convexas foram aqui omitidos por questão de objetividade e podem ser encontrados em [8] e no artigo tutorial citado ao final da Seção 1.2 .

2.1 Problemas de otimização convexa

Como descrito resumidamente na Subsecção 1.1.3, um problema de otimização convexa em sua forma padrão é assim descrito:

$$\begin{aligned} & \text{minimize} && f_0(\mathbf{x}) \\ & \text{sujeito a} && f_i(\mathbf{x}) \leq 0, && i = 1, \dots, m, \\ & && h_i(\mathbf{x}) = \mathbf{a}_i^T \mathbf{x} - b_i = 0, && i = 1, \dots, p \end{aligned} \quad (2.1)$$

em que f_0, \dots, f_m são funções convexas, h_i são funções afins, e o domínio do problema $D = \bigcap_{i=0}^m \text{dom}(f_i)$ é convexo. O conjunto de pontos factíveis C de um problema de otimização convexa é convexo, uma vez que ele é definido pela intersecção entre m conjuntos subníveis $\{\mathbf{x} \mid f_i(\mathbf{x}) \leq 0\}$ e p hiperplanos $\{\mathbf{x} \mid \mathbf{a}_i^T \mathbf{x} - b_i = 0\}$. A solução do problema de otimização convexa resulta em um conjunto de pontos ótimos X_{oti} dado por $X_{\text{oti}} = \{\mathbf{x} \mid \mathbf{x} \in D, \mathbf{x} \in C, f_0(\mathbf{x}) = v^*\}$. Caso $f_0(\mathbf{x})$ seja uma função estritamente convexa ou por consequência da restrição imposta por C , X_{oti} se limitará a apenas um ponto, correspondente a \mathbf{x}^* . É permitido que v^* assuma os valores estendidos $\pm\infty$. Se $v^* = +\infty$, diz-se que o problema é infactível e o conjunto de pontos factíveis C é vazio. Se existem pontos factíveis x_k com $f_0(x_k) \rightarrow -\infty$ com $k \rightarrow \infty$, desta forma $v^* = -\infty$ e então diz-se que o problema é ilimitado para baixo. Outra possível situação é o caso de existirem pontos factíveis x_k com $f_0(x_k) \rightarrow a$ com $k \rightarrow \infty$ e $a \in \mathbb{R}$, desta forma $v^* = a$ não será alcançável. Como exemplos, considere os problemas onde se deseja minimizar $f_0(x) = -\log x$ ou $f_0(x) = 1/x$, para $x \in \mathbb{R}$ com $\text{dom} f_0 = \mathbb{R}_{++}$. No problema em que deseja-se minimizar $f_0(x) = -\log x$, o valor de $v^* = -\infty$ não é alcançável, caracterizando o problema como ilimitado para baixo. No problema em que deseja-se minimizar $f_0(x) = 1/x$, o valor de $v^* = 0$ também não é alcançável, por maior que seja o valor de x . Em ambos os problemas, o conjunto solução X_{oti} será vazio.

Um tipo de problema de otimização convexa é o bicritério escalarizado sem restrições [8], dado por

$$\text{minimize} \quad f_0(\mathbf{x}; \lambda) = \lambda^T (F_1(\mathbf{x}), F_2(\mathbf{x})) = \lambda_1 F_1(\mathbf{x}) + \lambda_2 F_2(\mathbf{x}), \quad (2.2)$$

em que $\mathbf{x} \in \mathbb{R}^n$ é a variável de otimização, $\lambda = (\lambda_1, \lambda_2) \in \mathbb{R}_{++}^2$ é o vetor de parâmetros de escalarização e $F_1, F_2 : \mathbb{R}^n \rightarrow \mathbb{R}$ são os objetivos do problema (funções convexas) que, ponderados pelos seus respectivos parâmetros de escalarização (λ_1 e λ_2) e posteriormente somados, constituem a função objetivo $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ convexa.

Os parâmetros de escalarização podem ser interpretados como pesos dados aos objetivos do problema, uma vez que cada parâmetro pondera seu respectivo objetivo. Os objetivos do problema podem ser concorrentes entre si ou não: quando F_1 e F_2 são concorrentes, existe uma relação de compromisso entre eles de tal forma que, para que se tenha F_1 menor dentro da solução do problema, maior deverá ser o valor de F_2 na solução e vice-versa. Por outro lado, se não existe concorrência entre F_1 e F_2 , não existe nenhuma relação de compromisso entre eles de tal forma que o valor de F_1 na solução do problema deve ser o menor possível independente do valor de F_2 e vice-versa. Quando não há concorrência entre os objetivos, a solução do problema resultará em um único ponto ótimo, independente do vetor λ escolhido. Quando há concorrência entre os objetivos, os pesos a eles atribuídos relacionam-se com seus respectivos valores na solução do problema de otimização da seguinte forma: quanto maior for o peso λ_i dado a seu respectivo objetivo F_i , menor será o valor de F_i na solução do problema; quanto menor for o peso λ_i , maior será o valor de F_i na solução. Neste caso, para cada λ escolhido pode-se ter um ponto ótimo diferente. Estes pontos ótimos são conhecidos como pontos ótimos de Pareto [8], em que cada ponto ótimo de Pareto \mathbf{x}^* tem seu respectivo valor ótimo de Pareto, dado por $(F_1(\mathbf{x}^*), F_2(\mathbf{x}^*))$. O conjunto de todos os valores ótimos de Pareto corresponde à curva de compromisso ótima (*optimal trade-off curve*) entre os objetivos F_1 e F_2 . Segue um exemplo de problema do tipo bi-critério escalarizado sem restrições. Nele está apresentado um método de obtenção da curva de compromisso ótima através da variação da razão $\mu = \lambda_2/\lambda_1$, conhecida como peso relativo ou importância relativa entre os objetivos do problema. Este exemplo será útil ao entendimento da técnica de redução de ruído proposta nesta dissertação.

Mínimos quadrados (LS, *Least-Squares*) regularizado: Antes de se entender um problema do tipo LS regularizado é conveniente que se entenda um problema do tipo LS, dado por

$$\text{minimize } \|\mathbf{Ax} - \mathbf{b}\|_2^2, \quad (2.3)$$

em que $\mathbf{x} \in \mathbb{R}^n$ é a variável do problema, $\mathbf{A} \in \mathbb{R}^{m \times n}$ é uma matriz dada e $\mathbf{b} \in \mathbb{R}^m$ é um vetor dado. Este é um problema do tipo QP sem restrições. Ele surge em muitos campos de aplicação e possui diferentes nomes, tais como aproximação por LS ou análise de regressão. Sua função objetivo mede o desajuste entre \mathbf{Ax} e \mathbf{b} , dado pelo vetor residual $\mathbf{r} = \mathbf{Ax} - \mathbf{b}$. Sendo assim, seu objetivo é encontrar o valor de \mathbf{x} que minimiza este desajuste. Considere agora que se deseja encontrar o vetor \mathbf{x} que dê um bom ajuste entre \mathbf{Ax} e \mathbf{b} ($\|\mathbf{r}\|_2^2$ pequena) e que não seja um vetor com grande norma, ou seja, $\|\mathbf{x}\|_2^2$ seja pequena. É possível formular este problema tal como o Problema

(2.2), por

$$\text{minimize } \lambda^T(\|\mathbf{Ax} - \mathbf{b}\|_2^2, \|\mathbf{x}\|_2^2) = \lambda_1\|\mathbf{Ax} - \mathbf{b}\|_2^2 + \lambda_2\|\mathbf{x}\|_2^2, \quad (2.4)$$

em que λ_1 e λ_2 são os pesos dados ao ajuste ($F_1(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|_2^2$) e a norma do vetor ($F_2(\mathbf{x}) = \|\mathbf{x}\|_2^2$), respectivamente. Este problema é conhecido como LS regularizado, pois o objetivo é encontrar o melhor ajuste entre \mathbf{Ax} e \mathbf{b} regularizado pelo menor tamanho possível do vetor \mathbf{x} , a depender de λ_1 e λ_2 . O termo $\|\mathbf{x}\|_2^2$ é conhecido como termo de regularização. Para encontrar os pontos ótimos de Pareto e, conseqüentemente, a curva de compromisso ótima entre os objetivos $F_1(\mathbf{x})$ e $F_2(\mathbf{x})$, resolve-se o Problema (2.4) para diferentes valores do peso relativo $\mu = \lambda_2/\lambda_1$ para $0 < \mu < \infty$. Para cada valor de μ encontra-se um ponto ótimo de Pareto \mathbf{x}^* e o respectivo valor ótimo de Pareto ($F_1(\mathbf{x}^*), F_2(\mathbf{x}^*)$). Os pontos e valores ótimos de Pareto para os casos em que $\mu \rightarrow 0$ e $\mu \rightarrow \infty$ podem ser encontrados resolvendo-se o Problema (2.4) para $\lambda \rightarrow (1, 0)$ e $\lambda = (0, 1)$, respectivamente. O conjunto dos valores ótimos de Pareto constitui a curva de compromisso ótima entre $F_1(\mathbf{x})$ e $F_2(\mathbf{x})$. A Figura 2.1 mostra a curva de compromisso ótima entre os objetivos do Problema (2.4) para $\mathbf{A} \in \mathbb{R}^{100 \times 10}$ onde seus elementos estão compreendidos entre $-0,05$ e $0,05$ e $\mathbf{b} \in \mathbb{R}^{100}$ em que os elementos estão compreendidos entre $-0,5$ e $0,5$. Tanto os elementos da matriz \mathbf{A} quanto os elementos do vetor \mathbf{b} foram gerados através da função do Matlab `randi(a, m, n)`, que gera dados aleatórios uniformemente distribuídos compreendidos entre 0 e a formando uma matriz de m linhas e n colunas. As linhas de código que geram a matriz \mathbf{A} e \mathbf{b} estão apresentadas a seguir.

```
n = 10;
m = 100;
A = randi(100, m, n)/1000 - 0.05;
b = randi(1000, 100, 1)/1000 - 0.5;
```

Vizualizando a curva de compromisso ótima, pode-se obter algumas conclusões sobre as possíveis soluções do Problema (2.4) para diferentes valores de μ :

- O ponto do extremo direito da curva corresponde ao menor valor possível de F_2 sem considerar F_1 . Isto implica em ter-se uma solução cujo o vetor \mathbf{x} possuirá a menor norma euclidiana possível sem qualquer preocupação com o ajuste entre \mathbf{Ax} e \mathbf{b} . Esta solução ocorre quando $\mu \rightarrow \infty$ e pode ser obtida com $\lambda = (0, 1)$.
- O ponto do extremo esquerdo da curva corresponde ao menor valor possível de F_1 sem considerar F_2 . Isto implica em ter-se uma solução cujo o ajuste entre \mathbf{Ax}

e \mathbf{b} será o melhor possível, sem qualquer preocupação com a norma do vetor \mathbf{x} . Esta solução ocorre quando $\mu \rightarrow 0$ e pode ser obtida com $\lambda \rightarrow (1, 0)$.

- Ao encontrar a intersecção da curva com uma linha vertical em $F_1 = v$, é possível ver o quão grande F_2 deve ser para alcançar $F_1 \leq v$. Em outras palavras, verifica-se o quão pior será o ajuste entre \mathbf{Ax} e \mathbf{b} para se ter a norma de \mathbf{x} menor ou igual a v .
- Ao encontrar a intersecção da curva com uma linha horizontal em $F_2 = h$ é possível ver o quão grande F_1 deve ser para alcançar $F_2 \leq h$. Em outras palavras, verifica-se o quão maior será a norma de \mathbf{x} para se ter um ajuste entre \mathbf{Ax} e \mathbf{b} menor ou igual a h .
- A derivada da curva em um determinado ponto sobre ela mostra o compromisso ótimo local entre os dois objetivos. Onde a derivada é íngreme, pequenas mudanças em F_1 são acompanhadas por grandes mudanças em F_2 , ou seja, uma pequena melhora ou piora no ajuste entre \mathbf{Ax} e \mathbf{b} proporciona um grande aumento ou uma grande diminuição da norma de \mathbf{x} respectivamente.
- Um ponto de grande curvatura é aquele em que, a partir dele, para que se tenha um determinado decréscimo em um objetivo deve-se ter um acréscimo de maior proporção no outro objetivo e vice e versa. Este ponto é conhecido como joelho da curva de compromisso ótima e em muitas aplicações representa uma boa solução de compromisso.

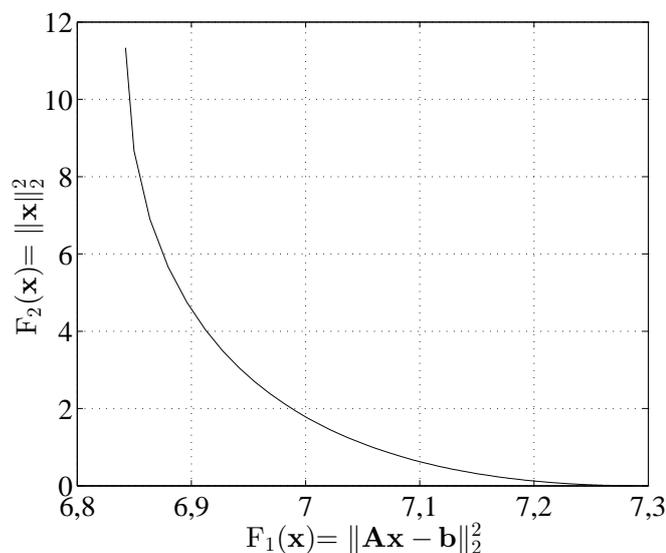


Figura 2.1: Curva de compromisso ótima entre os objetivos $\|\mathbf{Ax} - \mathbf{b}\|_2^2$ e $\|\mathbf{x}\|_2^2$.

2.2 O CVX e o conjunto de regras DCP

O CVX [19] é um sistema de modelagem em software, desenvolvido na plataforma Matlab, utilizado para a construção e solução de problemas de otimização baseadas em um conjunto de regras da programação convexa disciplinada (DCP, *disciplined convex optimization*) [9], ou conjunto de regras DCP. Ele é um software livre para uso em ambientes acadêmicos e comerciais. As informações completas sobre as plataformas suportadas, licenciamento e instalação podem ser obtidas diretamente em [19]. Por ser implementado no Matlab e utilizar grande parte das funções e comandos comuns do mesmo, este sistema de modelagem facilita a manipulação dos resultados obtidos na solução de um determinado problema modelado e a integração deste problema a um projeto maior, em que o problema de otimização convexa é apenas um de seus componentes.

Para utilizar o CVX é preciso seguir o conjunto de regras DCP. Estas regras baseiam-se nos princípios básicos de análise convexa e são de fácil entendimento. O CVX foi construído de maneira que o usuário seja obrigado a seguir o conjunto de regras DCP. Não segui-las implica na ocorrência de um erro. Este conjunto de regras é um conjunto de condições suficientes, mas não necessárias para estabelecer a convexidade de uma função. Sendo assim, existem funções que são convexas e que não são interpretadas como convexas pelo CVX. A descrição detalhada do conjunto de regras DCP pode ser encontrada em [19].

O CVX suporta a programação linear (LP), a programação quadrática (QP), a programação semidefinida (SDP, *semidefinite programming*), entre outras. Também é capaz de resolver problemas complexos de otimização nos quais a modelagem utiliza funções não diferenciáveis.

É importante ressaltar que a utilização eficaz do CVX requer do usuário o conhecimento prévio dos conceitos básicos sobre otimização convexa [8], sobre álgebra linear [20] e sobre o conjunto de regras DCP. É importante também, antes de inserir um problema no CVX, saber se o mesmo é um problema de otimização convexa, uma vez que o CVX não deve ser utilizado para verificação de convexidade. Há problemas de otimização convexa que não podem ser resolvidos pelo CVX, a menos que sejam traduzidos em uma forma equivalente, compatível com as regras DCP. Por outro lado, se o CVX declara um problema como resolvido, tem-se a garantia de que se trata de um problema de otimização convexa.

A estrutura genérica de um bloco de código CVX utilizado para modelar um pro-

blema de otimização no Matlab é:

```
cvx_begin %(comando de inicialização do bloco)
    % declaração das variáveis;
    % função objetivo;
    % restrições;
cvx_end %(comando de finalização do bloco)
```

Todo bloco de código CVX deve ser inicializado com o comando `cvx_begin` e terminado com o comando `cvx_end`. Todas as declarações de variáveis, a função objetivo e as restrições que modelam o problema de otimização devem estar entre estes dois comandos. Se o problema modelado for resolvido com sucesso, seu valor ótimo e seu ponto ótimo estarão disponíveis respectivamente na variável `cvx_optval` e nas variáveis do problema declaradas pelo usuário. Caso ocorra algum erro, tal como problema não factível, ilimitado ou falha na solução, mensagens de erros correspondentes aparecerão na janela de comandos do Matlab. A lista a seguir apresenta a sintaxe dos principais comandos que devem ou podem ser utilizados dentro de um bloco de código CVX:

- **Variáveis:** Todas as variáveis do problema de otimização devem ser declaradas antes de serem utilizadas nas funções. As variáveis podem ser escalares, vetorias, matricias ou arranjos multidimensionais, sendo reais ou complexas. A declaração de uma variável é feita através do comando `variable`, seguido pela designação da variável e, opcionalmente, por informações correspondentes à estrutura e às restrições da variável, conforme mostram os exemplos a seguir:

```
variable a;
variable v(6) nonnegative;
variable Z(2,2) semidefinite;
variable Q(3,3) complex semidefinite;
```

Nestes exemplos são declaradas respectivamente uma variável escalar a , um vetor v de 6 elementos não negativos, uma matriz Z de ordem 2×2 semidefinida positiva e uma matriz Q hermitiana complexa semidefinida positiva de ordem 3×3 . As diferentes informações correspondentes às possíveis estruturações de uma matriz são encontradas em [19]. É possível declarar mais de uma variável de uma só vez utilizando o comando `variables`, tal como:

```
variables a b c x(2) y(2,2) z(3,3,3);
```

Este recurso é disponível somente para as declarações de variáveis sem informações adicionais de estruturas ou restrições. As declarações de variáveis com informações adicionais devem ser feitas individualmente utilizando o comando `variable`.

O CVX permite ainda a declaração de um vetor de expressões através do comando `expression`. Como exemplo, considere o seguinte trecho de código pertencente a um bloco de código CVX:

```
expression x(M);
for i = 1:M
    x(i) = p(i)*square_pos(norm(S(i,:)'-d));
end
```

Primeiramente é criado o vetor de expressões `x` com `M` posições. Depois, dentro da estrutura de repetição, entre os comandos `for` e `end`, a `i`-ésima posição de `x` recebe a expressão convexa `p(i)*square_pos(norm(S(i,:)'-d))`, onde `p` é um vetor constante dado, `S` é uma matriz constante dada e `d` é uma variável. Por fim, tem-se o vetor de expressões `x` com todas suas posições ocupadas por expressões. Este vetor pode ser a entrada de uma função do CVX, como por exemplo:

```
%sum(f) = f(1)+f(2)+...+f(M) =
%= p(1)*square_pos(norm(S(1,:)'-x))+
%+ p(2)*square_pos(norm(S(2,:)'-x))+
%+...+p(M)*square_pos(norm(S(M,:)'-x)).
```

A função `sum(f)` pode ser utilizada, por exemplo, como a função objetivo de um determinado problema de otimização convexa no CVX.

Diferentemente de uma variável, um vetor de expressões pode ser escrito e sobrescrito em qualquer parte do bloco de código.

- **Função objetivo:** A declaração da função objetivo de um problema de otimização deve ser sempre precedida dos comandos `minimize` (`minimise`) ou `maximize` (`maximise`), a depender da curvatura da função em questão. Se a função objetivo é convexa, trata-se de um problema de minimização e, portanto, deve-se utilizar o comando `minimize`. Se a função objetivo é côncava,

trata-se de um problema de maximização e, portanto, deve-se utilizar o comando `maximize`. Se a função objetivo é afim (convexa e côncava), o problema de otimização pode ser tanto um problema de maximização quanto um problema de minimização, o que implica no uso dos comandos `minimize` ou `maximize` respectivamente. A função objetivo deve ser um escalar e não se pode declarar mais do que uma função objetivo por bloco de código CVX. Se nenhuma função objetivo for declarada, o problema passa a ser interpretado como um problema de factibilidade. Neste caso, `cvx_optval` será 0 caso o problema seja factível ou `+Inf` caso o problema seja infactível. Seguem dois exemplos de declarações de funções objetivos:

```
maximize sqrt(Ax + b);  
minimize abs(a'y);
```

A função objetivo do primeiro exemplo é côncava, pois trata-se da composição de uma função côncava com uma função afim e, portanto, constitui um problema de maximização. A função objetivo do segundo exemplo é convexa, pois trata-se da composição de uma função convexa com uma função afim e, portanto, constitui um problema de minimização. As regras de composição fazem parte do conjunto de regras da DCP.

- Restrições: O CVX aceita as seguintes restrições:
 - restrições de igualdade, `==`, onde ambos os lados são funções afins;
 - restrições de desigualdade menor igual, `<=`, onde o lado esquerdo é uma função convexa e o lado direito é uma função côncava;
 - restrições de desigualdade maior igual, `>=`, onde o lado esquerdo é uma função côncava e o lado direito é uma função convexa.

Note que o sinal `=` é diferente de `==`. O primeiro significa uma atribuição, enquanto o segundo corresponde a uma restrição de igualdade. As desigualdades estritas dadas por `<` ou `>` são aceitas pelo CVX, porém são tratadas da mesma maneira que as desigualdades não estritas `<=` ou `>=` respectivamente.

O CVX possui uma biblioteca de funções, chamada de biblioteca átomo, que inclui uma variedade de funções convexas, côncavas e afins adequadas à sintaxe do CVX. Estas funções aceitam como argumentos variáveis ou expressões do bloco de código CVX. Muitas funções dessa biblioteca são comuns no Matlab e foram reimplementadas para serem suportadas pelo CVX, tais como `sum`, `trace`, `diag`, `sqrt`,

\max , e \min . Outras funções são novas e não são encontradas no Matlab. Uma lista completa de funções da biblioteca átomo pode ser encontrada em [19]. Um exemplo de função desta biblioteca é a função `quad_form(x, P)`. Esta função, não encontrada no Matlab, representa a função $\mathbf{x}^T \mathbf{P} \mathbf{x}$ para $\mathbf{x} \in \mathbb{R}^n$ e $\mathbf{P} \in \mathbb{S}^n$ ou $\mathbf{x}^* \mathbf{P} \mathbf{x}$ para $\mathbf{x} \in \mathbb{C}^n$ e $\mathbf{P} \in \mathbb{H}^n$. Ela é convexa em \mathbf{x} para \mathbf{P} constante e semidefinida positiva, e côncava em \mathbf{x} para \mathbf{P} constante e semidefinida negativa.

Como já mencionado, para modelar um determinado problema de otimização convexa no CVX, é preciso reescrever este problema obedecendo a sintaxe de programação do CVX e o conjunto de regras DCP. Seguem exemplos de modelagens no CVX referentes aos Problemas (2.3) e (2.4) respectivamente:

```
%Modelagem do Problema LS

cvx_begin quiet
% O comando quiet enibe o aparecimento de
% informações referentes ao processo de solução do
% problema na janela de comando do Matlab.
    variable x(n);
    minimize square_pos(norm(A*x - b))
cvx_end

% x é o vetor que proporciona o melhor ajuste entre
% A*x e b.
```

```
%Modelagem do Problema LS Regularizado

% lambda = (lambda1, lambda2) > 0, em que:
% lambda1 é o peso de f(1) = square_pos(norm(A*x-b));
% lambda2 é o peso de f(2) = square_pos(norm(x)).

cvx_begin quiet
    variable x(n);
    expression f(2);
    f(1) = square_pos(norm(A*x-b));
    f(2) = square_pos(norm(x));
    minimize lambda'*f
cvx_end

% Dado lambda, x é o vetor que proporciona o melhor
% ajuste entre A*x e b possuindo a menor norma
% euclidiana possível.
```

2.3 Técnicas de reconstrução de sinais via otimização convexa

Para entender um problema de reconstrução de sinais, considere primeiramente que um sinal seja representado por suas amostras em um vetor $\mathbf{s} \in \mathbb{R}^n$, em que s_i para $i = 1, \dots, n$ é a i -ésima amostra do sinal. Em termos gerais considera-se que este sinal é suave. O sinal \mathbf{s} é corrompido pelo ruído aditivo \mathbf{r} , formando-se o sinal corrompido:

$$\mathbf{y} = \mathbf{s} + \mathbf{r}.$$

O ruído pode ser modelado de várias formas diferentes, mas neste caso admite-se apenas que ele é aleatório, pequeno em amplitude ($\mathbf{r} < \mathbf{s}$) e, diferentemente do sinal, varia rapidamente. O objetivo de um problema de reconstrução de sinais é formar uma estimativa $\hat{\mathbf{s}}$ do sinal original \mathbf{s} a partir do sinal corrompido \mathbf{y} . Este processo é chamado de reconstrução de sinal (uma vez que tenta-se reconstruir o sinal original a partir de sua versão corrompida) ou redução de ruídos (uma vez que tenta-se remover o ruído do sinal corrompido). A maioria dos métodos de reconstrução acabam realizando algum tipo de operação de suavização sobre \mathbf{y} para produzir $\hat{\mathbf{s}}$.

Uma formulação simples de um problema de reconstrução baseada no Problema bicritério escalarizado (2.2) pode ser dada por:

$$\text{minimize } \sum_{i=1}^n \phi_{12}(\hat{s}_i - y_i) + \delta \phi_{\text{suav}}(\hat{\mathbf{s}}), \quad (2.5)$$

em que

$$\phi_{12}(r) = r^2, \quad (2.6)$$

$\hat{\mathbf{s}} = [\hat{s}_1, \hat{s}_2, \dots, \hat{s}_n]$ é a variável do problema, $\mathbf{y} = [y_1, y_2, \dots, y_n]$ é o parâmetro de entrada e $\delta > 0$ (λ_2) é o parâmetro de escalarização referente ao segundo objetivo do problema, que no caso, é o próprio peso relativo pois $\lambda_1 = 1$. A função ϕ_{12} , conhecida como função de penalização quadrática, penaliza o desajuste entre o i -ésimo elemento do sinal estimado (\hat{s}_i) e o i -ésimo elemento do sinal corrompido (y_i), ou o i -ésimo resíduo $r_i = \hat{s}_i - y_i$. De acordo com a função quadrática mostrada na Figura 2.2, quanto maior for o módulo do resíduo, maior será sua penalização e menor a chance de valores correspondentes deste resíduo se manifestarem após a solução do problema. A soma das penalizações dadas a cada r_i constitui o primeiro objetivo do problema e é chamada de penalização total, que no caso penaliza o desajuste total entre o vetor $\hat{\mathbf{s}}$ e o vetor \mathbf{y} , tal como o objetivo do problema de aproximação por LS dado em (2.3). A função $\phi_{\text{suav}} : \mathbb{R}^n \rightarrow \mathbb{R}$ é convexa e é chamada de função de regularização ou objetivo de suavização. Ela controla a rugosidade ou a falta de suavidade da estimativa $\hat{\mathbf{s}}$. Desta forma, o objetivo do Problema (2.5) é encontrar o sinal que mais se aproxime do sinal corrompido com a maior suavidade possível, a depender do valor de δ . A estimativa $\hat{\mathbf{s}}$ será mais suave e menos parecida com \mathbf{y} quanto maior for o valor de δ ou será menos suave e mais parecida com \mathbf{y} quanto menor for o valor de δ .

A curva de compromisso ótima entre o ajuste e a suavização pode ser obtida resolvendo-se o Problema (2.5) para diferentes valores de $\delta > 0$.

Dois exemplos de problemas de reconstrução baseados no Problema (2.5) são o problema de reconstrução por suavização quadrática e o problema de reconstrução por variação total (*total variation*) [8], cujos os respectivos objetivos de suavização são:

$$\phi_{\text{quad}}(\hat{\mathbf{s}}) = \sum_{i=1}^{n-1} (\hat{s}_{i+1} - \hat{s}_i)^2, \quad (2.7)$$

$$\phi_{\text{tv}}(\hat{\mathbf{s}}) = \sum_{i=1}^{n-1} |\hat{s}_{i+1} - \hat{s}_i|. \quad (2.8)$$

A determinação de qual dos objetivos de suavização utilizar e, conseqüentemente, do método de reconstrução está relacionada com a suavidade do sinal original. A reconstrução por suavização quadrática é eficaz quando o sinal original é suave e o ruído varia rapidamente. Qualquer tipo de variação rápida no sinal original será atenuada ou removida pela suavização quadrática. Para sinais originais suaves com eventuais variações rápidas, recomenda-se o uso da reconstrução por variação total. Este método irá remover boa parte do ruído presente no sinal corrompido preservando as ocasionais variações rápidas do sinal original.

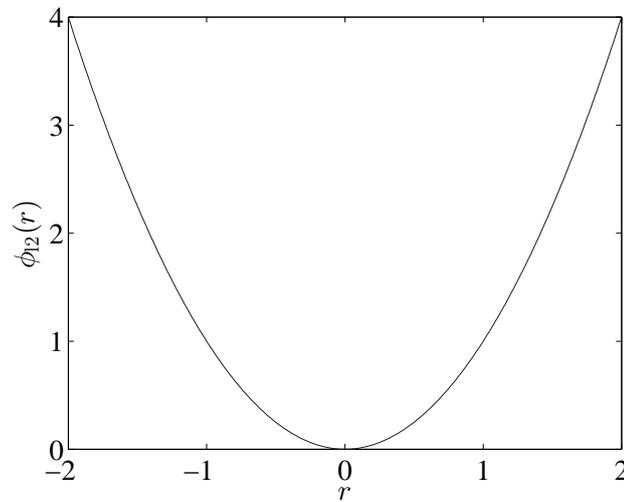


Figura 2.2: Função de penalização quadrática para $|r| \leq 2$.

A técnica de redução de ruído proposta para combater o ruído composto, designada por técnica COD, consiste na formulação de um problema de reconstrução de sinais parecido com o problema de reconstrução por suavização quadrática, dado por:

$$\text{minimize } \sum_{i=1}^n \phi_{\text{hub}}(\hat{s}_i - y_i) + \delta\phi_{\text{quad}}(\hat{\mathbf{s}}), \quad (2.9)$$

em que

$$\phi_{\text{hub}}(r) = \begin{cases} r^2 & \text{se } |r| \leq M, \\ M(2|r| - M) & \text{se } |r| > M. \end{cases} \quad (2.10)$$

A diferença deste problema para o problema de reconstrução por suavização quadrática está na construção do objetivo de aproximação: agora, a função de penalização escolhida para penalizar o desajuste individual entre os elementos dos vetores $\hat{\mathbf{s}}$ e \mathbf{y} não é mais a função quadrática e sim a função de Huber, ϕ_{hub} , que também é uma função convexa [8]. O parâmetro $M > 0$ é chamado de fator de forma da função de Huber.

A diferença da função de penalização quadrática para a função de penalização de Huber está na maneira com que a função de Huber penaliza os resíduos com valores em módulo maiores do que M . Isto pode ser observado na Figura 2.3, onde estão apresentadas a função quadrática e a função de Huber com $M = 0,5$ para valores de r compreendidos entre $-2 \leq r \leq 2$. Nota-se na figura que, assim como a função quadrática, a função de Huber penaliza da mesma forma os resíduos com valores em módulo menores e iguais a M , enquanto que para valores de resíduos maiores do que M a penalização é menor. Os resíduos maiores são ocasionados pelas amostras de ruído com maiores amplitudes, que no caso do ruído composto, correspondem as amostras impulsivas do ruído. Quanto menor o peso dado aos resíduos referentes às amostras impulsivas do ruído, menor será a influência destas amostras na aproximação. Desta forma, a aproximação realizada pela função de Huber irá ter menor influência das amostras impulsivas do ruído composto do que a aproximação realizada pela função quadrática, resultando em um sinal estimado mais aproximado do sinal original.

Em um problema de regressão linear, por exemplo, a aproximação feita pela função de penalização de Huber comparada com a aproximação feita pela função de penalização quadrática (aproximação LS) resulta em uma reta cuja a inclinação é menos influenciada pelos dados discrepantes, conhecidos como *outliers*, recebendo assim o nome de aproximação LS robusta. A Figura 2.4 apresenta soluções de um problema de regressão linear implementado com a função de penalização quadrática e com a função penalização de Huber. Na figura encontram-se 42 pontos circulares que podem ser bem aproximados por uma função linear, exceto por dois pontos *outliers*, localizados no canto superior esquerdo e no canto inferior direito. A linha sólida é o resultado de uma aproximação LS de uma reta $f(t) = at + b$ aos pontos. Pode-se observar que esta linha está rotacionada para fora do local principal de pontos, em direção aos *outliers*. Já a linha tracejada, referente ao resultado de uma aproximação LS robusta, apresenta melhor ajuste da reta aos dados não *outliers*.

Da perspectiva do ruído composto presente em sistemas UWAC, as amostras de ruído impulsivo podem ser vistas como *outliers* dentre as amostras do sinal recebido e a solução do problema LS robusto dentro do problema de regressão linear pode ser vista como uma aproximação linear de um segmento deste sinal.

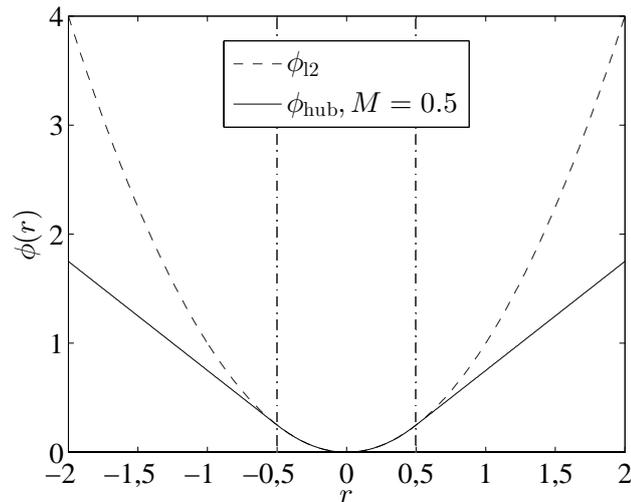


Figura 2.3: Funções de penalização quadrática e de Huber com $M = 0.5$.

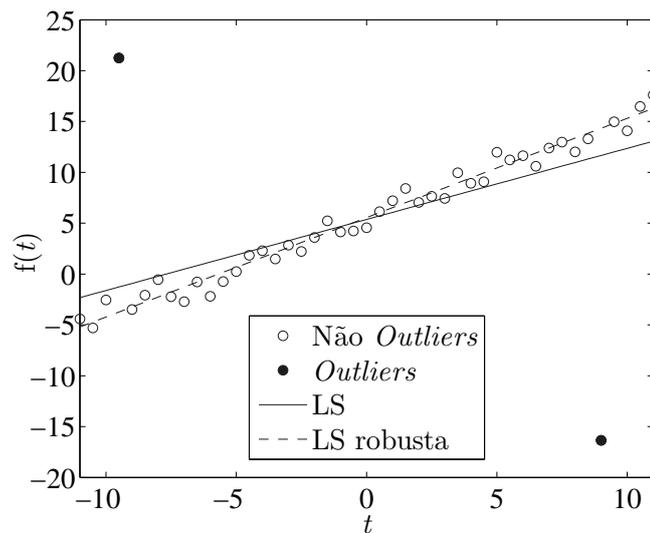


Figura 2.4: Soluções resultantes de problemas de regressão linear.

As escolhas dos objetivos de suavização e de aproximação do Problema (2.9) estão relacionadas com o conhecimento *a priori* de características do sinal original a ser estimado e do ruído. A escolha do objetivo de suavização quadrática é justificada pelo conhecimento *a priori* de que o sinal original é suave. Já a escolha do objetivo de aproximação LS robusta é justificada por saber que o ruído é impulsivo, com o objetivo de minimizar a influência das amostras impulsivas do ruído na aproximação. A robustez do objetivo de aproximação está relacionada com o fator de forma M . Combinando-se a aproximação LS robusta com o processo de suavização quadrática, tem-se um problema de reconstrução de sinais cujo o objetivo é reconstruir o sinal mais suave possível que melhor se ajuste ao sinal recebido, sem muita influência do ruído impul-

sivo. A relação de compromisso entre os objetivos de aproximação e de suavização é definida pelo peso relativo δ da seguinte forma: um menor valor de δ atua em favor de um melhor ajuste entre o sinal reconstruído e o sinal corrompido, no sentido da medida de Huber; um maior valor de δ atua em favor de um sinal reconstruído mais suave.

Existem diferentes maneiras de se modelar o Problema (2.9) no CVX. Um fator que facilita sua modelagem é que o CVX possui a função de Huber em sua biblioteca átomo. A Figura 2.5 apresenta o fluxograma correspondente a solução do Problema (2.9) por meio do CVX e na sequência está uma opção de modelagem deste problema no Matlab:

```
%Modelagem do problema de reconstrução de sinais da
%técnica COD

lambda = [1; delta]; % delta é o peso relativo.

cvx_begin quiet
    variable s_est(n);
    expression f(2);
    f(1) = sum(huber((s_est-y), M));
    f(2) = sum_square(s_est(2:(n))-s_est(1:(n-1)));
    minimize ((lambda')*f)
cvx_end

% s_est é o vetor de sinal reconstruído.
```

Para fins ilustrativos, a Figura 2.6 mostra 2000 amostras envolvidas em um processo de reconstrução de sinais utilizando a técnica proposta, cujo o fator de forma e o peso relativo escolhidos foram $M = 0,01$ e $\delta = 2$, respectivamente. Note que o vetor de sinal recebido y (b) está sob condição de ruído intenso, com forte impulsividade. Mesmo assim, o processo de reconstrução consegue reduzir drasticamente este ruído, como pode ser observado no sinal reconstruído \hat{s} (c).

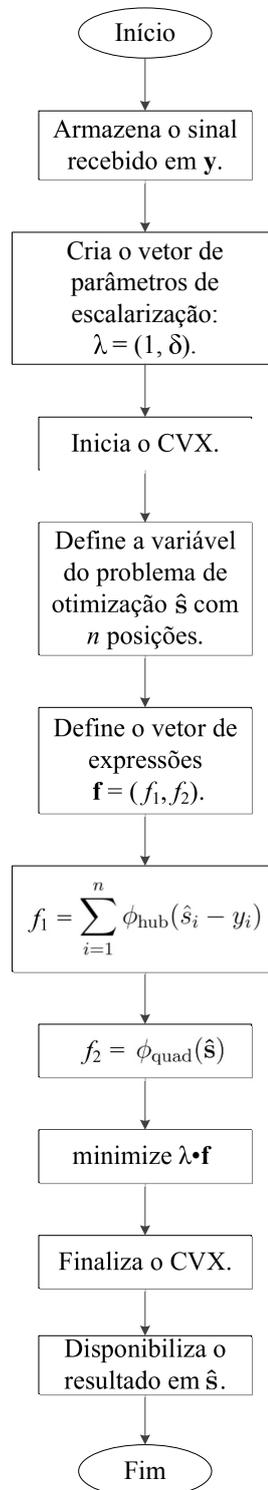


Figura 2.5: Fluxograma da solução do problema de otimização da técnica COD

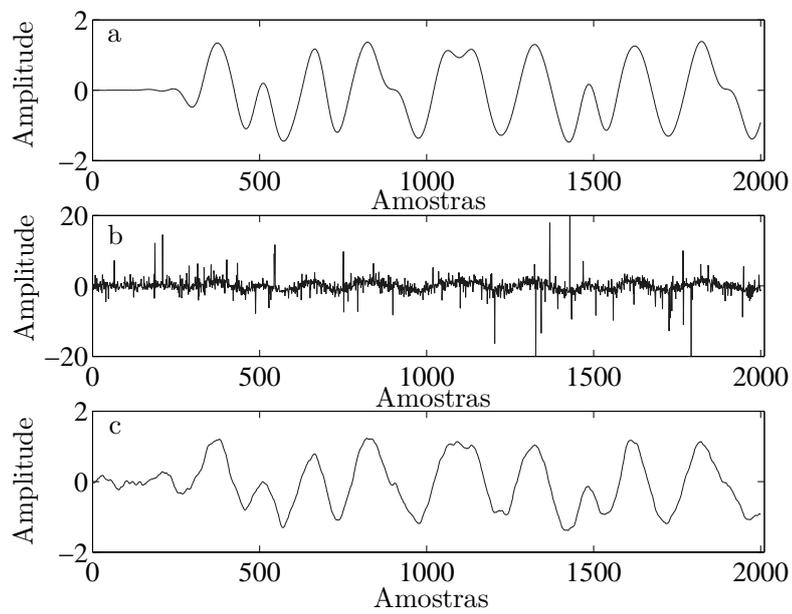


Figura 2.6: Exemplos de: a) vetor s , b) vetor y e c) vetor \hat{s} .

Capítulo 3

Otimização convexa aplicada à redução de ruído impulsivo em sistemas de comunicação sob a água

Este capítulo inicia-se com uma abordagem geral sobre sistemas UWAC. Na sequência são citados alguns sistemas UWAC comerciais, informando suas principais características, tais como, modulação, taxa de bits e alcance máximo. Em seguida apresenta-se a densidade espectral de potência (PSD, *power spectral density*) construída a partir de amostras reais do ruído composto e a função densidade de probabilidade (PDF, *probability density function*) utilizada na modelagem estatística do ruído. Posteriormente é mostrado o sistema UWAC adotado para a aplicação da técnica COD. Depois realiza-se a validação do modelo adotado, através da comparação da curva de desempenho obtida do sistema adotado com a respectiva curva teórica. Também é mostrado um diagrama de olho, com intuito de avaliar o projeto dos filtros de transmissão e de recepção. Para concluir o capítulo são apresentadas algumas formas de onda ilustrando a atuação da técnica COD no sistema adotado e são calculados os MSEs entre o sinal original e os outros sinais envolvidos no processo, com objetivo de quantificar o ganho de MSE proporcionado pela redução de ruído.

3.1 Sistemas de comunicação sob a água

Os sistemas UWAC podem ser classificados de acordo com o seu alcance como muito longo, longo, médio, curto ou muito curto [21], como mostra a Tabela 3.1. A

largura de banda disponível para um enlace UWAC depende de seu alcance. Os enlaces acústicos podem também ser classificados como vertical e horizontal, de acordo com a direção da radiação do som com relação ao fundo do oceano. A literatura tipicamente atribui o nome de águas rasas (*shallow water*) para águas com profundidades inferiores a 100 m, e águas profundas (*deep water*) para águas com maiores profundidades [18]. As comunicações acústicas subaquáticas são severamente afetadas pelas características do canal, tais como, elevada perda do percurso de propagação, ruídos, múltiplos percursos, atraso de propagação e espalhamento *Doppler* elevados e variáveis. Os múltiplos percursos afetam menos os enlaces verticais do que os enlaces horizontais e, de forma geral, as características dos canais em águas profundas impõem menores dificuldades para a comunicação acústica do que as características dos canais em águas rasas [21]. Desta forma, do ponto de vista do projeto de um sistema UWAC, menores serão os desafios encontrados para os enlaces verticais em águas profundas e maiores serão os desafios encontrados para os enlaces horizontais em águas rasas.

Para superar as dificuldades apresentadas pelo canal, os projetos dos sistemas UWAC disponíveis comercialmente baseavam-se principalmente no uso de técnicas de modulação com detecção não coerente, com baixa vazão de dados, na ordem de b/s [21]. Com o intuito de tornar mais eficiente o uso da largura de banda do canal acústico sub-aquático, técnicas de modulação com detecção coerente junto com processamento de sinais para exploração da diversidade espacial dos múltiplos percursos foram utilizadas. Esta nova abordagem resultou em um aumento de no mínimo uma ordem de magnitude na vazão dos dados dos sistemas UWAC, para a casa dos kb/s. Os sistemas UWAC disponíveis atualmente suportam principalmente aplicações ponto a ponto, com taxas de dados na ordem de kb/s e tolerantes a atrasos.

Tabela 3.1: Largura de banda disponível para diferentes alcances em canais UWAC.

Classificação	Alcance (km)	BW (kHz)
Muito longo	1000	< 1
Longo	10 – 100	2 – 5
Médio	1 – 10	≈ 10
Curto	0,1 – 1	20 – 50
Muito curto	< 0,1	> 100

As abordagens dos projetos de sistemas UWAC variam de acordo com a técnica utilizada para superar os efeitos da ISI e as variações de fase do sinal. Estas técnicas podem ser classificadas de acordo com o projeto do sinal (que diz respeito à escolha do método de modulação e detecção) e à estrutura do transmissor e do receptor (relacionada com a escolha do método de processamento e do método de equalização, se

houver). De acordo com o projeto de sinal, é possível encontrar sistemas UWAC disponíveis comercialmente ou experimentais [22] que utilizam modulações de uma única portadora, como a modulação não coerente MFSK (*M-ary frequency-shift keying*), as modulações coerentes MPSK (*M-ary phase-shift keying*) e MQAM (*M-ary quadrature amplitude modulation*), e a modulação diferencialmente coerente DPSK (*differential phase-shift keying*). Há ainda sistemas que utilizam a técnica DSSS (*direct-sequence spread-spectrum*), encontrada com ou sem coerência de fase e que permite o múltiplo acesso por divisão de código com sequência direta (DS-CDMA, *direct-sequence code-division multiple-access*). Também encontram-se sistemas UWAC implementados com multiplexação em múltiplas portadoras OFDM (*orthogonal frequency-division multiplexing*). Em relação à estrutura do transmissor e do receptor, existem sistemas UWAC que utilizam múltiplos transmissores e múltiplos receptores, técnica conhecida como MIMO (*multiple-input multiple-output*), com o objetivo de explorar a diversidade espacial ocasionada pelos múltiplos percursos e melhorar o desempenho do sistema. A equalização de canal é utilizada juntamente com a modulação coerente afim de garantir o sucesso na detecção. Um possível equalizador encontrado em sistemas UWAC é o equalizador adaptativo com realimentação de decisão (DFE, *decision feedback equalizer*).

Existem diversos modems acústicos disponíveis no mercado, produzidos por vários fabricantes tais como Aquatec [23], Develogic [24] e Teledyne Benthos [25].

O AQUAmodem 1000, da empresa Aquatec, é utilizado para telemetria e é configurável pela fábrica de acordo com a necessidade do cliente, devido às diferentes características de propagação acústica impostas pelos diferentes canais sub-aquáticos. Dependendo dos requisitos, este modem pode operar até a máxima profundidade dos oceanos (em torno de 11 km). Dependendo do meio, é informado que pode-se atingir um alcance de até 20 km. Uma recente aplicação em águas rasas atingiu um alcance de 5 km. A banda de frequência utilizada abrange as frequências de 7,5 a 12 kHz. As modulações podem ser MFSK ou DPSK com espalhamento espectral. A taxa de bits alcançada varia de 300 b/s a 2 kb/s.

Os sistemas HAM.NODE, desenvolvidos pela empresa Develogic, são modems acústicos para telemetria com configurações disponíveis para aplicações em águas rasas e águas profundas. As modulações utilizadas são MDPSK (combinada com OFDM) ou n-MFSK. A modulação MDPSK é utilizada com ou sem DFE para realizar comunicações com maiores taxas em canais verticais ou horizontais com múltiplos percursos limitados. A modulação n-MFSK, apesar de proporcionar menores taxas do que a modulação MDPSK com OFDM, é utilizada para comunicações de longo

alcance em águas rasas ou como alternativa dentro dos canais acústicos em geral, por ser uma modulação mais robusta. As bandas de frequências disponíveis são 2,5 a 4,5 kHz, 8 a 13 kHz, 11 a 20 kHz ou 17 a 29 kHz. Sistemas reais implementados por usuários em comunicações verticais alcançaram taxas maiores que 7 kb/s para 1,95 km e aproximadamente 3,4 kb/s para 6 km, ambos com a modulação MDPSK e OFDM. Em comunicação horizontal, um sistema real alcançou uma taxa de aproximadamente 145 b/s em uma distância de 30 km utilizando a modulação n-MFSK.

Os modems acústicos para telemetria da série ATM-900, produzidos pela empresa Teledyne Benthos, possuem configurações que trabalham com uma profundidade de até 6 km. Os alcances variam de 2 a 6 km, com possibilidades de alcances maiores dependendo das características do canal. As bandas de frequência podem abranger as frequências de 9 a 14 kHz, 16 a 21 kHz ou 22 a 27 kHz. As modulações podem ser MFSK e MPSK. As taxas de comunicação variam de 140 a 2400 b/s para MFSK e de 2560 a 15360 b/s para a modulação MPSK.

Devido à limitação espectral do canal marítimo, às limitações de energia e às exigências de BER e de taxa de bits por parte das aplicações [21], melhorar o desempenho ($BER \times E_b/N_0$) de um sistema UWAC torna possível a realização de aplicações com taxa de erro de bits cada vez menores e com maiores taxas de bits no ambiente subaquático, como por exemplo, telecontroles de maiores confiabilidades ($BER < 10^{-5}$ e taxa $\cong 1$ kb/s) e transmissões de vídeo com melhores qualidades (taxas > 100 kb/s e $BER \cong 10^{-4}$).

3.2 A técnica COD aplicada a um sistema UWAC

3.2.1 O ruído composto

Uma das principais diferenças entre o canal UWAC em águas quentes e rasas e outros canais está na estrutura do ruído ambiente. A estrutura do ruído ambiente determina a estrutura do ruído composto, dado pela soma do ruído térmico do receptor com o ruído ambiente. Na prática a potência do ruído térmico é muito menor quando comparada a potência do ruído ambiente. Desta forma, o ruído composto pode ser definido de forma aproximada pelo ruído ambiente.

PSD do ruído composto

A Figura 3.1 apresenta a PSD do ruído composto obtida a partir de dados reais coletados em águas de Singapura. O eixo vertical da figura é marcado como *uncalibrated* devido às exigências de confidencialidade impostas pela agência financiadora do projeto de coleta dos dados. A PSD correspondente às frequências menores que 1 kHz refere-se de forma dominante às embarcações, a PSD correspondente às frequências maiores que 2 kHz refere-se de forma dominante aos camarões de estalo e a PSD correspondente às frequências entre 300 Hz e 3 kHz refere-se à reverberação sonora.

A Figura 3.1 também compara a PSD obtida com curvas de PSD arquivadas de outras localidades, para baixas e altas frequências. Para as frequências referente às embarcações, foi necessário uma elevação da curva de PSD *heavy shipping curve* para concordar com a PSD do ruído ambiente de Singapura. Isto pode ser explicado pelo fato de Singapura ser um porto movimentado, gerando mais ruído do que a curva de referência correspondente. No caso das frequências referente aos estalos dos camarões, é possível notar uma boa concordância entre a PSD e a curva de referência *snapping shrimp curve*. Como a atenuação do som é um processo dependente da frequência, o oceano atua como um filtro passa baixas para o ruído ambiente. Desta forma, a PSD do ruído ambiente é aproximadamente rosa, o que implica que o ruído tem mais potência nas menores frequências e menos potência nas maiores frequências, como pode ser observado na Figura 3.1. O ruído impulsivo produzido pelos camarões de estalo dominam a PSD correspondente às altas frequências do ruído ambiente presente em águas marítimas quentes e rasas, exceto em lugares com ventos fortes ou chuvosos [14].

PDF do ruído composto

Embora cada camarão produza um impulso acústico distinto, o ruído ambiente observado é constituído de um grande número de impulsos sobrepostos vindos de muitas colônias de camarões e de reflexões no fundo e na superfície do mar. A partir de dados reais de valores de pressão do ruído ambiente, obteve-se a PDF empírica correspondente [14], constatando a característica não Gaussiana do ruído. Várias PDFs conhecidas foram testadas em [14] com o objetivo de se aproximar aos dados medidos, tais como a mistura de Gaussianas, a Gaussiana generalizada e a exponencial dupla Gaussiana. Entretanto, nenhuma destas PDFs apresentou precisão na aproximação com a PDF empírica.

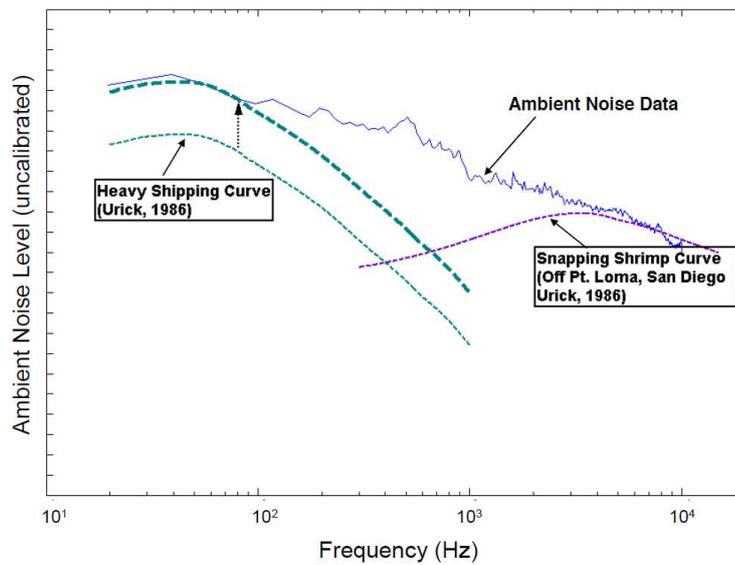


Figura 3.1: PSD do ruído ambiente (de [14], reproduzida com permissão).

Como o ruído ambiente impulsivo tende a produzir excursões de grandes amplitudes a partir da média nula mais frequentemente do que os sinais Gaussianos, vislumbrou-se a possibilidade de se utilizar a distribuição $S\alpha S$ (*symmetric α -stable*) na modelagem do ruído ambiente do oceano [14]. Esta distribuição é um caso particular da família de distribuições estáveis (*stable*), que são ferramentas teóricas úteis para a modelagem de ruídos com características impulsivas.

A distribuição estável é uma generalização direta da distribuição Gaussiana, incluindo-a como um caso limite. Um parâmetro da distribuição estável, conhecido como expoente característico α , controla a taxa de decaimento da cauda da distribuição. Quanto menor o valor de α , para $0 < \alpha \leq 2$, menor a taxa de decaimento da cauda da distribuição correspondente e, quanto maior o valor de α , maior a taxa. Um pequeno valor positivo de α representa um ruído altamente impulsivo, enquanto que um valor de α próximo de 2 indica um ruído com comportamento parecido com o Gaussiano. Quando $\alpha = 2$, a distribuição reduz-se para a distribuição Gaussiana. As distribuições estáveis são assim definidas por apresentarem a propriedade de estabilidade, que diz que a soma de duas variáveis aleatórias independentes definidas por distribuições estáveis com um determinado valor de α resulta em uma variável aleatória definida por uma distribuição estável com o mesmo valor de α . Por essa propriedade, as distribuições estáveis são também conhecidas como α estáveis (*α -stables*).

Uma distribuição estável pode ser escrita através de sua função característica, dada

por [14]:

$$\varphi(t) = \exp[jat - \gamma|t|^\alpha \{1 + j\beta \operatorname{sgn}(t)\omega(t, \alpha)\}], \quad (3.1)$$

em que

$$j = \sqrt{-1}; \operatorname{sgn}(t) = 1, t \geq 0 \text{ e } \operatorname{sgn}(t) = -1, t < 0;$$

$$\omega(t, \alpha) = \begin{cases} \tan \frac{\alpha\pi}{2} & \text{se } \alpha \neq 1, \\ \frac{2}{\pi} \log |t| & \text{se } \alpha = 1. \end{cases}; \quad (3.2)$$

α é o expoente característico, β é o parâmetro de simetria, γ é o parâmetro de dispersão e a é o parâmetro de localização. O parâmetro de dispersão determina a amplitude da distribuição, tal como a variância em uma distribuição Gaussiana. Quando $\alpha = 2$, γ é igual à metade da variância da distribuição Gaussiana resultante. Para todos os outros valores de α , a variância da distribuição é infinita. Um parâmetro frequentemente utilizado com as distribuições estáveis é o parâmetro de escala $c = \gamma^{\frac{1}{\alpha}}$, equivalente ao desvio padrão nas variáveis aleatórias Gaussianas. O parâmetro de localização corresponde à média da distribuição quando $\alpha > 1$ e à mediana da distribuição para os outros casos.

A distribuição $S\alpha S$ tem os parâmetro de simetria β e de localização a iguais a 0. A expressão fechada de uma PDF estável pode ser encontrada através da transformada de Fourier inversa da função característica (3.1). Infelizmente, tal como o caso da PDF $S\alpha S$, não existe expressões fechadas para as PDFs estáveis de um modo geral, exceto para a Gaussiana ($\alpha = 2$), Cauchy ($\alpha = 1, \beta = 0$) e Pearson ($\alpha = 0,5, \beta = -1$). Porém, existem métodos numéricos eficientes para calcular a PDF de uma distribuição $S\alpha S$.

Para certificar que as distribuições $S\alpha S$ são de fato boas aproximações do ruído ambiente real, várias amostras de ruído ambiente coletadas em diferentes horários e em diferentes partes das águas de Singapura foram testadas [14]. Todas produziram PDFs que puderam ser bem aproximadas pela PDF $S\alpha S$ com parâmetros apropriados. Os valores de α estimados estiveram dentro da faixa de 1,5 a 1,9. A Figura 3.2 apresenta a PDF empírica de dados reais do ruído ambiente comparada à PDF $S\alpha S$ aproximada, em que os parâmetros estimados foram $\alpha = 1,563$ e $c = 0,1178$ Pa. É possível visualizar uma boa concordância entre as duas PDFs. Testes de rejeição foram feitos em [14] para avaliar a aproximação realizada pela distribuição $S\alpha S$. Os resultados comprovaram a eficácia da aproximação.

A partir de um pacote de funções $S\alpha S$ para o Matlab disponível em [26], foi gerada

a PDF S α S para valores de $c = 0,1178$ e $\alpha = 1,563$, correspondentes aos valores estimados citados anteriormente. Para comparar, gerou-se também a PDF S α S para valores de $c = 0,1178$ e $\alpha = 2$, equivalente a uma distribuição Gaussiana de média 0 e $\sigma = c\sqrt{2} \approx 0,167$. Estas PDFs são mostradas na Figura 3.3 (a), em que as PDFs S α S com valores de $\alpha = 2$ e $\alpha = 1,563$ estão representadas pela linha seccionada e linha contínua, respectivamente. Nota-se nesta figura que a PDF S α S para $\alpha = 1,563$ apresenta maiores probabilidades de ocorrência de valores dos módulos das amplitudes maiores que 2σ do que a PDF S α S para $\alpha = 2$, resultado da ocorrência dos ruídos impulsivos. A Figura 3.3 (b) apresenta a forma de onda de um ruído composto gerada a partir de uma distribuição S α S com $c = 0,1178$ e $\alpha = 1,563$.

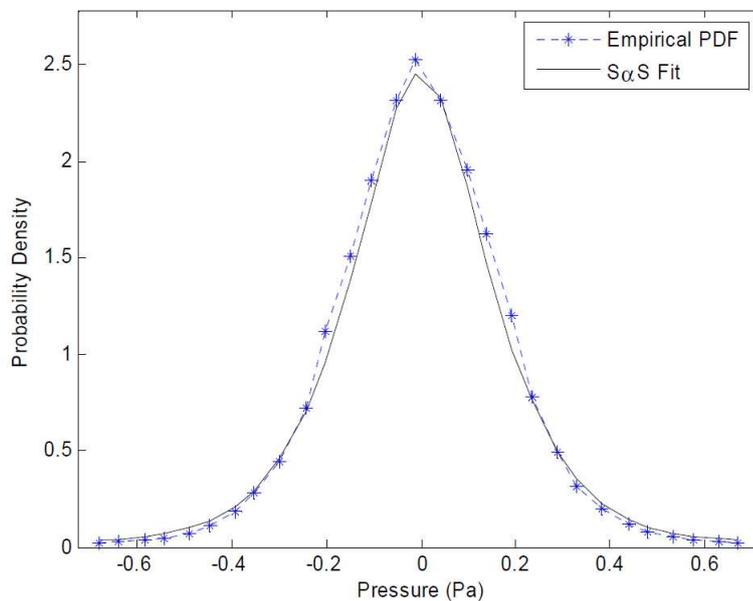


Figura 3.2: Melhor ajuste S α S (de [14], reproduzida com permissão).

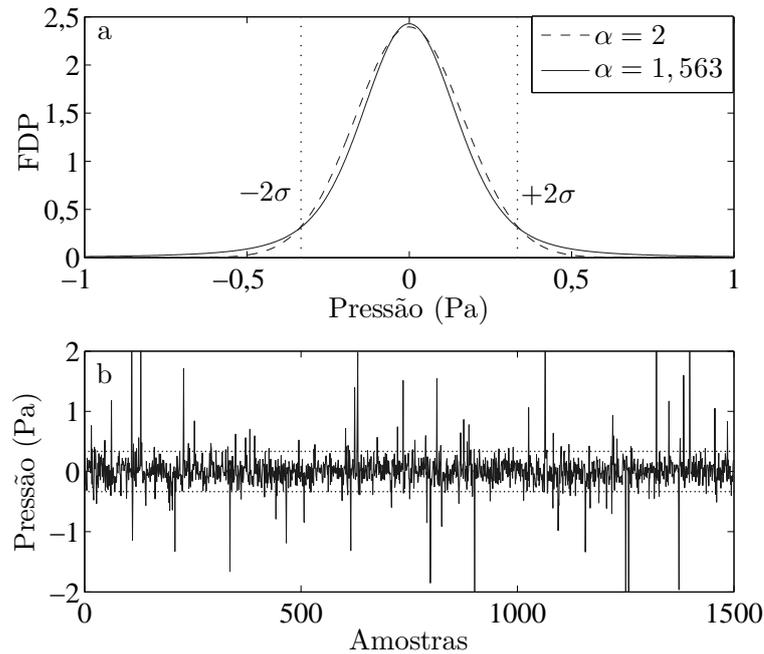


Figura 3.3: (a) PDFs SaS com $c = 0, 1178$. (b) forma de onda do ruído composto.

3.2.2 Modelo de Sistema UWAC

Seja o vetor s a representação de um sinal transmitido com taxa de variação de suas amplitudes muito mais lenta do que a taxa de variação das amplitudes do vetor de ruído composto r , de tal forma que a técnica COD trabalhe como desejado. Esta condição pode ser satisfeita pelos sinais UWAC modulados e filtrados, pois além da suavidade produzida pela filtragem, a portadora nestes sistemas é uma onda acústica, tipicamente com frequência na ordem de poucas dezenas de quilohertz [12]. Além disso, o conteúdo de frequência do ruído impulsivo gerado pelos camarões de estalo atinge facilmente 250 kHz [4], resultando em uma taxa de variação muito maior do que a taxa de variação do sinal transmitido modulado e filtrado de um sistema UWAC. Além do ruído composto, os sinais UWAC são também sujeitos a desvanecimento, mesmo em comunicações de curto alcance, onde o desvanecimento é similar, mas ligeiramente menos severo do que o previsto pela distribuição de Rayleigh [14]. Desta forma é necessário considerar o efeito do desvanecimento no desempenho de um sistema UWAC.

Com o objetivo de aderir aos requisitos citados no parágrafo anterior, adotou-se o modelo de sistema apresentado na Figura 3.4, equivalente ao sistema de comunicação digital em banda passante com modulação BPSK coerente mostrado na Figura 1.1, mas

com os modelos da propagação e do ruído do canal UWAC em águas marítimas quentes e rasas [14]. O funcionamento do sistema da Figura 3.4 partindo do mapeamento do vetor de bits transmitidos \mathbf{b} até a obtenção do vetor de bits estimados $\hat{\mathbf{b}}$ é descrito a seguir:

- (1) O vetor \mathbf{b} carrega bits de informação aleatórios equiprováveis que são mapeados em símbolos BPSK, em que os bits 0 e 1 são mapeados para os símbolos -1 e 1 respectivamente, formando o vetor de símbolos \mathbf{d} . As Figuras 3.5 e 3.6 exemplificam formas de ondas correspondentes aos vetores \mathbf{b} e \mathbf{d} respectivamente.
- (2) Embora os canais UWAC de águas marítimas quentes e rasas apresentarem tipicamente o desvanecimento seletivo e rápido [11], adotou-se o melhor caso possível que é o desvanecimento plano e lento [11] com o objetivo de avaliar isoladamente o efeito do ruído composto na atuação das técnicas de redução de ruído. O efeito do desvanecimento plano e lento é modelado pela multiplicação dos símbolos do vetor \mathbf{d} por uma variável aleatória Rayleigh com o segundo momento unitário, constante durante a duração do símbolo e independente de símbolo para símbolo. Nota-se que esta é uma maneira simples de garantir que, dentro deste modelo de canal, a componente do desvanecimento que afeta a variável de decisão seja distribuída de acordo com uma distribuição Rayleigh, permitindo a comparação do desempenho $\text{BER} \times E_b/N_0$ em relação aos resultados teóricos e diminuindo o tempo de simulação do sistema. Se um canal com desvanecimento fosse adotado ao invés desta abordagem, as variações da envoltória do desvanecimento deveriam ser muito lentas para concordar com a suposição de desvanecimento constante durante os intervalos de símbolos. Isto levaria a um tempo final de simulação maior para garantir que o efeito médio do desvanecimento se manifestasse. As variações de fase do canal não foram consideradas pois se está admitindo detecção coerente. A Figura 3.7 corresponde a forma de onda de um vetor de símbolos desvanecidos.
- (3) O vetor \mathbf{d} desvanecido é então filtrado por um filtro RRC (*root-raised-cosine*) equipado com uma equalização IS (*inverse sinc*), uma vez que os pulsos em sua entrada são retangulares. O filtro possui um fator de *roll-off* igual a 1 e $100k$ taps, em que 100 é o número de amostras por intervalo de símbolos e k é o número de intervalos de símbolos dentro da duração da resposta ao impulso do filtro. Foi adotado um $k = 10$ de tal forma que a maior parte da cauda da resposta ao impulso fosse levada em conta na síntese dessa resposta ao impulso. A Figura 3.8 é um exemplo da forma de onda do sinal amostrado e filtrado, em que o

atraso ocasionado na entrada e na saída do processo de filtragem corresponde a 5 símbolos ou 500 amostras.

- (4) O sinal filtrado modula uma portadora acústica, correspondente a uma onda cossenoidal com frequência dentro da faixa acústica (tipicamente de 1 kHz a 100kHz). Este processo converte as componentes de frequência do sinal para a banda passante (UC, *up converter*), formando o vetor de sinal transmitido s . A Figura 3.9 apresenta um exemplo de forma de onda correspondente a s .

Para transmitir o sinal no canal sub-aquático é necessário converter o sinal elétrico em ondas sonoras. Esta conversão pode ser realizada por um dispositivo conhecido como transdutor. Considera-se no sistema que o transdutor possui sua função de transferência linear.

- (5) O sinal acústico transmitido é corrompido pelo ruído ambiente acústico durante a sua propagação no canal sub-aquático. A recepção do sinal deve ser realizada por um dispositivo que converte as ondas sonoras em sinal elétrico. Esta função pode ser realizada por um dispositivo conhecido como hidrofone. Considera-se no sistema que o hidrofone possui sua função de transferência linear e sua resposta de frequência plana dentro da banda do ruído ambiente, garantido assim que as características estatísticas e de frequência do sinal transmitido e do ruído ambiente sejam mantidas no sinal elétrico resultante após a conversão. O sinal elétrico resultante é corrompido pelo ruído térmico.

Para simular a soma do ruído ambiente e do ruído térmico com o sinal transmitido, o vetor s é corrompido pelo ruído composto r gerado pelo processo $S\alpha S$ que, como já informado, modela adequadamente o ruído ambiente de canais UWAC em águas marítimas quentes e rasas juntamente com o ruído térmico. O sinal corrompido pelo ruído composto é representado pelo vetor y . As Figuras 3.10 e 3.11 representam as formas de onda do ruído r sem impulsividade ($\alpha = 2$) e do sinal y respectivamente. Na próxima seção serão apresentadas formas de ondas de sinais envolvidos em processos de redução de ruído impulsivo pela técnica COD, em que o ruído composto considerado é impulsivo, gerado por um processo $S\alpha S$ com $\alpha = 2$.

- (6) O vetor y é então submetido a um processo de redução de ruído, gerando o sinal estimado \hat{s} . Para fins de análises e comparações de desempenhos, o processo de redução de ruído foi realizado neste trabalho separadamente por três técnicas: pela técnica COD e por duas técnicas de redução de ruído via dispositivo não linear [27] [14], as técnicas END e END modificada, a serem abordadas

no próximo capítulo. Para a validação do modelo adotado, assunto da próxima subseção, não é realizado o processo de redução de ruídos.

- (7) O sinal \hat{s} (ou y para validação do modelo) é multiplicado novamente por uma portadora acústica, igual e sincronizada em fase com a portadora do processo de UC, mas agora com o objetivo de converter as componentes de frequência do sinal para a banda base (DC, *down converter*). A Figura 3.12 apresenta a forma de onda de um sinal convertido para a banda base.
- (8) O sinal em banda base é então filtrado por um filtro RRC, sem equalização IS, com o fator de *roll-off* e números de taps iguais aos do filtro de transmissão. A Figura 3.13 é um exemplo de forma de onda do sinal recebido filtrado. Tal como a etapa de filtragem da transmissão, esta etapa também ocasiona um atraso que corresponde a 5 símbolos ou 500 amostras na entrada e na saída do processo de filtragem. Desta forma, o atraso total no sinal transmitido ocasionado pelas duas etapas de filtragem corresponde aos 10 símbolos ou as 100 amostras iniciais e finais do sinal recebido. Estas amostras de atraso são eliminadas e apenas a informação restante referente ao sinal transmitido é utilizada.
- (9) A saída do filtro de recepção é amostrada e retida e os resultados são comparados ao valor 0 para se estimarem os símbolos transmitidos, sendo que um valor maior que 0 é decidido como o símbolo 1 e um valor menor que 0 é decidido como o símbolo -1. Os resultados de todas as decisões formam o vetor de símbolos estimados \hat{d} . A Figura 3.14 é um exemplo de forma de onda correspondente ao vetor \hat{d} .
- (10) Por fim os símbolos do vetor \hat{d} são demapeados em bits pelo demapeador BPSK, em que os símbolos -1 e 1 são demapeados para os bits 0 e 1, respectivamente, formando o vetor de bits estimados \hat{b} . Os bits estimados do vetor \hat{b} são então comparados aos bits transmitidos do vetor b para o cálculo da BER. A Figura 3.15 é um exemplo da forma de onda correspondente ao vetor \hat{b} .

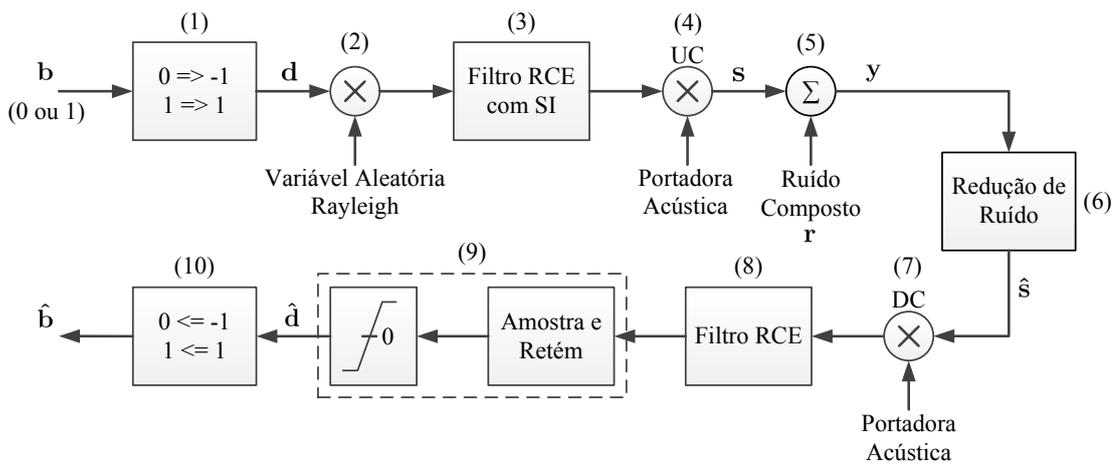


Figura 3.4: Modelo de sistema adotado para a análise dos desempenhos.

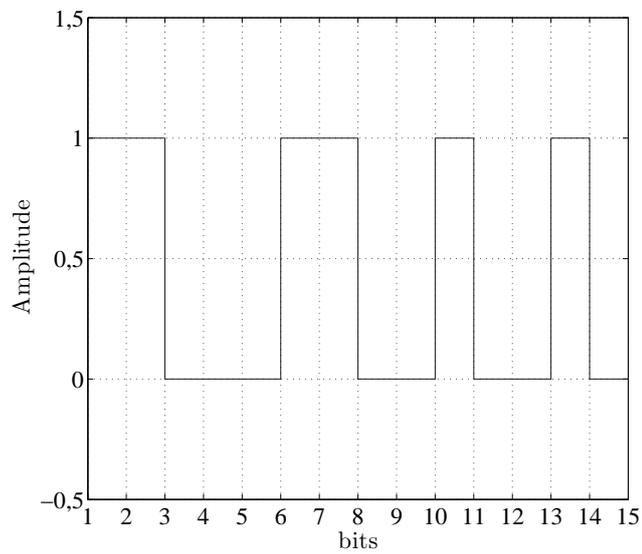


Figura 3.5: Exemplo de forma de onda do vetor de bits \mathbf{b} .

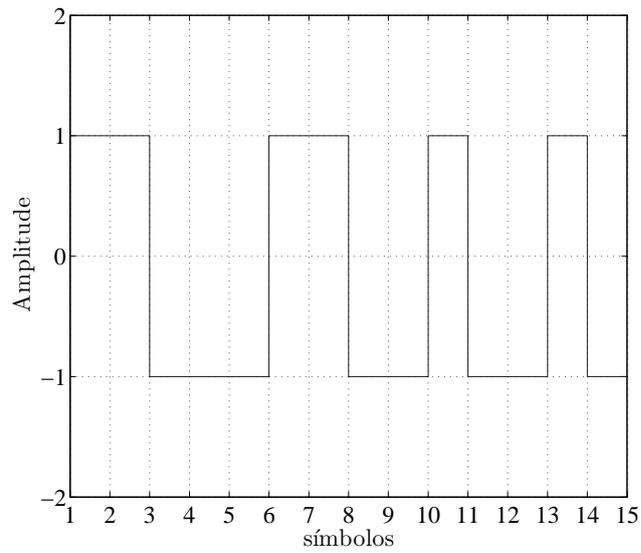


Figura 3.6: Exemplo de forma de onda do vetor de símbolos *d*.

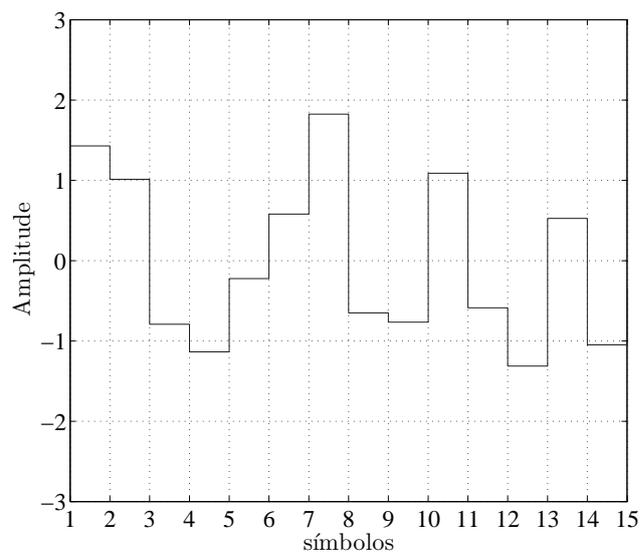


Figura 3.7: Exemplo de forma de onda do vetor de símbolos desvanecidos.

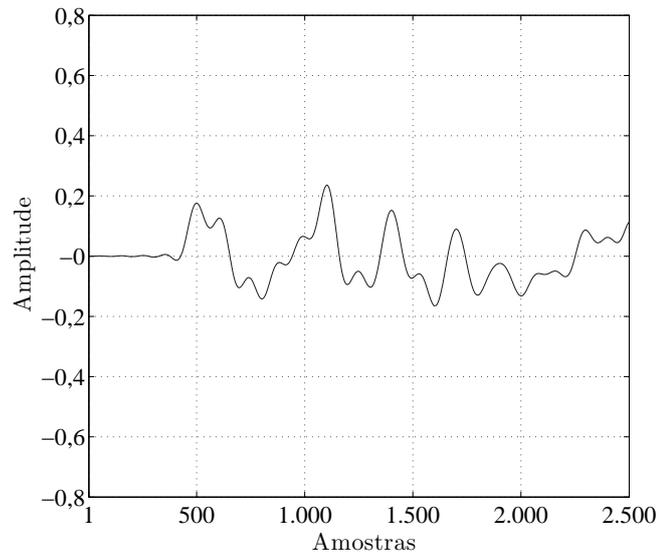


Figura 3.8: *Exemplo de forma de onda do sinal amostrado e filtrado.*

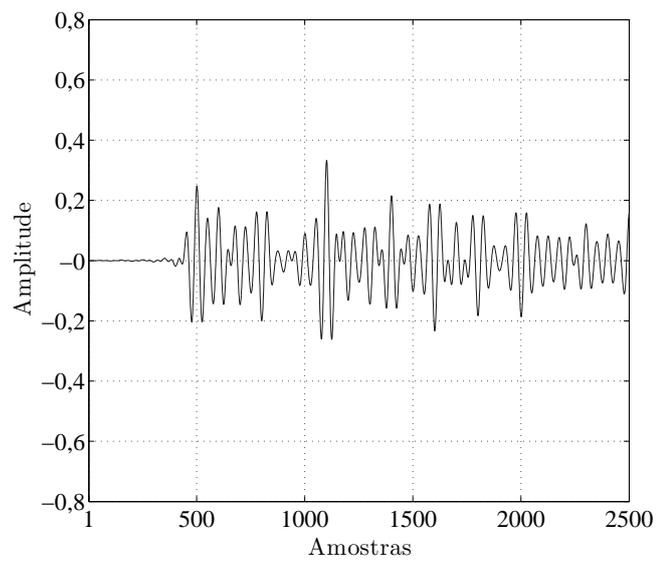


Figura 3.9: *Exemplo de forma de onda do sinal transmitido s.*

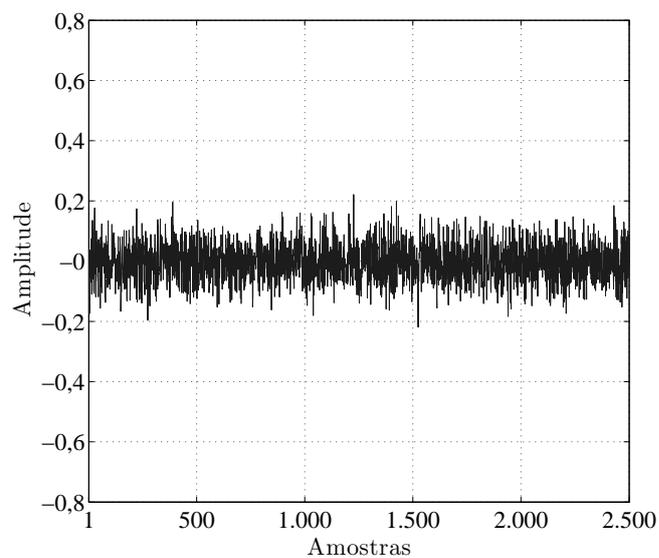


Figura 3.10: *Exemplo de forma de onda do ruído r sem impulsividade.*

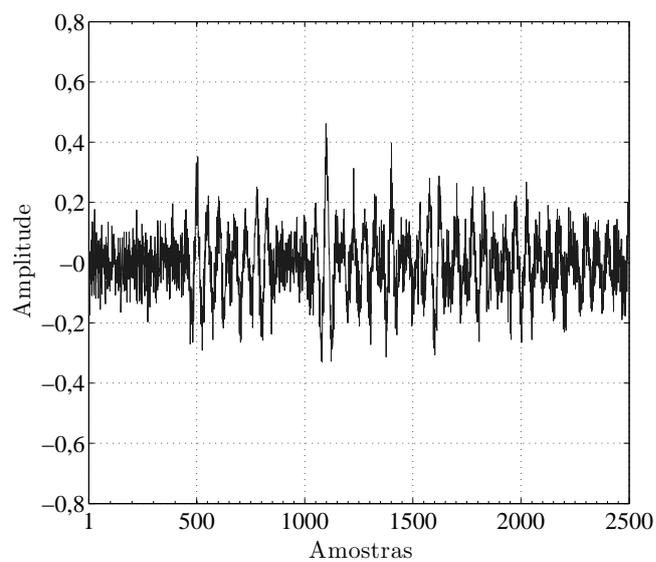


Figura 3.11: *Exemplo de forma de onda do sinal y .*

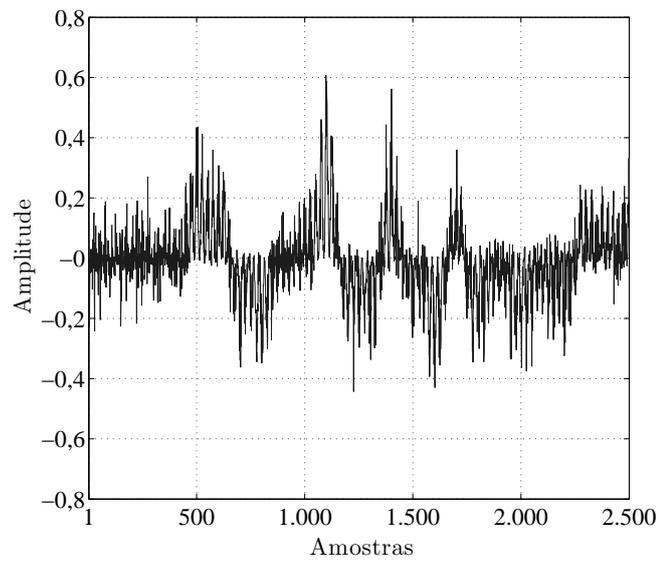


Figura 3.12: Exemplo de forma de onda do sinal convertido para a banda base.

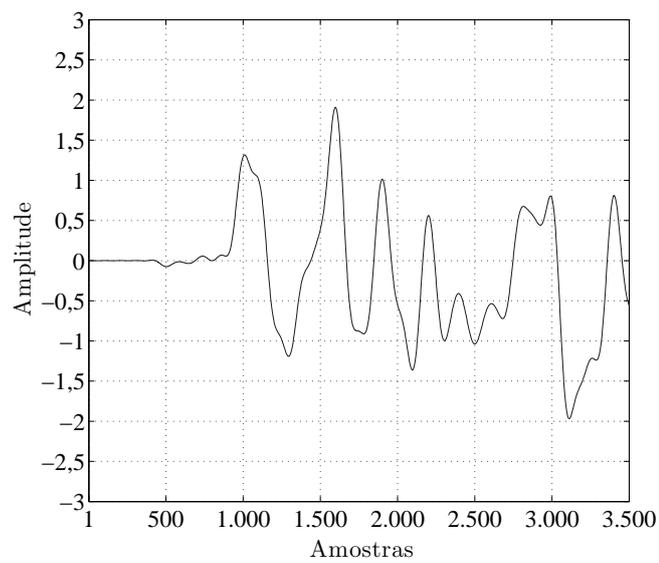


Figura 3.13: Exemplo de forma de onda do sinal recebido filtrado.

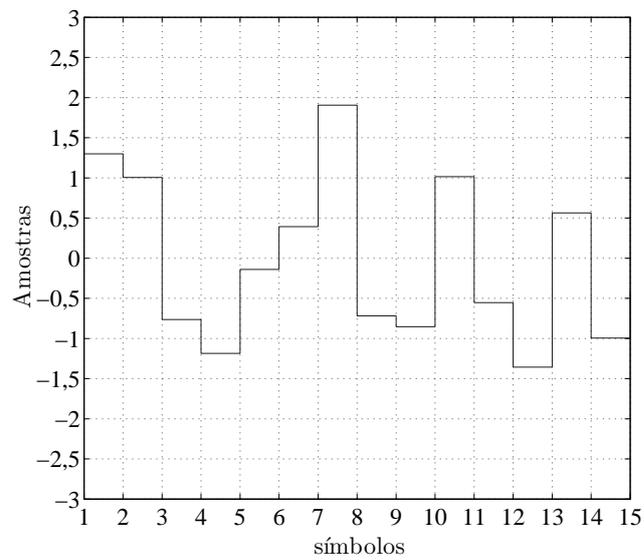


Figura 3.14: Exemplo de forma de onda do vetor de símbolos estimados $\hat{\mathbf{d}}$.

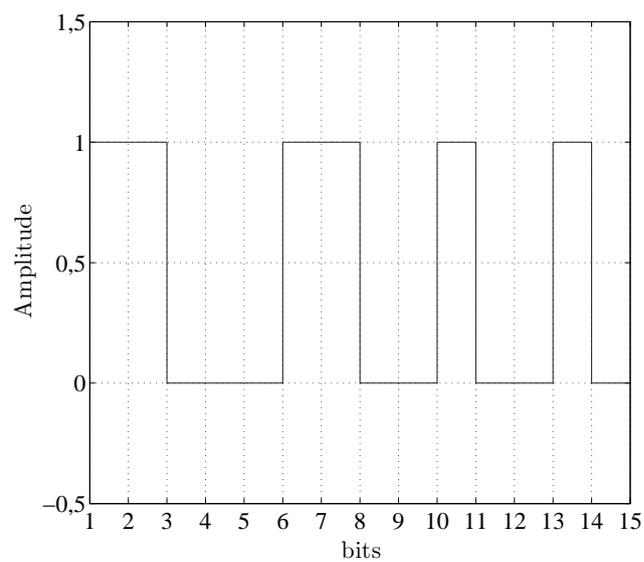


Figura 3.15: Exemplo de forma de onda do vetor de bits estimados $\hat{\mathbf{b}}$.

3.2.3 Validação de modelo e formas de onda dos sinais envolvidos na técnica COD

O modelo do sistema adotado da Figura 3.4 foi implementado no Matlab e posteriormente validado. O software utilizado para a validação do modelo encontra-se no Apêndice A. A validação foi realizada obtendo-se a curva de desempenho ($\text{BER} \times E_b/N_0$) do sistema sem a redução de ruído para vários valores de E_b/N_0 e comparando-a com a curva de desempenho teórico de um modulador BPSK coerente sobre o canal Rayleigh-AWGN. Para isto modificou-se o processo $S\alpha S$ para um processo gaussiano fazendo $\alpha = 2$ e ajustou-se o parâmetro de escala c em função do E_b/N_0 desejado, a partir da equação [5]:

$$c = \frac{\sigma}{\sqrt{2}} = \frac{\sqrt{N_0}}{2} = \sqrt{\frac{E_b}{4 \times 10^{\frac{E_b/N_0}{10}}}}, \quad (3.3)$$

em que σ é o desvio padrão da componente de ruído Gaussiano dentro do processo $S\alpha S$, $E_b = P/R_b$ é uma função da potência média do sinal recebido P e da taxa de bits R_b , e E_b/N_0 está em dB. É importante ressaltar que este cálculo de c é exato somente quando $\alpha = 2$, o único caso em que a variância do processo $S\alpha S$ é finita. Desta forma, deve-se considerar que o valor de E_b/N_0 é aproximadamente igual ao valor desejado quando $\alpha \neq 2$. A curva de desempenho do sistema e a curva teórica são mostradas na Figura 3.16. Cada ponto da curva de desempenho do sistema foi obtido através de eventos de Monte Carlo, onde o critério de parada foi 100 erros. Observa-se na figura uma clara concordância entre as curvas de desempenho, validando o sistema proposto.

Para avaliar o dimensionamento dos filtros de transmissão e recepção, utilizou-se a ferramenta gráfica conhecida como diagrama de olho [11], que permite avaliar a influência do ruído e da ISI no sistema de comunicação. O casamento entre os filtros é avaliado pela ISI, sendo que uma ISI nula indica um perfeito casamento. No diagrama de olho, o fechamento vertical do olho pode ser ocasionado por ruído, ISI ou ambos. O efeito de *jitter* no instante de amostragem não foi considerado na simulação do sistema em questão, de tal forma que o instante de amostragem ocorre sempre no instante de maior abertura vertical do diagrama de olho. Para permitir a análise de uma possível contribuição individual ocorrida pela ISI no fechamento vertical do olho, o diagrama foi obtido na saída do filtro de recepção do sistema sem considerar o efeito do desvanecimento e do ruído composto. O resultado está mostrado na Figura 3.17. Como pode ser visto na figura, a seta indica uma abertura vertical máxima do olho

no instante ótimo de amostragem, o que implica em uma ISI ≈ 0 . Do ponto de vista prático, isto reflete um casamento quase perfeito entre os filtros. A perfeição não é alcançada devido ao truncamento das respostas ao impulso de ambos os filtros.

Para ilustrar a atuação da técnica COD no sistema da Figura 3.4, foi realizada a redução de ruído do sistema sob condição de desvanecimento e ruído composto com $\alpha = 1,5$ em relações $E_b/N_0 = 7$ dB, 21 dB e 35 dB, cujas formas de ondas envolvidas estão apresentadas respectivamente nas Figuras 3.18, 3.19 e 3.20. Os parâmetros da técnica de redução de ruído foram escolhidos como $M = 0,01$ e $\delta = 1$. No próximo capítulo será apresentado o método utilizado para a escolha destes parâmetros. As formas de ondas correspondentes às letras (a), (b) e (c) das figuras se referem aos sinais sem ruído composto s , corrompido y e estimado \hat{s} , respectivamente. Os resultados mostram que foram realizadas boas estimações nos três casos, em que os MSEs calculados entre os sinais \hat{s} e s pela Equação (1.2) para $E_b/N_0 = 7$ dB, 21 dB e 35 dB foram $e_{\text{MSE}-7} = 9,1 \times 10^{-3}$, $e_{\text{MSE}-21} = 6,94 \times 10^{-4}$ e $e_{\text{MSE}-35} = 5,44 \times 10^{-5}$ respectivamente.

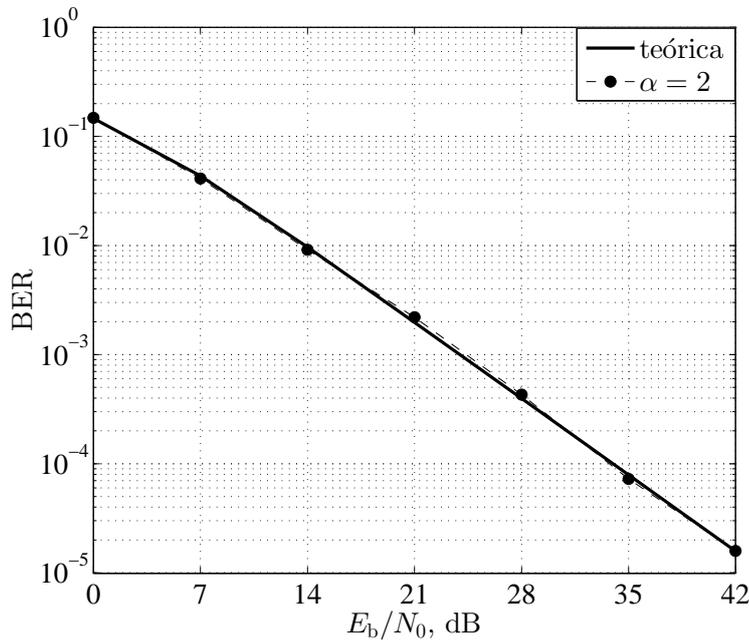


Figura 3.16: Validação do modelo adotado.

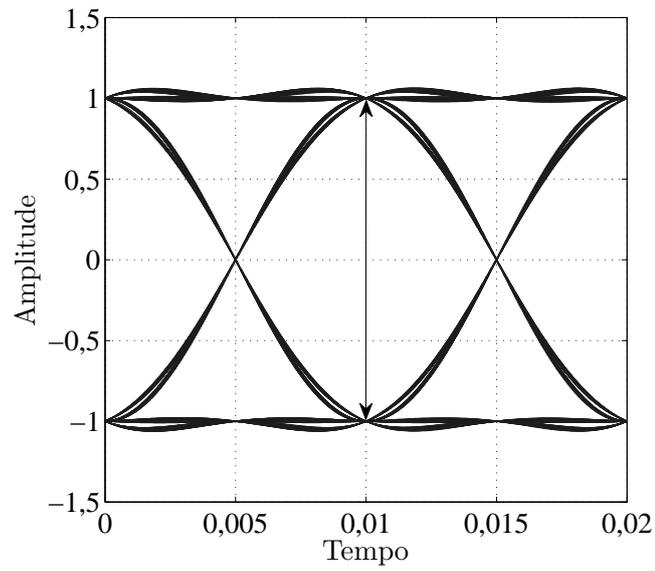


Figura 3.17: Diagrama de olho obtido na saída do filtro de recepção.

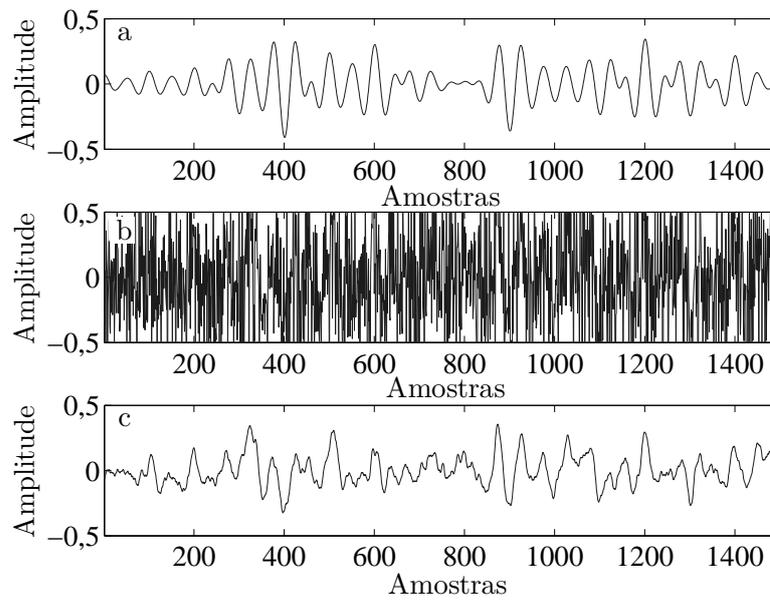


Figura 3.18: Formas de onda para $E_b/N_0 = 7$ dB: (a) s , (b) y e (c) \hat{s}

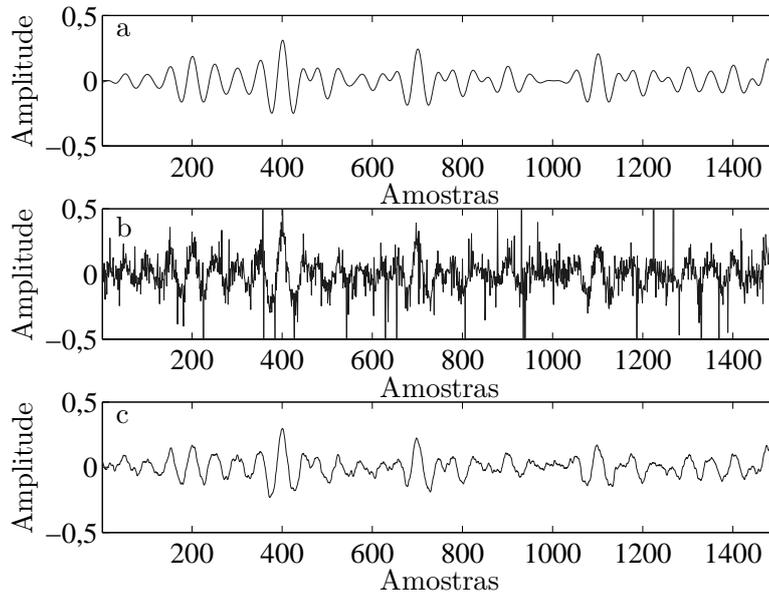


Figura 3.19: Formas de onda para $E_b/N_0 = 21$ dB: (a) s , (b) y e (c) \hat{s}

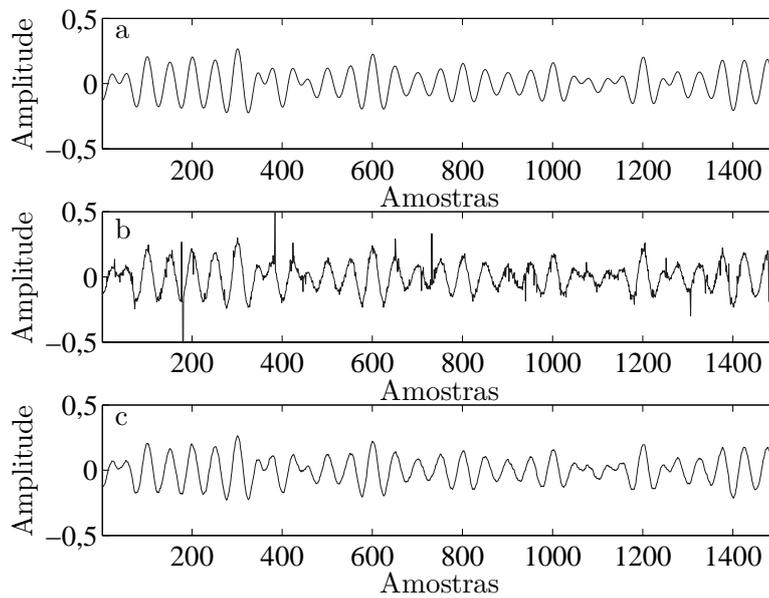


Figura 3.20: Formas de onda para $E_b/N_0 = 35$ dB: (a) s , (b) y e (c) \hat{s}

Afim de quantificar a melhoria de MSE relacionada com a redução do ruído composto para cada caso, foram calculados também os MSEs entre os sinais y e s e em seguida as relações dadas por:

$$\text{ganho por redução, dB} = 10 \log \left\{ \frac{e_{\text{MSE}}(\mathbf{y} - \mathbf{s}) - e_{\text{MSE}}(\hat{\mathbf{s}} - \mathbf{s})}{e_{\text{MSE}}(\mathbf{y} - \mathbf{s})} \right\},$$

em que $e_{\text{MSE}}(\mathbf{y} - \mathbf{s})$ e $e_{\text{MSE}}(\hat{\mathbf{s}} - \mathbf{s})$ são os erros quadráticos médios calculados entre os sinais \mathbf{y} e \mathbf{s} , e $\hat{\mathbf{s}}$ e \mathbf{s} , respectivamente. Os resultados estão apresentados na Tabela 3.2. Verifica-se que as reduções de ruído proporcionaram ganhos de MSE de até 19 dB.

Tabela 3.2: *Ganhos de MSE proporcionados pela redução do ruído composto.*

E_b/N_0 , dB	$e_{\text{MSE}}(\mathbf{y} - \mathbf{s})$	$e_{\text{MSE}}(\hat{\mathbf{s}} - \mathbf{s})$	ganho por redução, dB
7	0,7312	$9,1 \times 10^{-3}$	≈ 19
21	0,0337	$6,94 \times 10^{-4}$	$\approx 16,8$
35	$1,6 \times 10^{-3}$	$5,443 \times 10^{-5}$	$\approx 14,53$

No próximo capítulo, são apresentadas as curvas de desempenhos $\text{BER} \times E_b/N_0$ do sistema para valores de $\alpha = 1,9$, $\alpha = 1,7$ e $\alpha = 1,5$, sem e com a redução de ruído realizada por dispositivos não lineares e pela técnica COD.

Capítulo 4

Resultados numéricos

Neste capítulo apresenta-se primeiramente a configuração da simulação para a obtenção das curvas de desempenho do sistema. Em seguida, descreve-se o método adotado para determinar o fator de escalarização e o parâmetro de forma da função de penalização, utilizada na função objetivo do problema de otimização da técnica COD. Depois são apresentadas as curvas de desempenho sem a etapa de redução de ruído, para ruídos compostos com diferentes níveis de impulsividade. Na sequência é feita uma introdução sobre o funcionamento de dispositivos não lineares na entrada do receptor como forma de combater o ruído impulsivo e são descritas duas técnicas baseadas nesta abordagem: a técnica END (*empirical nonlinear-device denoising*) e a técnica END modificada. Curvas de desempenho destas duas técnicas para os diferentes cenários de ruído impulsivo considerados são apresentadas. Depois são apresentadas as curvas de desempenho do sistema com a redução de ruído realizada pela técnica COD para todos os níveis de impulsividade considerados. Por fim realiza-se a comparação destas curvas com os melhores resultados obtidos pelas técnicas de redução de ruído via detecção não linear analisadas.

4.1 Configuração da simulação

Para acelerar os cálculos foi escolhida uma frequência de amostragem da simulação de 10 k amostras/s, o que implica em uma largura de banda do ruído composto simulada igual a 5 kHz. Uma vez que a largura de banda do ruído ambiente real pode ultrapassar 250 kHz [4], 500 k amostras por segundo é uma frequência de amostragem suficiente para a maior parte das aplicações práticas. Desta forma, um fator de

50 é observado entre esta frequência e a frequência de amostragem da simulação. A fim de manter esta proporção nas simulações, foi utilizada uma portadora com uma frequência de 200 Hz, correspondente à uma onda portadora acústica real de 10 kHz. A taxa de símbolo foi reduzida de um valor típico real de 5 kb/s para 100 b/s, resultando em uma duração de símbolo correspondente a 2 períodos de portadora e 100 amostras da simulação. A Tabela 4.1 apresenta os valores adotados de parâmetros encontrados em aplicações práticas reais e os correspondentes valores simulados, em que o $fator = (\text{valores práticos}) / (\text{valores simulados}) = 50$. Na tabela, o parâmetro correspondente à largura de banda do ruído composto está representado por $BW(r)$, a frequência de amostragem da simulação por f_a , a frequência da portadora por f_p , e a taxa de bits por R_b . O processo $S\alpha S$ foi gerado para valores de α entre 1,5 e 1,9, em que $\alpha = 1,5$ simula o ruído composto com a maior impulsividade encontrada nas medidas realizadas em [5].

Tabela 4.1: Valores práticos e simulados dos parâmetros do sistema adotado.

$fator = 50$	$BW(r)$, kHz	f_a , amostra/s	f_p , kHz	R_b , kb/s
<i>Valores Práticos</i>	250	500	10	5
<i>Valores Simulados</i>	5	10	0,2	0,1

A simulação opera quadro a quadro, onde a duração do quadro define a dimensão dos vetores envolvidos no processo de redução de ruído: s , r , y e \hat{s} . Na prática, a duração do quadro deverá ser definida pela dependência entre a dimensão do vetor e a quantidade de tempo necessária para resolver o problema de otimização (2.9) antes de um novo quadro ser carregado. Se os dados recebidos não precisarem estar disponíveis em tempo real, as restrições sobre a duração do quadro e sobre o tempo de solução para o Problema (2.9) serão menores, que é o caso de algumas aplicações de redes de sensores submarinos [18]. Aqui, foi utilizado um tamanho de quadro $n = 1500$ amostras, com o objetivo de maximizar a relação entre o número de bits transmitidos por quadro e o tempo final da simulação, em que:

- menores valores de n reduzem o tempo necessário para resolver o problema de otimização, mas diminuem o número de bits por quadro, aumentando o número de quadros transmitidos e, desta forma, estendendo o tempo final de simulação;
- maiores valores de n aumentam o número de bits transmitidos por quadro, mas aumentam em maior proporção o tempo necessário para resolver o problema de otimização, estendendo o tempo final de simulação.

Assim como no capítulo 3, o critério de parada na simulação de Monte Carlo para

a obtenção dos resultados numéricos foi de 100 erros de bit a cada valor de E_b/N_0 . Esta quantidade de erros corresponde a um intervalo de confiança de aproximadamente $\pm 16\%$ em torno do valor real, com 90% de confiança [11].

4.1.1 Escolha dos valores dos parâmetros M e δ

O Problema de Otimização (2.9) da técnica COD não tem solução analítica e foi resolvido utilizando o CVX [19], seguindo a modelagem apresentada na seção 2.3. Para a escolha dos valores dos parâmetros do Problema (2.9), correspondentes ao fator de forma M da função de *Huber* e ao peso relativo δ , baseou-se na medida de MSE, dada pela Equação (1.2), entre o sinal transmitido livre de ruído s e o sinal estimado \hat{s} . Foi considerado o caso de ruído composto com a maior impulsividade registrada em [14], modelado pelo processo $S\alpha S$ com $\alpha = 1,5$. Sabendo-se que quanto menor o valor de $M > 0$ maior a robustez da técnica aos ruídos impulsivos, optou-se em buscar o melhor M entre $0,01 \leq M \leq 0,1$ com passos de 0,01. Para cada M , foram obtidas curvas de $MSE \times \delta$ resolvendo-se o problema de otimização da técnica COD para $0,1 \leq \delta \leq 6$, com passos de 0,01, para todos E_b/N_0 considerados na curva de validação. As Figuras, 4.1, 4.2 e 4.3 apresentam as curvas $MSE \times \delta$ para $E_b/N_0 = 7, 21, 35$ dB respectivamente. Os pontos indicados pelos números de 1 a 10 referem-se aos mínimos valores das curvas $MSE \times \delta$ para M de 0,01 a 0,1, respectivamente. O valor mínimo de MSE das curvas obtidas dentro da variação considerada de δ é simbolizado por MSE_{\min} . Os resultados mostram que, para cada E_b/N_0 , os valores mínimos de MSE para as curvas correspondentes aos diferentes valores de M ficam aproximados, porém com diferentes valores de δ ótimo, que é o δ responsável por levar a curva de MSE para o seu mínimo. Com o intuito de evitar a necessidade de se estimar o valor de E_b/N_0 para definir os valores dos parâmetros e proporcionar um sinal estimado cujo o MSE entre este sinal e o sinal desejado seja próximo a um dos pontos de MSE_{\min} obtidos, optou-se em escolher $M = 0,01$ e $\delta = 1$, pois é uma combinação que atinge um ponto próximo ao ponto de MSE_{\min} de número 1 em todos os cenários de E_b/N_0 analisados.

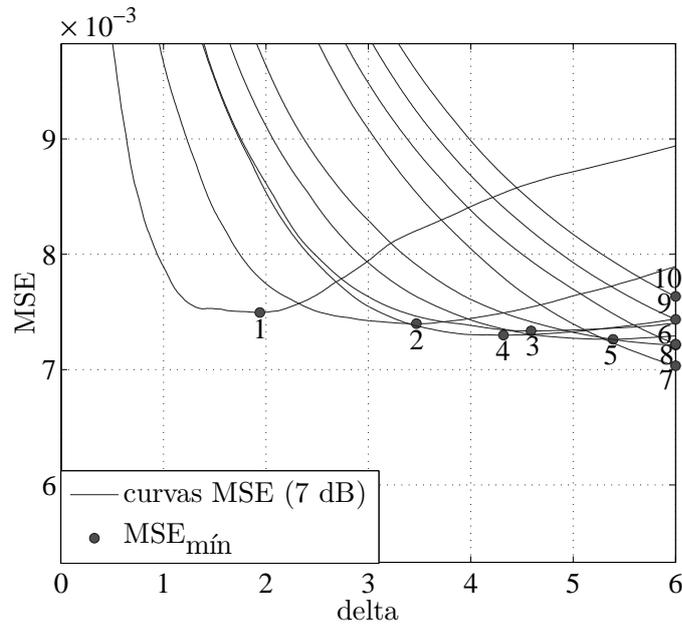


Figura 4.1: Curvas $MSE \times \delta$ para $E_b/N_0 = 7$ dB.

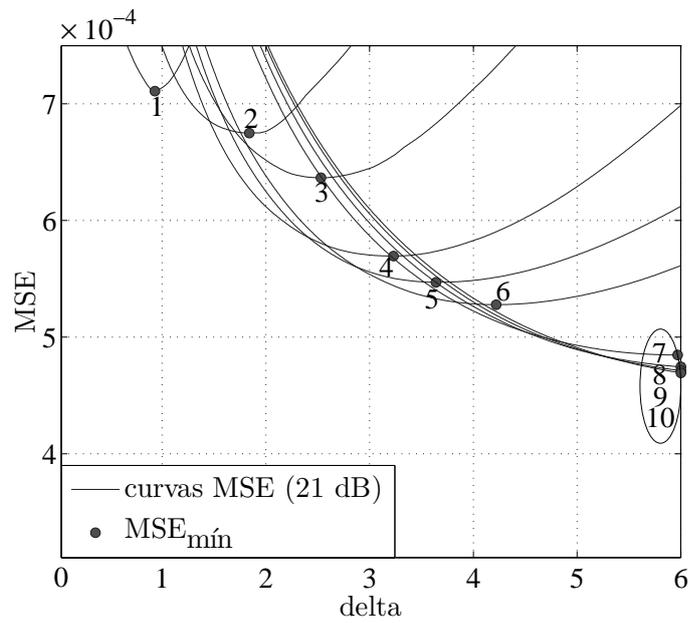


Figura 4.2: Curvas $MSE \times \delta$ para $E_b/N_0 = 21$ dB.

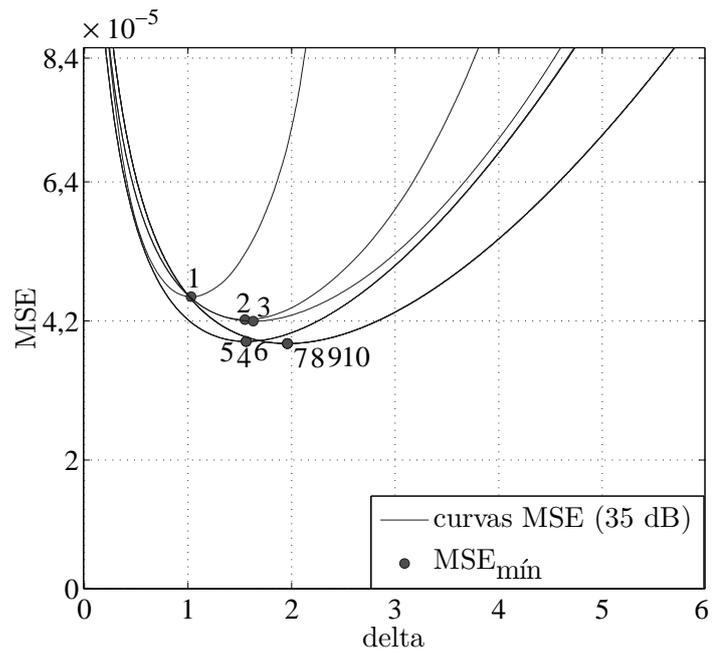


Figura 4.3: Curvas $MSE \times \delta$ para $E_b/N_0 = 35$ dB.

4.2 Desempenho do sistema UWAC sem redução de ruído

A Figura 4.4 apresenta a curva de desempenho teórica para o canal Rayleigh-AWGN e as curvas de desempenho do sistema adotado sob desvanecimento Rayleigh e ruído composto modelado por um processo aleatório $S\alpha S$ para $1,5 \leq \alpha \leq 2$, sem a etapa de redução de ruído. Observa-se que o desempenho é pior para o caso de $\alpha = 1,5$, o que representa o ruído composto com maior impulsividade encontrado em canais UWAC [5]. Para maiores valores de α , menores são as degradações do desempenho em relação à curva teórica, representando ruídos compostos com menores impulsividades. Para $\alpha = 2$ tem-se um ruído composto sem nenhuma impulsividade, representando um canal de ruído puramente Gaussiano. Neste caso, a curva de desempenho concorda com a curva teórica, uma vez que a curva teórica refere-se a um canal Rayleigh-AWGN. Estas curvas são as referências para as curvas de desempenho do sistema com a etapa de redução de ruído implementada.

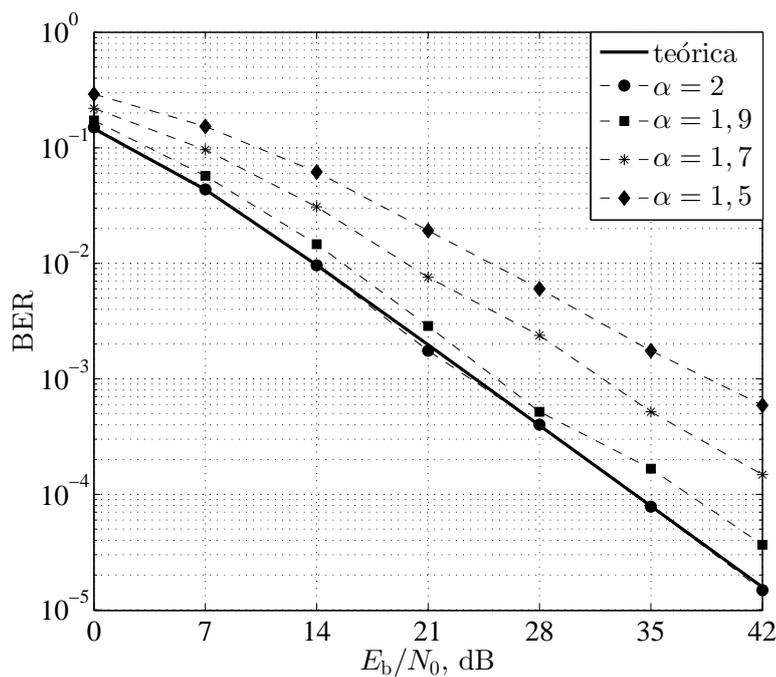


Figura 4.4: Desempenho do sistema para $1,5 \leq \alpha \leq 2$, sem redução de ruído.

4.3 Desempenho do sistema UWAC com redução de ruído via dispositivo não linear

Existem vários métodos para projetar detectores ótimos e subótimos locais para sinais corrompidos por ruído impulsivo [27] [5]. Um deles é implementado introduzindo um dispositivo não linear antes do filtro casado ou do correlator no receptor [5]. Entretanto, este dispositivo exige o conhecimento dos valores reais das amplitudes do sinal e das características impulsivas do ruído e geralmente não é adequado para sinais onde tais características são variantes no tempo. Aqui foi adotado um dispositivo com a seguinte função de transferência subótima:

$$y = \begin{cases} s & \text{se } |s| \leq t, \\ \text{sign}(s)t & \text{se } |s| > t. \end{cases}, \quad (4.1)$$

onde s é a entrada do dispositivo de ceifamento (*clipping device*), y é a saída e t é o limiar do corte. Esta função mantém sinais com baixas amplitudes sem distorção enquanto ceifa grandes ruídos impulsivos, melhorando o desempenho do detector. Entretanto, quando o sinal também é forte (com módulo de sua amplitude maior do que t), ele também é ceifado e o desempenho do detector piora. Para evitar isso, o limiar de ceifamento t deve ser adaptado de acordo com o desvio padrão σ da componente Gaussiana do processo SaS e de acordo com a amplitude real do sinal recebido. Esta amplitude depende do fator de desvanecimento η (que representa a variável aleatória com distribuição Rayleigh filtrada) e da amplitude do sinal modulado e filtrado (que por sua vez depende do *roll-off* do filtro RRC e da sequência de dados). Desconsiderando a sequência de dados, o valor subótimo de ceifamento deve ser

$$t = k_1\sigma + k_2\eta\sqrt{2P}, \quad (4.2)$$

em que P é a potência média do sinal recebido e k_1 e k_2 são escolhidos de modo a garantir que o sinal desejado não seja ceifado e que o ruído Gaussiano em torno deste sinal não seja totalmente ceifado. No entanto, em regime de alta SNR, as variações de amplitude do sinal dependente dos dados devem ser monitoradas com mais precisão para produzir uma melhoria considerável no desempenho. Sendo assim, o valor subótimo de t obtido a partir da Equação (4.2) não deverá ser capaz de representar corretamente o valor de ceifamento ideal para regime de alta SNR, reduzindo o desempenho do sistema.

Nota-se que a adaptação subótima acima não pode ser facilmente implementada na

prática, uma vez que vários parâmetros devem ser estimados ou conhecidos *a priori*. Assim, propõe-se uma adaptação alternativa de t em função da potência média de séries temporais, ρ , das amostras recebidas em cada quadro, depois da etapa de conversão DC, de acordo com a regra empírica simples dada por

$$t = F\sqrt{\rho}, \quad (4.3)$$

em que

$$\rho = \frac{1}{n} \sum_{i=1}^n (s_i)^2 \quad (4.4)$$

e F é um fator de calibração. Esta técnica foi designada por técnica de redução de ruído por dispositivo não linear empírica (END, *empirical nonlinear-device denoising*).

As Figuras 4.5, 4.6 e 4.7 apresentam as curvas de desempenho da técnica END para alguns valores de F para $\alpha = 1, 5; 1, 7$ e $1, 9$ respectivamente.

Na Figura 4.5 observa-se que o desempenho do sistema com a técnica END é melhor do que o desempenho do sistema sem redução de ruído para todos os valores de F considerados. A curva referente a $F = 1$ apresenta os melhores resultados para $E_b/N_0 < 42$ dB mas tem o desempenho penalizado em $E_b/N_0 = 42$ dB. A curva referente a $F = 2$ apresenta o melhor resultado em $E_b/N_0 = 42$ dB mas tem seu desempenho penalizado para $E_b/N_0 < 42$ dB. As curvas referentes aos valores intermediários de F apresentam resultados comparáveis em todos valores de E_b/N_0 considerados.

Na Figura 4.6 observa-se que o desempenho do sistema com a técnica END é melhor do que o desempenho do sistema sem redução de ruído para todos os valores de F considerados. A curva referente a $F = 1$ apresenta bons resultados para $E_b/N_0 \leq 28$ dB mas é penalizada para $E_b/N_0 > 28$. A curva referente a $F = 3$ apresenta o melhor resultado em $E_b/N_0 = 42$ dB mas é penalizada nos outros casos. As curvas referentes aos valores intermediários de F apresentam resultados comparáveis para $E_b/N_0 \leq 35$ dB e em $E_b/N_0 = 42$ dB o melhor resultado é da curva referente a $F = 2$.

Na Figura 4.7 nota-se que todas as curvas apresentam resultados comparáveis para $E_b/N_0 \leq 28$ dB. Para $E_b/N_0 > 28$ dB os melhores resultados apresentados referem-se a curva cujo $F = 3$. Em $E_b/N_0 = 42$ dB a curva com $F = 3$ foi a única que apresentou desempenho melhor do que a curva sem redução de ruído, seguida pelas curvas com $F = 2; 1, 5$ e 1 . Isto ocorre porque para este E_b/N_0 o sinal possui o valor

do módulo da sua amplitude maior do que o valor do limiar de ceifamento t estimado para $F = 2; 1,5$ e 1 , resultando no ceifamento do sinal e consequentemente na queda do desempenho.

A partir das análises realizadas anteriormente pode-se concluir que existe uma relação de compromisso entre as taxas de erro do sistema com a técnica END para os regimes de baixo e alto E_b/N_0 , em que: se F é ajustado para produzir um bom desempenho para $E_b/N_0 < 42$ dB, o desempenho em $E_b/N_0 = 42$ dB é penalizado; se F é ajustado para produzir um desempenho satisfatório em $E_b/N_0 = 42$ dB, o desempenho em valores menores de E_b/N_0 é penalizado.

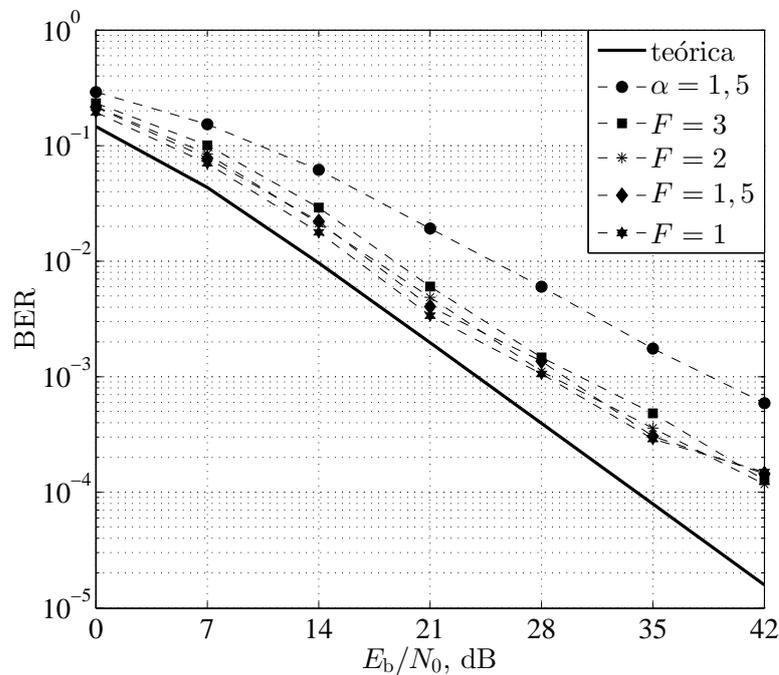


Figura 4.5: Desempenho do sistema com redução de ruído pela técnica END para $F = 3; 2; 1,5; 1$ e $\alpha = 1,5$.

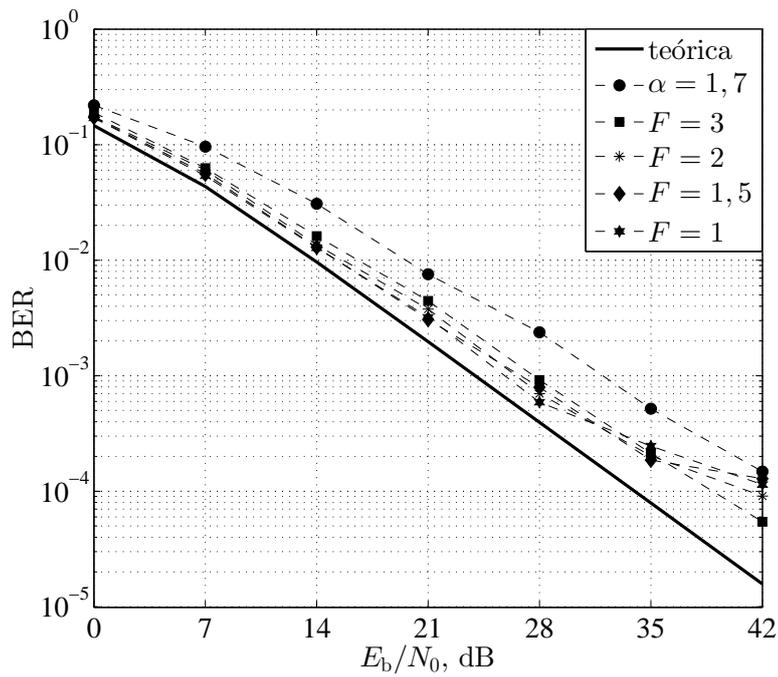


Figura 4.6: Desempenho do sistema com redução de ruído pela técnica END para $F = 3; 2; 1,5; 1$ e $\alpha = 1,7$.

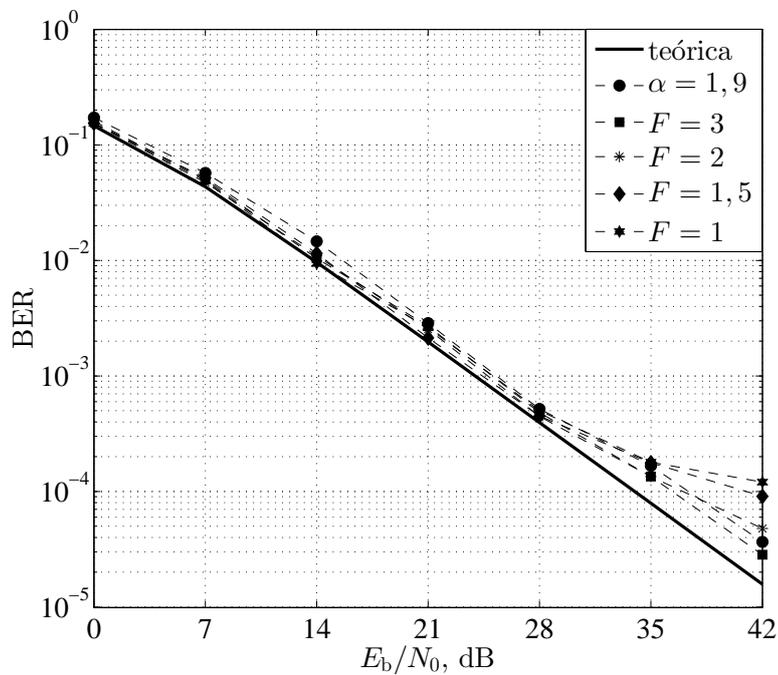


Figura 4.7: Desempenho do sistema com redução de ruído pela técnica END para $F = 3; 2; 1,5; 1$ e $\alpha = 1,9$.

Outra abordagem para a obtenção de t com base na Equação (4.3) foi realizada ordenando-se as amostras de cada quadro em ordem crescente e posteriormente calculando-se ρ excluindo as amostras iniciais e finais do quadro, objetivando eliminar grande parte das amostras de ruído impulsivo para a obtenção de um melhor limiar do que o limiar obtido pela técnica END. Esta nova abordagem foi designada por técnica END modificada.

As Figuras 4.8, 4.9 e 4.10 apresentam as curvas de desempenho da técnica END modificada para $\alpha = 1, 5; 1, 7$ e $1, 9$ respectivamente. Aqui o cálculo de ρ foi realizado a partir da exclusão das 200 amostras iniciais e finais de cada quadro ordenado.

Na Figura 4.8 observa-se que a atuação da técnica END modificada melhora o desempenho do sistema quando comparada a atuação da técnica END em $E_b/N_0 \leq 35$ dB para todos valores de F considerados. Em $E_b/N_0 = 42$ não ocorre melhoria de desempenho somente na curva referente a $F = 3$.

Na Figura 4.9 é possível observar que a atuação da técnica END modificada melhora o desempenho do sistema quando comparada a atuação da técnica END em $E_b/N_0 \leq 28$ para todos valores de F considerados. Para $E_b/N_0 > 28$ o desempenho do sistema com a técnica END modificada continua sendo melhor do que o desempenho do sistema com a técnica END nas curvas com $F = 1$ e $1, 5$, enquanto que para as curvas com $F = 2$ e 3 o desempenho do sistema é melhor com a técnica END.

Na Figura 4.10 nota-se que todas as curvas apresentam resultados comparáveis entre si e aos respectivos resultados proporcionados pela técnica END para $E_b/N_0 \leq 28$ dB. Para $E_b/N_0 = 35$ dB a atuação da técnica END modificada é melhor do que a atuação da técnica END para as curvas com valores de $F = 1; 1, 5$ e 2 enquanto que para a curva com $F = 3$ o desempenho do sistema é melhor com a técnica END. Para $E_b/N_0 = 42$ dB a atuação da técnica END modificada continua sendo melhor do que a atuação da técnica END apenas para a curva com $F = 1$, enquanto que para a curva com $F = 1, 5$ o desempenho é comparável e para as curvas com $F = 2$ e 3 o desempenho é melhor para a técnica END.

Assim como na técnica END, também é possível observar a partir dos resultados obtidos pela técnica END modificada que existe uma relação de compromisso entre as taxas de erro de bit do sistema com a técnica END modificada para os regimes de baixo e alto E_b/N_0 .

Comparando todos os resultados obtidos das duas técnicas de redução de ruído via detecção não linear abordadas, conclui-se que os melhores resultados de uma maneira

geral são proporcionados pela técnica END modificada com $F = 1$. Estes resultados serão utilizados para fins de comparação com os respectivos resultados obtidos pela técnica COD. Também serão utilizados para comparações os resultados proporcionados pela técnica END com $F = 3$ por apresentarem os melhores resultados para $E_b/N_0 = 42$ dB em $\alpha = 1,7$ e $1,9$. As comparações estão apresentadas na próxima seção.

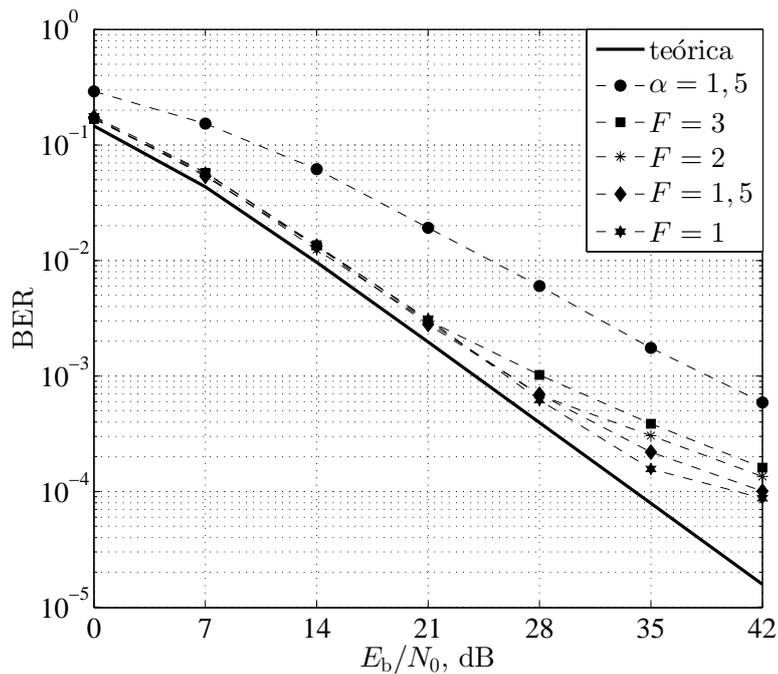


Figura 4.8: Desempenho do sistema com redução de ruído pela técnica END modificada para $F = 3; 2; 1,5; 1$ e $\alpha = 1,5$ com o cálculo de ρ para $i = 200 \dots 1300$.

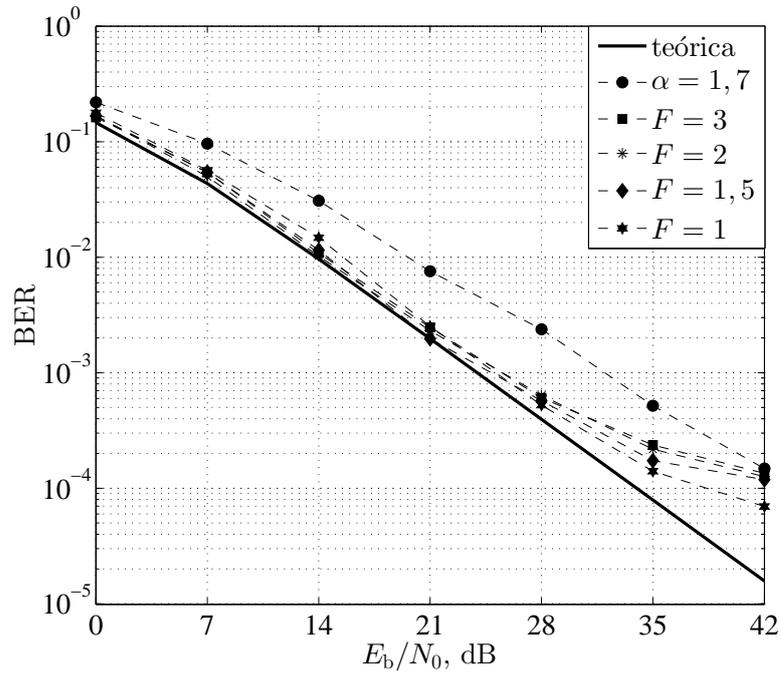


Figura 4.9: Desempenho do sistema com redução de ruído pela técnica END modificada para $F = 3; 2; 1, 5; 1$ e $\alpha = 1, 7$ com o cálculo de ρ para $i = 200 \dots 1300$.

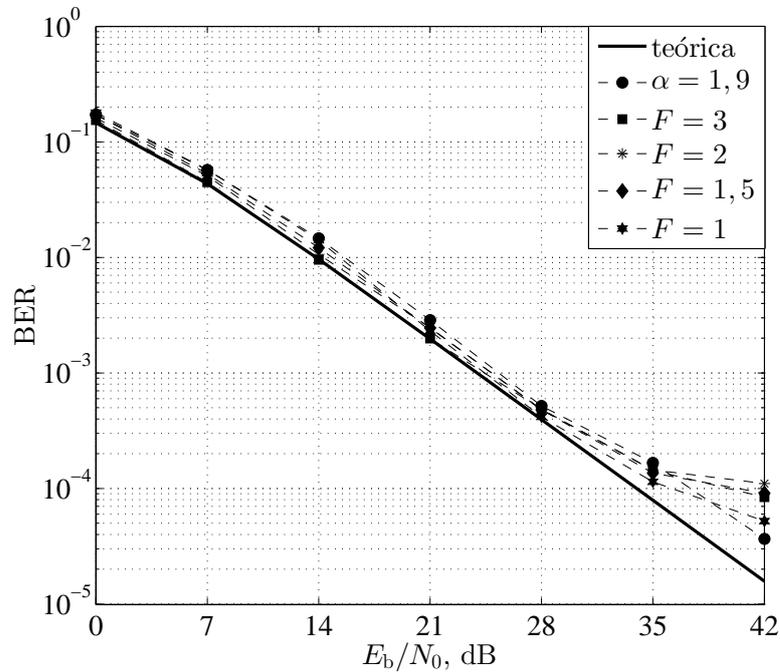


Figura 4.10: Desempenho do sistema com redução de ruído pela técnica END modificada para $F = 3; 2; 1, 5; 1$ e $\alpha = 1, 9$ com o cálculo de ρ para $i = 200 \dots 1300$.

4.4 Desempenho do sistema UWAC com redução de ruído via otimização convexa

As curvas de desempenho do sistema adotado com redução de ruído realizada pela técnica COD com $M = 0,01$ e $\delta = 1$ estão apresentadas nas Figuras 4.11, 4.12 e 4.13, para $\alpha = 1,5$; $1,7$ e $1,9$, respectivamente. Também estão apresentadas nas figuras as curvas de desempenho do sistema com redução de ruído realizada pela técnica END modificada com $F = 1$ e pela técnica END com $F = 3$ para fins de comparação.

Na Figura 4.11 verifica-se que a redução de ruído realizada pela técnica COD proporciona significativa melhora de desempenho em comparação com a curva sem redução de ruído para $\alpha = 1,5$ em todos valores de E_b/N_0 considerados. Verifica-se também que o desempenho proporcionado pela técnica COD é melhor do que o desempenho da técnica END para todos valores de E_b/N_0 e melhor do que o desempenho da técnica END modificada para $E_b/N_0 \geq 14$ dB. Para se atingir uma taxa de erro de bit de 10^{-4} por exemplo, são necessários aproximadamente 6 dB a mais de potência de transmissão no sistema com a técnica END modificada do que no sistema com a técnica COD. Com $E_b/N_0 = 42$ dB, o sistema com a técnica COD alcança uma taxa de erro de bit de aproximadamente $2,8 \times 10^{-5}$ enquanto que com a técnica END modificada o alcance limita-se a aproximadamente $8,76 \times 10^{-5}$ e com a técnica END limita-se a aproximadamente $1,27 \times 10^{-4}$. Nota-se que a curva de desempenho da técnica END modificada afasta-se da curva teórica para $E_b/N_0 \geq 35$ dB.

Na Figura 4.12 nota-se que a redução de ruído realizada pela técnica COD proporciona significativa melhora de desempenho em comparação com a curva sem redução de ruído para $\alpha = 1,7$ em todos valores de E_b/N_0 considerados. Também observa-se que o desempenho da técnica COD é melhor do que o desempenho da técnica END para todos valores de E_b/N_0 e melhor do que o desempenho da técnica END modificada para $E_b/N_0 \geq 21$ dB, a menos para $E_b/N_0 = 28$ dB. Para se atingir uma taxa de erro de bit de 10^{-4} por exemplo, são necessários aproximadamente 3 dB a mais de potência de transmissão no sistema com a técnica END modificada do que no sistema com a técnica COD. Com $E_b/N_0 = 42$ dB, o sistema com a técnica COD alcança uma taxa de erro de bit de aproximadamente $2,1 \times 10^{-5}$ enquanto que com a técnica END modificada o alcance limita-se a aproximadamente 7×10^{-5} e com a técnica END limita-se a aproximadamente $5,4 \times 10^{-5}$. Também pode-se observar nesta figura que a curva de desempenho da técnica END modificada afasta-se da curva teórica para $E_b/N_0 \geq 35$.

Em relação a Figura 4.13, verifica-se que a redução de ruído realizada pela técnica COD proporciona significativa melhora de desempenho em comparação com a curva sem redução de ruído para $\alpha = 1,9$ em $E_b/N_0 \geq 35$ dB e resultados comparáveis nos outros casos. Também verifica-se que a técnica END apresenta melhores resultados para $E_b/N_0 \leq 14$ dB, a técnica END modificada apresenta melhores resultados para $14 < E_b/N_0 \leq 28$ dB e a técnica COD é melhor para $E_b/N_0 > 28$ dB. As curvas que apresentam piores resultados para $E_b/N_0 \leq 35$ dB apresentam resultados comparáveis. Para $E_b/N_0 = 42$ dB, o sistema com a técnica COD alcança uma taxa de erro de bit de aproximadamente $1,9 \times 10^{-5}$ enquanto que com a técnica END modificada o alcance limita-se a aproximadamente $5,2 \times 10^{-5}$ e com a técnica END limita-se a aproximadamente $2,8 \times 10^{-5}$. A técnica END modificada neste ponto apresenta pior resultado que o sistema sem redução de ruído que alcança uma taxa de aproximadamente $3,8 \times 10^{-5}$. Também é possível observar na figura o afastamento da curva da técnica END modificada a medida que se aumenta o valor de E_b/N_0 dB, especificamente para $E_b/N_0 \geq 35$ dB.

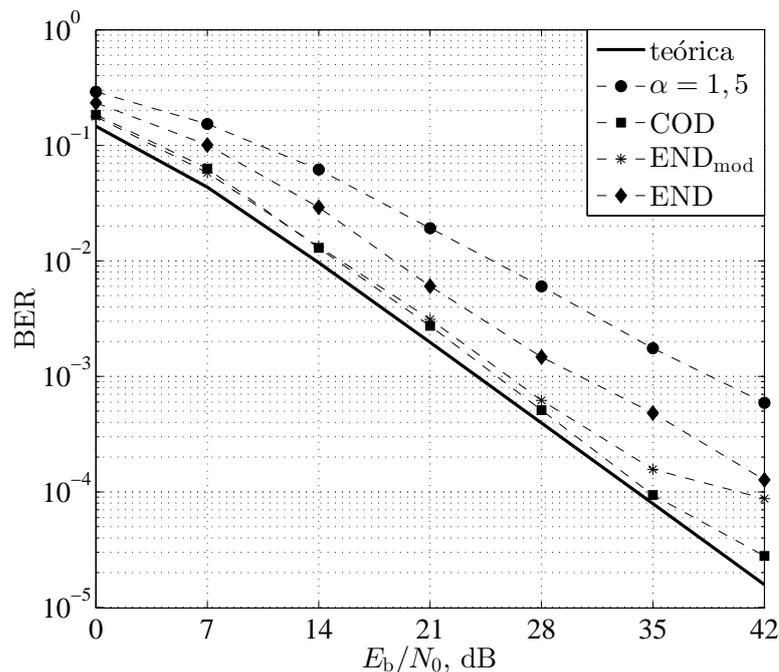


Figura 4.11: Desempenho das técnicas COD com $M = 0,01$ e $\delta = 1$, END modificada com $F = 1$ e ρ com $i = 200 \dots 1300$ e END com $F = 3$ para $\alpha = 1,5$.

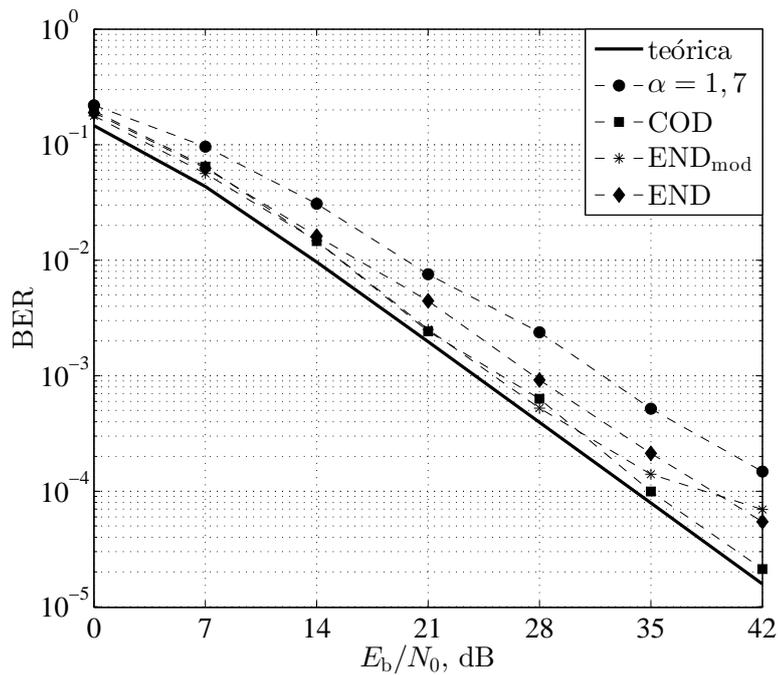


Figura 4.12: Desempenho das técnicas COD com $M = 0,01$ e $\delta = 1$, END modificada com $F = 1$ e ρ com $i = 200 \dots 1300$ e END com $F = 3$ para $\alpha = 1,7$.

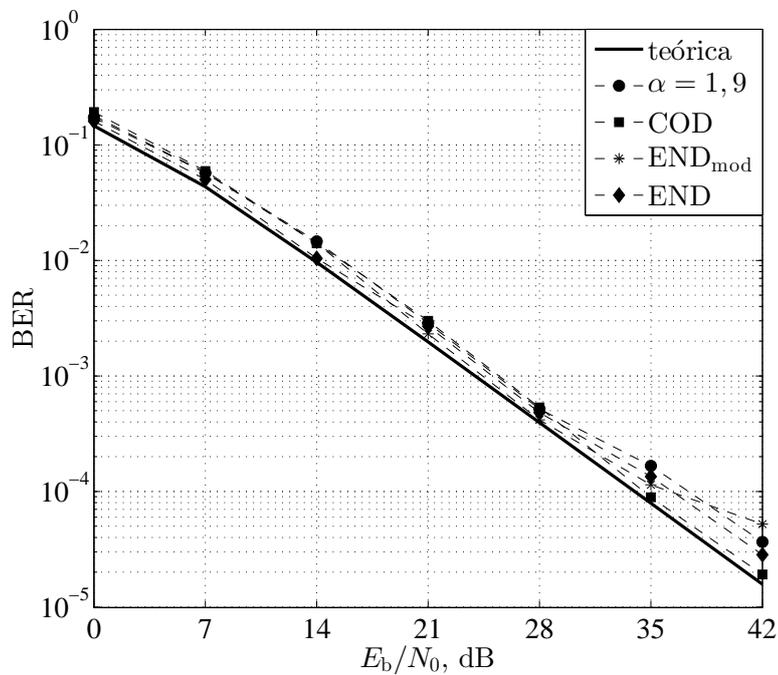


Figura 4.13: Desempenho das técnicas COD com $M = 0,01$ e $\delta = 1$, END modificada com $F = 1$ e ρ com $i = 200 \dots 1300$ e END com $F = 3$ para $\alpha = 1,9$.

Capítulo 5

Conclusões e oportunidades para novas pesquisas

Neste último capítulo são destacadas as principais vantagens e desvantagens da técnica COD e descritas as oportunidades para continuidade das pesquisas.

5.1 Conclusões

Nesta dissertação foi proposta uma técnica de redução de ruídos baseada em otimização convexa, buscando reduzir o ruído composto pelo ruído térmico mais ruído ambiente típico em sistemas UWAC em águas marítimas quentes e rasas, onde o ruído ambiente impulsivo produzido pelos camarões de estalo é comum e degrada severamente o desempenho dos sistemas de comunicação. Foi visto que a técnica COD pode reduzir significativamente a BER do sistema UWAC na presença de ruído composto sem a necessidade de nenhuma estimação ou adaptação de parâmetros em função da SNR. Também não há a necessidade de detecção do ruído impulsivo antes do sinal recebido ser processado, condição necessária para outras técnicas de redução de ruído impulsivo existentes.

Para as condições de forte e média impulsividade, simuladas por $\alpha = 1, 5$ e $1, 7$ no processo $S\alpha S$ respectivamente, a técnica COD proporcionou melhoria de desempenho para todos casos de E_b/N_0 analisados, restritos à faixa de $0 \leq E_b/N_0 \leq 42$ dB. Para a condição de fraca impulsividade, simulada por $\alpha = 1, 9$, verificou-se que a técnica COD proporcionou melhoria de desempenho para $E_b/N_0 \geq 35$ dB e resultados com-

paráveis à curva sem redução de ruído para os outros casos. Para todas as condições de ruído consideradas, verificou-se que as curvas da técnica END modificada se afastam da curva teórica para $E_b/N_0 > 28$ dB, resultando em piores desempenhos quando comparados aos respectivos pontos das curvas da técnica COD. Comparada a técnica END, a técnica COD apresentou melhores resultados em todos os pontos na condição de $\alpha = 1,5$, para $E_b/N_0 \geq 14$ dB na condição de $\alpha = 1,7$ e para $E_b/N_0 \geq 35$ dB na condição de $\alpha = 1,9$.

Há uma limitação no tempo necessário para resolver o problema de otimização da técnica COD, uma desvantagem que pode não existir em muitas aplicações de redes de sensores subaquáticas. Nestas aplicações, o dado recebido pode ser processado em um tempo maior por um nó de recepção com alta capacidade computacional. Além disso, aplicações em tempo real [18] podem ser realizadas em um futuro próximo devido à proliferação do conceito de otimização convexa em tempo real [28].

5.2 Oportunidades para novas pesquisas

Uma vez que a otimização convexa em tempo real ainda não está difundida ao ponto de ser aplicável ao problema em questão, uma oportunidade para nova pesquisa seria tentar acelerar a solução do problema utilizando os métodos de solução por primeira ordem acelerada [29]. Isto porque os métodos de ponto-interior adotados no CVX são computacionalmente ineficientes para resolver o problema da técnica COD, devido à utilização da função de Huber, principalmente. Outra possível continuidade das pesquisas, seria a proposta de métodos alternativos ao adotado neste trabalho para a escolha dos valores dos parâmetros M e δ do problema de otimização da técnica COD, com o objetivo de se obter curvas de desempenho mais próximas da curva teórica. Seguindo este mesmo objetivo, somado ao objetivo de se obter técnicas de redução de ruído cada vez menos complexas, poderiam ser sugeridas também novas modelagens de problemas de otimização convexa para a redução do ruído impulsivo. Além disso, seria interessante comparar o desempenho e a complexidade da técnica COD com outras soluções para o problema de redução do ruído provocado pelos camarões de estalo, como, por exemplo, as apresentadas em [30] e [31].

Referências Bibliográficas

- [1] S. V. Vaseghi, *Advanced digital signal processing and noise reduction*. John Wiley & Sons, 2008.
- [2] J. Harikiran, B. Saichandana, and B. Divakar, “Impulse noise removal in digital images,” *International Journal of Computer Applications*, vol. 10, no. 8, 2010.
- [3] B. N. Kim, J. Hahn, B. K. Choi, and B. C. Kim, “Acoustic characteristics of pure snapping shrimp noise measured under laboratory conditions,” *Proceedings of Symposium on Ultrasonic Electronics*, vol. 30, pp. 167–168, 2009.
- [4] M. Legg, A. Zaknich, A. Duncan, and M. Greening, “Analysis of impulsive biological noise due to snapping shrimp as a point process in time,” in *OCEANS 2007 - Europe*, June 2007, pp. 1–6.
- [5] M. Chitre, J. Potter, and S.-H. Ong, “Optimal and near-optimal signal detection in snapping shrimp dominated ambient noise,” *Oceanic Engineering, IEEE Journal of*, vol. 31, no. 2, pp. 497–503, April 2006.
- [6] J. Panaro, F. Lopes, L. M. Barreira, and F. E. Souza, “Underwater acoustic noise model for shallow water communications,” in *Brazilian Telecommunication Symposium*, 2012.
- [7] P. Chatterjee and P. Milanfar, “Is denoising dead?” *Image Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 895–911, April 2010.
- [8] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2009.
- [9] M. Grant, *Disciplined Convex Programming*, Information Systems Laboratory, Department of Electrical Engineering, Stanford University, 2004. [Online]. Available: http://stanford.edu/boyd/disc_cvx_prog.html
- [10] M. Grant, S. Boyd, and Y. Ye, *Disciplined convex programming*. Springer, 2006.

- [11] D. A. Guimaraes, *Digital Transmission - A Simulation-Aided Introduction with VisSim/Comm*. Springer, 2010.
- [12] J. G. Proakis, *Wiley encyclopedia of telecommunications*. John Wiley & Sons, 2003, vol. 4.
- [13] A. Quazi and W. Konrad, "Underwater acoustic communications," *Communications Magazine, IEEE*, vol. 20, no. 2, pp. 24–30, March 1982.
- [14] M. Chitre, "Underwater acoustic communications in warm shallow water channels," *P. hD. Thesis, Singapore: National University of Singapore*, 2006.
- [15] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li, "Research challenges and applications for underwater sensor networking," in *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, vol. 1. IEEE, 2006, pp. 228–235.
- [16] F. Yunus, S. Ariffin, and Y. Zahedi, "A survey of existing medium access control (mac) for underwater wireless sensor network (uwsn)," in *Mathematical/Analytical Modelling and Computer Simulation (AMS), 2010 Fourth Asia International Conference on*, May 2010, pp. 544–549.
- [17] M. Erol-Kantarci, H. T. Mouftah, and S. Oktug, "A survey of architectures and localization techniques for underwater acoustic sensor networks," *Communications Surveys & Tutorials, IEEE*, vol. 13, no. 3, pp. 487–502, 2011.
- [18] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad hoc networks*, vol. 3, no. 3, pp. 257–279, 2005.
- [19] M. Grant, S. Boyd, and Y. Ye, "Cvx: Matlab software for disciplined convex programming," 2008. [Online]. Available: <http://cvxr.com/cvx>
- [20] G. Strang, *Introduction to linear algebra*, 2003.
- [21] M. Stojanovic, "Acoustic (underwater) communications," *Encyclopedia of Telecommunications*, 2003.
- [22] T. Melodia, H. Kulhandjian, L.-C. Kuo, and E. Demirors, "Advances in underwater acoustic networking," *Mobile Ad Hoc Networking: Cutting Edge Directions*, Eds. S. Basagni, M. Conti, S. Giordano and I. Stojmenovic, John Wiley and Sons, Inc., Hoboken, NJ, 2013.

-
- [23] “Aquatec group.” [Online]. Available: <http://www.aquatecgroup.com/images/datasheets/aquamodem1000.pdf>
- [24] “develogic subsea systems.” [Online]. Available: http://www.develogic.de/wp-content/uploads/2014/04/HAM.Node_HAM.Base-04-2014.pdf
- [25] “Teledyne benthos.” [Online]. Available: http://benthos.teledyneweb.com/product/acoustic_modems/910-series-atm-915
- [26] “Função do matlab para gerar dados aleatórios a partir da distribuição αs .” [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/37514-stbl--alpha-stable-distributions-for-matlab>
- [27] J. Miller and J. Thomas, “The detection of signals in impulsive noise modeled as a mixture process,” *IEEE Transactions on Communications*, vol. 24, pp. 559–563, 1976.
- [28] J. Mattingley and S. Boyd, “Real-time convex optimization in signal processing,” *Signal Processing Magazine, IEEE*, vol. 27, no. 3, pp. 50–61, May 2010.
- [29] T. L. Jensen, *First-order Convex Optimization Methods for Signal and Image Processing*. Multimedia Information and Signal Processing, Department of Electronic Systems, Aalborg University, 2011.
- [30] H. Ou, J. S. Allen, and V. L. Syrmos, “Frame-based time-scale filters for underwater acoustic noise reduction,” *Oceanic Engineering, IEEE Journal of*, vol. 36, no. 2, pp. 285–297, 2011.
- [31] T. Suzuki, H. M. Tran, and T. Wada, “An underwater acoustic ofdm communication system with shrimp (impulsive) noise cancelling,” in *Computing, Management and Telecommunications (ComManTel), 2014 International Conference on*. IEEE, 2014, pp. 152–156.

Apêndice A

Simulação do sistema no MATLAB

Nas próximas páginas, estão apresentadas as linhas de código comentadas referentes a dois *Softwares* desenvolvidos no Matlab para o levantamento de todas as curvas de desempenho apresentadas nesta dissertação. O primeiro *Software* possibilita o levantamento das curvas de desempenho do sistema sem redução de ruído e com redução de ruído realizada pela técnica COD. O segundo *Software* possibilita o levantamento das curvas de desempenho do sistema sem redução de ruído e com redução de ruído realizada pelas técnicas de redução via dispositivos não lineares, as técnicas END e END adaptada.

Antes de simular o código, deve-se instalar o CVX, disponível em [19]. Também é necessário a inclusão da função referente a geração de dados aleatórios pela distribuição $S\alpha S$, disponível em [26]. Para caber nas páginas, algumas funções do código foram divididas ao meio. Isto pode causar algum erro caso queira-se replicar o código diretamente no Matlab para rodar a simulação.

```

%Software utilizado para a obtenção das curvas de desempenho da técnica
%COD apresentadas na dissertação:
%"Otimização Convexa Aplicada à Redução de Ruído Impulsivo em Sistemas de
%Comunicação Acústica Sob a Água.

%Desenvolvido por:
%Lucas Silvestre Chaves

%Data: 31/01/2015
%Revisão: R2

%Este software foi utilizado para simular o sistema adotado na dissertação
%e obter as curvas de desempenho do sistema adotado sem redução de ruído e
%com redução de ruído para todos cenários de ruído impulsivo
%considerados. A técnica de redução de ruído utilizada nesta simulação é a
%técnica proposta, designada por técnica COD.
%A simulação está configurada para obter e apresentar as curvas de
%desempenho referente ao sistema adotado sem redução e com redução por
%técnica COD para  $\alpha = 1,9$ ;  $1,7$  e  $1,5$ . Também é obtida e apresentada a
%curva de validação do modelo, referente ao sistema adotado sem redução de
%ruído para  $\alpha = 2$ .

%% Inicialização da simulação

clear all;
close all;
clc;

%% Definições dos parâmetros da distribuição Salphas

alpha = 1.5; %Definição do expoente característico.
skew = 0; %Definição do parâmetro de simetria.
scale = 0.01; %Definição do parâmetro de escala.
location = 0; %Definição do parâmetro de localização.

%% Definição dos parâmetros do sistema adotado.

Ms = 2; %Número de símbolos da constelação.
k = log2(Ms); %Número de bits por símbolo.
Cpacote = 1500; %Tamanho do quadro.
n = 10*Cpacote; %Número de bits enviados por laço para estimação da BER.

nSamp = 100; %Número de amostras por símbolo.

Eb = 1; %Energia por bit.

Rb = 100; %Taxa de bits enviados.
Rs = Rb/k; %Taxa de símbolos enviados.
Tsimb = 1/Rs; %Tempo por símbolo.

Fs = nSamp*Rs; %Frequência de amostragem da simulação.
Ts = 1/Fs; %Tempo de amostragem.

fosc = 200; %Frequência da portadora.

Nerros = 100; %Critério de parada para a simulação MonteCarlo.

%% Parâmetros do problema de otimização

M = 0.01; %Parâmetro de forma da função Huber.
delta = 1; %Peso relativo do problema de reconstrução de sinais.
lambda = [1; delta]; %Cria o vetor de parâmetros de escalarização do
%problema bicritério escalarizado.

%% Criação do Modulador e do Demodulados

hMod = modem.pskmod(Ms); % Cria um modulador M-PSK.
hMod.InputType = 'Bit'; % Define a entrada como bits.
hMod.SymbolOrder = 'Gray'; % Define um mapeamento Gray para os símbolos.
hDemod = modem.pskdemod(hMod); % Cria o B-PSK com as definições anteriores.

```

```

%% Projeto do filtro Raiz Cosseno Elevado

N = 10*nSamp; % Define a ordem do filtro.
rolloff = 1; % Define o fator de Rolloff do filtro.
filtDef = fdesign.pulseshaping(nSamp, 'Square Root Raised Cosine', ...
    'N,Beta', N, rolloff); % Define a resposta ao impulso do filtro.
rrcFilter = design(filtDef); % Cria o filtro.
rrcFilter.Numerator = rrcFilter.Numerator * sqrt(nSamp); % Normaliza.

%% Laço da simulação

EbNo = 0:7:42; %Define o vetor E_b/N_0 para o cálculo da BER.

BER = zeros(7,length(EbNo)); %Define a matriz BER para armazenar as BERs.

for flag = 1:1:7 %Laço de levantamento das curvas de desempenho do sistema.

    if(flag == 1)
        alpha = 2; %Curva de validação.
    end
    if(flag == 2)
        %Desempenho do sistema adotado sem redução de ruído para alpha=1,9.
        alpha = 1.9;
    end
    if(flag == 3)
        %Desempenho do sistema adotado sem redução de ruído para alpha=1,7.
        alpha = 1.7;
    end
    if(flag == 4)
        %Desempenho do sistema adotado sem redução de ruído para alpha=1,5.
        alpha = 1.5;
    end
    if(flag == 5)
        %Desempenho do sistema adotado com redução realizada pela técnica
        %COD para alpha=1,9.
        alpha = 1.9;
    end
    if(flag == 6)
        %Desempenho do sistema adotado com redução realizada pela técnica
        %COD para alpha=1,7.
        alpha = 1.7;
    end
    if(flag == 7)
        %Desempenho do sistema adotado com redução realizada pela técnica
        %COD para alpha=1,5.
        alpha = 1.5;
    end
end

for a = 1:1:length(EbNo) %Laço de cada curva.

    Nerros_aux = 0; %Zera a variável de contagem dos erros.
    cont = 0; %Contagem dos laços para contar o número de bits enviados.

    while (Nerros_aux <= Nerros) %Laço do levantamento de cada BER.
        %% Fonte binária

        %Gera uma sequência de bits aleatórios equiprováveis.
        b = randi([0 1],n,1);

        %% Modulação

        d = (-1)*modulate(hMod,b); %Modula a sequência dos bits geradas.

        %% Canal Rayleigh

        %Simula um canal Rayleigh plano e lento
        d = raylrnd(sqrt(2)/2,length(d),1).*real(d);

        %% Filtragem do sinal

        drealm = real(d); %Recebe a parte real de d.

```

```

dimag = imag(d); %Recebe a parte imaginária de d.

%Amostra cada símbolo em nsamp amostras.
dUpreal = upsample(dreal, nSamp); %Parte real.
dUpimag = upsample(dimag, nSamp); %Parte imaginária.

%Filtra os símbolos amostrados pelo filtro raiz cosseno elevado.
%Não foi necessário a utilização da equalização Sinc Inversa (SI)
%com foi apresentado no sistema adotado,
%pois o sinal a ser filtrado aqui é um impulso e uma sequência
%de zeros. Desta forma, o resultado da convolução deste filtro
%com o trem de impulsos espaçados por zeros é o mesmo que a
%convolução de um filtro raiz cosseno elevado com SI realizado
%com um trem de pulsos.
dTxreal = filter(rrcFilter, dUpreal); %Parte real.
dTximag = filter(rrcFilter, dUpimag); %Parte imaginária.

%% Conversão para banda passante (C.p/C.)

%Define a base de tempo para as funções cossenoidais.
t = [0: 1/Fs: (length(dTxreal) - 1)*1/Fs];

%Gera a portadora. O fator sqrt(2) garante que a potência da
%frequência do sinal em banda passante seja igual a potência
%da frequência duplicada do sinal em banda base.
%Ver: Passband Modulation with Adjacent Channel Interference.
carrier = sqrt(2)*exp(1i*2*pi*fosc*t);

%Multiplica o sinal a ser transmitido pela portador.
sTxup = (dTxreal + i.*dTximag).*carrier';

s = real(sTxup); %Envia somente a envoltória do sinal.

%% Canal

%Define o parâmetro de escala a partir da equação 3.3.
%Ver: http://math.bu.edu/people/mveillet/html/alphastablepub.html
scale = (sqrt(Eb*(10^(EbNo(a)/10))^(-1)/2);

%Gera o ruído composto a partir da distribuição Salphas com os
%parâmetros pré-definidos.
%A função da distribuição Salphas utilizada pode ser encontrada
%na referência:
%http://www.mathworks.com/matlabcentral/fileexchange/37514-stbl--alpha-stable-distributions-for-matlab
r = stblrnd(alpha,skew,scale,location,length(s),1);

%Soma o ruído composto com o sinal transmitido.
y = s + r;

%% Redução de ruído impulsivo pela técnica COD

if(flag >= 5)
    %Divisão do número de bits enviados em quadros com comprimentos
    %iguais a Cpacote.
    Nframes = length(y)/Cpacote;
    iNframes = floor(Nframes);
    Cresto = round(Cpacote*(Nframes - iNframes));
    iNframes*Cpacote + Cresto;

    %Cria o vetor que receberá o resultado da otimização de todos os
    %quadros.
    yResult = zeros(length(y), 1);

    %Laço da resolução do problema de otimização por quadro.
    for f = 1:iNframes

        %Modelagem do problema de reconstrução de sinais da técnica COD.
        cvx_begin quiet
            variable s_est(Cpacote); %Variável do problema.
            expression func(2); %Vetor de expressões.

```

```

%Objetivo de aproximação.
func(1) = sum(huber((s_est - y(1 + (f-1)*Cpacote:f*Cpacote)), M));
%Objetivo de suavização.
func(2) = sum_square(s_est(2:(Cpacote))-s_est(1:(Cpacote-1)));
%Problema bicritério escalarizado sem restrições.
minimize ((lambda')*func)
cvx_end

%Armazena o resultado da otimização do quadro f no vetor
%yResult.
yResult(1 + (f-1)*Cpacote:f*Cpacote) = s_est;

end
%Armazena o resultado da redução de ruído na variável s_est_total.
s_est_total = yResult;

%Para permitir o funcionamento automático dos levantamentos das
%curvas sem ocasionar nenhum erro, é necessário atribuir
%novamente o valor do sinal estimado a variável y.
y = s_est_total;

end

%% Conversão para banda base (C.p/B)

%Multiplica o sinal yNoisy pelo conjugado da portadora e armazena em
%yTxdn. Esta multiplicação converte o sinal em banda passante
%para seu equivalente em banda base.
s_est_down = y.*conj(carrier');

s_est_downreal = real(s_est_down); %Parte real.
s_est_downimag = imag(s_est_down); %Parte imaginária.

%% Sinal recebido

%Realiza a filtragem no sinal recebido através do filtro raiz
%cosseno elevado casado com o filtro de transmissão.
d_est_real = filter(rrcFilter, s_est_downreal); %Parte real.
d_est_imag = filter(rrcFilter, s_est_downimag); %Parte imaginária.

%Calcula o atraso de grupo do filtro dado por número de bits.
delay = N/(nSamp*2);

%Realiza a amostragem para baixo do sinal recebido, em que cada
%nSamp amostras é convertida em um símbolo.
d_est_real = downsample(d_est_real, nSamp); %Parte real.
d_est_imag = downsample(d_est_imag, nSamp); %Parte imaginária.

%Desconta os atrasos referentes às convoluções realizadas pelos
%filtros de transmissão e de recepção.
d_est_real = d_est_real(2*delay+1:end-2*delay); %Parte real.
d_est_imag = d_est_imag(2*delay+1:end-2*delay); %Parte imaginária.

%Compõe o vetor de símbolos estimado.
d_est = d_est_real + i*d_est_imag;

%% Demodulação

%Demodula o vetor de símbolos estimado no correspondente vetor de
%bits estimados.
b_est = demodulate(hDemod, (-1)*d_est);

%% Atualização dos dados para cálculo da BER

%Compara o vetor de bits estimados com o vetor de bits
%transmitidos. Nota-se que forma descontados os últimos 4*delay
%bits, correspondentes aos bits que ainda não foram estimados
%devido aos atrasos dos filtros de transmissão e de recepção.
[number_of_errors, ber] = biterr(b(1:end-4*delay*k), b_est);

```

```

        %Atualiza a contagem dos erros de bits estimados.
        Nerros_aux = Nerros_aux + number_of_errors;

        %Atualiza a contagem dos laços.
        cont = cont + 1;

        if(number_of_errors > 0)
            Nerros_aux = Nerros_aux
            save('Nerros_aux', 'Nerros_aux');
            save('cont', 'cont');
        end
    end
    %% Cálculo da BER

    %Calcula a BER correspondente a curva flag para o valor de
    %EbN0(a). Este cálculo é realizado dividindo-se o número de
    %estimativas erradas de bits contabilizadas pela variável Nerros_aux
    %pela quantidade de bits enviados, dada por cont*(n - 4*delay).
    BER(flag,a) = Nerros_aux/(cont*(n - 4*delay))

    %Salva a atualização da matriz BER no arquivo backup.mat.
    save('backup', 'BER');
end
end
%% Plota as curvas de desempenho encontradas

close all

%Curva de desempenho teórica para um sistema de comunicação digital BPSK
%coerente em um canal Rayleigh-AWGN.
BER_Teorica = berfading(EbNo, 'psk', 2, 1);
semilogy(EbNo, BER_Teorica, 'color', 'k', 'Linewidth', 2);
axis([0 42 0.0000099 1])

%Curva de desempenho obtida para o sistema adotado sem redução de ruído
%para alpha = 2. Esta é a curva de validação do sistema.
figure(1)
hold on
semilogy(EbNo, BER(1, :), 'LineStyle', '--', 'color', 'k', 'Marker', 'o', ...
    'MarkerEdge', 'k', 'MarkerFaceColor', 'k', 'MarkerSize', 7 );
set(gca, 'xtick', [0 7 14 21 28 35 42])
grid on

%Curva de desempenho obtida para o sistema adotado sem redução de ruído
%para alpha = 1,9.
figure(1)
hold on
semilogy(EbNo, BER(2, :), 'LineStyle', '--', 'color', 'k', 'Marker', 's', ...
    'MarkerEdge', 'k', 'MarkerFaceColor', 'k', 'MarkerSize', 7 );
grid on

%Curva de desempenho obtida para o sistema adotado sem redução de ruído
%para alpha = 1,7.
figure(1)
hold on
semilogy(EbNo, BER(3, :), 'LineStyle', '--', 'color', 'k', 'Marker', '*', ...
    'MarkerEdge', 'k', 'MarkerFaceColor', 'k', 'MarkerSize', 7 );
grid on

%Curva de desempenho obtida para o sistema adotado sem redução de ruído
%para alpha = 1,5.
figure(1)
hold on
semilogy(EbNo, BER(4, :), 'LineStyle', '--', 'color', 'k', 'Marker', 'd', ...
    'MarkerEdge', 'k', 'MarkerFaceColor', 'k', 'MarkerSize', 7 );
grid on

%Curva de desempenho obtida para o sistema adotado com redução de ruído
%realizada pela técnica COD para alpha = 1,9.
figure(1)
hold on

```

```

semilogy(EbNo,BER(5,:), 'LineStyle','--','color','k','Marker','h',...
'MarkerEdge','k','MarkerFaceColor','k','MarkerSize', 7 );
grid on

%Curva de desempenho obtida para o sistema adotado com reduao de ruído
%realizada pela tcnica COD para alpha = 1,7.
figure(1)
hold on
semilogy(EbNo,BER(6,:), 'LineStyle','--','color','k','Marker','p',...
'MarkerEdge','k','MarkerFaceColor','k','MarkerSize', 7 );
grid on

%Curva de desempenho obtida para o sistema adotado com reduao de ruído
%realizada pela tcnica COD para alpha = 1,5.
figure(1)
hold on
semilogy(EbNo,BER(7,:), 'LineStyle','--','color','k','Marker','>',...
'MarkerEdge','k','MarkerFaceColor','k','MarkerSize', 7 );
grid on

%Escrita dos eixos.
xlabel('$E_{\rm b}/N_{\rm 0}$, dB','Interpreter','latex');
ylabel('BER','Interpreter','latex')

%Legenda.
hl = legend('teorica', '$\alpha = 2$', '$\alpha = 1,9$',...
'$\alpha = 1,7$', '$\alpha = 1,5$', 'COD, $\alpha = 1,9$',...
'COD, $\alpha = 1,7$', 'COD, $\alpha = 1,5$' );

set(hl,'Interpreter','latex')
hold on

```

```
%Software utilizado para a obtenção das curvas de desempenho da técnica END
%e END modificada apresentadas na dissertação:
%"Otimização Convexa Aplicada à Redução de Ruído Impulsivo em Sistemas de
%Comunicação Acústica Sob a Água.
```

```
%Desenvolvido por:
%Lucas Silvestre Chaves
```

```
%Data: 31/01/2015
%Revisão: R2
```

```
%Este software foi utilizado para simular o sistema adotado na dissertação
%e obter as curvas de desempenho do sistema adotado sem redução de ruído e
%com redução de ruído para todos cenários de ruído impulsivo
%considerados. As técnicas de redução de ruído utilizadas possíveis de
%serem utilizadas aqui são as técnicas de redução via dispositivo não
%linear, designadas por técnicas END e END adaptada. A simulação está
%configurada para obter e apresentar as curvas de desempenho referentes
%ao sistema adotado sem redução e com redução pela técnica END para
%alpha = 1,9; 1,7 e 1,5. Também é obtida e apresentada a curva de validação
%do modelo, referente ao sistema adotado sem redução para alpha = 2.
```

```
%Para utilizar a técnica END modificada, deve-se comentar dentro do bloco
%"Redução de ruído via Detecção Não linear" o bloco da técnica END e
%descomentar o bloco da técnica END modificada. O fator F dos dispositivos
%não lineares está configurado com 1 e pode ser redefinido no bloco
%localizado no início do programa, chamado de "Parâmetros dos dispositivos
%não lineares". Outros parâmetros possíveis de serem reconfigurados neste
%bloco, são os parâmetros INICIO e FIM. Estes parâmetros têm influência na
%técnica END modificada e restringem quais serão as amostras de um quadro
%ordenado que constituem o cálculo da potência média para a obtenção do
%limiar t.
```

```
%% Inicialização da simulação
```

```
clear all;
close all;
clc;
```

```
%% Definições dos parâmetros da distribuição Salphas
```

```
alpha = 1.5; %Definição do expoente característico.
skew = 0; %Definição do parâmetro de simetria.
scale = 0.01; %Definição do parâmetro de escala.
location = 0; %Definição do parâmetro de localização.
```

```
%% Definição dos parâmetros do sistema adotado.
```

```
Ms = 2; %Número de símbolos da constelação.
k = log2(Ms); %Número de bits por símbolo.
Cpacote = 1500; %Tamanho do quadro.
n = 10*Cpacote; %Número de bits enviados por laço para estimação da BER.
```

```
nSamp = 100; %Número de amostras por símbolo.
```

```
Eb = 1; %Energia por bit.
```

```
Rb = 100; %Taxa de bits enviados.
Rs = Rb/k; %Taxa de símbolos enviados.
Tsimb = 1/Rs; %Tempo por símbolo.
```

```
Fs = nSamp*Rs; %Frequência de amostragem da simulação.
Ts = 1/Fs; %Tempo de amostragem.
```

```
fosc = 200; %Frequência da portadora.
```

```
Nerros = 100; %Critério de parada para a simulação MonteCarlo.
```

```
%% Parâmetros dos dispositivos não lineares
```

```
F = 1; %Fator F dos dispositivos não Lineares.
```

```

%Os parâmetros INICIO e FIM têm influência na técnica END adaptada e
%restringem quais serão as amostras de um quadro ordenado que constituem
%o cálculo da potência média para o posterior cálculo do limiar t
INICIO = 200; %Parâmetro que define a amostra inicial de um quadro ordenado;
FIM = 1300; %%Parâmetro que define a amostra final de um quadro ordenado;

%% Criação do Modulador e do Demodulados

hMod = modem.pskmod(Ms); % Cria um modulador M-PSK.
hMod.InputType = 'Bit'; % Define a entrada como bits.
hMod.SymbolOrder = 'Gray'; % Define um mapeamento Gray para os símbolos.
hDemod = modem.pskdemod(hMod); % Cria o B-PSK com as definições anteriores.

%% Projeto do filtro Raiz Cosseno Elevado

N = 10*nSamp; % Define a ordem do filtro.
rolloff = 1; % Define o fator de Rolloff do filtro.
filtDef = fdesign.pulseshaping(nSamp, 'Square Root Raised Cosine', ...
    'N,Beta', N, rolloff); % Define a resposta ao impulso do filtro.
rrcFilter = design(filtDef); % Cria o filtro.
rrcFilter.Numerator = rrcFilter.Numerator * sqrt(nSamp); % Normaliza.

%% Laço da simulação

EbNo = 0:7:42; %Define o vetor E_b/N_0 para o cálculo da BER.

BER = zeros(7,length(EbNo)); %Define a matriz BER para armazenar as BERs.

for flag = 4:1:7 %Laço de levantamento das curvas de desempenho do sistema.

    if(flag == 1)
        alpha = 2; %Curva de validação.
    end
    if(flag == 2)
        %Desempenho do sistema adotado sem redução de ruído para alpha=1,9.
        alpha = 1.9;
    end
    if(flag == 3)
        %Desempenho do sistema adotado sem redução de ruído para alpha=1,7.
        alpha = 1.7;
    end
    if(flag == 4)
        %Desempenho do sistema adotado sem redução de ruído para alpha=1,5.
        alpha = 1.5;
    end
    if(flag == 5)
        %Desempenho do sistema adotado com redução realizada pela técnica
        %END para alpha=1,9.
        alpha = 1.9;
    end
    if(flag == 6)
        %Desempenho do sistema adotado com redução realizada pela técnica
        %END para alpha=1,7.
        alpha = 1.7;
    end
    if(flag == 7)
        %Desempenho do sistema adotado com redução realizada pela técnica
        %END para alpha=1,5.
        alpha = 1.5;
    end

    for a = 1:1:length(EbNo) %Laço de cada curva.

        Nerros_aux = 0; %Zera a variável de contagem dos erros.
        cont = 0; %Contagem dos laços para contar o número de bits enviados.

        while (Nerros_aux <= Nerros) %Laço do levantamento de cada BER.
            %% Fonte binária
            %Gera uma sequência de bits aleatórios equiprováveis.
            b = randi([0 1],n,1);

```

```

%% Modulação
d = (-1)*modulate(hMod,b); %Modula a sequência dos bits geradas.

%% Canal Rayleigh
%Simula um canal Rayleigh plano e lento
d = raylrnd(sqrt(2)/2,length(d),1).*real(d);

%% Filtragem do sinal
dreal = real(d); %Recebe a parte real de d.
dimag = imag(d); %Recebe a parte imaginária de d.

%Amsotra cada símbolo em nsamp amostras.
dUpreal = upsample(dreal, nSamp); %Parte real.
dUpimag = upsample(dimag, nSamp); %Parte imaginária.

%Filtra os símbolos amostrados pelo filtro raiz cosseno elevado.
%Não foi necessário a utilização da equalização Sinc Inversa (SI)
%com foi apresentado no sistema adotado,
%pois o sinal a ser filtrado aqui é um impulso e uma sequência
%de zeros. Desta forma, o resultado da convolução deste filtro
%com o trem de impulsos espaçoes por zeros é o mesmo que a
%convolução de um filtro raiz cosseno elevado com SI realizado
%com um trem de pulsos.
dTxreal = filter(rrcFilter, dUpreal); %Parte real.
dTximag = filter(rrcFilter, dUpimag); %Parte imaginária.

%% Conversão para banda passante (C.p/C.)
%Define a base de tempo para as funções cossenoidais.
t = [0: 1/Fs: (length(dTxreal) - 1)*1/Fs];

%Gera a portadora. O fator sqrt(2) garante que a potência da
%frequência do sinal em banda passante seja igual a potência
%da frequência duplicada do sinal em banda base.
%Ver: Passband Modulation with Adjacent Channel Interference.
carrier = sqrt(2)*exp(1i*2*pi*fosc*t);

%Multiplica o sinal a ser transmitido pela portador.
sTxup = (dTxreal + i.*dTximag).*carrier';

s = real(sTxup); %Envia somente a envoltória do sinal.

%% Canal

%Define o parâmetro de escala a partir da equação 3.3.
%Ver: http://math.bu.edu/people/mveillet/html/alphastablepub.html
scale = (sqrt(Eb*(10^(EbNo(a)/10))^(-1)/2);

%Gera o ruído composto a partir da distribuição SalphaS com os
%parâmetros pré-definidos.
%A função da distribuição SalphaS utilizada pode ser encontrada
%na referência:
%http://www.mathworks.com/matlabcentral/fileexchange/37514-stbl--alpha-stable-distributions-for-matlab
r = stblrnd(alpha,skew,scale,location,length(s),1);

%Soma o ruído composto com o sinal transmitido.
y = s + r;

%% Conversão para banda base (C.p/B)

%Multiplica o sinal yNoisy pelo conjugado da portadora e armazena em
%yTxdown. Esta multiplicação converte o sinal em banda passante
%para seu equivalente em banda base.
s_est_down = y.*conj(carrier');

s_est_downreal = real(s_est_down); %Parte real.
s_est_downimag = imag(s_est_down); %Parte imaginária.

% Redução de ruído via Detecção Não linear
if(flag >= 5)

```

```

%Divisão do número de bits enviados em quadros com comprimentos
%iguais a Cpacote.
Nframes = length(s_est_downreal)/Cpacote;
iNframes = floor(Nframes);
Cresto = round(Cpacote*(Nframes - iNframes));
iNframes*Cpacote + Cresto;

%Cria o vetor que receberá o resultado da redução de ruído
%impulsivo via detecção não linear de todos os quadros.
yResult_real = zeros(length(s_est_downreal), 1);
yResult_imag = zeros(length(s_est_downimag), 1);

for f = 1:1:iNframes
%% END
%Armazena o sinal recebido em variáveis auxiliares.
yTxdownreal_aux = s_est_downreal(1 + (f-1)*Cpacote:f*Cpacote);
yTxdownimag_aux = s_est_downimag(1 + (f-1)*Cpacote:f*Cpacote);

%Calcula o limiar t para o fator F(flag-1).
t_real = F*sqrt(sum(yTxdownreal_aux.^2)/length(yTxdownreal_aux));
t_imag = F*sqrt(sum(yTxdownimag_aux.^2)/length(yTxdownimag_aux));

%%
%% %% END modificada
%%
%% %Ordena as amostras do sinal recebido em ordem crescente e
%% %armazena em variáveis auxiliares.
yTxdownreal_aux = sort((s_est_downreal(1 + (f-1)*Cpacote:f*Cpacote)));
yTxdownimag_aux = sort((s_est_downimag(1 + (f-1)*Cpacote:f*Cpacote)));

%%
%% %Calcula o limiar t para o fator F(flag-1) com o cálculo da
%% %potência
%% %realizado para as amostras centrais do vetor ordenado, da
%% %posição INICIO a posição FIM.
t_real = F*sqrt(sum(yTxdownreal_aux(INICIO:FIM).^2)/
length(yTxdownreal_aux(INICIO:FIM)));
t_imag = F*sqrt(sum(yTxdownimag_aux(INICIO:FIM).^2)/
length(yTxdownimag_aux(INICIO:FIM)));

%%
%% %Recarrega o vetor do sinal recebido nas variáveis auxiliares
%% %para a realização da detecção não linear com base no limiar t
%% %calculado.
yTxdownreal_aux = s_est_downreal(1 + (f-1)*Cpacote:f*Cpacote);
yTxdownimag_aux = s_est_downimag(1 + (f-1)*Cpacote:f*Cpacote);

%%
%% %% Processamento da Técnica de Detecção Não Linear
%% %Este processamento baseia-se na função de transferência não
%% %linear dada pela equação (4.1).
for cont_ceif = 1:1:length(yTxdownreal_aux)
    if(abs(yTxdownreal_aux(cont_ceif)) < t_real)
        yTxdownreal_aux(cont_ceif) = yTxdownreal_aux(cont_ceif);
    else
        if(yTxdownreal_aux(cont_ceif) < 0)
            yTxdownreal_aux(cont_ceif) = (-1)*t_real;
        else
            yTxdownreal_aux(cont_ceif) = t_real;
        end
    end

    if(abs(yTxdownimag_aux(cont_ceif)) < t_imag)
        yTxdownimag_aux(cont_ceif) = yTxdownimag_aux(cont_ceif);
    else
        if(yTxdownimag_aux(cont_ceif) < 0)
            yTxdownimag_aux(cont_ceif) = (-1)*t_imag;
        else
            yTxdownimag_aux(cont_ceif) = t_imag;
        end
    end
end
end
end
end

```

```

        %Armazena o resultado da redução de ruído do quadro f.
        yResult_real(1 + (f-1)*Cpacote:f*Cpacote) = yTxdownreal_aux;
        yResult_imag(1 + (f-1)*Cpacote:f*Cpacote) = yTxdownimag_aux;

    end

    %Armazena o resultado da redução de ruído impulsivo de todos os
    %quadros.
    s_est_downreal = yResult_real;
    s_est_downimag = yResult_imag;
end

%% Sinal recebido

%Realiza a filtragem no sinal recebido através do filtro raiz
%cosseno elevado casado com o filtro de transmissão.
d_est_real = filter(rrcFilter, s_est_downreal); %Parte real.
d_est_imag = filter(rrcFilter, s_est_downimag); %Parte imaginária.

%Calcula o atraso de grupo do filtro dado por número de bits.
delay = N/(nSamp*2);

%Realiza a amostragem para baixo do sinal recebido, em que cada
%nSamp amostras é convertida em um símbolo.
d_est_real = downsample(d_est_real, nSamp); %Parte real.
d_est_imag = downsample(d_est_imag, nSamp); %Parte imaginária.

%Desconta os atrasos referentes às convoluções realizadas pelos
%filtros de transmissão e de recepção.
d_est_real = d_est_real(2*delay+1:end-2*delay); %Parte real.
d_est_imag = d_est_imag(2*delay+1:end-2*delay); %Parte imaginária.

%Compõe o vetor de símbolos estimado.
d_est = d_est_real + i*d_est_imag;

%% Demodulação

%Demodula o vetor de símbolos estimado no correspondente vetor de
%bits estimados.
b_est = demodulate(hDemod, (-1)*d_est);

%% Atualização dos dados para cálculo da BER
%Compara o vetor de bits estimados com o vetor de bits
%transmitidos. Nota-se que forma descontados os últimos 4*delay
%bits, correspondentes aos bits que ainda não foram estimados
%devido aos atrasos dos filtros de transmissão e de recepção.
[number_of_errors, ber] = biterr(b(1:end-4*delay*k),b_est);

%Atualiza a contagem dos erros de bits estimados.
Nerros_aux = Nerros_aux + number_of_errors;

%Atualiza a contagem dos laços.
cont = cont + 1;

if(number_of_errors > 0)
    Nerros_aux = Nerros_aux
    save('Nerros_aux', 'Nerros_aux');
    save('cont', 'cont');
end
end
%% Cálculo da BER

%Calcula a BER correspondente a curva flag para o valor de
%EbN0(a). Este cálculo é realizado dividindo-se o número de
%estimativas erradas de bits contabilizadas pela variável Nerros_aux
%pela quantidade de bits enviados, dada por cont*(n - 4*delay).
BER(flag,a) = Nerros_aux/(cont*(n - 4*delay))

%Salva a atualização da matriz BER no arquivo backup.mat.
save('backup', 'BER');
```

```

    end
end
%% Plota as curvas de desempenho encontradas
close all

%Curva de desempenho teórica para um sistema de comunicação digital BPSK
%coerente em um canal Rayleigh-AWGN.
BER_Teorica = berfading(EbNo,'psk',2,1);
semilogy(EbNo, BER_Teorica, 'color','k', 'Linewidth',2);
axis([0 42 0.0000099 1])

%Curva de desempenho obtida para o sistema adotado sem redução de ruído
%para alpha = 2. Esta é a curva de validação do sistema.
figure(1)
hold on
semilogy(EbNo,BER(1,:), 'LineStyle','--', 'color','k', 'Marker','o',...
    'MarkerEdge','k', 'MarkerFaceColor','k', 'MarkerSize', 7 );
set(gca, 'xtick',[0 7 14 21 28 35 42])
grid on

%Curva de desempenho obtida para o sistema adotado sem redução de ruído
%para alpha = 1,9.
figure(1)
hold on
semilogy(EbNo,BER(2,:), 'LineStyle','--', 'color','k', 'Marker','s',...
    'MarkerEdge','k', 'MarkerFaceColor','k', 'MarkerSize', 7 );
grid on

%Curva de desempenho obtida para o sistema adotado sem redução de ruído
%para alpha = 1,7.
figure(1)
hold on
semilogy(EbNo,BER(3,:), 'LineStyle','--', 'color','k', 'Marker','*',...
    'MarkerEdge','k', 'MarkerFaceColor','k', 'MarkerSize', 7 );
grid on

%Curva de desempenho obtida para o sistema adotado sem redução de ruído
%para alpha = 1,5.
figure(1)
hold on
semilogy(EbNo,BER(4,:), 'LineStyle','--', 'color','k', 'Marker','d',...
    'MarkerEdge','k', 'MarkerFaceColor','k', 'MarkerSize', 7 );
grid on

%Curva de desempenho obtida para o sistema adotado com redução de ruído
%realizada pela técnica END para alpha = 1,9.
figure(1)
hold on
semilogy(EbNo,BER(5,:), 'LineStyle','--', 'color','k', 'Marker','h',...
    'MarkerEdge','k', 'MarkerFaceColor','k', 'MarkerSize', 7 );
grid on

%Curva de desempenho obtida para o sistema adotado com redução de ruído
%realizada pela técnica END para alpha = 1,7.
figure(1)
hold on
semilogy(EbNo,BER(6,:), 'LineStyle','--', 'color','k', 'Marker','p',...
    'MarkerEdge','k', 'MarkerFaceColor','k', 'MarkerSize', 7 );
grid on

%Curva de desempenho obtida para o sistema adotado com redução de ruído
%realizada pela técnica END para alpha = 1,5.
figure(1)
hold on
semilogy(EbNo,BER(7,:), 'LineStyle','--', 'color','k', 'Marker','>',...
    'MarkerEdge','k', 'MarkerFaceColor','k', 'MarkerSize', 7 );
grid on

%Escrita dos eixos.
xlabel('$E_{\rm b}/N_{\rm 0}$, dB', 'Interpreter','latex');
ylabel('BER', 'Interpreter','latex')

```

```
%Legenda.  
h1 = legend('teorica', '$\alpha = 2$', '$\alpha = 1,9$',...  
           '$\alpha = 1,7$', '$\alpha = 1,5$', 'END, $\alpha = 1,9$',...  
           'END, $\alpha = 1,7$', 'END, $\alpha = 1,5$' );  
  
set(h1,'Interpreter','latex')  
hold on
```

