

Inatel

Instituto Nacional de Telecomunicações

Análise de Mobilidade e um
Autoencoder Robusto

PEDRO MARCIO RAPOSO PEREIRA

07 / 2022



ANÁLISE DE MOBILIDADE E UM AUTOENCODER ROBUSTO

PEDRO MARCIO RAPOSO PEREIRA

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

ORIENTADOR: Prof. Dr. Rausley Adriano Amaral de Souza.

Pereira, Pedro Marcio Raposo

P436a

Análise de Mobilidade e Autoencoder Robusto. / Pedro Marcio Raposo Pereira – Santa Rita do Sapucaí, 2022.
96 p.

Orientador: Prof. Dr. Rausley Adriano Amaral de Souza
Dissertação de Mestrado em Telecomunicações – Instituto Nacional de Telecomunicações – INATEL.

Inclui bibliografia e anexo.

1. Mobilidade 2. Desvanecimento 3. Sombreamento 4. IA 5. Autoencoder 6. Mestrado em Telecomunicações. I. Souza, Rausley Adriano Amaral de II. Instituto Nacional de Telecomunicações – INATEL. III. Título.

CDU 621.39

FOLHA DE APROVAÇÃO

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

Prof. Dr. Rausley Adriano Amaral de Souza
INATEL

Prof. Dr. Roberto César Dias Vilela Bonfin
TECHNISCHE UNIVERSITÄT DRESDEN – TUD

Pro. Dr. Felipe Augusto Pereira de Figueiredo
INATEL

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

“A sabedoria começa na reflexão.”

Sócrates

*Aos meus pais,
pelos ensinamentos, compreensão e amor.*

Agradecimentos

Agradeço, primeiramente, a minha família.

Agradeço aos meus amigos pela companhia.

Agradeço, também, aos meus professores, em todas as etapas de minha formação, por suas contribuições imensuráveis e a gentileza de compartilhar seus conhecimentos.

Agradeço ao Instituto Nacional de Telecomunicações pela formação ímpar.

Este trabalho foi parcialmente financiado pelo CNPq (contrato: 311470/2021-1), pela FAPEMIG (contrato: APQ 00966-17), pela FAPESP (contrato: 2021/06946-0) e pela RNP, com recursos do MCTIC, processo No 01245.010604/2020-14, sob o projeto Sistemas de Comunicações Móveis de 6ª Geração (6G) do Centro de Referência em Radiocomunicações (CRR) do Instituto Nacional de Telecomunicações – Inatel, Brasil.

Pedro Marcio Raposo Pereira

Sumário

Sumário	x
Lista de Figuras	xi
Lista de Tabelas	xiii
Lista de Abreviaturas e Siglas	xv
Lista de Símbolos	xvii
Resumo	xxi
Abstract	xxiii
1 Introdução	1
2 Fundamentos de Modelo de Canal	5
2.1 Desvanecimento de Curto Prazo	6
2.2 Sombreamento	10
2.3 Mobilidade	11
3 Machine Learning e Autoencoders	15
3.1 Fundamentos	15
3.2 Autoencoder	18
3.3 Autoencoder como mecanismo de comunicação	19
3.3.1 Modelo de Comunicação	19
3.3.2 Implementação	20
3.3.3 Desempenho em Canal AWGN	22
4 Contribuições e Resultados	25
4.1 Modelo de Desvanecimento α - μ /Gama	25
4.2 Mobilidade Incorporada ao Desvanecimento	27
4.3 Métricas de Desempenho	29
4.3.1 Probabilidade de erro de símbolo média	29
4.3.2 Probabilidade de Indisponibilidade	31
4.3.3 Capacidade	32
4.4 Resultados	33

4.5	Autoencoder em Canal com Desvanecimento e Mobilidade	38
5	Conclusões e Propostas Para Novas Pesquisas	41
A	Função H de Fox	43
A.1	Função G de Meijer: Caso Especial	45
B	Algoritmo para cálculo da função H de Fox	47
	Referências Bibliográficas	49
I	Implementação, treinamento e validação do Autoencoder	1
II	Processo de geração de amostras para o modelo de deslocamento <i>random waypoint</i>	5
III	Algoritmos para geração e validação dos resultados e métricas de desempenho	8
III.1	Função densidade de probabilidade	8
III.2	Função de distribuição acumulada	9
III.3	Probabilidade de erro de símbolo média e assintótica	10
III.4	Probabilidade de indisponibilidade	12
III.5	Probabilidade de indisponibilidade com sinal interferente	13
III.6	Capacidade	14

Lista de Figuras

2.1	Cenário de propagação por múltiplos percursos.	6
2.2	FDP da envoltória e potência da distribuição α - μ e seus casos particulares.	9
2.3	Comparação da FDP da distribuição Gama com a equivalente distribuição lognormal.. . . .	10
2.4	FDP da distribuição espacial para o modelo RWP unificada para as topologias uni, bi e tridimensional.	12
3.1	Arquitetura de uma rede neural densa.	16
3.2	Arquitetura de um autoencoder.	19
3.3	Diagrama em blocos de um sistema de comunicação.	19
3.4	Diagrama da implementação do autoencoder.	20
3.5	Custo por iteração e custo médio por época para o Autoencoder.	23
3.6	Disposição dos símbolos aprendida pelo Autoencoder.	23
3.7	Probabilidade de erro de símbolo média comparada entre um Autoencoder e 4-QAM.	24
4.1	PDF e histograma da potência de um sinal sob desvanecimento α - μ /Gama com deslocamento aleatório RWP.	33
4.2	Comportamento da FDP da potência de um sinal sob desvanecimento α - μ /Gama com deslocamento aleatório RWP.	34
4.3	Comportamento da CDF da potência de um sinal sob desvanecimento α - μ /Gama com deslocamento aleatório RWP mediante variação das dimensões.	35
4.4	ASEP em função da potência de transmissão para diferentes parâmetros de desvanecimento.	36
4.5	ASEP para análise dos ganhos de diversidade e codificação.	36
4.6	PI para cenários com e sem canal interferente.	37
4.7	Capacidade média em função da potência média na máxima distância de cobertura.	38
4.8	Análise da Probabilidade de Erro de bloco do Autoencoder para diferentes configurações.	39
4.9	Análise da capacidade de generalização do Autoencoder para diferentes condições de canal.	40

Lista de Tabelas

2.1	Coeficientes polinomiais	12
3.1	Funções de ativação	17
3.2	Funções custo	17
3.3	Configuração das DNNs	21
4.1	Parâmetros c_1 e c_2 para diversos esquemas de modulação	29

Lista de Abreviaturas e Siglas

5G	Quinta geração de redes móveis
6G	Sexta geração de redes móveis
Adam	Estimação de momento adaptativa (<i>adaptive moment estimation</i>)
AE	Autoencoder
API	Interface de programação de aplicação (<i>application programming interface</i>)
ASEP	Probabilidade de erro de símbolo média (<i>average symbol error probability</i>)
AWGN	Ruído aditivo Gaussiano branco (<i>additive white Gaussian noise</i>)
BFSK	Modulação binária por deslocamento de frequência (<i>binary frequency shift keying</i>)
bps	Bits por segundo
BPSK	Modulação binária por deslocamento de fase (<i>binary phase shift keying</i>)
FDC	Função de distribuição cumulativa
CNN	Rede neural convolucional (<i>convolutional neural network</i>)
dB	Decibel
dBW	Decibel-Watt
DL	Aprendizado profundo (<i>deep learning</i>)
DNN	Rede neural densa (<i>dense neural network</i>)
FNN	<i>Feedforward Neural Network</i>
GAN	Rede adversarial generativa (<i>generative adversarial network</i>)
GDE	Gradiente descendente estocástico
GM	Gauss Markov
AI	Inteligência artificial (<i>artificial intelligence</i>)
i.i.d	Independente e identicamente distribuído
LOS	Linha de visada (<i>line-of-sight</i>)
LUT	Tabela de consulta (<i>lookup table</i>)
M-FSK	Modulação M -ária por deslocamento de frequência (<i>M-ary Frequency Shift Keying</i>)
ML	Aprendizado da máquina (<i>machine learning</i>)
MLP	<i>Multi layer perceptrons</i>
M-PSK	Modulação M -ária por deslocamento de fase (<i>M-ary phase shift keying</i>)
M-QAM	Modulação M -ária por amplitude em quadratura (<i>M-ary quadrature amplitude modulation</i>)
MSE	Erro médio quadrático (<i>mean square error</i>)
NN	Rede neural (<i>neural network</i>)

FDP	Função densidade de probabilidade
PI	Probabilidade de indisponibilidade
RD	Direção aleatória (<i>random direction</i>)
relu	<i>Rectified linear unit</i>
RSI	Relação sinal-interferência
RSR	Relação sinal-ruído
RSRI	Relação sinal-ruído-interferência
RW	Caminhada aleatória (<i>random walk</i>)
RWP	Modelo de paradas aleatórias (<i>random waypoint model</i>)
SDR	Rádio definido por software (<i>software-defined radio</i>)
THz	Terahertz
VA	Variável aleatória

Lista de Símbolos

i	Constante inteira
$f_A(a)$	Função densidade de probabilidade da variável aleatória A
\mathbb{C}	Conjunto dos números complexos
$P[a b]$	Probabilidade de a condicionada à b
$f_A(A B)$	Função densidade de probabilidade da variável aleatória A condicionada a B
$F_A(a)$	Função de distribuição cumulativa da variável aleatória A
$\text{erfc}(\cdot)$	Função erro complementar
$\exp(\cdot)$	Exponencial natural
$\Gamma(\cdot)$	Função Gama completa
$\mathbb{E}[\cdot]$	Operador esperança matemática
$\mathbb{V}[\cdot]$	Operador variância
\ln	Logaritmo natural
$\psi(\cdot)$	Função digama
$\psi'(\cdot)$	Função trigama
\mathbb{R}	Conjunto dos números reais
P_{Tx}	Potência do sinal transmitido
N_p	Número total de múltiplos percursos
R_{I_i}	Componente <i>em fase</i> da i -ésima componente do sinal
R_{Q_i}	Componente <i>quadratura</i> da i -ésima componente do sinal
σ^2	Variância
R	Envoltória do sinal
\hat{w}	Potência média de desvanecimento
W	Potência de desvanecimento
m_f	Fator de desvanecimento Nakagami
\hat{r}	Raiz α -ésima do valor médio da envoltória
μ	Parâmetro de desvanecimento
P_{Rx}	Potência do sinal recebido
α	Fator de não linearidade
Y	Potência do sombreamento
Ω_{dBW}	Potência média do sombreamento em decibel-Watt
σ_{dBW}	Desvio padrão do sombreamento em decibel-Watt
C_o	Constante numérica de valor $10/\ln(10)$
m_s	Parâmetro de sombreamento
Ω	Potência média de sombreamento
D	Distância aleatória

D_0	Máxima distância de deslocamento
u	Número de coeficientes polinomiais
G_e	Produto dos ganhos lineares das antenas
B_i	i -ésimo coeficiente polinomial
β_i	i -ésimo expoente polinomial
λ	Comprimento do sinal transmitido
d	Distância entre transmissor e receptor
f_0	Frequência de transmissão do sinal
$s_r(t)$	Descrição temporal do sinal de recepção
a_i	Ganho do canal para i -ésima componente do sinal
ϕ_i	Rotação de fase do canal para i -ésima componente do sinal
$\varphi_l(\cdot)$	Transformação realizada pela l -ésima camada densa
\mathbf{m}_i	i -ésima amostra de treinamento
τ_i	i -ésimo rótulo de treinamento
S	Número de dados de treinamento
$J(\cdot)$	Custo médio de treinamento
$j(\cdot)$	Custo por amostra de treinamento
η	Passo de aprendizagem
∇	Operador gradiente
\tilde{J}	Custo por subconjunto de treinamento
ι	Iteração de treinamento
S_b	Subconjunto de dados de treinamento
κ_l	Vetor de entrada de uma l -ésima camada densa
S_b	Número de dados do subconjunto de treinamento
L	Número de camadas de uma rede neural densa
$\hat{\tau}_i$	i -ésima resposta estimada por uma rede neural densa
∂	Operador derivada parcial
\mathbb{M}	Conjunto de M mensagens
m	Mensagem a ser transmitida
h	Código de um autoencoder
n	Utilização do canal de comunicação
R_b	Taxa de bits por ocupação do canal
$f(\cdot)$	Função de codificação
ξ_l	Parâmetros da l -ésima camada densa
$g(\cdot)$	Função de decodificação
\mathbf{X}	Mensagem codificada
\mathbf{Y}	Mensagem filtrada pelo canal
\hat{m}	Mensagem estimada
E	Energia média do sinal transmitido
\mathcal{P}_s	Probabilidade de erro de símbolo/bloco
\mathbf{p}	Vetor de probabilidades
N_0	Densidade espectral de potência de ruído
Z	Ruído aditivo Gaussiano branco
Q_l	Dimensão da l -ésima camada densa
ζ_l	Operação linear da l -ésima camada densa
Ψ_l	Matriz de pesos da l -ésima camada densa

ϵ_l	Vetor de <i>bias</i> da l -ésima camada densa
$\rho(\cdot)$	Função de ativação de uma camada densa
\mathcal{S}	Conjunto de dados de treinamento
t	Variável auxiliar
c_2	Parâmetro de modulação
$\tilde{f}_W(w)$	Função densidade de probabilidade do desvanecimento aproximada
\mathcal{O}_i	Termos da função densidade de probabilidade do desvanecimento aproximada
$\tilde{\mathcal{P}}_s$	Probabilidade de erro de símbolo média aproximada
\mathcal{M}_i	Termos da probabilidade de erro de símbolo média aproximada
\mathcal{G}_c	Ganho de codificação
\mathcal{G}_d	Ganho de diversidade
P_{OP}	Probabilidade de indisponibilidade
θ	Limiar de indisponibilidade
P	Potência de sinal interferente
\mathcal{C}	Termo constante em relação à integração
Ω_{Is}	Potência média por canal interferente
m_{If}	Fator de desvanecimento do interferente
m_{Is}	Parâmetro de sombreamento do interferente
N	Número de canais interferentes
Υ	Relação sinal-interferência
$\hat{\lambda}$	Relação sinal-interferência média na máxima distância de cobertura
C	Capacidade
$Z_p^v(\cdot)$	Função de Krätzel
$H_{p,q}^{m,n}(\cdot)$	Função H de Fox
δ	Expoente de perda de percurso
U	Variável auxiliar de integração
Ω_0	Potência do sinal recebido na máxima distância de cobertura
$G_{p,q}^{m,n}(\cdot)$	Função G de Meijer
c_1	Parâmetro de modulação

Resumo

Pereira, P.M. Análise de mobilidade e um Autoencoder Robusto [dissertação de mestrado]. Santa Rita do Sapucaí: Instituto Nacional de Telecomunicações; 2022.

A modelagem estatística de canais desempenha um papel importante no desenvolvimento de redes de comunicações, com o advento da quinta geração de redes móveis (5G) e sexta geração de redes móveis (6G), já que se espera redes cada vez mais diversificadas quanto aos dispositivos conectados e com maior necessidade de recursos e eficiência. Um paradigma promissor para redes modernas é a inteligência artificial (*artificial intelligence*, AI), com o papel de otimização, integração e gerência em vários níveis.

Este trabalho procura avaliar um modelo generalista de desvanecimento α - μ afetado por um sombreamento Gama em um cenário de mobilidade do tipo modelo de paradas aleatórias (*random waypoint model*, RWP) para diferentes ambientes de propagação e topologias. Obtiveram-se novas expressões para função densidade de probabilidade (FDP), função de distribuição cumulativa (FDC), probabilidade de erro de símbolo média (*average symbol error probability*, ASEP), probabilidade de indisponibilidade (PI) e capacidade. Também, verificou-se a aplicação de um sistema de comunicação, baseado em rede neural densa (*dense neural network*, DNN), na forma de um autoencoder (AE) no canal proposto. Com o conhecimento apenas das amostras do canal, o AE obteve desempenho similar às modulações tradicionais e se mostrou robusto para variações no canal.

Palavras-Chave: 5G, 6G, mobilidade, sombreamento, IA, autoencoder, desvanecimento α - μ .

Abstract

Pereira, P.M. Análise de mobilidade e um Autoencoder Robusto [dissertação de mestrado]. Santa Rita do Sapucaí: Instituto Nacional de Telecomunicações; 2022.

Statistical channel modeling plays an important role in the development of communication networks. With the advent of 5th generation of mobile networks (5G) and 6th generation of mobile networks (6G), it is necessary to use generalist models, since networks are expected to be increasingly diversified in terms of connected devices and with greater need for resources and efficiency. A promising paradigm for modern networks is artificial intelligence (AI), with the role of optimization, integration and management at various levels.

This work seeks to evaluate a general α - μ fading model affected by Gamma shadowing in a random waypoint model (RWP) mobility scenario for different propagation environments and physical network topologies. New expressions were obtained for probability density function (PDF), cumulative distribution function (CDF), average symbol error probability (ASEP), outage probability (OP) and capacity. Then, the application of a communication system based on dense neural network (DNN) as an autoencoder (AE) in the proposed channel is investigated. With only the knowledge of the channel samples, the AE obtained a performance similar to traditional modulations and proved to be robust for channel variations.

Keywords: 5G, 6G, mobility, shadowing, IA, autoencoder, α - μ fading.

Capítulo 1

Introdução

AO projetar um sistema de comunicação digital, a fonte de informação e o receptor (usuário final) são geralmente especificados de acordo com os parâmetros de desempenho para um canal de comunicação modelado por distribuições probabilísticas, uma vez que não se tem controle sobre o mesmo. Fundamentalmente, a tarefa de um sistema de comunicação é guiar a informação da origem ao destino, isso feito seguindo os procedimentos de I) geração de uma mensagem, II) descrição desta mensagem com um certo grau de precisão usando um conjunto de símbolos, III) codificação desses símbolos em uma forma adequada para o meio físico, IV) transmissão dos símbolos codificados para o destino, V) decodificação e reprodução dos símbolos originais e VI) restauração da informação original com uma degradação aceitável, intrínseca ao sistema. A grande dificuldade se encontra em codificar ou modular o sinal de mensagem produzido pela fonte de informação, transmiti-lo pelo canal e gerar uma estimativa na saída do receptor que satisfaça os requisitos do usuário final e tudo isso com uma complexidade aceitável [1].

Para a sexta geração de redes móveis (6G) espera-se que o sistema tenha uma taxa de transferência de até 1×10^{12} bits por segundo (bps) e, para o usuário, uma taxa superior a 1×10^9 bps. Quanto à mobilidade, estima-se uma velocidade máxima suportada de 1000 km/h, além de melhorias como latências e eficiências espectral e energética, em relação à quinta geração de redes móveis (5G). A inteligência artificial (*artificial intelligence*, AI) é frequentemente considerada a mais promissora das tecnologias diretoras da 6G, não só porque desempenhará um papel fundamental no *design* e na otimização da infraestrutura de rede, mas também, o sistema 6G será projetado para suportar um aumento no fluxo de dados originados de diversas aplicações móveis de AI [2].

Muitas tarefas de otimização se tornaram intratáveis à medida que as redes móveis

ficaram mais complexas e diversificadas, apresentando uma oportunidade para abordagens avançadas de aprendizado da máquina (*machine learning*, ML). Técnicas de ML estão sendo exploradas como método orientado a dados para fornecer inteligências de rádio e rede computacional, como previsão de qualidade de serviço, análise de tráfego, previsão de canais sob desvanecimento, classificação de tráfego, etc. O aprendizado profundo (*deep learning*, DL), como um sub-ramo de ML, oferece uma gama de aplicações em comunicações e redes sem-fio, como transmissão adaptativa, previsão e estimação mais precisas em canais sob desvanecimento [2, 3].

Quando o desvanecimento é dito como de curto prazo, observam-se rápidas variações da intensidade do sinal. Tradicionalmente, esse fenômeno é modelado por distribuições tais como Rayleigh [4], Rice [5] e Nakagami- m [6]. Para contemplar canais em que não admitem o campo de espalhamento como sendo homogêneo, outros modelos foram propostos como, por exemplo, as distribuições κ - μ [7], η - μ [7] e α - μ [8]. A variedade de aplicações práticas e a versatilidade desta distribuição generalizada, resultam em grande aceitabilidade da comunidade científica. Fato corroborado pelo estudo em [9], em que é demonstrado que distribuições convencionais resultam em baixa acurácia, enquanto a distribuição mais generalista α - μ se ajusta, quase, perfeitamente a um conjunto de medidas experimentais na faixa de terahertz (THz), possível banda a ser utilizada em sistemas da 6G [2].

Normalmente, os modelos de desvanecimento de curto prazo admitem que a potência média do sinal de recepção seja constante. Entretanto, a presença de obstáculos no meio de transmissão pode fazer com que a potência média seja aleatória. Esse fenômeno é nomeado sombreamento (ou também, desvanecimento de longo prazo). A ocorrência dos fenômenos de curto e longo prazo é conjunta, por isso são modelados por distribuições tais quais Rayleigh-Lognormal [10], Suzuki [11], Nakagami-Lognormal [12], e Nakagami-Gama [13, 14]. A potência média também é afetada pelo movimento relativo entre transmissor e receptor em uma rede móvel. A mobilidade do usuário resulta em variações aleatórias da potência. Existem várias técnicas que modelam o deslocamento dos nós e permitem que se avalie seu impacto sobre o sistema. Dentre elas, o modelo de paradas aleatórias (*random waypoint model*, RWP), sendo comumente usado em redes *Ad Hoc*.

O deslocamento aleatório RWP em canais com desvanecimento foi investigado para os casos Rayleigh, Rice, Nakagami- m , α - μ , η - μ e κ - μ [15–23]. A função densidade de probabilidade (FDP) e função de distribuição cumulativa (FDC) da potência recebida, para os respectivos casos, foram deduzidas em [15–23]. A probabilidade de erro de símbolo média (*average symbol error probability*, ASEP) e a probabilidade de indisponibilidade (PI) foram adotadas como métricas de desempenho do sistema em [15–23].

A PI considerando múltiplas fontes de interferência foi explorada em [17, 22, 24]. A capacidade média do canal é averiguada em [18, 25]. Em [19] foi analisado o desempenho quando há o emprego de diversidade. Os casos com sombreado agregado foram investigados em [18] para as distribuições Nakagami- m e κ - μ .

O estudo estatístico do desvanecimento ocupa um papel crucial no projeto e na análise de desempenho de redes sem-fio, porém, é parte de um problema muito maior. Convencionalmente, entende-se a divisão do sistema de comunicação em blocos, como codificação e decodificação, modulação e demodulação, estimação de canal, sincronismo, equalização etc. Por sua vez, esses são arquitetados, matematicamente, de maneira individual. Em cenários práticos, frequentemente, surgem efeitos que não foram pensados e, também, a otimização é feita para cada bloco de processamento, mas não se pode garantir a otimização global. Assim, o sistema ótimo desejado varia com fatores do meio e aplicação [26].

O conceito de um sistema de comunicação fim-a-fim interpretado como um autoencoder (AE) foi inicialmente proposto em [27, 28]. Um AE descreve uma rede neural densa (*dense neural network*, DNN) treinada para reconstruir as informações da entrada na saída. Técnicas de DL podem ser completamente orientadas a dados, abrindo mão de tratamentos matemáticos. Por outro lado, os conhecimentos já adquiridos podem ser combinados com DL para criar sistemas de comunicação, orientados a modelos baseados em DL, que possam ser mais robustos, eficientes e com menor tempo de execução [26].

Em seu marco inicial, os autores de [28] demonstraram a possibilidade de se treinar um sistema de comunicação fim-a-fim com desempenho comparado às técnicas tradicionais. Também verificou-se a possibilidade da utilização de um AE em um sistema com múltiplos usuários competindo pelo mesmo canal. Outro importante passo foi dado em [29], em que foi feita a implementação de um AE em rádio definido por software (*software-defined radio*, SDR), novamente, com desempenho competitivo. A alternativa de se utilizar um AE modelado por uma rede neural convolucional (*convolutional neural network*, CNN) foi averiguada em [30] e também foi comparado o desempenho em canal sob desvanecimento Rayleigh, em [31] aferiu-se para o modelo α - μ . Em [32, 33] foi proposta a utilização de uma rede neural (*neural network*, NN) para estimar a informação mútua [34] entre as amostras de entrada e saída do canal, deste modo não há necessidade de conhecimento do meio de comunicação. Outro método agnóstico quanto ao canal se dá pelo uso de redes adversariais generativas (*generative adversarial networks*, GANs) em [35, 36].

O presente trabalho tem por objetivo o estudo de um canal com desvanecimento

α - μ afetado por sombreamento e mobilidade simultaneamente. Os resultados apresentados dialogam com os trabalhos [17, 20], ao se utilizar o deslocamento RWP. São deduzidas expressões para FDP, FDC, ASEP, PI e capacidade. Mais ainda, investiga-se o comportamento de um AE, nos moldes propostos por [28], no modelo de canal proposto.

O subsequente da dissertação é dividido da seguinte maneira: No Capítulo 2 são apresentados os conceitos principais que fomentam o presente trabalho, sendo eles, desvanecimentos de curto e longo prazo e mobilidade. No Capítulo 3, são apresentados os conceitos fundamentais para a compreensão e estruturação de um AE. No Capítulo 4 é feita a fundamentação matemática e estudo dos resultados obtidos. Por fim, conclui-se o trabalho no Capítulo 5.

Capítulo 2

Fundamentos de Modelo de Canal

O modelo mais simples de canal considera um caminho livre de obstruções e é chamado linha de visada (*line-of-sight*, LOS). Nele o sinal se propaga em uma linha reta, sem desvios, do transmissor para o receptor [37]. Assim pode-se determinar a relação entre as potências transmitida e recebida como [37, eq. (2.7)]

$$P_{\text{Rx}} = P_{\text{Tx}} \left(\frac{\sqrt{G_e} \lambda}{4\pi d} \right)^2, \quad (2.1)$$

sendo P_{Tx} e P_{Rx} as potências de transmissão e recepção, respectivamente, G_e o produto dos ganhos lineares das antenas de transmissão e recepção, λ o comprimento de onda do sinal transmitido e d a distância entre transmissor e receptor. Dado que a potência transmitida, os ganhos e o comprimento de onda são sempre constantes, a potência de recepção será uma grandeza determinística e decairá com o quadrado da distância.

Por vezes, o modelo LOS não é suficiente para representar o comportamento do canal. Frequentemente, o sinal transmitido não atinge diretamente o receptor. Parte da potência é perdida devido à absorção. Além disso, o sinal, ao interagir com objetos presentes no ambiente de comunicação, sofre reflexão, refração, espalhamento e difração [38]. Assim, o sinal que chega ao receptor é composto pela soma de diversas componentes dos múltiplos percursos causados pelo ambiente de propagação, como ilustrado na Figura 2.1.

Devido à natureza estocástica do canal, cada uma das componentes pode sofrer diferentes atenuações, atrasos e deslocamentos de fase e frequência. Desta forma o sinal recebido tem amplitude, fase, frequência e, conseqüentemente, potência descritas por variáveis aleatórias (VAs) [37, 39, 40]. Quando os múltiplos caminhos são independentes e não encontram nenhuma outra obstrução, o fenômeno de variação da potência é chamado desvanecimento de curto prazo (*short-term fading*). Quando o

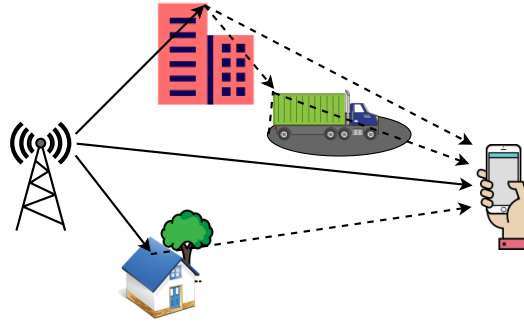


Figura 2.1: Cenário de propagação por múltiplos percursos.

sinal sofre múltiplos espalhamentos, isso é, colide em mais de uma ocasião, ocorre o fenômeno do sombreamento (*shadowing*), ou desvanecimento de longo prazo (*long-term fading*) [39].

É importante ressaltar que para o restante da Seção 2 admite-se o canal como não variante no tempo e não seletivo em frequência (plano), ou seja, na transmissão de um único tom não ocorre alargamento espectral e tons distintos são igualmente afetados pelo canal [40]. Os fenômenos que atuam sobre a potência do sinal, descritos a seguir, são base para o desenvolvimento de novas expressões que permitam analisar o comportamento conjunto deles e seu impacto sobre o desempenho de um sistema de comunicação abordados no Capítulo 4.

2.1 Desvanecimento de Curto Prazo

Sendo um sinal composto por um único tom de frequência f_0 transmitido através de um canal não variante no tempo e plano, o sinal recebido $s_r(t)$ é matematicamente descrito por

$$s_r(t) = \sum_{i=1}^{N_p} a_i \cos(2\pi f_0 t + \phi_i), \quad (2.2)$$

em que a_i e ϕ_i são, respectivamente, o ganho do canal e a rotação de fase da i -ésima componente de um total de N_p múltiplos percursos. Devido ao sinal ser de banda-estreita (um único tom) os atrasos entre os vários caminhos são desconsiderados. Aplicando a identidade trigonométrica do cosseno da soma de dois arcos, é possível reescrever (2.2) como

$$s_r(t) = R_I \cos(2\pi f_0 t) - R_Q \sin(2\pi f_0 t), \quad (2.3)$$

sendo

$$R_I = \sum_{i=1}^{N_p} a_i \cos(\phi_i) \quad (2.4)$$

e

$$R_Q = \sum_{i=1}^{N_p} a_i \sin(\phi_i), \quad (2.5)$$

em que R_I e R_Q denotam, respectivamente, as componentes *em fase* e *quadratura* do sinal recebido.

Admitindo um campo de espalhamento homogêneo e difuso, isto é, um grande número de componentes de espalhamento com amplitudes idênticas, diferentes componentes são descorrelacionadas, a componente de fase ϕ_i e amplitude a_i são descorrelacionadas e a distribuição da fase é uniforme de $[0, 2\pi)$ [41]. Por meio do teorema do limite central, é possível deduzir que R_I e R_Q são VAs com distribuição Gaussiana [42, eq. (7.122)] e $\mathbb{E}[R_I] = \mathbb{E}[R_Q] = 0$, $\mathbb{E}[R_I R_Q] = 0$ e $\mathbb{E}[R_I^2] = \mathbb{E}[R_Q^2] = \sigma^2$, em que $\mathbb{E}[\cdot]$ denota o operador esperança matemática [42, eq. (5.44)]. Com isso, a envoltória do sinal será dada por

$$R = \sqrt{R_I^2 + R_Q^2}, \quad (2.6)$$

em que R é uma VA com distribuição Rayleigh [42, eq. (6.70)], tendo sua FDP expressa por

$$f_R(r) = \frac{2r}{\hat{w}} \exp\left(-\frac{r^2}{\hat{w}}\right), \quad (2.7)$$

para $r \geq 0$, em que $\exp(\cdot)$ é a função exponencial natural e \hat{w} é a potência média do sinal recebido, ou seja, $\hat{w} = \mathbb{E}[R^2] = 2\sigma^2$.

A potência instantânea W é encontrada por meio do processo de transformação de VAs [42, eq. (6.115)], ou seja,

$$W = R^2. \quad (2.8)$$

Neste caso, como a envoltória tem distribuição Rayleigh, a potência instantânea tem distribuição exponencial, assim sua FDP é descrita por [39, eq. (4.7)]

$$f_W(w) = \frac{1}{\hat{w}} \exp\left(-\frac{w}{\hat{w}}\right), \quad (2.9)$$

para $w \geq 0$.

Agora, admitem-se que os obstáculos que causam o espalhamento estejam concentrados em agrupamentos (*clusters*) e aleatoriamente distribuídos. Para cada *cluster* admite-se que os atrasos produzidos sobre as ondas sejam próximos e, entre diferentes

clusters, que os atrasos sejam suficientemente grandes resultando em uma envoltória descrita por

$$R = \sum_{i=1}^{m_f} \sqrt{R_{I_i}^2 + R_{Q_i}^2}, \quad (2.10)$$

em que m_f é o número de *clusters* e R_{I_i} e R_{Q_i} correspondem à componente *em fase* e *quadratura* advinda do i -ésimo *cluster*, respectivamente [39]. Nesse cenário, a envoltória terá distribuição Nakagami- m com FDP descrita por [39, eq. (4.35)]

$$f_R(r) = \frac{2r^{2m_f-1}m_f^{m_f}}{\hat{w}^{m_f}\Gamma(m_f)} \exp\left(-m_f\frac{r^2}{\hat{w}}\right), \quad (2.11)$$

par $r \geq 0$, em que $\Gamma(\cdot)$ é função Gama completa definida na forma integral como [43, eq. (6.1.1)]

$$\Gamma(t) = \int_0^{\infty} x^{t-1}e^{-x}dx, \quad (2.12)$$

\hat{w} é a potência média do sinal e m_f é mais comumente tratado como o parâmetro da distribuição Nakagami, ou fator de desvanecimento, embora fisicamente esteja ligado ao número de *clusters*, matematicamente e é definido como o inverso da variância normalizada de R^2 , sendo

$$m_f = \frac{\hat{w}^2}{\mathbb{V}[R^2]}, \quad (2.13)$$

em que $\mathbb{V}[\cdot]$ representa o operador variância [42, eq. (5.60)].

A FDP da potência instantânea é obtida a partir de (2.8), resultando em

$$f_W(w) = \frac{w^{m_f-1}m_f^{m_f}}{\hat{w}^{m_f}\Gamma(m_f)} \exp\left(-m_f\frac{w}{\hat{w}}\right), \quad (2.14)$$

para $w \geq 0$. De fato, um sinal cuja envoltória tem distribuição Nakagami terá uma potência com distribuição Gama [39, eq. (4.44)].

Nem sempre é possível admitir que o campo de espalhamento seja homogêneo. Em um ambiente real, dificilmente os obstáculos estão aleatoriamente distribuídos. A fim de englobar a heterogeneidade do canal de propagação, distribuições mais abrangentes foram propostas como κ - μ [7], η - μ [7] e α - μ [8], sendo a última o foco deste trabalho.

A distribuição α - μ é obtida ao elevar-se a envoltória resultante em (2.10) a um fator de não linearidade $\alpha/2$ [8]. O que resulta em

$$R^\alpha = \sum_{i=1}^{\mu} R_{I_i}^2 + R_{Q_i}^2. \quad (2.15)$$

A partir de (2.15), obtém-se a FDP da envoltória α - μ como [8, eq. (1)]

$$f_R(r) = \frac{\alpha \mu^\mu r^{\alpha\mu-1}}{\hat{r}^{\alpha\mu} \Gamma(\mu)} \exp \left[-\mu \left(\frac{r}{\hat{r}} \right)^\alpha \right], \quad (2.16)$$

para $r \geq 0$, em que \hat{r} é a raiz α -ésima do valor médio da envoltória,

$$\hat{r} = \sqrt[\alpha]{\mathbb{E}[R^\alpha]} \quad (2.17)$$

e μ é o inverso da variância normalizada de R^α , isto é

$$\mu = \frac{\mathbb{E}^2[R^\alpha]}{\mathbb{V}[R^\alpha]}. \quad (2.18)$$

O parâmetro $\mu > 0$ é uma extensão real do número de *clusters*, portanto não se limita a somente valores inteiros. Também é chamado de parâmetro de desvanecimento.

A distribuição α - μ engloba outros modelos de desvanecimento. A partir dela pode-se obter as distribuições, tais como, Weibull [44] ($\mu = 1$), Rayleigh ($\mu = 1$ e $\alpha = 2$) e Nakagami- m ($\alpha = 2$ e $\mu = m$). Novamente, utilizando de $W = R^2$ encontra-se a FDP da potência, como

$$f_W(w) = \frac{\alpha \mu^\mu w^{\frac{\alpha\mu}{2}-1}}{2\hat{w}^{\frac{\alpha\mu}{2}} \Gamma(\mu)} \exp \left[-\mu \left(\frac{w}{\hat{w}} \right)^{\frac{\alpha}{2}} \right], \quad (2.19)$$

para $w \geq 0$, em que $\hat{w} = \mathbb{E}[W]$. Na Figura 2.2(a) são demonstrados alguns resultados para a envoltória da distribuição α - μ e na Figura 2.2(b) são demonstrados resultados para a potência da distribuição α - μ .

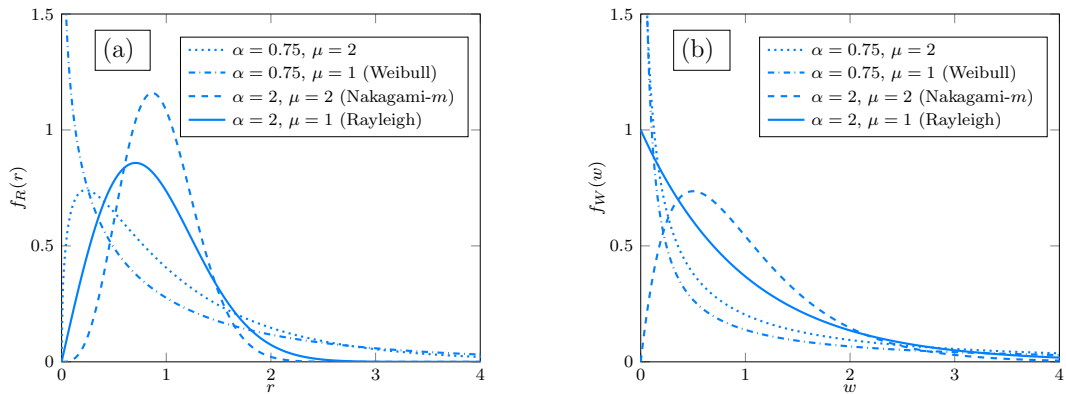


Figura 2.2: FDPs da envoltória e potência da distribuição α - μ e seus casos particulares.

2.2 Sombreamento

Empiricamente, resultados demonstram que a FDP que descreve o sombreamento pode ser modelada por uma VA Lognormal. Teoricamente, devido às múltiplas reflexões causadas pelo meio de propagação, o sombreamento pode ser modelado como uma VA resultante do produto das parcelas de múltiplos espalhamentos [45], isto é

$$Y = \prod_i W_i, \quad (2.20)$$

em que W_i representa a potência da i -ésima parcela de espalhamento. Quando há um grande número de componentes, pelo teorema do limite central para o produto [42, eq. (7.141)], a FDP de Y será dada por [39, eq. (4.67)]

$$f_Y(y) = \frac{C_o}{\sqrt{2\pi}\sigma_{\text{dBW}}y} \exp \left[-\frac{(10 \log_{10} y - \Omega_{\text{dBW}})^2}{2\sigma_{\text{dBW}}^2} \right], \quad (2.21)$$

dados que Ω_{dBW} e σ_{dBW} são, respectivamente, a potência média e o desvio padrão do sombreamento em decibel-Watt (dBW), e que

$$C_o = \frac{10}{\ln(10)}. \quad (2.22)$$

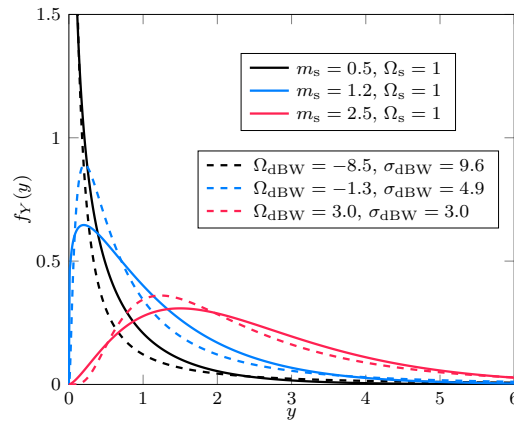


Figura 2.3: Comparação da FDP da distribuição Gama com a equivalente distribuição log-normal.

Outras distribuições foram propostas para modelar o sombreamento além da Lognormal, como a Gama [46,47] e a Gama Inversa [48]. A distribuição Gama tem FDP dada por

$$f_Y(y) = \frac{y^{m_s-1}}{\Omega^{m_s}\Gamma(m_s)} \exp \left(-\frac{y}{\Omega} \right), \quad (2.23)$$

para $y \geq 0$, em que Ω é a potência média do sombreamento e m_s é o parâmetro de sombreamento. A relação entre os modelos Lognormal (2.21) e Gama (2.23) é demonstrada em [39, eq. (4.73)]

$$\Omega_{\text{dBW}} = C_o [\ln(\Omega) + \psi(m_s)] \quad (2.24)$$

e em [39, eq. (4.72)]

$$\sigma_{\text{dBW}}^2 = C_o^2 \psi'(m_s), \quad (2.25)$$

em que $\psi(\cdot)$ é a função digama e $\psi'(\cdot)$ a função trigama [43, eq. (6.4.1)]. Na Figura 2.3 comparam-se as curvas obtidas por meio da expressão (2.23) com os valores correspondentes de Ω_{dBW} e σ_{dBW} para a expressão (2.21).

2.3 Mobilidade

Os modelos que representam os efeitos do desvanecimento de curto prazo e do sombreamento, anteriormente citados, consideram que a distância entre o transmissor e o receptor é constante. Porém, em sistemas móveis, o deslocamento dos nós na rede também afeta a potência média do sinal recebido [17] de forma aleatória.

Há diversos modelos que buscam interpretar a mobilidade dos nós dentro de uma rede, como, por exemplo, Gauss Markov (GM), caminhada aleatória (*random walk*, RW), direção aleatória (*random direction*, RD) e o RWP e são classificados de acordo com suas propriedades estatísticas [49]. Para o modelo RWP, a distribuição espacial dos nós da rede é, em geral, não-uniforme [50], razão que o torna muito utilizado para representar redes dinâmicas, por isso é empregado para analisar o desempenho em sistemas de comunicação sem-fio [17]. Além disso, a distribuição espacial dos nós é expressa como um polinômio em função da distância entre o transmissor e o receptor [15, 51].

No modelo RWP consideram-se que os nós receptores estão localizados em pontos de coordenadas selecionados aleatoriamente dentro de uma área de serviço, a qual é dependente da topologia da rede. Para uma topologia unidimensional, 1-D, considera-se uma linha reta de comprimento D_0 em que o transmissor ou ponto de acesso está localizado na origem. A topologia bidimensional, 2-D, considera que o transmissor está localizado no centro (origem) de uma região circular de raio D_0 (região máxima de cobertura) e o modelo tridimensional, 3-D, é similar ao modelo 2-D, porém, o transmissor está no centro de uma região esférica de raio D_0 . Os nós se movimentam de

forma aleatória dentro desta região conforme a topologia utilizada (1-D, 2-D ou 3-D), percorrendo um caminho em linha reta entre a origem e o destino com velocidade constante. Chegando ao destino, o nó móvel possui duas opções, escolher um novo destino aleatoriamente e iniciar o movimento em outra direção ou esperar um tempo aleatório e posteriormente iniciar o movimento em uma outra direção. Mais detalhes sobre o modelo RWP podem ser vistos em [51].

A distribuição espacial dos nós em estado de repouso é representada por polinômios em função da distância d entre o transmissor e o receptor. A FDP dessa distância é representada da seguinte forma [17, eq. (5)]

$$f_D(d) = \sum_{i=1}^u B_i \frac{d^{\beta_i}}{D_0^{\beta_i+1}}, \quad (2.26)$$

para $0 < d < D_0$, em que D_0 corresponde à máxima distância na área de cobertura e os parâmetros u , B_i e β_i dependem da topologia empregada [17] e são apresentados na Tabela 2.1.

Tabela 2.1: Coeficientes polinomiais

Topologia	u	B_i	β_i
1-D	2	[6, -6]	[1, 2]
2-D	3	(1/73)[324, -420, 96]	[1, 3, 5]
3-D	3	(1/72)[735, -1190, 455]	[2, 4, 6]

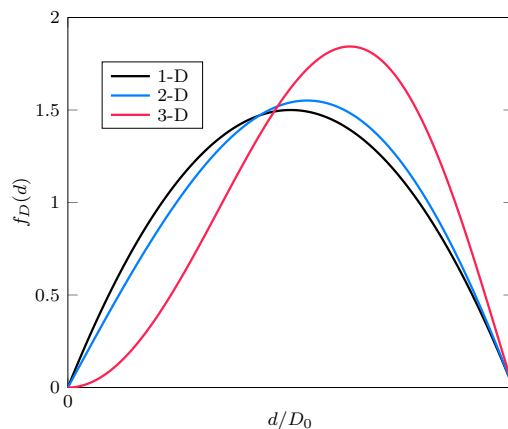


Figura 2.4: FDP da distribuição espacial para o modelo RWP unificada para as topologias uni, bi e tridimensional.

Pela Tabela 2.1 pode-se observar que para cada topologia há um conjunto de valores para os parâmetros u , B_i e β_i , o que levará a um polinômio resultante diferente. A

representação de (2.26) através de um somatório de termos apresentada em [17, eq. (5)] tem por objetivo criar uma FDP unificada de mobilidade que represente qualquer uma das topologias por meio da configuração adequada dos parâmetros apresentados. Na Figura 2.4 é apresentada a FDP da posição espacial dos nós para os 3 possíveis casos.

Capítulo 3

Machine Learning e Autoencoders

A AI é um paradigma computacional que atribui às máquinas a capacidade de aprender e reagir como humanos. O ML é um subconjunto de AI que busca conceder a um algoritmo a capacidade de realizar previsões, sejam elas para problemas de regressão, classificação ou *clustering* (agrupamento), sem que ele seja explicitamente programado para tal tarefa. DL é uma das possíveis técnicas de ML que imita sistemas nervosos biológicos construídos a partir da interconexão de múltiplas camadas. Os algoritmos de DL extraem conhecimento hierarquicamente de dados brutos por meio de várias camadas de unidades de processamento não linear para realizar previsões. Os modelos de aprendizado profundo mais famosos são as NNs e a unidade básica de processamento é chamada neurônio [3].

Quanto ao método de aprendizado, são divididos em três principais grupos, sendo eles supervisionado, não-supervisionado e por reforço. Aprendizado supervisionado é aquele em que os dados de treinamento são rotulados de seus valores verdadeiros ou classe de pertencimento. No aprendizado não-supervisionado os dados não são rotulados, resultando no aprendizado de estruturas-relações-ocultas/latentes. Por fim, o aprendizado por reforço aplica o conceito de punição e premiação, permitindo o algoritmo a se aprimorar e aprender novos conceitos por meio da tentativa e erro.

3.1 Fundamentos

Feedforward Neural Networks (FNNs), ou também chamadas de *multi layer perceptrons* (MLP) são um modelo fundamental de DL. O nome de *feedforward*, que em tradução livre significa “alimentar para frente”, vem do fato de o fluxo de informação ser unidirecional, ou seja, da entrada para a saída sem que haja retroalimentação. A primeira camada de uma FNN é a camada de entrada (*input layer*) e a última recebe

o nome camada de saída (*output layer*). As camadas intermediárias são chamadas camadas ocultas (*hidden layers*), como apresentado na Figura 3.1.

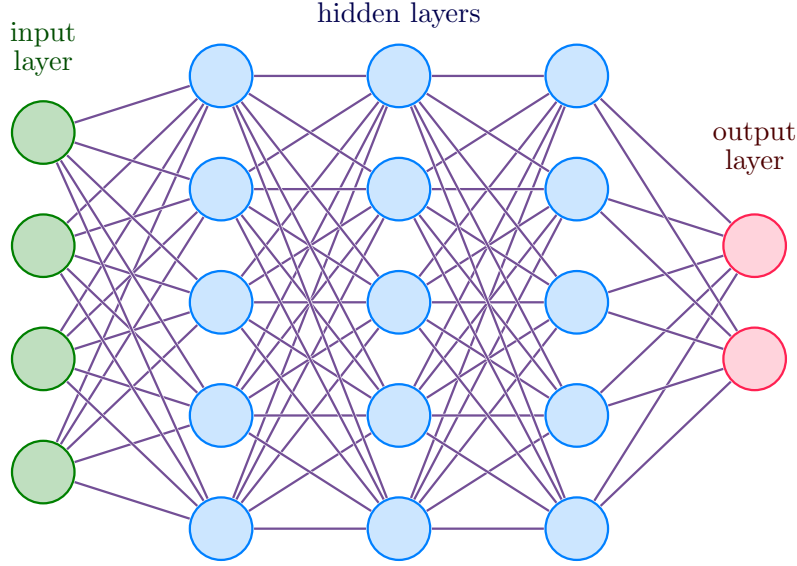


Figura 3.1: Arquitetura de uma rede neural densa.

Como descrito em [28], uma FNN com L camadas realiza o mapeamento $\varphi(\boldsymbol{\kappa}_0, \boldsymbol{\xi}) : \mathbb{R}^{Q_0} \mapsto \mathbb{R}^{Q_L}$ do vetor de entrada $\boldsymbol{\kappa}_0 \in \mathbb{R}^{Q_0}$ para o vetor de saída $\boldsymbol{\kappa}_L \in \mathbb{R}^{Q_L}$, em que Q_0 é a dimensão de entrada e Q_L a dimensão de saída, por meio do processo iterativo

$$\boldsymbol{\kappa}_l = \varphi_l(\boldsymbol{\kappa}_{l-1}, \boldsymbol{\xi}_l), \quad l = 1, 2, \dots, L \quad (3.1)$$

sendo $\varphi_l(\boldsymbol{\kappa}_{l-1}, \boldsymbol{\xi}_l) : \mathbb{R}^{Q_{l-1}} \mapsto \mathbb{R}^{Q_l}$ o mapeamento realizado para a l -ésima camada de dimensionalidade Q_l , $\boldsymbol{\xi} = \{\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_L\}$ o conjunto de todos os pesos da rede e \mathbb{R} o conjunto dos números reais. Uma camada é dita como densa quando está densamente conectada com a camada anterior, isto é, cada um de seus neurônios se conecta com todos os outros da camada progressa. Assim, a l -ésima camada densa realizará a seguinte operação

$$\boldsymbol{\zeta}_l = \boldsymbol{\Psi}_l \boldsymbol{\kappa}_{l-1} + \boldsymbol{\epsilon}_l, \quad (3.2)$$

em que $\boldsymbol{\Psi}_l \in \mathbb{R}^{Q_l \times Q_{l-1}}$ é a matriz dos pesos e $\boldsymbol{\epsilon}_l \in \mathbb{R}^{Q_l}$ é o vetor de *bias* e $\boldsymbol{\zeta}_l$ a saída da camada, ou seja, cada neurônio faz uma combinação linear de suas entradas em relação aos pesos.

A operação (3.2) é linear, desta forma não há benefício em acrescentar múltiplas camadas a um modelo, pois sempre resultaria em uma transformação linear [28]. Para um maior conjunto de possibilidades de mapeamento se faz necessário adicionar não linearidade à NN, o que é feito por meio de funções de ativação. Finalmente, pode-se

escrever o mapeamento realizado pela l -ésima camada como

$$\varphi_l(\boldsymbol{\kappa}_{l-1}, \xi_l) = \rho(\boldsymbol{\zeta}_l) = \rho(\boldsymbol{\Psi}_l \boldsymbol{\kappa}_{l-1} + \boldsymbol{\epsilon}_l), \quad (3.3)$$

em que $\rho(\cdot)$ é a função de ativação e $\xi_l = \{\boldsymbol{\Psi}_l, \boldsymbol{\epsilon}_l\}$ os parâmetros referentes à l -ésima camada. Geralmente, as funções de ativação são aplicadas individualmente a cada elemento resultante da respectiva camada, as mais comuns são apresentadas na Tabela 3.1. Para uma visão mais detalhada recomenda-se a documentação [52].

Tabela 3.1: *Funções de ativação*

Função	$\rho(\boldsymbol{\zeta})$	Domínio
linear	ζ_i	$(-\infty, \infty)$
ReLU	$\max(0, \zeta_i)$	$[0, \infty)$
tanh	$\tanh(\zeta_i)$	$(-1, 1)$
sigmoid	$\frac{1}{1+e^{-\zeta_i}}$	$(0, 1)$
softmax	$\frac{e^{\zeta_i}}{\sum_j e^{\zeta_j}}$	$(0, 1)$

No processo de treinamento, utiliza-se de um conjunto de S amostras rotuladas (portanto se trata de um aprendizado supervisionado) na forma de

$$\mathbb{S} = \{(\boldsymbol{m}_1, \boldsymbol{\tau}_1), \dots, (\boldsymbol{m}_S, \boldsymbol{\tau}_S)\}, \quad (3.4)$$

sendo que para a i -ésima entrada \boldsymbol{m}_i na NN há uma respectiva resposta desejada $\boldsymbol{\tau}_i$. Durante o processo de treinamento o objetivo é minimizar o custo, ou erro, dado por

$$J(\boldsymbol{\xi}) = \frac{1}{S} \sum_{i=1}^S j(\hat{\boldsymbol{\tau}}_i, \boldsymbol{\tau}_i), \quad (3.5)$$

em que $j(\hat{\boldsymbol{\tau}}_i, \boldsymbol{\tau}_i) : \mathbb{R}^{Q_L} \times \mathbb{R}^{Q_L} \mapsto \mathbb{R}$ é a função de custo e $\hat{\boldsymbol{\tau}}_i$ é a resposta estimada pela NN para a entrada \boldsymbol{m}_i . Há um grande conjunto de funções custo que podem ser encontradas em [52]. São listadas na Tabela 3.2 as funções custo para o erro médio quadrático (*mean square error*, MSE) e a entropia cruzada categórica (*categorical cross-entropy*), sendo a primeira mais indicada para problemas de regressão e a segunda mais utilizada para problemas de classificação.

Tabela 3.2: *Funções custo*

Função	$j(\hat{\boldsymbol{\tau}}, \boldsymbol{\tau})$
MSE	$(\boldsymbol{\tau} - \hat{\boldsymbol{\tau}})^2$
<i>Categorical cross-entropy</i>	$-\sum_i \tau_i \log(\hat{\tau}_i)$

Matematicamente, a arquitetura da NN é diferenciável, portanto, uma das maneiras de se encontrar o conjunto de parâmetros ξ é por meio do gradiente descendente estocástico (GDE). O GDE inicia de um valor aleatório ξ_0 e atualizar os valores pela iteração

$$\xi_{\iota+1} = \xi_{\iota} - \eta \nabla \tilde{J}(\xi_{\iota}), \quad (3.6)$$

em que $\iota = \{0, 1, \dots\}$ denota a ι -ésima iteração, η é o passo de aprendizagem, ∇ é o operador gradiente e $\tilde{J}(\xi_{\iota})$ é a aproximação da função erro computada a partir de um subconjunto das amostras de treinamento (*mini batch*) $\mathbb{S}_b \subset \mathbb{S}$, ou seja,

$$\tilde{J}(\xi_{\iota}) = \frac{1}{S_b} \sum_{i=1}^{S_b} j(\hat{\tau}_i, \tau_i), \quad (3.7)$$

em que $0 < S_b \leq S$. O gradiente em (3.6) é eficientemente encontrado pelo método da retro-propagação do erro. Primeiramente, computa-se o gradiente da função custo $J(\xi_t)$ em relação aos pesos da última camada oculta, então atualizam-se os pesos por

$$\xi_l \Big|_{\iota+1} = \left[\xi_l - \eta \frac{\partial J(\xi)}{\partial \xi_l} \right]_{\iota}, \quad (3.8)$$

para $\iota = \{0, 1, \dots\}$ e $l = \{1, \dots, L\}$, sendo que ∂ denota o operador de derivada parcial. A mesma operação é realizada para todas as demais camadas, sendo a derivada parcial para cada conjunto de pesos encontrada pela regra da cadeia [3].

3.2 Autoencoder

Um AE é uma NN treinada para reconstruir uma cópia da entrada em sua saída. Tradicionalmente é utilizado para redução de dimensionalidade e é dividido em três partes: codificador (*encoder*), código (*code*) e decodificador (*decoder*), como ilustrado na Figura 3.2.

A camada interna oculta h representa o código de profundidade Q (dimensionalidade da palavra codificada) utilizado para representação da informação da entrada. O codificador tem a função de aplicar uma transformação $f : \mathbb{M} \mapsto \mathbb{R}^Q$ para a entrada $\mathbf{m} \in \mathbb{M}$ para gerar a função de codificação $h = f(\mathbf{m}) \in \mathbb{R}^Q$. O decodificador utiliza do mapeando $g : \mathbb{R}^Q \mapsto \mathbb{M}$ para reconstruir, a mensagem de entrada por $\hat{\mathbf{m}} = g(h)$. O processo de aprendizagem do AE pode ser descrito pela minimização do custo da função de reconstrução

$$J(\mathbf{m}, g(f(\mathbf{m}))), \quad (3.9)$$

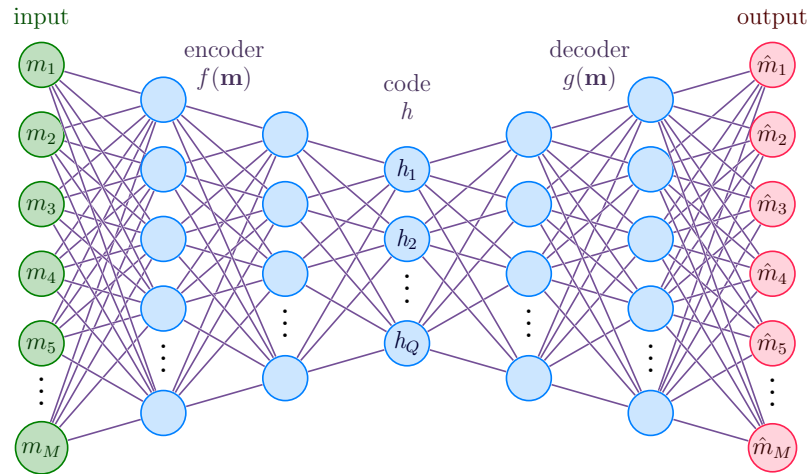


Figura 3.2: Arquitetura de um autoencoder.

em que $J(\cdot)$ uma função custo capaz de penalizar a diferença entre a entrada e a saída estimada.

AEs são vistos como um caso particular de FNN, portanto também são treinados da mesma maneira, geralmente utilizando gradiente descendente e retro-propagação. Dependendo da aplicação, podem ser do tipo incompleto, esparso, *denoising* ou contrativo [53].

3.3 Autoencoder como mecanismo de comunicação

3.3.1 Modelo de Comunicação

Um sistema de comunicação pode ser analisado em três blocos, sendo eles, um transmissor e um receptor conectados por um canal, como apresentado na Figura 3.3.

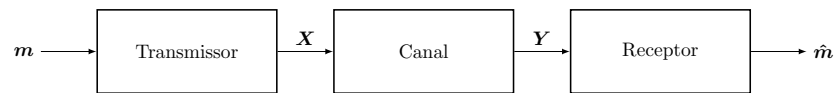


Figura 3.3: Diagrama em blocos de um sistema de comunicação.

A mensagem $m \in \mathbb{M} = \{0, 1, \dots, M - 1\}$ é enviada por um canal com uma utilização complexa $n = \{1, 2, \dots\}$ a uma taxa dada por

$$R_b = \frac{\log_2(M)}{n} \text{ [bits/uso complexo do canal] }, \quad (3.10)$$

em que $\log_2 M$ é o número de bits por mensagem, o uso complexo do canal n equivale a $2n$ usos reais do canal. O transmissor aplica uma transformação $f : \mathbb{M} \mapsto \mathbb{C}^n$ à

mensagem m para gerar os símbolos transmitidos $\mathbf{X} = f(m) \in \mathbb{C}^n$, em que \mathbb{C} é o conjunto dos números complexos. Após passar pelo canal, o decodificador realiza um mapeamento $g : \mathbb{C}^n \mapsto \mathbb{M}$ dos símbolos recebidos $\mathbf{Y} \in \mathbb{C}^n$ a fim de, por meio da função $\hat{m} = g(\mathbf{Y})$, estimar a mensagem recebida. Ao sinal transmitido \mathbf{X} é imposta uma restrição de potência, de modo que a energia média E , dada por

$$E = \frac{1}{n} \sum_{i=1}^n |\mathbf{x}_i|^2 = 1, \quad (3.11)$$

seja unitária, sendo $\mathbf{x}_i \in \mathbb{C}$ o i -ésimo símbolo complexo gerado pelo transmissor. A taxa de erro de mensagem \mathcal{P}_s é definida como a probabilidade de erro sobre todas as possíveis mensagens, sendo

$$\mathcal{P}_s = \frac{1}{S_b} \sum_{i=1}^{S_b} P[\hat{m}_i \neq m_i | m_i], \quad (3.12)$$

em que $P[\hat{m}_i \neq m_i | m_i]$ é a probabilidade de não reconhecimento da i -ésima mensagem m_i condicionada ao envio do mesmo.

3.3.2 Implementação

A arquitetura do AE utilizada nesta dissertação está implementada tal qual o diagrama da Figura 3.4. O transmissor consiste em uma camada de incorporação (*embedding layer*), uma DNN composta por duas camadas, uma camada de transformação de real para complexo e uma camada de restrição de energia. Já o decodificar segue uma lógica reversa, sendo composto por uma camada de conversão, seguida por mais duas camadas densas e por fim uma função para estimação da mensagem [28, 29, 32].

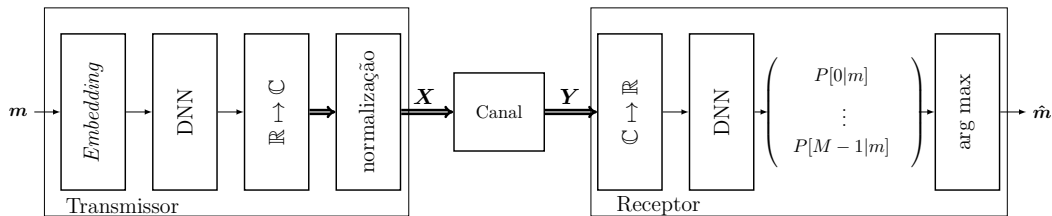


Figura 3.4: Diagrama da implementação do AE.

A camada de incorporação (*Embedding Layer*) tem funcionamento análogo a uma tabela de consulta (*lookup table*, LUT). Dado um vocabulário de tamanho fixo, que para este caso é M , a camada será responsável por fazer o mapeamento da entrada discreta m em um vetor denso de dimensões fixas composto de valores contínuos e

Tabela 3.3: Configuração das DNNs

DNN	camada densa	função de ativação	dimensão de saída
codificador	1	relu	M
	2	linear	$2n$
decodificador	1	relu	M
	2	softmax	M

seus pesos são aleatoriamente iniciados [52]. Em seguida, tem-se a DNN, que está disposta como indicado na Tabela 3.3. É composta por duas camadas densas ocultas de dimensões de saída M e $2n$ e têm como funções de ativação *rectified linear unit* (relu) e linear, respectivamente. Na saída da DNN tem-se a conversão de $2n$ valores reais para n valores complexos. Por fim, a camada de normalização tem o papel de garantir que o conjunto de símbolos tenha uma restrição de energia média unitária. Essa camada é necessária para garantir que o transmissor aprenda símbolos com energia limitada, ou seja, impede que as saídas sejam desnecessariamente grandes e gere instabilidade no processo de aprendizagem. Uma alternativa à não adoção de valores complexos é concatenar os vetores que representam a parte real e a parte imaginária, ou seja, $\mathbf{X} \in \mathbb{R}^{2n}$.

A estrutura do decodificador consiste em uma camada de conversão de complexo para real, concatenando a partes real e imaginária em uma única dimensão. Em seguida, a DNN como na Tabela 3.3 é composta por duas camadas densas de dimensão de saída M e tendo por funções de ativação relu e softmax, ordenadamente.

A função softmax realiza um mapeamento $\rho : \mathbb{R}^M \mapsto (0, 1)^M$ retornando um vetor $\mathbf{p} \in (0, 1)^M$. Dado um vetor $\mathbf{b} \in \mathbb{R}^M$, a probabilidade p_i associada ao i -ésimo elemento b_i é obtida por

$$p_i = \frac{\exp(b_i)}{\sum_{j=0}^{M-1} \exp(b_j)}. \quad (3.13)$$

De fato, o que se tem na saída é um vetor que estima a probabilidade da mensagem enviada m , após passar por todo o sistema, pertencer a cada uma das M classes (possíveis mensagens). A mensagem estimada \hat{m} , portanto, consiste no índice de maior valor absoluto de \mathbf{p} , de forma que pode ser encontrado pela função *argmax*.

Uma vez que a tarefa do decodificador consiste em um problema de classificação, como função de custo utiliza-se a entropia cruzada categórica esparsa (*sparse categorical crossentropy*), uma vez que é uma função mais eficiente e evita o uso de vetores *one-hot* diretamente, que são esparsos.

Para o treinamento, são geradas amostras uniformemente distribuídas no intervalo

$0 \leq m \leq M - 1$ para cada iteração. É importante ressaltar que, neste caso, admite-se que o decodificador tenha conhecimento da mensagem enviada, o que na prática pode ser obtido com um gerador de sequências aleatórias. Por fim, utiliza-se o algoritmo de estimação de momento adaptativa (*adaptive moment estimation*, Adam) [54], que é uma extensão do gradiente descendente estocástico, para a atualização dos pesos.

Ainda, é preciso notar que, sobre o treinamento, a natureza estocástica do canal atua como uma forma de regularização das amostras, o que impossibilita que ocorra o sobre-ajuste do modelo. Em outras palavras, para cada mensagem de índice m existem infinitas possibilidades de amostras resultantes do processo de filtragem do canal [29].

Para a implementação utilizou-se a linguagem de programação Python [55], tendo como principal biblioteca o Tensorflow [56], auxiliada pela interface de programação de aplicação (*application programming interface*, API) Keras [52], baseado no trabalho prévio de [32]. Mais detalhes sobre a implementação estão dispostas no Apêndice I.

3.3.3 Desempenho em Canal AWGN

A seguir, apresentam-se resultados para um canal com ruído aditivo Gaussiano branco (*additive white Gaussian noise*, AWGN). Os parâmetros de treinamento utilizados são: passo de aprendizagem $\eta = 0.001$, *batch size* $S_b = 100$, passos = 512, épocas = 5 (número de iterações sobre os passos, totalizando 2560). O canal é representado por uma camada aditiva de ruído Gaussiano com variância $\sigma^2 = (2E/N_0)^{-1}$, em que E/N_0 denota a relação entre a energia média por símbolo, E , pela densidade espectral de potência unilateral do ruído, N_0 . A média para a normalização de energia é tomada sobre a dimensão do sinal e também sobre o *batch size* e o ruído é independente para cada canal, assim como feito em [32]. Para todo o treinamento é utilizado $E/N_0 = 7$ decibel (dB). O valor de E/N_0 deve ser grande suficiente para que o AE seja capaz de generalizar seu desempenho para demais valores de N_0 mas não tão grande a ponto de que o sistema aprenda apenas a forma de ruído, de forma que o valor ótimo é encontrado empiricamente.

São analisados dois cenários. O primeiro considera um conjunto de 4 mensagens para uma utilização unitária de canal, ou seja $M = 4$ e $n = 1$. O segundo, para o mesmo conjunto de mensagens transmitidas dobra-se a utilização do canal, portanto, $M = 4$ e $n = 2$. Na Figura 3.5(a) é apresentada a progressão do custo por iteração conjuntamente com o custo médio por época para o caso em que $M = 4$ e $n = 1$ e na Figura 3.5(b), para o caso em que $M = 4$ e $n = 2$. Pode-se observar que para $M = 4$ e $n = 2$ é possível atingir um menor valor da função custo, o que reflete diretamente no

desempenho do sistema. Devido à adição de redundância ao sinal, ou seja, uma mesma mensagem é representada por um maior número de símbolos, a chance de mensagem ser corretamente decodificada no receptor é maior e isso se reflete no erro, o qual é diretamente relacionado à mensagem recebida.

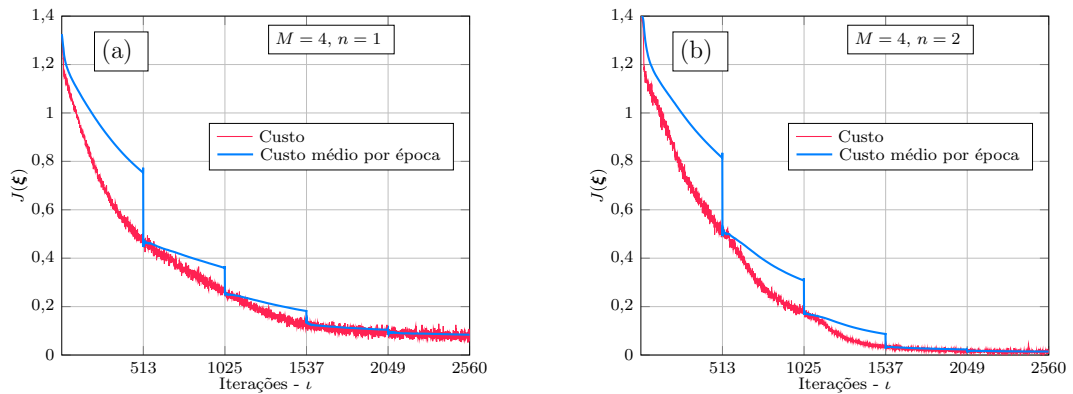


Figura 3.5: Custo por iteração e custo médio por época para o AE.

A disposição dos símbolos aprendida pelo AE é apresentada na Figura 3.6(a) para $M = 4$ e $n = 1$ e na Figura 3.6(b) para $M = 4$ e $n = 2$. Quando a utilização do canal é unitária, a disposição dos símbolos se assemelha a uma constelação de uma modulação M -ária por amplitude em quadratura (M -ary quadrature amplitude modulation, M -QAM) tradicional. Quando a utilização do canal sobe para 2, pode-se analisar como se fossem transmitidas duas constelações paralelas, cada uma em seu canal. Ainda, nota-se que um símbolo que tenha uma representação de menor energia em um canal terá, no outro, uma de maior energia. É importante ressaltar que a disposição geométrica, bem como a mensagem a ser representada, pode variar de acordo com os parâmetros da estrutura e treinamento do AE.

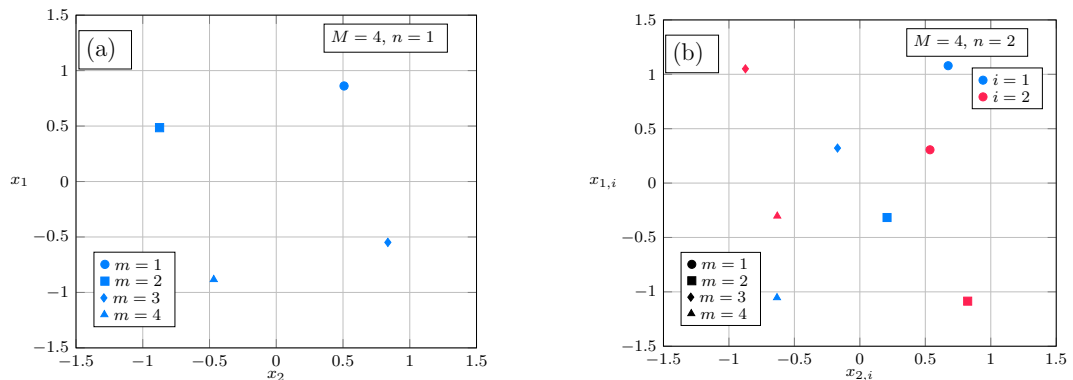


Figura 3.6: Disposição dos símbolos aprendida pelo AE.

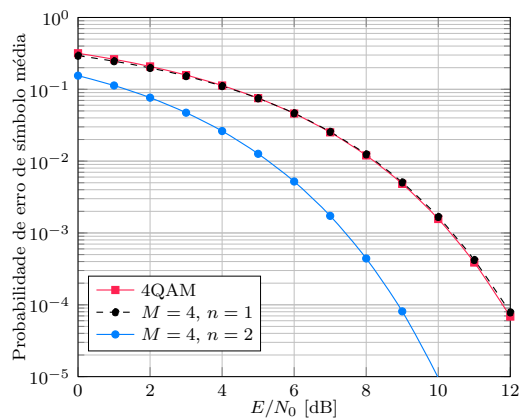


Figura 3.7: Probabilidade de erro de símbolo média comparada entre um AE e 4-QAM.

Por fim, comparam-se os desempenhos dos dois sistemas implementados via um AE com um sistema 4-QAM e modulação binária por deslocamento de fase (*binary phase shift keying*, BPSK) na Figura 3.7. A análise da probabilidade de erro é feita sobre os símbolos, o que é equivalente aos blocos em um AE. Com a utilização unitária do canal, o desempenho do AE é semelhante ao teórico para o canal AWGN. Quando a utilização do canal passa para $n = 2$, tem-se uma melhora significativa de desempenho. O custo da redundância pode ser visto de duas maneiras, um maior consumo de banda de transmissão, ou, caso haja uma concatenação temporal (serial), uma redução da taxa de comunicação. Os resultados referentes ao AE para $n = 2$ indicam que ele é equivalente ao código de repetição, o que significa que teria desempenho consideravelmente inferior à outros códigos conhecidos na literatura. A comparação do AE e outros esquemas de comunicação utilizando códigos é abordada em [28].

Capítulo 4

Contribuições e Resultados

COMO apresentado anteriormente, a mobilidade do usuário em um sistema de comunicação tem impacto sobre seu desempenho. No decorrer do presente capítulo, será primeiramente apresentado o processo necessário para incorporar o sombreamento Gama ao desvanecimento α - μ , que foram individualmente analisados no Capítulo 2. Depois, será acrescentado um deslocamento aleatório do tipo RWP ao modelo de canal, a partir do qual deduz-se expressões para métricas de desempenho do sistema e apresentam-se análises dos resultados obtidos. São deduzidas expressões para FDP, FDC, ASEP, PI e capacidade. Os resultados apresentados dialogam com os trabalhos [17, 20], porém são mais generalistas, sendo portanto novas contribuições. Por fim, é apresentado o processo de treinamento e validação do desempenho de um AE, que foi discutido no Capítulo 3, sobre o modelo de canal α - μ /Gama com deslocamento aleatório RWP.

4.1 Modelo de Desvanecimento α - μ /Gama

A FDP da potência instantânea da distribuição α - μ é expressa em (2.19), aqui reproduzida por

$$f_W(w) = \frac{\alpha \mu^\mu w^{\frac{\alpha\mu}{2}-1}}{2\hat{w}^{\frac{\alpha\mu}{2}} \Gamma(\mu)} \exp \left[-\mu \left(\frac{w}{\hat{w}} \right)^{\frac{\alpha}{2}} \right]. \quad (4.1)$$

Para incorporar o sombreamento, faz-se $\hat{w} = Y$, de modo que Y seja uma VA Gama com distribuição expressa em (2.23), aqui reproduzida por

$$f_Y(y) = \frac{y^{m_s-1}}{\Omega^{m_s} \Gamma(m_s)} \exp \left(-\frac{y}{\Omega} \right), \quad (4.2)$$

resultando em uma FDP condicional $f_W(w|y)$ da forma

$$f_W(w|y) = \frac{\alpha\mu^\mu w^{\frac{\alpha\mu}{2}-1}}{2y^{\frac{\alpha\mu}{2}} \Gamma(\mu)} \exp \left[-\mu \left(\frac{w}{y} \right)^{\frac{\alpha}{2}} \right], \quad (4.3)$$

para $w \geq 0$ e $y \geq 0$. A FDP da potência é obtida descondicionando $f_W(w|y)$, isto é, tomando sua média sobre a potência média de sombreamento Y , portanto

$$f_W(w) = \int_0^\infty f_W(w|y) f_Y(y) dy. \quad (4.4)$$

Substituindo (4.2) e (4.3) em (4.4), obtém-se

$$f_W(w) = \frac{\alpha\mu^\mu w^{\frac{\alpha\mu}{2}-1}}{2\Omega^{m_s} \Gamma(\mu) \Gamma(m_s)} \int_0^\infty y^{m_s - \frac{\alpha\mu}{2} - 1} \exp \left[-\frac{y}{\Omega} - \mu \left(\frac{w}{y} \right)^{\frac{\alpha}{2}} \right] dy. \quad (4.5)$$

A solução da integral em (4.5) necessita de algumas manipulações. Para melhor visualização e para não tornar o processo massante, as simplificações dos termos constantes em relação à integração serão omitidas. Primeiramente, realiza-se a mudança de variáveis de integração, da forma

$$y = \frac{w\mu^{\frac{2}{\alpha}}}{t}. \quad (4.6)$$

Após a mudança de variáveis, a integral em (4.5) resulta em

$$f_W(w) = \mathcal{C} \int_0^\infty t^{\frac{\alpha\mu}{2} - m_s - 1} \exp \left(-t^{\frac{\alpha}{2}} - \frac{w\mu^{\frac{2}{\alpha}}}{t\Omega} \right) dt, \quad (4.7)$$

em que \mathcal{C} é o termo referente às constantes omitidas. Então, a integral (4.7), pode ser representada na forma da função de Krätzel definida como [57, eq. (1.120)]

$$Z_p^v(z) = \int_0^\infty t^{v-1} \exp \left(-t^p - \frac{z}{t} \right) dt, \quad (4.8)$$

para $z > 0$, $p \in \mathbb{R}$ e $v \in \mathbb{C}$, muito comum em análises físico-químicas, mecânica estatística e geração de energia nuclear [58]. Utilizando a transformada de Mellin, é possível reescrever a função de Krätzel em termos da função H de Fox $H_{p,q}^{m,n}(\cdot|\cdot)$ [57, eq. (1.2)], mais prática e já amplamente utilizada para análises de canais. Mais detalhes sobre a função H de Fox estão disponíveis no Anexo A. Assim, é possível estabelecer

a seguinte relação [57, eq. (1.141)]

$$Z_p^v(z) = \frac{1}{p} H_{0,2}^{2,0} \left(z \left| \begin{matrix} - \\ (0,1), (\frac{v}{p}, \frac{1}{p}) \end{matrix} \right. \right). \quad (4.9)$$

Então, com o auxílio de (4.8) e (4.9) resolve-se (4.7) e, após a manipulação dos termos constantes, obtém-se

$$f_W(w) = \frac{w^{-1}}{\Gamma(\mu)\Gamma(m_s)} \left(\frac{w\mu^{\frac{2}{\alpha}}}{\Omega} \right)^{m_s} H_{0,2}^{2,0} \left(\frac{w\mu^{\frac{2}{\alpha}}}{\Omega} \left| \begin{matrix} - \\ (0,1), (\mu - \frac{2m_s}{\alpha}, \frac{2}{\alpha}) \end{matrix} \right. \right). \quad (4.10)$$

Por fim, ainda é possível simplificar (4.10), empregando-se da propriedade [57, eq. (1.60)]. Finalmente, a FDP da potência instantânea α - μ com sombreado Gama é expressa por

$$f_W(w) = \frac{w^{-1}}{\Gamma(\mu)\Gamma(m_s)} H_{0,2}^{2,0} \left(\frac{w\mu^{\frac{2}{\alpha}}}{\Omega} \left| \begin{matrix} - \\ (m_s,1), (\mu, \frac{2}{\alpha}) \end{matrix} \right. \right). \quad (4.11)$$

4.2 Mobilidade Incorporada ao Desvanecimento

Até aqui, a potência média de recepção Ω foi tratada como uma constante. Agora, adotando processo análogo ao realizado em [17], o parâmetro Ω será função de uma potência de transmissão constante P_{Tx} , um expoente de perda de percurso δ ($2 \leq \delta \leq 5$) e uma distância de propagação variável d . Substituindo $\Omega = P_{Tx}d^{-\delta}$ em (4.11), a FDP da potência condicionada à distância de propagação será dada por

$$f_W(w|d) = \frac{1}{w\Gamma(\mu)\Gamma(m_s)} H_{0,2}^{2,0} \left(\frac{\mu^{\frac{2}{\alpha}}w}{P_{Tx}d^{-\delta}} \left| \begin{matrix} - \\ (m_s,1), (\mu, \frac{2}{\alpha}) \end{matrix} \right. \right). \quad (4.12)$$

O modelo de deslocamento aleatório RWP tem FDP expressa em (2.26), aqui reproduzida por

$$f_D(d) = \sum_{i=1}^u B_i \frac{d^{\beta_i}}{D_0^{\beta_i+1}}, \quad (4.13)$$

para $0 < d < D_0$. Novamente, realizando o descondicionamento de variáveis, substituindo (4.13) e (4.12) em

$$f_W(w) = \int_0^{D_0} f_W(w|d) f_D(d) dd, \quad (4.14)$$

obtém-se

$$f_W(w) = \frac{w^{-1}}{\Gamma(\mu)\Gamma(m_s)} \sum_{i=1}^u B_i \int_0^{D_0} \frac{d^{\beta_i}}{D_0^{\beta_i+1}} H_{0,2}^{2,0} \left(\frac{\mu^{\frac{2}{\alpha}} w}{P_{Tx} d^{-\delta}} \middle| \begin{matrix} - \\ (m_s, 1), (\mu, \frac{2}{\alpha}) \end{matrix} \right) dd. \quad (4.15)$$

Para a solução da integral em (4.15) adota-se procedimento semelhante ao realizado em [17]. Partindo da mudança de variáveis

$$U = \left(\frac{d}{D_0} \right)^{\delta}, \quad (4.16)$$

a integral resulta em

$$f_W(w) = \frac{(\delta w)^{-1}}{\Gamma(\mu)\Gamma(m_s)} \sum_{i=1}^u B_i \int_0^1 U^{\frac{\beta_i+1}{\delta}-1} H_{0,2}^{2,0} \left(\frac{\mu^{\frac{2}{\alpha}} w U}{P_{Tx} D_0^{-\delta}} \middle| \begin{matrix} - \\ (m_s, 1), (\mu, \frac{2}{\alpha}) \end{matrix} \right) dU. \quad (4.17)$$

Com o auxílio de [57, eq. (2.53)], é possível determinar a FDP do desvanecimento α - μ /Gama com deslocamento aleatório RWP incorporado como sendo

$$f_W(w) = \frac{(\delta w)^{-1}}{\Gamma(\mu)\Gamma(m_s)} \sum_{i=1}^u B_i H_{1,3}^{2,1} \left(\frac{\mu^{\frac{2}{\alpha}} w}{\Omega_0} \middle| \begin{matrix} (1 - \frac{\beta_i+1}{\delta}, 1) \\ (m_s, 1), (\mu, \frac{2}{\alpha}), (-\frac{\beta_i+1}{\delta}, 1) \end{matrix} \right), \quad (4.18)$$

para $w > 0$, em que Ω_0 é definido como a potência na máxima distância de cobertura D_0 da célula, isto é, $\Omega_0 = P_{Tx} D_0^{-\delta}$.

Por definição, a FDC é obtida por

$$F_W(w) = \int_0^w f_W(t) dt. \quad (4.19)$$

Para a distribuição em questão, aplicando (4.18) em (4.19) e com o auxílio de [57, eq. (2.53)], imediatamente resulta em

$$F_W(w) = \frac{\delta^{-1}}{\Gamma(\mu)\Gamma(m_s)} \sum_{i=1}^u B_i H_{2,4}^{2,2} \left(\frac{\mu^{\frac{2}{\alpha}} w}{\Omega_0} \middle| \begin{matrix} (1, 1), (1 - \frac{\beta_i+1}{\delta}, 1) \\ (m_s, 1), (\mu, \frac{2}{\alpha}), (-\frac{\beta_i+1}{\delta}, 1), (0, 1) \end{matrix} \right). \quad (4.20)$$

Fazendo $\alpha = 2$ em (4.18) e (4.20), obtém-se um caso particular que é a distribuição Gama/Gama com a mobilidade RWP incorporada. O que equivale à potência de uma distribuição de envoltória Nakagami/Gama. Então, fazendo $\mu = m_f$, o parâmetro de desvanecimento Nakagami, com o auxílio da propriedade [59, eq. (2.9.1)], obtém-se

$$f_W(w) = \frac{1}{w\delta\Gamma(m_f)\Gamma(m_s)} \sum_{i=1}^u B_i G_{1,3}^{2,1} \left(\frac{m_f w}{\Omega_0} \middle| \begin{matrix} 1 - \frac{\beta_i+1}{\delta} \\ m_s, m_f, -\frac{\beta_i+1}{\delta} \end{matrix} \right), \quad (4.21)$$

$$F_W(w) = \frac{1}{\delta\Gamma(m_f)\Gamma(m_s)} \sum_{i=1}^u B_i G_{2,4}^{2,2} \left(\frac{m_f w}{\Omega_0} \left| \begin{matrix} 1, 1 - \frac{\beta_i + 1}{\delta} \\ m_s, m_f, -\frac{\beta_i + 1}{\delta}, 0 \end{matrix} \right. \right), \quad (4.22)$$

em que $G_{p,q}^{m,n}(\cdot|\cdot)$ é a função G de Meijer [57, eq. (1.112)].

4.3 Métricas de Desempenho

4.3.1 Probabilidade de erro de símbolo média

A ASEP para BPSK, modulação binária por deslocamento de frequência (*binary frequency shift keying*, BFSK), modulação M -ária por deslocamento de fase (*M-ary phase shift keying*, M -PSK), modulação M -ária por deslocamento de frequência (*M-ary Frequency Shift Keying*, M -FSK) e M -QAM com constelações quadradas pode ser encontrada por [60, eq. (19)]

$$\mathcal{P}_s = \int_0^\infty c_1 \operatorname{erfc}(\sqrt{c_2 w}) f_W(w) dw, \quad (4.23)$$

em que $\operatorname{erfc}(\cdot)$ é a função erro complementar [43, eq. (7.1.2)] e os coeficientes c_1 e c_2 dependem da modulação utilizada. Na Tabela 4.1 são apresentados os valores de c_1 e c_2 para os possíveis esquemas de modulação.

Tabela 4.1: Parâmetros c_1 e c_2 para diversos esquemas de modulação

Modulação	c_1	c_2	M
BPSK	1/2	1	2
BFSK	1/2	1/2	2
M -PSK	1	$\operatorname{sen}^2(\pi/M)$	≥ 4
M -FSK	$(M-1)/2$	1/2	> 2
M -QAM	$2 - 2/\sqrt{M}$	$3/(2M-2)$	≥ 4

A ASEP também pode ser calculada em função da FDC por meio de um processo de integração por partes, fazendo $u = c_1 \operatorname{erfc}(\sqrt{c_2 w})$ e $dv = f_W(w)dw$ em (4.23), assim, obtém-se

$$\mathcal{P}_s = c_1 \sqrt{\frac{c_2}{\pi}} \int_0^\infty w^{-\frac{1}{2}} \exp(-c_2 w) F_W(w) dw. \quad (4.24)$$

Substituindo (4.20) em (4.24) e com auxílio de [61, eq. (2.25.2.3)] obtém-se a ASEP para o canal α - μ /Gama com deslocamento aleatório RWP como sendo

$$\mathcal{P}_s = \frac{c_1}{\sqrt{\pi}\delta\Gamma(\mu)\Gamma(m_s)} \sum_{i=1}^u B_i H_{3,4}^{2,3} \left(\frac{\mu^{\frac{2}{\alpha}}}{\Omega_0 c_2} \left| \begin{matrix} (\frac{1}{2}, 1), (1, 1), (1 - \frac{\beta_i + 1}{\delta}, 1) \\ (m_s, 1), (\mu, \frac{2}{\alpha}), (-\frac{\beta_i + 1}{\delta}, 1), (0, 1) \end{matrix} \right. \right). \quad (4.25)$$

Novamente, fazendo $\alpha = 2$ e $\mu = m_f$, obtém-se a simplificação para a distribuição Nakagami/Gama como

$$\mathcal{P}_s = \frac{c_1}{\sqrt{\pi}\delta\Gamma(m_f)\Gamma(m_s)} \sum_{i=1}^u B_i G_{3,4}^{2,3} \left(\frac{m_f}{\Omega_0 c_2} \middle| \begin{matrix} \frac{1}{2}, 1, 1 - \frac{\beta_i + 1}{\delta} \\ m_s, m_f, -\frac{\beta_i + 1}{\delta}, 0 \end{matrix} \right). \quad (4.26)$$

A fim de analisar o impacto que o desvanecimento provoca no desempenho do sistema, será feita a análise assintótica da ASEP em regimes de elevada relação sinal-ruído (RSR). Fazendo $\Omega_0 \rightarrow \infty$ em (4.18) e com o auxílio [59, eq. (1.8.4)] é possível representar o valor aproximado da FDP para uma RSR elevada na máxima distância de cobertura como

$$\tilde{f}_W(w) = \frac{1}{\delta\Gamma(\mu)\Gamma(m_s)} \sum_{i=1}^u B_i \mathcal{O}_i, \quad (4.27)$$

em que

$$\mathcal{O}_i = \begin{cases} \frac{\Gamma(\mu - \frac{2m_s}{\alpha})}{\frac{\beta_i + 1}{\delta} + m_s} \left(\frac{\mu}{\Omega_0} w \right)^{m_s}, & m_s < \frac{\alpha\mu}{2} \\ \frac{\alpha\Gamma(m_s - \frac{\alpha\mu}{2})}{2(\frac{\beta_i + 1}{\delta} + \frac{\alpha\mu}{2})} \left(\frac{\mu}{\Omega_0} w \right)^{\frac{\alpha\mu}{2}}, & m_s > \frac{\alpha\mu}{2}. \end{cases} \quad (4.28)$$

Transferindo (4.27) em (4.23) e com o auxílio de [62, eq. (6.455.1)] a ASEP assintótica será representada por

$$\tilde{\mathcal{P}}_s = \frac{c_1}{\sqrt{\pi}\Gamma(\mu)\Gamma(m_s)} \sum_{i=1}^u B_i \mathcal{M}_i, \quad (4.29)$$

em que

$$\mathcal{M}_i = \begin{cases} \frac{\Gamma(\mu - \frac{2m_s}{\alpha})\Gamma(m_s + \frac{1}{2})\mu^{\frac{2m_s}{\alpha}}}{(\beta_i + 1 + \delta m_s)m_s c_2^{m_s}} \Omega_0^{-m_s}, & m_s < \frac{\alpha\mu}{2} \\ \frac{\Gamma(m_s - \frac{\alpha\mu}{2})\Gamma(\frac{\alpha\mu + 1}{2})\mu^\mu}{(\beta_i + 1 + \delta \frac{\alpha\mu}{2})\mu c_2^{\frac{\alpha\mu}{2}}} \Omega_0^{-\frac{\alpha\mu}{2}}, & m_s > \frac{\alpha\mu}{2}. \end{cases} \quad (4.30)$$

As equações (4.27) e (4.29) são válidas somente para $m_s \neq \alpha\mu/2$. Em um regime de elevada RSR, a ASEP de um sistema pode ser aproximadamente representada por

$$\tilde{\mathcal{P}}_s = (\mathcal{G}_c \Omega_0)^{-\mathcal{G}_d}, \quad (4.31)$$

sendo \mathcal{G}_c o ganho de codificação e \mathcal{G}_d o ganho, ou ordem, de diversidade. Claramente, comparando com (4.29), para ambos os casos, estático e com mobilidade, o ganho de diversidade é m_s para $m_s < \alpha\mu/2$ e será $\alpha\mu/2$ quando $m_s > \alpha\mu/2$. Entretanto, o ganho de codificação difere do caso estático, pois está relacionado com a dimensão da

rede, coeficiente de perda de percurso além do ganho de diversidade.

4.3.2 Probabilidade de Indisponibilidade

A PI é uma métrica importante para sistemas de comunicação sem fio. Neste trabalho, ela indica a chance de a potência do sinal cair abaixo de um valor de referência θ , a partir do qual o funcionamento será prejudicado. Em um canal livre de interferência, define-se a PI como

$$P_{OP} = F_W(\theta). \quad (4.32)$$

Agora, será analisado o impacto da interferência de canal adjacente na PI. Primeiramente, admite-se que N sinais com distribuição Nakagami e estáticos são independentes e identicamente distribuídos (i.i.d). A FDP do sinal interferente resultante será dada por [17, eq. (16)]

$$f_P(p) = \left(\frac{m_{If}}{\Omega_{If}} \right)^{m_{If}N} \frac{p^{m_{If}N-1}}{\Gamma(m_{If}N)} \exp\left(-\frac{m_{If}p}{\Omega_{If}}\right), \quad (4.33)$$

sendo m_{If} o fator de desvanecimento para o canal interferente e Ω_{If} a potência média de cada sinal. Para incluir o fenômeno do sombreamento, novamente, condiciona-se a potência P em (4.42) a uma VA com distribuição Gama de parâmetro de desvanecimento m_{Is} e potência média Ω_{Is} . Seguindo o mesmo procedimento de (4.11) e utilizando de [62, eq. (3.471.9)] e [61, eq. (8.4.23.1)], a FDP do sinal interferente Nakagami com sombreamento Gama é dada por

$$f_P(p) = \frac{1}{p\Gamma(m_{If}N)\Gamma(m_{Is})} G_{0,2}^{2,0} \left(\frac{m_{If}p}{\Omega_{Is}} \middle| \begin{matrix} - \\ m_{Is}, m_{If}N \end{matrix} \right). \quad (4.34)$$

Admitindo que o sinal interferente seja dominante, o efeito do ruído passa a ser negligenciável. Então, a relação sinal-ruído-interferência (RSRI) será aproximada por meio da equivalente relação sinal-interferência (RSI) dada pela VA

$$\Upsilon = \frac{W}{P}. \quad (4.35)$$

A FDP de Υ pode ser obtida por meio da FDP de W e P utilizando-se de [42, eq. (6.60)]

$$f_\Upsilon(\gamma) = \int_0^\infty p f_W(\gamma p) f_P(p) dp. \quad (4.36)$$

Substituindo (4.18) e (4.34) em (4.36) e utilizando-se de [61, eq. (2.25.1.1)] obtém-se

$$f_{\Upsilon}(\gamma) = \sum_{i=1}^u \frac{B_i H_{3,3}^{2,3} \left(\frac{\mu \frac{2}{\alpha} \gamma}{m_{\text{If}} \hat{\gamma}} \middle| \begin{matrix} (1-m_{\text{Is}},1), (1-m_{\text{If}},1), (\frac{\delta-\beta_i-1}{\delta},1) \\ (m_s,1), (\mu, \frac{2}{\alpha}), (-\frac{\beta_i+1}{\delta},1) \end{matrix} \right)}{\gamma \delta \Gamma(\mu) \Gamma(m_s) \Gamma(m_{\text{If}} N) \Gamma(m_{\text{Is}})}, \quad (4.37)$$

em que

$$\hat{\gamma} = \frac{\Omega_0}{\Omega_{\text{Is}}} \quad (4.38)$$

é a RSI média na distância máxima de cobertura D_0 . A FDC de Υ é novamente encontrada com o auxílio de [57, eq. (2.53)], sendo

$$F_{\Upsilon}(\gamma) = \sum_{i=1}^u \frac{B_i H_{4,4}^{2,4} \left(\frac{\mu \frac{2}{\alpha} \gamma}{m_{\text{If}} \hat{\gamma}} \middle| \begin{matrix} (1,1), (1-m_{\text{Is}},1), (1-m_{\text{If}},1), (\frac{\delta-\beta_i-1}{\delta},1) \\ (m_s,1), (\mu, \frac{2}{\alpha}), (-\frac{\beta_i+1}{\delta},1), (0,1) \end{matrix} \right)}{\delta \Gamma(\mu) \Gamma(m_s) \Gamma(m_{\text{If}} N) \Gamma(m_{\text{Is}})}. \quad (4.39)$$

Também, pode-se obter o caso particular em que o sinal desejado tem distribuição Nakagami/Gama fazendo $\alpha = 2$ e $\mu = m_{\text{f}}$, assim obtendo

$$F_{\Upsilon}(\gamma) = \sum_{i=1}^u \frac{B_i G_{4,4}^{2,4} \left(\frac{m_{\text{f}} \gamma}{m_{\text{If}} \hat{\gamma}} \middle| \begin{matrix} 1, 1-m_{\text{Is}}, 1-m_{\text{If}} N, \frac{\delta-\beta_i-1}{\delta} \\ m_s, m_{\text{f}}, -\frac{\beta_i+1}{\delta}, 0 \end{matrix} \right)}{\delta \Gamma(m_{\text{f}}) \Gamma(m_s) \Gamma(m_{\text{If}} N) \Gamma(m_{\text{Is}})}. \quad (4.40)$$

Então a PI na presença de um canal ajacente causador de interferência será encontrada por

$$P_{\text{OP}} = F_{\Upsilon}(\theta). \quad (4.41)$$

4.3.3 Capacidade

A capacidade média de um canal normalizada pela banda é expressa por [37, eq. (4.4)]

$$C = \int_0^{\infty} \log_2(w+1) f_W(w) dw. \quad (4.42)$$

Para a resolução de (4.42) é necessário realizar a mudança da base logarítmica

$$\log_2(w+1) = \frac{\ln(1+w)}{\ln(2)} \quad (4.43)$$

e, em seguida, a transformação [61, eq. (8.4.6.5)]

$$\ln(1+w) = G_{2,2}^{1,2} \left(w \middle| \begin{matrix} 1,1 \\ 1,0 \end{matrix} \right) = H_{2,2}^{1,2} \left(w \middle| \begin{matrix} (1,1), (1,1) \\ (1,1), (0,1) \end{matrix} \right), \quad (4.44)$$

assim, após substituir (4.18) em (4.42) e utilizando-se de [61, eq. (2.25.1.1)], encontra-se a capacidade do canal afetado pelo desvanecimento α - μ /Gama com mobilidade RWP como

$$C = \frac{1}{\delta\Gamma(\mu)\Gamma(m_s)\ln(2)} \sum_{i=1}^u B_i H_{5,3}^{2,4} \left(\frac{\Omega_0}{\mu^{\frac{2}{\alpha}}} \middle| \begin{matrix} (1,1),(1,1),(1-m_s,1),(1-\mu,\frac{2}{\alpha}),(\frac{\delta+\beta_i+1}{\delta},1) \\ (1,1),(\frac{\beta_i+1}{\delta},1),(0,1) \end{matrix} \right). \quad (4.45)$$

Consequentemente, fazendo $\alpha = 2$ e $\mu = m_f$, a capacidade para o desvanecimento Nakagami/Gama é encontrada como

$$C = \frac{1}{\delta\Gamma(m_f)\Gamma(m_s)\ln(2)} \sum_{i=1}^u B_i G_{3,5}^{4,2} \left(\frac{m_f}{\Omega_0} \middle| \begin{matrix} 0,1-\frac{\beta_i+1}{\delta},1 \\ 0,0,m_s,m_f,-\frac{\beta_i+1}{\delta} \end{matrix} \right). \quad (4.46)$$

4.4 Resultados

De início, apresentam-se na Figura 4.1 a FDP da distribuição α - μ /Gama com deslocamento aleatório RWP descrita por (4.18) e o histograma normalizado para validação dos resultados. Utilizam-se os seguintes parâmetros: $\alpha = 3.1$, $\mu = 2.7$, $m_s = 1.8$ e $\delta = 3$ para $\Omega_0 = 1$ e um modelo de deslocamento unidimensional. Como observado, as barras verticais coincidem com a curva teórica. A avaliação do canal para diferentes parâmetros será feita conjuntamente com as expressões de desempenho, uma vez que, se elas forem válidas, consequentemente, as distribuições do canal também serão. O processo de geração das amostras de mobilidade é detalhado no Apêndice II, as distribuições Gama e α - μ são geradas a partir de funções já existentes para a variável Gama. O algoritmo para o cálculo da função H de Fox está disponível no Anexo B e os códigos para computar as funções estão dispostos no Apêndice III.

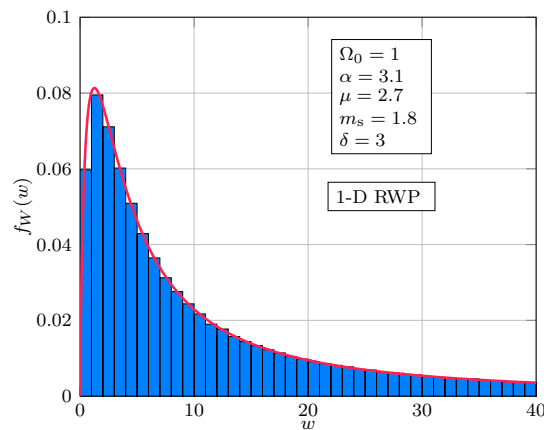


Figura 4.1: FDP e histograma da potência de um sinal sob desvanecimento α - μ /Gama com deslocamento aleatório RWP.

Para a análise do modelo proposto, na Figura 4.2 variam-se os principais parâmetros

em relação à FDP. Para todos eles foi adotado o modelo de deslocamento unidimensional e potência unitária na máxima distância de cobertura. Na Figura 4.2(a) admitem-se os valores do fator de não linearidade como sendo $\alpha = \{1, 2, 3.4, 4.2\}$. Na Figura 4.2(b) admite-se o parâmetro de desvanecimento como sendo $\mu = \{0.5, 1.0, 1.5, 2.5\}$. Na Figura 4.2(c) admite-se o parâmetro de sombreamento $m_s = \{0.5, 1.8, 2.6, 4.8\}$. Por fim, Na Figura 4.2(d) admite-se o expoente de perda como sendo $\delta = \{2, 2.5, 3.5, 4\}$. Os valores dos demais parâmetros estão indicados para cada caso.

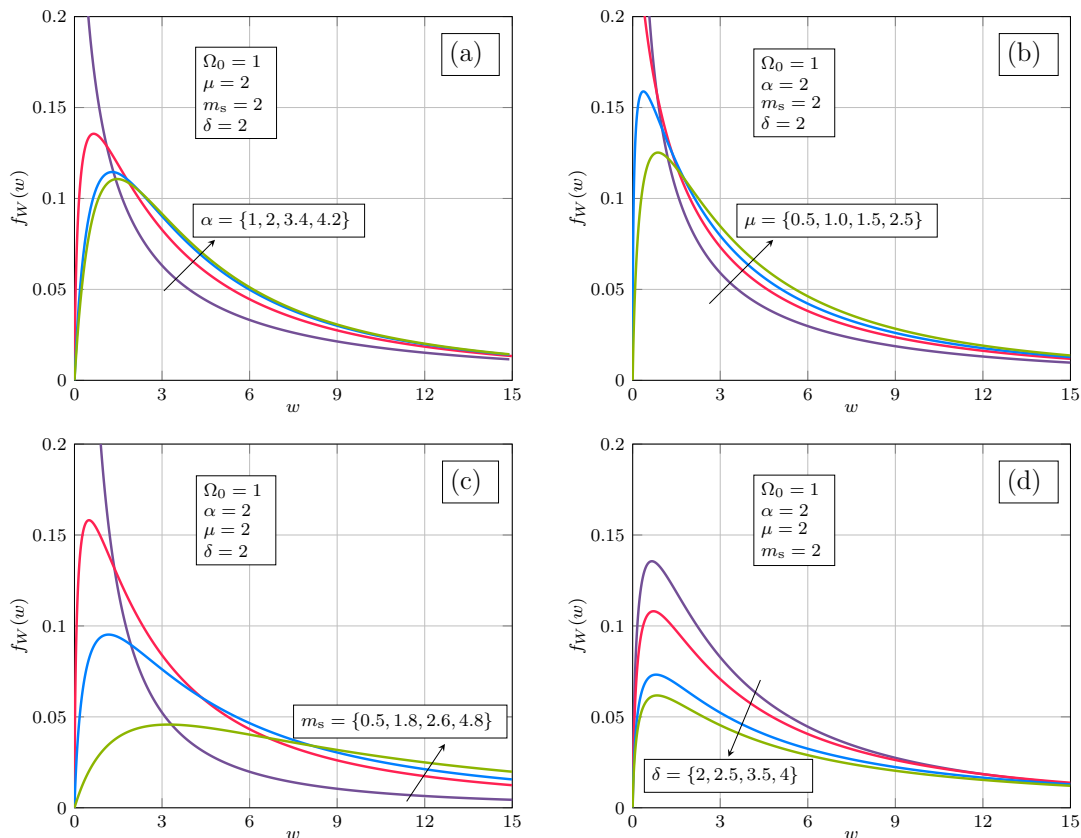


Figura 4.2: Comportamento da FDP da potência de um sinal sob desvanecimento α - μ /Gama com deslocamento aleatório RWP.

Observa-se que quanto maior os valores dos parâmetros α , μ e m_s , com os demais parâmetros fixos, as curvas deslocam-se mais para a direita. Isto significa que menor é a probabilidade de se encontrar valores próximos de zero do sinal sob desvanecimento. Consequentemente, o canal torna-se menos severo em termos de desvanecimento. O mesmo comportamento é esperado com o decréscimo do parâmetro δ , porém, para o cenário em questão, observa-se que o sistema teria um melhor desempenho com maiores valores de δ . Esclarecendo: para uma potência na máxima distância de cobertura $\Omega_0 = 1$ a potência de transmissão será $P_{Tx} = D_0^\delta$, de modo que, a maior atenuação representará uma maior potência média na região de cobertura.

Agora, será averiguado o impacto das diferentes topologias de deslocamento aleatório empregadas sobre o sinal. Para isso, utilizam-se os seguintes parâmetros, $\Omega_0 = 1$, $\alpha = 2.2$, $\mu = 1.8$, $m_s = 2.3$ e $\delta = 2$ para gerar as FDCs dos modelos uni, bi e tridimensional em (4.20). Os resultados são dispostos na Figura 4.3. O modelo tridimensional é o mais severo e isso é demonstrado pelo rápido crescimento da distribuição, em outras palavras, há uma maior concentração da potência ao redor de valores pequenos. Teoricamente, a diferença entre os modelos bidimensional e unidimensional é pequena, desta forma, pode-se economizar recursos ao modelar os casos mais simples para 1-D e 3-D em cenários nos quais os deslocamentos de elevação sejam suficientemente relevantes.

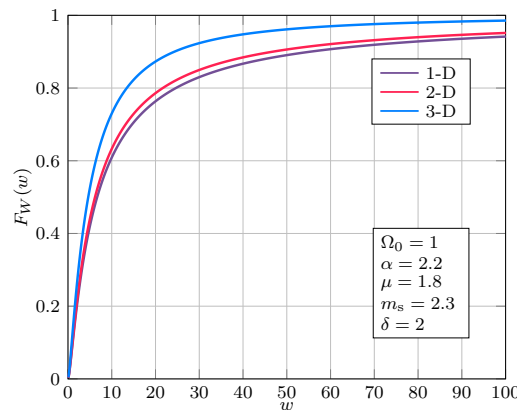


Figura 4.3: Comportamento da FDC da potência de um sinal sob desvanecimento α - μ /Gama com deslocamento aleatório RWP mediante variação das dimensões.

A seguir, serão discutidos resultados referentes às métricas de desempenho. Em todos os gráficos, as linha cheias, tracejadas e pontilhadas representam valores teóricos. Os marcadores representam resultados numéricos obtidos por simulação. A não ser que seja indicado o contrário, as simulações se referem a um sistema BPSK e com a distância máxima de movimento $D_0 = 50\text{m}$.

A Figura 4.4 demonstra a variação da ASEP em relação à potência de transmissão para diversos parâmetros de sistema buscado avaliar diferentes cenários de desvanecimento. A potência de transmissão está expressa aqui em dB, o que significa uma potência de transmissão em relação a uma potência de ruído fixa de 1 Watt. As retas verdes pontilhadas representam os valores assintóticos calculados a partir da equações (4.27) e (4.29).

É importante notar que as expressões deduzidas são precisas, uma vez que há correspondência entre os valores teóricos e os pontos simulados. Também é possível verificar a convergência das expressões assintóticas para elevados valores de RSR. Portanto, também reforçam os resultados deduzidos. Quanto maior a distância entre

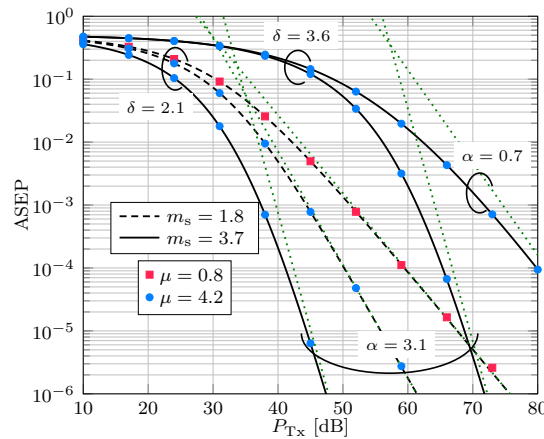


Figura 4.4: ASEP em função da potência de transmissão para diferentes parâmetros de desvanecimento.

os valores de $\alpha\mu/2$ e m_s , as expressões assintóticas convergem mais rapidamente para os valores exatos em situações de baixa RSR. Isso ocorre devido ao fato da restrição $m_s \neq \alpha\mu/2$ em (4.27) e (4.29). Para todos os casos, as assintotas como limitante superior. Como esperado, há um melhor desempenho do sistema para menores valores do expoente de perda de percurso δ , uma vez que a potência média na máxima cobertura será maior.

Na Figura 4.5 é fixado o parâmetro $\alpha = 1.5$ para demonstrar a variação da ASEP em diferentes cenários de desvanecimento e mobilidade, porém, agora com o foco na análise dos ganhos de diversidade e codificação.

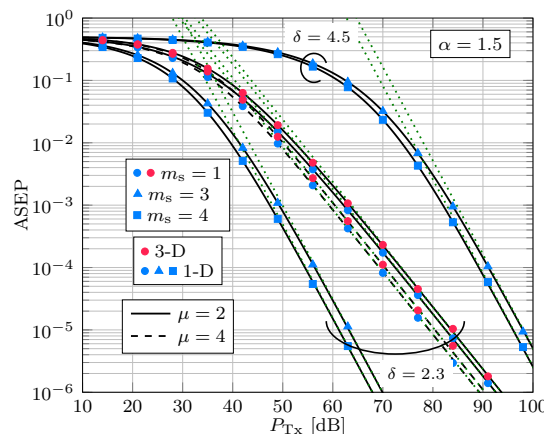


Figura 4.5: ASEP para análise dos ganhos de diversidade e codificação.

Para um mesmo cenário de mobilidade, isto é, mesma dimensão de deslocamento e expoente de perda de percurso, quando o ganho de diversidade é igual a $\mathcal{G}_d = \alpha\mu/2 = 1.5$ as curvas para $m_s = 3$ e $m_s = 4$ decaem com mesma inclinação em elevada RSR. Desta forma, a ordem de diversidade, baseada na ASEP, depende somente da não li-

nearidade do canal e do número de *clusters*, assim como indicado pelos resultados teóricos. Elevados valores do parâmetro de sombreamento m_s , i.e. menor severidade, resultam em uma melhora no desempenho do sistema, i.e. menor ASEP, devido ao fato de o ganho de codificação aumentar com m_s . Por outro lado, quando o ganho de diversidade é igual a $\mathcal{G}_d = m_s = 1$ as curvas para $\mu = 2$ e $\mu = 4$ têm a mesma inclinação para uma potência de transmissão elevada, resultando que, para a condição em que $m_s < \alpha\mu/2$ o ganho de diversidade depende apenas do parâmetro de sombreamento m_s . Além do mais, os resultados das curvas assintóticas demonstram que a ASEP em um cenário de mobilidade é degradada com o aumento da dimensionalidade de rede e também com o incremento do coeficiente de perda de percurso, como já discutido anteriormente. Portanto, o canho de codificação diminui com o aumento da dimensionalidade e atenuação provocada pelo canal.

A PI, para cenários com e sem interferência, é apresentada na Figura 4.6. Os parâmetros do canal são $\alpha = 1.1$, $\mu = m_{If} = 2.3$, $\Omega_{Is} = 15\text{dBW}$ e $\delta = 3.1$. O limiar é admitido como sendo $\theta = 4$. De maneira generalista, pode-se admitir que o sombreamento que afeta os sinais desejado e interferente têm mesmas características, porém são independentes, nesse caso propõe-se dois cenários, $m_s = m_{Is} = 1.8$ e $m_s = m_{Is} = 3.9$.

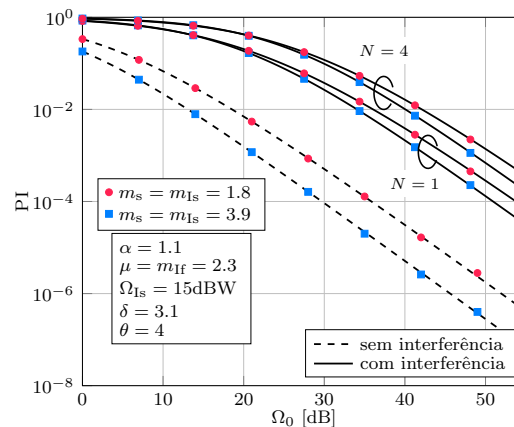


Figura 4.6: *PI para cenários com e sem canal interferente.*

Tomando como base o caso ótimo, em que não há interferência, a presença de um único canal interferente $N = 1$ é suficiente para degradar o desempenho do sistema. Quando o número de canais interferentes cresce para $N = 4$, o desempenho é impactado, porém, relativamente menos do que o primeiro caso. A sensibilidade do sistema, nesse contexto, classifica-o como limitado por interferência.

O gráfico da capacidade média normalizada em relação à largura de banda do canal em função de Ω_0 é apresentado na Figura 4.7. São fixados os valores de $\alpha = 1.8$ e

$m_s = 1.8$ e varia-se $\mu = \{0.7, 1.2, 4.8\}$. Com o incremento do parâmetro de desvanecimento de $\mu = 0.7$, severo, para $\mu = 4.8$, brando, a capacidade do canal aumenta. Novamente, o melhor desempenho é observado para uma maior atenuação do canal devido à normalização em relação à potência na máxima distância de cobertura.

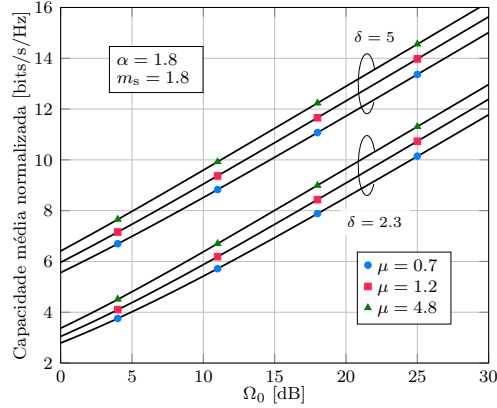


Figura 4.7: Capacidade média em função da potência média na máxima distância de cobertura.

4.5 Autoencoder em Canal com Desvanecimento e Mobilidade

A implementação do AE é feita tal qual descrita no Capítulo 3. O sinal transmitido é afetado pelo desvanecimento α - μ com sombreamento Gama e deslocamento RWP dada por (4.18). O sinal recebido é contaminado por um ruído AWGN e normalizado em relação à potência de recepção. Assim, cada amostra que modela o sinal transmitido é gerado de acordo com a seguinte expressão

$$y_i = (\sqrt{w_i}x_i + z_i)/\sqrt{w_i}, \quad (4.47)$$

para $i = 1, 2, \dots, 2n$, em que z_i é uma realização da variável aleatória Z_i com distribuição Gaussiana e variância $\sigma^2 = (2E/N_0)^{-1}$ e i.i.d sobre as $2n$ dimensões de $\mathbf{Z} \in \mathbb{R}^{2n}$, w_i é uma realização da variável aleatória W_i com distribuição dada por (4.18), também i.i.d sobre as $2n$ dimensões de $\mathbf{W} \in \mathbb{R}^{2n}$, x_i é a i -ésima componente dos símbolos gerados pela função de codificação $\mathbf{X} \in \mathbb{R}^{2n}$ e, por fim, y_i é a i -ésima componente dos símbolos recebidos $\mathbf{Y} \in \mathbb{R}^{2n}$. A energia média dos símbolos é unitária, i.e. $E = 1$, de modo que a RSR é dada pelo valor da densidade espectral de potência unilateral do ruído, i.e. N_0 .

Os parâmetros de treinamento do canal são $\alpha = 2.1$, $\mu = 1.8$, $m_s = 1.3$, $\delta =$

2. A distância máxima é $D_0 = 75$ m, a dimensão de deslocamento utilizada é a unidimensional, já que possui menor complexidade computacional, e $E/N_0 = 45$ dB. $\eta = 0.001$ e $S_b = 500$ mensagens para 500 passos e 10 épocas. Detalhes da implementação estão dispostos no Apêndice I.

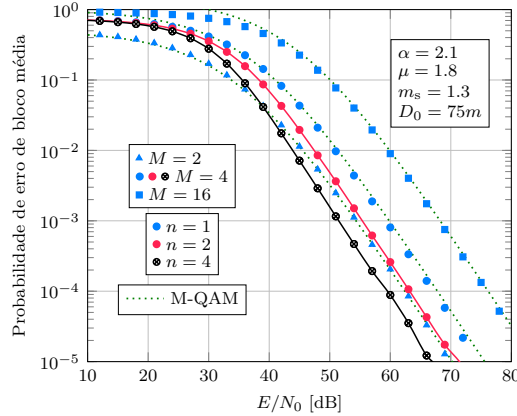


Figura 4.8: Análise da Probabilidade de Erro de bloco do Autoencoder para diferentes configurações.

A seguir, será feita a análise de desempenho do AE tomando para isso a probabilidade de erro de símbolo na Figura 4.8. As curvas teóricas são dadas pelas linhas pontilhadas para cada esquema de modulação M -QAM, sendo $M = \{2, 4, 6\}$, para o AE, cada utilização complexa do canal recebe uma cor diferente, sendo elas azul, vermelha e preta para os respectivos valores de utilização $n = \{1, 2, 4\}$. Ao se comparar as curvas do AE com as curvas teóricas para as modulações M -QAM, percebe-se que o desempenho do AE é similar à técnica tradicional para os casos em que a utilização complexa do canal é unitária, i.e. $n = 1$. Com $M = 4$ símbolos, fazer o uso complexo de canal $n = 2$ resulta em um melhor desempenho do sistema, com um ganho de aproximadamente de 3 dB quando comparado com o caso em que $n = 1$. Novamente, fazendo $n = 4$ resulta em um ganho de 3 dB quando comparado ao caso em que $n = 2$. A melhoria do desempenho é acompanhando com um decréscimo da taxa de bits por uso do canal R_b .

Também, é importante ressaltar a capacidade de generalização do AE, isto é, como ele desempenha quando atuando fora do cenário para o qual foi treinado. Para isso, mantém-se os mesmos parâmetros de treinamento porém, faz-se $E/N_0 = 65$ dB e são variados, individualmente, os parâmetros α , μ e m_s para $M = 2$ e $n = 1$, os resultados são apresentados na Figura 4.9. Observam-se três casos, o primeiro, mantêm-se fixos $\mu = 1.8$ e $m_s = 1.3$, e varia-se α no intervalo de 0.7 a 3, para gerar a curva com marcadores azuis, o segundo, mantêm-se fixos $\alpha = 2.1$ e $m_s = 1.3$, e varia-se μ no intervalo de 0.7 a 3, para gerar a curva com marcadores vermelhos e , por último, mantêm-se

fixos $\alpha = 2.1$ e $\mu = 1.8$, e varia-se m_s no intervalo de 0.7 a 3, para gerar a curva com marcadores verdes. Nota-se que, dentro os valores analisados, o comportamento do AE se assemelha à variação teórica de um sistema BPSK nas mesmas condições de canal, o que revela a capacidade de generalização do AE em manter a menor probabilidade de erro de símbolo sob diferentes condições de canais em relação aos parâmetros de treinamento.

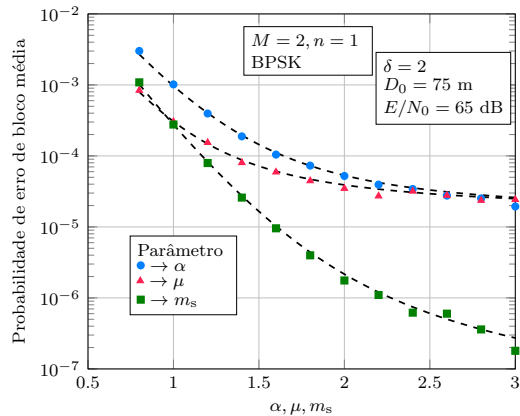


Figura 4.9: Análise da capacidade de generalização do AE para diferentes condições de canal.

Finalizando, há uma ressalva sobre a relação entre a taxa de erro de símbolo e de bit. Não se pode presumir que o mapeamento dos bits em símbolos será sempre ótima, ou seja, buscará a menor quantidade de erros para os vizinhos mais próximos, o que em técnicas tradicionais é feito pelo Código Gray [1]. Portanto, a conversão de bits para símbolos, que para esse caso seria uma mudança de base decimal para binária, não reflete o melhor caso para probabilidade de erro de bit. Para que essa comparação seja feita, é necessário alterar a arquitetura e treinamento do AE para que sejam capazes de tratar os bits de informação.

Capítulo 5

Conclusões e Propostas Para Novas Pesquisas

Neste trabalho deduziu-se a distribuição α - μ /Gama com deslocamento aleatório RWP, um modelo mais generalista do que os anteriormente propostos e expressões para FDP, FDC, ASEP, PI e capacidade. Como foi demonstrado ao longo do trabalho, as expressões e resultados possuem acurácia quando comparados aos dados obtidos por simulações. Adequando-se os parâmetros pode-se cobrir uma grande variedade de cenários. É promissora em se tratando de projetos para redes como 5G e 6G que têm como desafio a heterogeneidade de dispositivos conectados. Um modelo de deslocamento 2-D pode, por exemplo, ser usado para a modelagem comportamento de uma rede veicular, já o 3-D, para veículos aéreos não tripulados. Outras distribuições podem ser descritas em termos da função H de Fox, como demonstrado em [63]. Existem, portanto, vários outros modelos de desvanecimento aos quais há a possibilidade de incorporar o deslocamento aleatório, seguindo processos similares aos adotados neste trabalho. Vê-se como oportunidade o estudo e validação desses casos.

O AE provou ser sólido e tem desempenho comparável com técnicas já conhecidas e bastante estudadas, que são soluções ótimas, quando analisados os blocos individualmente, mas não necessariamente serão uma solução ótima quando analisado o sistema como um todo, ou seja fim-a-fim. O AE possui uma grande flexibilidade na forma de sua arquitetura e implementação. O presente trabalho, porém, se limitou aos efeitos do canal sobre a envoltória do sinal, portanto, como pesquisa futura, é oportuna a utilização de um modelo de canal que leva em conta a utilização dos sinais a partir de sua definição formal, ou seja, utilizando às funções bases que compõem as componentes *em fase* e *quadratura* do sinal, afim de se analisar os efeitos sobre a fase do sinal recebido. Também, ao gerar as amostras do sinal para o treinamento do AE é

necessária a normalização do sinal devido às diferentes potências de recepção resultadas da mobilidade. Esse processo ajuda o sistema uma vez que se faz a estimação e a equalização do sinal recebido. Há a oportunidade de se propor um sistema baseado em DL que seja capaz de realizar determinada tarefa, podendo ser, mas não necessariamente, um AE.

Outra área a ser abordada é a utilização de canais que contenham correlação temporal, adequando-se o modelo, pode-se extrair mais informações do comportamento do canal e sua comparação com técnicas mais complexas. Também, cenários em que se pode fazer a implementação prática permitem uma comparação mais direta com tecnologias já existentes.

Anexo A

Função H de Fox

A função H, ou popularmente conhecida na literatura como a função H de Fox, pode ser aplicada em uma grande variedade de problemas relacionados à comunicação, equações diferenciais e integrais fracionárias, muitas áreas da física teórica e teoria da distribuição estatística, por exemplo. A seguir, define-se matematicamente a função H de Fox.

Notação A.1.

$$H_{p,q}^{m,n} \left(z \left| \begin{matrix} (a_p, A_p) \\ (b_q, B_q) \end{matrix} \right. \right) = H_{p,q}^{m,n} \left(z \left| \begin{matrix} (a_1, A_1) \dots (a_p, A_p) \\ (b_1, B_1) \dots (b_q, B_q) \end{matrix} \right. \right) : \text{Função H} \quad (\text{A.1})$$

Definição A.1. A Função H de Fox é definida por meio de uma integral do tipo Mellin–Barnes da subseqüente maneira

$$H_{p,q}^{m,n} \left(z \left| \begin{matrix} (a_p, A_p) \\ (b_q, B_q) \end{matrix} \right. \right) = \frac{1}{2\pi i} \int_{\mathcal{L}} \Theta(s) z^{-s} ds, \quad (\text{A.2})$$

em que $i = (-1)^{\frac{1}{2}}$, $z \neq 0$ e $z^{-s} = \exp[-s(\ln|z| + i \arg z)]$, em que $\ln|z|$ é o logaritmo natural de $|z|$ e $\arg z$ não é necessariamente o valor principal do argumento de z , e

$$\Theta(s) = \frac{\{\prod_{j=1}^m \Gamma(b_j + B_j s)\} \{\prod_{k=1}^n \Gamma(1 - a_k - A_k s)\}}{\{\prod_{j=m+1}^q \Gamma(1 - b_j - B_j s)\} \{\prod_{k=n+1}^p \Gamma(a_k + A_k s)\}}. \quad (\text{A.3})$$

Um produto vazio é sempre interpretado como a unidade, $m, n, p, q \in \mathbb{N}_0$ sendo $0 \leq n \leq p$, $1 \leq m \leq q$, $A_k, B_j \in \mathbb{R}_+$, $a_k, b_j \in \mathbb{C}$, $k = 1, \dots, p$ e $j = 1, \dots, q$. \mathcal{L} é um contorno adequado separando os polos

$$\xi_{jv} = - \left(\frac{b_j + v}{B_j} \right), \quad (\text{A.4})$$

das funções Gama $\Gamma(b_j + B_j s)$ para $j = 1, \dots, m$ e $v = 0, 1, 2, \dots$ dos polos

$$\omega_{ku} = - \left(\frac{1 - a_k + u}{A_k} \right), \quad (\text{A.5})$$

das funções Gama $\Gamma(1 - a_k - A_k s)$ para $k = 1, \dots, n$ e $u = 0, 1, 2, \dots$, isto é

$$A_k(b_j + v) \neq B_j(a_k - u - 1), \quad (\text{A.6})$$

para $v = 0, 1, 2, \dots$, $u = 0, 1, 2, \dots$, $j = 1, \dots, m$ e $k = 1, \dots, n$.

O contorno \mathcal{L} existe em razão de (A.6), essas suposições serão mantidas adiante, e é um dos possíveis $\mathcal{L}_{-\infty}$, $\mathcal{L}_{+\infty}$ ou $\mathcal{L}_{i\gamma\infty}$, que são definidos a seguir.

- i $\mathcal{L} = \mathcal{L}_{-\infty}$ é um laço que começa e termina em $-\infty$ envolvendo todos os polos de $\Gamma(b_j + B_j s)$, $j = 1, \dots, m$, na direção positiva, mas nenhum dos polos de $\Gamma(1 - a_k - A_k s)$, $k = 1, \dots, n$. Sendo

$$\beta = \left\{ \prod_{k=1}^p (A_k)^{-A_k} \right\} \left\{ \prod_{j=1}^q (B_j)^{-B_j} \right\} \quad (\text{A.7})$$

$$\mu = \sum_{j=1}^q B_j - \sum_{k=1}^p A_k, \text{ e} \quad (\text{A.8})$$

$$\delta = \sum_{j=1}^q b_j - \sum_{k=1}^p a_k + \frac{p - q}{2}, \quad (\text{A.9})$$

A integral converge para todo z se $\mu > 0$ e $z \neq 0$, ou $\mu = 0$ e $0 < |z| < \beta$. A integral também converge se

$$\mu = 0, |z| = \beta \text{ e } \text{Re}(\delta) < -1, \quad (\text{A.10})$$

em que $\text{Re}(\cdot)$ denota a parte real de um número complexo.

- ii $\mathcal{L} = \mathcal{L}_{+\infty}$ é um laço começando e terminando em $\mathcal{L}_{+\infty}$ envolvendo todos os polos de $\Gamma(1 - a_k - A_k s)$, $k = 1, \dots, n$, na direção negativa, mas nenhum dos polos de $\Gamma(b_j + B_j s)$, $j = 1, \dots, m$. A integral converge para todo z se

$$\mu \leq 0 \text{ e } z \neq 0 \text{ ou } \mu = 0 \text{ e } |z| > \beta. \quad (\text{A.11})$$

A integral também converge se todas as condições (A.10) forem atendidas.

- iii $\mathcal{L}_{i\gamma\infty}$ é um contorno começando no ponto $\gamma - i\infty$ e indo para $\gamma + i\infty$, em que $\gamma \in \mathbb{R} = (-\infty, +\infty)$ tal qual todos os polos de $\Gamma(b_j + B_j s)$, $j = 1, \dots, m$,

estão separados dos de $\Gamma(1 - a_k - A_k s)$, $k = 1, \dots, n$. A integral converge se

$$\alpha > 0, |\arg z| \leq \frac{1}{2}\pi\alpha, a_k \neq 0. \quad (\text{A.12})$$

Sendo

$$\alpha = \sum_{k=1}^n A_k - \sum_{k=n+1}^p A_k + \sum_{j=1}^m B_j - \sum_{j=m+1}^q B_j, \quad (\text{A.13})$$

A integral também converge se $\alpha = 0$, $\gamma\mu + \text{Re}(\delta) < -1$, $\arg z = 0$ e $z \neq 0$.

As condições de existência da função H são enumeradas a seguir. Presume-se que a condição (A.6) seja satisfeita.

Teorema A.1. *A função H de Fox é uma função analítica de z e existe diante dos casos enumerados:*

1. $q \geq 1, \mu > 0$, Função H existe para todo $z \neq 0$;
2. $q \geq 1, \mu = 0$, Função H existe para $0 < |z| < \beta$;
3. $q \geq 1, \mu = 0, \text{Re}(\delta) < -1$ Função H existe para $|z| = \beta$;
4. $p \geq 1, \mu < 0$, Função H existe para todo $z \neq 0$;
5. $p \geq 1, \mu = 0$, Função H existe para $|z| > \beta$;
6. $p \geq 1, \mu = 0, \text{Re}(\delta) < -1$, Função H existe para $|z| = \beta$;
7. $\alpha > 0, |\arg z| \leq \frac{1}{2}\pi\alpha$, Função H existe para todo $z \neq 0$;
8. $\alpha = 0, \gamma\mu + \text{Re}(\delta) < -1$, Função H existe para $\arg z = 0$ e $z \neq 0$.

A.1 Função G de Meijer: Caso Especial

Notação A.2.

$$G_{p,q}^{m,n} \left(z \left| \begin{matrix} a_p \\ b_q \end{matrix} \right. \right) = G_{p,q}^{m,n} \left(z \left| \begin{matrix} a_1 \dots a_p \\ b_1 \dots b_q \end{matrix} \right. \right) : \text{Função G} \quad (\text{A.14})$$

Definição A.2.

$$G_{p,q}^{m,n} \left(z \left| \begin{matrix} a_p \\ b_q \end{matrix} \right. \right) = \frac{1}{2\pi i} \int_{\mathcal{L}} \frac{\{\prod_{j=1}^m \Gamma(b_j + s)\} \{\prod_{k=1}^n \Gamma(1 - a_k - s)\}}{\{\prod_{j=m+1}^q \Gamma(1 - b_j - s)\} \{\prod_{k=n+1}^p \Gamma(a_k + s)\}} z^{-s} ds, \quad (\text{A.15})$$

Em que, $0 \leq m \leq q$, $0 \leq n \leq p$, a_k para $k = 1, \dots, p$ e b_j para $j = 1, \dots, q$ são números complexos e tal que

$$a_k - b_j \neq 0, 1, \dots, \quad (\text{A.16})$$

para $j = 1, \dots, n$ e $k = 1, \dots, m$. Os parâmetros são tais que os pontos

$$s = -(b_j + v), \quad (\text{A.17})$$

para $j = 1, \dots, n$ e $v \in \mathbb{N}_0$, e

$$s = -(a_k - v - 1), \quad (\text{A.18})$$

para $k = 1, \dots, m$ e $v \in \mathbb{N}_0$ estão separados. Aqui a linha \mathcal{L} é o mesmo contorno do que foi tomado para a função H de Fox. Note, que a Função G surge fazendo os elementos A_p e B_q unitários em (A.2). Para uma estudo mais detalhada recomenda-se [57], do qual se extraiu esse excerto.

Anexo B

Algoritmo para cálculo da função H de Fox

Um simples algoritmo para o cálculo da função H de Fox proposto por [64] em linguagem MATLAB [65].

```
1 function Fox_H_value=func_Fox_H_univariate(z, O, P)
2 % parameters defination here
3 length=20;
4 leave_from_real_axis=0.05;
5 % disamble the O and P
6 m=O(1);
7 n=O(2);
8 p=O(3);
9 q=O(4);
10
11 k=P{1};
12 c=P{2};
13 a=P{3};
14 b=P{4};
15 A=P{5};
16 B=P{6};
17
18 Fox_H_value_temp=real(integral(@(s)mpq_part_generation(c*z,s,m,n,
19     p,q,a,b,A,B), leave_from_real_axis-li*length,
20     leave_from_real_axis+li*length, 'AbsTol', 1e-7, 'RelTol', 1.e-7, '
21     Waypoints', true)/(2*pi*li));
22
23 Fox_H_value=Fox_H_value_temp*k;
24 end
25 function mpq_part=mpq_part_generation(z,s,m,n,p,q,a,b,A,B)
26 % initialization for the integration theta(s)
```

```
24 m_part= 1;
25 n_part= 1;
26 p_part= 1;
27 q_part= 1;
28 % create the integration theta(s)
29 for iii=1:1:m
30     m_part= m_part.*gammaz(b(iii)-B(iii).*s);
31 end
32 for iii=1:1:n
33     n_part= n_part.*gammaz(1-a(iii)+A(iii).*s);
34 end
35 for iii=n+1:1:p
36     p_part= p_part.*gammaz(a(iii)-A(iii).*s);
37 end
38 for iii=m+1:1:q
39     q_part= q_part.*gammaz(1-b(iii)+B(iii).*s);
40 end
41 mnpq_part=z.^s.*m_part.*n_part./p_part./q_part;
42 end
```

Referências Bibliográficas

- [1] S. Haykin, *Digital Communication Systems*. Wiley, 2013.
- [2] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, “The road towards 6G: A comprehensive survey,” *IEEE Open Journal of the Communications Society*, vol. 2, pp. 334–366, 2021.
- [3] C. Zhang, P. Patras, and H. Haddadi, “Deep learning in mobile and wireless networking: A survey,” *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, 2019.
- [4] L. Rayleigh, “On the resultant of a large number of vibrations of the same pitch and of arbitrary phase,” *Philos. Mag.*, no. 60, pp. 73–78, Aug. 1880.
- [5] S. O. Rice, “Statistical properties of a sine wave plus random noise,” *Bell Syst. Tech. J.*, vol. 27, no. 1, pp. 109–157, Jan. 1948.
- [6] M. Nakagami, “The m -distribution—A general formula of intensity distribution of rapid fading,” *Statistical Methods in Radio Wave Propagation*, vol. 26, no. 2, pp. 3–36, Apr. 1960.
- [7] M. D. Yacoub, “The κ - μ distribution and the η - μ distribution,” *IEEE Trans. Antennas Propag.*, vol. 49, no. 1, pp. 68–81, Feb. 2007.
- [8] ———, “The α - μ distribution: A physical fading model for the Stacy distribution,” *IEEE Trans. Veh. Technol.*, vol. 56, no. 1, pp. 27–34, Jan. 2007.
- [9] E. N. Papasotiriou, A.-A. A. Boulogeorgos, K. Haneda, M. F. de Guzman, and A. Alexiou, “An experimentally validated fading model for thz wireless systems,” *Scientific Reports*, vol. 11, no. 1, pp. 1–14, 2021.
- [10] F. Hansen and F. I. Meno, “Mobile fading—Rayleigh and Lognormal superimposed,” *IEEE Trans. Veh. Technol.*, vol. 26, no. 4, pp. 332–335, Nov. 1977.
- [11] H. Suzuki, “A statistical model for urban radio propagation,” *IEEE Trans. Commun.*, vol. 25, no. 7, pp. 673–680, Jul. 1977.
- [12] T. T. Tjhung and C. C. Chai, “Fade statistics in Nakagami-lognormal channels,”

- IEEE Trans. Commun.*, vol. 47, no. 12, pp. 1769–1772, Dec. 1999.
- [13] P. Bithas, N. Sagias, P. Mathiopoulos, G. Karagiannidis, and A. Rontogiannis, “On the performance analysis of digital communications over generalized- k fading channels,” *IEEE Commun. Lett.*, vol. 10, no. 5, pp. 353–355, 2006.
- [14] P. M. Shankar, “Outage probabilities in shadowed fading channels using a compound statistical model,” *IEE Proc. Commun.*, vol. 152, no. 6, pp. 828–832, Dec. 2005.
- [15] K. Govindan, K. Zeng, and P. Mohapatra, “Probability density of the received power in mobile networks,” *IEEE Trans. Wireless Commun.*, vol. 10, no. 11, pp. 3613–3619, Nov. 2011.
- [16] C. Li *et al.*, “Effect of mobile wireless on outage and BER performances over Rician fading channel,” *IEEE Access*, vol. 8, pp. 91 799–91 806, May. 2020.
- [17] V. A. Aalo, C. Mukasa, and G. P. Efthymoglou, “Effect of mobility on the outage and BER performances of digital transmissions over Nakagami- m fading channels,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2715–2721, Apr. 2016.
- [18] V. Aalo *et al.*, “On the impact of user mobility on the performance of wireless receivers,” *IEEE Access*, vol. 8, pp. 197 300–197 311, Oct. 2020.
- [19] M. Das, B. Sahu, and U. Bhanja, “Analysis of the effects of diversity on mobile wireless networks in a Nakagami fading channel,” *Physical Commun.*, vol. 39, p. 101031, Apr. 2020.
- [20] E. Meesa-ard and S. Pattaramalai, “Analyzing the impact of mobility over α - μ generalized fading channels in wireless communication,” in *2018 3rd Int. Conf. on Comput. and Commun. Syst. (ICCCS)*, Apr. 2018, pp. 318–322.
- [21] A. Al-Rimawi, J. Siam, and A. Abdo, “The outage probability of mobile wireless networks over η - μ fading channel,” in *2018 IEEE Middle East and North Africa Commun. Conf. (MENACOMM)*, Apr. 2018, pp. 1–5.
- [22] E. Meesa-Ard and S. Pattaramalai, “Evaluating the mobility impact on the performance of heterogeneous wireless networks over η - μ fading channels,” *IEEE Access*, vol. 9, pp. 65 017–65 032, Apr. 2021.
- [23] E. Meesa-Ard, S. Pattaramalai, and M. D. C. Madapatha, “Evaluating the impact of mobility over κ - μ generalized fading channels in digital communication,” in *2018 8th Int. Conf. on Electron. Inf. and Emerg. Commun. (ICEIEC)*, Jun. 2018.
- [24] E. Meesa-ard, S. Pattaramalai, and M. D. C. Madapatha, “Outage probability of mobility incorporated α - μ fading distribution with co-channel interference in heterogeneous networks,” in *2018 IEEE Int. Conf. on Smart Internet of Things*

- (*SmartIoT*), Aug. 2018, pp. 76–80.
- [25] F. T. Al Rabee, A. Al-Rimawi, and R. D. Gitlin, “Channel capacity in a dynamic random waypoint mobility model,” in *2018 9th IEEE Annual Ubiquitous Comp., Elect. Mob. Commun. Conf. (UEMCON)*, Nov. 2018, pp. 711–715.
- [26] Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang, “Deep learning in physical layer communications,” *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 93–99, 2019.
- [27] T. J. O’Shea, K. Karra, and T. C. Clancy, “Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention,” in *2016 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2016, pp. 223–228.
- [28] T. O’Shea and J. Hoydis, “An introduction to deep learning for the physical layer,” *IEEE Trans. on Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, 2017.
- [29] S. Dörner, S. Cammerer, J. Hoydis, and S. t. Brink, “Deep learning based communication over the air,” *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, 2018.
- [30] N. Wu, X. Wang, B. Lin, and K. Zhang, “A CNN-based end-to-end learning framework toward intelligent communication systems,” *IEEE Access*, vol. 7, pp. 110 197–110 204, 2019.
- [31] S. Gomes and M. Yacoub, “CNN-based learning system in a generalized fading environment,” in *XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, 01 2020.
- [32] R. Fritschek, R. F. Schaefer, and G. Wunder, “Deep learning for channel coding via neural mutual information estimation,” in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2019, pp. 1–5.
- [33] R. Fritschek, R. F. Schaefer, and G. Wunder, “Neural mutual information estimation for channel coding: State-of-the-art estimators, analysis, and performance comparison,” in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2020, pp. 1–5.
- [34] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, “Mutual information neural estimation,” in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 531–540. [Online]. Available: <https://proceedings.mlr.press/v80/belghazi18a.html>

- [35] H. Ye, G. Y. Li, B.-H. F. Juang, and K. Sivanesan, “Channel agnostic end-to-end learning based communication systems with conditional gan,” in *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2018, pp. 1–5.
- [36] T. J. O’Shea, T. Roy, N. West, and B. C. Hilburn, “Physical layer communications system design over-the-air using adversarial networks,” in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 529–532.
- [37] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.
- [38] R. Heath, *Introduction to Wireless Digital Communication: A Signal Processing Perspective*, ser. Prentice Hall communications engineering and emerging technologies series. Prentice Hall, 2017.
- [39] P. M. Shankar, *Fading and shadowing in wireless systems*. Springer, 2017.
- [40] J. Proakis and M. Salehi, *Digital Communications*. McGraw-Hill, 2008.
- [41] W. Braun and U. Dersch, “A physical mobile radio channel model,” *IEEE Transactions on Vehicular Technology*, vol. 40, no. 2, pp. 472–482, 1991.
- [42] A. Papoulis and S. Pillai, *Probability, Random Variables, and Stochastic Processes*, ser. McGraw-Hill series in electrical engineering: Communications and signal processing. McGraw-Hill, 2002.
- [43] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*. New York, NY, USA: Dover, 1972.
- [44] N. Sagias, D. Zogas, G. Karagiannidis, and G. Tombras, “Channel capacity and second-order statistics in weibull fading,” *IEEE Commun. Lett.*, vol. 8, no. 6, pp. 377–379, 2004.
- [45] H. Hashemi, “The indoor radio propagation channel,” *IEE Proceedings*, vol. 81, no. 7, pp. 943–968, 1993.
- [46] I. Kostic, “Analytical approach to performance analysis for channel subject to shadowing and fading,” *Communications, IEE Proceedings*, vol. 152, pp. 821 – 827, 01 2006.
- [47] A. Abdi and M. Kaveh, “On the utility of gamma pdf in modeling shadow fading (slow fading),” in *1999 IEEE 49th Vehicular Technology Conference (Cat. No.99CH36363)*, vol. 3, 1999, pp. 2308–2312 vol.3.
- [48] S. K. Yoo, S. L. Cotton, L. Zhang, and P. C. Sofotasios, “The inverse gamma distribution: A new shadowing model,” in *2019 8th Asia-Pacific Conference on Antennas and Propagation (APCAP)*, 2019, pp. 475–476.

- [49] J. B. T. Camp and V. Davies, “A survey of mobility models for ad hoc network research,” *Wirel. Commun. Mob. Comput.*, vol. 2, pp. 483–502, Sep 2002.
- [50] C. Bettstetter, G. Resta, and P. Santi, “The node distribution of the random waypoint mobility model for wireless ad hoc networks,” *IEEE Trans. Mobile Comput.*, vol. 2, no. 3, pp. 257–269, Jul, 2003.
- [51] E. Hyttiä and J. Virtamo, “Random waypoint model in n -dimensional space,” *Oper. Res. Lett.*, vol. 33, no. 6, pp. 567–571, 2005.
- [52] F. Chollet *et al.*, “Keras,” <https://keras.io>, 2015.
- [53] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, ser. Adaptive Computation and Machine Learning series. MIT Press, 2016.
- [54] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations*, 12 2014.
- [55] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.
- [56] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: <http://tensorflow.org/>
- [57] A. M. Mathai, *The H-function theory and applications*. New York, NY, USA: Springer, 2009.
- [58] T. Princy, “Krätzel function and related statistical distributions,” *Communications in Mathematics and Statistics*, vol. 2, no. 3-4, pp. 413–429, 2014.
- [59] A. A. Kilbas, *H-transforms: Theory and Applications*. CRC Press, 2004.
- [60] N. C. Sagias, D. A. Zogas, and G. K. Karagiannidis, “Selection diversity receivers over nonidentical weibull fading channels,” *IEEE Trans. Veh. Technol.*, vol. 54, no. 6, pp. 2146–2151, Nov. 2005.
- [61] A. P. Prudnikov, Y. A. Brychov, and O. I. Marichev, *Integrals and Series: More special functions*, ser. Integrals and Series. New York, NY, USA: Gordon and Breach Science Publishers, 1986, vol. 3.

- [62] I. Gradshteyn and I. Ryzhik, *Table of Integrals, Series, and Products*, 17th ed. Burlington, MA, USA: Academic Press, 2007.
- [63] F. Yilmaz and M.-S. Alouini, “A novel unified expression for the capacity and bit error probability of wireless communication systems over generalized fading channels,” *IEEE Trans. Commun.*, vol. 60, no. 7, pp. 1862–1876, 2012.
- [64] M. You, H. Sun, J. Jiang, and J. Zhang, “Unified framework for the effective rate analysis of wireless communication systems over miso fading channels,” *IEEE Trans. Commun.*, vol. 65, no. 4, pp. 1775–1785, 2017.
- [65] MATLAB, *version 7.10.0 (R2010a)*. Natick, Massachusetts: The MathWorks Inc., 2010.
- [66] A. Leon-Garcia, *Probability, Statistics, and Random Processes for Electrical Engineering*. Pearson/Prentice Hall, 2008.
- [67] W. R. Inc., “Mathematica, Version 13.0.0,” champaign, IL, 2021. [Online]. Available: <https://www.wolfram.com/mathematica>
- [68] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, v. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, and A. Scopatz, “SymPy: symbolic computing in python,” *PeerJ Computer Science*, vol. 3, p. e103, Jan. 2017. [Online]. Available: <https://doi.org/10.7717/peerj-cs.103>

Apêndice I

Implementação, treinamento e validação do Autoencoder

A seguir disponibiliza-se o código de implementação do autoencoder (AE) em linguagem Python. Inicia-se pela importação das bibliotecas, definições de parâmetros e definição do modelo.

```
1 #importing main libraries
2 import numpy as np
3 import tensorflow as tf
4 from tensorflow import keras
5 from tensorflow.keras import layers
6 from tensorflow.math import sqrt
7
8 #setting random seed
9 np.random.seed(151)
10 tf.random.set_seed(151)
11
12 #setting communication parameters
13 M = 16
14 n = 1
15
16 #setting channel parameters
17 D = 75
18 alpha = 2.1
19 mu = 1.8
20 v = 1.3
21 delta = 2
22 dim = 1
23 snrdb = 45
24
```

```

25 # custom functions layers
26 norm_layer = keras.layers.Lambda(lambda x: tf.divide(x,tf.sqrt(2*tf.
    reduce_mean(tf.square(x))))))
27 shape_layer_encoder = keras.layers.Lambda(lambda x: tf.reshape(x,
    shape=[-1,2,n]))
28 shape_layer_decoder = keras.layers.Lambda(lambda x: tf.reshape(x,
    shape=[-1,2*n]))
29
30 #alpha-mu\gamma - 1-D rwp channel
31 def channel_model(x,snrdb,alpha,mu,v,delta,D):
32     size = tf.shape(x)
33     snr = tf.cast(10**(snrdb/10),tf.float32)
34     noise_std = sqrt(1/(2*snr))
35
36     #1-D rwp samples
37     U = tf.cast(tf.random.uniform(size),tf.complex64)
38
39     j = sqrt(tf.constant(-1,dtype = tf.complex64))
40     a0 = tf.constant(3,dtype = tf.complex64)
41
42     samples = tf.math.real((1/4)*(2 + (1 - (j)*sqrt(a0) + (1 + (j)*
        sqrt(a0))*(-1 + 2*U + 2*sqrt((-1 + U)*U))**(2/3))/(-1 + 2*U + 2*
        sqrt((-1 + U)*U))**(1/3)))
43
44     omegaS = (samples*D)**(-delta)
45     omega = (tf.random.gamma(shape = [],alpha = v,beta = 1/omegaS))**(
        alpha/2)
46     h = (tf.random.gamma(shape = [],alpha = mu,beta = mu/omega))**(1/
        alpha)
47     h = tf.reshape(h,tf.shape(x))
48     ns = tf.random.normal(size, stddev = noise_std)
49     y = x + ns/h
50
51     return tf.cast(y, tf.float32)
52
53 channel_layer = keras.layers.Lambda(lambda x:channel_model(x,snrdb,
    alpha,mu,v,delta,D))
54
55 #encoder
56 encoder = keras.models.Sequential([
57     keras.layers.Embedding(M, M, input_length=1),
58     keras.layers.Dense(M, activation="relu"),
59     keras.layers.Dense(2*n, activation=None),
60     shape_layer_encoder,
61     norm_layer,

```

```
62     ])  
63  
64 #decoder  
65 decoder = keras.models.Sequential([  
66     keras.layers.InputLayer(input_shape=[2,n]),  
67     shape_layer_decoder,  
68     keras.layers.Dense(M, activation="relu"),  
69     keras.layers.Dense(M, activation="softmax"),  
70     ])  
71  
72 autoencoder = keras.models.Sequential([encoder,channel,decoder])
```

Agora, apresentam-se os parâmetros e código para o treinamento do modelo anteriormente criado.

```
1 #training parameters  
2 learning_rate = 0.001  
3 batch_size = 512  
4 n_epochs = 10  
5 n_steps = 500  
6  
7  
8 #training model  
9 autoencoder.reset_states()  
10 optimizer = keras.optimizers.Adam(learning_rate = learning_rate)  
11 cost = keras.losses.SparseCategoricalCrossentropy( )  
12  
13 min_loss_value = np.inf  
14 for i in range(1,n_epochs+1):  
15     for j in range(1,n_steps+1):  
16         with tf.GradientTape() as tape:  
17             m_train = np.random.randint(M, size=(batch_size,1))  
18             m_pred = autoencoder(m_train,training = True)  
19  
20             loss = tf.reduce_mean(cost(m_train,m_pred))  
21             grad = tape.gradient(loss, autoencoder.trainable_variables)  
22             optimizer.apply_gradients(zip(grad,autoencoder.  
trainable_variables))  
23  
24             if loss < min_loss_value:  
25                 min_loss_value = loss  
26                 best_score = autoencoder.get_weights()  
27                 print('weights updated! best score:{}'.format(min_loss_value  
28 ))  
28 autoencoder.set_weights(best_score)
```

Por fim, avalia-se o AE em função da frequência relativa de erros de mensagens, utilizando-se o código seguinte.

```
1 #evaluating autoencoder
2 n_eval = int(1e2)
3 pt_db = np.arange(0,100,3)
4 sle = np.zeros(n_eval)
5 sler = np.zeros(len(pt_db))
6 ssize = 50000
7
8 for i in range(len(pt_db)):
9     snrdb = pt_db[i]
10    for j in range(n_eval):
11        m_test = np.random.randint(M, size=(ssize,1))
12        x_test = encoder(m_test)
13        y_test = channel_model(x_test, snrdb, alpha, mu, v, delta, D)
14        m_hat = np.argmax(decoder(y_test).numpy(), axis=1).reshape(-1,1)
15        sle[j] = np.mean(np.not_equal(m_test, m_hat))
16    sler[i] = np.mean(sle)
17    print('pt [dB]: {} BLER: {}'.format(pt_db[i], sler[i]))
```

Apêndice II

Processo de geração de amostras para o modelo de deslocamento *random waypoint*

Para o processo de geração das amostras de para o modelo de paradas aleatórias (*random waypoint model*, RWP) propõem-se duas formas, o primeiro pelo método da transformação e o segundo, da rejeição [66].

Para o método da transformação, necessita-se encontrar a função de distribuição cumulativa (FDC) e em seguida fazer sua inversão. Uma vez que a função densidade de probabilidade (FDP) dada em (2.26) é uma função polinomial, encontrar a FDC é um processo simples. Entretanto, a inversão não é trivial e carece do estudo de raízes, para encontrar qual(ais) resultam em uma distribuição válida. Processo que é facilitado por ferramentas como Mathematica [67] e SymPy [68]. Apenas os modelos uni e bidimensional possuem raízes factíveis de serem computadas. Também é importante ressaltar que o processo envolve cálculos complexos o que diminui o desempenho para geração das amostras. O algoritmo para o método da transformação é descrito subsequentemente em linguagem Python.

```
1 #required libraries
2 import numpy as np
3 from cmath import sqrt
4 #1-D Model
5 def rwp_1d(n_samples,D):
6     #D -> coverage distance
7     #n_samples -> number of samples desired
8     u = np.random.rand(n_samples) #uniform distribution (0,1)
9     sample = []
10    for U in u:
```

```

11     sample.append(((1/4)*(2 + (1 - (1j)*sqrt(3) + (1 + (1j)*sqrt(3))
12         *(-1 + 2*U + 2*sqrt((-1 + U)*U))**(2/3))/(-1 + 2*U + 2*sqrt
13         ((-1 + U)*U))**(1/3))).real)
14     return np.array(sample)*D
15 #2-D Model
16 def rwp_2d(n_samples,D):
17     u = np.random.rand(n_samples)
18     sample = []
19     for U in u:
20         sample.append(D*(sqrt(-(1/32)*(16*sqrt(73)*sqrt(4672*U**2 -
21             2485*U - 2187) + 9344*U - 2485)**(1/3)) +
22             (1j/32)*sqrt(3)*(16*sqrt(73)*sqrt(4672*U**2 - 2485*U - 2187) +
23             9344*U - 2485)**(1/3) -
24             (361j*sqrt(3))/(32*(16*sqrt(73)*sqrt(4672*U**2 - 2485*U - 2187)
25             + 9344*U - 2485)**(1/3)) -
26             361/(32*(16*sqrt(73)*sqrt(4672*U**2 - 2485*U - 2187) + 9344*U -
27             2485)**(1/3)) + 35/16)).real)
28     return np.array(sample)*D

```

Para o método da rejeição, propõe-se uma função simples, que apesar de resultar em uma baixa taxa de aceitação, em torno de 52%, possui uma baixa complexidade, portanto maior desempenho para gerar amostras. Seja f_{sup} a função que limita superiormente (2.26), definida como

$$f_{\text{sup}} = \frac{n}{D} \text{sen} \left(\frac{\pi d}{D} \right), \quad (\text{II.1})$$

para $0 \leq d \leq D_0$, em que n é o número de elementos no somatório de (2.26), ou seja, $n = 2$ para o caso unidimensional e $n = 3$ para bi e tridimensional. O algoritmo proposto está implementado adiante em linguagem Python.

```

1 #required libraries
2 import numpy as np
3 def f_r(r,dim): #pdf
4     s = np.zeros((n,1))
5     for i in range(n):
6         s[i] = B[i]*np.power(r,beta[i])/(np.power(D,beta[i]+1))
7     return np.sum(s,axis=0)
8 def RWP_Sample_Gen(D,n_samples,dim):
9     #D -> coverage distance
10    #n_samples -> number of samples desired
11    #dim -> dimension
12
13    if dim == 1:
14        B = [6,-6]

```



```
15     beta = [1,2]
16     n = 2
17     elif dim == 2:
18         B = [324/73, -420/73, 96/73]
19         beta = [1,3,5]
20         n = 3
21     elif dim == 3:
22         B = [735/72, -1190/72, 455/72]
23         beta = [2,4,6]
24         n = 3
25
26     samples = []
27     k=0
28     rejected = 0
29     while(k<n_samples):
30         u = np.random.uniform(0,1)
31         r = D/np.pi*np.arccos(1-2*u)
32         b = n/D*np.sin(np.pi*r/D)
33         y = np.random.uniform(0,b)
34         f = f_r(r,dim)
35         if y <= f:
36             samples.append(r)
37             k += 1
38         else:
39             rejected += 1
40     print('acceptance ratio = {}'.format(k/(k+rejected)))
41     return samples
42
```

Apêndice III

Algoritmos para geração e validação dos resultados e métricas de desempenho

Nas próximas sessões são disponibilizados códigos para a geração das funções deduzidas no Capítulo 4, para MATLAB.

III.1 Função densidade de probabilidade

```
1 Pt = 10;
2 D = 10;
3 alpha = 2;
4 b = 2;
5 mu = 2;
6 delta = 4;
7 lamb_max = 100;
8
9 lamb = 0.1:.1:(lamb_max+0.1);
10
11 B = [6,-6];
12 beta = [1,2];
13 n = 2;
14
15 u = 1; #convergence parameter. Default = 1
16 omega_0 = Pt*D^(-delta);
17 f_A2 = zeros(n,1);
18 f_x = zeros(length(lamb),1);
19 for j=1: (length(lamb))
20     arg1 = u*1/(lamb(j)*delta*gamma(mu)*gamma(b));
```

```
21 z = mu^(2/alpha)*lamb(j)/omega_0;
22 for i=1:n
23     ap = [1-((beta(i)+1)/delta)];
24     bq = [b,mu,-((beta(i)+1)/delta)];
25     Ap = [1];
26     Bq = [1,2/alpha,1];
27     P={ [1], [1], ap,bq,Ap*u,Bq*u};
28     O=[2 1 1 3];
29     f_A2(i) = B(i).*func_Fox_H_univariate(z^u, O, P);
30     f_x(j) = arg1*sum(f_A2);
31 end
32 end
```

III.2 Função de distribuição acumulada

```
1 Pt = 1;
2 D = 100;
3 alpha = 2.2;
4 b = 2.3;
5 mu = 1.8;
6 delta = 2;
7 lamb_max = 100;
8 step=0.1;
9 dim=3;
10
11 lamb = 0.1:step:(lamb_max+step);
12
13 if dim==1
14     B = [6,-6];
15     beta = [1,2];
16     n = 2;
17 elseif dim == 2
18     B = [324/73,-420/73,96/73];
19     beta = [1,3,5];
20     n=3;
21 elseif dim == 3
22     B =[735/72,-1190/72,455/72];
23     beta=[2,4,6];
24     n=3;
25 end
26
27 omega_0 = Pt*D^(-delta);
28
29 f_A2 = zeros(n,1);
```

```

30 F_x = zeros(length(lamb),1);
31 for j=1:(length(lamb))
32     arg1 = 1/(delta*gamma(mu)*gamma(b));
33     z = mu^(2/alpha)*lamb(j)/omega_0;
34     for i=1:n
35         O=[2 2 2 4];
36         P={ [1],[1],[1,1-((beta(i)+1)/delta)], [b,mu,-((beta(i)+1)/delta)
37             ,0],[1,1],[1,2/alpha,1,1]};
38         f_A2(i) = B(i).*func_Fox_H_univariate(z, O, P);
39     end
40     F_x(j) = arg1*sum(f_A2);
41 end

```

III.3 Probabilidade de erro de símbolo média e assintótica

```

1 D = 50;
2 alpha = 1.5;
3 mu = 2;
4 b = 3;
5 delta = 2.3;
6 PtdB_min = 50;
7 PtdB_max = 70;
8 Npoints = 100;
9
10 dim =1;
11
12 M = 2;
13
14 if M==2 %bpsk mod
15     h = 1/2;
16     w = 1;
17 else %Square M-QAM (M>=4)
18     w = 3/(2*(M-1));%a
19     h = 2-2/sqrt(M);%b
20 end
21
22 stp_sim = 7;
23
24 stp = (PtdB_max-PtdB_min)/Npoints;
25 PtdBVec = PtdB_min:stp:PtdB_max;
26
27 if dim==1
28     B = [6,-6];
29     beta = [1,2];

```

```

30     n = 2;
31 elseif dim == 2
32     n=3;
33     B = [324/73,-420/73,96/73];
34     beta = [1,3,5];
35 elseif dim == 3
36     n=3;
37     B =[735/72,-1190/72,455/72];
38     beta=[2,4,6];
39 end
40 f_A2 = zeros(n,1);
41 ps = zeros(length(PtdBVec),1);
42
43 u = 0.18;
44 for j=1: (length(PtdBVec))
45     Pt = 10^(PtdBVec(j)/10.0);
46     omega_0 = Pt*D^(-delta);
47     arg1 = u*h/(sqrt(pi)*delta*gamma(mu)*gamma(b));
48     z = (mu^(2/alpha))/(omega_0*w);
49     for i=1:n
50         v =(beta(i)+1)/delta;
51         ap = [1/2,1,1-v];
52         bq = [b,mu,-v,0];
53         Ap = [1,1,1];
54         Bq = [1,2/alpha,1,1];
55         O=[2 3 3 4];
56         P={ [1], [1], ap,bq,Ap*u,Bq*u};
57         f_A2(i) = B(i).*func_Fox_H_univariate(z^u, O, P);
58         ps(j) = arg1*sum(f_A2);
59     end
60 end
61
62 ass_A2 = zeros(n,1);
63 assps = zeros(length(PtdBVec),1);
64 for j=1: (length(PtdBVec))
65     Pt = 10^(PtdBVec(j)/10.0);
66     omega_0 = Pt*D^(-delta);
67     arg1 = h/(sqrt(pi)*delta*gamma(mu)*gamma(b));
68     z = (mu^(2/alpha))/(omega_0*w);
69     for i=1:n
70         if b < alpha*mu/2
71             k = gamma(mu-2*b/alpha)*gamma(b+1/2)*z^b/(((beta(i)+1)/delta+b)*
72             b);
73         else
74             k = gamma(b-alpha*mu/2)*gamma(0.5*(alpha*mu+1))*z^(alpha*mu/2)

```

```

/(((beta(i)+1)/delta+alpha*mu/2)*mu);
74 end
75 ass_A2(i) = B(i).*k;
76 assps(j) = arg1*sum(ass_A2);
77 end
78 end

```

III.4 Probabilidade de indisponibilidade

```

1 Pt_dB_min = 0;
2 Pt_dB_max = 55;
3
4 D = 1;
5 b = 1.8;
6 alpha = 1.1;
7 mu = 2.3;
8 delta = 3.1;
9
10 kappa = 4;
11
12 B = [6, -6];
13 beta = [1, 2];
14 n = 2;
15
16 f_A2 = zeros(n, 1);
17 Pout = zeros(length(Pt_dB_vec), 1); %outage
18 for j=1:(length(Pt_dB_vec))
19   Pt = 10^(Pt_dB_vec(j)/10);
20   omega_0 = Pt*D^(-delta);
21   arg1 = 1/(delta*gamma(mu)*gamma(b));
22   z = mu^(2/alpha)*kappa/omega_0;
23   for i=1:n
24     O=[2 2 2 4];
25     P={ [1], [1], [1, 1-((beta(i)+1)/delta)], [b, mu, -((beta(i)+1)/delta)
, 0], [1, 1], [1, 2/alpha, 1, 1]};
26     f_A2(i) = B(i).*func_Fox_H_univariate(z, O, P);
27   Pout(j) = arg1*sum(f_A2);
28   end
29 end

```

III.5 Probabilidade de indisponibilidade com sinal interferente

```
1 Pt_dB_min = 0;
2 Pt_dB_max = 55;
3
4 D = 1;
5 b = 1.8;
6 alpha = 1.1;
7 mu = 2.3;
8 delta = 3.1;
9
10 Pc_dB = 15;
11 Dc = 1;
12 b_c = b;
13 m_c = mu;
14 delta_c = delta;
15 N = 4;
16
17 kappa = 4;
18
19 Pt_dB_vec = Pt_dB_min:stp_sim:Pt_dB_max;
20 Pc = 10^(Pc_dB/10);
21 omega_c = Pc*Dc^(-delta_c);
22
23 B = [6,-6];
24 beta = [1,2];
25 n = 2;
26
27 f_A2 = zeros(n,1);
28 Pout = zeros(length(Pt_dB_vec),1);
29 sim_rep = zeros(length(PrecSim),1);
30 Pout_sim = zeros(length(Pt_dB_vec),1);
31 for j=1:(length(Pt_dB_vec))
32     Pt = 10^(Pt_dB_vec(j)/10);
33     omega_0 = Pt*D^(-delta);
34     arg1 = 1/(delta*gamma(mu)*gamma(b)*gamma(N*m_c)*gamma(b_c));
35     z = mu^(2/alpha)*omega_c*kappa/(omega_0*m_c);
36     for i=1:n
37         O=[2 4 4 4];
38         P={ [1], [1], [1,1-b_c,1-N*m_c,1-(beta(i)+1)/delta)], [b,mu,-(beta
39             (i)+1)/delta), 0], [1,1,1,1], [1,2/alpha,1,1]};
40         f_A2(i) = B(i).*func_Fox_H_univariate(z, O, P);
41     end
42     Pout(j) = arg1*sum(f_A2);
```

```

41 end
42 end

```

III.6 Capacidade

```

1 D = 1;
2 alpha = 1.8;
3 mu = 0.7;
4 b = 1.8;
5 delta = 4.2;
6
7 PtdB_min = -10;
8 PtdB_max = 60;
9 Npoints = 10;
10
11 stp_sim = 7;
12
13 stp = (PtdB_max-PtdB_min)/Npoints;
14 PtVec = PtdB_min:stp:PtdB_max;
15
16 B = [6,-6];
17 beta = [1,2];
18 n = 2;
19
20 f_A2 = zeros(n,1);
21 Cap = zeros(length(PtVec),1);
22
23 u = 2/(alpha*mu)^b;
24 u=0.05;
25 t = 0;
26 for j=1: (length(PtVec))
27     Pt = 10^(PtVec(j)/10);
28     omega_0 = Pt*D^(-delta);
29     z = (omega_0/mu^(2/alpha));
30     arg1 = u/(log(2)*delta*gamma(mu)*gamma(b));
31     for i=1:n
32         v =(beta(i)+1)/delta;
33         ap = [1,1,1-b,1-mu,1+v];
34         bq = [1,v,0];
35         Ap = [1,1,1,2/alpha,1];
36         Bq = [1,1,1];
37
38         O=[2 4 5 3];
39         P={ [1],[1], ap,bq, Ap*u, Bq*u};

```



```
40     f_A2(i) = B(i).*func_Fox_H_univariate(z^u, O, P);  
41     end  
42     Cap(j) = arg1*sum(f_A2);  
43 end
```