

MODELAGEM ESTOCÁSTICA DO APRENDIZADO ONLINE DE UMA  
ESTRUTURA LINEAR ADAPTATIVA IMPLEMENTADA EM BLOCOS

Thiago Rangel Pessel Gonzaga

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ, como parte dos requisitos necessários à obtenção do grau de mestre.

Orientadores:  
Diego Barreto Haddad  
Felipe da Rocha Henriques

Rio de Janeiro,  
Fevereiro de 2022

**Modelagem Estocástica do Aprendizado Online de uma Estrutura  
Linear Adaptativa Implementada em Blocos**

Dissertação de Mestrado em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ.

Thiago Rangel Pessel Gonzaga

Aprovada por:

---

Presidente, Prof. Diego Barreto Haddad, D.Sc. (orientador)

---

Coorientador, Prof. Felipe da Rocha Henriques, D.Sc. (coorientador)

---

Eduardo Bezerra da Silva, D.Sc.

---

Tadeu Nagashima Ferreira, D.Sc.

Rio de Janeiro,

Fevereiro de 2022

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

G643 Gonzaga, Thiago Rangel Pessel

Modelagem estocástica do aprendizado online de uma estrutura linear adaptativa implementada em blocos / Thiago Rangel Pessel Gonzaga. — 2022.

116f. + apêndice : il. (algumas color.) , enc.

Dissertação (Mestrado) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2022.

Bibliografia : f. 110-116

Orientador: Diego Barreto Haddad

Coorientador: Felipe da Rocha Henriques

1. Teoria do ruído estocástico. 2. Teoria dos sinais (Telecomunicação). 3. Teoria da comunicação estatística. 4. Algoritmos. 5. Computação. I. Haddad, Diego Barreto (Orient.). II. Henriques, Felipe da Rocha (Coorient.). III. Título.

CDD 621.38224

## DEDICATÓRIA

Dedico este trabalho aos meus pais, Vera e Nelson, a minha Esposa Cristhiane, ao meu filho Davi e a minha Irmã Daniela (*In memoriam*). Sem eles, certamente, eu não seria o que sou hoje e não chegaria até aqui.

## **AGRADECIMENTOS**

Agradeço primeiramente ao Centro Federal de Educação Tecnológica Celso Suckow da Fonseca - CEFET/RJ, esta instituição que me ensinou e me orientou quando jovem e que mais uma vez me abrigou, desta vez no Programa de Pós-Graduação em Ciência da Computação com seus professores competentes e dedicados, fazendo com que eu pudesse contribuir com a ciência e a pesquisa no Brasil.

Agradeço ao professor Diego Barreto Haddad, que aceitou ser meu orientador, por sua paciência, dedicação e por ter servido de guia e exemplo nesta trajetória de aprendizado.

Também estendo meus agradecimentos ao meu co-orientador, o professor Felipe da Rocha Henriques, por seus conselhos, compreensão e auxílio nesta jornada.

Um agradecimento à minha amada esposa Cristhiane Miniguelo por seu amor, dedicação, incentivo e paciência.

E por fim, não poderia deixar de agradecer à Claro Brasil, empresa onde trabalho e que com a compreensão do meu coordenador Iuri Napoleão e de minha Diretora Yeda Matta me proporcionaram a oportunidade de poder concluir este aprendizado.

## **RESUMO**

Modelagem Estocástica do Aprendizado Online de uma Estrutura Linear Adaptativa  
Implementada em Blocos

Thiago Rangel Pesset Gonzaga

Orientadores:

Diego Barreto Haddad  
Felipe da Rocha Henriques

Resumo da Dissertação submetida ao Programa de Pós-graduação em Ciência da Computação do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ como parte dos requisitos necessários à obtenção do grau de mestre.

Os algoritmos de filtragem adaptativa constituem uma família de técnicas com ampla aplicação em problemas de grande relevância, tais como equalização de canais, cancelamento de eco acústico, cancelamento de ruído, identificação de sistemas e séries temporais. Este trabalho propõe um modelo estocástico capaz de prever as características de aprendizado dos algoritmos LMS e NLMS implementados em blocos. A análise é simplificada por um modelo que divorcia a distribuição radial dos vetores de entrada da distribuição angular, a qual é discretizada. A despeito desta simplificação, o modelo utilizado de sinal de entrada é coerente com a matriz de autocorrelação original dos dados de entrada. A partir desta análise foi possível modelar o comportamento de divergência dos algoritmos estudados, relacionar o MSE em excesso a quantidade de blocos utilizados e de forma proporcional ao tamanho do filtro. As previsões teóricas serão comparadas com curvas de desempenho oriundas de simulação, de modo a aferir a acurácia das estimativas resultantes.

Palavras-chave:

Aprendizagem On-line; Filtragem Adaptativa; Modelos Estocásticos.

Rio de Janeiro,  
Fevereiro de 2022

## **ABSTRACT**

Stochastic Modeling of Online Learning of an Adaptive Linear Structure Implemented in  
Blocks

Thiago Rangel Pesset Gonzaga

Advisors:

Diego Barreto Haddad

Felipe da Rocha Henriques

Abstract of dissertation submitted to Programa de Pós-graduação em Ciência da Computação - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ as partial fulfillment of the requirements for the degree of master.

Adaptive filtering algorithms are a family of techniques with wide application in highly relevant problems, such as channel equalization, acoustic echo cancellation, noise cancellation, systems identification and time series. This work proposes a stochastic model capable of predicting the learning characteristics of the block LMS algorithm. The analysis is simplified by a model that divorces the radial distribution of the input vectors from the angular distribution, which is discretized. Despite this simplification, the model used for the input signal is consistent with the original autocorrelation matrix of the input data. From this analysis it was possible to model the divergence behavior of the studied algorithms, relating the excess MSE to the amount of blocks used and proportionally to the filter size. Theoretical predictions will be compared with performance curves derived from simulation, in order to assess the accuracy of the resulting estimates.

Key-words:

On-line Learning; Adaptive Filtering; Stochastic Models.

Rio de Janeiro,

Fevereiro de 2022

## Lista de Figuras

II.1	Filtro passa-baixas projetado. . . . .	32
II.2	Resposta frequencial do filtro projetado no exemplo. . . . .	32
II.3	Exemplo de um problema de equalização de canal. A função de transferência $H(z)$ emula a distorção provocada pelo canal. Cabe ao filtro $W(z)$ desfazer tal distorção (ou seja, efetuar uma equalização). . . . .	33
II.4	Evolução de alguns coeficientes $w_i(k)$ do exemplo ilustrativo de equalização, para $i \in \{0, 4, 9\}$ e apenas uma execução do algoritmo. O algoritmo de atualização utilizado é o CMA, com $\beta = 10^{-3}$ . . . . .	35
II.5	Evolução média de alguns coeficientes $w_i(k)$ do exemplo ilustrativo de equalização, para $i \in \{0, 4, 9\}$ , obtida com 20 ensaios independentes de Monte Carlo. O algoritmo de atualização utilizado é o CMA, com $\beta = 10^{-3}$ . . . . .	36
II.6	Função de transferência equivalente, para a iteração $k = 10^4$ do algoritmo CMA, com $\beta = 10^{-3}$ . A configuração é a mesma da Figura II.4. . . . .	37
II.7	Evolução média da ISI (em dB), obtida com 10 ensaios independentes de Monte Carlo. A atualização foi feita pelo algoritmo CMA, com diferentes valores de $\beta$ . . . . .	38
II.8	Diagrama de blocos de um algoritmo adaptativo atuando na identificação de sistemas. . . . .	41
II.9	Função de transferência considerada no exemplo ilustrativo. . . . .	42
II.10	Evolução média de alguns coeficientes adaptativos, obtida com 100 ensaios independentes de Monte Carlo. As linhas tracejadas horizontais sinalizam os valores respectivos da função de transferência ótima. . . . .	43
II.11	Evolução do MSE, computada com $10^3$ ensaios independentes de Monte Carlo, para diferentes valores de $\beta$ . . . . .	44
II.12	Evolução do MSD do LMS com $\beta = 2 \times 10^{-3}$ , computada com $10^3$ ensaios independentes de Monte Carlo, para sinal de entrada branco e sinal de entrada colorido com filtro $H(z) = \frac{1}{1-0.8z^{-1}}$ . . . . .	44
II.13	Diagrama de blocos de um algoritmo de Filtragem Adaptativa atuando em apri- moramento de Sinal. . . . .	47

II.14	Diagrama de blocos de um algoritmo de Filtragem Adaptativa atuando em Predição de Sinais. . . . .	47
III.1	Distribuição em probabilidade estimada empiricamente pelo método de <i>kernels</i> \mathbf{x}(k), a partir de $10^6$ vetores. . . . .	55
III.2	Distribuição em probabilidade estimada empiricamente pelo método de <i>kernels</i> \mathbf{x}(k), a partir de $10^6$ vetores (em azul). Em vermelho tracejado, a distribuição teórica de Rayleigh que descreve a solução analítica para este caso. . . . .	56
III.3	Distribuição em probabilidade estimada empiricamente pelo método de <i>kernels</i> \mathbf{x}(k), a partir de $10^6$ vetores (em azul), obtidos de acordo com o modelo de média móvel (III.19). . . . .	58
V.1	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 1$ e $N = 10$ . . . . .	86
V.2	Evolução do $\tilde{\lambda}_i(k)$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BLMS com $L = 1$ e $N = 10$ . . . . .	86
V.3	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BLMS com $L = 1$ e $N = 10$ . . . . .	87
V.4	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 1$ e $N = 10$ . . . . .	87
V.5	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 2$ e $N = 10$ . . . . .	88
V.6	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BLMS com $L = 2$ e $N = 10$ . . . . .	88
V.7	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BLMS com $L = 2$ e $N = 10$ . . . . .	89
V.8	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 2$ e $N = 10$ . . . . .	89
V.9	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 3$ e $N = 10$ . . . . .	90
V.10	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BLMS com $L = 3$ e $N = 10$ . . . . .	90
V.11	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BLMS com $L = 3$ e $N = 10$ . . . . .	91

V.12	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 3$ e $N = 10$ . . . . .	91
V.13	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 4$ e $N = 10$ . . . . .	92
V.14	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BLMS com $L = 4$ e $N = 10$ . . . . .	92
V.15	Evolução do $\tilde{\lambda}_i(k)$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BLMS com $L = 4$ e $N = 10$ . . . . .	93
V.16	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 4$ e $N = 10$ . . . . .	93
V.17	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 1$ e $N = 60$ . . . . .	94
V.18	Evolução do $\tilde{\lambda}_i(k)$ teórico e simulado com $i = 3$ ao longo das iterações do algoritmo BLMS com $L = 1$ e $N = 60$ . . . . .	95
V.19	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 7$ ao longo das iterações do algoritmo BLMS com $L = 1$ e $N = 60$ . . . . .	95
V.20	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 1$ e $N = 60$ . . . . .	96
V.21	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 2$ e $N = 60$ . . . . .	96
V.22	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 3$ ao longo das iterações do algoritmo BLMS com $L = 2$ e $N = 60$ . . . . .	97
V.23	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 7$ ao longo das iterações do algoritmo BLMS com $L = 2$ e $N = 60$ . . . . .	97
V.24	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 2$ e $N = 60$ . . . . .	98
V.25	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 3$ . . . . .	98
V.26	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 3$ ao longo das iterações do algoritmo BLMS com $L = 3$ e $N = 60$ . . . . .	99
V.27	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 7$ ao longo das iterações do algoritmo BLMS com $L = 3$ e $N = 60$ . . . . .	99
V.28	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 3$ e $N = 60$ . . . . .	100

V.29	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 4$	100
V.30	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 3$ ao longo das iterações do algoritmo BLMS com $L = 4$ e $N = 60$	101
V.31	Evolução do $\tilde{\lambda}_i(k)$ teórico e simulado com $i = 7$ ao longo das iterações do algoritmo BLMS com $L = 4$ e $N = 60$	101
V.32	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 4$ e $N = 60$	102
V.33	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$	103
V.34	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$	103
V.35	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$	104
V.36	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$	104
V.37	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 2$ e $N = 10$	105
V.38	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BNLMS com $L = 2$ e $N = 10$	105
V.39	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BNLMS com $L = 2$ e $N = 10$	106
V.40	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 2$ e $N = 10$	106
V.41	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 3$ e $N = 10$	107
V.42	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BNLMS com $L = 3$ e $N = 10$	107
V.43	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BNLMS com $L = 3$ e $N = 10$	108
V.44	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 3$ e $N = 10$	108
V.45	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 4$ e $N = 10$	109

V.46	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BNLMS com $L = 4$ e $N = 10$ . . . . .	109
V.47	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BNLMS com $L = 4$ e $N = 10$ . . . . .	110
V.48	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 4$ e $N = 10$ . . . . .	110
V.49	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 60$ . . . . .	111
V.50	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 18$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 60$ . . . . .	112
V.51	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 21$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 60$ . . . . .	112
V.52	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 60$ . . . . .	113
V.53	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 2$ e $N = 60$ . . . . .	113
V.54	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 18$ ao longo das iterações do algoritmo BNLMS com $L = 2$ e $N = 60$ . . . . .	114
V.55	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 21$ ao longo das iterações do algoritmo BNLMS com $L = 2$ e $N = 60$ . . . . .	114
V.56	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 2$ e $N = 60$ . . . . .	115
V.57	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 3$ e $N = 60$ . . . . .	115
V.58	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 18$ ao longo das iterações do algoritmo BNLMS com $L = 3$ e $N = 60$ . . . . .	116
V.59	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 21$ ao longo das iterações do algoritmo BNLMS com $L = 3$ e $N = 60$ . . . . .	116
V.60	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 3$ e $N = 60$ . . . . .	117
V.61	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 4$ e $N = 60$ . . . . .	117
V.62	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 18$ ao longo das iterações do algoritmo BNLMS com $L = 4$ e $N = 60$ . . . . .	118

V.63	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 21$ ao longo das iterações do algoritmo BNLMS com $L = 4$ e $N = 60$ . . . . .	118
V.64	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 4$ e $N = 60$ . . . . .	119
V.65	Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 1$ e $N = 10$ . . . . .	120
V.66	Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 2$ e $N = 10$ . . . . .	121
V.67	Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 3$ e $N = 10$ . . . . .	121
V.68	Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 4$ e $N = 10$ . . . . .	122
V.69	Curva de probabilidade de divergência simulada versus quantidade de blocos $L$ do algoritmo BLMS com $N = 10$ . . . . .	123
V.70	Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 1$ e $N = 60$ . . . . .	124
V.71	Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 2$ e $N = 60$ . . . . .	124
V.72	Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 3$ e $N = 60$ . . . . .	125
V.73	Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 4$ e $N = 60$ . . . . .	125
V.74	Curva de probabilidade de divergência simulada versus quantidade de blocos $L$ do algoritmo BLMS com $N = 60$ . . . . .	126
V.75	Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 2$ e $N = 10$ . . . . .	127
V.76	Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 3$ e $N = 10$ . . . . .	128
V.77	Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 4$ e $N = 10$ . . . . .	128
V.78	Curva de probabilidade de divergência simulada versus quantidade de blocos $L$ do algoritmo BNLMS com $N = 10$ . . . . .	129
V.79	Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 2$ e $N = 60$ . . . . .	130

V.80	Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 3$ e $N = 60$ . . . . .	131
V.81	Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 4$ e $N = 60$ . . . . .	131
V.82	Curva de probabilidade de divergência simulada versus quantidade de blocos $L$ do algoritmo BNLMS com $N = 60$ . . . . .	132
VII.1	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	142
VII.2	Evolução do $\tilde{\lambda}_i(k)$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	143
VII.3	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	143
VII.4	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	144
VII.5	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	144
VII.6	Evolução do $\tilde{\lambda}_i(k)$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	145
VII.7	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	145
VII.8	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	146
VII.9	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	146
VII.10	Evolução do $\tilde{\lambda}_i(k)$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	147
VII.11	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	147
VII.12	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	148
VII.13	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	148
VII.14	Evolução do $\tilde{\lambda}_i(k)$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	149

VII.15	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	149
VII.16	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	150
VII.17	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	150
VII.18	Evolução do $\tilde{\lambda}_i(k)$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	151
VII.19	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	151
VII.20	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	152
VII.21	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	152
VII.22	Evolução do $\tilde{\lambda}_i(k)$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	153
VII.23	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	153
VII.24	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	154
VII.25	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	154
VII.26	Evolução do $\tilde{\lambda}_i(k)$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	155
VII.27	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	155
VII.28	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	156
VII.29	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	156
VII.30	Evolução do $\tilde{\lambda}_i(k)$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	157
VII.31	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	157

VII.32	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	158
VII.33	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	158
VII.34	Evolução do $\tilde{\lambda}_i(k)$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	159
VII.35	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	159
VII.36	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	160
VII.37	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	160
VII.38	Evolução do $\tilde{\lambda}_i(k)$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	161
VII.39	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	161
VII.40	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	162
VII.41	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	162
VII.42	Evolução do $\tilde{\lambda}_i(k)$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	163
VII.43	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	163
VII.44	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	164
VII.45	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	164
VII.46	Evolução do $\tilde{\lambda}_i(k)$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	165
VII.47	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	165
VII.48	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BLMS com $L = 5$ e $N = 10$ . . . . .	166

VII.49	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	166
VII.50	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	167
VII.51	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	167
VII.52	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	168
VII.53	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	168
VII.54	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	169
VII.55	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	169
VII.56	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	170
VII.57	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	170
VII.58	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	171
VII.59	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	171
VII.60	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	172
VII.61	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	172
VII.62	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	173
VII.63	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	173
VII.64	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	174
VII.65	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	174

VII.66	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	175
VII.67	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	175
VII.68	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	176
VII.69	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	176
VII.70	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	177
VII.71	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	177
VII.72	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	178
VII.73	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	178
VII.74	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	179
VII.75	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	179
VII.76	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	180
VII.77	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	180
VII.78	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	181
VII.79	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	181
VII.80	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	182
VII.81	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	182
VII.82	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	183

VII.83	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	183
VII.84	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	184
VII.85	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	184
VII.86	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	185
VII.87	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	185
VII.88	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	186
VII.89	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	186
VII.90	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	187
VII.91	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	187
VII.92	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	188
VII.93	Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	188
VII.94	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 1$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	189
VII.95	Evolução do $\tilde{\lambda}_i$ teórico e simulado com $i = 4$ ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	189
VII.96	Evolução do $\bar{\lambda}$ médio teórico e simulado ao longo das iterações do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	190
VII.97	Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 1$ e $N = 10$ . . . . .	190
VII.98	Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 1$ e $N = 10$ . . . . .	191
VII.99	Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 1$ e $N = 10$ . . . . .	191

VII.100 Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 1$ e $N = 10$ . . . . .	192
VII.101 Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 1$ e $N = 10$ . . . . .	192
VII.102 Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 1$ e $N = 10$ . . . . .	193
VII.103 Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 1$ e $N = 10$ . . . . .	193
VII.104 Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 1$ e $N = 10$ . . . . .	194
VII.105 Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 1$ e $N = 10$ . . . . .	194
VII.106 Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 1$ e $N = 10$ . . . . .	195
VII.107 Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 1$ e $N = 10$ . . . . .	195
VII.108 Curva de probabilidade de divergência simulada do algoritmo BLMS com $L = 1$ e $N = 10$ . . . . .	196
VII.109 Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	196
VII.110 Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	197
VII.111 Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	197
VII.112 Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	198
VII.113 Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	198
VII.114 Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	199
VII.115 Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	199
VII.116 Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	200

VII.117 Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	200
VII.118 Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	201
VII.119 Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	201
VII.120 Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	202
VII.121 Curva de probabilidade de divergência simulada do algoritmo BNLMS com $L = 1$ e $N = 10$ . . . . .	202

## Lista de Tabelas

I.1	Tabela com referências de trabalhos relacionados.	29
II.1	Algumas funções custo utilizadas na literatura para a derivação de algoritmos de filtragem adaptativa utilizando gradiente estocástico. $\ \mathbf{w}\ _0$ denota uma aproximação da pseudo-norma $\ell_0$ , $n \in \mathbb{N}$ e $\lambda(\gamma)$ é uma função densidade de probabilidade válida com suporte no intervalo $(1,2)$ .	41
V.1	Tabela com valores de $\beta$ simulados da probabilidade de divergência nas faixas 0%, 50% e 100% para o algoritmo BLMS e N=10.	122
V.2	Tabela com valores de $\beta$ simulados da probabilidade de divergência nas faixas 0%, 50% e 100% para o algoritmo BLMS e N=60.	126
V.3	Tabela com valores de $\beta$ simulados da probabilidade de divergência nas faixas 0%, 50% e 100% para o algoritmo BNLMS e N=10.	129
V.4	Tabela com valores de $\beta$ simulados da probabilidade de divergência nas faixas 0%, 50% e 100% para o algoritmo BNLMS e N=60.	132

## **Lista de Abreviações**

## Sumário

<b>I</b>	<b>Introdução</b>	<b>26</b>
I.1	Introdução . . . . .	26
I.2	Motivações . . . . .	27
I.3	Objetivo . . . . .	28
I.4	Metodologia . . . . .	28
I.5	Contribuições . . . . .	28
I.6	Publicação Derivada deste Trabalho . . . . .	29
I.7	Trabalhos Relacionados . . . . .	29
I.8	Notação matemática . . . . .	29
I.9	Estrutura deste trabalho . . . . .	29
<b>II</b>	<b>Algoritmos de Filtragem Adaptativa</b>	<b>31</b>
II.1	Estruturas Transversais Adaptativas . . . . .	31
II.1.1	Exemplo ilustrativo de equalização . . . . .	34
II.2	Gradiente Estocástico . . . . .	38
II.2.1	Exemplo ilustrativo de identificação . . . . .	42
II.3	Derivação de Algoritmos de Filtragem Adaptativa via Princípio da Distorção Mínima	45
II.4	Aplicações . . . . .	46
II.4.1	Aprimoramento de sinal . . . . .	47
II.4.2	Predição de sinais . . . . .	47
II.5	Comentários Finais . . . . .	48
<b>III</b>	<b>Modelos Estocásticos de Algoritmos de Filtragem Adaptativa</b>	<b>49</b>
III.1	Introdução . . . . .	49
III.2	<i>Modelo de Slock</i> . . . . .	51
III.2.1	Exemplos ilustrativos . . . . .	54
III.3	Predição do MSE . . . . .	57
III.4	Análise do Algoritmo LMS pelo <i>Modelo de Slock</i> . . . . .	59
III.5	Análise do Algoritmo NLMS pelo <i>Modelo de Slock</i> . . . . .	63

III.6	Comentários Finais . . . . .	67
<b>IV</b>	<b>Análise dos Algoritmos LMS e NLMS em Blocos</b>	<b>68</b>
IV.1	Implementação em Blocos de Algoritmos de Filtragem Adaptativa . . . . .	68
IV.2	Modelo do sinal de entrada . . . . .	69
IV.3	Análise do BLMS . . . . .	69
IV.4	Análise do BNLMS . . . . .	77
IV.5	Comentários Finais . . . . .	83
<b>V</b>	<b>Resultados</b>	<b>84</b>
V.1	Introdução . . . . .	84
V.2	Simulação 1 - MSE e $\tilde{\lambda}_i$ . . . . .	85
V.2.1	BLMS . . . . .	85
V.2.2	BNLMS . . . . .	102
V.3	Simulação 2 - Divergência em função de $\beta$ . . . . .	119
V.3.1	BLMS . . . . .	120
V.3.2	BNLMS . . . . .	127
V.4	Comentários Finais . . . . .	133
<b>VI</b>	<b>Conclusões</b>	<b>134</b>
<b>Referências Bibliográficas</b>		<b>134</b>
<b>VII</b>	<b>Apêndice A</b>	<b>142</b>

## Capítulo I Introdução

### I.1 Introdução

Vivemos na era do dilúvio de dados, na qual a conversão de informação em conhecimento permite a emergência de diversas aplicações inéditas, desde que haja um processamento inteligente para lidar com *big data*. O grande volume, bem como a dimensionalidade elevada dos dados acabam por impossibilitar a adoção de métodos inferenciais tradicionais, motivando o emprego de técnicas de processamento *online* de processamento de dados [Slavakis et al., 2014]. Neste contexto, Solo [2019] enfatiza a relevância atual dos algoritmos adaptativos:

“Because adaptive algorithms have short memory they can extract patterns effectively from data streams while needing little storage. In a “big data” world, this gives adaptive algorithms a powerful comparative advantage even in offline or batch settings. For this simple reason adaptive algorithms (and the simplest of them all, LMS) have a guaranteed future. ***The internet of things won’t be possible without them.***<sup>1</sup>”

Diversas transformações em sinais de interesse (como os de voz e os utilizados em comunicações digitais) são engendradas por sistemas variantes no tempo e desconhecidos. Ademais, as estatísticas dos sinais envolvidos (frequentemente não estacionários) são não raro desconhecidas, inviabilizando o projeto de estruturas de filtragem fixas. A ubiquidade de tais sistemas motiva a utilização de técnicas *online* de aprendizado, capazes de lidar de modo computacionalmente eficiente com a não estacionariedade subjacente. Particularmente, observa-se neste caso uma grande sobreposição de técnicas advindas da área de processamento adaptativo de sinais, as quais permitem tratar problemas como equalização de canais, cancelamento de eco acústico, identificação de sistemas, cancelamento de vibrações e controle adaptativo [Haykin, 2002].

Este trabalho insere-se numa subárea do processamento adaptativo de sinais, conhecida como filtragem adaptativa, a qual mantém relações estreitas com a área de aprendizado *online*.

Algoritmos de filtragem adaptativa costumam requerer um baixo custo computacional, são flexíveis e exigem pouca informação *a priori*. Desafortunadamente, tais vantagens implicam um preço, já que o processo de aprendizagem resultante é estocástico, não linear e variante no tempo.

---

<sup>1</sup>O negrito foi adicionado por nós.

Dada esta sofisticação, quaisquer análises teóricas que almejam modelar o seu processo de aprendizagem se deparam com dificuldades teóricas por vezes intransponíveis. Por esta razão, intensos esforços têm sido investidos em tais análises, pois estas oferecem diretrizes importantes para o projetista de tais algoritmos, assim como fornecem garantias de desempenho, de estabilidade e de capacidade de rastreamento.

Os algoritmos de filtragem adaptativa podem ser classificados em dois grandes grupos: os supervisionados e os não supervisionados, sendo que naqueles é necessária a aquisição de um sinal de referência, por vezes adquirido de forma engenhosa (a depender da aplicação em particular), com o qual coteja-se a saída do filtro. Já algoritmos cegos (ou não supervisionados) atuam na ausência de tal sinal de referência, como no caso de problemas de separação cega de fontes. Este trabalho enfoca alguns algoritmos pertencentes à classe das técnicas supervisionadas.

## I.2 Motivações

O amplo leque de aplicações potenciais das técnicas de filtragem adaptativa justifica os intensos esforços de pesquisa para propostas de novas variantes destes algoritmos, assim como para redução de custo computacional, robustez perante ruídos de elevada energia (ou impulsivos) e para o desenvolvimento de modelos estocásticos capazes de prover garantias para o projetista. Ademais, é sabido que as abordagens adotadas são de natureza *ad hoc*, já que as ferramentas são coletadas em um largo espectro de áreas, e escassas são as abordagens sistemáticas.

Algoritmos de filtragem adaptativa por vezes são implementados em blocos, normalmente com o propósito de reduzir a complexidade computacional, ou mesmo aumentar a taxa de convergência e o desempenho assintótico. Os dois algoritmos adaptativos mais tradicionais desta classe são o BLMS (do inglês *Block Least Mean Squares*) e o BNLMS (do inglês *Block Normalized Least Mean Squares*). O fato de a implementação eficiente destes algoritmos ser tradicionalmente a preocupação de pesquisadores oriundos da área de processamento de sinais explica porque a maior parte das referências deste trabalho é desta área, embora, como vimos, tais técnicas possam ser equivalentemente descritas como soluções de aprendizado *online*.

Para que o projetista de tais algoritmos tenha segurança de utilizá-los, é importante que haja modelos teóricos que ofereçam garantias de desempenho e de estabilidade. Assim, o projetista teria uma expectativa prévia sobre quanto bem tais algoritmos irão funcionar na prática.

Explicitadas tais motivações, podemos, em termos sucintos, dizer que este trabalho pretende aplicar um modelo estocástico para predizer características de aprendizado dos algoritmos BLMS e BNLMS.

Estruturas não lineares de processamento, em tese, apresentam maior expressividade em obter o mapeamento entrada-saída desejado. Neste trabalho, optamos por nos restringir a estruturas

lineares<sup>2</sup>, já que elas são as necessárias quando os sistemas subjacentes são lineares (como em comunicações digitais e em algumas aplicações de cancelamento de eco) e porque é possível neste caso efetuar análises teóricas mais amplas.

### I.3 Objetivo

Este trabalho tem como objetivo utilizar um modelo estocástico que divorta as características radiais das angulares dos vetores de entrada do algoritmo para analisar teoricamente os algoritmos BLMS e BNLMS. Tal modelo será capaz de caracterizar tanto a taxa de convergência, o valor máximo do passo de aprendizagem que evita divergência e o desempenho em regime permanente destes algoritmos.

### I.4 Metodologia

No propósito de modelar o desempenho dos algoritmos BLMS e BNLMS, equações recursivas relativas à matriz de autocorrelação dos desvios (os quais são a diferença entre os coeficientes adaptativos e os seus respectivos valores ótimos) serão obtidas por meio de manipulações matemáticas. A caracterização média do desempenho do algoritmo será obtida pela aplicação do operador de valor esperado. Para se evitar a necessidade de se computar momentos conjuntos de diversas variáveis aleatórias, serão empregadas algumas hipóteses estatísticas comuns na literatura. Uma delas ainda não utilizada na análise dos algoritmos adaptativos em bloco, será a principal inovação deste trabalho. Tal hipótese assume que o vetor de entrada do algoritmo adaptativo num determinado instante apresenta distribuições radiais e angulares estatisticamente independentes. Para aferição da qualidade do modelo, curvas teóricas serão cotejadas com curvas obtidas via simulações.

### I.5 Contribuições

As contribuições deste trabalho podem ser descritas em duas partes. O primeira delas é a aplicação do modelo estocástico avançado proposto por Slock [1993] para prever as habilidades de aprendizagem tanto do BLMS quanto do BNLMS. Este modelo desacopla a distribuição radial (distribuição da função de densidade de probabilidade da norma do vetor) e angular (distribuição da função de densidade de probabilidade do coeficiente angular do vetor) do vetor de entrada  $\mathbf{x}(k)$  a fim de obter novos *insights* sobre o comportamento de convergência dependente da distribuição dos autovalores da matriz de autocorrelação do sinal de entrada. A segunda é a inclusão desses

---

<sup>2</sup>As estruturas utilizadas são aqui consideradas lineares por abuso de linguagem; já que o seriam apenas se seus coeficientes fossem fixos, o que não é o caso. Dada a retroalimentação presente em seu aprendizado, o mapeamento engendrado por tais estruturas é eminentemente não linear, já que viola o princípio da superposição.

algoritmos de forma determinística (em vez de estocástica), quadro que se revelou útil para a derivação de novos algoritmos adaptativos [Gonzaga et al., 2021].

## I.6 Publicação Derivada deste Trabalho

**Gonzaga, Thiago R. P.**; Junior, Valmir dos S. N.; de Barros, Ana L. F.; Henriques, Felipe da R.; Haddad, Diego B. “*Transient Analysis of the Block Least Mean Squares Algorithm*”, *IEEE Communications Letters*, v. 25, n. 2, p. 608 - 612, Fev. 2021.

Revista com fator de impacto 3,419 e estratificada como Qualis A1 da CAPES.

## I.7 Trabalhos Relacionados

O modelo Estocástico proposto por Slock e utilizado nesta dissertação, deu origem a diversos trabalhos relacionados em temas variados na área de Filtragem Adaptativa, como pode se visto na tabela I.1 abaixo:

Referência	Titulo
[Bhutto and Antoniou, 2014]	Affine-Projection-Like Adaptive-Filtering Algorithms Using Gradient-Based Step Size
[Das and Chakraborty, 2015]	On Convergence of Proportionate-Type Normalized Least Mean Square Algorithms
[Jimaa et al., 2008]	Convergence Evaluation of a Variable Step-Size LMSE Adaptive Switching Algorithm
[Paul and Ogunfunmi, 2011]	On the Convergence Behavior of the Affine Projection Algorithm for Adaptive Filters

Tabela I.1: Tabela com referências de trabalhos relacionados.

## I.8 Notação matemática

Ao longo deste texto, o símbolo  $\mathbb{E}[\cdot]$  denota o operador de valor esperado. As variáveis escalares são denotadas por letras minúsculas e em itálico (como  $a$ ), os vetores (normalmente do tipo coluna, a menos que se diga o contrário) por letras minúsculas, em negrito e em itálico (como  $\mathbf{b}$ ) e matrizes por letras maiúsculas, em negrito e em itálico (como  $\mathbf{C}$ ). A expressão  $\text{Prob}[A]$  equivale à probabilidade de o evento  $A$  ocorrer, enquanto que  $\text{Tr}[\mathbf{X}]$  significa o traço da matriz quadrada  $\mathbf{X}$  (isto é, a soma dos elementos de sua diagonal principal).

## I.9 Estrutura deste trabalho

Este trabalho apresenta a seguinte estrutura: o Capítulo II apresenta uma introdução aos algoritmos de filtragem adaptativa. O Capítulo III trata do método de análise utilizado. O Capítulo IV, o central deste trabalho, propõe um modelo estocástico capaz de prever suas características de

aprendizado. O Capítulo V apresenta as simulações e resultados, os quais precedem o último capítulo, de conclusões e de trabalhos futuros.

## Capítulo II Algoritmos de Filtragem Adaptativa

### II.1 Estruturas Transversais Adaptativas

Filtros são estruturas responsáveis por extrair de um sinal uma determinada informação de interesse. A estrutura mais popular para os algoritmos de filtragem adaptativa é a chamada estrutura transversal, a qual atenta para uma *janela* de  $N \in \mathbb{N}$  amostras do sinal de entrada  $x(k) \in \mathbb{R}$ , dando origem ao sinal de saída  $y(k) \in \mathbb{R}$ :

$$y(k) = w_0(k)x(k) + w_1(k)x(k-1) + \dots + w_{N-1}(k)x(k-N+1) = \sum_{i=0}^{N-1} w_i(k)x(k-i), \quad (\text{II.1})$$

onde os pesos  $\{w_i(k)\}_{i=0}^{N-1}$ , quando adaptativos, variam no tempo (daí a necessidade de se incluir o índice  $k$ ).

Quando as estatísticas do sinal de entrada e de saída são conhecidas, é possível se efetuar um projeto de tais coeficientes, dando origem a um processamento que privilegia determinadas frequências do sinal de entrada, em detrimento de outras.

Como exemplo didático de tal tipo de projeto, podemos projetar um filtro digital passa-baixas com uma frequência de corte em  $\omega_0$  rad/s, cuja magnitude em frequência seja aproximadamente unitária em todas as frequências inferiores a  $\omega_0$  e cuja magnitude seja aproximadamente nula nas demais frequências por meio da transformada inversa de Fourier:

$$w_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(w) e^{j\omega n} dw = \frac{1}{2\pi} \int_{-\omega_0}^{\omega_0} e^{j\omega n} dw = \frac{\operatorname{sen}(\omega_0 n)}{\pi n}, \quad (\text{II.2})$$

onde  $H(w)$  denota o valor da transformada de Fourier na frequência  $w$  (em rad/s) e  $j$  significa a raiz quadrada de -1. Logo, o filtro definido por

$$w_n = \begin{cases} \frac{\operatorname{sen}(0.3n)}{\pi n}, & \text{para } |n| \leq 25 \\ 0, & \text{no resto,} \end{cases}, \quad (\text{II.3})$$

apresentado na Figura II.1, apresenta a característica frequencial da Figura II.2, cujo comportamento é similar ao pretendido.

O projeto de filtros fixos constitui uma área bastante madura de processamento digital de sinais,

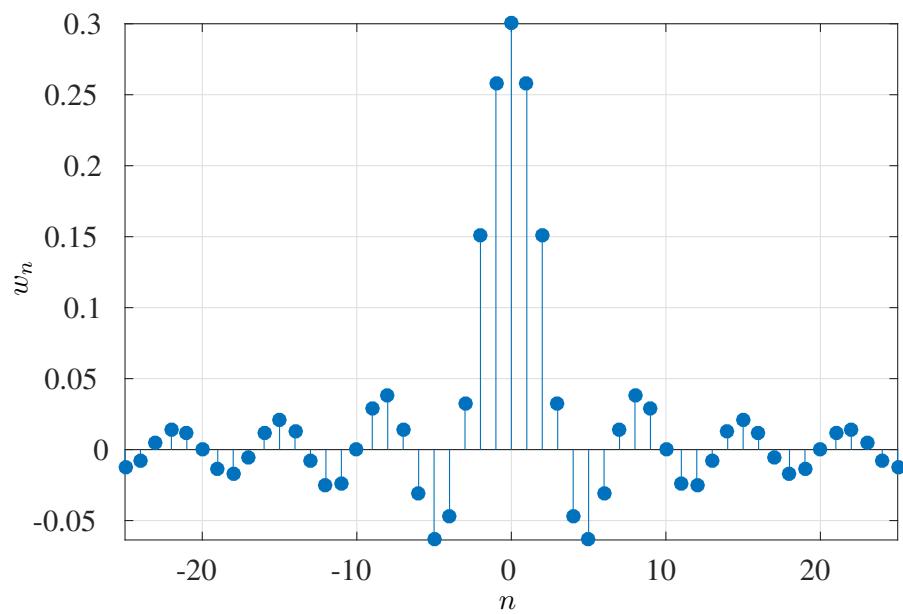


Figura II.1: Filtro passa-baixas projetado.

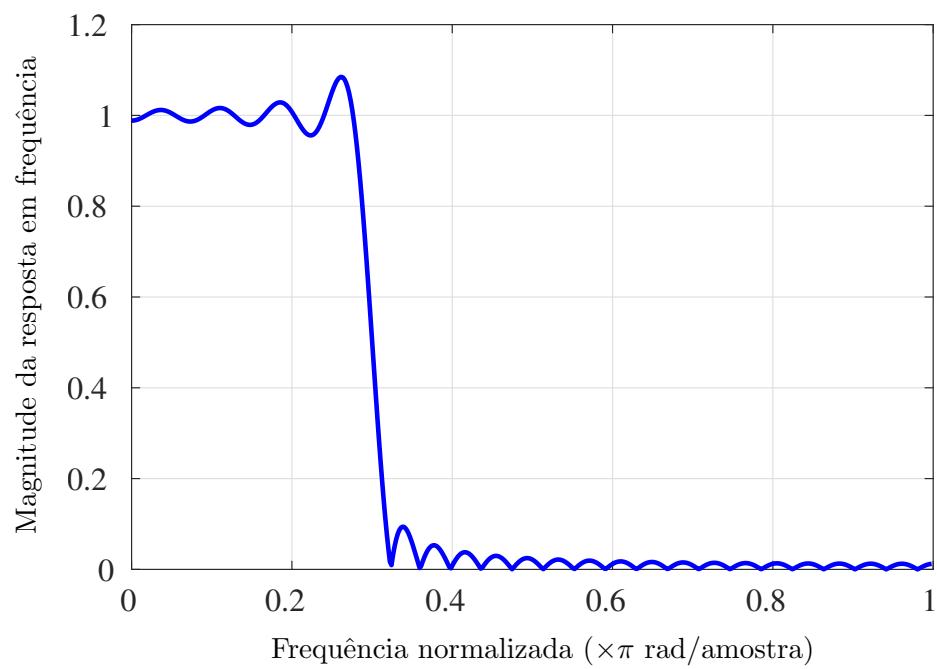


Figura II.2: Resposta frequencial do filtro projetado no exemplo.



Figura II.3: Exemplo de um problema de equalização de canal. A função de transferência  $H(z)$  emula a distorção provocada pelo canal. Cabe ao filtro  $W(z)$  desfazer tal distorção (ou seja, efetuar uma equalização).

com técnicas muito mais sofisticadas do que o exemplo apresentado [Diniz et al., 2010].

Ao longo deste trabalho, assumiremos que os filtros adaptativos assumem a estrutura transversal da Equação (II.1). Entretanto, nas aplicações de interesse, não há informação *a priori* que permita o projeto único de um filtro capaz de lidar com todos os possíveis cenários. Desafortunadamente, cada cenário exige uma solução particular.

Como exemplo de tal problema, considere um sistema de comunicação digital, no qual um filtro<sup>1</sup>  $H(z)$ , associado ao canal, distorce um sinal transmitido  $s(k)$ , conforme ilustrado pela Figura II.3. Na classe de problemas conhecida como *equalização de canal*, caberia a um filtro transversal com coeficientes  $w_i(k)$  (para  $i \in \{0, 1, \dots, N - 1\}$ ) reconstruir aproximadamente, em sua saída, uma versão atrasada do sinal transmitido  $s(k)$ . Logrando êxito em tal intento, podemos escrever a saída do filtro como:

$$y(k) \approx cs(k - \Delta), \quad (\text{II.4})$$

onde  $c \in \mathbb{R}$  é uma constante de escala e  $\Delta \in \mathbb{N}$  é o atraso introduzido pelo equalizador, o qual deve reverter o efeito do canal.

Como na prática o canal varia em cada realização do sistema de comunicação,  $H(z)$  não é conhecido *a priori*. Ademais, ele pode ser variante no tempo. Neste cenário, não é possível recorrer a um projeto bem especificado de filtro fixo.

Uma solução possível para este desafio de equalização consiste em se inicializar o filtro adaptativo  $\mathbf{w}(k) \in \mathbb{R}^N$ , com o vetor  $\mathbf{w}(0)$  coletando os  $N$  coeficientes adaptativos iniciais:

$$\mathbf{w}(0) = [w_0(0) \ w_1(0) \ w_2(0) \ \dots \ w_{N-1}(0)]^T. \quad (\text{II.5})$$

Tal inicialização dificilmente resultaria num processamento ótimo da estrutura de equalização, já que não atenta para as especificidades da configuração em particular. Neste contexto, é de grande valia o que poderíamos chamar de *princípio da adaptação*, a partir do qual os coeficientes são atualizados recursivamente de modo a progressivamente se aproximarem de uma solução que atenda aos requisitos da aplicação em particular:

$$\mathbf{w}(k + 1) = \mathbf{w}(k) + \Delta \mathbf{w}(k), \quad (\text{II.6})$$

---

<sup>1</sup>Aqui, a transformada no domínio  $\mathcal{Z}$  de um sinal no domínio do tempo é denotada pela letra maiúscula correspondente.

onde  $\Delta\mathbf{w}(k) \in \mathbb{R}^N$  é um termo de correção vetorial que efetua um ajuste (na média, benéfico) da solução atual  $\mathbf{w}(k)$ . O fato de tal ajuste se dar continuamente confere ao processo de filtragem a sua peculiaridade adaptativa, de modo que na  $k$ -ésima iteração o vetor  $\mathbf{w}(k)$  possa ser escrito como

$$\mathbf{w}(k) \triangleq \begin{bmatrix} w_0(k) & w_1(k) & w_2(k) & \dots & w_{N-1}(k) \end{bmatrix}^T. \quad (\text{II.7})$$

No caso do CMA (do inglês *Constant Modulus Algorithm*), o termo de atualização é  $\Delta\mathbf{w}(k) = \beta\mathbf{x}(k)y(k)(\bar{\gamma} - y^2(k))$ , o que especifica equação de atualização do CMA [Schniter and Johnson, 1999]:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \beta\mathbf{x}(k)y(k)(\bar{\gamma} - y^2(k)), \quad (\text{II.8})$$

onde  $\beta \in \mathbb{R}_+$  denota o passo de aprendizagem,  $\bar{\gamma}$  é dependente da modulação escolhida e  $\mathbf{x}(k)$  é o chamado vetor de entrada na  $k$ -ésima iteração:

$$\mathbf{x}(k) \triangleq \begin{bmatrix} x(k) & x(k-1) & \dots & x(k-N+1) \end{bmatrix}^T. \quad (\text{II.9})$$

O fator de aprendizagem  $\beta$  deve ser escolhido de modo judicioso, porque seu valor tem grande impacto na taxa de convergência, no desempenho em regime permanente, na característica de rastreamento e mesmo na estabilidade.

### II.1.1 Exemplo ilustrativo de equalização

Para concretizarmos alguns conceitos importantes, seja o caso (didático) no qual o sinal transmitido assume valores  $+1$  ou  $-1$ , de modo equiprovável. Assim:

$$s(k) = \begin{cases} 1, & \text{com probabilidade } 0,5 \\ -1, & \text{com probabilidade } 0,5 \end{cases}. \quad (\text{II.10})$$

Assumamos que tal sinal seja modificado por um canal linear, cuja função de transferência, no domínio da transformada  $\mathcal{Z}$ , seja

$$H(z) = 2,0474z^{-1} + 0,5411z^{-2} - 1,0607z^{-3} - 1,2009z^{-4}, \quad (\text{II.11})$$

e que o sinal recebido  $x(k)$  seja corrompido por um ruído aditivo branco, gaussiano, de média zero e variância  $10^{-6}$ . Observe que na prática o receptor somente tem acesso a  $x(k)$ , desconhecendo por conseguinte tanto  $H(z)$  quanto a variância do ruído aditivo.

Utilizaremos para equalização de canal o algoritmo CMA (vide Equação (II.8)), com  $\bar{\gamma} = 1$  e

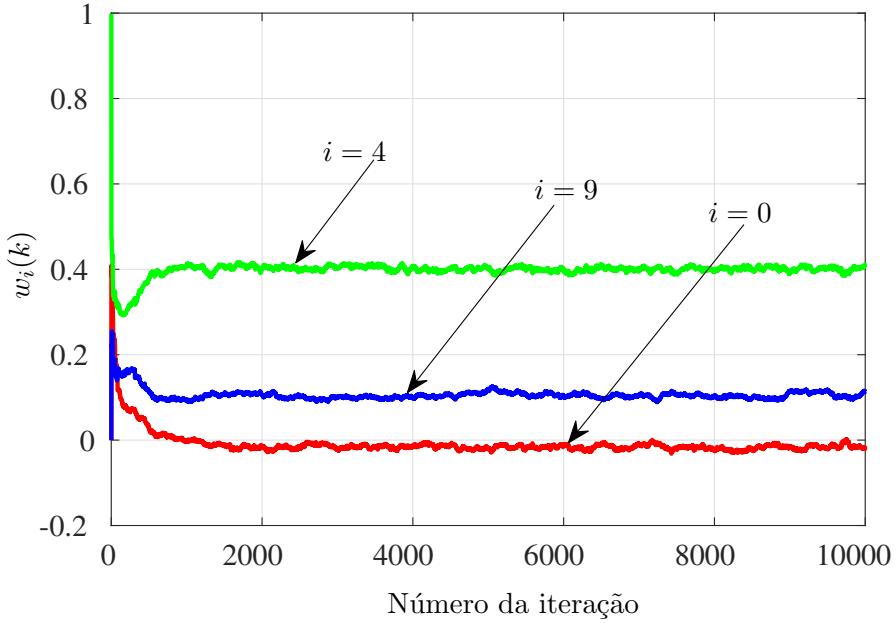


Figura II.4: Evolução de alguns coeficientes  $w_i(k)$  do exemplo ilustrativo de equalização, para  $i \in \{0, 4, 9\}$  e apenas uma execução do algoritmo. O algoritmo de atualização utilizado é o CMA, com  $\beta = 10^{-3}$ .

um filtro  $\mathbf{w}(k)$  com<sup>2</sup>  $N = 10$  coeficientes, inicializado da seguinte forma:

$$\mathbf{w}(0) = \begin{bmatrix} \mathbf{0}_{N/2-1} \\ 1 \\ \mathbf{0}_{N/2} \end{bmatrix}, \quad (\text{II.12})$$

onde  $\mathbf{0}_M$  denota o vetor nulo (do tipo coluna) com  $M$  elementos.

A Figura II.4 apresenta a evolução dos coeficientes  $w_0(k)$ ,  $w_4(k)$  e  $w_9(k)$  ao longo das iterações, com a atualização sendo efetuada pelo algoritmo CMA com  $\beta = 10^{-3}$  e  $\bar{\gamma} = 1$ . O coeficiente  $w_4(k)$  é inicializado com o valor unitário, enquanto que os demais são inicializados com zeros. Observe que a evolução de tais coeficientes é ruidosa. Tal característica é típica de resultados oriundos de uma única execução do algoritmo, com a variabilidade destas curvas normalmente aumentando quando o fator de aprendizagem  $\beta$  assume valores maiores.

Sendo resultado de uma única execução do algoritmo, as curvas apresentadas na Figura II.4 não permitem a caracterização do comportamento médio da evolução destes coeficientes. Normalmente, tal caracterização, quando feita de modo empírico (isto é, simulado), requer a realização de diversos ensaios independentes de Monte Carlo, com cada um destes ensaios correspondendo a uma realização particular dos processos estocásticos envolvidos (ou seja, tanto do sinal  $s(k)$  quanto do ruído aditivo). Quanto maior o número de ensaios de Monte Carlo, mais suaves são as curvas obtidas. Assim, utilizando-se 20 ensaios de Monte Carlo, podemos verificar a variação do compor-

<sup>2</sup>A escolha do tamanho do filtro  $N$  foi apenas ilustrativa.

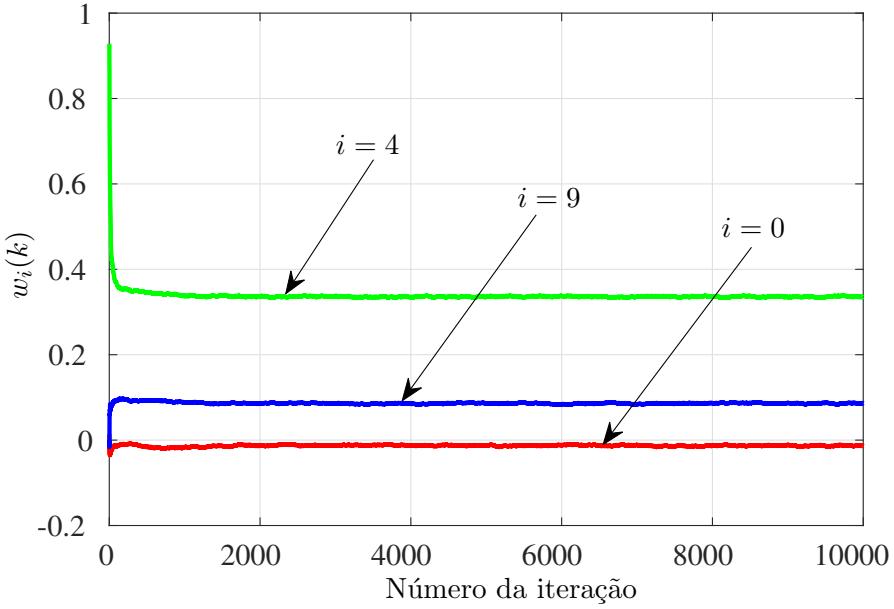


Figura II.5: Evolução *média* de alguns coeficientes  $w_i(k)$  do exemplo ilustrativo de equalização, para  $i \in \{0, 4, 9\}$ , obtida com 20 ensaios independentes de Monte Carlo. O algoritmo de atualização utilizado é o CMA, com  $\beta = 10^{-3}$ .

tamento *médio* dos coeficientes na Figura II.5.

Observe que a evolução dos coeficientes vista na Figura II.5 não permite inferir quão bem o algoritmo está funcionando. Neste caso, é de grande utilidade a utilização de uma *métrica de aferição de desempenho*.

Como os sistemas desconhecido  $H(z)$  e o adaptativo  $W(z)$  encontram-se em série (vide Figura II.3), tal métrica é mais fácil de ser calculada apelando-se ao sistema *equivalente*  $C(z)$ , obtido pela convolução entre  $H(z)$  e  $W(z)$ . Embora em termos rigorosos, a utilização da convolução para o cômputo do sistema equivalente seja possível apenas para sistemas lineares e invariantes no tempo, e portanto inapropriada para sistemas variantes no tempo (afinal,  $W(z)$  é um filtro adaptativo), o fato de os coeficientes adaptativos evoluírem numa dinâmica bem mais lenta do que a do sinal de entrada garante ser esta uma boa aproximação. Afinal, pode-se assumir que os coeficientes adaptativos são praticamente fixos, num curto intervalo de tempo, caso o algoritmo opere numa região de estabilidade. Assim, a Figura II.6 apresenta a função de transferência equivalente na iteração  $k = 10000$ . Observe a emergência de um coeficiente bem maior do que os outros em  $n = 5$ , significando que o sistema global canal + equalizador resulta num atraso de 5 amostras. Idealmente, pela Equação (II.4), a saída deveria conter apenas um atraso (no caso,  $\Delta = 5$ ), porém observe que os demais coeficientes do sistema global não são nulos. O fato de tais coeficientes não serem nulos significa que o equalizador não consegue eliminar totalmente a interferência inter-simbólica (ISI, do inglês *intersymbol interference*).

A ISI pode ser computada dividindo-se a energia do maior coeficiente pela energia dos demais

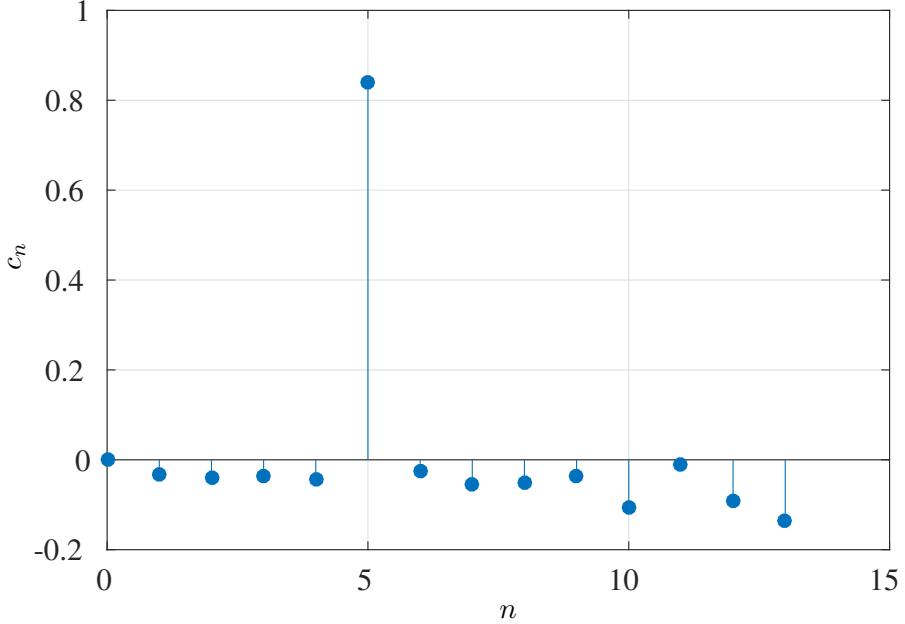


Figura II.6: Função de transferência equivalente, para a iteração  $k = 10^4$  do algoritmo CMA, com  $\beta = 10^{-3}$ . A configuração é a mesma da Figura II.4.

coeficientes da resposta ao impulso global  $C(z)$ . Assim, definindo-se

$$i_{\max} = \arg \max_i |c(i)|, \quad (\text{II.13})$$

podemos descrever a ISI como

$$\text{ISI} = \mathbb{E} \left[ \frac{c_{i_{\max}}^2}{\sum_{i=0, i \neq i_{\max}}^{\bar{M}-1} c_i^2} \right], \quad (\text{II.14})$$

onde  $\bar{M}$  é o tamanho do filtro global  $C(z)$ . A ISI é uma métrica objetiva de aferição de desempenho, a qual, como muitas outras, não raro é apresentada em dB:

$$\text{ISI (dB)} = 10 \log_{10} \left\{ \mathbb{E} \left[ \frac{c_{i_{\max}}^2}{\sum_{i=0, i \neq i_{\max}}^{\bar{M}-1} c_i^2} \right] \right\}. \quad (\text{II.15})$$

Observe ademais que a ISI depende do valor médio de grandezas que são funções quadráticas dos coeficientes  $w_i(k)$ . Isso significa que a ISI também depende de estatísticas que não são de primeira ordem. A predição de estatísticas de segunda ordem, por exemplo, é de muito interesse em filtragem adaptativa, porque a variância (que é uma medida de variabilidade) depende de estatísticas de segunda ordem.

Como último gráfico deste exemplo ilustrativo, a Figura II.7 apresenta a evolução da ISI (em dB) no exemplo considerado. Observa-se um aumento contínuo da métrica de desempenho ao longo das três curvas diferentes (cada uma com um valor específico de  $\beta$ ). Observa-se que a escolha de um  $\beta$  menor torna o aprendizado mais lento. Também cabe ressaltar a existência de duas regiões

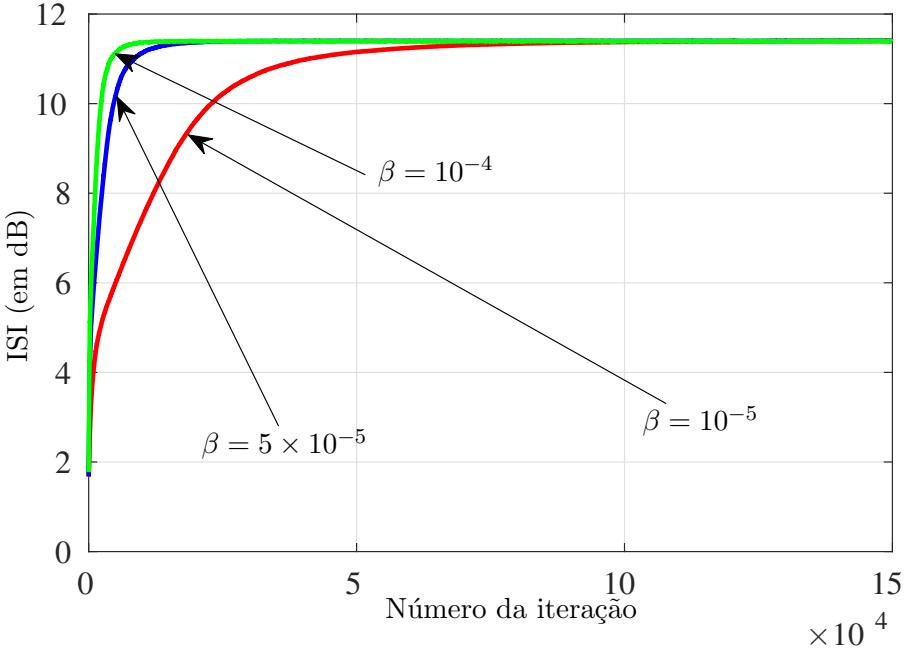


Figura II.7: Evolução média da ISI (em dB), obtida com 10 ensaios independentes de Monte Carlo. A atualização foi feita pelo algoritmo CMA, com diferentes valores de  $\beta$ .

na dinâmica de aprendizado: a de *transiente*, durante a qual o algoritmo aprende progressivamente, e a de *regime permanente* (ou assintótica), caracterizada por um patamar constante, atingido *após* o regime de transiente. Cumpre notar que independentemente da escolha de  $\beta$ , o patamar obtido no regime permanente é quase o mesmo. Esse fato não é geralmente observado em filtragem adaptativa; normalmente, a escolha de valores mais elevados de  $\beta$  aumenta a oscilação dos coeficientes adaptativos em regime permanente, contribuindo para degradação de seu desempenho assintótico.

## II.2 Gradiente Estocástico

O exemplo de equalização visto anteriormente trata de uma configuração não supervisionada, já que não é necessário nenhum sinal de referência para proceder à equalização. Em sistemas de comunicação digital, tal sinal de referência poderia ser obtido mediante apelo a uma sequência de treinamento, pré-acordada entre o transmissor e o receptor, que permitiria ao filtro adaptativo conhecer qual deveria ser a sua resposta ideal. Neste caso, o algoritmo assumiria uma configuração supervisionada, que é o foco deste trabalho.

Assim, um algoritmo supervisionado tem conhecimento de um sinal desejado  $d(k) \in \mathbb{R}$ , do qual sua saída

$$y(k) = \mathbf{w}^T(k) \mathbf{x}(k) \quad (\text{II.16})$$

deveria se aproximar. Estando disponíveis as amostras  $d(k)$  e a saída  $y(k)$ , é possível o cômputo

de uma amostra do sinal de erro  $e(k) \in \mathbb{R}$ :

$$e(k) = d(k) - y(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k), \quad (\text{II.17})$$

o qual deve alimentar o algoritmo adaptativo. O fato de este erro ensejar modificações do vetor  $\mathbf{w}(k)$  é o responsável pela retroalimentação existente no aprendizado do filtro (afinal, o erro depende de  $y(k)$ , o qual também depende do vetor de coeficientes adaptativos). Esta retroalimentação explica a grande dificuldade de se modelar teoricamente o comportamento dos algoritmos de filtragem adaptativa, já que mesmo algoritmos simples acabam por apresentar uma dinâmica de aprendizado bastante sofisticada.

Parte significativa dos algoritmos de filtragem adaptativa podem ser obtidos mediante o paradigma do *gradiente estocástico*. Neste paradigma, existe uma função global do erro que geralmente se deseja minimizar. Assim, podemos descrever *em termos intuitivos* o problema que o algoritmo adaptativo *tenta* resolver como

$$\mathbf{w}(k+1) \approx \min_{\mathbf{w}} \bar{\mathcal{F}}\{e[\mathbf{w}]\}, \quad (\text{II.18})$$

para  $k = 0, 1, \dots, \infty$ , onde  $\bar{\mathcal{F}}\{e[\mathbf{w}]\}$  é o valor esperado de uma função custo. No caso em que a função custo é o erro quadrático médio, temos que desejamos encontrar o vetor  $\mathbf{w}$  que minimiza  $\frac{1}{2}\mathbb{E}[e^2(k)]$ . Quando temos acesso a uma grande quantidade de dados ( $K$  amostras, digamos), podemos associar a esta função custo a seguinte função custo *empírica*:

$$\bar{\mathcal{F}}_{\text{emp}}[\mathbf{w}] = \frac{1}{2K} \sum_{k=0}^{K-1} e^2(k), \quad (\text{II.19})$$

onde  $e(k)$  depende de  $\mathbf{w}$ .

A minimização de (II.19) pode ser empreendida pela técnica de mínimos quadrados (LS, do inglês *least squares*), a qual entretanto apresenta alguns inconvenientes, como o de exigir alta complexidade computacional, bem como assumir que a solução ótima do vetor  $\mathbf{w}$  é fixa ao longo de todas as amostras. Na prática, entretanto, esta última restrição é muito forte, já que diversos sistemas reais são variantes no tempo. Uma forma relativamente simples (mas com repercussões teóricas e práticas bastante complexas) de se contornar estes problemas consiste em se recorrer a uma função custo estocástica, a qual utiliza apenas a amostra atual (corrente) do sinal de erro. Assim, a versão estocástica da função custo (II.19) pode ser escrita como

$$\bar{\mathcal{F}}_{\text{est}}[\mathbf{w}] = \frac{1}{2}e^2(k). \quad (\text{II.20})$$

A técnica do gradiente estocástico efetua uma atualização recursiva no vetor de coeficientes adap-

tativos  $\mathbf{w}(k)$  segundo o seguinte padrão:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \beta \nabla_{\mathbf{w}(k)} \bar{\mathcal{F}}_{\text{est}}[\mathbf{w}(k)], \quad (\text{II.21})$$

onde  $\nabla_{\mathbf{w}} f$  denota o gradiente de  $f$  com relação a  $\mathbf{w}$ ; assim:

$$\nabla_{\mathbf{w}} \bar{\mathcal{F}}_{\text{est}}[\mathbf{w}(k)] = \begin{bmatrix} \frac{\partial \bar{\mathcal{F}}_{\text{est}}[\mathbf{w}(k)]}{\partial w_0(k)} \\ \frac{\partial \bar{\mathcal{F}}_{\text{est}}[\mathbf{w}(k)]}{\partial w_1(k)} \\ \vdots \\ \frac{\partial \bar{\mathcal{F}}_{\text{est}}[\mathbf{w}(k)]}{\partial w_{N-1}(k)} \end{bmatrix}. \quad (\text{II.22})$$

Observe que (II.21) pode ser considerada um caso particular de (II.6) quando

$$\Delta \mathbf{w}(k) = -\beta \nabla_{\mathbf{w}(k)} \bar{\mathcal{F}}_{\text{est}}[\mathbf{w}(k)]. \quad (\text{II.23})$$

Ademais, cabe ressaltar que o gradiente  $\nabla_{\mathbf{w}(k)} \bar{\mathcal{F}}_{\text{est}}[\mathbf{w}(k)]$  sinaliza a direção de subida mais íngreme da função custo estocástica, e que o fato de esta função depender de apenas uma amostra significa que tal gradiente é degradado por um ruído (daí a expressão gradiente *estocástico*). O sinal de negativo em  $\Delta \mathbf{w}(k)$  significa que estamos indo na direção contrária à de subida mais íngreme, devido ao fato de almejarmos minimizar a função custo. Por fim, o fator de aprendizagem  $\beta$  neste contexto pode ser associado ao fato de a atualização estar mais preocupada com a direção fornecida pelo gradiente do que propriamente com sua magnitude.

Quando a função custo estocástica escolhida é o erro quadrático instantâneo (vide Equação (II.20)) podemos, utilizando (II.17) e (II.21), especificar a equação de atualização do algoritmo LMS, o mais tradicional dos algoritmos de filtragem adaptativa:

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) - \beta \nabla_{\mathbf{w}(k)} \left[ \frac{1}{2} e^2(k) \right] = \mathbf{w}(k) - \beta e(k) \nabla_{\mathbf{w}(k)} [e(k)] \\ &= \mathbf{w}(k) - \beta e(k) \nabla_{\mathbf{w}(k)} [d(k) - \mathbf{w}^T(k) \mathbf{x}(k)] \\ &= \mathbf{w}(k) + \beta e(k) \nabla_{\mathbf{w}(k)} [\mathbf{w}^T(k) \mathbf{x}(k)] \\ &= \mathbf{w}(k) + \beta \mathbf{x}(k) e(k). \end{aligned} \quad (\text{II.24})$$

Assim, o gradiente estocástico é um paradigma de otimização iterativa de primeira ordem que contempla uma vasta classe de algoritmos adaptativos. Diferentes escolhas da função custo estocástica  $\bar{\mathcal{F}}_{\text{est}}[\mathbf{w}]$ , quando acopladas à Equação (II.21), dão origem a diversos algoritmos propostos na literatura, conforme mostra a Tabela II.1. A escolha da função custo tem grande influência no desempenho do algoritmo adaptativo resultante, podendo conferir a ele robustez perante ruído impulsivo, aumento na taxa de convergência, melhor desempenho em regime permanente, incor-

$\bar{\mathcal{F}}_{\text{est}}[\mathbf{w}]$	Referência
$ e(k) $	Verhoeckx and Claasen [1984]
$e^{2n+1}(k)$	Walach and Widrow [1984]
$\gamma e(k) + (1 - \gamma)e^3(k)$	Chambers et al. [1994]
$e^2(k) + \gamma \ \mathbf{w}(k)\ _0$	Gu et al. [2009]
$\frac{1}{\gamma} \tanh \left[ \frac{\gamma}{2} \left( \frac{e(k)}{\ \mathbf{x}(k)\ } \right)^2 \right]$	Song et al. [2013]
$\frac{1}{2\gamma} \ln \left[ 1 + \gamma \left( \frac{e(k)}{\ \mathbf{x}(k)\ } \right)^2 \right]$	Song et al. [2013]
$e^2(k) - \frac{1}{\gamma} \ln (1 + \gamma e^2(k))$	Sayin et al. [2014]
$ e(k)  - \frac{1}{\gamma} \ln (1 + \gamma  e(k) )$	Sayin et al. [2014]
$\int_1^2 \lambda(\gamma)  e(k) ^\gamma d\gamma$	Zayyani [2014]

Tabela II.1: Algumas funções custo utilizadas na literatura para a derivação de algoritmos de filtragem adaptativa utilizando gradiente estocástico.  $\|\mathbf{w}\|_0$  denota uma aproximação da pseudo-norma  $\ell_0$ ,  $n \in \mathbb{N}$  e  $\lambda(\gamma)$  é uma função densidade de probabilidade válida com suporte no intervalo  $(1,2)$ .

puração de informação *a priori* acerca da esparsidade da função de transferência a identificar, etc.

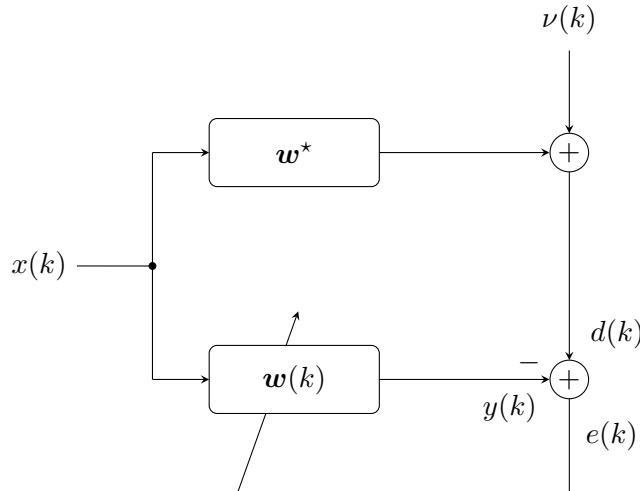


Figura II.8: Diagrama de blocos de um algoritmo adaptativo atuando na identificação de sistemas.

Nesta dissertação, estamos essencialmente interessados em aplicações de identificação de sistemas, a qual nos permite a construção de modelos matemáticos importantes para a predição de sistemas dinâmicos, conforme ilustra a Figura II.13. Observe que nesta configuração o sinal de entrada é compartilhado entre a função desconhecida a identificar (armazenada nos coeficientes do vetor  $\mathbf{w}^* \in \mathbb{R}^N$ ), ao contrário do que ocorre no cenário de equalização. Ademais, cabe observar na Figura II.13 que o sinal de referência  $d(k)$  encontra-se corrompido pelo ruído aditivo  $\nu(k)$ , podendo ser expresso pela seguinte regressão linear nos parâmetros:

$$d(k) = [\mathbf{w}^*]^T \mathbf{x}(k) + \nu(k). \quad (\text{II.25})$$

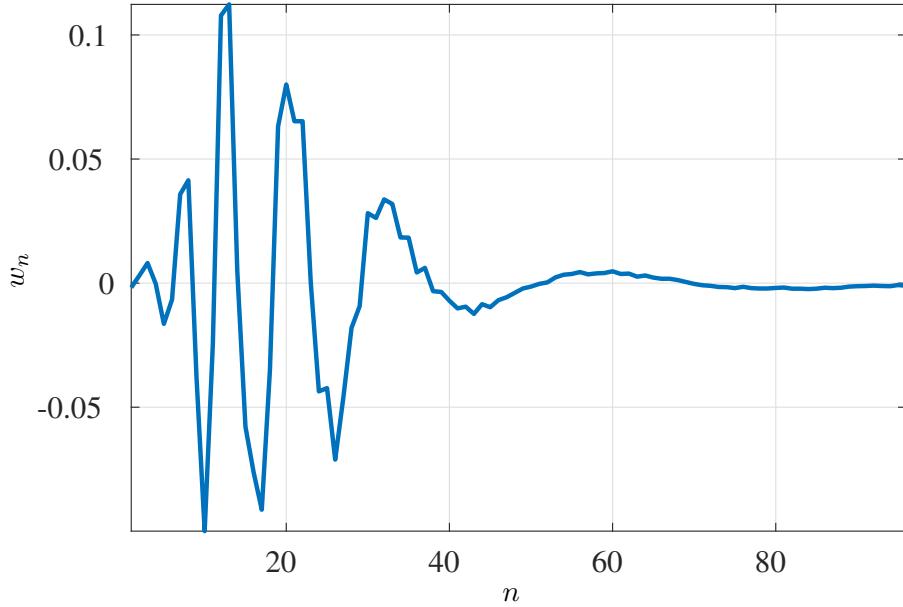


Figura II.9: Função de transferência considerada no exemplo ilustrativo.

A aplicação de identificação de sistemas tem bastante apelo prático. No caso de cancelamento de eco acústico (essencial para a inteligibilidade de sistemas de telecomunicação viva-voz), cabe ao algoritmo de identificação modelar o acoplamento acústico entre a caixa de som e o microfone. Caso esta identificação seja bem sucedida, é possível digitalmente eliminar do sinal registrado no microfone a componente oriunda da retroalimentação acústica da caixa de som, responsável pelo eco. Observe neste caso que o sistema tem acesso ao sinal de entrada  $x(k)$  para efetuar a identificação, pois este seria justamente o sinal enviado para a caixa de som. Ademais, neste cenário, a solução ótima para o vetor  $\mathbf{w}(k)$  (em termos de minimização do erro quadrático médio) quando o ruído aditivo é gaussiano é justamente a planta desconhecida  $\mathbf{w}^*$ .

### II.2.1 Exemplo ilustrativo de identificação

Consideremos neste exemplo o caso em que a função de transferência a se identificar é o segundo modelo de TSG [2004], obtido a partir de experimentos em redes digitais reais. A função de transferência apresenta 96 coeficientes, sendo apresentada na Figura II.10.

A evolução de alguns coeficientes adaptativos para o caso em que estes são adaptados de acordo com o algoritmo LMS com um fator de aprendizagem  $\beta = 10^{-2}$ , um sinal de entrada branco, gaussiano, de média zero e variância unitária e um ruído aditivo  $\nu(k)$  branco, gaussiano, de média zero e de variância  $\sigma_\nu^2 = 10^{-6}$  é mostrada na Figura II.10. Nesta figura, pode-se observar que os coeficientes convergem para seus respectivos valores ótimos, na média. Isso ilustra uma propriedade desejável do algoritmo LMS: a não polarização em regime permanente, uma característica já devidamente caracterizada, em termos teóricos, na literatura [Diniz et al., 2010]. Em outras palavras, em termos de comportamento médio (ou seja, de primeira ordem), os coeficientes do LMS oscilam

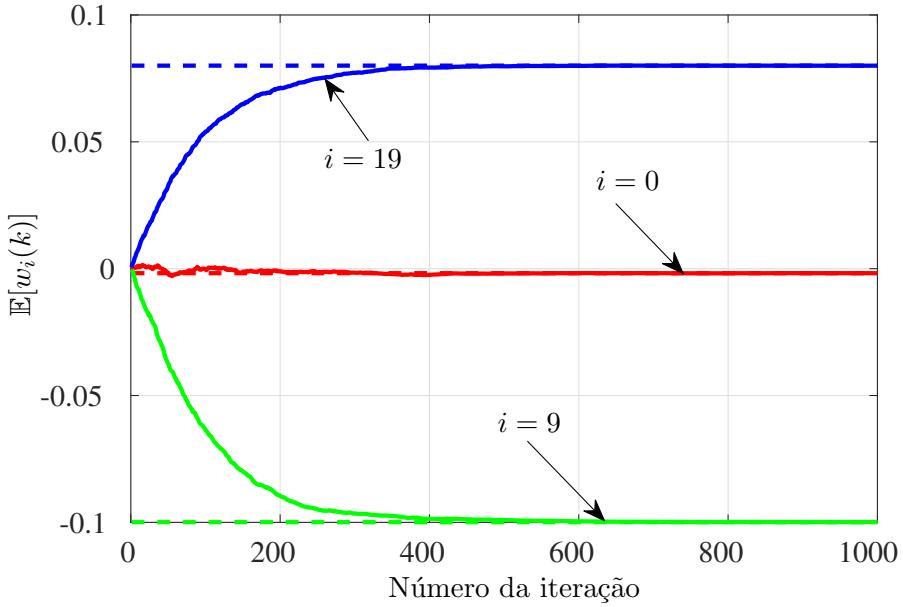


Figura II.10: Evolução média de alguns coeficientes adaptativos, obtida com 100 ensaios independentes de Monte Carlo. As linhas tracejadas horizontais sinalizam os valores respectivos da função de transferência ótima.

em torno de seus valores ótimos quando em regime permanente.

Uma métrica de aferição de desempenho bastante utilizada é o erro quadrático médio (MSE, do inglês *mean squared error*), descrito por

$$\text{MSE} = \mathbb{E} [e^2(k)]. \quad (\text{II.26})$$

A Figura II.11 apresenta a evolução do MSE (Um caso particular de estatística de segunda ordem pois utiliza momentos de segunda ordem dos coeficientes adaptativos) para a mesma configuração da Figura II.10, exceto pelo fato de que três diferentes valores de  $\beta$  foram utilizados. Observa-se que valores mais altos de  $\beta$  aceleraram o processo de convergência, embora degradem o desempenho assintótico, já que acabam por reforçar as oscilações dos coeficientes adaptativos.

Outra métrica de aferição de desempenho bastante popular (e também associada a estatísticas de segunda ordem) é o desvio quadrático médio (MSD, do inglês *mean square deviation*), descrito por

$$\text{MSD} = \mathbb{E} [\|\mathbf{w}^* - \mathbf{w}(k)\|^2]. \quad (\text{II.27})$$

A Figura II.12 apresenta a evolução do MSD no mesmo cenário considerado na Figura II.11, exceto pelo fato de que  $\beta$  assume o valor de  $2 \times 10^{-3}$  e o resultado do aprendizado quando o filtro é submetido a um sinal de entrada colorido é também acrescentado. O sinal de entrada colorido foi obtido pela convolução de um ruído branco, gaussiano e de variância unitária pelo filtro  $H(z) = \frac{1}{1-0,8z^{-1}}$ . Cabe observar o grande impacto que as propriedades estatísticas do sinal de

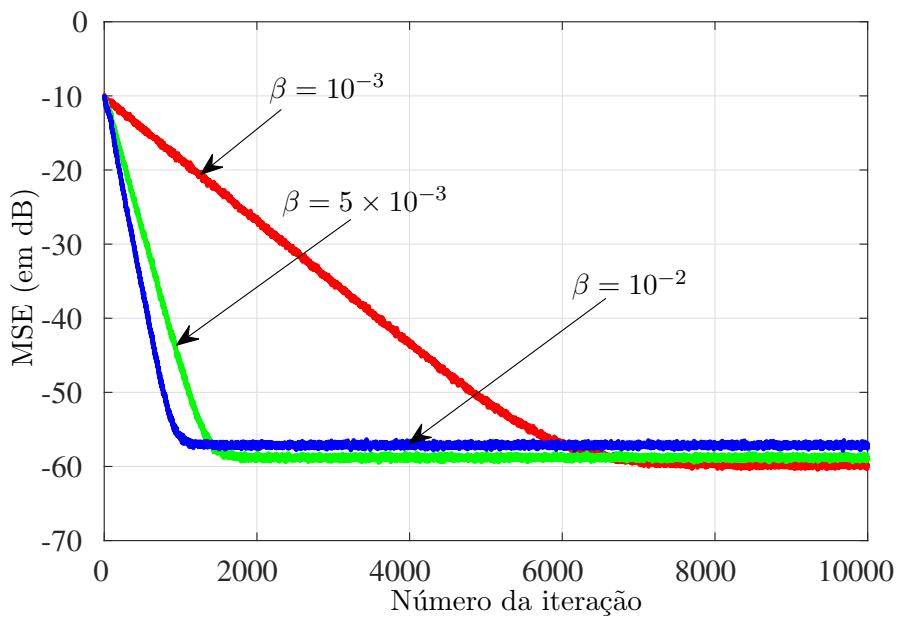


Figura II.11: Evolução do MSE, computada com  $10^3$  ensaios independentes de Monte Carlo, para diferentes valores de  $\beta$ .

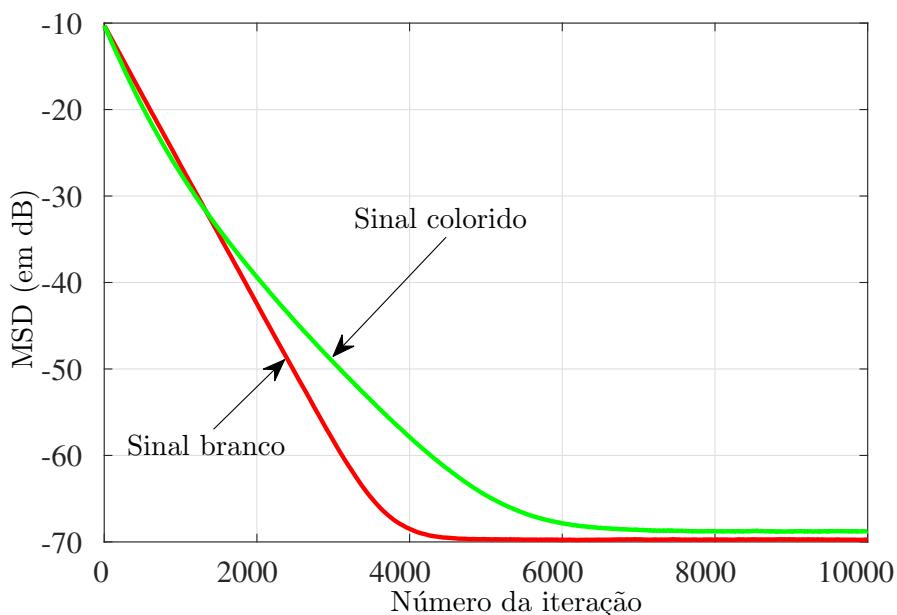


Figura II.12: Evolução do MSD do LMS com  $\beta = 2 \times 10^{-3}$ , computada com  $10^3$  ensaios independentes de Monte Carlo, para sinal de entrada branco e sinal de entrada colorido com filtro  $H(z) = \frac{1}{1-0.8z^{-1}}$ .

entrada apresenta no desempenho do algoritmo. Normalmente, verifica-se que o colorimento do sinal de entrada se traduz numa taxa de convergência mais lenta do algoritmo. Intuitivamente, podemos atribuir tal lentidão ao fato de que a inovação contida em cada novo vetor de entrada é menor, dado que as suas amostras são correlacionadas com as de vetores de entrada pregressos, os quais foram utilizados para adaptação.

### II.3 Derivação de Algoritmos de Filtragem Adaptativa via Princípio da Distorção Mínima

Uma ferramenta matemática muito útil para a obtenção de novos algoritmos de filtragem adaptativa consiste em interpretá-los sob a ótica do princípio da distorção mínima, o qual atualiza o filtro de modo conservador, ou seja, sem perturbar em demasia a última solução. Por vezes, a depender dos problemas de otimização apresentados, o mesmo algoritmo de filtragem adaptativa pode ser obtido, seja pela otimização via gradiente estocástico, seja como solucionador de um problema de otimização de perturbação mínima.

Normalmente, quando definimos um problema de otimização com restrições lineares para efeitos de gerar um algoritmo de filtragem adaptativa, a técnica utilizada para obtenção de sua equação de atualização é a dos multiplicadores de Lagrange. Tais problemas com restrições por vezes carregam uma interpretação geométrica, a qual pode ser útil para elucidar algumas características do algoritmo.

Uma grandeza muito utilizada nesta classe de problemas de otimização é o chamado erro *a posteriori*  $e_p(k)$ , o qual reutiliza no vetor obtido  $\mathbf{w}(k+1)$  os últimos dados  $\{\mathbf{x}(k), d(k)\}$  que foram utilizados para determiná-lo:

$$e_p(k) \triangleq d(k) - \mathbf{w}^T(k+1)\mathbf{x}(k). \quad (\text{II.28})$$

Como exemplo de utilização da técnica de otimização que utiliza o princípio da distorção mínima, seja o seguinte problema de otimização:

$$\min_{\mathbf{w}(k+1)} \frac{1}{2} \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 \text{ sujeito a } e_p(k) = (1 - \beta \|\mathbf{x}(k)\|^2)e(k), \quad (\text{II.29})$$

no qual o termo  $\frac{1}{2}\|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2$  incorpora o princípio da distorção mínima, já que penaliza soluções muito distantes da solução atual.

Por meio dos multiplicadores de Lagrange, podemos reescrever o problema de otimização com restrições (II.29) como o sendo equivalente à minimização *sem restrições* da seguinte função custo:

$$\min_{\mathbf{w}(k+1)} \mathcal{F}_{\text{LMS}}[\mathbf{w}(k+1)] \triangleq \frac{1}{2} \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 + \lambda [e_p(k) - (1 - \beta \|\mathbf{x}(k)\|^2)e(k)], \quad (\text{II.30})$$

onde  $\lambda \in \mathbb{R}$  é o multiplicador de Lagrange. A minimização de (II.30) pode ser feita zerando-se seu gradiente com relação a  $\mathbf{w}(k+1)$ :

$$\nabla_{\mathbf{w}(k+1)} \mathcal{F}_{\text{LMS}}[\mathbf{w}(k+1)] = \mathbf{0} \Rightarrow \mathbf{w}(k+1) - \mathbf{w}(k) - \lambda \mathbf{x}(k) = \mathbf{0}, \quad (\text{II.31})$$

o que leva a

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \lambda \mathbf{x}(k), \quad (\text{II.32})$$

equação que depende do multiplicador de Lagrange. Impondo que a solução deve respeitar a restrição de (II.29), obtemos:

$$\lambda = \beta e(k), \quad (\text{II.33})$$

identidade que, quando conjugada a (II.32), dá origem à equação de atualização do LMS, já apresentada:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \beta \mathbf{x}(k) e(k). \quad (\text{II.34})$$

Assim, temos que o algoritmo LMS pode ser derivado de duas formas alternativas: por meio da técnica de gradiente estocástico ou por meio do princípio da distorção mínima. Existem algoritmos que somente são obtidos por meio de uma dessas duas técnicas de otimização. Já outros são mais naturalmente obtidos por uma delas. Este é o caso do algoritmo LMS normalizado (NLMS, do inglês *normalized LMS*), o qual pode ser descrito como o solucionador exato do seguinte problema de otimização:

$$\min_{\mathbf{w}(k+1)} \frac{1}{2} \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 \text{ sujeito a } e_p(k) = (1 - \beta)e(k), \quad (\text{II.35})$$

cuja resolução, utilizando os mesmos passos descritos para o caso do LMS, dá origem à equação de atualização do algoritmo NLMS:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \beta \frac{\mathbf{x}(k)e(k)}{\|\mathbf{x}(k)\|^2}, \quad (\text{II.36})$$

o qual apresenta algumas vantagens com relação ao algoritmo LMS. Umas das mais relevantes destas vantagens reside no fato de o intervalo no qual o fator de aprendizagem garante estabilidade do aprendizado é basicamente independente das propriedades estatísticas do sinal de entrada. Assim, escolhendo  $\beta$  no intervalo  $(0, 1]$ , temos garantia de estabilidade, independentemente, por exemplo, do espalhamento espectral da matriz de autocorrelação do sinal de entrada.

## II.4 Aplicações

Os algoritmos de filtragem adaptativa possuem diversas aplicações. Duas delas foram exemplificadas nas seções **II.1.1** (Equalização de Sinais) e **III.2.1** (Identificação de Sistemas), mas diversas

outras aplicações práticas podem ser citadas. Nesta seção, adicionalmente duas destas aplicações são mencionadas nesta parte do trabalho para haja uma melhor compreensão.

#### II.4.1 Aprimoramento de sinal

Na implementação de um algoritmo de filtragem adaptativa para aprimoramento de sinal (Figura II.13), em geral, se utiliza a configuração em que um sinal de entrada  $x(k)$  está adicionado por um ruído  $\nu_1(k)$  que desconhecemos; no entanto temos acesso a um sinal  $\nu_2(k)$  que sabidamente está correlacionado ao ruído  $\nu_1(k)$  do sinal original. Desta forma, podemos utilizar o sinal  $\nu_2(k)$  correlacionado ao ruído original como entrada de um filtro adaptativo, servindo assim como referência para o mesmo. Ao final, após a convergência do filtro adaptativo, o sinal de erro  $e(k)$  será o sinal de entrada  $x(k)$  aprimorado/intensificado, pois estará subtraído pelo sinal correlacionado que foi ajustado pelo filtro.

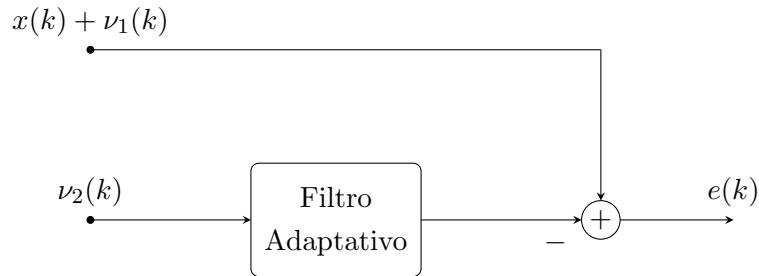


Figura II.13: Diagrama de blocos de um algoritmo de Filtragem Adaptativa atuando em aprimoramento de Sinal.

#### II.4.2 Predição de sinais

Na implementação de um algoritmo de filtragem adaptativa para predição de sinais (Figura II.14), o filtro adaptativo é utilizado para encontrar uma relação entre o sinal de entrada  $x(k)$  e a próxima amostra, de tal forma que o sinal desejado seja uma versão adiantada do original. Depois que o filtro adaptativo converge e se ajusta ao sinal de referência  $x(k)$  a saída do filtro pode ser utilizada como um preditor do sinal de entrada.

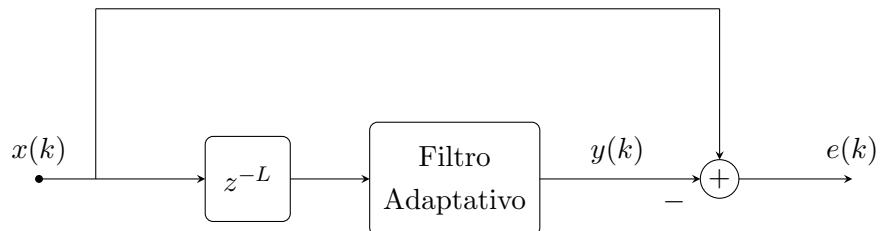


Figura II.14: Diagrama de blocos de um algoritmo de Filtragem Adaptativa atuando em Predição de Sinais.

## II.5 Comentários Finais

Este capítulo introduziu alguns conceitos e terminologia básicos de algoritmos de filtragem adaptativa. Foram ademais introduzidos os algoritmos tradicionais LMS e NLMS, bem como algumas simulações foram apresentadas para a discussão de alguns aspectos importantes do desempenho destes algoritmos. Foram apresentados alguns exemplos práticos para contextualização e melhor compreensão do trabalho. Um tema de grande relevo na literatura consiste no desenvolvimento de modelos estocásticos de algoritmos de filtragem adaptativa. Tais modelos são capazes de oferecer diversos *insights* e garantias para o projetista. No próximo capítulo, trataremos de um método de análise que será empregado na principal contribuição deste trabalho, a ser descrita no Capítulo IV.

## Capítulo III Modelos Estocásticos de Algoritmos de Filtragem Adaptativa

### III.1 Introdução

Diversas técnicas de análise, entre determinísticas e estocásticas, têm sido mobilizadas para elucidar os principais fatores responsáveis pelo desempenho de algoritmos de filtragem adaptativa. Nesta introdução, sem a pretensão de exaurir toda a literatura correlata, trataremos de algumas referências importantes neste campo. Embora algumas delas sejam bastante antigas (algumas datando da década de 1980), diversas delas mantém sua atualidade, já que podem ser reutilizadas na análise de novos algoritmos.

Uma das primeiras análises teóricas referentes a algoritmos de filtragem adaptativa referia-se ao LMS [Widrow et al., 1976]. Tendo como principal autor o inventor do LMS (o pesquisador Bernard Widrow), o artigo já caracterizava o MSE como um paraboloide em funções dos pesos adaptativos com um único ponto de mínimo, o qual poderia ser procurado por meio de técnicas de gradiente. O processo de estimação do gradiente já é então reconhecido como ruidoso, com uma energia média proporcional à taxa de aprendizagem  $\beta$  e ao número de coeficientes adaptativos. Tal referência também efetua, de maneira precoce, o tratamento estocástico do caso não-estacionário, acentuando a dependência do aprendizado do LMS com relação aos autovalores da matriz de autocorrelação  $\mathbf{R}$  dos sinais de entrada, definida por:

$$\mathbf{R} \triangleq \mathbb{E} [\mathbf{x}(k)\mathbf{x}^T(k)]. \quad (\text{III.1})$$

Análises estocásticas normalmente aplicam o operador de valor esperado para capturar o comportamento médio de determinadas grandezas estatísticas ao longo de todas as possíveis realizações do processo estocástico de aprendizagem. Normalmente, a aplicação de tal operador requer o cômputo de momentos conjuntos dos coeficientes adaptativos  $w_i(k)$  e das amostras do sinal de entrada. Os cálculos resultantes seriam bastante simplificados se tais variáveis aleatórias fossem consideradas estatisticamente independentes, o que infelizmente não é verdade, já que a estrutura transversal induz um acoplamento *determinístico* entre vetores de entrada consecutivos (por exemplo,  $\mathbf{x}(k)$  e  $\mathbf{x}(k - 1)$  compartilham  $N - 1$  amostras idênticas, ainda que deslocadas). Assim, para simplificar a análise, a maior parte da literatura de modelos estocásticos lança mão da chamada

“hipótese da independência”, embora recentemente tenha se argumentado que a expressão mais adequada seja “heurística da independência”, conforme Solo [2019], já que tal hipótese decerto é violada na prática. Tal hipótese assevera que as amostras do sinal de entrada e os coeficientes adaptativos num determinado instante são estatisticamente independentes, o que tende a negligenciar o fato de que o estimador adaptativo apresenta dependência com relação a todo o conjunto de dados já apresentado [Quirk et al., 1998]. Existe uma família de artigos que não se vale da hipótese da independência como em Douglas [1992, 1993]; Douglas and W. Pan [1995]; Lara et al. [2019a]; Lara et al. [2018]; Lara et al. [2019b,c]; Douglas and Meng [1992]; Lara et al. [2020]; Silva et al. [2020, 2021], geralmente exigindo alta complexidade computacional para lidar com as equações resultantes. Uma vertente de técnicas que contorna a adoção de tal hipótese assume que os filtros são extremamente longos em Butterweck [2001]; Rupp and Butterweck [2003]; Butterweck [1999, 1996a,b, 1995], a qual curiosamente apresenta conexões com técnicas oriundas da física estatística [Miyoshi and Kajikawa, 2013, 2015; Murata et al., 2017].

Uma família bastante popular de técnicas de análise almeja encontrar uma recursão matricial (RM) para a matriz de autocorrelação dos desvios

$$\mathbf{R}_{\tilde{\mathbf{w}}}(k) \triangleq \mathbb{E} [\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)], \quad (\text{III.2})$$

onde o desvio denota a diferença entre os valores ótimos dos coeficientes adaptativos e as suas estimativas atuais:

$$\tilde{\mathbf{w}}(k) \triangleq \mathbf{w}^* - \mathbf{w}(k). \quad (\text{III.3})$$

A determinação da evolução de  $\mathbf{R}_{\tilde{\mathbf{w}}}(k)$  é de grande interesse, porque de tal grandeza podemos inferir, mediante a adoção de algumas hipóteses comuns (como a aludida hipótese da independência), métricas de desempenho, tal como o MSE e o MSD. Tal fato atrai bastante atenção para família de técnicas de RM. Assim, como exemplares da aplicação de tal família de técnicas, temos que esta é mobilizada para analisar os algoritmos LMS e NLMS com sinais de entrada brancos, gaussianos e ciclo-estacionários em Bershad et al. [2014], para analisar o desempenho do LMS com sinais de entrada fractais e não estacionários em Gupta and Joshi [2008], para prever a rejeição de interferência em banda estreita do algoritmo LMS em Iltis and Milstein [1985] e para prever o desempenho do algoritmo LMS quando a planta a identificar é variante no tempo, mesmo quando os coeficientes são quantizados [Farhang-Boroujeny and Gazor, 1996; Bermudez and Bershad, 1996].

Uma outra técnica de análise estocástica possível para se modelar o comportamento de algoritmos de filtragem adaptativa se baseia em argumentos de conservação de energia, cuja referência seminal é Yousef and Sayed [2001], a qual tem grande dependência da análise determinística apresentada em [Sayed and Rupp, 1996]. Tal técnica foi mobilizada em Zhao and Sayed [2012] para analisar

os limites de desempenho de técnicas de estimação distribuída em redes adaptativas utilizando-se o LMS, para análise de transiente em filtros normalizados em Al-Naffouri and Sayed [2003], para analisar a habilidade de filtros adaptativos em acompanhar desvios na frequência de portadoras no caso de canais não estacionários em Yousef and Sayed [2002], para analisar as habilidades de rastreamento do algoritmo sign-LMS em Yousef and Sayed [2000], para analisar algoritmos conscientes da esparsidade em Haddad et al. [2016]; Lima et al. [2014] e para modelar o desempenho de algoritmos baseados em kernel em [Chen et al., 2017].

Cabe ressaltar que quando os algoritmos adaptativos atuam como equalizadores transversais na presença de interferência aditiva em banda estreita, alguns fenômenos inesperados podem surgir, como a possibilidade de o LMS superar o desempenho da solução de Wiener, fato que vem sendo elucidado por uma importante literatura correlata [Ikuma et al., 2008; Quirk et al., 1998]. Tal literatura apresenta conexões matemáticas com a chamada análise determinística de algoritmos, a qual explora sinais não estocásticos (como somas de senoides) Sayed and Rupp [1996]; Olmos and Laguna [2000]; Clarkson and White [1987]; Titchener [1990] e pode ser usada para demonstrar a otimalidade  $\mathcal{H}^\infty$  do algoritmo LMS [Hassibi et al., 1996].

Outras análises desvendaram fatores importantes que influenciam o desempenho dos algoritmos. Assim, por exemplo, em Ali et al. [2021], a dependência do MSE em excesso do NLMS (ou seja, acima do nível da variância do ruído) é caracterizada como uma função dos autovalores da matriz de autocorrelação de entrada, com tal função apresentando a propriedade de convexidade de Schur. Citar todas estas análises alternativas, muito numerosas, encontra-se fora do escopo deste trabalho.

Ao longo da seção seguinte, entraremos em maiores detalhes em uma técnica específica de análise proposta em Slock [1993], a qual denominaremos doravante de *modelo de Slock*. Esta técnica tem a vantagem de permitir encontrar fórmulas que associam características de desempenho a parâmetros escolhidos para o projetista, sendo capaz de fornecer *insights* úteis para a implementação na prática. Ela será utilizada no próximo capítulo para modelar o comportamento dos algoritmos adaptativos LMS e NLMS, quando implementados em blocos, o que consiste na principal contribuição deste trabalho.

### **III.2 *Modelo de Slock***

A despeito de haver sido publicado em 1993, o *modelo de Slock* vem se revelando bastante útil para a modelagem de diversos algoritmos adaptativos até os dias atuais. Como exemplos da utilização deste modelo, temos a análise de algoritmos proporcionais em Das and Chakraborty [2015], de algoritmos de atualização parcial com seleção de dados em Werner et al. [2004], de algoritmos do tipo *affine projection* em Paul and Ogunfunmi [2011] e algoritmos com reúso de dados binormalizados [Apolinario et al., 2000]. Sendo possível também aplicar em outras classes

de Algoritmos de filtragem Adaptativa, tais como o algoritmo BC-LMS(Bias Compensated) em Pimenta et al. [2021] e o algoritmo RC-LMS(Reusing Coefficient) em Pimenta et al. [2018].

O modelo é importante porque permite uma exploração analítica do comportamento qualitativo de algoritmos de filtragem adaptativa. Devido à coerência determinística entre vetores de entrada consecutivos, o modelo markoviano não é respeitado na prática. Porém, como levar em consideração esta dependência gera muitas dificuldades analíticas, os sucessivos vetores de entrada  $\{\mathbf{x}(k-2), \mathbf{x}(k-1), \mathbf{x}(k), \dots\}$  são considerados estatisticamente independentes, o que implica que o *modelo de Slock* utiliza a hipótese da independência [Silva et al., 2020]. Como normalmente os vetores de entrada apresentam média zero, sabemos que suas estatísticas de primeira e segunda ordens devem satisfazer às restrições seguintes:

$$\mathbb{E} [\mathbf{x}(k)] = \mathbf{0}, \quad (\text{III.4})$$

$$\mathbb{E} [\mathbf{x}(k)\mathbf{x}^T(k)] = \mathbf{R}, \quad (\text{III.5})$$

onde a matriz  $\mathbf{R}$  apresenta autovalores  $\lambda_i \in \mathbb{R}_+$  (para  $i \in \{0, 1, \dots, N-1\}$ ), cada um dos quais associados a seus autovetores respectivos  $\mathbf{v}_i$ , para  $i \in \{0, 1, \dots, N-1\}$ . Definindo as matrizes

$$\mathbf{V} \triangleq \begin{bmatrix} \mathbf{v}_0 & \mathbf{v}_1 & \dots & \mathbf{v}_{N-1} \end{bmatrix} \quad (\text{III.6})$$

e

$$\Sigma \triangleq \begin{bmatrix} \lambda_0 & 0 & 0 & \dots & 0 \\ 0 & \lambda_1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & \lambda_{N-1} \end{bmatrix}, \quad (\text{III.7})$$

é possível decompor espectralmente a matriz de autocorrelação  $\mathbf{R}$  por

$$\mathbf{R} = \mathbb{E} [\mathbf{x}(k)\mathbf{x}^T(k)] = \mathbf{V}\Sigma\mathbf{V}^T = \sum_{i=0}^{N-1} \lambda_i \mathbf{v}_i \mathbf{v}_i^T. \quad (\text{III.8})$$

O *modelo de Slock* assume que o vetor  $\mathbf{x}(k)$  é gerado pelo produto de duas variáveis aleatórias ( $s_k \in \{+1, -1\}$  e  $r_k \in \mathbb{R}_+$ ) e um vetor aleatório  $\mathbf{V}_k \in \mathbb{R}^N$ ; ou seja:

$$\mathbf{x}(k) = s_k r_k \mathbf{V}_k, \quad (\text{III.9})$$

onde  $s_k$  é uma variável aleatória discreta que assume os valores  $+1$  e  $-1$  de forma equiprovável (podendo ser associada ao sinal de  $\mathbf{x}(k)$ ),  $r_k$  apresenta a mesma distribuição *radial* que o vetor  $\mathbf{x}(k)$  original (ou seja, sem a adoção do modelo) e  $\mathbf{V}_k$  é um vetor aleatório discreto que pode assumir  $N$

direções (ou *ângulos*), cada uma das quais sendo caracterizada por um autovetor  $\mathbf{v}_i$ :

$$\text{Prob}[\mathcal{V}_k = \mathbf{v}_i] = p_i = \frac{\lambda_i}{\text{Tr}[\mathbf{R}]}, \quad (\text{III.10})$$

onde a positividade de  $p_i$  (necessária para que seja uma probabilidade válida) decorre do fato de os autovetores  $\lambda_i$  serem positivos, assim como o traço da matriz  $\mathbf{R}$ . Ademais, como a soma dos autovalores  $\lambda_i$  é igual ao traço da matriz  $\mathbf{R}$ , temos que a soma  $\sum_{i=0}^{N-1} p_i$  é unitária, o que atende às condições necessárias de uma variável aleatória:

$$\sum_{i=0}^{N-1} p_i = \sum_{i=0}^{N-1} \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} = \frac{1}{\text{Tr}[\mathbf{R}]} \sum_{i=0}^{N-1} \lambda_i = 1. \quad (\text{III.11})$$

Estando o vetor aleatório discreto  $\mathcal{V}_k$  associado aos possíveis ângulos (finitos) que  $\mathbf{x}(k)$  pode assumir (no modelo), percebemos que o *modelo de Slock* divorcia a distribuição radial incorporada em  $r_k$  da distribuição angular do vetor de entrada  $\mathbf{x}(k)$ . Ademais, observa-se que a distribuição angular é discretizada, podendo assumir apenas uma de  $N$  possíveis direções (fornecidas pelos autovetores da matriz  $\mathbf{R}$ ). Em termos pouco rigorosos, tais direções tentam aproximar as direções preferenciais assumidas pelo vetor  $\mathbf{x}(k)$  (que na prática, são infinitas e não discretas, ao contrário do que preconiza o modelo).

O *modelo de Slock* é claramente compatível com as estatísticas apresentadas em (III.4)-(III.5); afinal, assumindo-o como válido, encontramos

$$\begin{aligned} \mathbb{E}[\mathbf{x}(k)] &= \mathbb{E}[s_k r_k \mathcal{V}_k] = \overbrace{\mathbb{E}[s_k]}^{=0} \mathbb{E}[r_k] \mathbb{E}[\mathcal{V}_k] = \mathbf{0}, \\ \mathbb{E}[\mathbf{x}(k)\mathbf{x}^T(k)] &= \mathbb{E}[s_k^2 r_k^2 \mathcal{V}_k \mathcal{V}_k^T] = \overbrace{\mathbb{E}[s_k^2]}^{=1} \overbrace{\mathbb{E}[r_k^2]}^{=\text{Tr}[\mathbf{R}]} \mathbb{E}[\mathcal{V}_k \mathcal{V}_k^T] \\ &= \text{Tr}[\mathbf{R}] \sum_{i=0}^{N-1} p_i \mathbf{v}_i \mathbf{v}_i^T = \text{Tr}[\mathbf{R}] \sum_{i=0}^{N-1} \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \mathbf{v}_i \mathbf{v}_i^T \\ &= \sum_{i=0}^{N-1} \lambda_i \mathbf{v}_i \mathbf{v}_i^T = \mathbf{R}, \end{aligned} \quad (\text{III.12})$$

onde em (III.12) utilizamos o fato de que

$$\mathbb{E}[s_k] = \text{Prob}[s_k = 1](1) + \text{Prob}[s_k = -1](-1) = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot (-1) = 0 \quad (\text{III.14})$$

e em (III.13) utilizamos a decomposição espectral (III.8).

Ademais, o *modelo de Slock* é também compatível com a hipótese da independência, já que dele decorre que os vetores  $\mathbf{x}(k-i)$  e  $\mathbf{x}(k-j)$  são estatisticamente independentes para  $i \neq j$ ; assim,

como da independência estatística decorre a descorrelação, temos:

$$\begin{aligned}\mathbb{E} [\mathbf{x}(k-i)\mathbf{x}^T(k-j)] &= \mathbb{E} [s_{k-i}r_{k-i}\mathbf{\mathcal{V}}_{k-i}s_{k-j}r_{k-j}\mathbf{\mathcal{V}}_{k-j}^T] \\ &= \underbrace{\mathbb{E}[s_{k-i}]}_{=0} \underbrace{\mathbb{E}[s_{k-j}]}_{=0} \mathbb{E}[r_{k-i}] \mathbb{E}[r_{k-j}] \mathbb{E}[\mathbf{\mathcal{V}}_{k-i}] \mathbb{E}[\mathbf{\mathcal{V}}_{k-j}^T] = \mathbf{0},\end{aligned}\quad (\text{III.15})$$

para  $i \neq j$ .

### III.2.1 Exemplos ilustrativos

Para elucidar melhor as características e restrições decorrentes do *modelo de Slock*, assumamos que o sinal  $x(k)$  seja advindo de um processo estocástico estacionário branco, gaussiano, de variância unitária (isto é,  $\sigma_x^2 = 1$ ) e de média zero. Por simplicidade, seja o caso em que o comprimento do filtro seja  $N = 2$ . Neste contexto, a matriz de autocorrelação da entrada é a matriz identidade:

$$\begin{aligned}\mathbf{R} &= \mathbb{E}[\mathbf{x}(k)\mathbf{x}^T(k)] = \begin{bmatrix} \mathbb{E}[x^2(k)] & \mathbb{E}[x(k)x(k-1)] \\ \mathbb{E}[x(k)x(k-1)] & \mathbb{E}[x^2(k-1)] \end{bmatrix} \\ &= \begin{bmatrix} \sigma_x^2 & \mathbb{E}[x(k)]\mathbb{E}[x(k-1)] \\ \mathbb{E}[x(k)]\mathbb{E}[x(k-1)] & \sigma_x^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\end{aligned}$$

cujos autovetores são  $\mathbf{v}_0 = [1 \ 0]^T$  e  $\mathbf{v}_1 = [0 \ 1]^T$ , cada um dos quais sendo associado a um autovalor unitário (isto é,  $\lambda_0 = \lambda_1 = 1$ ). A distribuição angular dos possíveis vetores  $\mathbf{x}(k)$  é uniformemente distribuída ao longo do intervalo  $[0, \pi]$ , desde que restrinjamos os ângulos possíveis ao primeiro e segundo quadrantes. Quando geramos inúmeros vetores segundo esta distribuição, podemos estimar de modo empírico tal distribuição de probabilidade, por meio de *kernels gaussianos*. A distribuição angular estimada a partir de  $10^6$  vetores é apresentada na Figura III.1. Note que não se observa uma distribuição exatamente uniforme devido aos artefatos oriundos do fato de se tratar de uma estimativa baseada nos dados. De todo modo, observamos que tal distribuição oscila em torno do valor teórico  $1/\pi \approx 0,3183$ , o que é bastante próximo de uma distribuição uniforme.

A distribuição em probabilidade da norma  $r_k = \sqrt{x^2(k) + x^2(k-1)}$  pode ser expressa analiticamente, sendo um caso particular da distribuição de Rayleigh para a distribuição considerada para os vetores  $\mathbf{x}(k)$ :

$$f_r(r) = \begin{cases} r \exp\left[-\frac{r^2}{2}\right], & \text{para } r > 0, \\ 0, & \text{no resto} \end{cases}, \quad (\text{III.16})$$

a qual pode ser comparada com sua respectiva estimativa empírica na Figura III.2.

Para esta distribuição específica dos vetores  $\mathbf{x}(k)$ , o *modelo de Slock* assume-os como gerados

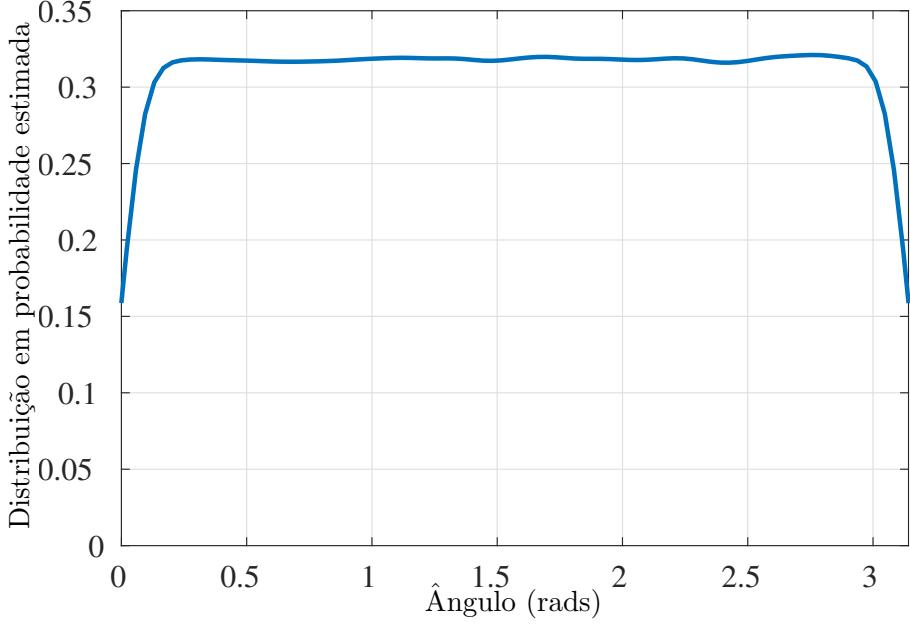


Figura III.1: Distribuição em probabilidade estimada empíricamente pelo método de *kernels gaussianos* dos ângulos dos vetores  $\mathbf{x}(k)$ , a partir de  $10^6$  vetores.

pelo produto de duas variáveis aleatórias e um vetor aleatório, todos eles independentes entre si:

$$\mathbf{x}(k) = s_k r_k \mathbf{v}_k, \quad (\text{III.17})$$

onde  $s_k$  assume de modo equiprovável valores no conjunto  $\{+1, -1\}$ ,  $r_k$  assume a distribuição (III.16) e  $\mathbf{v}_k$  assume de forma equiprovável os valores  $\mathbf{v}_0 = [1 0]^T$  e  $\mathbf{v}_1 = [0 1]^T$ . Tais valores são equiprováveis porque os autovalores são idênticos, de modo que

$$\text{Prob}[\mathbf{v}_k = \mathbf{v}_i(k)] = \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} = \frac{\lambda_i}{\sum_{i=0}^{N-1} \lambda_i} = \frac{1}{2}. \quad (\text{III.18})$$

Ademais, como apenas duas direções são possíveis, observa-se uma discretização angular advinda do *modelo de Slock*, a qual é bastante diferente da distribuição angular real dos vetores, que permite infinitos valores possíveis, conforme visto na Figura III.1. Como veremos, tal simplificação facilita ao *modelo de Slock* encontrar de modo interpretável fatores que influenciam o desempenho do algoritmo sob análise.

Como vetores sucessivos de entrada compartilham diversos elementos em comum (já que seus elementos são obtidos a partir de uma janela deslizante que extrai amostras consecutivas do sinal de entrada), sabemos que há um acoplamento determinístico entre estes vetores. Tal acoplamento dificulta a análise dos algoritmos de filtragem adaptativa, de modo que normalmente se apela à hipótese da independência para contorná-lo, conforme visto. Assim, a matriz de correlação cruzada

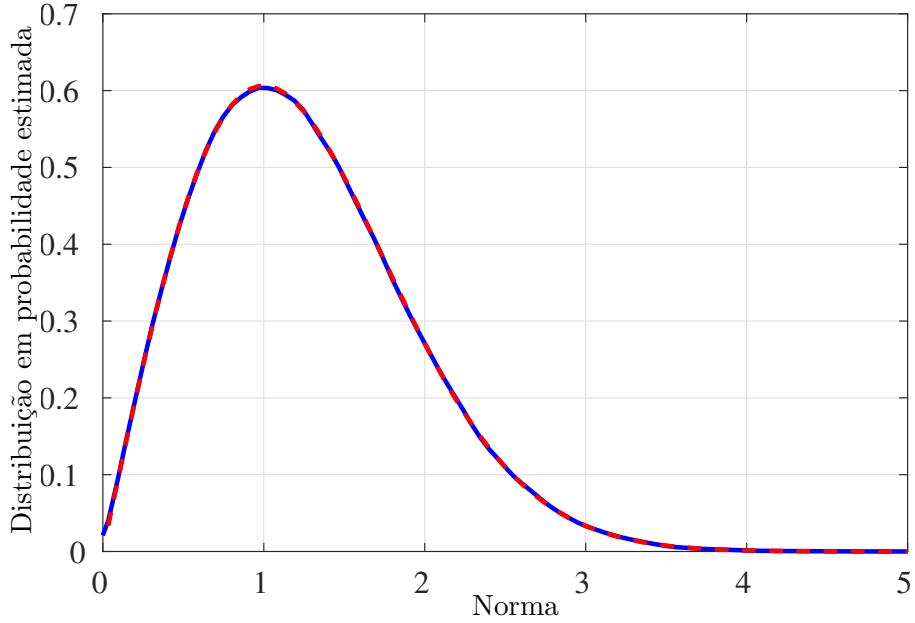


Figura III.2: Distribuição em probabilidade estimada empiricamente pelo método de *kernels gaussianos* das normas dos vetores  $\mathbf{x}(k)$ , a partir de  $10^6$  vetores (em azul). Em vermelho tracejado, a distribuição teórica de Rayleigh que descreve a solução analítica para este caso.

entre  $\mathbf{x}(k)$  e  $\mathbf{x}(k - 1)$  não é exatamente nula:

$$\begin{aligned} \mathbb{E} [\mathbf{x}(k) \mathbf{x}^T(k - 1)] &= \mathbb{E} \left\{ \begin{bmatrix} x(k) \\ x(k - 1) \end{bmatrix} \begin{bmatrix} x(k - 1) & x(k - 2) \end{bmatrix} \right\} \\ &= \begin{bmatrix} \mathbb{E}[x(k)x(k - 1)] & \mathbb{E}[x(k)x(k - 2)] \\ \mathbb{E}[x^2(k - 1)] & \mathbb{E}[x(k - 1)x(k - 2)] \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \end{aligned}$$

embora o *modelo de Slock*, que incorpora a hipótese da independência, assuma que tal matriz seja nula (vide (III.15)).

Como segundo exemplo ilustrativo, seja a configuração com  $N = 2$ , na qual o sinal de entrada  $x(k)$  seja obtido pelo seguinte modelo de média móvel de um sinal  $u(k)$ :

$$x(k) = 0.9578u(k) - 0.2873u(k - 1), \quad (\text{III.19})$$

onde  $u(k)$  é modelado como branco, gaussiano, de média zero e variância unitária. O modelo (III.19) induz uma correlação estatística entre amostras sucessivas de  $x(k)$ . Na linguagem corrente em processamento de sinais, diz-se que tal sinal é *colorido*. Tal dependência pode ser mostrada

facilmente; por exemplo:

$$\begin{aligned}\mathbb{E}[x(k)x(k-1)] &= \mathbb{E}\{[0, 9578u(k) - 0, 2873u(k-1)][0, 9578u(k-1) - 0, 2873u(k-2)]\} \\ &= 0,9174\underbrace{\mathbb{E}[u(k)u(k-1)]}_{=0} - 0,2752\underbrace{\mathbb{E}[u(k)u(k-2)]}_{=0} \\ &\quad + 0,0825\underbrace{\mathbb{E}[u(k-1)u(k-2)]}_{=0} + 0,2752\underbrace{\mathbb{E}[u^2(k-1)]}_{=1} = 0,2752 \neq 0, \quad (\text{III.20})\end{aligned}$$

o que mostra que as amostras  $x(k)$  e  $x(k-1)$  não são mais descorrelacionadas.

A estimativa da distribuição angular de  $x(k)$ , quando gerado segundo o modelo (III.19), pode ser vista na Figura III.3, onde observa-se que as direções não são mais equiprováveis. No caso do modelo gerativo (III.19) para o sinal de entrada  $x(k)$ , a matriz de autocorrelação é dada por

$$\mathbf{R} = \begin{bmatrix} 1,0347 & 0,2803 \\ 0,2803 & 1,0347 \end{bmatrix},$$

cujos autovetores são

$$\mathbf{v}_0 = \begin{bmatrix} -0,7071 \\ 0,7071 \end{bmatrix}, \mathbf{v}_1 = \begin{bmatrix} 0,7071 \\ 0,7071 \end{bmatrix},$$

cujos autovalores respectivos são  $\lambda_0 = 0,7544$  e  $\lambda_1 = 1,3150$ . Ao contrário do exemplo anterior, o *modelo de Slock* assume o vetor  $\mathbf{x}(k)$  assume cada uma das duas direções possíveis com probabilidades distintas:

$$\text{Prob}[\mathcal{V}_k = \mathbf{v}_0] = \frac{\lambda_0}{\lambda_0 + \lambda_1} = \frac{0,7544}{2,0694} = 0,3646, \quad (\text{III.21})$$

$$\text{Prob}[\mathcal{V}_k = \mathbf{v}_1] = \frac{\lambda_1}{\lambda_0 + \lambda_1} = \frac{1,3150}{2,0694} = 0,6354. \quad (\text{III.22})$$

Ademais, a distribuição radial de  $\mathbf{x}(k)$  também equivale a uma distribuição de Rayleigh, como a vista em (III.16). Independentemente da direção assumida pelo vetor  $\mathbf{x}(k)$  numa realização específica, a variável aleatória associada à distribuição radial do vetor assumirá um valor consoante à distribuição de Rayleigh de modo independente, já que a variável aleatória  $r_k$  e o vetor aleatório  $\mathcal{V}_k$  são considerados independentes no *modelo de Slock*.

### III.3 Predição do MSE

Assumindo que o sinal de referência adira ao modelo linear preconizado em (II.25), o *modelo de Slock* pode ser utilizado para prever o MSE de um algoritmo. Para esclarecer tal conexão, cumpre notar que o MSE, em uma dada iteração, pode ser reescrito como

$$\text{MSE}(k) = \xi(k) \triangleq \mathbb{E}[e^2(k)] = \mathbb{E}\left\{\left[\tilde{\mathbf{w}}^T(k)\mathbf{x}(k) + \nu(k)\right]^2\right\}, \quad (\text{III.23})$$

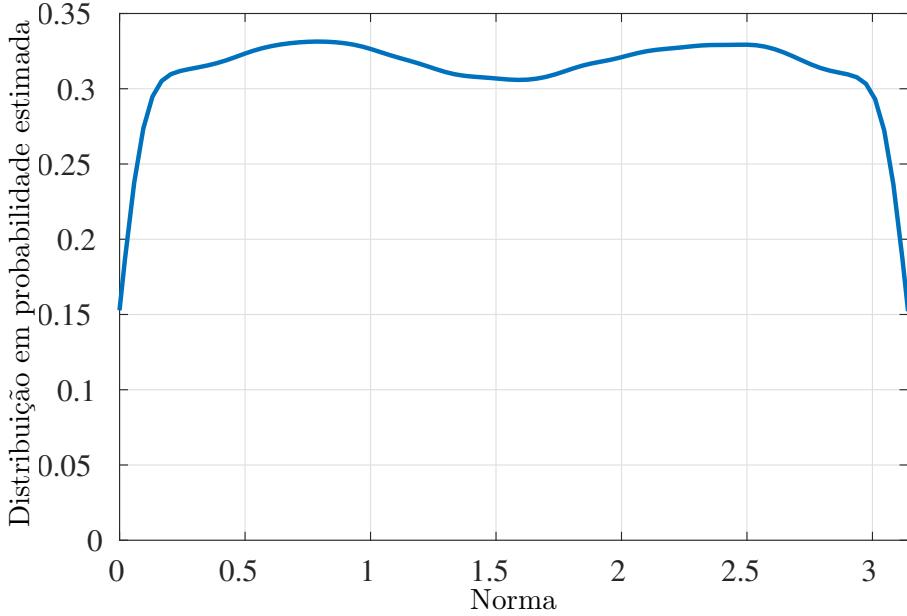


Figura III.3: Distribuição em probabilidade estimada empiricamente pelo método de *kernels gaussianos* dos ângulos dos vetores  $\mathbf{x}(k)$ , a partir de  $10^6$  vetores (em azul), obtidos de acordo com o modelo de média móvel (III.19).

onde foi feito uso da definição (III.3). A equação (III.23) pode ser simplificada por meio da adoção da hipótese do ruído, a qual pode ser descrita como:

*Hipótese do Ruído.* As amostras do processo estocástico  $\nu(k)$  são independentes e estatisticamente distribuídas (iid), apresentam média zero, variância  $\sigma_\nu^2$  constante e independência estatística com relação às demais variáveis aleatórias envolvidas.

A hipótese do ruído é ubíqua na literatura de análise de algoritmos de filtragem adaptativa, sendo Lara et al. [2019c] uma rara exceção. A despeito de ser uma hipótese estatisticamente forte, é fisicamente plausível, já que o fenômeno físico responsável pelo ruído de medição é não raro independente da fonte geradora de informação.

Utilizando-se a hipótese do ruído, podemos simplificar (III.23), obtendo

$$\xi(k) = \mathbb{E}[e^2(k)] = \mathbb{E}\left\{\left[\tilde{\mathbf{w}}^T(k)\mathbf{x}(k)\right]^2\right\} + \sigma_\nu^2, \quad (\text{III.24})$$

equação que pode ser simplificada, por apelo à hipótese da independência, para Slock [1993]:

$$\xi(k) = \sigma_\nu^2 + \text{Tr}[\mathbf{R}\mathbf{R}_{\tilde{\mathbf{w}}(k)}], \quad (\text{III.25})$$

onde  $\mathbf{R}$  é a matriz de autocorrelação do sinal de entrada (vide (III.1)) e  $\mathbf{R}_{\tilde{\mathbf{w}}(k)}$  é a matriz de autocorrelação dos desvios (vide (III.2)). Como as direções preferenciais do *modelo de Slock* dependem diretamente dos autovalores e dos autovetores da matriz de autocorrelação do sinal de entrada (vide

(III.10)), nos é de interesse explicitar tal relação por meio das seguintes manipulações:

$$\begin{aligned}
\xi(k) \triangleq \mathbb{E}[e^2(k)] &= \sigma_\nu^2 + \mathbb{E}[\|\tilde{\mathbf{w}}^T(k)\mathbf{x}(k)\|^2] \\
&= \sigma_\nu^2 + \text{Tr}[\mathbf{R}\mathbf{R}_{\tilde{\mathbf{w}}}] \\
&= \sigma_\nu^2 + \text{Tr}[\mathbf{V}\Sigma\mathbf{V}^T\mathbf{R}_{\tilde{\mathbf{w}}}] \\
&= \sigma_\nu^2 + \text{Tr}[\Sigma\mathbf{V}^T\mathbf{R}_{\tilde{\mathbf{w}}}\mathbf{V}] (\star) \\
&= \sigma_\nu^2 + \sum_{i=0}^{N-1} \lambda_i \tilde{\lambda}_i(k),
\end{aligned} \tag{III.26}$$

onde em  $(\star)$  se utilizou a identidade  $\text{Tr}[\mathbf{AB}] = \text{Tr}[\mathbf{BA}]$ , com  $\tilde{\lambda}_i(k)$  sendo definido por

$$\tilde{\lambda}_i(k) \triangleq \mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}(k)} \mathbf{v}_i. \tag{III.27}$$

A identidade (III.26) é crucial para os nossos propósitos, já que permite converter os esforços teóricos de predição do MSE na predição da evolução dos escalares  $\tilde{\lambda}_i(k)$  (para  $i \in \{0, 1, \dots, N-1\}$ ). Para ilustrar a forma como o *modelo de Slock* é utilizado, a seguir iremos analisar o comportamento dos algoritmos LMS preconizados por este modelo, seguindo a abordagem de Slock [1993].

### III.4 Análise do Algoritmo LMS pelo *Modelo de Slock*

O algoritmo LMS é definido pela equação de atualização (II.24) a qual, após manipulações matemáticas (com o suporte de (III.3) e (II.25)), nos permite encontrar uma recursão em termos dos desvios  $\tilde{\mathbf{w}}(k)$ :

$$\begin{aligned}
\mathbf{w}(k+1) &= \mathbf{w}(k) + \beta \mathbf{x}(k) [\mathbf{x}^T(k)\tilde{\mathbf{w}}(k) + \nu(k)] \\
\Rightarrow \tilde{\mathbf{w}}(k+1) &= \tilde{\mathbf{w}}(k) - \beta \mathbf{x}(k) \mathbf{x}^T(k) \tilde{\mathbf{w}}(k) - \beta \mathbf{x}(k) \nu(k) \\
\Rightarrow \tilde{\mathbf{w}}(k+1) &= [\mathbf{I} - \beta \mathbf{x}(k) \mathbf{x}^T(k)] \tilde{\mathbf{w}}(k) - \beta \mathbf{x}(k) \nu(k),
\end{aligned} \tag{III.28}$$

o que é um resultado importante, porque o MSE pode ser concisamente descrito como uma função do vetor de desvios  $\tilde{\mathbf{w}}(k)$  (vide (III.24)). Transpondo-se (III.28), obtemos

$$\tilde{\mathbf{w}}^T(k+1) = \tilde{\mathbf{w}}^T(k) [\mathbf{I} - \beta \mathbf{x}(k) \mathbf{x}^T(k)] - \beta \mathbf{x}^T(k) \nu(k). \tag{III.29}$$

Com o objetivo de se encontrar uma recursão em  $\mathbf{R}_{\tilde{\mathbf{w}}(k)}$  (também importante para a deter-

mineração do MSE, vide (III.26) e (III.27)), podemos multiplicar (III.28) por (III.29), obtendo:

$$\begin{aligned}\tilde{\mathbf{w}}(k+1)\tilde{\mathbf{w}}^T(k+1) &= [\mathbf{I} - \beta\mathbf{x}(k)\mathbf{x}^T(k)]\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)[\mathbf{I} - \beta\mathbf{x}(k)\mathbf{x}^T(k)] \\ &\quad + \beta^2\nu^2(k)\mathbf{x}(k)\mathbf{x}^T(k) + \mathcal{O}[\nu(k)],\end{aligned}\tag{III.30}$$

onde  $\mathcal{O}[\nu(k)]$  é um termo que concatena todas as expressões que dependem linearmente do ruído  $\nu(k)$ . Com a adoção da hipótese do ruído, tal termo é eliminado das análises estocásticas.

Aplicando-se o valor esperado, a hipótese referente à independência do ruído e a hipótese da independência, temos:

$$\mathbf{R}_{\tilde{\mathbf{w}}}(k+1) = \mathbb{E}\left\{[\mathbf{I} - \beta\mathbf{x}(k)\mathbf{x}^T(k)]\mathbf{R}_{\tilde{\mathbf{w}}}(k)[\mathbf{I} - \beta\mathbf{x}(k)\mathbf{x}^T(k)]\right\} + \beta^2\sigma_\nu^2\mathbf{R},\tag{III.31}$$

a qual é uma recursão para a matriz  $\mathbf{R}_{\tilde{\mathbf{w}}}(k)$  a qual ainda não se encontra no formato adequado, já que estamos essencialmente interessados em encontrar recursões para  $\tilde{\lambda}_i(k)$ , necessárias para se prever o MSE, segundo (III.26).

Ao multiplicarmos a recursão matricial (III.31) à esquerda por  $\mathbf{v}_i^T$  e à direita por  $\mathbf{v}_i$ , obtemos a seguinte recursão *escalar*:

$$\begin{aligned}\mathbf{v}_i^T\mathbf{R}_{\tilde{\mathbf{w}}}(k+1)\mathbf{v}_i &= \mathbb{E}\left\{\mathbf{v}_i^T[\mathbf{I} - \beta\mathbf{x}(k)\mathbf{x}^T(k)]\mathbf{R}_{\tilde{\mathbf{w}}}(k)[\mathbf{I} - \beta\mathbf{x}(k)\mathbf{x}^T(k)]\mathbf{v}_i\right\} + \beta^2\sigma_\nu^2\mathbf{v}_i^T\mathbf{R}\mathbf{v}_i \\ \Rightarrow \tilde{\lambda}_i(k+1) &= \mathbb{E}\left\{\mathbf{v}_i^T[\mathbf{I} - \beta\mathbf{x}(k)\mathbf{x}^T(k)]\mathbf{R}_{\tilde{\mathbf{w}}}(k)[\mathbf{I} - \beta\mathbf{x}(k)\mathbf{x}^T(k)]\mathbf{v}_i\right\} + \beta^2\sigma_\nu^2\lambda_i,\end{aligned}\tag{III.32}$$

onde fizemos uso da identidade da álgebra linear  $\mathbf{R}\mathbf{v}_i = \lambda_i\mathbf{v}_i$ . Ademais, utilizamos também a propriedade da ortonormalidade dos autovetores, a qual implica que

$$\mathbf{v}_i^T\mathbf{v}_j = \begin{cases} 1, & \text{se } i = j \\ 0, & \text{se } i \neq j. \end{cases}\tag{III.33}$$

A Equação (III.32) pode ser simplificada por meio das hipóteses subjacentes ao *modelo de Slock*:

$$\begin{aligned}\tilde{\lambda}_i(k+1) &= \sum_{j=0}^{N-1} \frac{\lambda_j}{\text{Tr}[\mathbf{R}]} \mathbb{E}\left\{\mathbf{v}_i^T[\mathbf{I} - \beta s^2 r^2 \mathbf{v}_j \mathbf{v}_j^T]\mathbf{R}_{\tilde{\mathbf{w}}}(k)[\mathbf{I} - \beta s^2 r^2 \mathbf{v}_j \mathbf{v}_j^T]\mathbf{v}_i\right\} + \beta^2\sigma_\nu^2\lambda_i \\ &= \mathbf{v}_i^T\mathbf{R}_{\tilde{\mathbf{w}}}(k)\mathbf{v}_i - \beta\mathbf{v}_i^T\mathbf{R}_{\tilde{\mathbf{w}}}(k)\mathbb{E}[\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{v}_i] - \beta\mathbb{E}[\mathbf{v}_i^T\mathbf{x}(k)\mathbf{x}^T(k)]\mathbf{R}_{\tilde{\mathbf{w}}}(k)\mathbf{v}_i \\ &\quad + \beta^2\mathbb{E}[\mathbf{v}_i^T\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{R}_{\tilde{\mathbf{w}}}(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{v}_i] + \beta^2\sigma_\nu^2\lambda_i \\ &= \tilde{\lambda}_i(k) - 2\beta\frac{\lambda_i}{\text{Tr}[\mathbf{R}]}\tilde{\lambda}_i(k) + \beta^2\sigma_\nu^2\lambda_i + \beta^2\underbrace{\mathbb{E}[\mathbf{v}_i^T\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{R}_{\tilde{\mathbf{w}}}(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{v}_i]}_{\triangleq d},\end{aligned}\tag{III.34}$$

onde o termo  $d$  pode ser computado a partir de:

$$d = \mathbb{E} [\mathbf{v}_i^T \mathbf{x}(k) \mathbf{x}^T(k) \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathbf{x}(k) \mathbf{x}^T(k) \mathbf{v}_i] = \sum_{j=0}^{N-1} \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \overbrace{\mathbb{E}[s_k^4]}^{=1} \mathbb{E}[r_k^4] \mathbf{v}_i^T \mathbf{v}_j \mathbf{v}_j^T \mathbf{R}_{\tilde{\mathbf{w}}(k)} \mathbf{v}_j \mathbf{v}_j^T \mathbf{v}_i,$$

somatório que pode ser simplificado apelando-se a (III.33), o que resulta em

$$d = \mathbb{E}[r^4] \frac{\lambda_i \tilde{\lambda}_i(k)}{\text{Tr}[\mathbf{R}]}.$$
 (III.35)

Aplicando-se (III.35) em (III.34), obtemos, após breves manipulações:

$$\tilde{\lambda}_i(k+1) = \underbrace{\left\{ 1 - 2\beta\lambda_i + \beta^2 \mathbb{E}[r^4] \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \right\}}_{\triangleq \alpha_{\text{LMS}}} \tilde{\lambda}_i(k) + \beta^2 \sigma_\nu^2 \lambda_i,$$
 (III.36)

o que é uma equação a diferenças que descreve a evolução de  $\tilde{\lambda}_i(k)$ , na qual o fator  $\alpha_{\text{LMS}}$  deve apresentar magnitude inferior à unidade para que o algoritmo seja estável (caso contrário, o MSE cresce sem limite, segundo (III.26)). O valor de  $\beta$  que maximiza a convergência pode ser encontrado via

$$\frac{\partial \alpha_{\text{LMS}}}{\partial \beta} = 0 \Rightarrow \beta_{\text{fast}} = \frac{\text{Tr}[\mathbf{R}]}{\mathbb{E}[r^4]},$$
 (III.37)

o qual é um valor único para todos os modos  $i \in \{0, 1, \dots, N-1\}$ . Infelizmente, entretanto, as estatísticas do sinal de entrada são desconhecidas, de modo que não é trivial impor um  $\beta$  que maximiza a convergência de acordo com (III.37). Ademais, o valor de  $\beta$  que maximiza a convergência nem sempre é de interesse para o projetista, já que pode resultar num desempenho em regime permanente aquém dos requerimentos da aplicação em particular.

O *modelo de Slock* permite também estabelecer teoricamente qual o maior valor para o passo de aprendizagem que evita a divergência do algoritmo:

$$|\alpha_{\text{LMS}}| < 1 \Rightarrow \beta_{\max} < 2 \frac{\text{Tr}[\mathbf{R}]}{\mathbb{E}[r^4]},$$
 (III.38)

o qual demonstra a grande dependência que a estabilidade do LMS tem com relação à estatísticas do sinal de entrada. Isso significa que o projetista, que em geral desconhece tais estatísticas, tem que ser muito cuidadoso com a possibilidade de o algoritmo divergir.

A partir da análise da equação (III.26), podemos também relacionar o comportamento do MSE em regime permanente com o tamanho da entrada  $N$ . Para tanto, devemos assumir duas hipóteses:

- Em sendo o sinal de entrada branco,  $\lambda_i$  é igual para todo  $i$  que situa-se entre 0 e  $N-1$  ao longo da execução do algoritmo. Deste modo podemos substituí-lo por um  $\bar{\lambda}$  médio.

$$= \sigma_\nu^2 + \bar{\lambda} \sum_{i=0}^{N-1} \tilde{\lambda}_i(k), \quad (\text{III.39})$$

- O algoritmo converge, logo podemos considerar o comportamento quando  $k$  (quantidade de amostras) tende ao infinito (regime permanente).

$$\lim_{k \rightarrow +\infty} \tilde{\lambda}_i(k+1) = \tilde{\lambda}_i(k), \quad (\text{III.40})$$

Utilizando a equação (III.36), substituindo  $\lambda_i$  por  $\bar{\lambda}$  e fazendo  $\tilde{\lambda}_i(k+1)$  igual a  $\tilde{\lambda}_i(k)$  teremos a seguinte expressão:

$$\tilde{\lambda}_i(k) = \left\{ 1 - 2\beta\bar{\lambda} + \beta^2 \left[ \frac{\mathbb{E}[r^4]\bar{\lambda}}{\text{Tr}[\mathbf{R}]} \right] + \beta^2\bar{\lambda}^2 \right\} \tilde{\lambda}_i(k) + \beta^2\sigma_\eta^2\bar{\lambda} \quad (\text{III.41})$$

Desenvolvendo:

$$0 = \left\{ -2\beta\bar{\lambda} + \beta^2 \left[ \frac{\mathbb{E}[r^4]\bar{\lambda}}{\text{Tr}[\mathbf{R}]} \right] \right\} \tilde{\lambda}_i(k) + \beta^2\sigma_\eta^2\bar{\lambda} \\ \tilde{\lambda}_i(k) \left\{ 2\beta\bar{\lambda} - \beta^2 \left[ \frac{\mathbb{E}[r^4]\bar{\lambda}}{\text{Tr}[\mathbf{R}]} \right] \right\} = \beta^2\sigma_\eta^2\bar{\lambda} \quad (\text{III.42})$$

Dividindo toda a equação por  $\beta^2\bar{\lambda}$ :

$$\tilde{\lambda}_i(k) \left\{ \frac{2}{\beta} - \frac{\mathbb{E}[r^4]}{\text{Tr}[\mathbf{R}]} \right\} = \sigma_\eta^2 \quad (\text{III.43})$$

Isolando  $\tilde{\lambda}_i(k)$ :

$$\tilde{\lambda}_i(k) = \frac{\sigma_\eta^2\beta\text{Tr}[\mathbf{R}]}{2\text{Tr}[\mathbf{R}] - \beta\mathbb{E}[r^4]} \quad (\text{III.44})$$

Agora que temos o  $\tilde{\lambda}_i(k)$  podemos aplicar a mesma hipótese de  $k$  tendendo ao infinito ao MSE

da equação (III.26):

$$\begin{aligned} \lim_{k \rightarrow +\infty} \xi(k) &= \sigma_\nu^2 + \bar{\lambda}N \underbrace{\tilde{\lambda}_i(k)}_{=\frac{\sigma_\eta^2 \beta \text{Tr}[\mathbf{R}]}{2\text{Tr}[\mathbf{R}] - \beta \mathbb{E}[r^4]}} \\ \lim_{k \rightarrow +\infty} \xi(k) &= \sigma_\nu^2 + \bar{\lambda}N \frac{\sigma_\eta^2 \beta \text{Tr}[\mathbf{R}]}{2\text{Tr}[\mathbf{R}] - \beta \mathbb{E}[r^4]} \end{aligned} \quad (\text{III.45})$$

Conseguimos assim, considerando a hipótese de regime permanente, relacionar o MSE em excesso (ou seja, o MSE que supera o valor mínimo  $\sigma_\nu^2$ ), o qual é linearmente proporcional ao tamanho do filtro  $N$ .

### III.5 Análise do Algoritmo NLMS pelo *Modelo de Slock*

O algoritmo NLMS, definido pela Equação (II.36), quando associado à relação (II.25) e à Equação (III.3) leva a

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \beta \frac{\mathbf{x}(k)[\mathbf{x}^T(k)\tilde{\mathbf{w}}(k) + \nu(k)]}{\|\mathbf{x}(k)\|^2}, \quad (\text{III.46})$$

relação que, após algumas manipulações, permite-nos estabelecer a seguinte recursão:

$$\tilde{\mathbf{w}}(k+1) = \left[ \mathbf{I} - \frac{\beta \mathbf{x}(k) \mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \tilde{\mathbf{w}}(k) - \frac{\beta \mathbf{x}(k) \nu(k)}{\|\mathbf{x}(k)\|^2}. \quad (\text{III.47})$$

Efetuando a transposição de (III.47), temos:

$$\tilde{\mathbf{w}}^T(k+1) = \tilde{\mathbf{w}}^T(k) \left[ \mathbf{I} - \frac{\beta \mathbf{x}(k) \mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] - \frac{\beta \mathbf{x}^T(k) \nu(k)}{\|\mathbf{x}(k)\|^2}. \quad (\text{III.48})$$

Multiplicando-se (III.47) por (III.48), temos:

$$\begin{aligned} \tilde{\mathbf{w}}(k+1) \tilde{\mathbf{w}}^T(k+1) &= \left[ \mathbf{I} - \frac{\beta \mathbf{x}(k) \mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \tilde{\mathbf{w}}(k) \tilde{\mathbf{w}}^T(k) \left[ \mathbf{I} - \frac{\beta \mathbf{x}(k) \mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \\ &\quad + \frac{\beta^2 \nu^2(k) \mathbf{x}(k) \mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^4} + \mathcal{O}[\nu(k)] \end{aligned} \quad (\text{III.49})$$

Aplicando o valor esperado e assumindo a hipótese do ruído, obtemos:

$$\mathbf{R}_{\tilde{\mathbf{w}}}(k+1) = \mathbb{E} \left\{ \left[ \mathbf{I} - \frac{\beta \mathbf{x}(k) \mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \tilde{\mathbf{w}}(k) \tilde{\mathbf{w}}^T(k) \left[ \mathbf{I} - \frac{\beta \mathbf{x}(k) \mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \right\} + \beta^2 \sigma_\nu^2 \mathbb{E} \left[ \frac{\mathbf{x}(k) \mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^4} \right]. \quad (\text{III.50})$$

Recorrendo à hipótese da independência, podemos simplificar (III.50), obtendo:

$$\mathbf{R}_{\tilde{\mathbf{w}}}(k+1) = \mathbb{E} \left\{ \left[ \mathbf{I} - \beta \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \mathbf{R}_{\tilde{\mathbf{w}}}(k) \left\{ \mathbf{I} - \beta \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right\} + \beta^2 \sigma_\nu^2 \underbrace{\mathbb{E} \left[ \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^4} \right]}_{=\mathcal{X}}, \quad (\text{III.51}) \right.$$

onde o termo  $\mathcal{X}$  pode ser equivalentemente descrito como:

$$\begin{aligned} \mathcal{X} \triangleq \mathbb{E} \left[ \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^4} \right] &= \mathbb{E} \left[ \frac{s^2 r^2 \mathbf{V} \mathbf{V}^T}{s^4 r^4 \underbrace{\mathbf{V}^T \mathbf{V}}_{=1}} \right] = \mathbb{E} \left[ \frac{\mathbf{V} \mathbf{V}^T}{s^2 r^2} \right] = \underbrace{\mathbb{E} \left[ \frac{1}{s^2} \right]}_{=1} \mathbb{E} \left[ \frac{1}{r^2} \right] \mathbb{E} \left[ \mathbf{V} \mathbf{V}^T \right] = \mathbb{E} \left[ \frac{1}{r^2} \right] \sum_{i=0}^{N-1} p_i \mathbf{v}_i \mathbf{v}_i^T \\ &= \mathbb{E} \left[ \frac{1}{r^2} \right] \sum_{i=0}^{N-1} \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \mathbf{v}_i \mathbf{v}_i^T = \mathbb{E} \left[ \frac{1}{r^2} \right] \frac{1}{\text{Tr}[\mathbf{R}]} \sum_{i=0}^{N-1} \lambda_i \mathbf{v}_i \mathbf{v}_i^T = \mathbb{E} \left[ \frac{1}{r^2} \right] \frac{\mathbf{R}}{\text{Tr}[\mathbf{R}]}, \end{aligned} \quad (\text{III.52})$$

onde o índice  $k$  foi omitido, por simplicidade.

Aplicando (III.52) em (III.51), temos:

$$\mathbf{R}_{\tilde{\mathbf{w}}}(k+1) = \mathbb{E} \left\{ \left[ \mathbf{I} - \beta \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \mathbf{R}_{\tilde{\mathbf{w}}}(k) \left\{ \mathbf{I} - \beta \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right\} + \frac{\beta^2 \sigma_\nu^2}{\text{Tr}[\mathbf{R}]} \mathbb{E} \left[ \frac{1}{r^2} \right] \mathbf{R}. \quad (\text{III.53}) \right.$$

Utilizando o mesmo procedimento da análise do LMS, multiplicamos (III.53) à esquerda por  $\mathbf{v}_i^T$

e à direita por  $\mathbf{v}_i$ , o que nos leva a:

$$\mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}}(k+1) \mathbf{v}_i = \mathbb{E} \left\{ \mathbf{v}_i^T \left[ \mathbf{I} - \beta \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \mathbf{R}_{\tilde{\mathbf{w}}}(k) \left\{ \mathbf{I} - \beta \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right\} \mathbf{v}_i \right\} + \frac{\beta^2 \sigma_\nu^2}{\text{Tr}[\mathbf{R}]} \mathbb{E} \left[ \frac{1}{r^2} \right] \mathbf{v}_i^T \mathbf{R} \mathbf{v}_i, \quad (\text{III.54})$$

identidade (III.54) pode ser reescrita como:

$$\begin{aligned} \overbrace{\mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}}(k+1) \mathbf{v}_i}^{=\tilde{\lambda}_i(k+1)} &= \overbrace{\mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathbf{v}_i}^{=\tilde{\lambda}_i(k)} - \beta \mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) \underbrace{\mathbb{E} \left[ \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right]}_{=\mathbf{a}} \mathbf{v}_i - \beta \mathbf{v}_i^T \underbrace{\mathbb{E} \left[ \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right]}_{=\mathbf{b}} \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathbf{v}_i \\ &\quad + \beta^2 \mathbf{v}_i^T \underbrace{\mathbb{E} \left[ \frac{\mathbf{x}(k)\mathbf{x}^T(k) \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathbf{x}(k) \mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^4} \right]}_{=\mathbf{c}} \mathbf{v}_i + \beta^2 \sigma_\nu^2 \mathbb{E} \left[ \frac{1}{r^2} \right] \frac{\mathbf{v}_i^T \mathbf{R} \mathbf{v}_i}{\text{Tr}[\mathbf{R}]}, \end{aligned} \quad (\text{III.55})$$

relação que depende dos termos  $\mathbf{a}$ ,  $\mathbf{b}$  e  $c$ , que calculamos a seguir:

$$\mathbf{a} \triangleq \mathbb{E} \left[ \frac{\mathbf{x}(k)\mathbf{x}^T(k) \mathbf{v}_i}{\|\mathbf{x}(k)\|^2} \right] = \mathbb{E} \left[ \frac{s^2 r^2 \mathbf{V} \mathbf{V}^T \mathbf{v}_i}{s^4 r^4} \right] = \sum_{j=0}^{N-1} p_j \mathbf{v}_j \mathbf{v}_j^T \mathbf{v}_i = p_i \mathbf{v}_i \underbrace{\|\mathbf{v}_i\|^2}_{=1} = p_i \mathbf{v}_i = \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \mathbf{v}_i, \quad (\text{III.56})$$

$$\mathbf{b} \triangleq \mathbb{E} \left[ \frac{\mathbf{v}_i^T \mathbf{x}(k) \mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] = \mathbb{E} \left[ \frac{\mathbf{v}_i^T s^2 r^2 \mathbf{V} \mathbf{V}^T}{s^4 r^4} \right] = \sum_{j=0}^{N-1} \frac{\lambda_j}{\text{Tr}[\mathbf{R}]} \mathbf{v}_i^T \mathbf{v}_j \mathbf{v}_j^T = \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \mathbf{v}_i^T \mathbf{v}_i = \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \mathbf{v}_i^T, \quad (\text{III.57})$$

$$\begin{aligned}
c \triangleq \mathbb{E} \left[ \frac{\mathbf{v}_i^T \mathbf{x}(k) \mathbf{x}^T(k) \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathbf{x}(k) \mathbf{x}(k) \mathbf{v}_i}{\|\mathbf{x}(k)\|^4} \right] &= \mathbb{E} [\mathbf{v}_i^T \mathcal{V} \mathcal{V}^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathcal{V} \mathcal{V}^T \mathbf{v}_i] \\
&= \sum_{j=0}^{N-1} \frac{\lambda_j}{\text{Tr}[\mathbf{R}]} \mathbf{v}_i^T \mathbf{v}_j \underbrace{\mathbf{v}_j^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathbf{v}_j}_{=\tilde{\lambda}_j(k)} \mathbf{v}_j^T \mathbf{v}_i \\
&= \sum_{j=0}^{N-1} \frac{\lambda_j}{\text{Tr}[\mathbf{R}]} \mathbf{v}_i^T \mathbf{v}_j \tilde{\lambda}_j(k) \mathbf{v}_j^T \mathbf{v}_i \\
&= \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \tilde{\lambda}_i(k).
\end{aligned} \tag{III.58}$$

Substituindo-se (III.56), (III.57) e (III.58) em (III.55), obtém-se:

$$\tilde{\lambda}_i(k+1) = \overbrace{\left\{ 1 - \beta(2-\beta) \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \right\}}^{\triangleq \alpha_{\text{NLMS}}} \tilde{\lambda}_i(k) + \beta^2 \sigma_\nu^2 \mathbb{E} \left[ \frac{1}{r^2} \right] \frac{\lambda_i}{\text{Tr}[\mathbf{R}]}, \tag{III.59}$$

de cuja equação a diferenças pode-se inferir que o termo  $\alpha_{\text{NLMS}} \triangleq \left( 1 - \beta(2-\beta) \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \right)$  em (III.59) é o principal responsável pelas características de convergência. Assim, podemos verificar a faixa na qual  $\beta$  pode ser escolhido de modo a evitar divergência do algoritmo:

$$1 - \beta(2-\beta) \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} < 1 \Rightarrow \beta(2-\beta) > 0.$$

Como o único intervalo em que o produto das funções  $\beta$  e  $2-\beta$  é positivo é o intervalo  $(0,2)$ , temos que  $0 \leq \beta \leq 2$  deve ser satisfeita para que o algoritmo seja estável. Este é um resultado notável, pois mostra que o algoritmo NLMS é estável independentemente das estatísticas do sinal de entrada (em geral desconhecidas *a priori*), o que desafortunadamente não é o caso do LMS (vide (III.38)).

No caso do NLMS, o valor de  $\beta$  que maximiza a convergência pode ser obtido derivando-se  $\alpha_{\text{NLMS}}$  com relação a  $\beta$  e igualando a zero:

$$\frac{\partial \alpha}{\partial \beta} = -2 \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} + 2\beta \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} = 0 \Rightarrow \beta = 1, \tag{III.60}$$

o que demonstra que a taxa de convergência do algoritmo NLMS é maximizada de modo independente das estatísticas do sinal de entrada, o que é uma vantagem para o projetista não compartilhada pelo LMS (veja a Equação (III.37)).

A partir da análise da equação (III.26) podemos também relacionar o comportamento do MSE em regime permanente com o tamanho da entrada  $N$ , para tanto devemos assumir duas hipóteses:

Em sendo o sinal de entrada branco,  $\lambda_i$  é igual para todo  $i$  que situa-se entre 0 e  $N-1$  ao

longo da execução do algoritmo. Deste modo podemos substituí-lo por um  $\bar{\lambda}$  médio.

$$= \sigma_\nu^2 + \bar{\lambda} \sum_{i=0}^{N-1} \tilde{\lambda}_i(k), \quad (\text{III.61})$$

O algoritmo converge, logo podemos considerar o comportamento quando  $k$  (quantidade de amostras) tende ao infinito (regime permanente).

$$\lim_{k \rightarrow +\infty} \tilde{\lambda}_i(k+1) = \tilde{\lambda}_i(k), \quad (\text{III.62})$$

Utilizando a equação (III.59), substituindo  $\lambda_i$  por  $\bar{\lambda}$  e fazendo  $\tilde{\lambda}_i(k+1)$  igual a  $\tilde{\lambda}_i(k)$  teremos a seguinte expressão:

$$\tilde{\lambda}_i(k) = \left\{ 1 - \beta(2 - \beta) \frac{\bar{\lambda}}{\text{Tr}[\mathbf{R}]} \right\} \tilde{\lambda}_i(k) + \beta^2 \sigma_\nu^2 \mathbb{E} \left[ \frac{1}{r^2} \right] \frac{\bar{\lambda}}{\text{Tr}[\mathbf{R}]} . \quad (\text{III.63})$$

Desenvolvendo:

$$0 = \left\{ -\beta(2 - \beta) \frac{\bar{\lambda}}{\text{Tr}[\mathbf{R}]} \right\} \tilde{\lambda}_i(k) + \beta^2 \sigma_\nu^2 \mathbb{E} \left[ \frac{1}{r^2} \right] \frac{\bar{\lambda}}{\text{Tr}[\mathbf{R}]} \\ \tilde{\lambda}_i(k) \left\{ \beta(2 - \beta) \frac{\bar{\lambda}}{\text{Tr}[\mathbf{R}]} \right\} = \beta^2 \sigma_\nu^2 \mathbb{E} \left[ \frac{1}{r^2} \right] \frac{\bar{\lambda}}{\text{Tr}[\mathbf{R}]} . \quad (\text{III.64})$$

Dividindo toda a equação por  $\frac{\beta^2 \bar{\lambda}}{\text{Tr}[\mathbf{R}]}$ :

$$\tilde{\lambda}_i(k) \left\{ \frac{(2 - \beta)}{\beta} \right\} = \sigma_\nu^2 \mathbb{E} \left[ \frac{1}{r^2} \right] . \quad (\text{III.65})$$

Isolando  $\tilde{\lambda}_i(k)$ :

$$\tilde{\lambda}_i(k) = \frac{\sigma_\nu^2 \beta \mathbb{E} \left[ \frac{1}{r^2} \right]}{(2 - \beta)} . \quad (\text{III.66})$$

Agora que temos o  $\tilde{\lambda}_i(k)$  podemos aplicar a mesma hipótese de  $k$  tendendo ao infinito ao MSE da equação (III.26):

$$\begin{aligned}
&= \frac{\sigma_\nu^2 \beta \mathbb{E} \left[ \frac{1}{r^2} \right]}{(2-\beta)} \\
\lim_{k \rightarrow +\infty} \xi(k) &= \sigma_\nu^2 + \bar{\lambda} N \overbrace{\tilde{\lambda}_i(k)}^{} ; \\
\lim_{k \rightarrow +\infty} \xi(k) &= \sigma_\nu^2 + \bar{\lambda} N \frac{\sigma_\nu^2 \beta \mathbb{E} \left[ \frac{1}{r^2} \right]}{(2-\beta)}. \tag{III.67}
\end{aligned}$$

Conseguimos assim, considerando a hipótese de regime permanente, relacionar o MSE ao tamanho do filtro  $N$ . Novamente, percebe-se uma proporcionalidade entre o MSE em excesso e o tamanho do filtro adaptativo.

### III.6 Comentários Finais

O principal objetivo deste capítulo foi a exposição dos principais conceitos relacionados ao *modelo de Slock*, utilizado para a análise estocástica de algoritmos de filtragem adaptativa. Como exemplo da aplicação desta técnica, aplicamo-la aos algoritmos tradicionais LMS e NLMS, obtendo algumas informações qualitativas acerca do valor máximo do fator de aprendizagem que tais algoritmos devem utilizar para evitar divergência, assim como o valor deste passo que maximiza a taxa de convergência. Ainda neste capítulo utilizamos uma hipótese de regime permanente para relacionar o MSE ao tamanho do filtro  $N$ . Valendo-se desta técnica, o próximo capítulo efetua uma análise dos algoritmos LMS e NLMS quando implementados em blocos, o que é a principal contribuição deste trabalho.

## Capítulo IV Análise dos Algoritmos LMS e NLMS em Blocos

### IV.1 Implementação em Blocos de Algoritmos de Filtragem Adaptativa

Uma implementação possível de técnicas de filtragem adaptativa requer o cálculo de um bloco contendo um conjunto finito de saídas, a partir de um conjunto de valores de entrada [Clark et al., 1981]. Tal modalidade é conhecida como “implementação em blocos”, sendo costumeiramente adotada para redução de custo computacional, já que se permite a implementação de transformadas rápidas, tais como a transformada rápida de Fourier [Clark et al., 1981; Wang and Wang, 1992; Panda et al., 1986]. Por vezes, tal característica de eficiência computacional motiva a adoção de algoritmos em blocos, especialmente quando implementações em tempo real são de interesse como em Dacheng and Dejung [1988], embora outras motivações possam também existir, tal como o aumento da taxa de convergência [Mikhael and Ghosh, 1995; Hun Choi et al., 2002]. Tal modalidade de algoritmo também pode apresentar vantagens quando implementada em circuitos integrados e arquiteturas com processamento paralelizável como em Clark et al. [1981]; Dacheng and Dejung [1988] ou em arranjos sistólicos [Azimi-Sadjadi and Hongye Pan, 1994]. Tal tipo de implementação em blocos também apresenta melhor estabilidade e robustez a erros de aproximação [Azimi-Sadjadi and Hongye Pan, 1994], bem como permite a adoção de arquiteturas eficientes baseadas em aritmética distribuída [Gowtham and Babu, 2014].

Em termos de equação de atualização, o *Block LMS* (BLMS) pode ser descrito como

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \beta \sum_{l=0}^{L-1} \mathbf{x}(kL+l) e(kL+l), \quad (\text{IV.1})$$

enquanto que o *Block NLMS* (BNLMS) pode ser descrito como

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \beta \sum_{l=0}^{L-1} \frac{\mathbf{x}(kL+l) e(kL+l)}{\|\mathbf{x}(kL+l)\|^2}. \quad (\text{IV.2})$$

Tais equações de atualização induzem uma menor complexidade computacional do que suas versões clássicas (vide (II.24) e (II.36)), devido à utilização de alguns “truques” clássicos da área de processamento de sinais. Como este trabalho está interessado em modelar o comportamento do algoritmo, e não em replicar a metodologia já amplamente padronizada de redução de complexidade

computacional, iremos doravante focar nas equações de atualização destes algoritmos, abstraindo as técnicas de implementação eficaz destas equações.

## IV.2 Modelo do sinal de entrada

Embora já tenhamos apresentado o modelo do sinal de entrada no Capítulo III, por razões didáticas replicamo-lo aqui, já que será extensamente utilizado para a análise dos algoritmos BLM e BNLMS nas seções seguintes:

$$\mathbf{R} \triangleq \mathbb{E} [\mathbf{x}(k)\mathbf{x}^T(k)] = \mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^T = \sum_{i=0}^{N-1} \lambda_i \mathbf{v}_i \mathbf{v}_i^T \quad (\text{IV.3})$$

$$\mathbf{x}(k) = sr\mathcal{V} \quad (\text{IV.4})$$

$$\Pr\{s = \pm 1\} = \frac{1}{2} \quad (\text{IV.5})$$

$$r \sim \|\mathbf{x}(k)\| \quad (\text{IV.6})$$

$$\Pr\{\mathcal{V} = \mathbf{v}_i\} = p_i = \frac{\lambda_i}{\text{Tr}(\mathbf{R})}, i = 0, \dots, N-1 \quad (\text{IV.7})$$

## IV.3 Análise do BLMS

Como vimos, o algoritmo BLMS pode ser descrito por

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \beta \sum_{l=0}^{L-1} \mathbf{x}(kL+l)e(kL+l), \quad (\text{IV.8})$$

onde o erro  $e(kL+l)$  é computado via

$$e(kL+l) = d(kL+l) - y(kL+l). \quad (\text{IV.9})$$

Utilizando a definição de vetor de desvio (III.3) e o modelo para o sinal de referência (II.25), podemos reescrever o sinal de erro como:

$$\begin{aligned} e(kL+l) &= [\mathbf{w}^*]^T(k)\mathbf{x}(kL+l) + \nu(kL+l) - \mathbf{w}^T(k)\mathbf{x}(kL+l) \\ &= \tilde{\mathbf{w}}^T(k)\mathbf{x}(kL+l) + \nu(kL+l). \end{aligned} \quad (\text{IV.10})$$

Aplicando-se (II.25) em (IV.8), obtemos:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \beta \sum_{l=0}^{L-1} \mathbf{x}(kL+l)\mathbf{x}^T(kL+l)\tilde{\mathbf{w}}(k) + \beta \sum_{l=0}^{L-1} \mathbf{x}(kL+l) + \nu(kL+l), \quad (\text{IV.11})$$

de onde infere-se que:

$$\tilde{\mathbf{w}}(k+1) = \tilde{\mathbf{w}}(k) + \beta \sum_{l=0}^{L-1} \mathbf{x}(kL+l) \mathbf{x}^T(kL+l) \tilde{\mathbf{w}}(k) + \beta \sum_{l=0}^{L-1} \mathbf{x}(kL+l) + \nu(kL+l) \quad (\text{IV.12})$$

Efetuando a transposição em (IV.12), temos:

$$\tilde{\mathbf{w}}^T(k+1) = \tilde{\mathbf{w}}^T(k) + \beta \tilde{\mathbf{w}}^T(k) \sum_{l=0}^{L-1} \mathbf{x}^T(kL+l) \mathbf{x}(kL+l) + \beta \sum_{l=0}^{L-1} \mathbf{x}^T(kL+l) + \nu(kL+l) \quad (\text{IV.13})$$

Definamos os seguintes termos das equações (IV.12) e (IV.13):

$$\begin{aligned} \tilde{\mathbf{w}}(k+1) &= \underbrace{\tilde{\mathbf{w}}(k)}_{(I)} + \underbrace{\beta \sum_{l=0}^{L-1} \mathbf{x}(kL+l) \mathbf{x}^T(kL+l) \tilde{\mathbf{w}}(k)}_{(II)} + \underbrace{\beta \sum_{l=0}^{L-1} \mathbf{x}(kL+l) \nu(kL+l)}_{(III)} \\ \tilde{\mathbf{w}}^T(k+1) &= \underbrace{\tilde{\mathbf{w}}^T(k)}_{(a)} + \underbrace{\beta \tilde{\mathbf{w}}^T(k) \sum_{l=0}^{L-1} \mathbf{x}^T(kL+l) \mathbf{x}(kL+l)}_{(b)} + \underbrace{\beta \sum_{l=0}^{L-1} \mathbf{x}^T(kL+l) + \nu(kL+l)}_{(c)} \end{aligned}$$

Efetuando o produto entre (IV.12) e (IV.13), temos:

$$\begin{aligned}
\tilde{\mathbf{w}}(k+1)\tilde{\mathbf{w}}^T(k+1) &= \underbrace{\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)}_{(Ia)} + \underbrace{\tilde{\mathbf{w}}(k)\beta\tilde{\mathbf{w}}^T(k)\sum_{l=0}^{L-1}\mathbf{x}(kL+l)\mathbf{x}^T(kL+l)}_{(Ib)} \\
&\quad + \underbrace{\tilde{\mathbf{w}}(k)\beta\sum_{l=0}^{L-1}\mathbf{x}^T(kL+l)+\nu(kL+l)+\beta\sum_{l=0}^{L-1}\mathbf{x}(kL+l)\mathbf{x}^T(kL+l)\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)}_{(Ic)} \\
&\quad + \underbrace{\beta\sum_{l=0}^{L-1}\mathbf{x}(kL+l)\mathbf{x}^T(kL+l)\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\beta\sum_{l=0}^{L-1}\mathbf{x}(kL+l)\mathbf{x}^T(kL+l)}_{(IIa)} \\
&\quad + \underbrace{\beta\sum_{l=0}^{L-1}\mathbf{x}(kL+l)\mathbf{x}^T(kL+l)\tilde{\mathbf{w}}(k)\beta\sum_{l=0}^{L-1}\mathbf{x}^T(kL+l)\nu(kL+l)}_{(IIb)} \\
&\quad + \underbrace{\beta\sum_{l=0}^{L-1}\mathbf{x}(kL+l)\nu(kL+l)\tilde{\mathbf{w}}^T(k)}_{(IIc)} \\
&\quad + \underbrace{\beta\sum_{l=0}^{L-1}\mathbf{x}(kL+l)\nu(kL+l)\tilde{\mathbf{w}}^T(k)\beta\tilde{\mathbf{w}}^T(k)\sum_{l=0}^{L-1}\mathbf{x}(kL+l)\mathbf{x}^T(kL+l)}_{(IIIa)} \\
&\quad + \underbrace{\beta\sum_{l=0}^{L-1}\mathbf{x}(kL+l)\nu(kL+l)\tilde{\mathbf{w}}^T(k)\beta\tilde{\mathbf{w}}^T(k)\sum_{l=0}^{L-1}\mathbf{x}(kL+l)\mathbf{x}^T(kL+l)}_{(IIIb)} \\
&\quad + \underbrace{\beta\sum_{l=0}^{L-1}\mathbf{x}(kL+l)\nu(kL+l)\beta\sum_{l=0}^{L-1}\mathbf{x}^T(kL+l)\nu(kL+l)}_{(IIIc)}, \tag{IV.14}
\end{aligned}$$

a qual é uma equação exata e válida para todas as realizações do processo de aprendizado. Para caracterizar o comportamento médio de segunda ordem do algoritmo, cabe aplicar o operador de valor esperado, em conjunto com a hipótese de independência do ruído (o qual, por hipótese, apresenta média zero), o que resulta em:

$$\begin{aligned}
\mathbb{E} [\tilde{\mathbf{w}}(k+1)\tilde{\mathbf{w}}^T(k+1)] &= \underbrace{\mathbb{E} [\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)]}_{(Ia)} \\
&\quad + \underbrace{\mathbb{E} \left[ \tilde{\mathbf{w}}(k)\beta\tilde{\mathbf{w}}^T(k) \sum_{l=0}^{L-1} \mathbf{x}(kL+l)\mathbf{x}^T(kL+l) \right]}_{(Ib)} \\
&\quad + \underbrace{\mathbb{E} \left[ \tilde{\mathbf{w}}(k)\beta \sum_{l=0}^{L-1} \mathbf{x}^T(kL+l)\nu(kL+l) \right]}_{(Ic)} \\
&\quad + \underbrace{\mathbb{E} \left[ \beta \sum_{l=0}^{L-1} \mathbf{x}(kL+l)\mathbf{x}^T(kL+l)\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k) \right]}_{(IIa)} \\
&\quad + \underbrace{\mathbb{E} \left[ \beta \sum_{l=0}^{L-1} \mathbf{x}(kL+l)\mathbf{x}^T(kL+l)\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k) \beta \sum_{l=0}^{L-1} \mathbf{x}(kL+l)\mathbf{x}^T(kL+l) \right]}_{(IIb)} \\
&\quad + \underbrace{\mathbb{E} \left[ \beta \sum_{l=0}^{L-1} \mathbf{x}(kL+l)\mathbf{x}^T(kL+l)\tilde{\mathbf{w}}(k)\beta \sum_{l=0}^{L-1} \mathbf{x}^T(kL+l)\nu(kL+l) \right]}_{(IIc)} \\
&\quad + \underbrace{\mathbb{E} \left[ \beta \sum_{l=0}^{L-1} \mathbf{x}(kL+l)\nu(kL+l)\tilde{\mathbf{w}}^T(k) \right]}_{(IIIa)} \\
&\quad + \underbrace{\mathbb{E} \left[ \beta \sum_{l=0}^{L-1} \mathbf{x}(kL+l)\nu(kL+l)\tilde{\mathbf{w}}^T(k) \beta \tilde{\mathbf{w}}^T(k) \sum_{l=0}^{L-1} \mathbf{x}(kL+l)\mathbf{x}^T(kL+l) \right]}_{(IIIb)} \\
&\quad + \underbrace{\mathbb{E} \left[ \beta \sum_{l=0}^{L-1} \mathbf{x}(kL+l)\nu(kL+l)\beta \sum_{l=0}^{L-1} \mathbf{x}^T(kL+l)\nu(kL+l) \right]}_{(IIIc)} \quad (\text{IV.15})
\end{aligned}$$

Desenvolvendo:

$$\begin{aligned}
\mathbf{R}_{\tilde{\mathbf{w}}(k+1)} &= \underbrace{\mathbf{R}_{\tilde{\mathbf{w}}(k)}}_{(Ia)} + \beta \underbrace{\mathbb{E} \left[ \sum_{l=0}^{L-1} \mathbf{x}(kL+l) \mathbf{x}^T(kL+l) \right]}_{(Ib)} \\
&\quad + \beta \underbrace{\mathbb{E} \left[ \sum_{l=0}^{L-1} \mathbf{x}(kL+l) \mathbf{x}^T(kL+l) \tilde{\mathbf{w}}(k) \tilde{\mathbf{w}}^T(k) \right]}_{(IIa)} \\
&\quad + \underbrace{\beta^2 \mathbb{E} \left[ \sum_{l_1=0}^{L-1} \sum_{l_2=0}^{L-1} \mathbf{x}(kL+l_1) \mathbf{x}^T(kL+l_1) \tilde{\mathbf{w}}(k) \tilde{\mathbf{w}}^T(k) \mathbf{x}(kL+l_2) \mathbf{x}^T(kL+l_2) \right]}_{(IIb)} \\
&\quad + \underbrace{\beta^2 \mathbb{E} \left[ \sum_{l=0}^{L-1} \sum_{l_2=0}^{L-1} \mathbf{x}(kL+l) \nu(kL+l) \mathbf{x}^T(kL+l) \nu(kL+l) \right]}_{(IIIc)} \\
&\quad + \underbrace{\mathbb{E}[0]}_{(IIc)} + \beta \underbrace{\mathbb{E}[0]}_{(IIIa)} + \beta^2 \underbrace{\mathbb{E}[0]}_{(IIIb)} + \underbrace{\mathbb{E}[0]}_{(Ic)} \tag{IV.16}
\end{aligned}$$

Multiplicando-se (IV.16) por  $\mathbf{v}_i^T$  à esquerda e por  $\mathbf{v}_i$  à direita, podemos encontrar as recursões para os escalares  $\tilde{\lambda}_i(k)$ , essenciais para a predição das métricas de desempenho de segunda ordem (veja, por exemplo, (III.26)):

$$\begin{aligned}
\underbrace{\mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}(k+1)} \mathbf{v}_i}_{\tilde{\lambda}_i(k+1)} &= \underbrace{\mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}(k)} \mathbf{v}_i}_{\tilde{\lambda}_i(k)} - \underbrace{\beta \mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}(k)} \sum_{l=0}^{L-1} \mathbb{E} [\mathbf{x}(kL+l) \mathbf{x}^T(kL+l) \mathbf{v}_i]}_{Ib} \\
&\quad - \underbrace{\beta \sum_{l=0}^{L-1} \mathbb{E} [\mathbf{v}_i^T \mathbf{x}(kL+l) \mathbf{x}^T(kL+l)] \mathbf{R}_{\tilde{\mathbf{w}}(k)} \mathbf{v}_i}_{IIa} \\
&\quad + \underbrace{\beta^2 \sum_{l_1=0}^{L-1} \sum_{l_2=0}^{L-1} \mathbb{E} [\mathbf{v}_i^T \mathbf{x}(kL+l_1) \mathbf{x}^T(kL+l_1) \mathbf{R}_{\tilde{\mathbf{w}}(k)} \mathbf{x}(kL+l_2) \mathbf{x}^T(kL+l_2) \mathbf{v}_i]}_{IIb} \\
&\quad + \underbrace{\beta^2 \sum_{l=0}^{L-1} \mathbb{E} [\mathbf{v}_i^T \mathbf{x}(kL+l) \mathbf{x}^T(kL+l) \nu^2(kL+l) \mathbf{v}_i]}_{(IIIc)} \tag{IV.17}
\end{aligned}$$

Sendo:

$$\begin{aligned}
\mathbb{E} [\mathbf{x}(kL + l) \mathbf{x}^T(kL + l) \mathbf{v}_i] &= \mathbb{E} [s^2 r^2 \mathbf{V} \mathbf{V}^T \mathbf{v}_i] \\
&= \text{Tr}[\mathbf{R}] \mathbb{E} [\mathbf{V} \mathbf{V}^T \mathbf{v}_i] \\
&= \text{Tr}[\mathbf{R}] \sum_{j=0}^{N-1} p_j \mathbf{v}_j \mathbf{v}_j^T \mathbf{v}_i \\
&= \text{Tr}[\mathbf{R}] p_i \mathbf{v}_i \\
&= \text{Tr}[\mathbf{R}] \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \mathbf{v}_i \\
&= \lambda_i \mathbf{v}_i \\
&= \mathbb{E} [\mathbf{x}(kL + l) \mathbf{x}^T(kL + l) \mathbf{v}_i] \\
&= \lambda_i \mathbf{v}_i
\end{aligned} \tag{IV.18}$$

Por analogia:

$$\mathbb{E} [\mathbf{v}_i^T \mathbf{x}(kL + l) \mathbf{x}^T(kL + l)] = \lambda_i \mathbf{v}_i^T \tag{IV.19}$$

Aplicando (IV.18) e (IV.19) em IIb e IIa temos:

$$Ib = -\beta \mathbf{v}_i^T \mathbf{R}_{\tilde{w}(k+1)} L \lambda_i \mathbf{v}_i = -\beta \lambda_i L \tilde{\lambda}(k) \tag{IV.20}$$

$$IIa = -\beta L \lambda_i \mathbf{v}_i^T \mathbf{R}_{\tilde{w}(k+1)} \mathbf{v}_i = -\beta \lambda_i L \tilde{\lambda}(k) \tag{IV.21}$$

$$Ib + IIa = -2\beta \lambda_i L \tilde{\lambda}(k) \tag{IV.22}$$

Desenvolvendo IIIc:

$$\beta^2 \sum_{l=0}^{L-1} \mathbb{E} [\mathbf{v}_i^T \mathbf{x}(kL + l) \mathbf{x}^T(kL + l) \nu^2(kL + l) \mathbf{v}_i] = \beta^2 \sigma^2 \sum_{l=0}^{L-1} \lambda_i = \beta^2 \sigma^2 L \lambda_i \tag{IV.23}$$

Desenvolvendo IIb:

$$\begin{aligned}
&= \beta^2 \sum_{l_1=0}^{L-1} \sum_{l_2=0}^{L-1} \mathbb{E} [\mathbf{v}_i^T \mathbf{x}(kL + l_1) \mathbf{x}^T(kL + l_1) \mathbf{R}_{\tilde{\mathbf{w}}(k)} \mathbf{x}(kL + l_2) \mathbf{x}^T(kL + l_2) \mathbf{v}_i] \\
&= \beta^2 \sum_{l_1=0}^{L-1} \sum_{l_2=0}^{L-1} \mathbb{E} [s_{l_1}^2 s_{l_2}^2] \mathbb{E} [r_{l_1}^2 r_{l_2}^2] \mathbb{E} [\mathbf{v}_i^T \mathbf{V}_{l_1} \mathbf{V}_{l_1}^T \mathbf{R}_{\tilde{\mathbf{w}}(k)} \mathbf{V}_{l_2} \mathbf{V}_{l_2}^T \mathbf{v}_i] \\
&= \beta^2 \sum_{l_1=0}^{L-1} \sum_{l_2=0}^{L-1} \underbrace{\mathbb{E} [s_{l_1}^2 s_{l_2}^2]}_{=1} \mathbb{E} [r_l^4] \mathbb{E} [\mathbf{v}_i^T \mathbf{v}_j \mathbf{v}_j^T \mathbf{R}_{\tilde{\mathbf{w}}(k)} \mathbf{v}_{v_j} \mathbf{v}_j^T \mathbf{v}_i] \\
&= \beta^2 \sum_{l=0}^{L-1} \mathbb{E} [r_l^4] \sum_{j=0}^{N-1} \frac{\lambda_j}{\text{Tr}[\mathbf{R}]} \mathbf{v}_i^T \mathbf{v}_j \mathbf{v}_j^T \mathbf{R}_{\tilde{\mathbf{w}}(k)} \mathbf{v}_{v_j} \mathbf{v}_j^T \mathbf{v}_i \\
&\quad + \beta^2 \sum_{l=0}^{L-1} \mathbb{E} [r_l^4] \sum_{j_1=0}^{N-1} \sum_{j_2=0}^{N-1} \frac{\lambda_j 2\lambda_j 2}{\text{Tr}[\mathbf{R}]^2} \mathbf{v}_i^T \mathbf{v}_j \mathbf{v}_j^T \mathbf{R}_{\tilde{\mathbf{w}}(k)} \mathbf{v}_{v_j} \mathbf{v}_j^T \mathbf{v}_i \\
&= \beta^2 \sum_{l=0}^{L-1} \mathbb{E} [r_l^4] \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \tilde{\lambda}_i(k) + \beta^2 \text{Tr}[\mathbf{R}]^2 L(L-1) \frac{\lambda_i^2}{\text{Tr}[\mathbf{R}]^2} \lambda(\tilde{k}) \\
&= \beta^2 L \mathbb{E} [r_l^4] \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \tilde{\lambda}_i(k) + \beta^2 L(L-1) \lambda_i^2 \lambda(\tilde{k}) \\
&= \beta^2 L \mathbb{E} [r_l^4] \frac{\lambda_i}{\text{Tr}[\mathbf{R}]^2} \tilde{\lambda}_i(k) + \beta^2 L(L-1) \lambda_i^2 \lambda(\tilde{k})
\end{aligned} \tag{IV.24}$$

Juntando as equações obtemos:

$$\begin{aligned}
\tilde{\lambda}_i(k+1) &= \tilde{\lambda}_i(k) - 2\beta \lambda_i L \tilde{\lambda}(k) + \beta^2 L \mathbb{E} [r_l^4] \frac{\lambda_i}{\text{Tr}[\mathbf{R}]^2} \tilde{\lambda}_i(k) + \beta^2 L(L-1) \lambda_i^2 \lambda(\tilde{k}) \\
&\quad + \beta^2 \sigma^2 \sum_{l=0}^{L-1} \lambda_i + \beta^2 \sigma^2 L \lambda_i,
\end{aligned} \tag{IV.25}$$

o que nos permite chegar à seguinte expressão mais concisa:

$$\tilde{\lambda}_i(k+1) = \overbrace{\left\{ 1 - 2\beta L \lambda_i + \beta^2 L \left[ \frac{\mathbb{E} [r^4] \lambda_i}{\text{Tr}[\mathbf{R}]} \right] + \beta^2 \lambda_i^2 L(L-1) \right\}}^{=\alpha_{BLMS}} \tilde{\lambda}_i(k) + \beta^2 L \sigma_\nu^2 \lambda_i, \tag{IV.26}$$

a qual claramente degenera-se em (III.36) quando  $L = 1$ . Assim, obtemos uma equação capaz de descrever grandezas relacionadas ao desempenho quadrático médio do algoritmo BLMS em blocos. Observe que o modelo prevê estabilidade do algoritmo BLMS quando  $|\alpha_{BNLMS}| < 1$ . Esta é a primeira vez que o *modelo de Slock* é aplicado a esta modalidade de algoritmos adaptativos (em bloco).

A partir da análise da equação (III.26) podemos também relacionar o comportamento do MSE em regime permanente com o tamanho da entrada  $N$ , para tanto devemos assumir duas hipóteses:

- Em sendo o sinal de entrada branco,  $\lambda_i$  é igual para todo  $i$  que situa-se entre 0 e  $N - 1$  ao

longo da execução do algoritmo. Deste modo podemos substituí-lo por um  $\bar{\lambda}$  médio.

$$= \sigma_\nu^2 + \bar{\lambda} \sum_{i=0}^{N-1} \tilde{\lambda}_i(k). \quad (\text{IV.27})$$

O algoritmo converge, logo podemos considerar o comportamento quando  $k$  (quantidade de amostras) tende ao infinito (regime permanente).

$$\lim_{k \rightarrow +\infty} \tilde{\lambda}_i(k+1) = \tilde{\lambda}_i(k), \quad (\text{IV.28})$$

Utilizando a equação (IV.26), substituindo  $\lambda_i$  por  $\bar{\lambda}$  e fazendo  $\tilde{\lambda}_i(k+1)$  igual a  $\tilde{\lambda}_i(k)$  teremos a seguinte expressão:

$$\tilde{\lambda}_i(k) = \left\{ 1 - 2\beta L \bar{\lambda} + \beta^2 L \left[ \frac{\mathbb{E}[r^4] \bar{\lambda}}{\text{Tr}[\mathbf{R}]} \right] + \beta^2 \bar{\lambda}^2 L(L-1) \right\} \tilde{\lambda}_i(k) + \beta^2 L \sigma_\nu^2 \bar{\lambda} \quad (\text{IV.29})$$

Desenvolvendo:

$$0 = \left\{ -2\beta L \bar{\lambda} + \beta^2 L \left[ \frac{\mathbb{E}[r^4] \bar{\lambda}}{\text{Tr}[\mathbf{R}]} \right] + \beta^2 \bar{\lambda}^2 L(L-1) \right\} \tilde{\lambda}_i(k) + \beta^2 L \sigma_\nu^2 \bar{\lambda}$$

$$\tilde{\lambda}_i(k) \left\{ 2\beta L \bar{\lambda} - \beta^2 L \left[ \frac{\mathbb{E}[r^4] \bar{\lambda}}{\text{Tr}[\mathbf{R}]} \right] - \beta^2 \bar{\lambda}^2 L(L-1) \right\} = \beta^2 L \sigma_\nu^2 \bar{\lambda} \quad (\text{IV.30})$$

Dividindo toda a equação por  $\beta^2 L \bar{\lambda}$ :

$$\tilde{\lambda}_i(k) \left\{ \frac{2}{\beta} - \frac{\mathbb{E}[r^4]}{\text{Tr}[\mathbf{R}]} - (L-1) \right\} = \sigma_\nu^2$$

$$(IV.31)$$

Isolando  $\tilde{\lambda}_i(k)$ :

$$\tilde{\lambda}_i(k) = \frac{\sigma_\nu^2 \beta \text{Tr}[\mathbf{R}]}{\text{Tr}[\mathbf{R}] (2 - \beta(L-1)) - \beta \mathbb{E}[r^4]} \quad (\text{IV.32})$$

Agora que temos o  $\tilde{\lambda}_i(k)$  podemos aplicar a mesma hipótese de  $k$  tendendo ao infinito ao MSE da equação (III.26):

$$\begin{aligned} \lim_{k \rightarrow +\infty} \xi(k) &= \sigma_\nu^2 + \bar{\lambda}N \quad \tilde{\lambda}_i(k) \\ \lim_{k \rightarrow +\infty} \xi(k) &= \sigma_\nu^2 + \bar{\lambda}N \frac{\sigma_\nu^2 \beta \text{Tr}[\mathbf{R}]}{\text{Tr}[\mathbf{R}] (2 - \beta(L-1)) - \beta \mathbb{E}[r^4]} \end{aligned} \quad (\text{IV.33})$$

Conseguimos assim, considerando a hipótese de regime permanente, relacionar o MSE ao tamanho do filtro  $N$  e a quantidade de blocos  $L$ . Novamente, observa-se, pelo modelo adotado, que o MSE em excesso pode ser, no caso de sinais de entrada brancos, linearmente relacionado ao tamanho do filtro (ou seja, ao número de parâmetros a estimar). Cumpre lembrar que para quantidade de blocos  $L = 1$  a equação (IV.33) degenera-se naturalmente em (III.45), como é esperado.

#### IV.4 Análise do BNLMS

No caso do algoritmo NLMS em blocos, a equação de atualização pode ser descrita por:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \beta \sum_{l=0}^{L-1} \frac{\mathbf{x}(kL+l)e(kL+l)}{\|\mathbf{x}(kL+l)\|^2} \quad (\text{IV.34})$$

Erro:

$$e(kL+l) = d(kL+l) - y(kL+l) \quad (\text{IV.35})$$

Sendo:

$$\tilde{\mathbf{w}}(k) = \mathbf{w}^*(k) - \mathbf{w}(k) \quad (\text{IV.36})$$

Reescrevendo o erro:

$$\begin{aligned} e(kL+l) &= [\mathbf{w}^*]^T(k) \mathbf{x}(kL+l) + \nu(kL+l) - \mathbf{w}^T(k) \mathbf{x}(kL+l) \\ &= \mathbf{w}^T(k) \mathbf{x}(kL+l) + \nu(kL+l) \end{aligned} \quad (\text{IV.37})$$

Aplicando a equação (IV.37) em (IV.34):

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \beta \sum_{l=0}^{L-1} \frac{\mathbf{x}(kL+l)(k)\mathbf{x}^T(kL+l)\tilde{\mathbf{w}}(k)}{\|\mathbf{x}(kL+l)\|^2} + \beta \sum_{l=0}^{L-1} \frac{\mathbf{x}(kL+l)\nu(kL+l)}{\|\mathbf{x}(kL+l)\|^2} \quad (\text{IV.38})$$

Em decorrência da equação (IV.36) temos a equação abaixo, a qual nomearemos seus termos de I a III:

$$\tilde{\mathbf{w}}(k+1) = \overbrace{\tilde{\mathbf{w}}(k)}^I - \overbrace{\beta \sum_{l=0}^{L-1} \frac{\mathbf{x}(kL+l)(k)\mathbf{x}^T(kL+l)\tilde{\mathbf{w}}(k)}{\|\mathbf{x}(kL+l)\|^2}}^{II} - \overbrace{\beta \sum_{l=0}^{L-1} \frac{\mathbf{x}(kL+l)\nu(kL+l)}{\|\mathbf{x}(kL+l)\|^2}}^{III} \quad (\text{IV.39})$$

Efetuando a transposição temos a equação a seguir, a qual nomearemos seus termos de a até c:

$$\tilde{\mathbf{w}}^T(k+1) = \overbrace{\tilde{\mathbf{w}}^T(k)}^a - \overbrace{\beta \tilde{\mathbf{w}}^T(k) \sum_{l=0}^{L-1} \frac{\mathbf{x}(kL+l)(k)\mathbf{x}^T(kL+l)}{\|\mathbf{x}(kL+l)\|^2}}^b - \overbrace{\beta \sum_{l=0}^{L-1} \frac{\mathbf{x}^T(kL+l)\nu(kL+l)}{\|\mathbf{x}(kL+l)\|^2}}^c \quad (\text{IV.40})$$

Realizando o produto termo a termo de (IV.39) com (IV.40) teremos que:

$$\begin{aligned} \tilde{\mathbf{w}}(k+1)\tilde{\mathbf{w}}^T(k+1) &= \overbrace{\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)}^{Ia} - \overbrace{\beta \tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k) \sum_{l=0}^{L-1} \frac{\mathbf{x}(kL+l)(k)\mathbf{x}^T(kL+l)}{\|\mathbf{x}(kL+l)\|^2}}^{Ib} \\ &\quad - \overbrace{\beta \sum_{l=0}^{L-1} \frac{\mathbf{x}(kL+l)(k)\mathbf{x}^T(kL+l)\tilde{\mathbf{w}}(k)}{\|\mathbf{x}(kL+l)\|^2} \tilde{\mathbf{w}}^T(k)}^{IIa} \\ &\quad + \overbrace{\beta^2 \sum_{l_1=0}^{L-1} \frac{\mathbf{x}(kL+l_1)(k)\mathbf{x}^T(kL+l_1)}{\|\mathbf{x}(kL+l_1)\|^2} \mathbf{w}(k)\tilde{\mathbf{w}}^T(k) \sum_{l_2=0}^{L-1} \frac{\mathbf{x}(kL+l_2)(k)\mathbf{x}^T(kL+l_2)}{\|\mathbf{x}(kL+l_2)\|^2}}^{IIb} \\ &\quad + \overbrace{\beta^2 \sum_{l=0}^{L-1} \sum_{l_1=0}^{L-1} \frac{\mathbf{x}(kL+l_1)\nu(kL+l_1)}{\|\mathbf{x}(kL+l_1)\|^2} \frac{\mathbf{x}^T(kL+l_2)\nu(kL+l_2)}{\|\mathbf{x}(kL+l_2)\|^2}}^{IIIc} \\ &\quad + \overbrace{\mathcal{O}[\nu(kL, \dots, kL+L-1)]}^{Ic+IIc+IIIa+IIIb} \end{aligned} \quad (\text{IV.41})$$

Aplicando o operador de valor esperado  $\mathbb{E}[\cdot]$  em (IV.41) e supondo que o ruído seja estatística-

mente independente e tenha media zero:

$$\begin{aligned}
 \mathbf{R}_{\tilde{\mathbf{w}}}(k+1) &= \overbrace{\mathbf{R}_{\tilde{\mathbf{w}}}(k)}^{Ia} - \overbrace{\beta \sum_{l=0}^{L-1} \mathbb{E} \left[ \frac{\mathbf{x}(kL+l)(k)\mathbf{x}^T(kL+l)}{\|\mathbf{x}(kL+l)\|^2} \right]}^{Ib} \\
 &\quad - \overbrace{\beta \sum_{l=0}^{L-1} \mathbb{E} \left[ \frac{\mathbf{x}(kL+l)(k)\mathbf{x}^T(kL+l)}{\|\mathbf{x}(kL+l)\|^2} \right] \mathbf{R}_{\tilde{\mathbf{w}}}(k)}^{IIa} \\
 &\quad + \overbrace{\beta^2 \sum_{l_1=0}^{L-1} \sum_{l_2=0}^{L-1} \mathbb{E} \left[ \frac{\mathbf{x}(kL+l_1)(k)\mathbf{x}^T(kL+l_1)}{\|\mathbf{x}(kL+l_1)\|^2} \mathbf{R}_{\tilde{\mathbf{w}}}(k) \frac{\mathbf{x}(kL+l_2)(k)\mathbf{x}^T(kL+l_2)}{\|\mathbf{x}(kL+l_2)\|^2} \right]}^{IIb} \\
 &\quad + \overbrace{\beta^2 \sigma^2 \sum_{l=0}^{L-1} \mathbb{E} \left[ \frac{\mathbf{x}(kL+l)\mathbf{x}^T(kL+l)}{\|\mathbf{x}(kL+l)\|^4} \right]}^{IIIc} \tag{IV.42}
 \end{aligned}$$

Multiplicando (IV.43) à esquerda por  $\mathbf{v}_i^T$  e à direita por  $\mathbf{v}_i$ , temos:

$$\begin{aligned}
 \underbrace{\mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}}(k+1) \mathbf{v}_i}_{\tilde{\lambda}_i(k+1)} &= \underbrace{\mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathbf{v}_i}_{Ia} - \overbrace{\beta \mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) \sum_{l=0}^{L-1} \mathbb{E} \left[ \frac{\mathbf{x}(kL+l)(k)\mathbf{x}^T(kL+l)}{\|\mathbf{x}(kL+l)\|^2} \mathbf{v}_i \right]}^{Ib} \\
 &\quad - \overbrace{\beta \sum_{l=0}^{L-1} \mathbb{E} \left[ \mathbf{v}_i^T \frac{\mathbf{x}(kL+l)(k)\mathbf{x}^T(kL+l)}{\|\mathbf{x}(kL+l)\|^2} \right] \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathbf{v}_i}^{IIa} \\
 &\quad + \overbrace{\beta^2 \sum_{l_1=0}^{L-1} \sum_{l_2=0}^{L-1} \mathbb{E} \left[ \mathbf{v}_i^T \frac{\mathbf{x}(kL+l_1)(k)\mathbf{x}^T(kL+l_1)}{\|\mathbf{x}(kL+l_1)\|^2} \mathbf{R}_{\tilde{\mathbf{w}}}(k) \frac{\mathbf{x}(kL+l_2)(k)\mathbf{x}^T(kL+l_2)}{\|\mathbf{x}(kL+l_2)\|^2} \mathbf{v}_i \right]}^{IIb} \\
 &\quad + \overbrace{\beta^2 \sigma^2 \sum_{l=0}^{L-1} \mathbb{E} \left[ \mathbf{v}_i^T \frac{\mathbf{x}(kL+l)\mathbf{x}^T(kL+l)}{\|\mathbf{x}(kL+l)\|^4} \mathbf{v}_i \right]}^{IIIc} \tag{IV.43}
 \end{aligned}$$

O objetivo é representar de forma teórica o  $\tilde{\lambda}_i(k+1)$ , desta forma vamos desenvolver os termos Ib, IIa, IIb e IIIc:

$$\begin{aligned}
Ib &= -\beta \mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) \sum_{l=0}^{L-1} \mathbb{E} \left[ \frac{\mathbf{x}(kL+l)(k) \mathbf{x}^T(kL+l)}{\|\mathbf{x}(kL+l)\|^2} \mathbf{v}_i \right] \\
&= -\beta \mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) \sum_{l=0}^{L-1} \mathbb{E} \left[ \frac{\mathbf{s}r\mathcal{V}s r\mathcal{V}^T}{s^2 r^2} \mathbf{v}_i \right] \\
&= -\beta \mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) \sum_{l=0}^{L-1} \mathbb{E} [V\mathcal{V}^T \mathbf{v}_i] \\
&= -\beta \mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) \sum_{l=0}^{L-1} \mathbb{E} [V\mathcal{V}^T \mathbf{v}_i] \\
&= -\beta \mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) \sum_{l=0}^{L-1} \sum_{j=0}^{N-1} \frac{\lambda_j}{\text{Tr}[\mathbf{R}]} \mathbf{v}_j \mathbf{v}^T \mathbf{v}_i \\
&= -\beta \mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) \sum_{l=0}^{L-1} \frac{\lambda_j}{\text{Tr}[\mathbf{R}]} \mathbf{v}_i \\
&= -\beta \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} L \mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathbf{v}_i \\
&= -\beta L \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \tilde{\lambda}_i(k)
\end{aligned} \tag{IV.44}$$

$$\begin{aligned}
IIa &= -\beta \sum_{l=0}^{L-1} \mathbb{E} \left[ \mathbf{v}_i^T \frac{\mathbf{x}(kL+l)(k) \mathbf{x}^T(kL+l)}{\|\mathbf{x}(kL+l)\|^2} \right] \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathbf{v}_i \\
&= -\beta \sum_{l=0}^{L-1} \mathbb{E} \left[ \mathbf{v}_i^T \frac{\mathbf{s}r\mathcal{V}s r\mathcal{V}^T}{s^2 r^2} \right] \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathbf{v}_i \\
&= -\beta \sum_{l=0}^{L-1} \sum_{j=0}^{L-1} \mathbb{E} \left[ \mathbf{v}_i^T \frac{\lambda_j}{\text{Tr}[\mathbf{R}]} \mathbf{v}_j \mathbf{v}^T \right] \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathbf{v}_i \\
&= -\beta \frac{\lambda_j}{\text{Tr}[\mathbf{R}]} L \mathbf{v}_i^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathbf{v}_i \\
&= -\beta L \frac{\lambda_j}{\text{Tr}[\mathbf{R}]} \tilde{\lambda}_i(k)
\end{aligned} \tag{IV.45}$$

$$\begin{aligned}
IIIc &= +\beta^2 \sigma^2 \sum_{l=0}^{L-1} \mathbb{E} \left[ \mathbf{v}_i^T \frac{\mathbf{x}(kL+l) \mathbf{x}^T(kL+l)}{\|\mathbf{x}(kL+l)\|^4} \mathbf{v}_i \right] \\
&= +\beta^2 \sigma^2 \sum_{l=0}^{L-1} \mathbb{E} \left[ \mathbf{v}_i^T \frac{\mathbf{s}r \mathcal{V}_{sr} \mathcal{V}^T}{s^4 r^4} \mathbf{v}_i \right] \\
&= +\beta^2 \sigma^2 L \mathbb{E} \left[ \frac{\mathbf{1}}{r^2} \right] \mathbb{E} [\mathbf{v}_i^T V \mathcal{V}^T \mathbf{v}_i] \\
&= +\beta^2 \sigma^2 L \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \mathbb{E} \left[ \frac{\mathbf{1}}{r^2} \right]
\end{aligned} \tag{IV.46}$$

$$\begin{aligned}
IIb &= +\beta^2 \sum_{l_1=0}^{L-1} \sum_{l_2=0}^{L-1} \mathbb{E} \left[ \mathbf{v}_i^T \frac{\mathbf{x}(kL+l_1)(k) \mathbf{x}^T(kL+l_1)}{\|\mathbf{x}(kL+l_1)\|^2} \mathbf{R}_{\tilde{\mathbf{w}}}(k) \frac{\mathbf{x}(kL+l_2)(k) \mathbf{x}^T(kL+l_2)}{\|\mathbf{x}(kL+l_2)\|^2} \mathbf{v}_i \right] \\
&= +\beta^2 \sum_{l_1=0}^{L-1} \sum_{l_2=0}^{L-1} \mathbb{E} \left[ \frac{\mathbf{v}_i^T \mathbf{x}(kL+l_1)(k) \mathbf{x}^T(kL+l_1) \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathbf{x}(kL+l_2)(k) \mathbf{x}^T(kL+l_2) \mathbf{v}_i}{\|\mathbf{x}(kL+l_1)\|^2 \|\mathbf{x}(kL+l_2)\|^2} \right] \\
&= +\beta^2 \sum_{l_1=0}^{L-1} \sum_{l_2=0}^{L-1} \mathbb{E} \left[ \frac{\mathbf{v}_i^T s_{l_1}^2 r_{l_1}^2 \mathcal{V}_{l_1} \mathcal{V}_{l_1}^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) s_{l_2}^2 r_{l_2}^2 \mathcal{V}_{l_2} \mathcal{V}_{l_2}^T \mathbf{v}_i}{\|\mathbf{x}(kL+l_1)\|^2 \|\mathbf{x}(kL+l_2)\|^2} \right] \\
&= +\beta^2 \sum_{l_1=0}^{L-1} \sum_{l_2=0}^{L-1} \underbrace{\mathbb{E} [s_{l_1}^2 s_{l_2}^2]}_{=1} \underbrace{\mathbb{E} \left[ \frac{r_{l_1}^2 r_{l_2}^2}{\|\mathbf{x}(kL+l_1)\|^2 \|\mathbf{x}(kL+l_2)\|^2} \right]}_{=1} \mathbb{E} [\mathbf{v}_i^T \mathcal{V}_{l_1} \mathcal{V}_{l_1}^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathcal{V}_{l_2} \mathcal{V}_{l_2}^T \mathbf{v}_i] \\
&= +\beta^2 \sum_{l_1=0}^{L-1} \sum_{l_2=0}^{L-1} \mathbb{E} [\mathbf{v}_i^T \mathcal{V}_{l_1} \mathcal{V}_{l_1}^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathcal{V}_{l_2} \mathcal{V}_{l_2}^T \mathbf{v}_i] \\
&= +\beta^2 \sum_{l=0}^{L-1} \sum_{j=0}^{N-1} \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} (\mathbf{v}_i^T \mathbf{v}_j \mathbf{v}_j^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathbf{v}_j \mathbf{v}_j^T \mathbf{v}_i) + \beta^2 \sum_{l_1=0}^{L-1} \sum_{l_2=0}^{L-1} \left( \sum_{j_1=0}^{N-1} \sum_{j_2=0}^{N-1} \left( \frac{\lambda_{j_1} \lambda_{j_2}}{\text{Tr}[\mathbf{R}]^2} \mathbf{v}_i^T \mathbf{v}_{j_1} \mathbf{v}_{j_1}^T \mathbf{R}_{\tilde{\mathbf{w}}}(k) \mathbf{v}_{j_2} \mathbf{v}_{j_2}^T \mathbf{v}_i \right) \right) \\
&= +\beta^2 L \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \tilde{\lambda}_i(k) + \beta^2 (L-1) \frac{\lambda_i^2}{\text{Tr}[\mathbf{R}]^2} \tilde{\lambda}_i(k)
\end{aligned} \tag{IV.47}$$

Juntando todos os termos Ib, IIa, IIb e IIIc:

$$\begin{aligned}
\tilde{\lambda}_i(k+1) &= \tilde{\lambda}_i(k) - \beta L \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \tilde{\lambda}_i(k) - \beta L \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \tilde{\lambda}_i(k) + \beta^2 L \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \tilde{\lambda}_i(k) \\
&\quad + \beta^2 (L-1) \frac{\lambda_i^2}{\text{Tr}[\mathbf{R}]^2} \tilde{\lambda}_i(k) + \beta^2 \sigma^2 L \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} \mathbb{E} \left[ \frac{\mathbf{1}}{r^2} \right]
\end{aligned} \tag{IV.48}$$

$$\tilde{\lambda}_i(k+1) = \overbrace{\left\{ 1 - 2\beta L \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} + \beta^2 L \frac{\lambda_i}{\text{Tr}[\mathbf{R}]} + \beta^2(L-1) \frac{\lambda_i^2}{\text{Tr}[\mathbf{R}]^2} \right\} \tilde{\lambda}_i(k)}^{=\alpha_{\text{BNLMS}}} + \beta^2 \sigma_\nu^2 \mathbb{E} \left[ \frac{1}{r^2} \right] \frac{\lambda_i}{\text{Tr}[\mathbf{R}]}, \quad (\text{IV.49})$$

a qual generaliza a equação respectiva do NLMS (III.59). Observe que o modelo prevê estabilidade do algoritmo BNLMS quando  $|\alpha_{\text{BNLMS}}| < 1$ .

A partir da análise da equação (III.26) podemos também relacionar o comportamento do MSE em regime permanente com o tamanho da entrada  $N$ , para tanto devemos assumir duas hipóteses:

Em sendo o sinal de entrada branco,  $\lambda_i$  é igual para todo  $i$  que situa-se entre 0 e  $N-1$  ao longo da execução do algoritmo. Deste modo podemos substituí-lo por um  $\bar{\lambda}$  médio.

$$= \sigma_\nu^2 + \bar{\lambda} \sum_{i=0}^{N-1} \tilde{\lambda}_i(k), \quad (\text{IV.50})$$

O algoritmo converge, logo podemos considerar o comportamento quando  $k$  (quantidade de amostras) tende ao infinito (regime permanente).

$$\lim_{k \rightarrow +\infty} \tilde{\lambda}_i(k+1) = \tilde{\lambda}_i(k), \quad (\text{IV.51})$$

Utilizando a equação (IV.49), substituindo  $\lambda_i$  por  $\bar{\lambda}$  e fazendo  $\tilde{\lambda}_i(k+1)$  igual a  $\tilde{\lambda}_i(k)$  teremos a seguinte expressão:

$$\begin{aligned} \tilde{\lambda}_i(k) = & \left\{ 1 - 2\beta L \frac{\bar{\lambda}}{\text{Tr}[\mathbf{R}]} + \beta^2 L \frac{\bar{\lambda}}{\text{Tr}[\mathbf{R}]} + \beta^2(L-1) \frac{\bar{\lambda}^2}{\text{Tr}[\mathbf{R}]^2} \right\} \tilde{\lambda}_i(k) \\ & + \beta^2 \sigma_\nu^2 \mathbb{E} \left[ \frac{1}{r^2} \right] \frac{\bar{\lambda}}{\text{Tr}[\mathbf{R}]} \end{aligned} \quad (\text{IV.52})$$

Desenvolvendo:

$$\begin{aligned} 0 = & \left\{ -2\beta L \frac{\bar{\lambda}}{\text{Tr}[\mathbf{R}]} + \beta^2 L \frac{\bar{\lambda}}{\text{Tr}[\mathbf{R}]} + \beta^2(L-1) \frac{\bar{\lambda}^2}{\text{Tr}[\mathbf{R}]^2} \right\} \tilde{\lambda}_i(k) + \beta^2 \sigma_\nu^2 \mathbb{E} \left[ \frac{1}{r^2} \right] \frac{\bar{\lambda}}{\text{Tr}[\mathbf{R}]} \\ & \left\{ 2\beta L \frac{\bar{\lambda}}{\text{Tr}[\mathbf{R}]} - \beta^2 L \frac{\bar{\lambda}}{\text{Tr}[\mathbf{R}]} - \beta^2(L-1) \frac{\bar{\lambda}^2}{\text{Tr}[\mathbf{R}]^2} \right\} \tilde{\lambda}_i(k) = \beta^2 \sigma_\nu^2 \mathbb{E} \left[ \frac{1}{r^2} \right] \frac{\bar{\lambda}}{\text{Tr}[\mathbf{R}]} \end{aligned} \quad (\text{IV.53})$$

Dividindo toda a equação por  $\frac{\beta^2 L \bar{\lambda}}{\text{Tr}[\mathbf{R}]}$ :

$$\begin{aligned}
\tilde{\lambda}_i(k) \left\{ \frac{2}{\beta} - 1 - \frac{(L-1)\bar{\lambda}_i}{L\text{Tr}[\mathbf{R}]} \right\} &= \sigma_\nu^2 \mathbb{E} \left[ \frac{1}{r^2} \right] \\
\tilde{\lambda}_i(k) \left\{ \frac{2L\text{Tr}[\mathbf{R}] - \beta L\text{Tr}[\mathbf{R}] - \beta(L-1)\bar{\lambda}_i}{\beta L\text{Tr}[\mathbf{R}]} \right\} &= \sigma_\nu^2 \mathbb{E} \left[ \frac{1}{r^2} \right] \\
\tilde{\lambda}_i(k) \left\{ \frac{L\text{Tr}[\mathbf{R}](2-\beta) - \beta(L-1)\bar{\lambda}_i}{\beta L\text{Tr}[\mathbf{R}]} \right\} &= \sigma_\nu^2 \mathbb{E} \left[ \frac{1}{r^2} \right]
\end{aligned} \tag{IV.54}$$

Isolando  $\tilde{\lambda}_i(k)$ :

$$\tilde{\lambda}_i(k) = \frac{\sigma_\nu^2 \beta \mathbb{E} \left[ \frac{1}{r^2} \right] L\text{Tr}[\mathbf{R}]}{L\text{Tr}[\mathbf{R}] (2-\beta) - \beta(L-1)\bar{\lambda}_i} \tag{IV.55}$$

Agora que temos o  $\tilde{\lambda}_i(k)$  podemos aplicar a mesma hipótese de  $k$  tendendo ao infinito ao MSE da equação (III.26):

$$\begin{aligned}
\lim_{k \rightarrow +\infty} \xi(k) &= \sigma_\nu^2 + \bar{\lambda}N \quad \overbrace{\tilde{\lambda}_i(k)}^{\sigma_\nu^2 \beta \mathbb{E} \left[ \frac{1}{r^2} \right] L\text{Tr}[\mathbf{R}]} \\
&= \frac{\sigma_\nu^2 \beta \mathbb{E} \left[ \frac{1}{r^2} \right] L\text{Tr}[\mathbf{R}]}{L\text{Tr}[\mathbf{R}] (2-\beta) - \beta(L-1)\bar{\lambda}_i}
\end{aligned} \tag{IV.56}$$

Conseguimos assim, considerando a hipótese de regime permanente, relacionar o MSE ao tamanho do filtro  $N$  e a quantidade de blocos  $L$ , lembrando que para quantidade de blocos  $L = 1$  a equação (IV.56) degenera-se naturalmente em (III.67), como é esperado.

## IV.5 Comentários Finais

Neste capítulo, o *modelo de Slock* foi generalizado para os algoritmos adaptativos em bloco BLMS e BNLMS. Através destas generalizações é possível prever o comportamento do MSE nos algoritmos e modelar o comportamento de divergência e estabilidade em regime permanente. Também conseguimos através do uso de uma hipótese de regime permanente relacionar o MSE ao tamanho do filtro  $N$  e à quantidade de blocos  $L$ . No próximo capítulo, as predições do modelo resultante serão comparadas com curvas advindas de simulações a fim de testar a aderência do modelo teórico ao comportamento na prática dos algoritmos em questão.

## Capítulo V Resultados

### V.1 Introdução

A partir do modelo estocástico proposto por Slock e explicado na seção III.2, foram desenvolvidas as análises teóricas dos algoritmos BLMS e BNLMS nas seções IV.3 e IV.4 deste trabalho, respectivamente. Estas análises, por fim, modelam o comportamento do  $\tilde{\lambda}_i$  de tal forma que a partir dele podemos representar também de forma teórica o MSE, além de formular, também de forma teórica, considerações a respeito do limitante superior teórico para o fator de aprendizagem  $\beta$  que evita a divergência dos algoritmos analisados. O objetivo das simulações realizadas é comparar os resultados teóricos com os encontrados na execução dos algoritmos estudados. Afim de aferir a robustez da modelagem proposta foram escolhidos dois cenários com configurações diferentes para cada algoritmo simulado (BLMS e BNLMS), neste modo tentamos demonstrar que dentro da faixa aceitável de parametrização os algoritmos modelados comportam-se da maneira esperada, possibilitando a comparação entre as curvas teórica e simulada.

Todas os códigos para as simulações desta seção foram construídos na linguagem de programação do Software MATLAB. O ambiente experimental utilizado para simular e obter os resultados foi composto por um Hardware com a seguinte configuração:

- Fabricante: DELL
- Modelo: Latitude E6410
- CPU: Intel® Core™ i5 CPU M 560 @ 2.67GHz × 4
- Memoria: 8GB RAM
- Disco Rígido: 240GB SSD
- Sistema Operacional: Ubuntu 20.04.3 LTS 64 bits
- Versão do Software para Simulações: OCTAVE 6.4.0 for Linux Ubuntu 64 bits

## V.2 Simulação 1 - MSE e $\tilde{\lambda}_i$

Nesta simulação, os algoritmos BLMS e BNLMS foram implementados de forma a podermos comparar as curvas simuladas e teóricas das análises propostas, baseando-se no comportamento do MSE, de dois  $\tilde{\lambda}_i$  diferentes escolhidos ao acaso e um  $\bar{\lambda}$  médio. Com o objetivo de demonstrar que a capacidade de simulação do modelo não é afetado pelos parâmetros utilizados na simulação, foram utilizadas duas configurações de parâmetros diferentes para cada algoritmo, para cada cenário variou-se o tamanho do bloco ( $L$ ).

### V.2.1 BLMS

**Cenário 1 →  $N = 10$**

Os parâmetros utilizados para a simulação do BLMS neste cenário foram:  $N=10$ ,  $\beta = 3 \times 10^{-3}$ ,  $\sigma_\nu^2 = 10^{-2}$ , número de ensaios Monte Carlo = 1000 e um sinal de entrada colorido obtido pelo colorimento de um sinal branco, gaussiano e de variância unitária pelo filtro  $B(z) = 1 - 0,8z^{-1} + 0.6z^{-2}$ . Todos os coeficientes da função de transferência ideal são unitários.

$L = 1$

Nesta simulação foi utilizada a quantidade de blocos  $L = 1$ . Neste caso, o algoritmo em blocos se degenera na sua versão clássica (ou seja, LMS). Como a análise proposta é genérica, veremos que ela também modela o LMS, como um caso particular. A Figura V.1 apresenta a evolução do MSE empírico e do teórico. Observa-se boa aderência do modelo à curva, bem como aos parâmetros  $\tilde{\lambda}_1(k)$  e  $\tilde{\lambda}_4(k)$  (Figuras V.2 e V.3). Como há diversos  $\tilde{\lambda}_i(k)$ , dois deles foram escolhidos para efeitos de aferição da qualidade de predição. Também foi traçado na curva V.4 um  $\bar{\lambda}$  médio calculado através da média aritmética de todos os  $\tilde{\lambda}_i(k)$ .

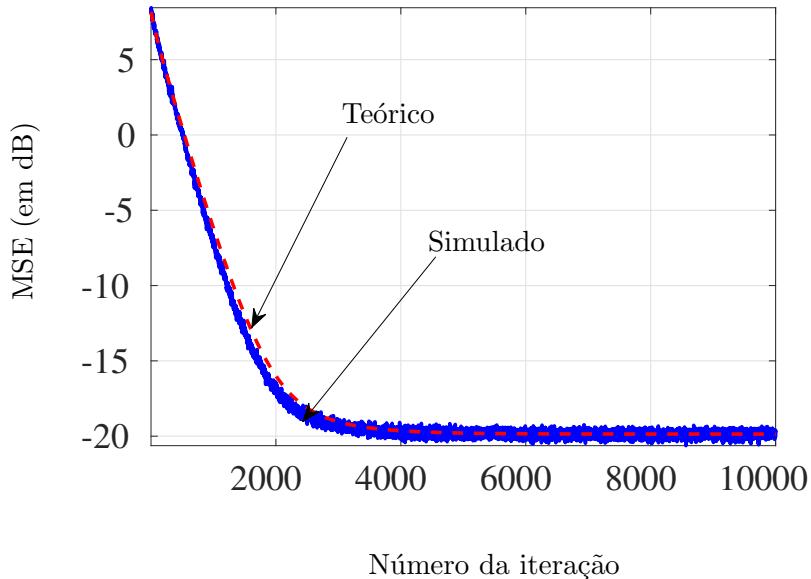


Figura V.1: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 1$  e  $N = 10$

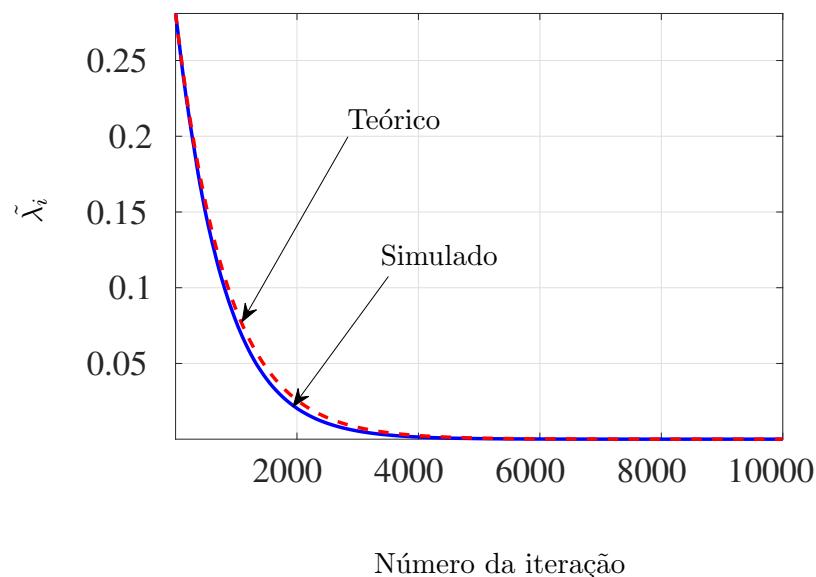


Figura V.2: Evolução do  $\tilde{\lambda}_i(k)$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BLMS com  $L = 1$  e  $N = 10$ .

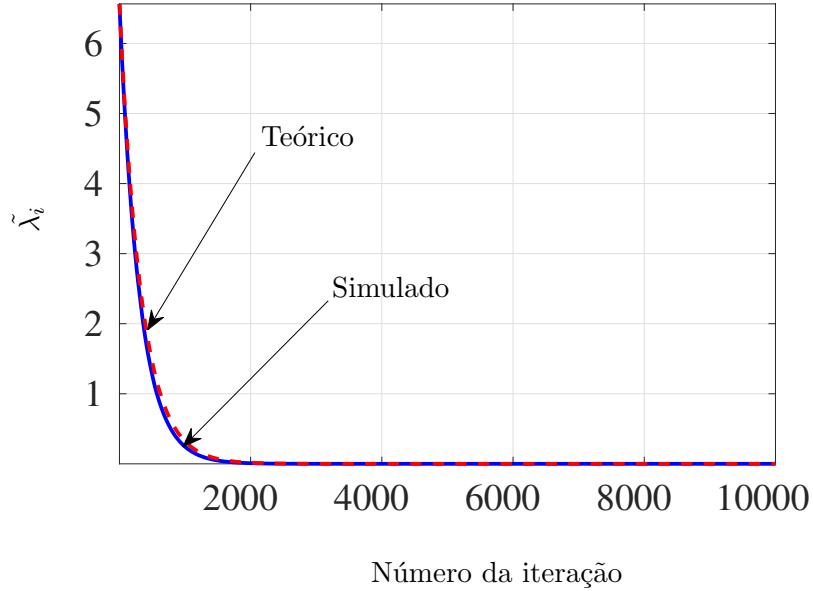


Figura V.3: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BLMS com  $L = 1$  e  $N = 10$

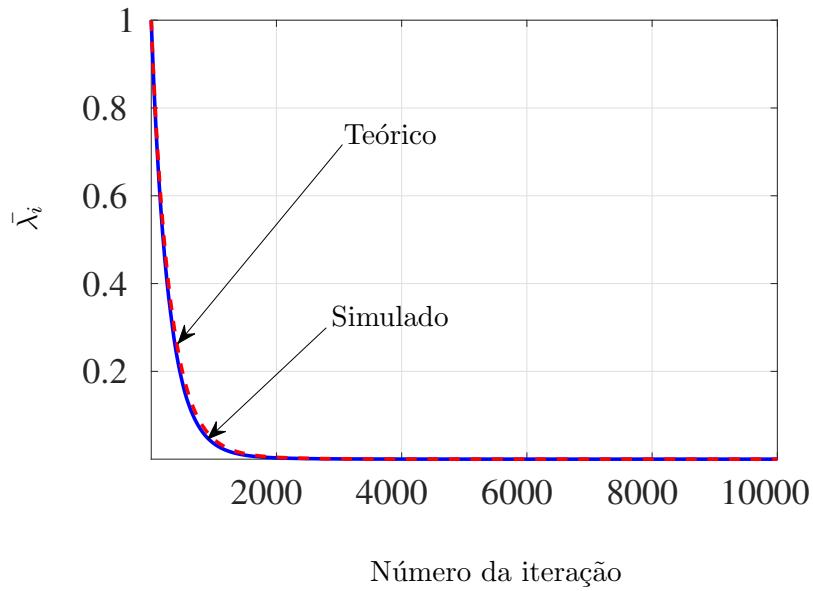


Figura V.4: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 1$  e  $N = 10$

$L = 2$

Nesta simulação foi utilizada a quantidade de blocos  $L = 2$ . A Figura V.5 apresenta a evolução do MSE empírico e do teórico. Observa-se boa aderência do modelo à curva, bem como aos parâmetros  $\tilde{\lambda}_1(k)$  e  $\tilde{\lambda}_4(k)$  (Figuras V.6 e V.7). Como há diversos  $\tilde{\lambda}_i(k)$ , dois deles foram escolhidos para efeitos de aferição da qualidade de predição. Também foi traçado na curva V.8 um  $\bar{\lambda}$  médio calculado através da média aritmética de todos os  $\tilde{\lambda}_i(k)$ .

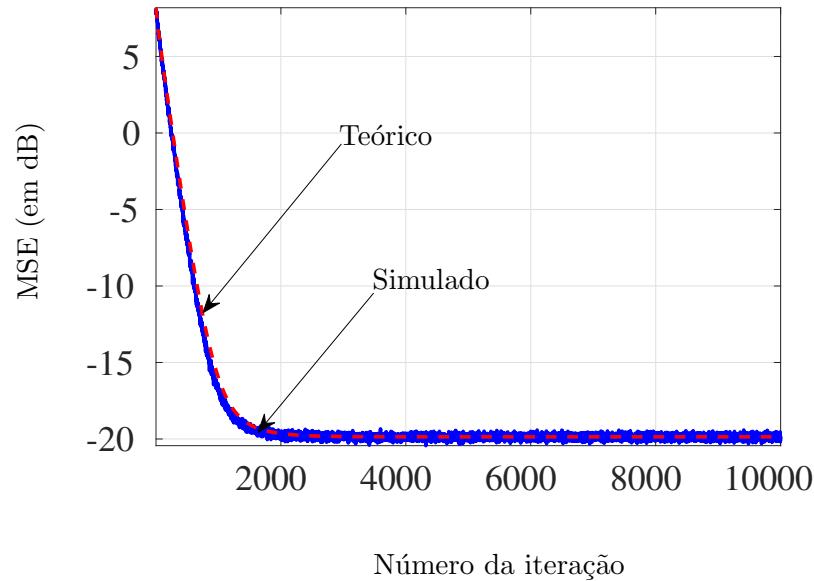


Figura V.5: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 2$  e  $N = 10$

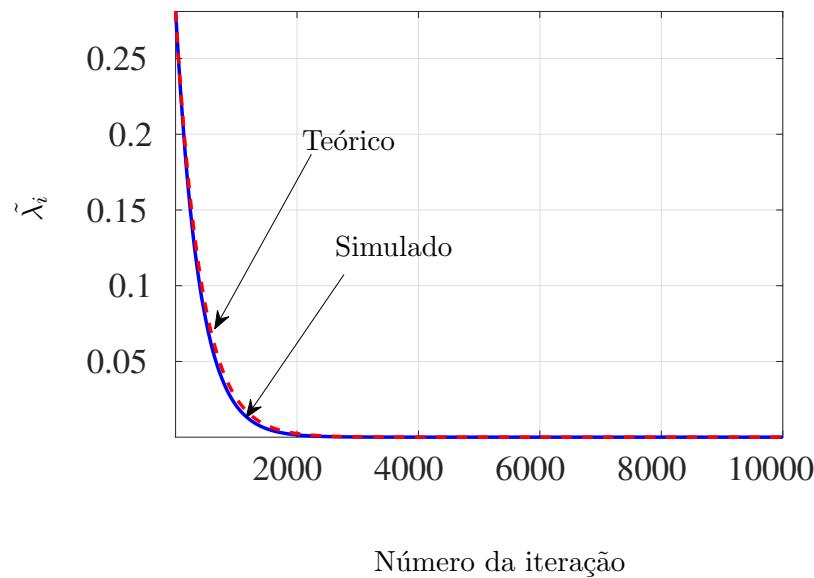


Figura V.6: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BLMS com  $L = 2$  e  $N = 10$ .

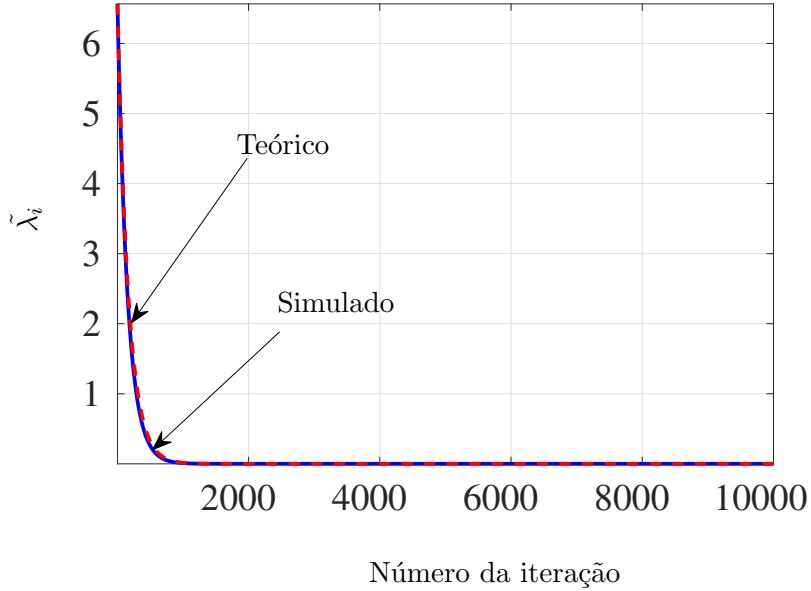


Figura V.7: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BLMS com  $L = 2$  e  $N = 10$ .

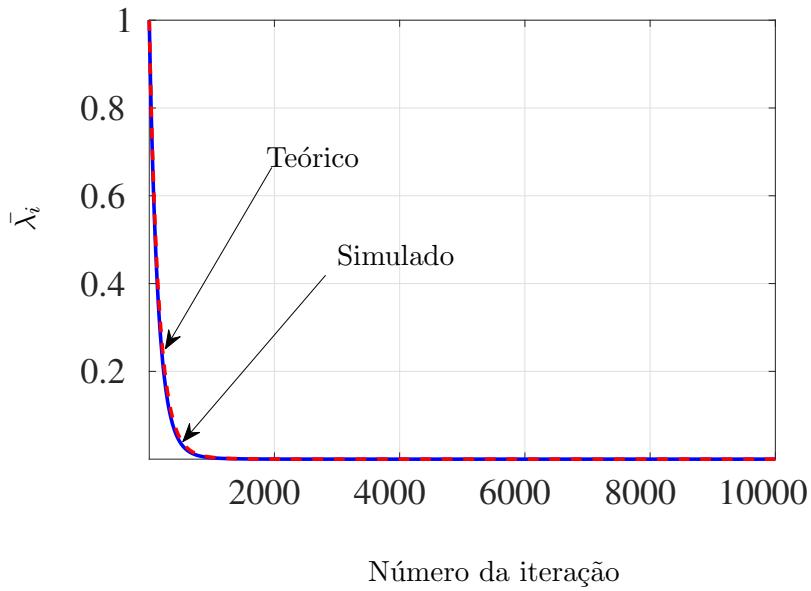


Figura V.8: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 2$  e  $N = 10$

$L = 3$

Nesta simulação foi utilizada a quantidade de blocos  $L = 3$ . A Figura V.9 apresenta a evolução do MSE empírico e do teórico. Observa-se boa aderência do modelo à curva, bem como aos parâmetros  $\tilde{\lambda}_1(k)$  e  $\tilde{\lambda}_4(k)$  (Figuras V.10 e V.11). Como há diversos  $\tilde{\lambda}_i(k)$ , dois deles foram escolhidos para efeitos de aferição da qualidade de predição. Também foi traçado na curva V.12 um  $\bar{\lambda}$  médio calculado através da média aritmética de todos os  $\tilde{\lambda}_i(k)$ .

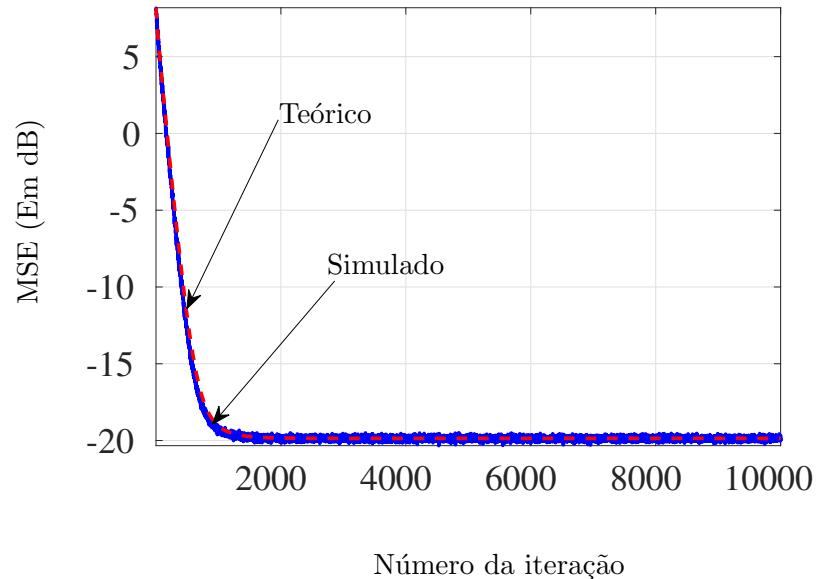


Figura V.9: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 3$  e  $N = 10$

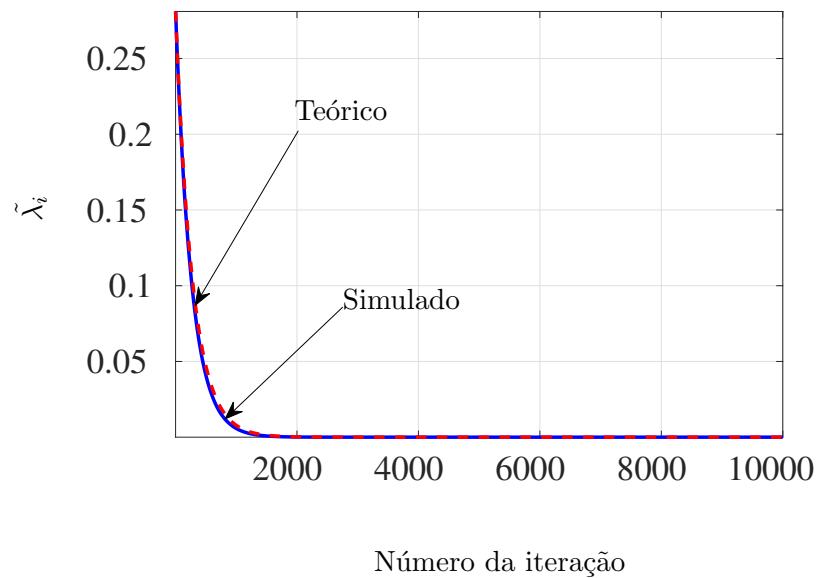


Figura V.10: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BLMS com  $L = 3$  e  $N = 10$

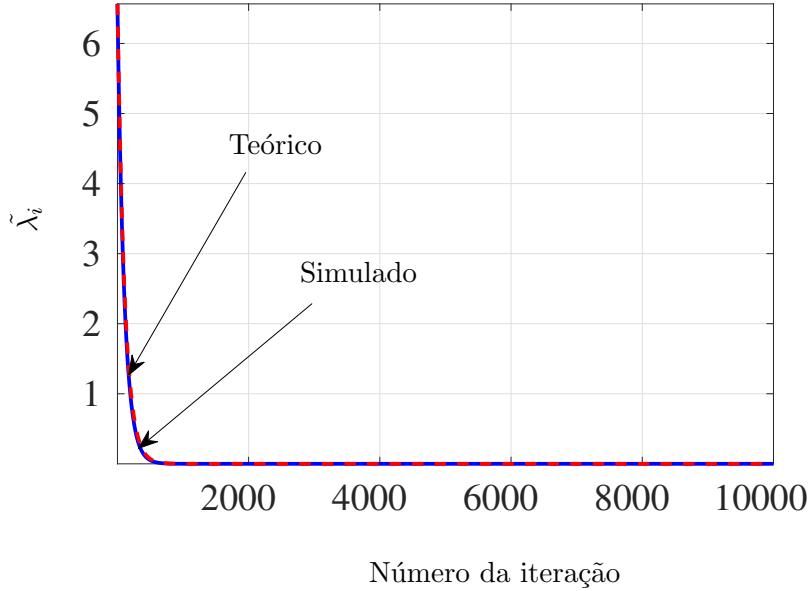


Figura V.11: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BLMS com  $L = 3$  e  $N = 10$

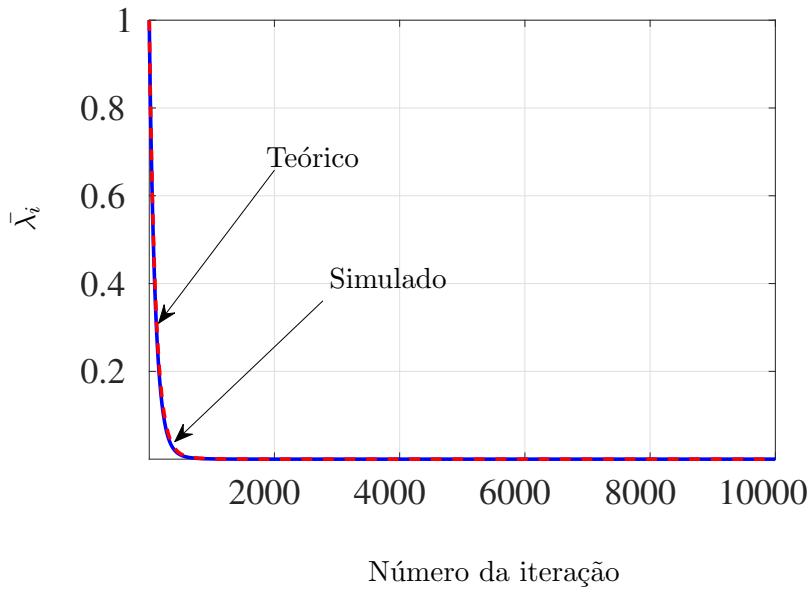


Figura V.12: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 3$  e  $N = 10$

$L = 4$

Nesta simulação foi utilizada a quantidade de blocos  $L = 4$ . A Figura V.13 apresenta a evolução do MSE empírico e do teórico. Observa-se boa aderência do modelo à curva, bem como aos parâmetros  $\tilde{\lambda}_1(k)$  e  $\tilde{\lambda}_4(k)$  (Figuras V.14 e V.15). Como há diversos  $\tilde{\lambda}_i(k)$ , dois deles foram escolhidos para efeitos de aferição da qualidade de predição. Também foi traçado na curva V.16 um  $\bar{\lambda}$  médio calculado através da média aritmética de todos os  $\tilde{\lambda}_i(k)$ .

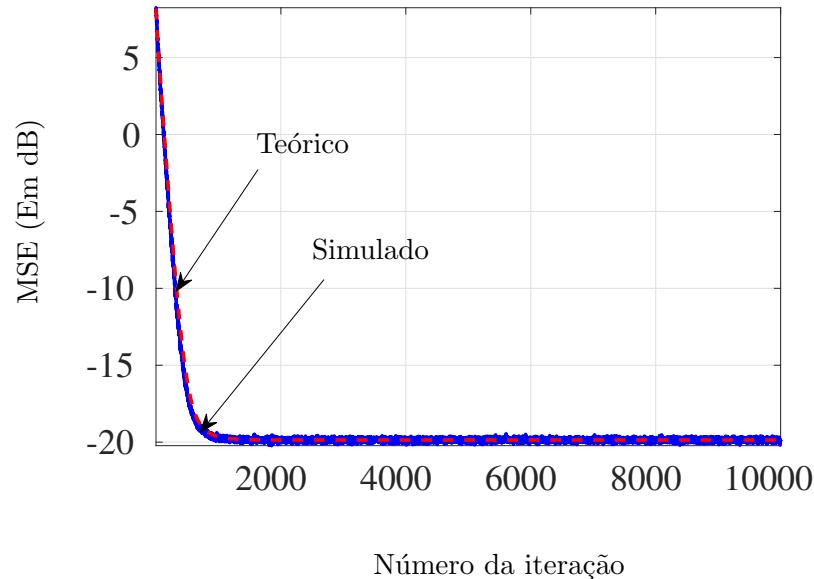


Figura V.13: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 4$  e  $N = 10$ .

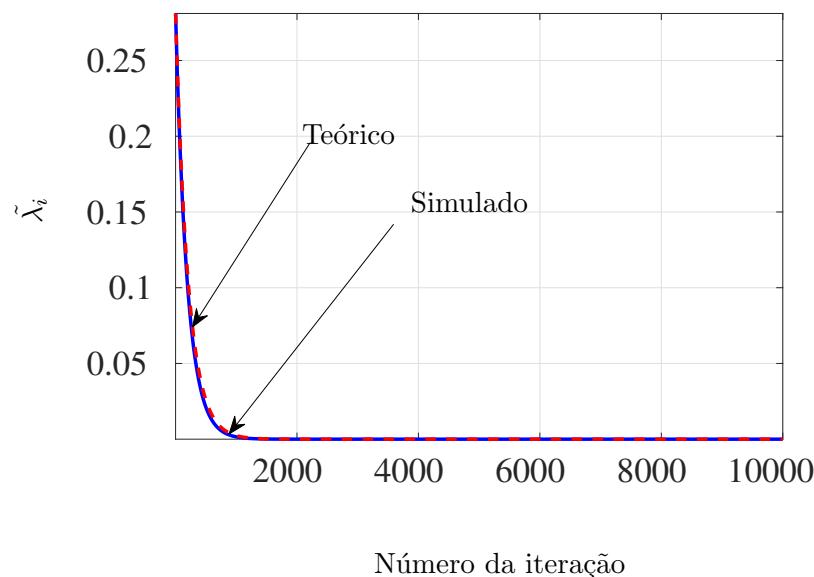


Figura V.14: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BLMS com  $L = 4$  e  $N = 10$

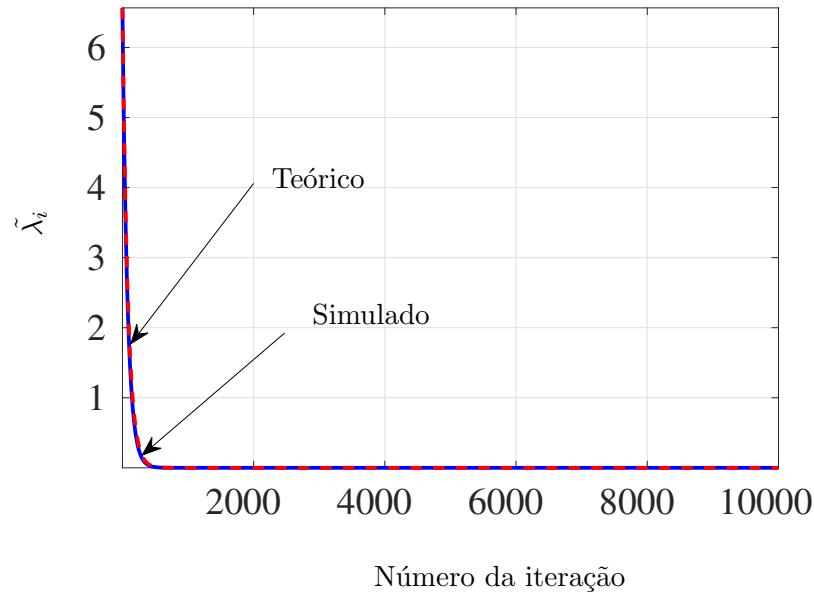


Figura V.15: Evolução do  $\tilde{\lambda}_i(k)$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BLMS com  $L = 4$  e  $N = 10$

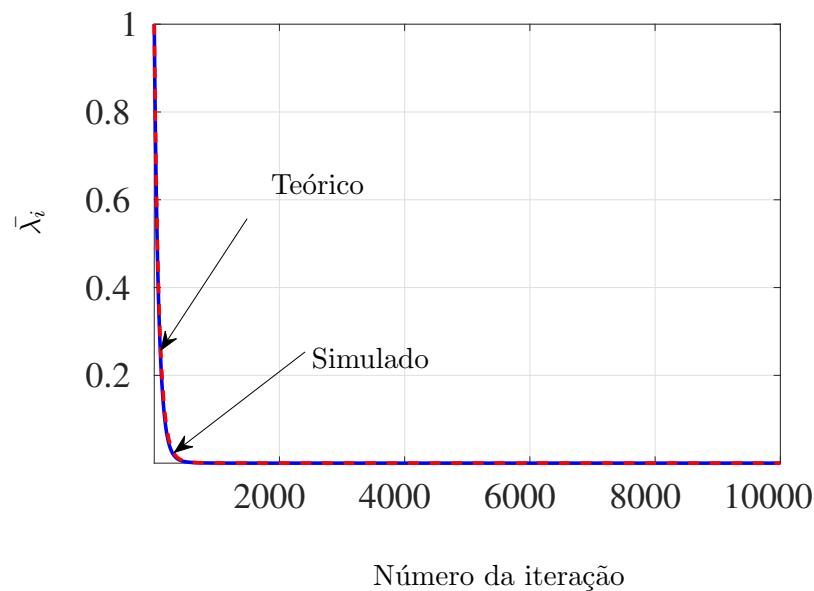


Figura V.16: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 4$  e  $N = 10$

## Comentários

As simulações deste cenário apresentaram todas uma excelente aderência do modelo simulado com o teórico. Na simulação com quantidade de blocos  $L = 4$ , o MSE claramente parece alcançar o regime permanente mais rapidamente, e isto se deve a própria característica da modelagem em blocos.

**Cenário 2 →  $N = 60$**

Os parâmetros utilizados para a simulação do BLMS neste cenário foram:  $N=60$ ,  $\beta = 1.5 \times 10^{-3}$ ,  $\sigma_\nu^2 = 2.510^{-2}$ , número de ensaios Monte Carlo = 1000 e um sinal de entrada colorido obtido pelo colorimento de um sinal branco, gaussiano e de variância unitária pelo filtro  $B(z) = 1 - 0,7z^{-1} + 0.8z^{-2}$ . Todos os coeficientes da função de transferência ideal são unitários.

$L = 1$

Nesta simulação foi utilizada a quantidade de blocos  $L = 1$ . Neste caso, o algoritmo em blocos se degenera na sua versão clássica (ou seja, LMS). Como a análise proposta é genérica, veremos que ela também modela o LMS, como um caso particular. A Figura V.17 apresenta a evolução do MSE empírico e do teórico. Observa-se boa aderência do modelo à curva, bem como aos parâmetros  $\tilde{\lambda}_1(k)$  e  $\tilde{\lambda}_4(k)$  (Figuras V.18 e V.19). Como há diversos  $\tilde{\lambda}_i(k)$ , dois deles foram escolhidos para efeitos de aferição da qualidade de predição. Também foi traçado na curva V.20 um  $\bar{\lambda}$  médio calculado através da média aritmética de todos os  $\tilde{\lambda}_i(k)$ .

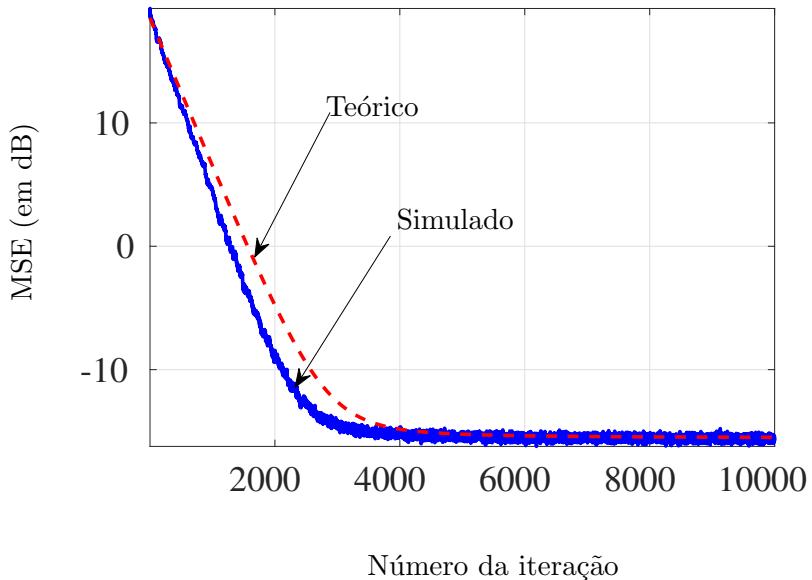


Figura V.17: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 1$  e  $N = 60$

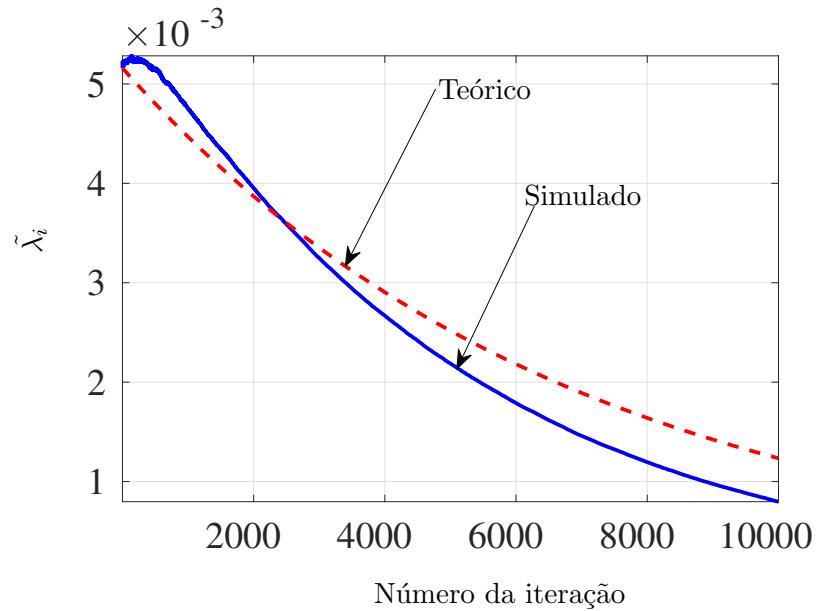


Figura V.18: Evolução do  $\tilde{\lambda}_i(k)$  teórico e simulado com  $i = 3$  ao longo das iterações do algoritmo BLMS com  $L = 1$  e  $N = 60$ .

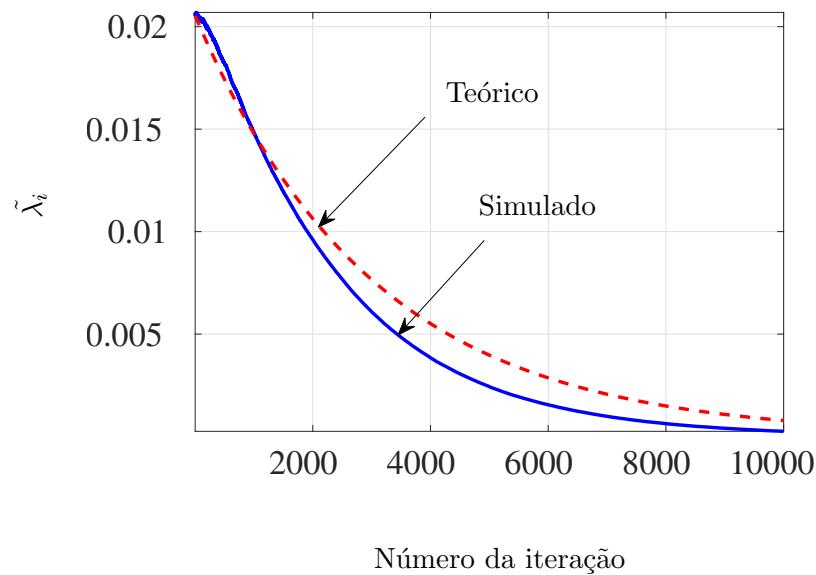


Figura V.19: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 7$  ao longo das iterações do algoritmo BLMS com  $L = 1$  e  $N = 60$

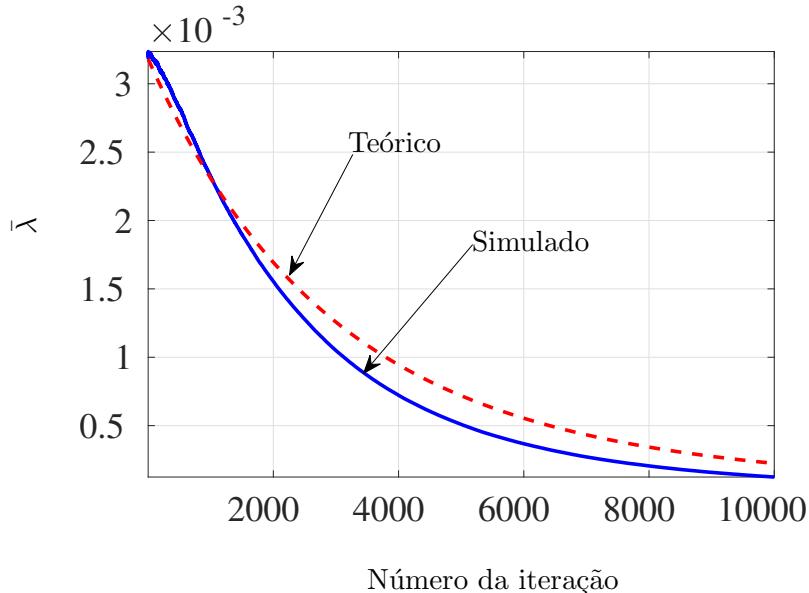


Figura V.20: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 1$  e  $N = 60$

$$L = 2$$

A Figura V.21 apresenta a evolução do MSE empírico e do teórico. Observa-se boa aderência do modelo à curva, bem como aos parâmetros  $\tilde{\lambda}_1(k)$  e  $\tilde{\lambda}_4(k)$  (Figuras V.22 e V.23). Como há diversos  $\tilde{\lambda}_i(k)$ , dois deles foram escolhidos para efeitos de aferição da qualidade de predição. Também foi traçado na curva V.24 um  $\bar{\lambda}$  médio calculado através da média aritmética de todos os  $\tilde{\lambda}_i(k)$ .

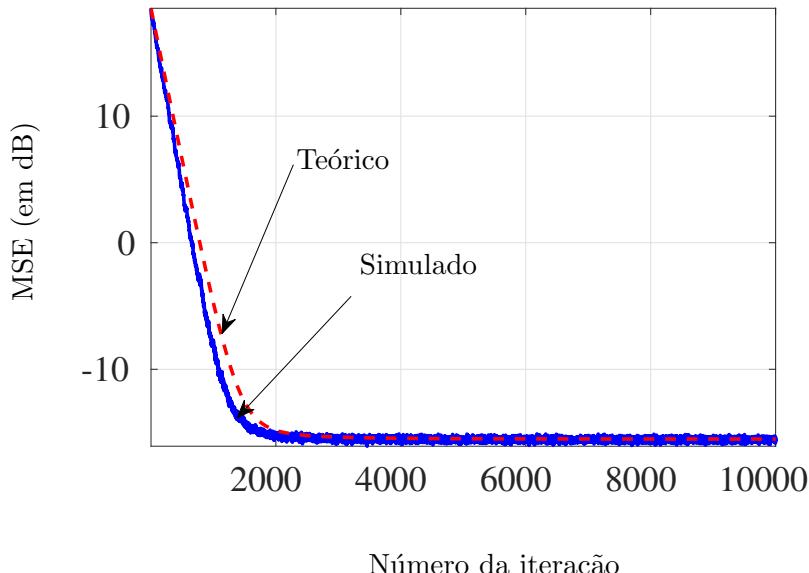


Figura V.21: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 2$  e  $N = 60$

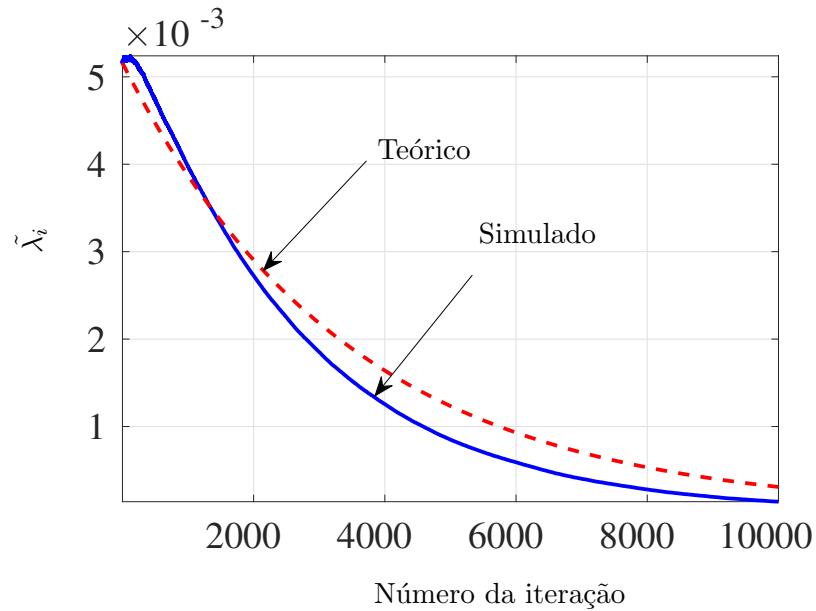


Figura V.22: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 3$  ao longo das iterações do algoritmo BLMS com  $L = 2$  e  $N = 60$ .

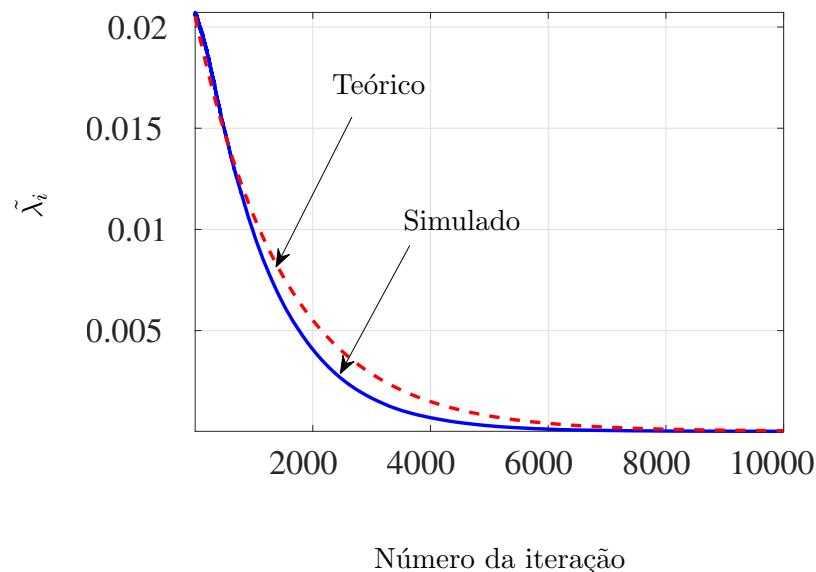


Figura V.23: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 7$  ao longo das iterações do algoritmo BLMS com  $L = 2$  e  $N = 60$ .

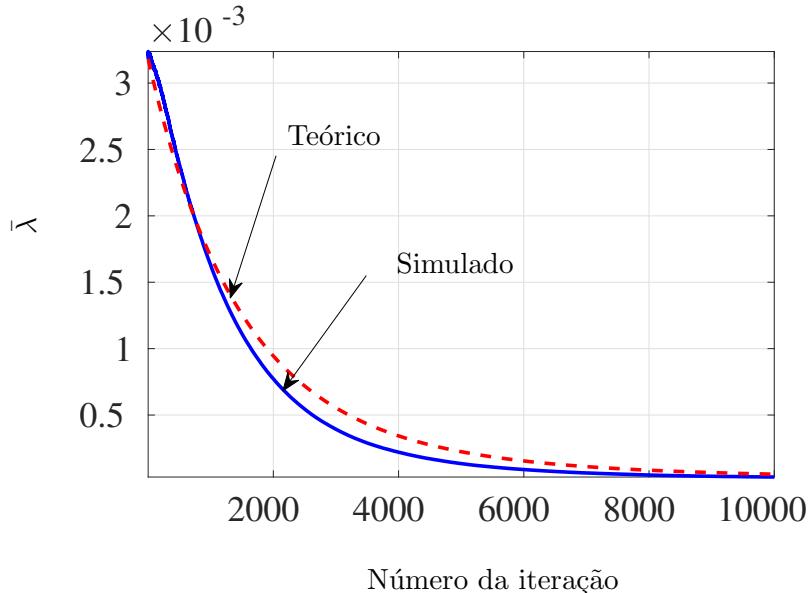


Figura V.24: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 2$  e  $N = 60$

$L = 3$

A Figura V.25 apresenta a evolução do MSE empírico e do teórico. Observa-se boa aderência do modelo à curva, bem como aos parâmetros  $\tilde{\lambda}_1(k)$  e  $\tilde{\lambda}_4(k)$  (Figuras V.26 e V.27). Como há diversos  $\tilde{\lambda}_i(k)$ , dois deles foram escolhidos para efeitos de aferição da qualidade de predição. Também foi traçado na curva V.32 um  $\bar{\lambda}$  médio calculado através da média aritmética de todos os  $\tilde{\lambda}_i(k)$ .

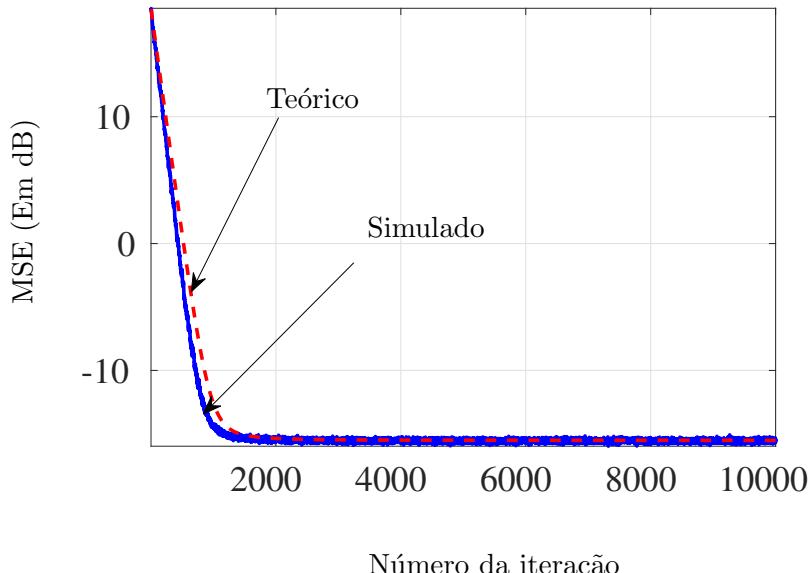


Figura V.25: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 3$

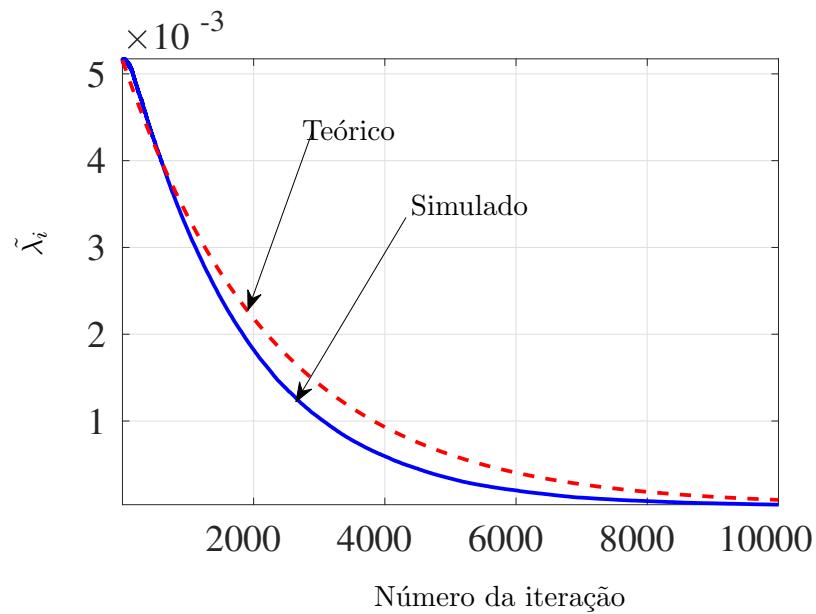


Figura V.26: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 3$  ao longo das iterações do algoritmo BLMS com  $L = 3$  e  $N = 60$

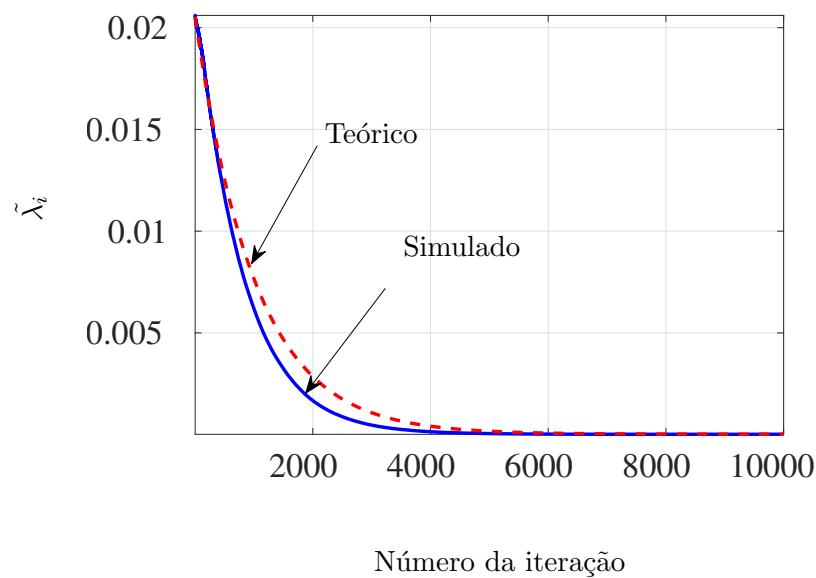


Figura V.27: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 7$  ao longo das iterações do algoritmo BLMS com  $L = 3$  e  $N = 60$

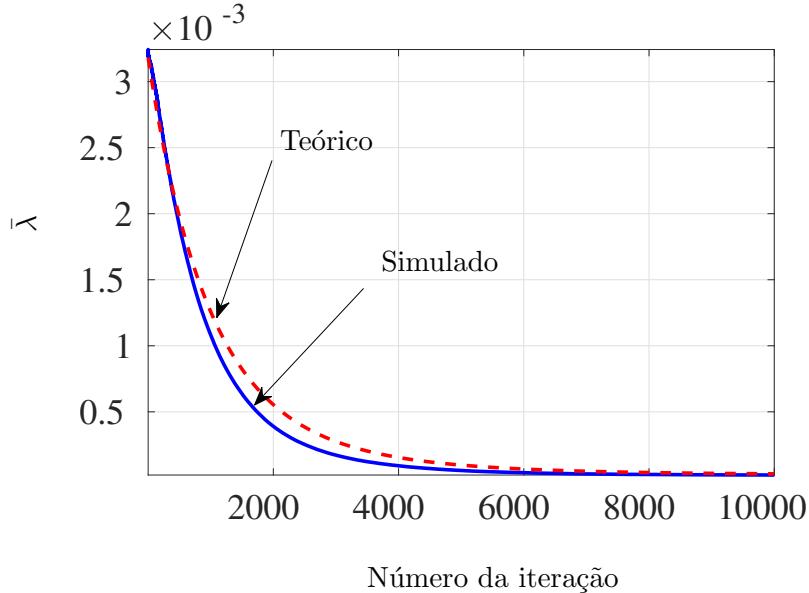


Figura V.28: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 3$  e  $N = 60$

$L = 4$

No cenário 4, o algoritmo BLMS foi representado pela Figura V.29 que apresenta a evolução do MSE empírico e do teórico. Observa-se boa aderência do modelo à curva, bem como aos parâmetros  $\tilde{\lambda}_1(k)$  e  $\tilde{\lambda}_4(k)$  (Figuras V.30 e V.31). Como há diversos  $\tilde{\lambda}_i(k)$ , dois deles foram escolhidos para efeitos de aferição da qualidade de predição. Também foi traçado na curva V.32 um  $\bar{\lambda}$  médio calculado através da média aritmética de todos os  $\tilde{\lambda}_i(k)$ .

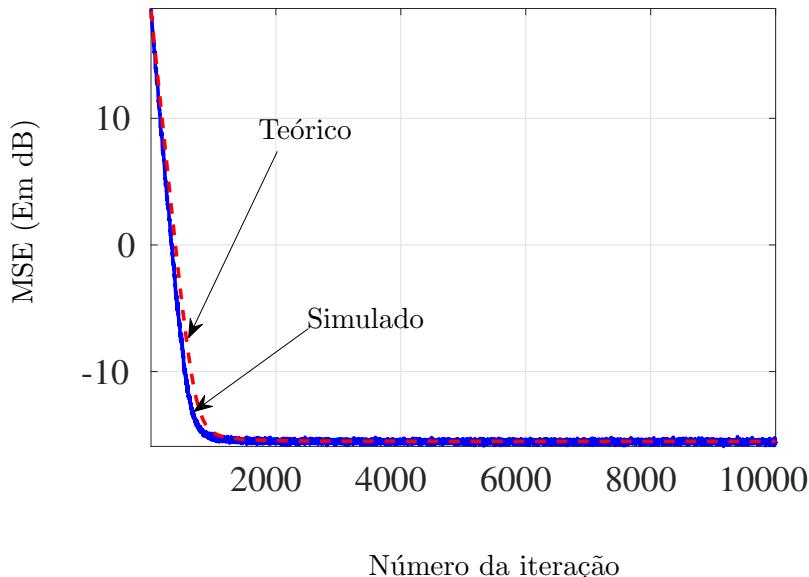


Figura V.29: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 4$ .

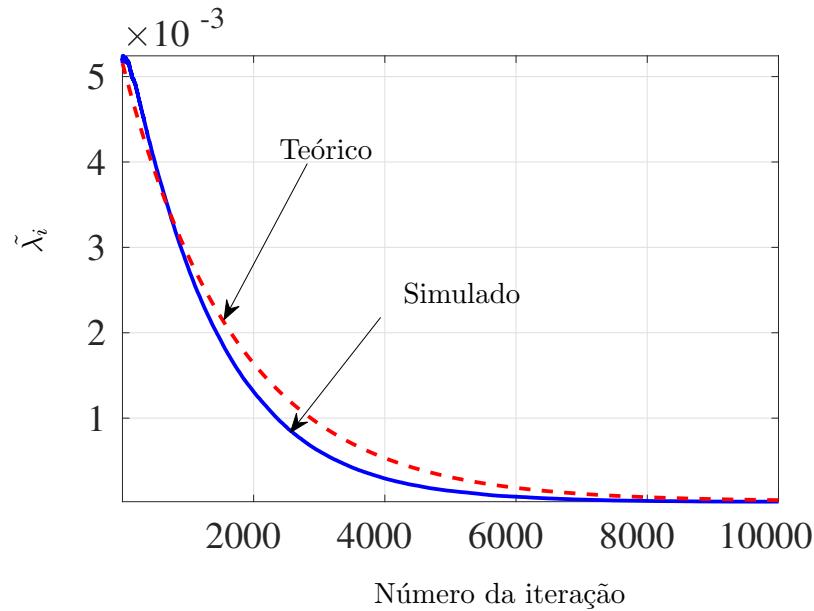


Figura V.30: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 3$  ao longo das iterações do algoritmo BLMS com  $L = 4$  e  $N = 60$

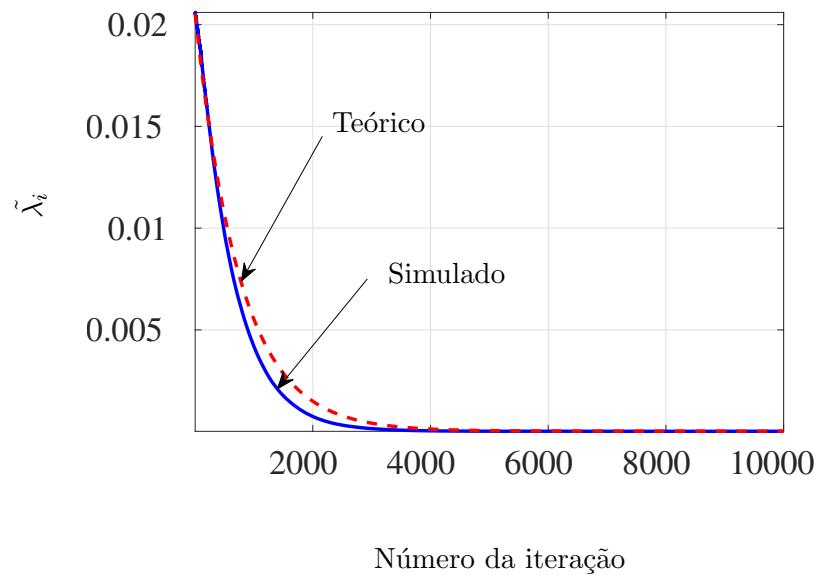


Figura V.31: Evolução do  $\tilde{\lambda}_i(k)$  teórico e simulado com  $i = 7$  ao longo das iterações do algoritmo BLMS com  $L = 4$  e  $N = 60$ .

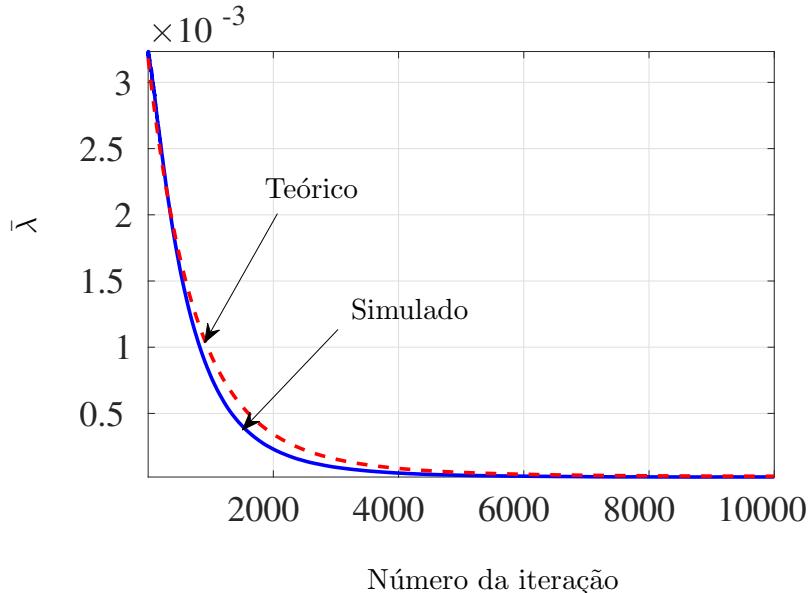


Figura V.32: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 4$  e  $N = 60$

### Comentários

O cenário com  $N = 60$  demonstra que mesmo variando parâmetros e aumentando muito o tamanho do filtro o modelo proposto continua aderente ao simulado. Em todos as quantidades de blocos  $L$  simuladas, as curvas com  $\tilde{\lambda}_i$  simulados demonstraram-se levemente deslocadas no inicio das iterações do algoritmo, podendo indicar que o tamanho do filtro cause alguma perturbação antes da fase de regime permanente, muito possivelmente em função das aproximações utilizadas tais como a hipótese da Independência e a hipótese do Ruído.

## V.2.2 BNLMS

### Cenário 1 $\rightarrow N = 10$

Os parâmetros utilizados para a simulação do BNLMS neste cenário foram:  $N=10$ ,  $0.4$ ,  $\sigma_\nu^2 = 10^{-2}$ , número de ensaios Monte Carlo = 1000 e um sinal de entrada colorido obtido pelo colorimento de um sinal branco, gaussiano e de variância unitária pelo filtro  $B(z) = 1 - 0,8z^{-1} + 0.6z^{-2}$ . Todos os coeficientes da função de transferência ideal são unitários.

$L = 1$

Nesta simulação foi utilizada a quantidade de blocos  $L = 1$ . Neste caso, o algoritmo em blocos se degenera na sua versão clássica (ou seja, NLMS). Como a análise proposta é genérica, veremos que ela também modela o NLMS, como um caso particular. A Figura VII.93 apresenta a evolução do MSE empírico e do teórico. Observa-se boa aderência do modelo à curva, bem como aos parâmetros

$\tilde{\lambda}_1(k)$  e  $\tilde{\lambda}_4(k)$  (Figuras VII.94 e VII.95). Como há diversos  $\tilde{\lambda}_i(k)$ , dois deles foram escolhidos para efeitos de aferição da qualidade de predição. Também foi traçado na curva VII.96 um  $\bar{\lambda}$  médio calculado através da média aritmética de todos os  $\tilde{\lambda}_i(k)$ .

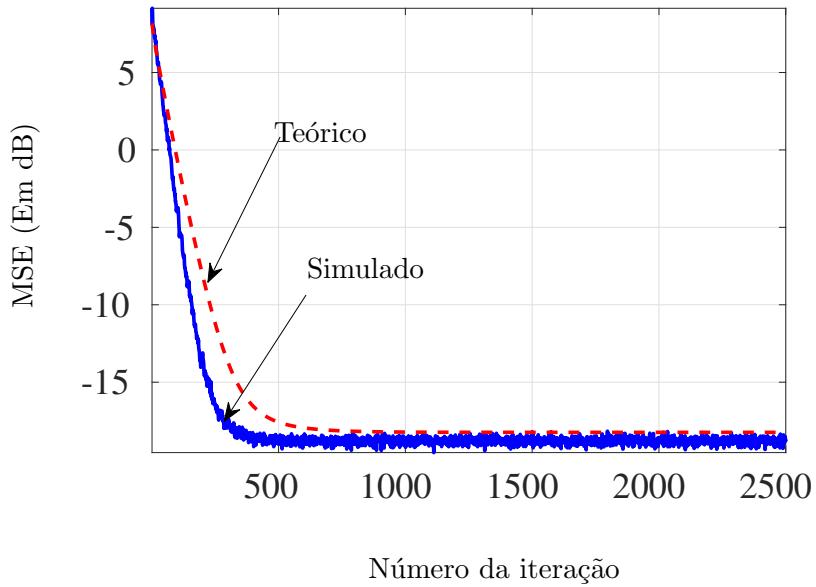


Figura V.33: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

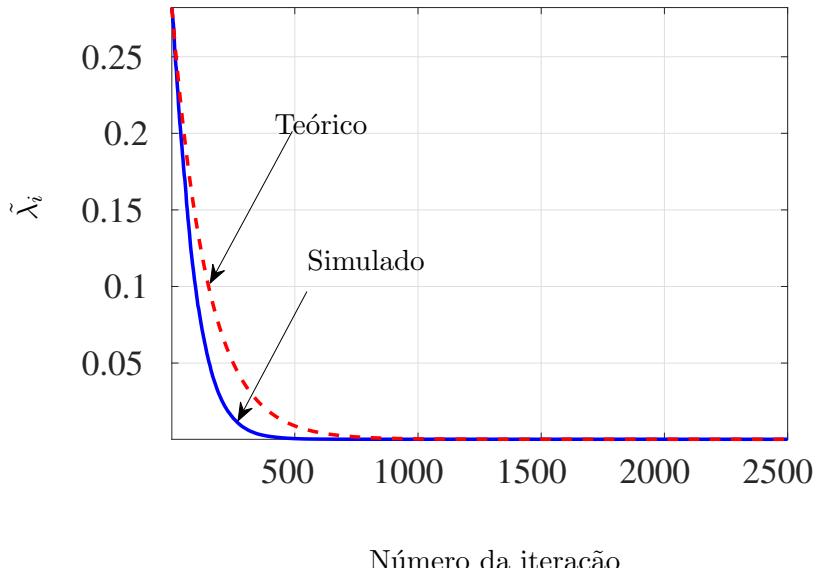


Figura V.34: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

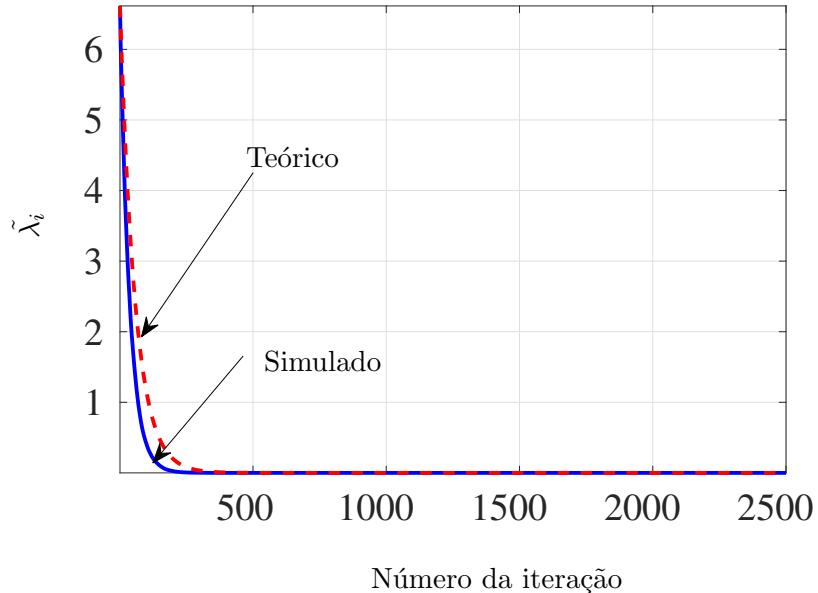


Figura V.35: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

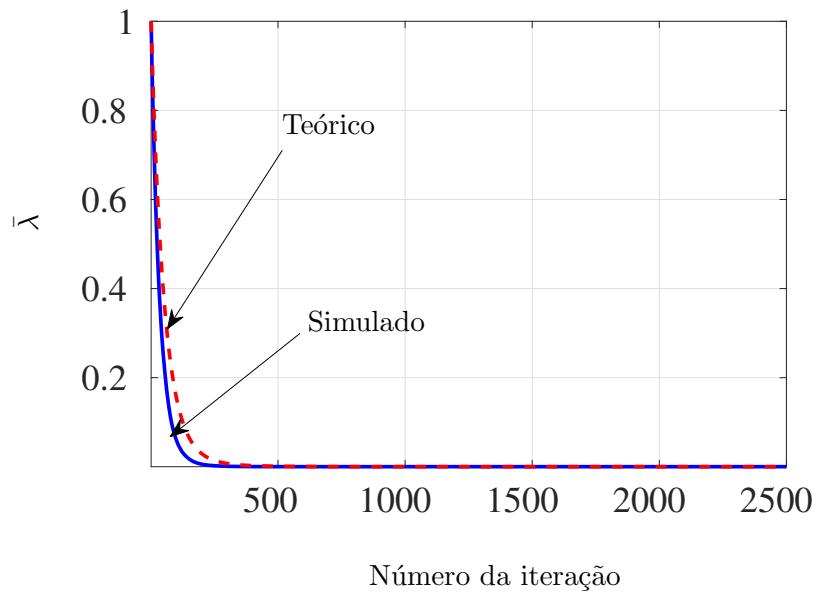


Figura V.36: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

$L = 2$

Nesta simulação foi utilizada a quantidade de blocos  $L = 1$ . A Figura V.37 apresenta a evolução do MSE empírico e do teórico. Observa-se boa aderência do modelo à curva, bem como aos parâmetros  $\tilde{\lambda}_1(k)$  e  $\tilde{\lambda}_4(k)$  (Figuras V.38 e V.39). Como há diversos  $\tilde{\lambda}_i(k)$ , dois deles foram escolhidos para efeitos de aferição da qualidade de predição. Também foi traçado na curva V.40 um  $\bar{\lambda}$  médio calculado através da média aritmética de todos os  $\tilde{\lambda}_i(k)$ .

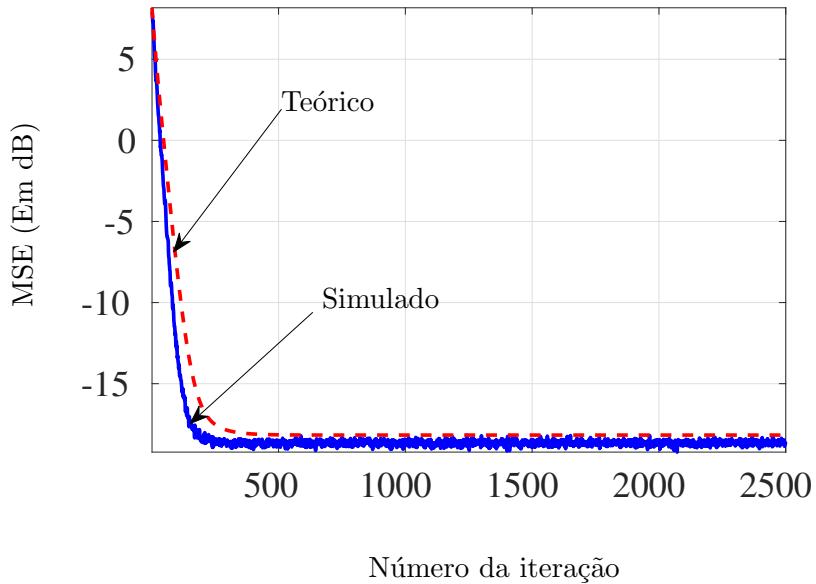


Figura V.37: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 2$  e  $N = 10$

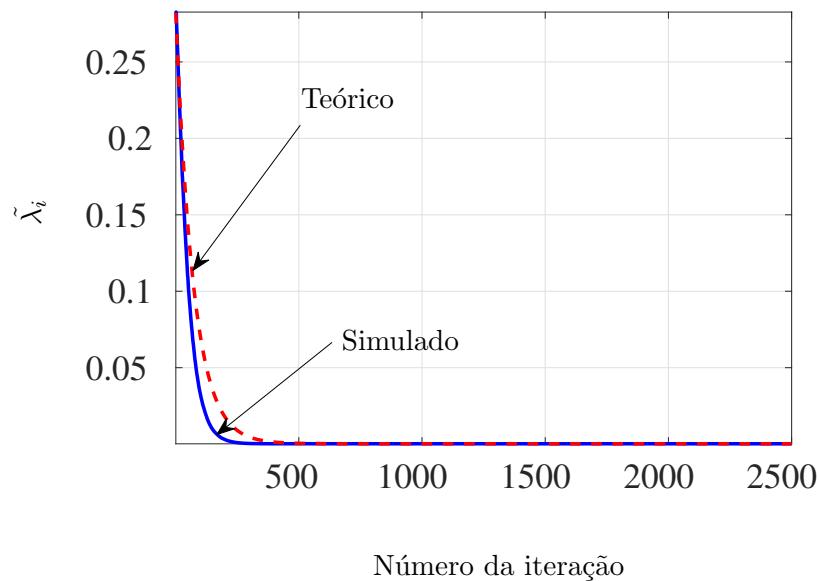


Figura V.38: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BNLMS com  $L = 2$  e  $N = 10$

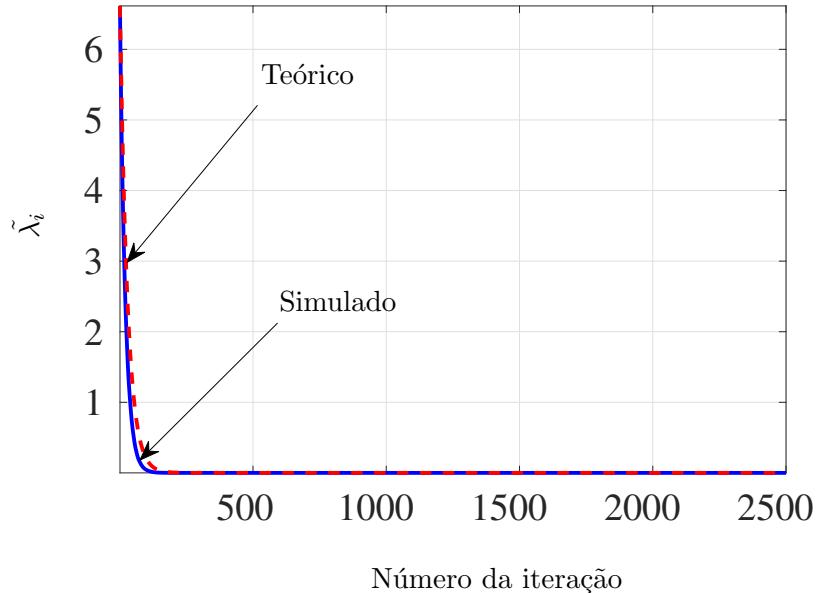


Figura V.39: Evolução do  $\bar{\lambda}_i^i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BNLMS com  $L = 2$  e  $N = 10$

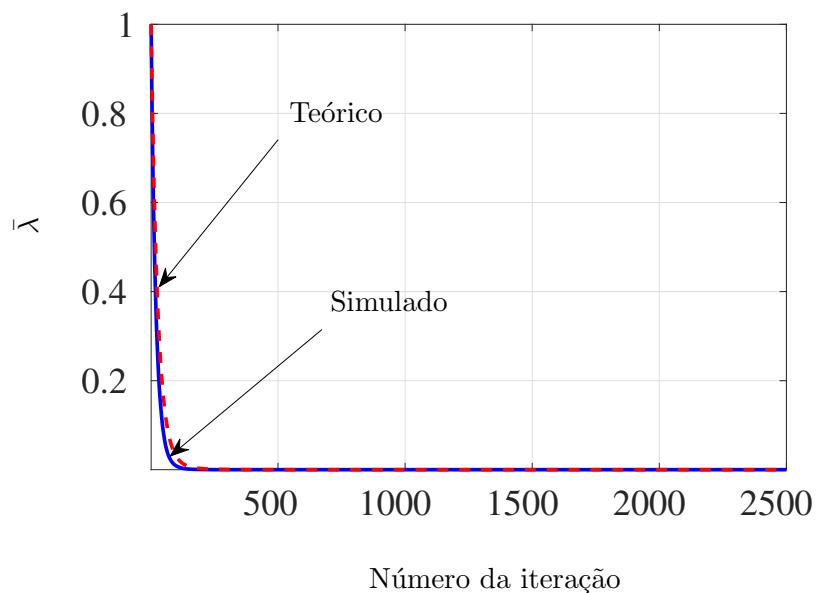


Figura V.40: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 2$  e  $N = 10$

$L = 3$

O terceiro cenário de simulação contempla o caso em que o número de blocos é  $L = 3$ . Mesmo com o aumento do número de blocos, as Figuras V.41, V.42, V.43, V.44 ilustram a coerência entre as previsões teóricas e as obtidas por meio de simulações.

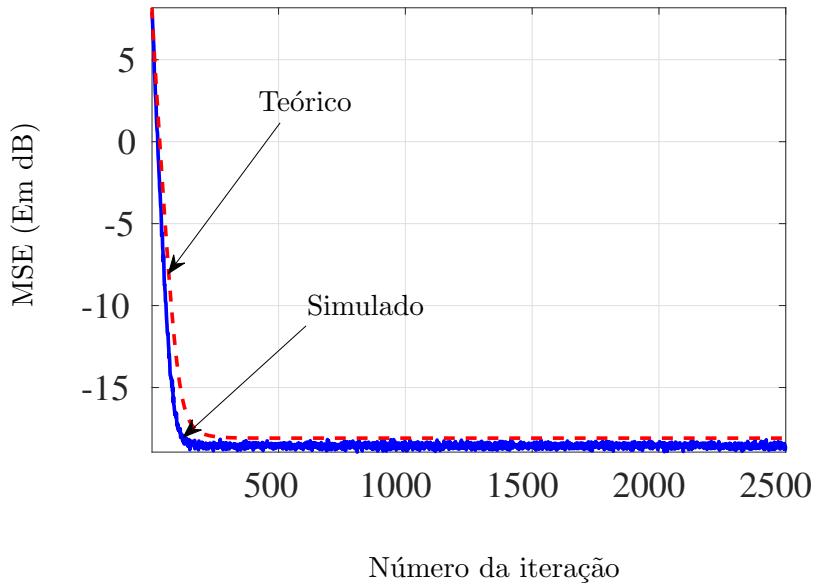


Figura V.41: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 3$  e  $N = 10$

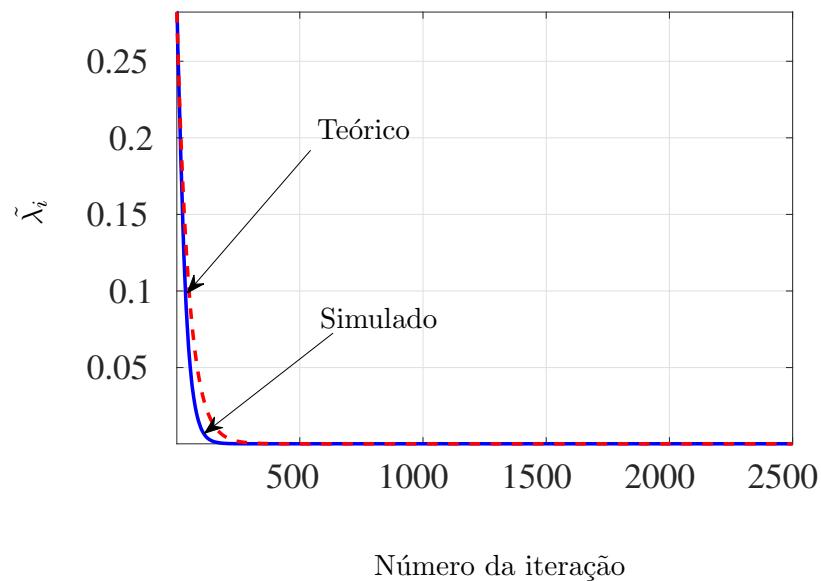


Figura V.42: Evolução do  $\lambda\tilde{i}$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BNLMS com  $L = 3$  e  $N = 10$ .

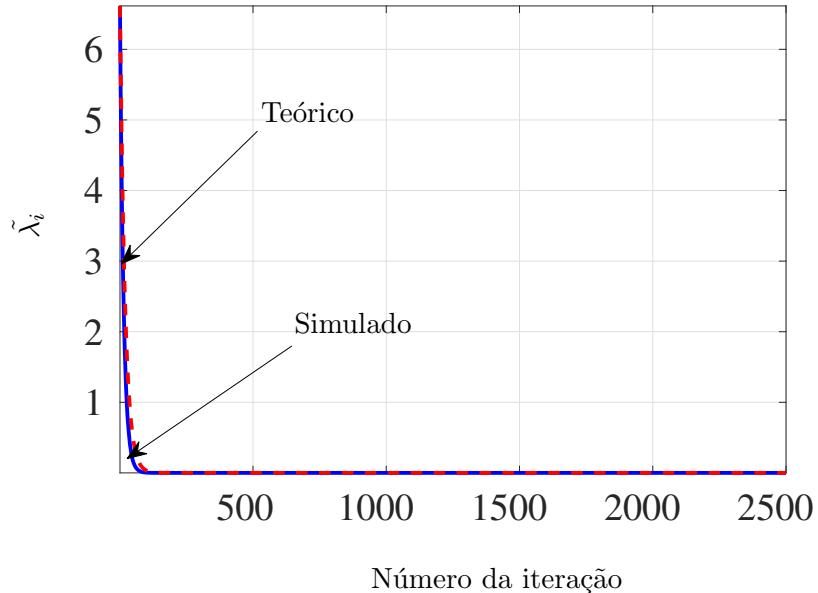


Figura V.43: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BNLMS com  $L = 3$  e  $N = 10$

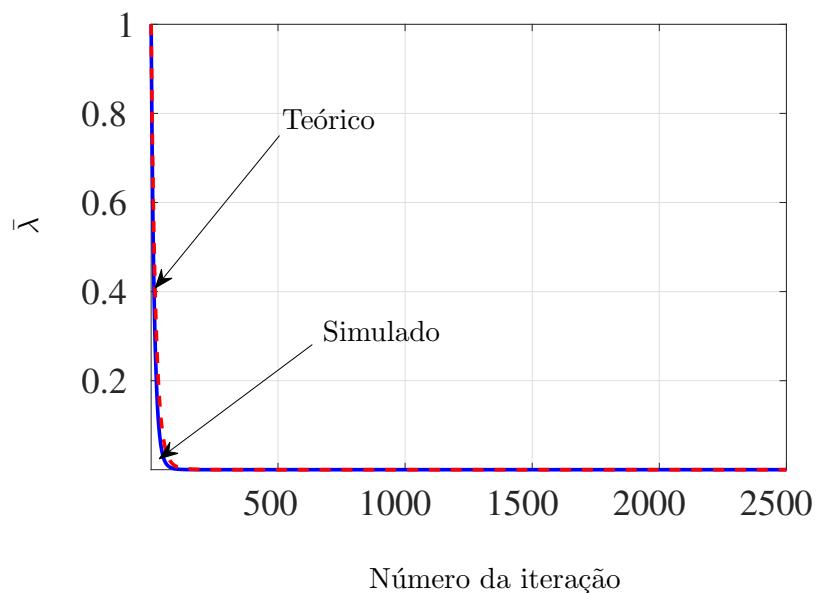


Figura V.44: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 3$  e  $N = 10$

$$L = 4$$

O cenário 4 adota um número de blocos  $L = 4$ . As Figuras V.45, V.46, V.47 e V.48 mostram que nem mesmo com um número alto de blocos a análise proposta perde coerência com os resultados experimentais.

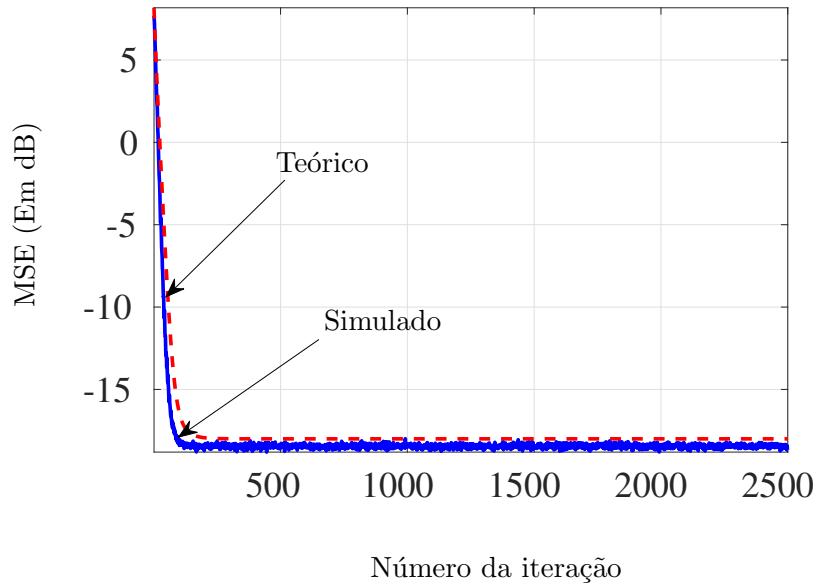


Figura V.45: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 4$  e  $N = 10$

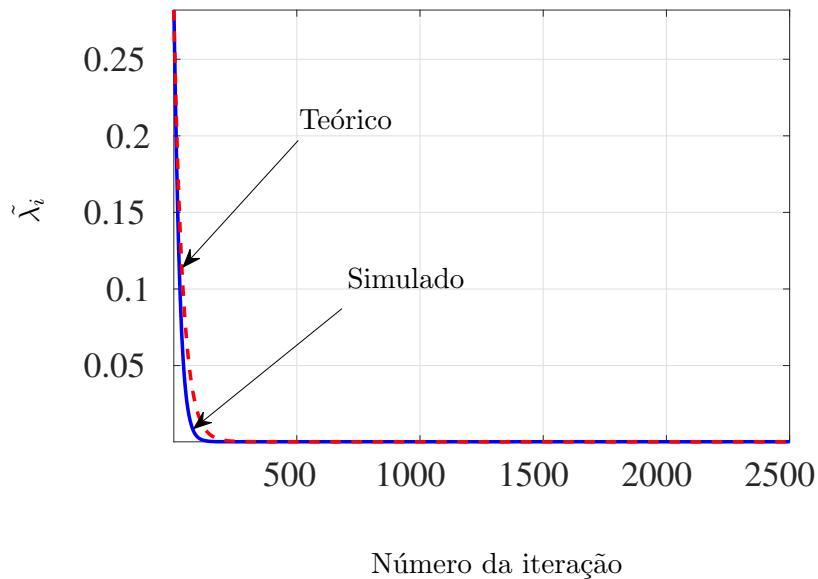


Figura V.46: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BNLMS com  $L = 4$  e  $N = 10$

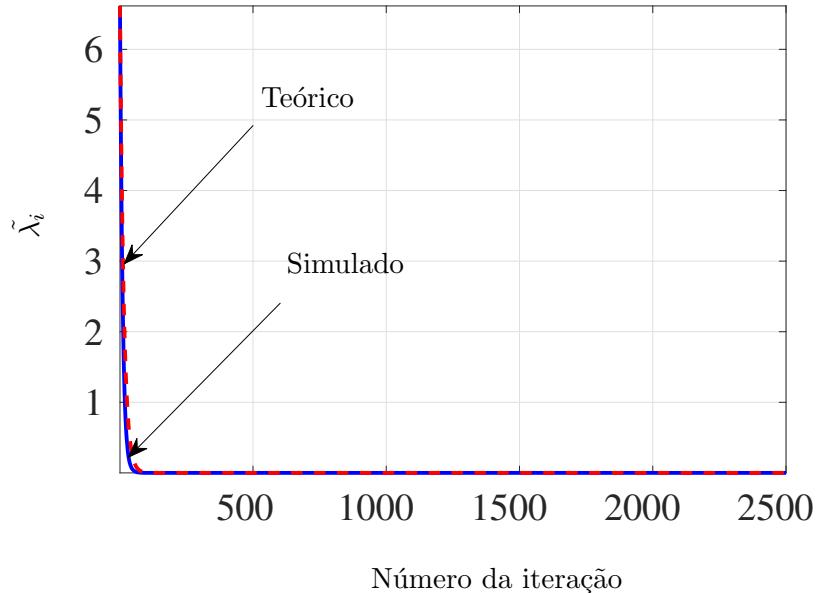


Figura V.47: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BNLMS com  $L = 4$  e  $N = 10$

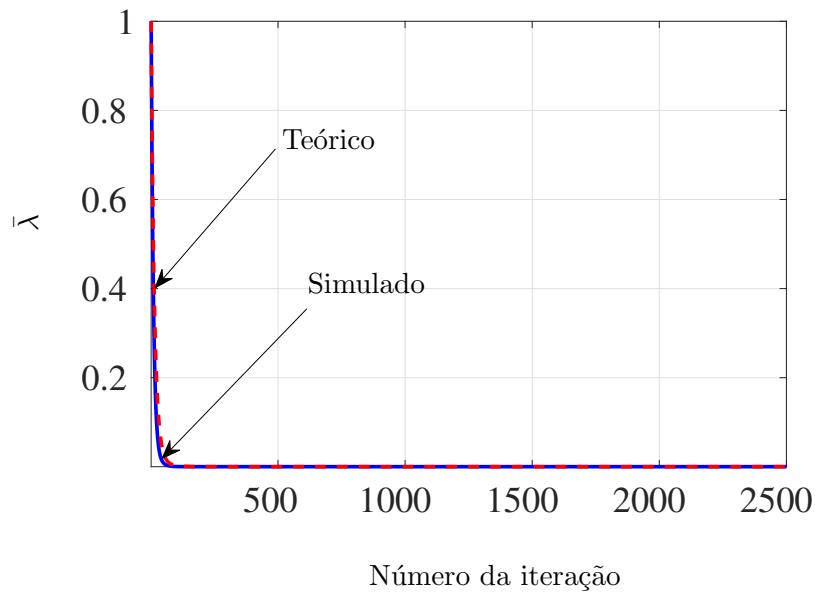


Figura V.48: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 4$  e  $N = 10$

### Comentários

As simulações realizadas no cenário um, com  $N = 10$  para o algoritmo Normalizado demonstraram boa aderência apesar de um pouco ruidoso em regime permanente, principalmente com quantidade de blocos  $L = 1$ . As curvas de  $\tilde{\lambda}_i(k)$  ficaram muito congruentes em todas as quantidades de blocos em especial nas curvas médias de  $\bar{\lambda}_i(k)$ .

**Cenário 2 →  $N = 60$**

Os parâmetros utilizados para a simulação do BLMS foram  $N = 10$ ,  $\beta = 10^{-2}$ ,  $\sigma_\nu^2 = 10^{-2}$ , 1000 ensaios independentes de Monte Carlo e um sinal de entrada obtido pelo colorimento de um sinal branco, gaussiano e de média zero pelo filtro  $B(z) = 1 - 0,8z^{-1} + 0,6z^{-2}$ . Todos os coeficientes da função de transferência ideal são unitários.

$L = 1$

O cenário 1 utiliza  $L = 1$  blocos, caso em que o algoritmo BNLMS se degenera no NLMS. As Figuras V.49, V.50, V.51 e V.52 ilustram a generalidade do modelo proposto, o qual prevê as características de desempenho do NLMS como um caso particular.

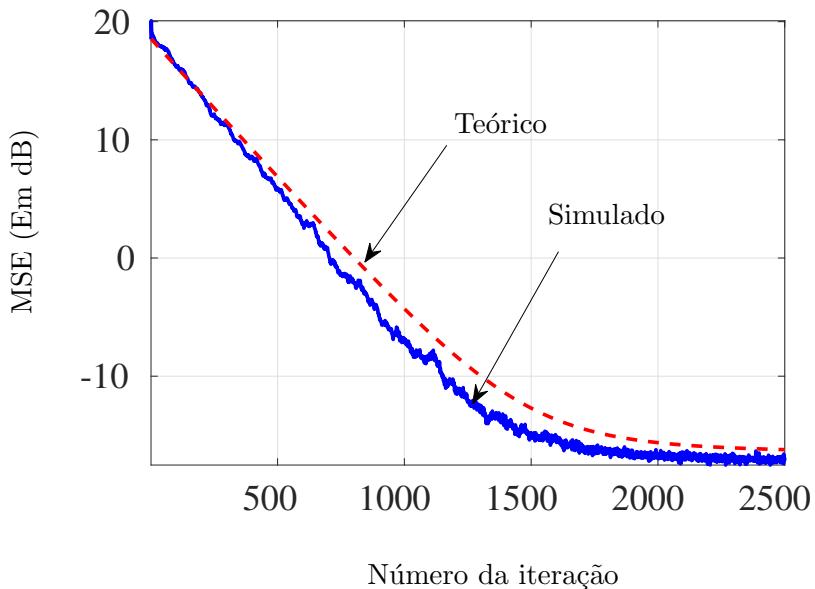


Figura V.49: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 60$

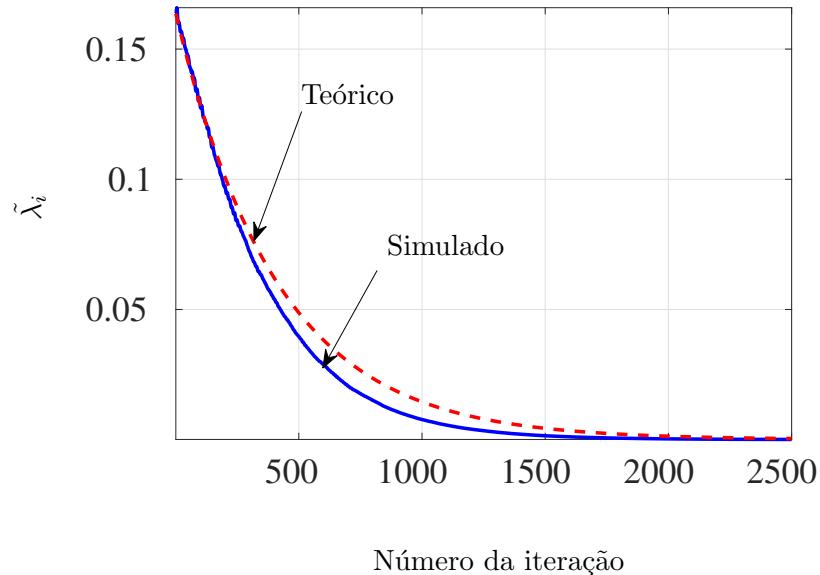


Figura V.50: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 18$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 60$

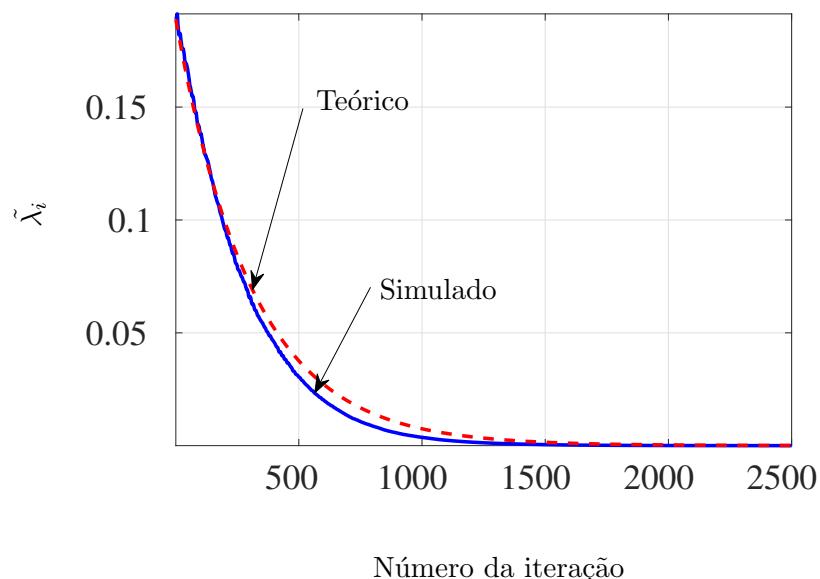


Figura V.51: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 21$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 60$

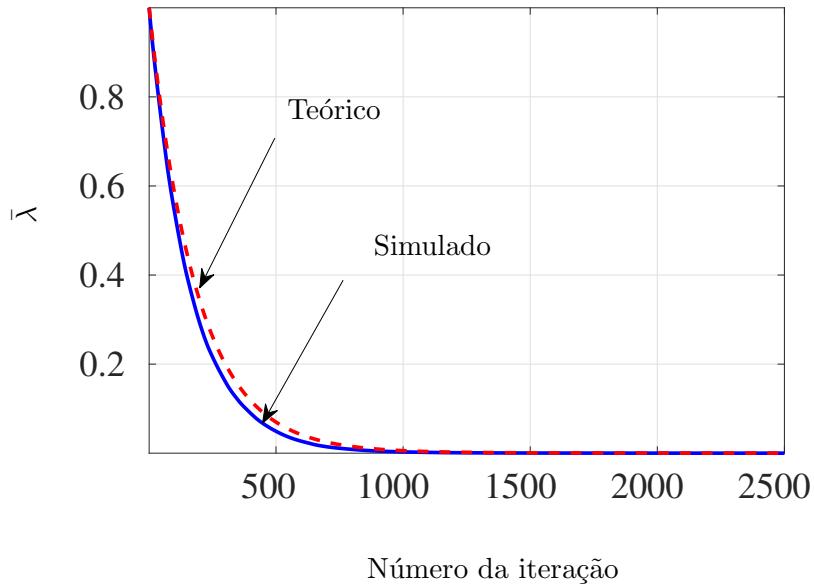


Figura V.52: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 60$

$$L = 2$$

O cenário 2 adota  $L = 2$  blocos. As Figuras V.53, V.54, V.55 e V.56 mostram a boa aderência do modelo proposto aos dados simulados.

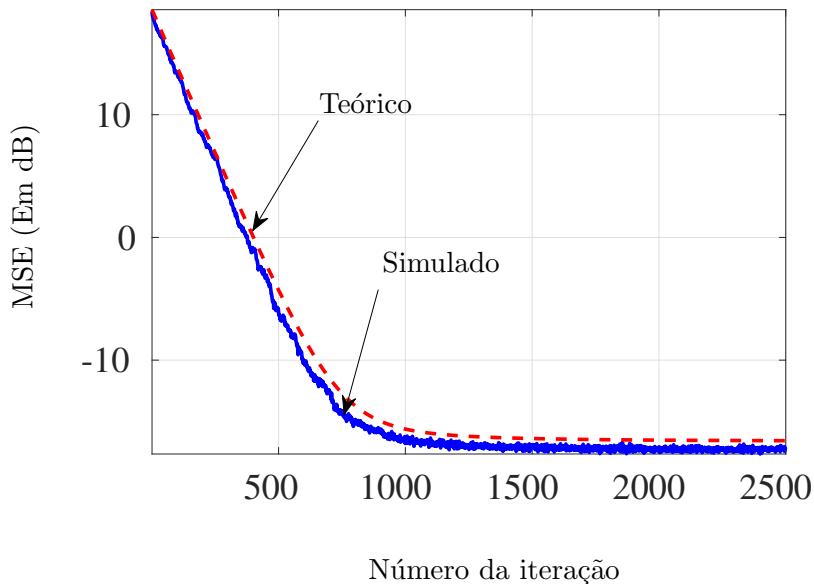


Figura V.53: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 2$  e  $N = 60$

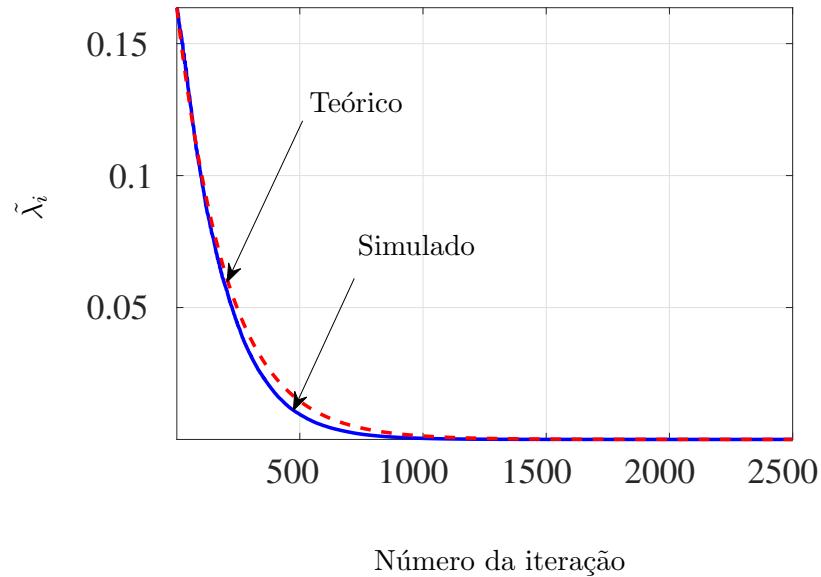


Figura V.54: Evolução do  $\tilde{\lambda}_i^i$  teórico e simulado com  $i = 18$  ao longo das iterações do algoritmo BNLMS com  $L = 2$  e  $N = 60$

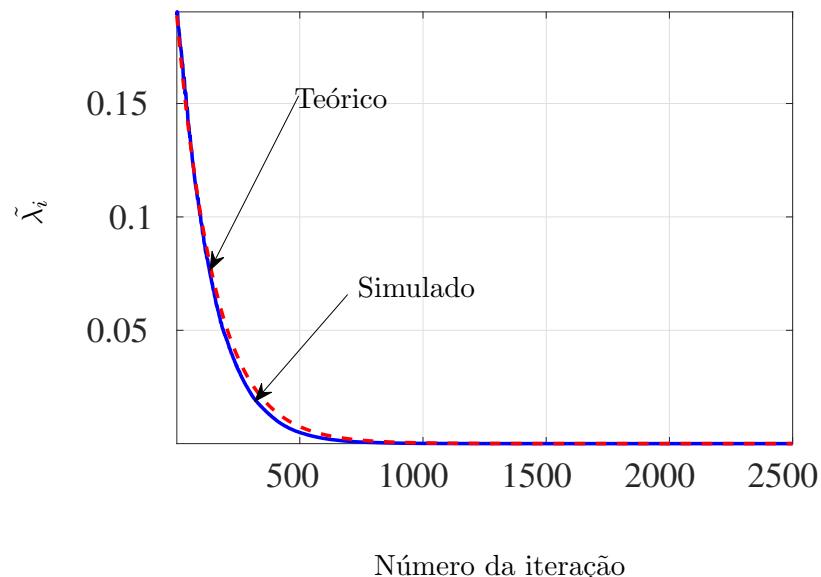


Figura V.55: Evolução do  $\tilde{\lambda}_i^i$  teórico e simulado com  $i = 21$  ao longo das iterações do algoritmo BNLMS com  $L = 2$  e  $N = 60$

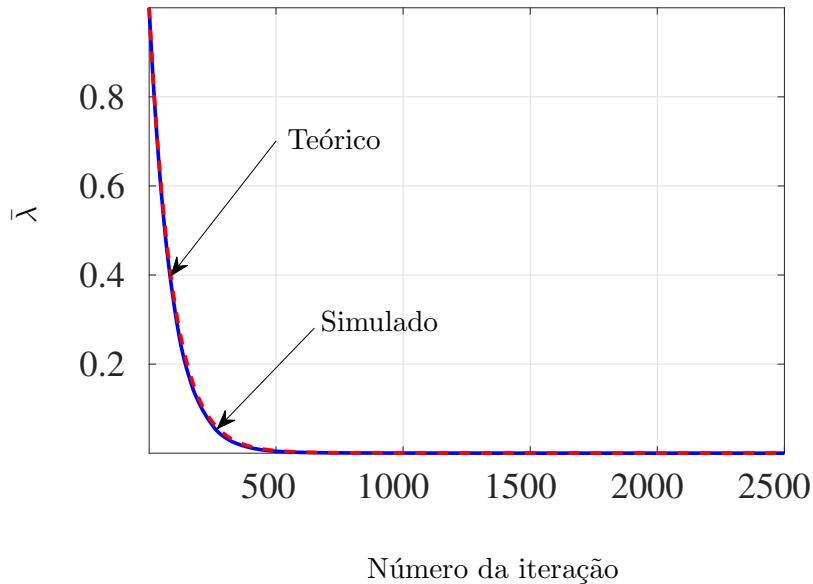


Figura V.56: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 2$  e  $N = 60$

$L = 3$

O terceiro cenário de simulação contempla o caso em que o número de blocos é  $L = 3$ . Mesmo com o aumento do número de blocos, as Figuras V.57, V.58, V.59 e V.60 ilustram a coerência entre as previsões teóricas e as obtidas por meio de simulações.

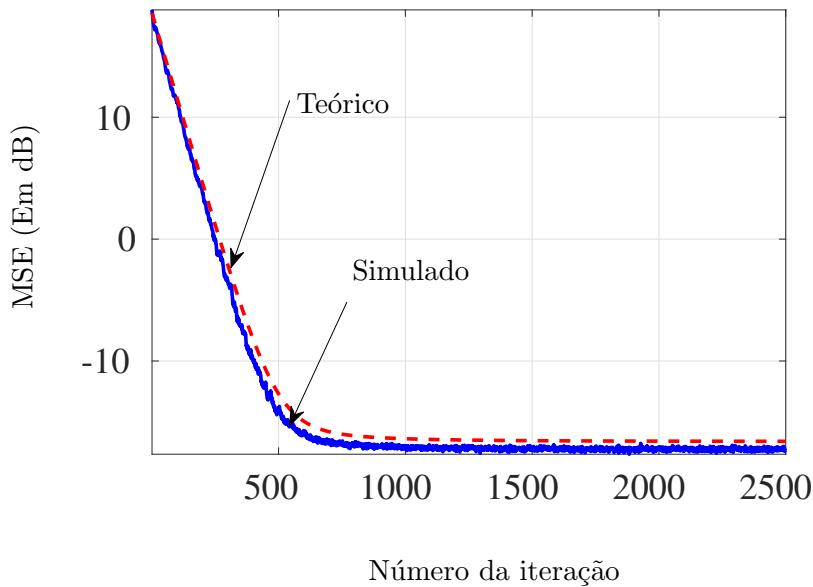


Figura V.57: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 3$  e  $N = 60$

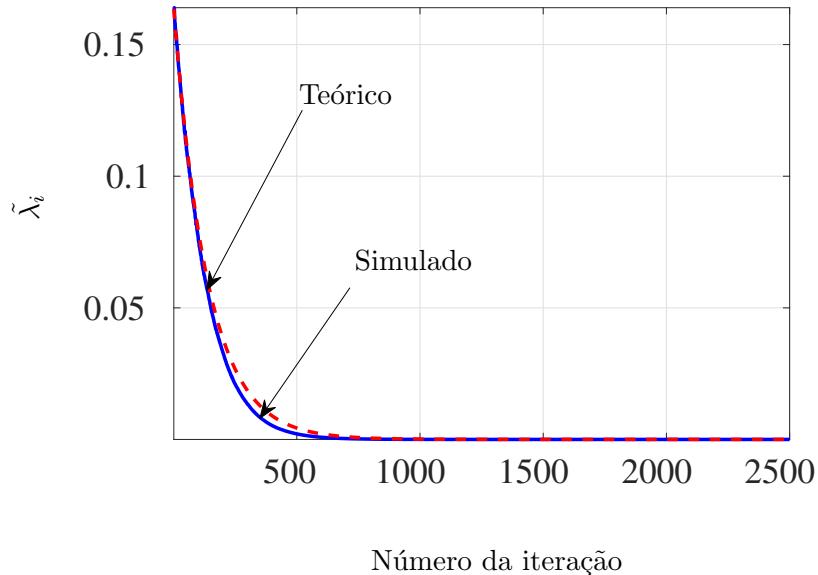


Figura V.58: Evolução do  $\tilde{\lambda}_i^i$  teórico e simulado com  $i = 18$  ao longo das iterações do algoritmo BNLMS com  $L = 3$  e  $N = 60$ .

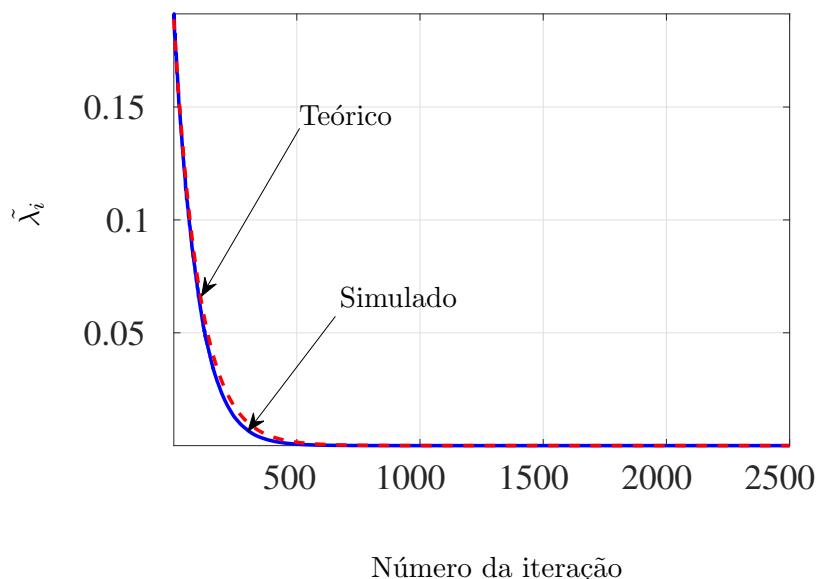


Figura V.59: Evolução do  $\tilde{\lambda}_i^i$  teórico e simulado com  $i = 21$  ao longo das iterações do algoritmo BNLMS com  $L = 3$  e  $N = 60$

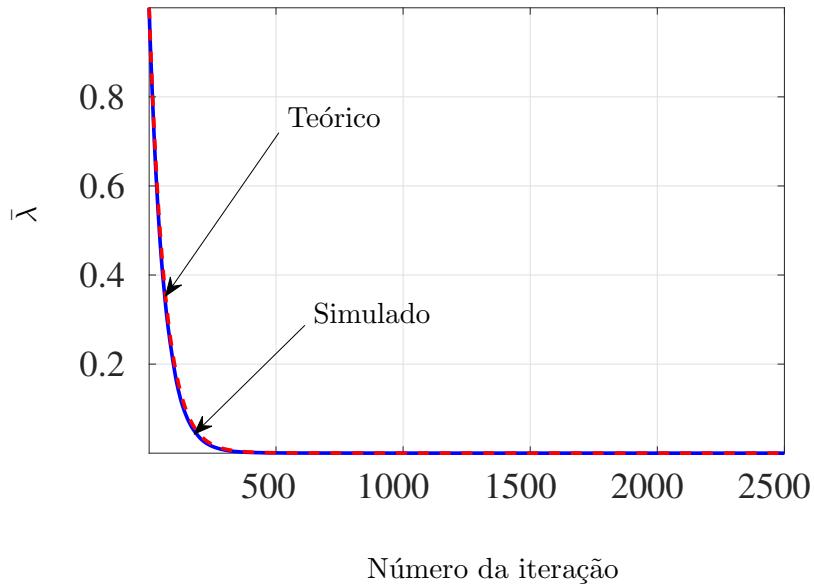


Figura V.60: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 3$  e  $N = 60$

$L = 4$

O cenário 4 adota um número de blocos  $L = 4$ . As Figuras V.61, V.62, V.63 e V.64 mostram que nem mesmo com um número alto de blocos a análise proposta perde coerência com os resultados experimentais.

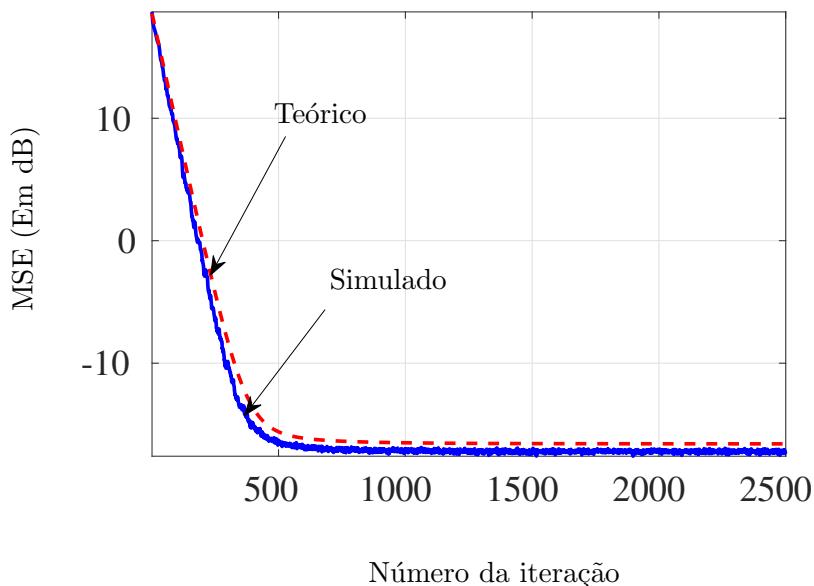


Figura V.61: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 4$  e  $N = 60$

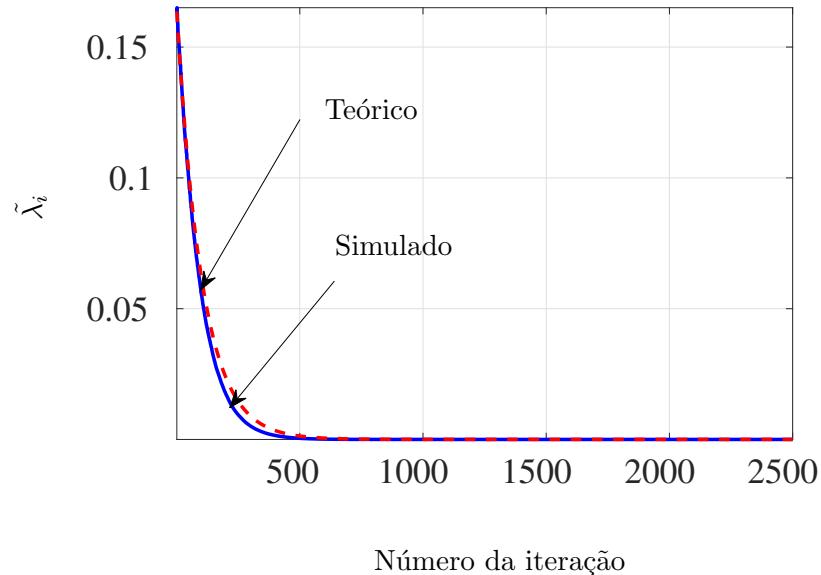


Figura V.62: Evolução do  $\tilde{\lambda}_i^i$  teórico e simulado com  $i = 18$  ao longo das iterações do algoritmo BNLMS com  $L = 4$  e  $N = 60$

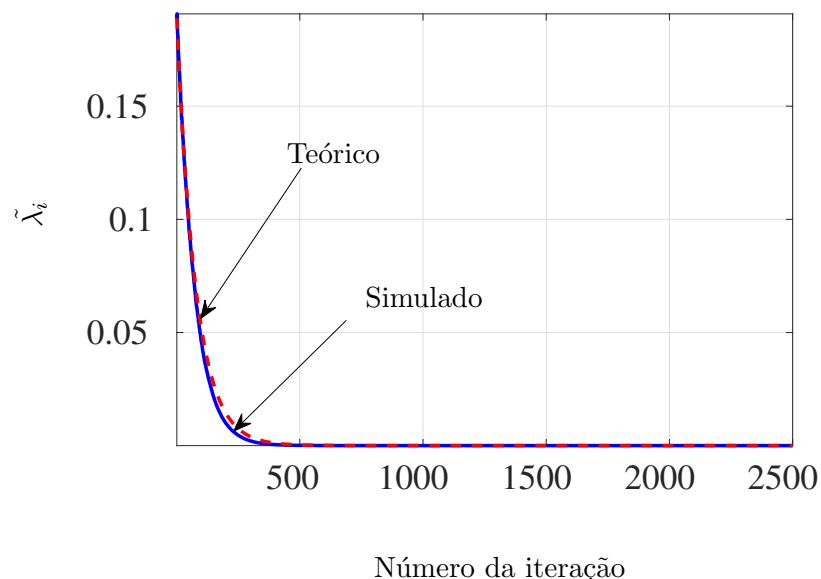


Figura V.63: Evolução do  $\tilde{\lambda}_i^i$  teórico e simulado com  $i = 21$  ao longo das iterações do algoritmo BNLMS com  $L = 4$  e  $N = 60$

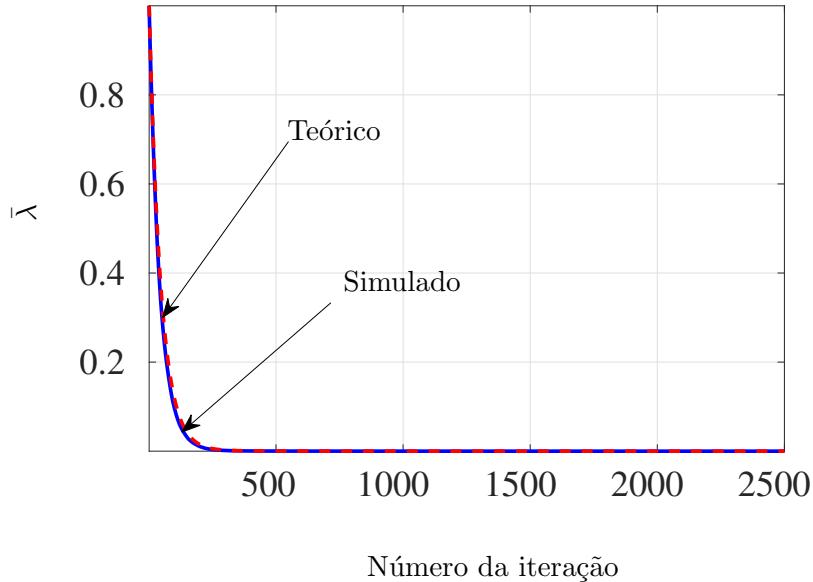


Figura V.64: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 4$  e  $N = 60$

### Comentários

Apesar de as curvas de simulação propostas demonstrarem-se um pouco ruidosas, principalmente nas de MSE, a simulação se provou satisfatória, reforçando o objetivo deste cenário com  $N$  maior, que é justamente mostrar que o modelo se mantém aderente mesmo aumentando o tamanho do filtro e utilizando outra função de transferência. O MSE em excesso em regime permanente é mais perceptível neste cenário. O mesmo tem relacionamento direto com o tamanho do filtro e com a quantidade de blocos conforme demonstrado em (IV.56).

### V.3 Simulação 2 - Divergência em função de $\beta$

Uma das mais importantes propriedades de um modelo estocástico reside na predição de limitantes superiores para o fator de aprendizagem de modo a garantir estabilidade do algoritmo. Cabe ressaltar que a divergência é um fenômeno extremamente indesejável, que cumpre evitar ao máximo. Nesta seção são apresentadas curvas simuladas com a probabilidade de divergência do algoritmo em função do  $\beta$  utilizado. Nestas curvas, foi assinalado o limitante superior teórico de  $\beta$  capaz de evitar a divergência, utilizando para tanto o valor calculado através das equações (IV.26) (BLMS) e (IV.49) (BNLMS). A partir desta mesma simulação foram computados em uma tabela os valores de beta mais próximos dos pontos escolhidos de probabilidade de divergência em 0%, 50% e 100%, em seguida foram traçadas novas curvas em cada cenário representando como estes três pontos de probabilidade de divergência escolhidos se comportam ao longo da variação da quantidade de blocos no algoritmo. Foram utilizados os mesmos parâmetros de execução por algoritmo das seções anteriores, variando apenas o tamanho do bloco ( $L$ ). Todos os coeficientes da função de

transferência ideal são unitários.

### V.3.1 BLMS

Após diversos testes foi escolhida uma faixa de  $\beta$  que melhor representasse a probabilidade de divergência em torno do valor teórico calculado, esta escolha foi baseada em múltiplas execuções do algoritmo até encontrar a faixa de  $\beta$  que melhor exibisse a curva, no caso desta simulação foram utilizados valores de  $\beta$  entre  $1.0 \times 10^{-2}$  e  $9.0 \times 10^{-2}$  com cerca de 80 pontos uniformemente espalhados para o cenário  $N = 10$  e valores de  $\beta$  entre  $2.0 \times 10^{-3}$  e  $9.0 \times 10^{-2}$  com cerca de 80 pontos uniformemente espalhados para o cenário  $N = 60$ .

**Cenário 1 →  $N = 10$**

$L = 1$

Neste cenário de simulação foi utilizado o parâmetro de quantidade de blocos  $L = 1$ . Observa-se na Figura VII.108 que o modelo estocástico proposto consegue prever com razoável precisão o “joelho” da curva de probabilidade de divergência.

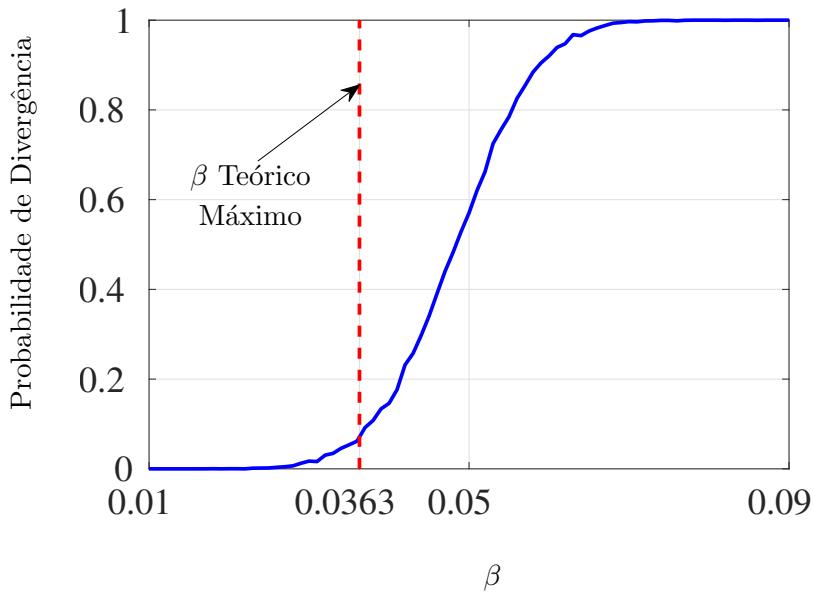


Figura V.65: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 1$  e  $N = 10$ .

$L = 2$

A Figura V.66 compara a curva empírica de probabilidade de divergência com o limitante superior de  $\beta$  inferido através do modelo proposto, para  $L = 2$ .

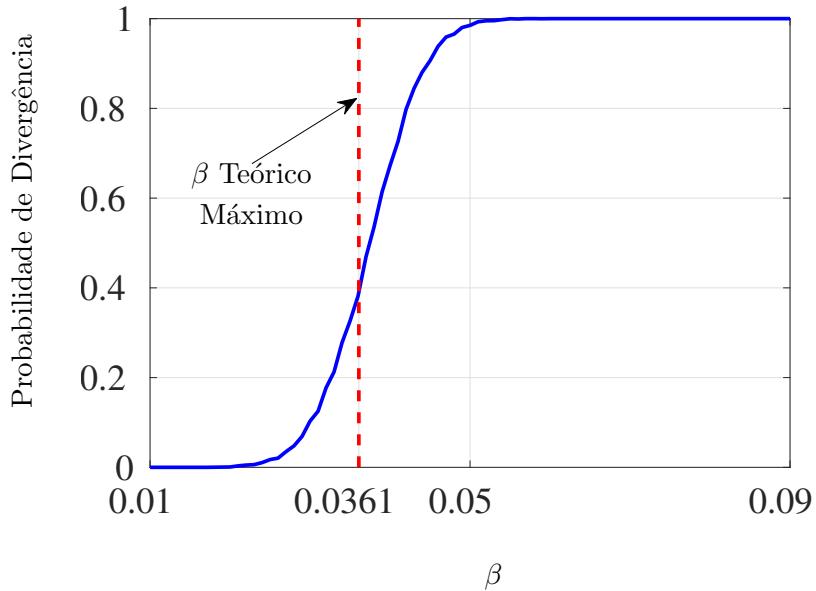


Figura V.66: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 2$  e  $N = 10$

$L = 3$

No caso do algoritmo BLMS com  $L = 3$ , a Figura V.67 mostra que o modelo desenvolvido consegue aproximar razoavelmente a partir de que ponto o algoritmo é eminentemente instável.

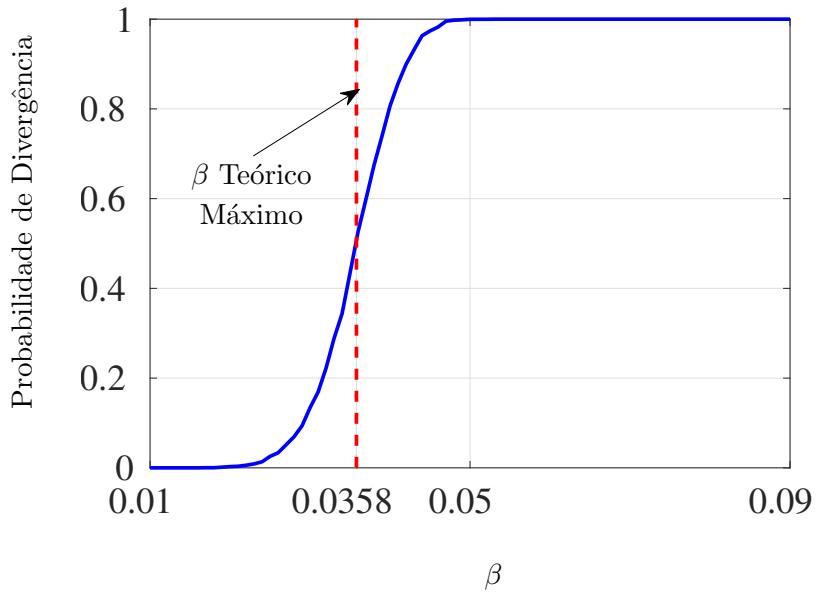


Figura V.67: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 3$  e  $N = 10$

$L = 4$

A Figura V.68 mostra que o modelo proposto é capaz de predizer razoavelmente a região de estabilidade do algoritmo BLMS quando  $L = 4$ .

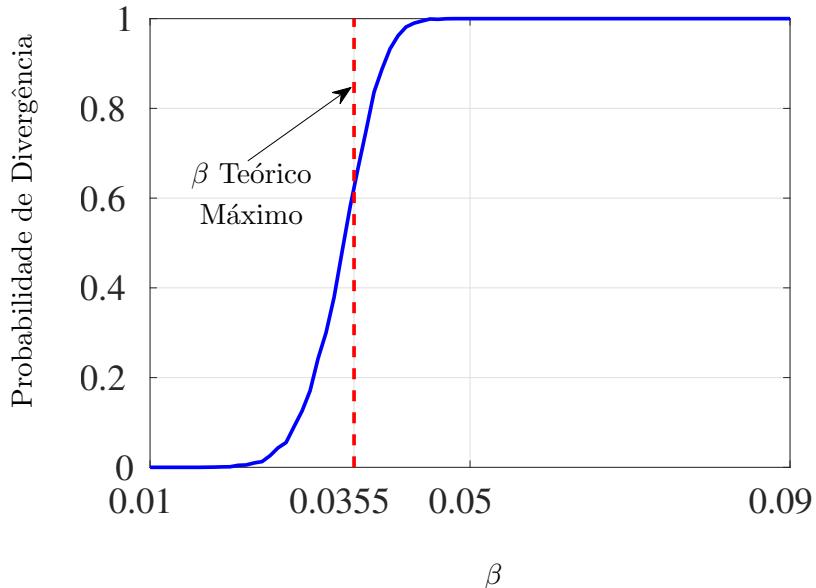


Figura V.68: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 4$  e  $N = 10$

$L$	Valores de $\beta$ para BLMS $N = 10$			
	$\beta$ máximo que evita a divergência	$\beta$ com probabilidade simulada de divergência 0%	$\beta$ com probabilidade simulada de divergência 50%	$\beta$ com probabilidade simulada de divergência 100%
1	0,036342	0,017	0,0485	0,0775
2	0,036058	0,017	0,0375	0,0565
3	0,035778	0,016	0,0355	0,0525
4	0,035503	0,016	0,0345	0,0475
5	0,035232	0,015	0,0325	0,0435
6	0,034965	0,015	0,0305	0,0405
7	0,034702	0,014	0,0295	0,0375
8	0,034443	0,016	0,0275	0,0355
9	0,034188	0,013	0,0265	0,0335
10	0,033937	0,011	0,0255	0,0315

Tabela V.1: Tabela com valores de  $\beta$  simulados da probabilidade de divergência nas faixas 0%, 50% e 100% para o algoritmo BLMS e  $N=10$ .

A partir dos dados coletados na execução das simulações representados na Tabela V.1 foram traçadas curvas de  $\beta$  com diferentes probabilidades de divergência conforme Figura V.69 abaixo:

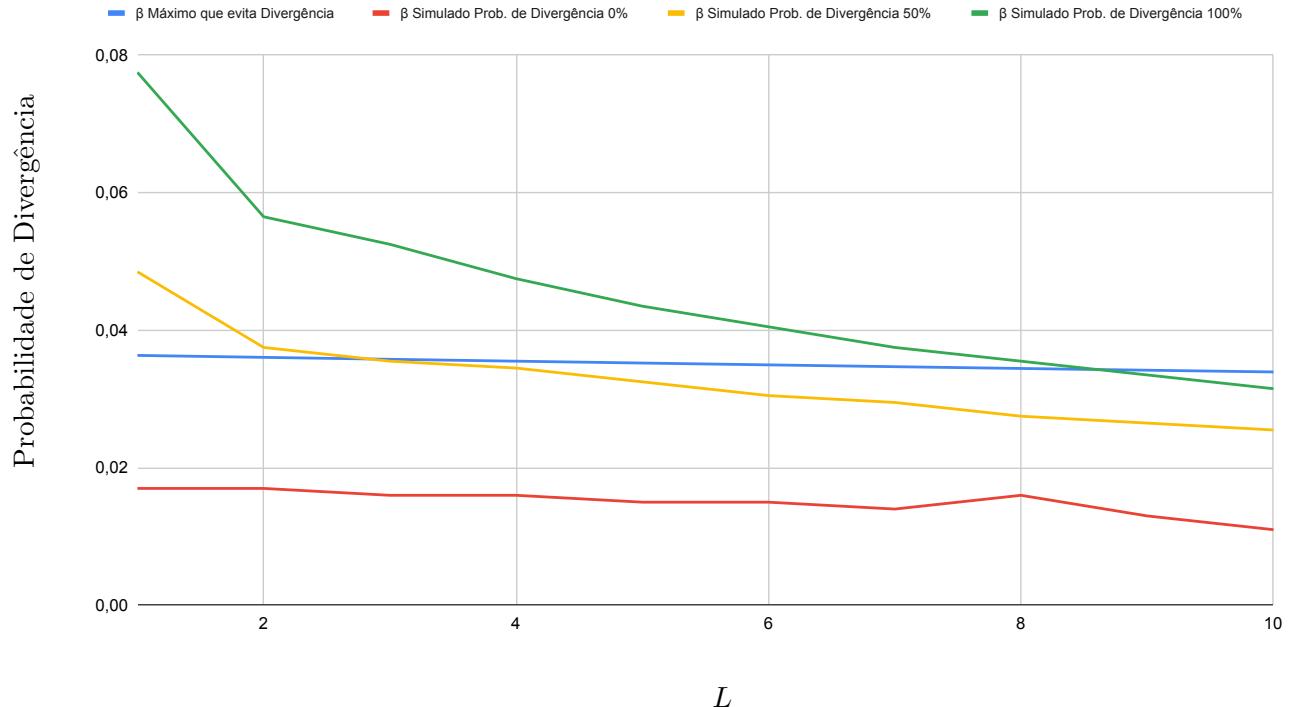


Figura V.69: Curva de probabilidade de divergência simulada versus quantidade de blocos  $L$  do algoritmo BLMS com  $N = 10$

## Comentários

Nesta simulação, o algoritmo se desenvolveu muito coerente com o objetivo proposto, pois em especial a curva (Figura V.69) que representa o  $\beta$  com probabilidade simulada de divergência 0% se manteve ao longo da variação da quantidade de blocos  $L$  sempre abaixo do valor máximo calculado que evita a divergência (IV.26). Isto traz grande segurança aos projetistas pois demonstra que o modelo proposto apresenta uma margem razoável de implementação sem eventos indesejáveis.

## Cenário 2 → $N = 60$

$$L = 1$$

Neste cenário de simulação foi utilizado o parâmetro de quantidade de blocos  $L = 1$ . Observa-se na Figura V.70 que o modelo estocástico proposto consegue prever com razoável precisão o “joelho” da curva de probabilidade de divergência.

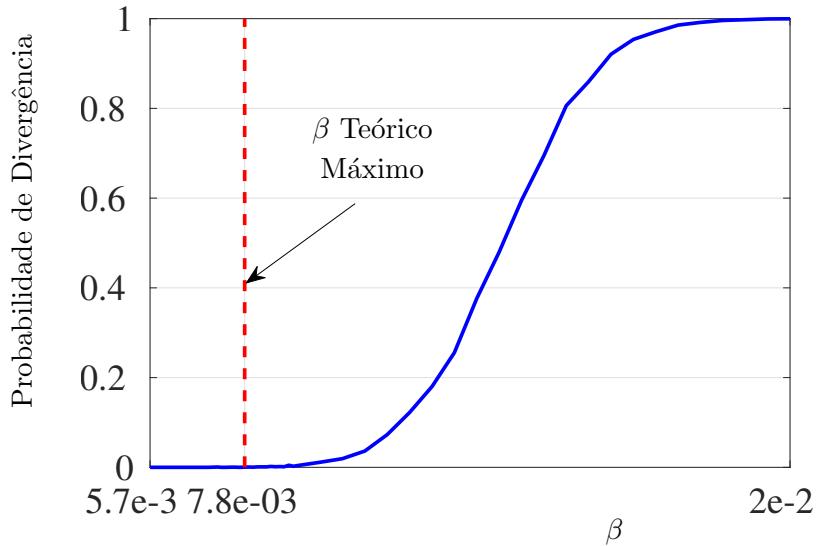


Figura V.70: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 1$  e  $N = 60$ .

$L = 2$

A Figura V.71 compara a curva empírica de probabilidade de divergência com o limitante superior de  $\beta$  inferido através do modelo proposto, para  $L = 2$ .

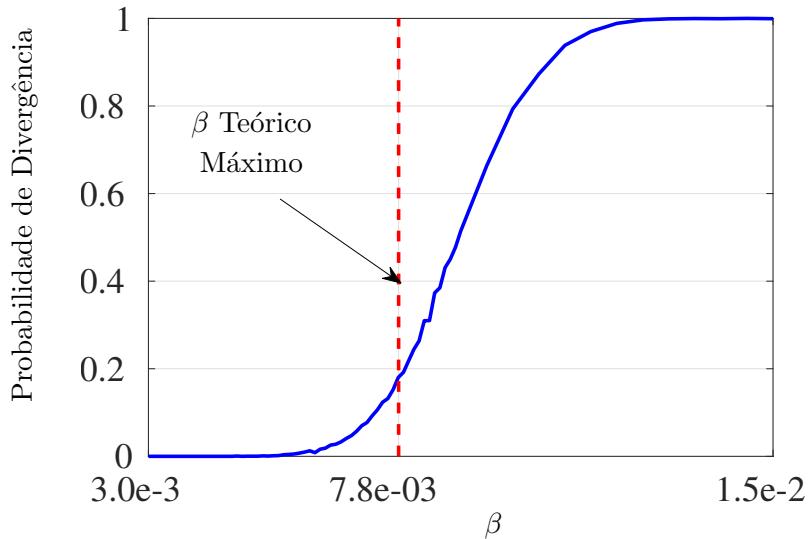


Figura V.71: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 2$  e  $N = 60$ .

$L = 3$

No caso do algoritmo BLMS com  $L = 3$ , a Figura V.72 mostra que o modelo desenvolvido consegue aproximar razoavelmente a partir de que ponto o algoritmo é eminentemente instável.

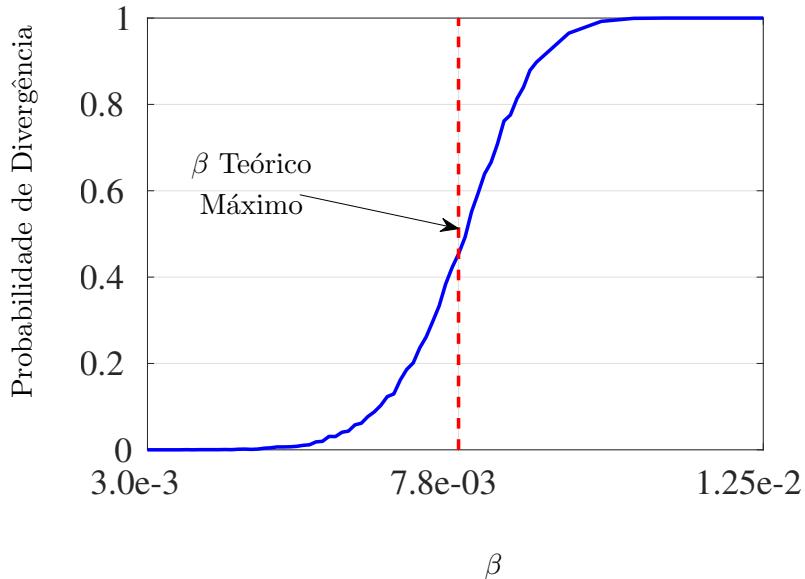


Figura V.72: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 3$  e  $N = 60$

$L = 4$

A Figura V.73 mostra que o modelo proposto é capaz de predizer razoavelmente a região de estabilidade do algoritmo BLMS quando  $L = 4$ .

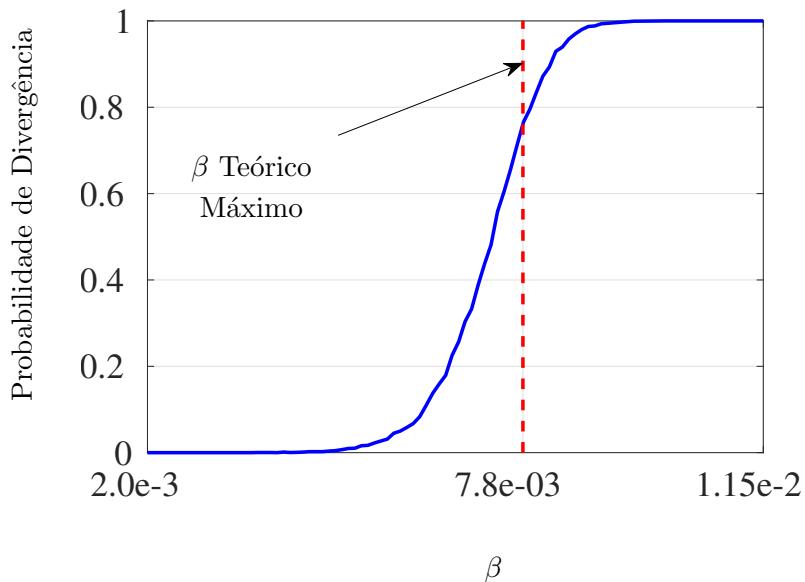


Figura V.73: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 4$  e  $N = 60$

$L$	$\beta$ máximo que evita a divergência	Valores de $\beta$ para BLMS $N = 10$		
		$\beta$ com probabilidade simulada de divergência 0%	$\beta$ com probabilidade simulada de divergência 50%	$\beta$ com probabilidade simulada de divergência 100%
1	0,007814	0,0071	0,01375	0,01925
2	0,007807	0,0046	0,00895	0,01325
3	0,007799	0,0037	0,00795	0,01075
4	0,007792	0,0037	0,00735	0,00975
5	0,007784	0,003	0,00675	0,00865
6	0,007777	0,0032	0,00645	0,00795
7	0,007770	0,003	0,00615	0,00755
8	0,007762	0,0031	0,00585	0,00725
9	0,007755	0,0027	0,00565	0,00695
10	0,007748	0,0032	0,00555	0,00685

Tabela V.2: Tabela com valores de  $\beta$  simulados da probabilidade de divergência nas faixas 0%, 50% e 100% para o algoritmo BLMS e  $N=60$ .

A partir dos dados coletados na execução das simulações representados na Tabela V.2 foram traçadas curvas de  $\beta$  com diferentes probabilidades de divergência conforme Figura V.74 abaixo:

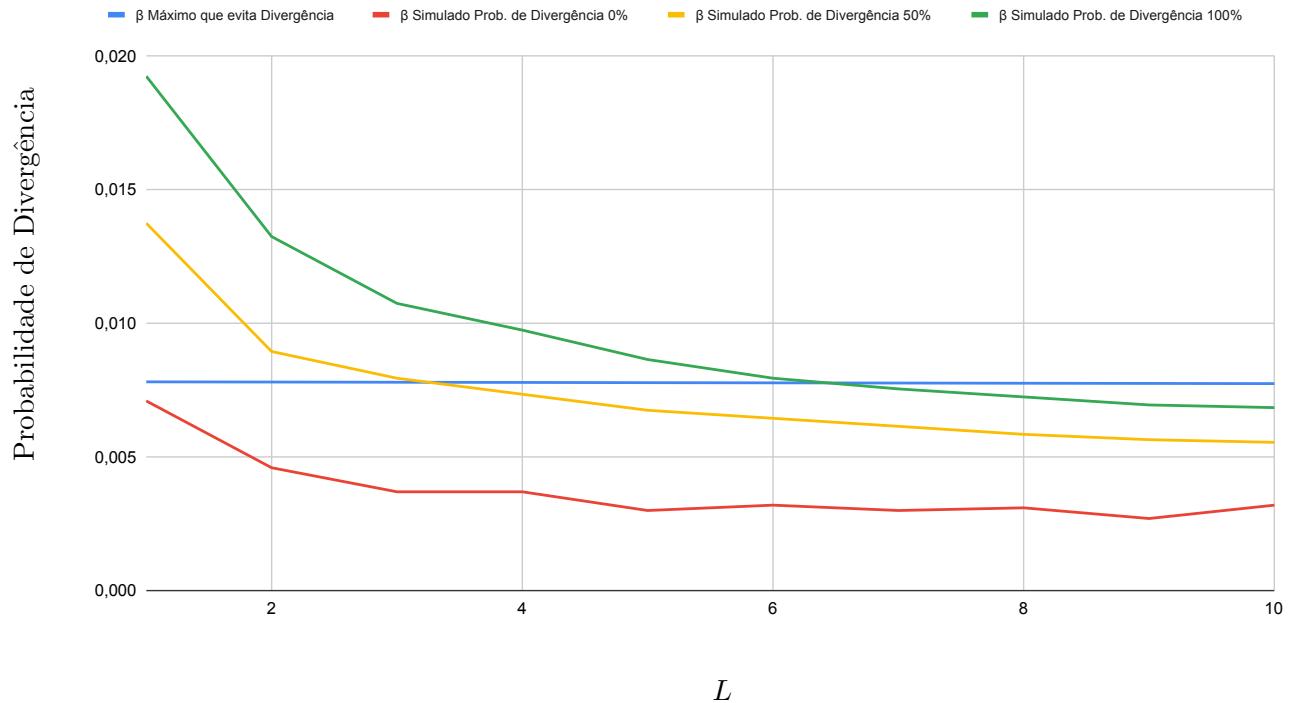


Figura V.74: Curva de probabilidade de divergência simulada versus quantidade de blocos  $L$  do algoritmo BLMS com  $N = 60$

## Comentários

A simulação com  $N = 60$  se mostrou tão aderente quanto quanto o cenário anterior com a curva de  $\beta$  com probabilidade simulada de divergência 0% (Figura V.74) abaixo da margem de segurança delimitada pelo  $\beta$  máximo que evita a divergência.

### V.3.2 BNLMS

Após diversos testes foi escolhida uma faixa de  $\beta$  que melhor representasse a probabilidade de divergência em torno do valor teórico calculado, esta escolha foi baseada em múltiplas execuções do algoritmo ate encontrar a faixa de  $\beta$  que melhor exibisse a curva, no caso desta simulação foram utilizados valores de  $\beta$  entre 0.1 e 2.5 com cerca de 80 pontos uniformemente espalhados.

**Cenário 1 →  $N = 10$**

$L = 1$

Neste cenário de simulação, o algoritmo BNLMS foi analisado, com  $L = 2$  blocos. A Figura V.75 mostra que o limitante superior de  $\beta$  predito é bem preciso.

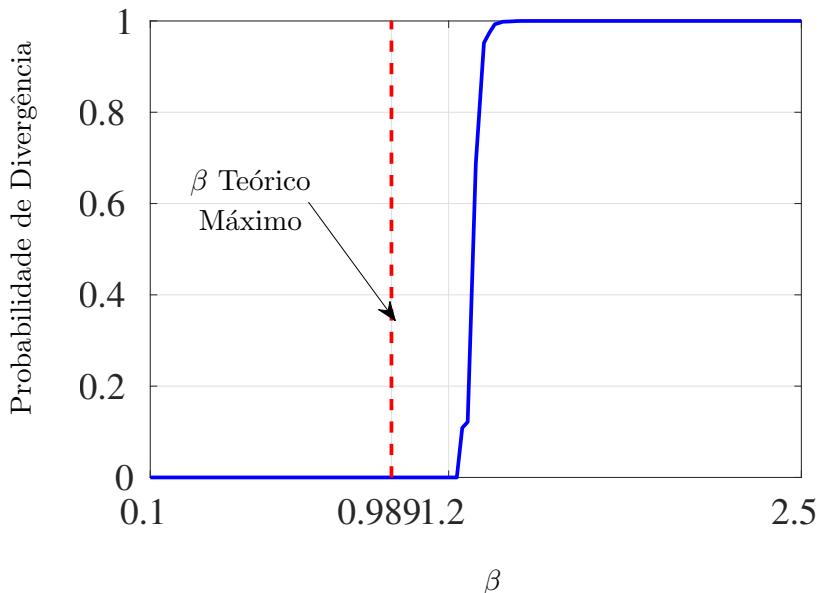


Figura V.75: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 2$  e  $N = 10$

$L = 3$

Para o caso em que o algoritmo BNLMS apresenta  $L = 3$  blocos, a Figura V.76 mostra uma razoável qualidade do intervalo teórico de convergência do algoritmo.

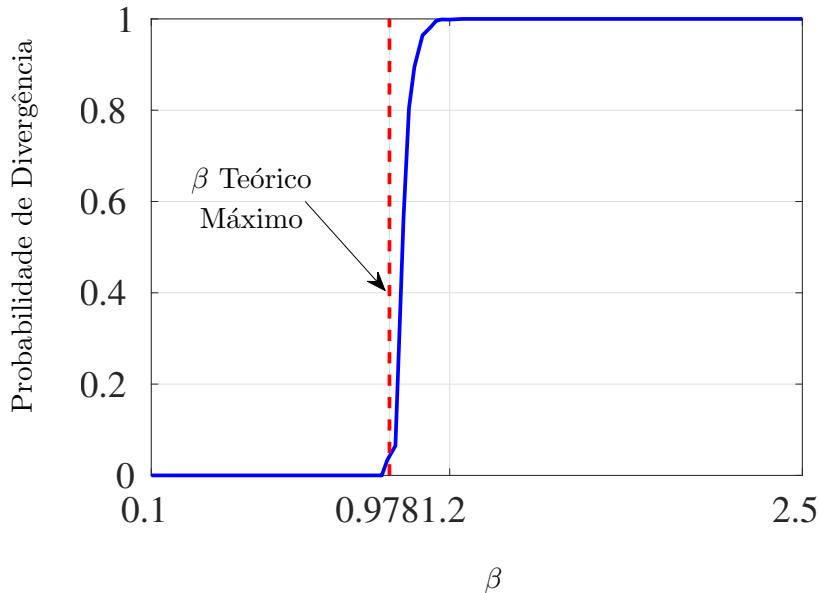


Figura V.76: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 3$  e  $N = 10$

$L = 4$

A Figura V.77 apresenta a probabilidade de divergência do BNLMS em termos de  $\beta$  com uma quantidade de blocos  $L = 4$ . Novamente, a análise teórica consegue delimitar com razoabilidade a faixa de  $\beta$  que garante estabilidade.

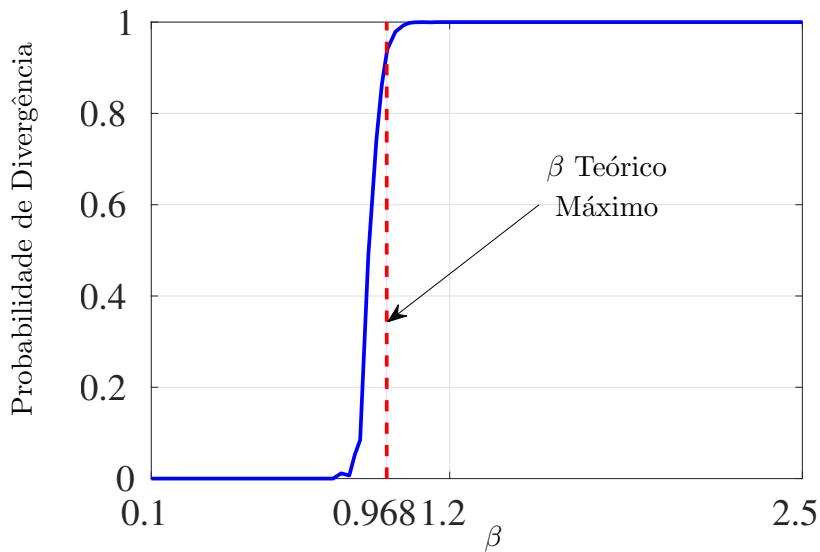


Figura V.77: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 4$  e  $N = 10$

$L$	$\beta$ máximo que evita a divergência	Valores de $\beta$ para BLMS $N = 10$		
		$\beta$ com probabilidade simulada de divergência 0%	$\beta$ com probabilidade simulada de divergência 50%	$\beta$ com probabilidade simulada de divergência 100%
1	1,000000	2	2,015	2,04
2	0,989281	1,23	1,285	1,46
3	0,978790	0,95	1,015	1,24
4	0,968518	0,77	0,915	1,085
5	0,958460	0,73	0,815	1,015
6	0,948609	0,67	0,785	0,94
7	0,938958	0,63	0,715	0,885
8	0,929502	0,6	0,685	0,815
9	0,920234	0,55	0,64	0,815
10	0,911149	0,53	0,585	0,74

Tabela V.3: Tabela com valores de  $\beta$  simulados da probabilidade de divergência nas faixas 0%, 50% e 100% para o algoritmo BNLMs e  $N=10$ .

A partir dos dados coletados na execução das simulações representados na Tabela V.3 foram traçadas curvas de  $\beta$  com diferentes probabilidades de divergência conforme Figura V.78 abaixo:

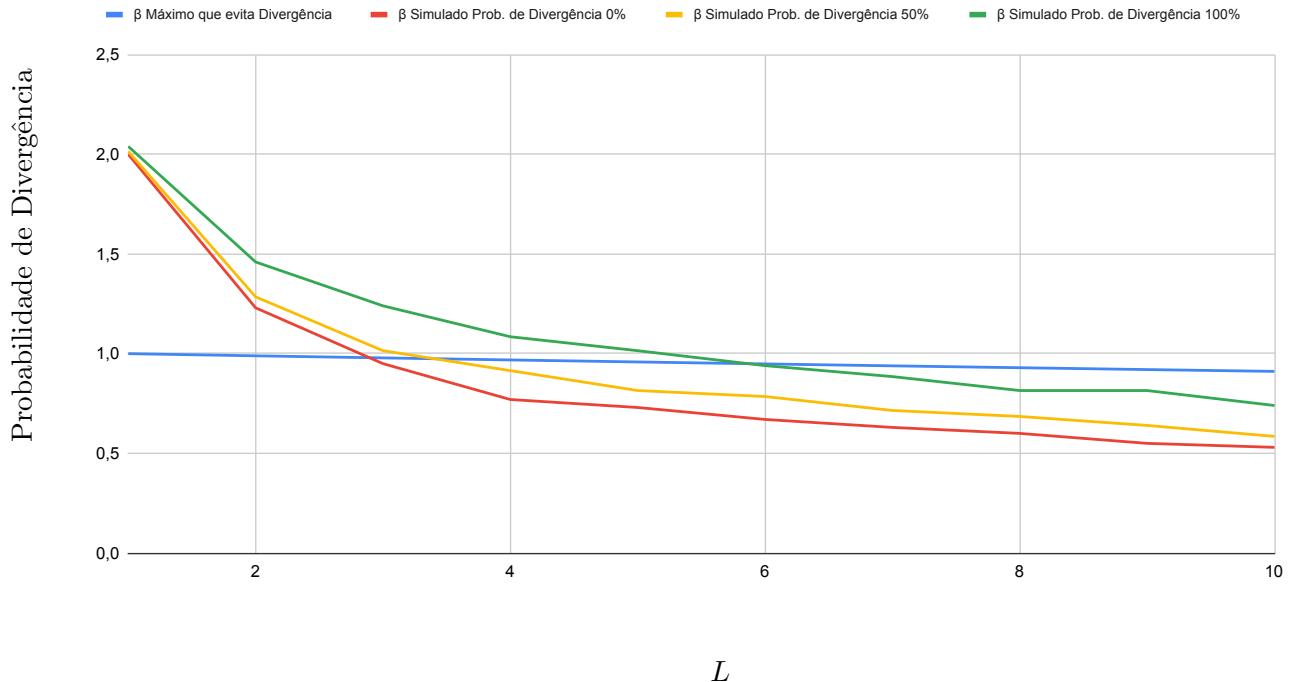


Figura V.78: Curva de probabilidade de divergência simulada versus quantidade de blocos  $L$  do algoritmo BNLMs com  $N = 10$

## Comentários

No cenário de simulação de divergência para o algoritmo BNLMS as curvas (Figura V.78) por quantidade de blocos  $L$  se mostraram bem menos suaves do que no algoritmo não Normalizado. Muito provavelmente devido as hipóteses utilizadas para modelagem os cenários com menos de quatro blocos apresentaram probabilidade de divergência diferente da esperada. Neste caso em particular a simulação se comportou como previsto, visto que a utilização do algoritmo Normalizado se dá sempre com um  $\beta$  abaixo de 1 por motivos de segurança.

**Cenário 2 → N = 60**

$$L = 1$$

Neste cenário de simulação, o algoritmo BNLMS foi analisado, com  $L = 2$  blocos. A Figura V.79 mostra que o limitante superior de  $\beta$  predito é bem preciso.

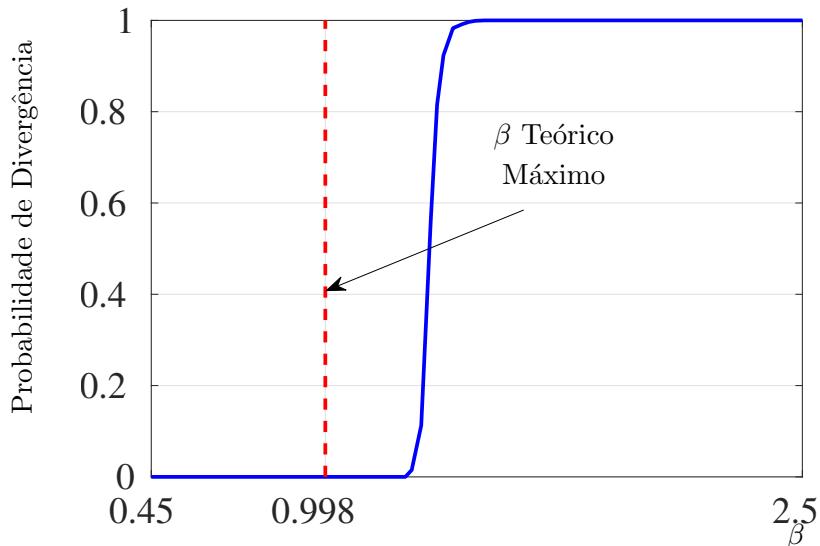


Figura V.79: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 2$  e  $N = 60$

$$L = 3$$

Para o caso em que o algoritmo BNLMS apresenta  $L = 3$  blocos, a Figura V.80 mostra uma razoável qualidade do intervalo teórico de convergência do algoritmo.

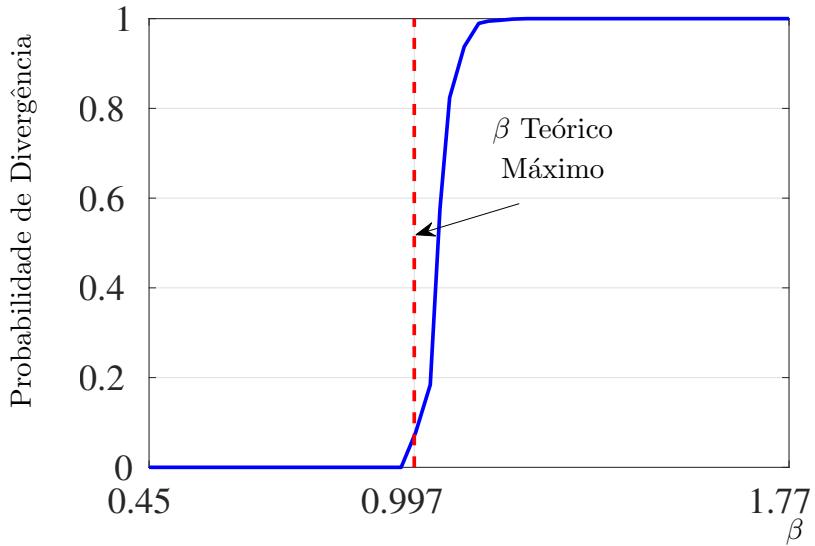


Figura V.80: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 3$  e  $N = 60$

$L = 4$

A Figura V.81 apresenta a probabilidade de divergência do BNLMS em termos de  $\beta$  com uma quantidade de blocos  $L = 4$ . Novamente, a análise teórica consegue delimitar com razoabilidade a faixa de  $\beta$  que garante estabilidade.

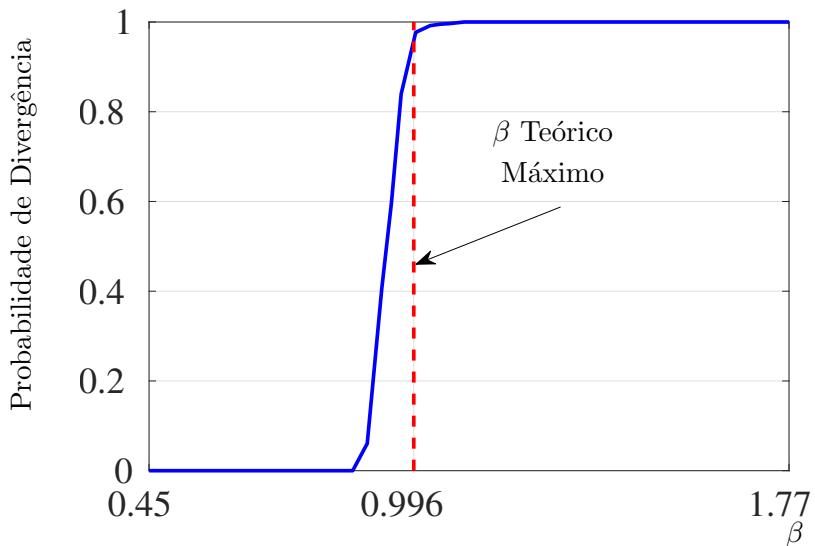


Figura V.81: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 4$  e  $N = 60$

$L$	$\beta$ máximo que evita a divergência	Valores de $\beta$ para BLMS $N = 10$		
		$\beta$ com probabilidade simulada de divergência 0%	$\beta$ com probabilidade simulada de divergência 50%	$\beta$ com probabilidade simulada de divergência 100%
1	1,000000	2,03	2,06	2,085
2	0,998988	1,27	1,315	1,485
3	0,997979	0,95	1,04	1,215
4	0,996971	0,83	0,94	1,115
5	0,995966	0,75	0,86	1,015
6	0,994962	0,73	0,815	0,985
7	0,993961	0,65	0,76	0,915
8	0,992961	0,65	0,74	0,885
9	0,991964	0,57	0,715	0,885
10	0,990968	0,57	0,715	0,815

Tabela V.4: Tabela com valores de  $\beta$  simulados da probabilidade de divergência nas faixas 0%, 50% e 100% para o algoritmo BNLMs e  $N=60$ .

A partir dos dados coletados na execução das simulações representados na Tabela V.4 foram traçadas curvas de  $\beta$  com diferentes probabilidades de divergência conforme Figura V.82 abaixo:

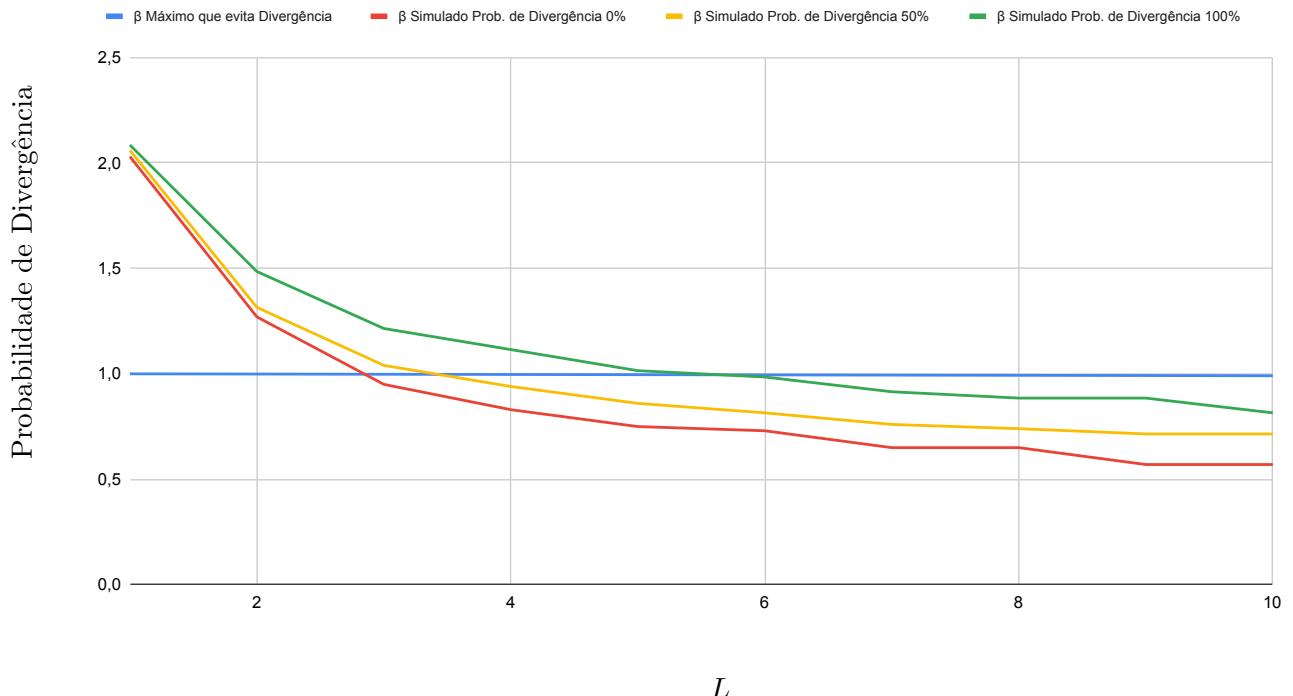


Figura V.82: Curva de probabilidade de divergência simulada versus quantidade de blocos  $L$  do algoritmo BNLMs com  $N = 60$

## Comentários

O segundo cenário de simulação de divergência com tamanho do filtro  $N = 60$  se mostrou bem semelhante ao primeiro com curvas de divergência (Figura V.82) seguindo o mesmo padrão e com a mesma irregularidade abaixo de aproximadamente  $L = 4$ . Isto demonstra claramente a aderência do modelo proposto com a simulação.

### V.4 Comentários Finais

Neste capítulo os resultados simulados foram comparados com os teóricos. Observou-se que o modelo estocástico desenvolvido é capaz de predizer tanto características de taxa de convergência quanto o desempenho assintótico dos algoritmos BLMS e BNLMS. Ademais, a faixa de  $\beta$  que garante convergência também se revelou apropriada, mesmo sabendo que para a análise proposta foram utilizadas as Hipóteses da Independência e do Ruido (III.1) o que sabidamente podem causar distorções nas simulações. Em um contexto prático de aplicação (para os projetistas de Algoritmos de Filtragem Adaptativa) a medida que se escolhe o valor de  $\beta = 1$  ele vai saber como se comporta a divergência a partir desta análise. No caso Normalizado do LMS em Blocos o algoritmo é mais insensível a pequenas variações em relação à  $\beta$ , no caso Não Normalizado do LMS em Blocos a sensibilidade maior é em relação ao tamanho do bloco  $L$ .

## Capítulo VI Conclusões

Este trabalho utilizou um modelo estocástico capaz de prever características de aprendizado de uma estrutura de processamento linear implementada em blocos, treinada tanto com o algoritmo LMS quanto com o algoritmo NLMS.

O modelo estocástico utilizado divorcia a distribuição radial dos vetores de entrada da distribuição angular, que se assume ser discreta. Foram encontradas fórmulas capazes de prever a evolução do MSE ao longo das iterações, em especial as características de divergência em função do fator de aprendizagem  $\beta$ . Também foram elaboradas análises para relacionar o MSE em excesso (em regime permanente) ao tamanho do filtro  $N$  e à quantidade de Blocos  $L$ , adotando para tanto uma hipótese de regime permanente.

Nas simulações foram adotados cenários com parametrizações diferentes, de modo a demonstrar a robustez do modelo, ainda que submetido a diferentes condições. A quantidade de blocos utilizados nas simulações variaram de  $L = 1$  até  $L = 10$ . Mesmo sabendo que para possibilitar as análises deste trabalho foram utilizadas as hipóteses do Ruído e da Independência, as previsões teóricas foram cotejadas com resultados oriundos de simulações, apresentando boa aderência a eles independente da quantidade de Blocos  $L$  adotado.

Como possíveis trabalhos futuros, pretendemos melhorar o *modelo de Slock*, inserindo dependências estatísticas entre as variáveis de sinal, de norma e de direção entre vetores de entrada consecutivos, já que se espera que tais variáveis não mudem de maneira muito abrupta, em geral, já que vetores de entrada sucessivos compartilham muitas amostras entre si. Espera-se que tal modelo mais sofisticado (por conseguinte mais realístico) seja capaz de fornecer diretrizes mais adequadas para o projetista de tais algoritmos. Outro possível trabalho futuro poderia ser expandir a análise através do modelo de Slock permitindo que características polinomiais de algumas classes de algoritmos sejam avaliadas, dentre outros podemos citar os filtros Volterra, que são estruturas de filtragem não-lineares com características polinomiais, cuja complexidade computacional cresce exponencialmente consoante a quantidade de graus de liberdade utilizada [Chao and Inomata, 1997], [Kapgate et al., 2018]. Neste contexto a análise proposta neste trabalho poderia se mostrar bastante valiosa.

## Referências Bibliográficas

- K. Slavakis, S. Kim, G. Mateos, and G. B. Giannakis. Stochastic approximation vis-a-vis online learning for big data analytics [lecture notes]. *IEEE Signal Processing Magazine*, 31(6):124–129, 2014. doi: 10.1109/MSP.2014.2345536. 26
- V. Solo. LMS: Past, present and future. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7740–7744, May. 2019. 26, 50
- Simon Haykin. *Adaptive filter theory*. Prentice Hall, Upper Saddle River, NJ, 4th edition, 2002. 26
- D. T. M. Slock. On the convergence behavior of the LMS and the normalized LMS algorithms. *IEEE Transactions on Signal Processing*, 41(9):2811–2825, Sep. 1993. 28, 51, 58, 59
- Thiago R. P. Gonzaga, Valmir dos S. N. Junior, Ana L. F. de Barros, Felipe da R. Henriques, and Diego B. Haddad. Transient analysis of the block least mean squares algorithm. *IEEE Communications Letters*, 25(2):608–612, 2021. doi: 10.1109/LCOMM.2020.3026610. 29
- Md. Zulfiquar Ali Bhotto and Andreas Antoniou. Affine-projection-like adaptive-filtering algorithms using gradient-based step size. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 61(7):2048–2056, 2014. doi: 10.1109/TCSI.2014.2304665. 29
- R. L. Das and M. Chakraborty. On convergence of proportionate-type normalized least mean square algorithms. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 62(5):491–495, 2015. 29, 51
- Shihab Jimaa, Tetsuya Shimamura, and Hideki Takekawa. Convergence evaluation of a variable step-size lmse adaptive switching algorithm. In *2008 IEEE International Networking and Communications Conference*, pages 23–27, 2008. doi: 10.1109/INCC.2008.4562685. 29
- T. K. Paul and T. Ogunfunmi. On the convergence behavior of the affine projection algorithm for adaptive filters. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 58(8):1813–1826, 2011. 29, 51
- Paulo S. R. Diniz, Eduardo A. B. da Silva, and Sergio L. Netto. *Digital Signal Processing: System Analysis and Design*. Cambridge University Press, 2 edition, 2010. doi: 10.1017/CBO9780511781667. 33, 42

- P. Schniter and C. R. Johnson. Dithered signed-error CMA: robust, computationally efficient blind adaptive equalization. *IEEE Transactions on Signal Processing*, 47(6):1592–1603, 1999. doi: 10.1109/78.765129. 34
- N. Verhoeckx and T. Claasen. Some considerations on the design of adaptive digital filters equipped with the sign algorithm. *IEEE Transactions on Communications*, 32(3):258–266, 1984. doi: 10.1109/TCOM.1984.1096064. 41
- E. Walach and B. Widrow. The least mean fourth (LMF) adaptive algorithm and its family. *IEEE Transactions on Information Theory*, 30(2):275–283, 1984. doi: 10.1109/TIT.1984.1056886. 41
- J.A. Chambers, O. Tanrikulu, and A.G. Constantinides. Least mean mixed-norm adaptive filtering. *Electronics Letters*, 30(19):1574–1575, September 1994. doi: 10.1049/el:19941060. URL <https://doi.org/10.1049/el:19941060>. 41
- Y. Gu, J. Jin, and S. Mei.  $l_0$  norm constraint LMS algorithm for sparse system identification. *IEEE Signal Processing Letters*, 16(9):774–777, 2009. doi: 10.1109/LSP.2009.2024736. 41
- I. Song, P. Park, and R. W. Newcomb. A normalized least mean squares algorithm with a step-size scaler against impulsive measurement noise. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 60(7):442–445, 2013. doi: 10.1109/TCSII.2013.2258266. 41
- M. O. Sayin, N. D. Vanli, and S. S. Kozat. A novel family of adaptive filtering algorithms based on the logarithmic cost. *IEEE Transactions on Signal Processing*, 62(17):4411–4424, 2014. doi: 10.1109/TSP.2014.2333559. 41
- H. Zayyani. Continuous mixed  $p$ -norm adaptive algorithm for system identification. *IEEE Signal Processing Letters*, 21(9):1108–1110, 2014. doi: 10.1109/LSP.2014.2325495. 41
- I. TSG. 5, *digital network echo cancellers (recommendation.* tech. rep., Tech. Rep. G. 168, ITU-T, 2004. 42
- B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson. Stationary and nonstationary learning characteristics of the lms adaptive filter. *Proceedings of the IEEE*, 64(8):1151–1162, 1976. doi: 10.1109/PROC.1976.10286. 49
- K. J. Quirk, J. R. Zeidler, and L. B. Milstein. Bounding the performance of the LMS estimator for cases where performance exceeds that of the finite Wiener filter. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, volume 3, pages 1417–1420 vol.3, 1998. 50, 51

- S. C. Douglas. Exact expectation analysis of the sign-data LMS algorithm for i.i.d. input data. In *[1992] Conference Record of the Twenty-Sixth Asilomar Conference on Signals, Systems Computers*, pages 566–570 vol.1, Oct. 1992. 50
- S. C. Douglas. Exact expectation analysis of the LMS adaptive filter for correlated Gaussian input data. In *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 519–522 vol.3, Apr. 1993. 50
- S. C. Douglas and W. Pan. Exact expectation analysis of the LMS adaptive filter. *IEEE Transactions on Signal Processing*, 43(12):2863–2871, Dec. 1995. 50
- P. Lara, L. D. T. J. Tarrataca, and D. B. Haddad. Exact expectation analysis of the deficient-length LMS algorithm. *Signal Processing*, 162:54 – 64, 2019a. 50
- P. Lara, K. d. S. Olinto, F. R. Petraglia, and D. B. Haddad. Exact analysis of the least-mean-square algorithm with coloured measurement noise. *Electronics Letters*, 54(24):1401–1403, 2018. 50
- Pedro Lara, Luís D.T.J. Tarrataca, and Diego B. Haddad. Exact expectation analysis of the deficient-length LMS algorithm. *Signal Processing*, 162:54 – 64, 2019b. 50
- Pedro Lara, Diego B. Haddad, and Luís Tarrataca. Advances on the analysis of the LMS algorithm with a colored measurement noise. *Signal, Image and Video Processing*, 14:529–536, Oct. 2019c. 50, 58
- S. C. Douglas and T. H.-Y. Meng. Exact expectation analysis of the LMS adaptive filter without the independence assumption. In *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 61–64, 1992. 50
- P. Lara, F. Igreja, T. T. P. Silva, L. D. T. J. Tarrataca, and D. B. Haddad. Exact expectation analysis of the LMS adaptive identification of non-linear systems. *Electronics Letters*, 56(1): 45–48, Jan. 2020. 50
- T. T. P. Silva, P. Lara, T. Gonzaga, D. Brandão, L. Tarrataca, and D. B. Haddad. Recent advances in overcoming the independence assumption in adaptive filtering analyses. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 317–322, 2020. doi: 10.1109/IWSSIP48289.2020.9145102. 50
- T. T. P. Silva, F. Igreja, P. Lara, L. Tarrataca, A. Kar, and D. B. Haddad. On the skewness of the LMS adaptive weights. *IEEE Transactions on Circuits and Systems II: Express Briefs*, pages 1–1, 2021. 50

- H. J. Butterweck. A wave theory of long adaptive filters. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(6):739–747, Jun. 2001. 50
- M. Rupp and H. . Butterweck. Overcoming the independence assumption in LMS filtering. In *The Thirty-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, volume 1, pages 607–611 Vol.1, 2003. 50
- H. J. Butterweck. Iterative analysis of the steady-state weight fluctuations in LMS-type adaptive filters. *IEEE Transactions on Signal Processing*, 47(9):2558–2561, 1999. 50
- H. J. Butterweck. An approach to LMS adaptive filtering without use of the independence assumption. In *1996 8th European Signal Processing Conference (EUSIPCO 1996)*, pages 1–4, 1996a. 50
- H. J. Butterweck. A new interpretation of the misadjustment in adaptive filtering. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 3, pages 1641–1643 vol. 3, 1996b. 50
- H. J. Butterweck. A steady-state analysis of the LMS adaptive algorithm without use of the independence assumption. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1404–1407 vol.2, 1995. 50
- S. Miyoshi and Y. Kajikawa. A theory of the FXLMS algorithm based on statistical-mechanical method. In *2013 8th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 645–650, 2013. 50
- S. Miyoshi and Y. Kajikawa. Statistical-mechanical analysis of the FXLMS algorithm with actual primary path. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3502–3506, 2015. 50
- T. Murata, Y. Kajikawa, and S. Miyoshi. Statistical-mechanical analysis of the FXLMS algorithm for multiple-channel active noise control. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 136–139, 2017. 50
- N. J. Bershad, E. Eweda, and J. C. M. Bermudez. Stochastic analysis of the LMS and NLMS algorithms for cyclostationary white gaussian inputs. *IEEE Transactions on Signal Processing*, 62(9):2238–2249, 2014. 50
- A. Gupta and S. Joshi. Variable step-size LMS algorithm for fractal signals. *IEEE Transactions on Signal Processing*, 56(4):1411–1420, 2008. 50

- R. Iltis and L. Milstein. An approximate statistical analysis of the Widrow LMS algorithm with application to narrow-band interference rejection. *IEEE Transactions on Communications*, 33(2):121–130, 1985. 50
- B. Farhang-Boroujeny and S. Gazor. Performance of LMS-based adaptive filters in tracking a time-varying plant. *IEEE Transactions on Signal Processing*, 44(11):2868–2871, 1996. 50
- J. C. M. Bermudez and N. J. Bershad. Transient and tracking performance analysis of the quantized LMS algorithm for time-varying system identification. *IEEE Transactions on Signal Processing*, 44(8):1990–1997, 1996. 50
- N. R. Yousef and A. H. Sayed. A unified approach to the steady-state and tracking analyses of adaptive filters. *IEEE Transactions on Signal Processing*, 49(2):314–324, 2001. 50
- A. H. Sayed and M. Rupp. Error-energy bounds for adaptive gradient algorithms. *IEEE Transactions on Signal Processing*, 44(8):1982–1989, 1996. 50, 51
- X. Zhao and A. H. Sayed. Performance limits for distributed estimation over LMS adaptive networks. *IEEE Transactions on Signal Processing*, 60(10):5107–5124, 2012. 50
- T. Y. Al-Naffouri and A. H. Sayed. Transient analysis of data-normalized adaptive filters. *IEEE Transactions on Signal Processing*, 51(3):639–652, 2003. 51
- N. R. Yousef and A. H. Sayed. Ability of adaptive filters to track carrier offsets and channel nonstationarities. *IEEE Transactions on Signal Processing*, 50(7):1533–1544, 2002. 51
- N. R. Yousef and A. H. Sayed. Steady-state and tracking analyses of the sign algorithm without the explicit use of the independence assumption. *IEEE Signal Processing Letters*, 7(11):307–309, 2000. 51
- D. B. Haddad, M. R. Petraglia, and A. Petraglia. A unified approach for sparsity-aware and maximum correntropy adaptive filters. In *2016 24th European Signal Processing Conference (EUSIPCO)*, pages 170–174, 2016. 51
- M. V. S. Lima, I. Sobron, W. A. Martins, and P. S. R. Diniz. Stability and MSE analyses of affine projection algorithms for sparse system identification. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6399–6403, 2014. 51
- B. Chen, Z. Qin, and L. Sun. Steady-state mean square performance of a sparsified kernel least mean square algorithm. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2701–2705, 2017. 51

- T. Ikuma, A. A. Beex, and J. R. Zeidler. Non-wiener mean weight behavior of LMS transversal equalizers with sinusoidal interference. *IEEE Transactions on Signal Processing*, 56(9):4521–4525, 2008. 51
- S. Olmos and P. Laguna. Steady-state MSE convergence of LMS adaptive filters with deterministic reference inputs with applications to biomedical signals. *IEEE Transactions on Signal Processing*, 48(8):2229–2241, 2000. 51
- P. Clarkson and P. White. Simplified analysis of the LMS adaptive filter using a transfer function approximation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(7):987–993, 1987. 51
- P. F. Titchener. Transient behavior of the LMS algorithm: a deterministic approach. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 1491–1494 vol.3, 1990. 51
- B. Hassibi, A. H. Sayed, and T. Kailath.  $\mathcal{H}^\infty$ -optimality of the LMS algorithm. *IEEE Transactions on Signal Processing*, 44(2):267–280, 1996. 51
- A. Ali, M. Moinuddin, and T. Y. Al-Naffouri. The NLMS is steady-state schur-convex. *IEEE Signal Processing Letters*, 28:389–393, 2021. doi: 10.1109/LSP.2021.3055460. 51
- S. Werner, M. L. R. de Campos, and P. S. R. Diniz. Partial-update NLMS algorithms with data-selective updating. *IEEE Transactions on Signal Processing*, 52(4):938–949, 2004. 51
- J. Apolinario, M. L. R. Campos, and P. S. R. Diniz. Convergence analysis of the binormalized data-reusing LMS algorithm. *IEEE Transactions on Signal Processing*, 48(11):3235–3242, 2000. 51
- Rodrigo Pimenta, Newton Siqueira, Mariane Petraglia, and Diego Haddad. Transient analysis of the bias-compensated lms algorithm. *Journal of Communication and Information Systems*, 36(1):114–118, Jun. 2021. doi: 10.14209/jcis.2021.12. URL <https://jcis.sbrt.org.br/jcis/article/view/762>. 52
- Rodrigo M. S. Pimenta, Leonardo C. Resende, Newton N. Siqueira, Idiego B. Haddad, and Mariane R. Petraglia. A new proportionate adaptive filtering algorithm with coefficient reuse and robustness against impulsive noise. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 465–469, 2018. doi: 10.23919/EUSIPCO.2018.8553593. 52
- Thiago T. P. Silva, Pedro Lara, Filipe Igreja, Fernanda D. V. R. Oliveira, Luís Tarrataca, and Diego B. Haddad. An exact expectation model for the lms tracking abilities. *IEEE Transactions on Signal Processing*, 68:5882–5893, 2020. doi: 10.1109/TSP.2020.3028993. 52

- G. Clark, S. Mitra, and S. Parker. Block implementation of adaptive digital filters. *IEEE Transactions on Circuits and Systems*, 28(6):584–592, 1981. 68
- T. Wang and C. . Wang. A new optimum block adaptive FIR filtering algorithm. In *TENCON'92 - Technology Enabling Tomorrow*, pages 725–729 vol.2, 1992. 68
- G. Panda, B. Mulgrew, C. Cowan, and P. Grant. A self-orthogonalizing efficient block adaptive filter. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(6):1573–1582, 1986. 68
- Y. Dacheng and W. Dejung. A simplified block LMS algorithm. In *1988., IEEE International Symposium on Circuits and Systems*, pages 277–280 vol.1, 1988. 68
- W. B. Mikhael and S. M. Ghosh. Two-dimensional block adaptive filtering algorithms with optimum convergence factors. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 42(8):505–515, 1995. 68
- Hun Choi, Geun-Taek Ryu, Dae-SungKim, Ki-Tae Lim, Tack-Won Kwon, Hae-Jin Lim, and Hyeon-Deok Bae. Variable block LMS adaptive filters with gaussian input. In *Proceedings of 2002 IEEE 10th Digital Signal Processing Workshop, 2002 and the 2nd Signal Processing Education Workshop.*, pages 217–220, 2002. 68
- M. R. Azimi-Sadjadi and Hongye Pan. Two-dimensional block diagonal LMS adaptive filtering. *IEEE Transactions on Signal Processing*, 42(9):2420–2429, 1994. 68
- N. Gowtham and P. Babu. An efficient architecture for BLMS adaptive filter based on distributed arithmetic technique. In *2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE)*, pages 1–5, 2014. 68
- Jinhui Chao and A. Inomata. A convergence analysis of volterra adaptive filters. In *1997 IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 4, pages 2477–2480 vol.4, 1997. doi: 10.1109/ISCAS.1997.612826. 134
- Sachin N. Kapgate, Saurav Gupta, and Ajit Kumar Sahoo. Adaptive volterra modeling for nonlinear systems based on lms variants. In *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 258–263, 2018. doi: 10.1109/SPIN.2018.8474036. 134

## Capítulo VII Apêndice A

Este apêndice é composto por todos os cenários de simulação que ficaram de fora da parte principal deste trabalho, caso o leitor queira aferir melhor a acurácia dos modelos desenvolvidos.

### Simulação 1 - MSE e $\tilde{\lambda}_i$

#### BLMS

Cenário 1 →  $N = 10$

$L = 5$

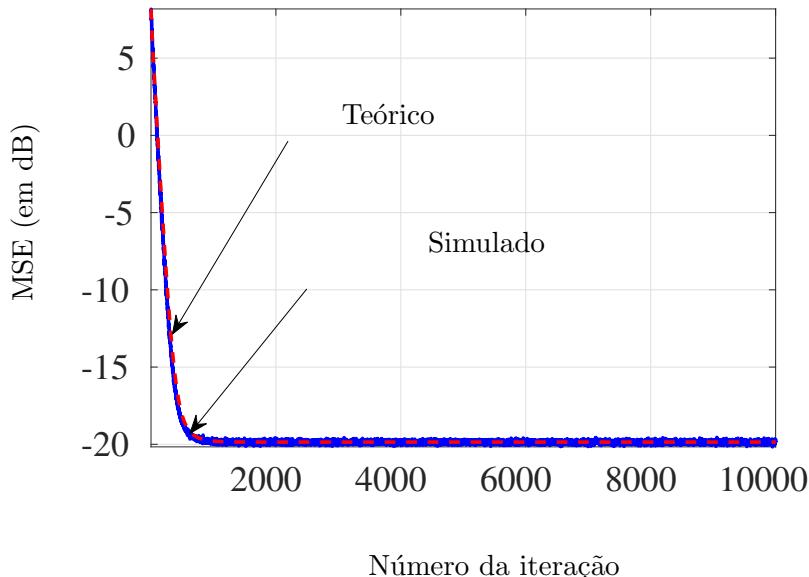


Figura VII.1: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

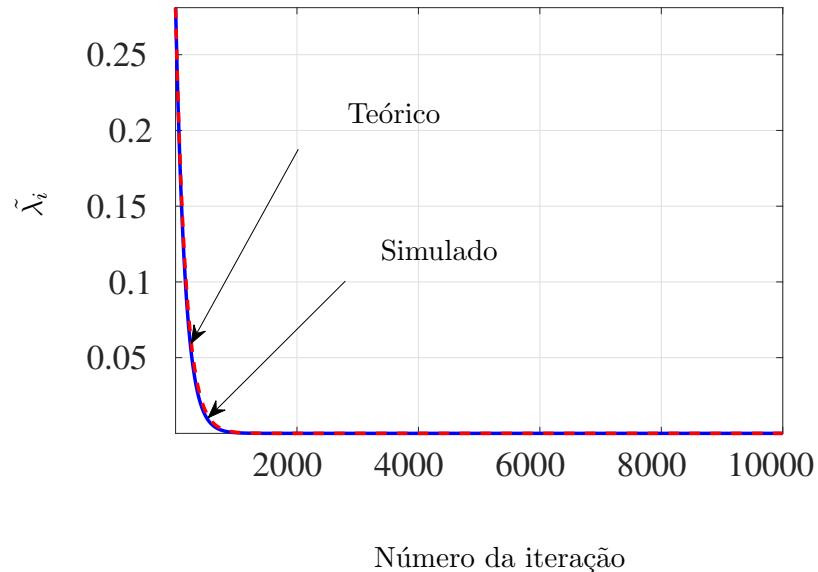


Figura VII.2: Evolução do  $\tilde{\lambda}_i(k)$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$ .

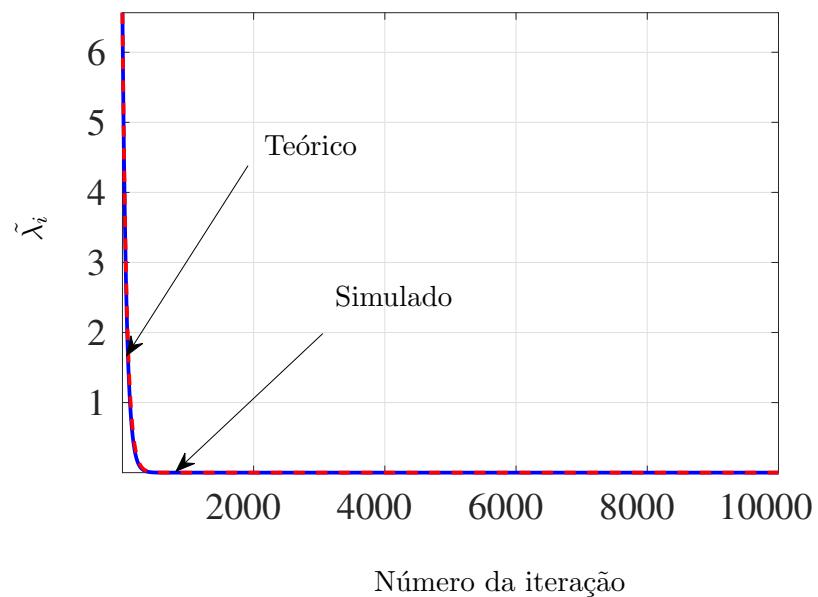


Figura VII.3: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

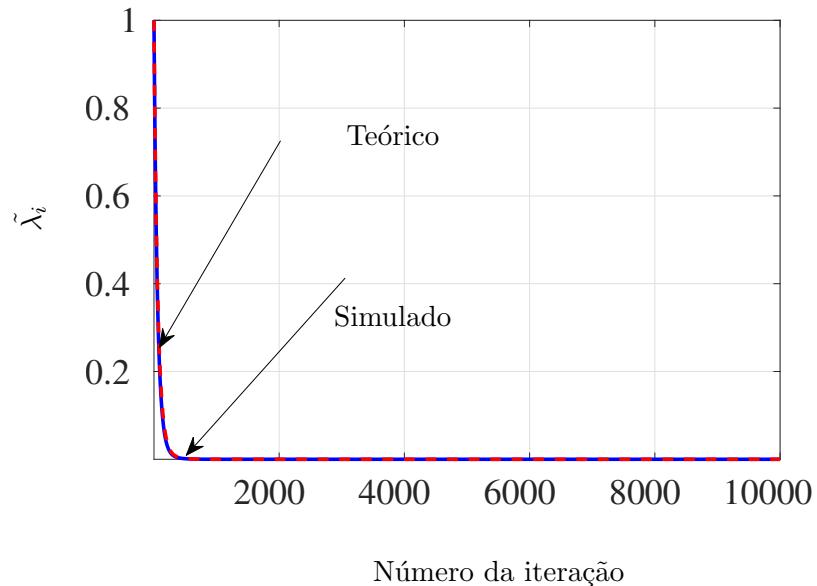


Figura VII.4: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

$L = 6$

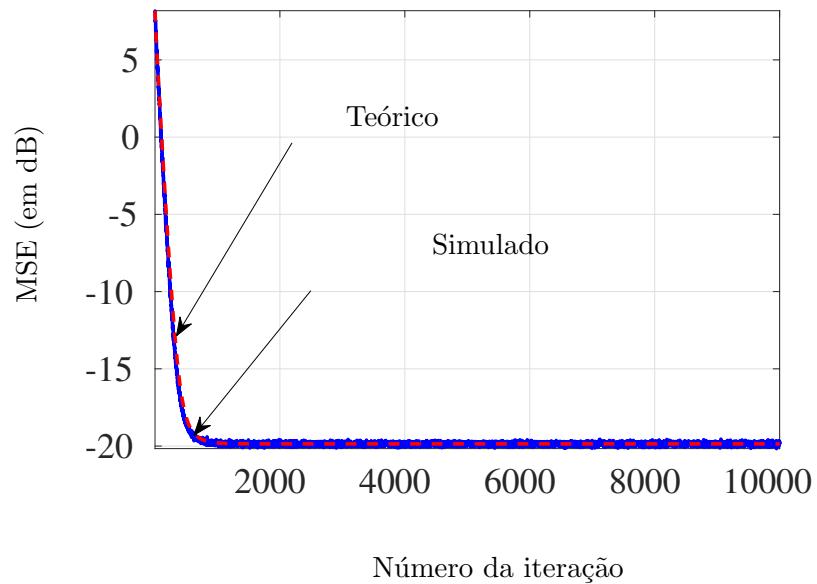


Figura VII.5: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

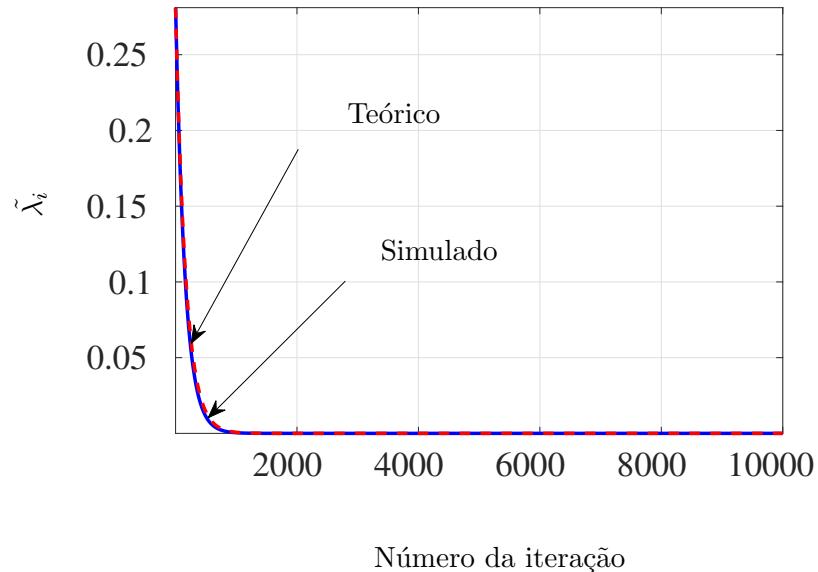


Figura VII.6: Evolução do  $\tilde{\lambda}_i(k)$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$ .

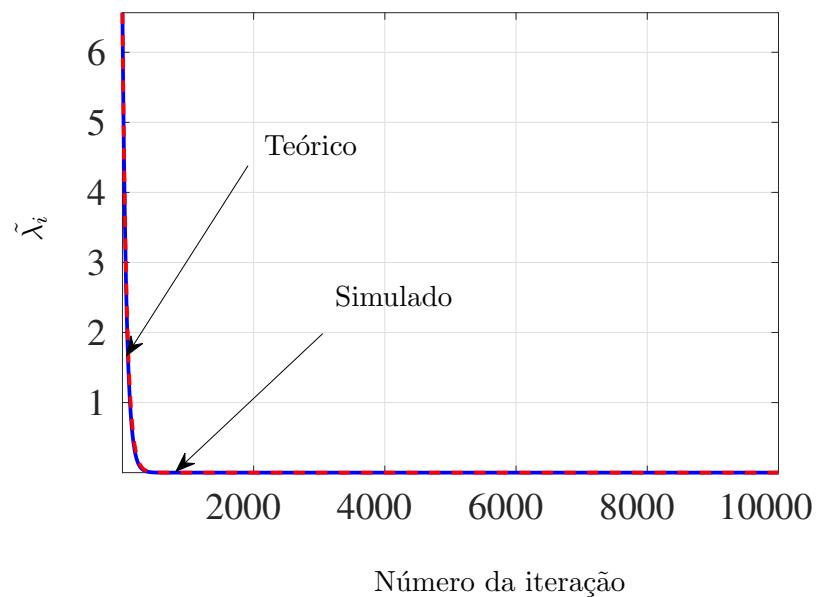


Figura VII.7: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

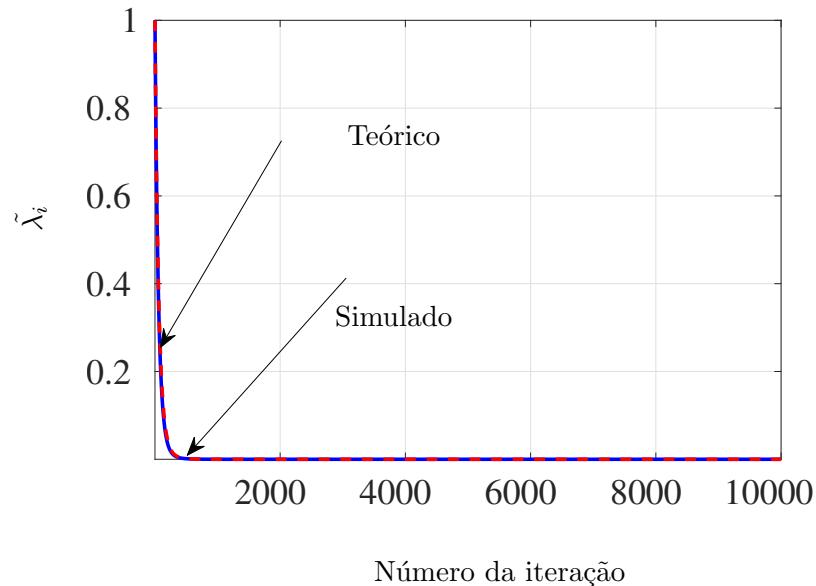


Figura VII.8: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

$L = 7$

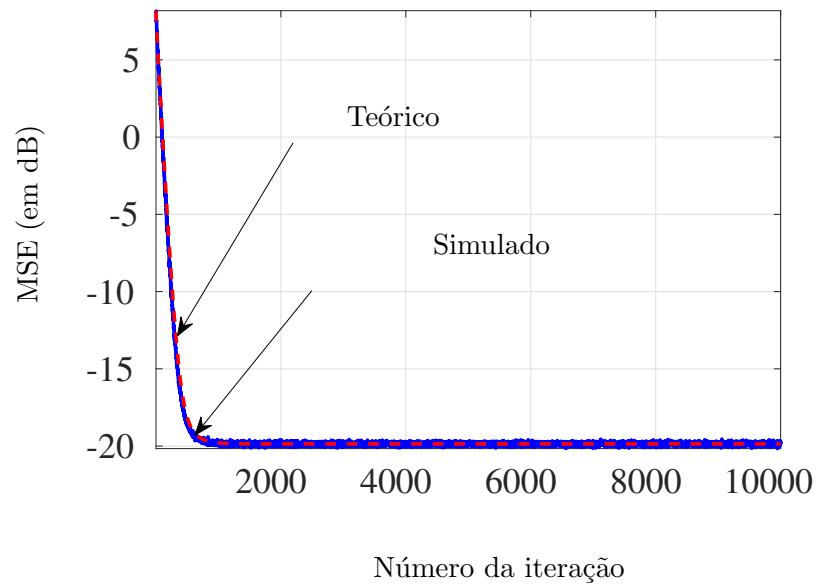


Figura VII.9: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

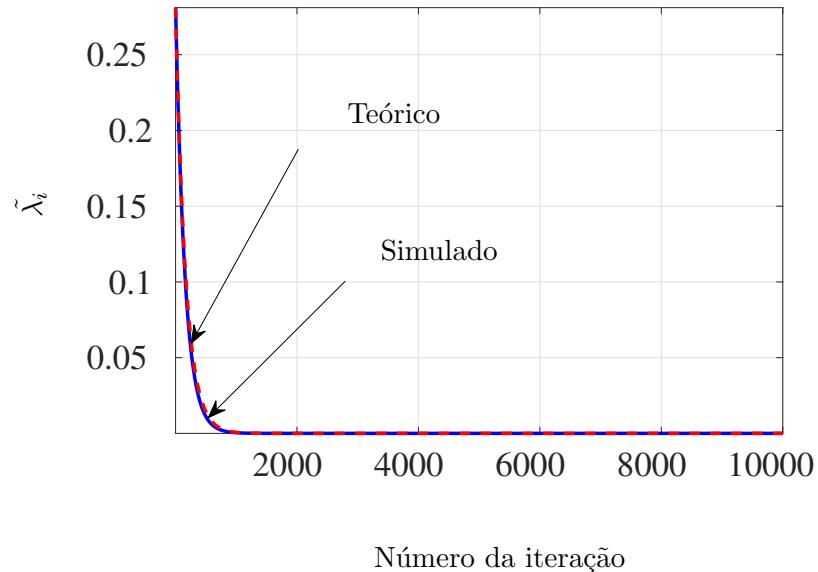


Figura VII.10: Evolução do  $\tilde{\lambda}_i(k)$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$ .

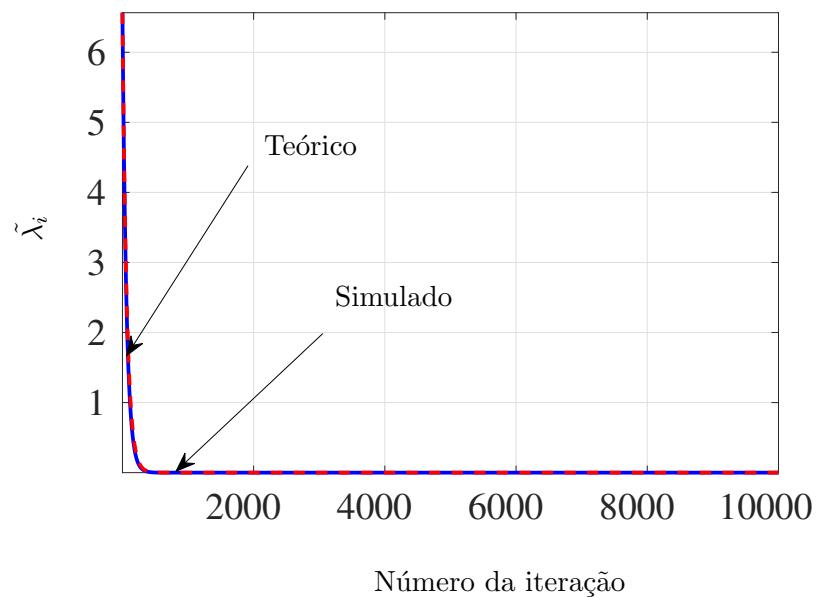


Figura VII.11: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

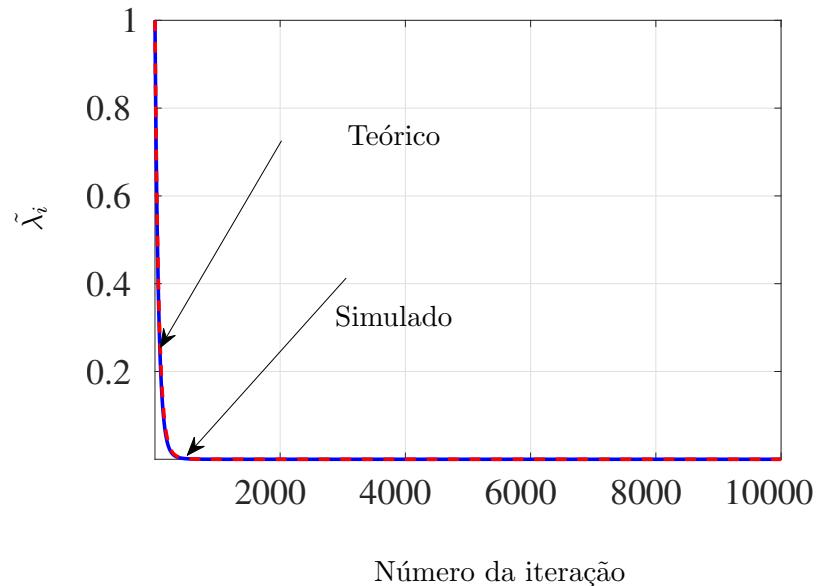


Figura VII.12: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

$L = 8$

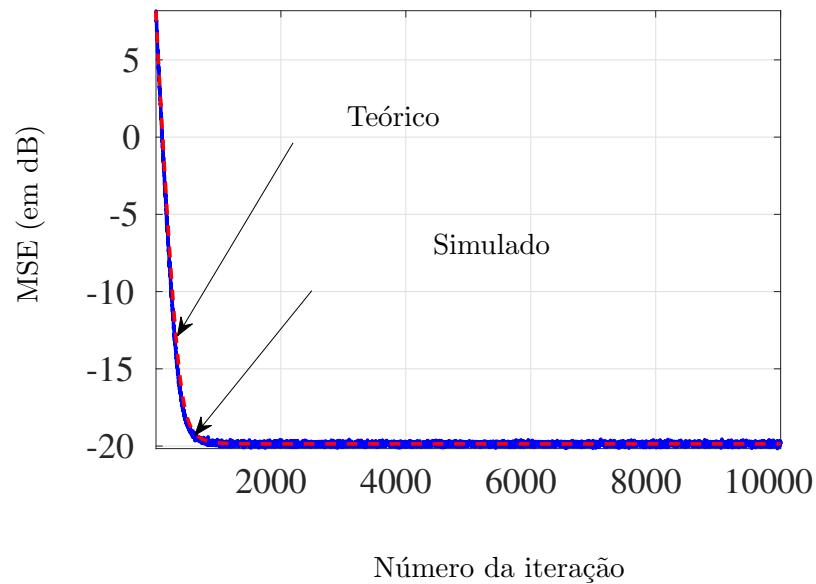


Figura VII.13: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

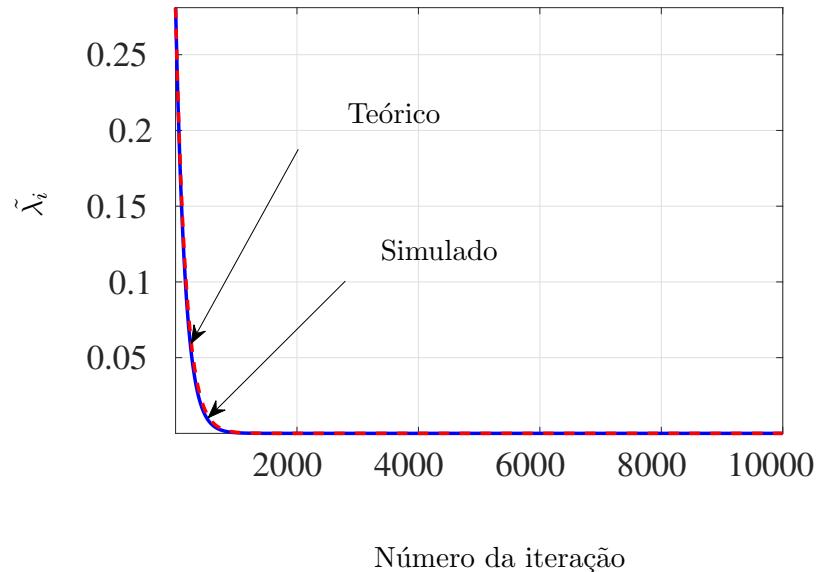


Figura VII.14: Evolução do  $\tilde{\lambda}_i(k)$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$ .

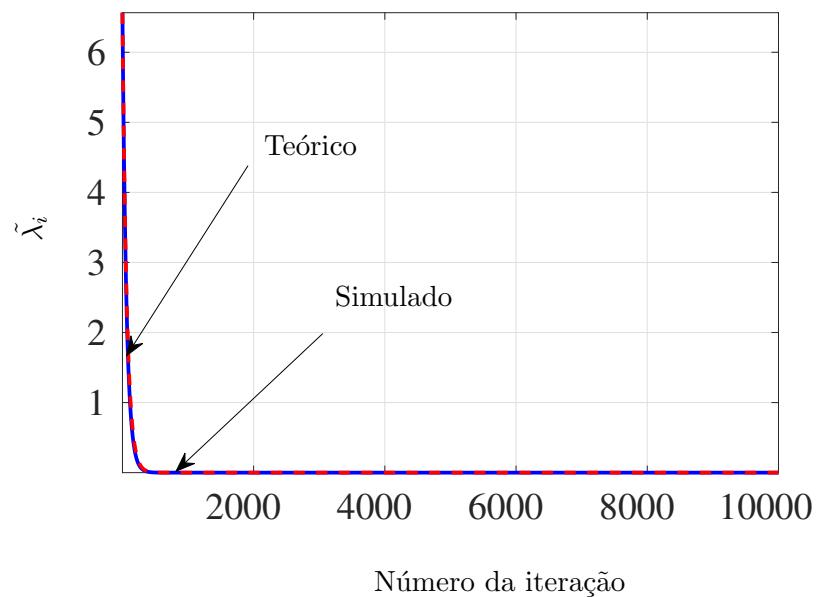


Figura VII.15: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

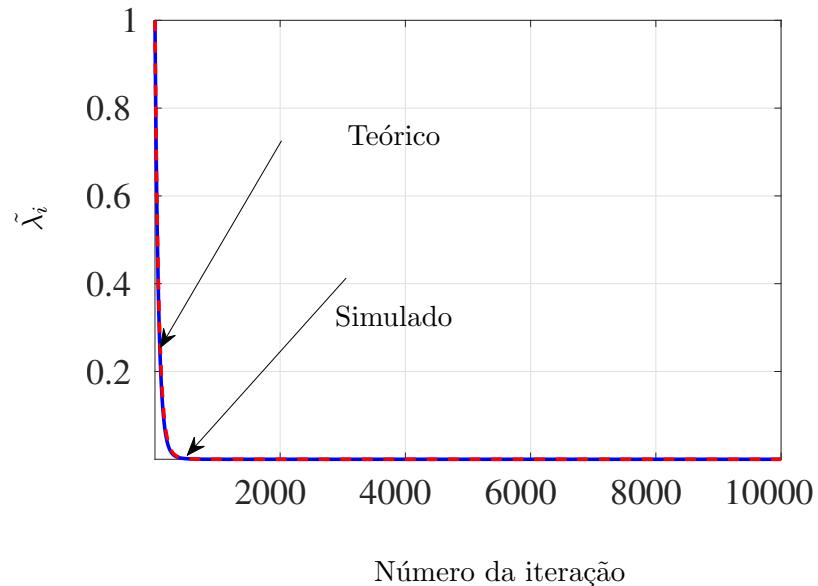


Figura VII.16: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

$L = 9$

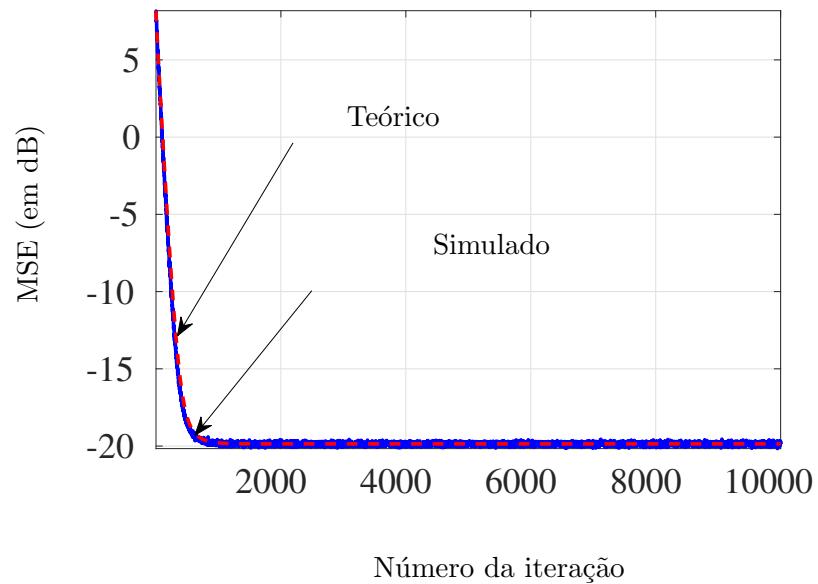


Figura VII.17: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

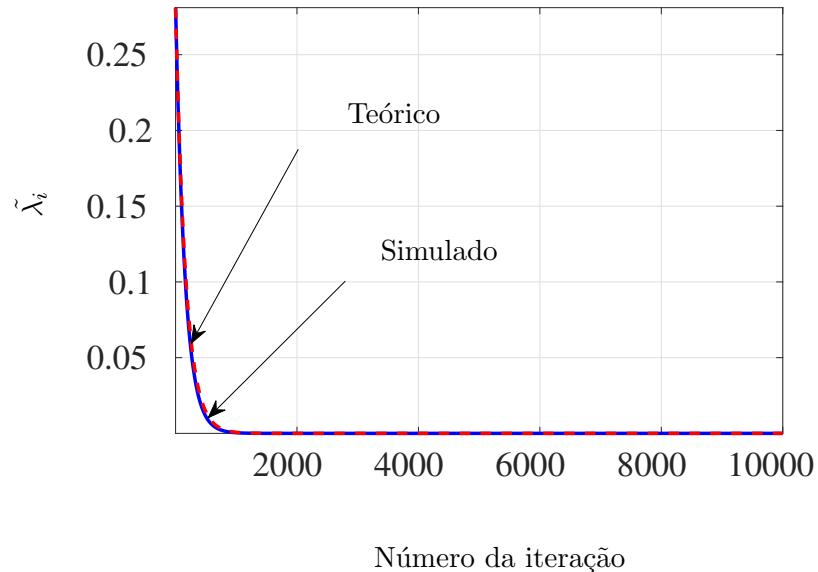


Figura VII.18: Evolução do  $\tilde{\lambda}_i(k)$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$ .

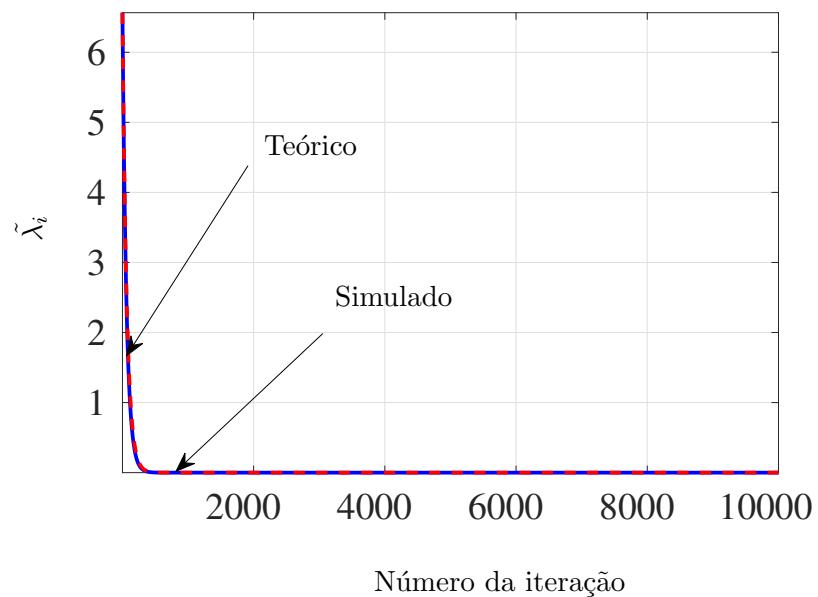


Figura VII.19: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

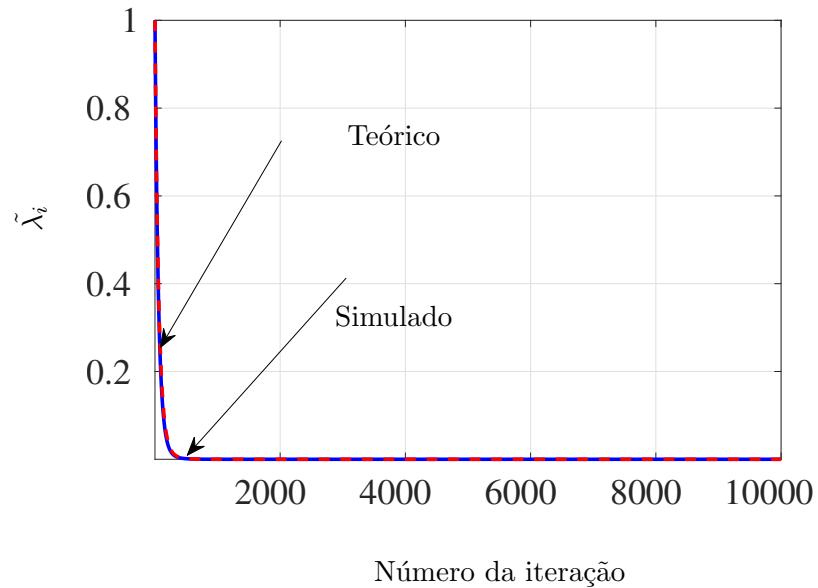


Figura VII.20: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

$L = 10$

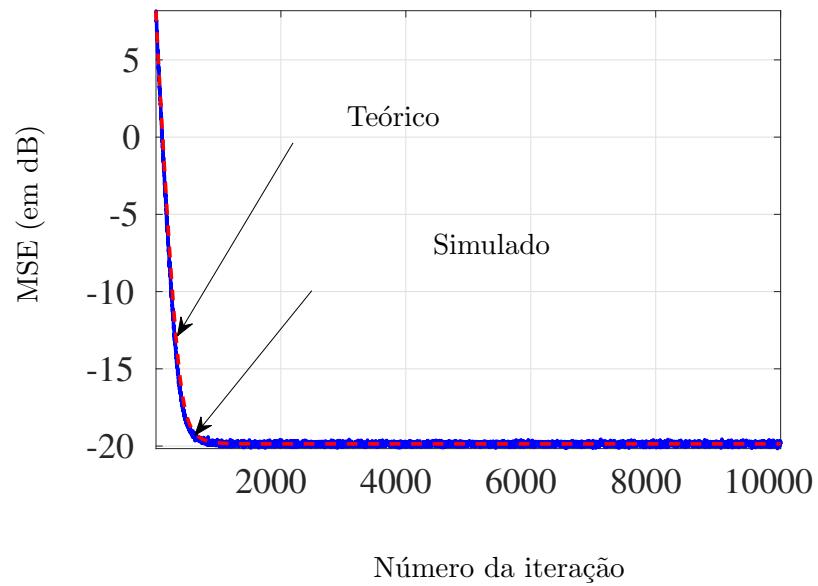


Figura VII.21: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

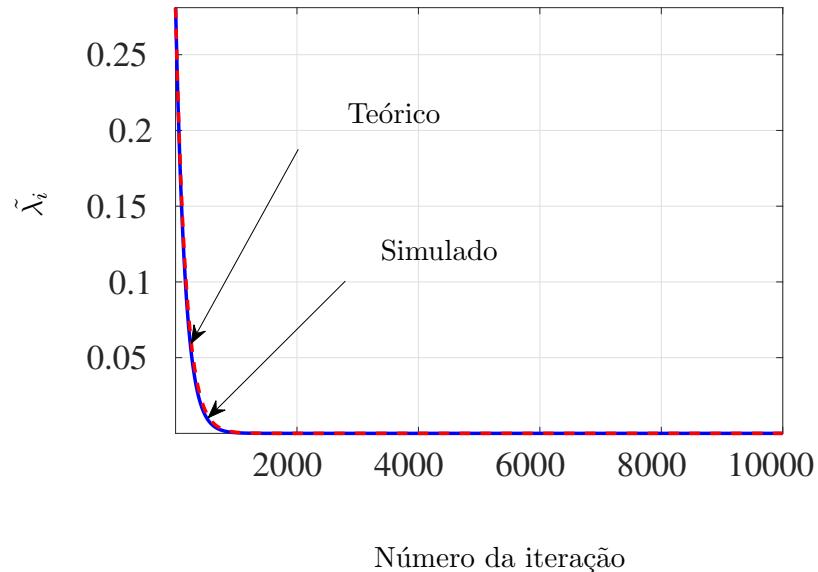


Figura VII.22: Evolução do  $\tilde{\lambda}_i(k)$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$ .

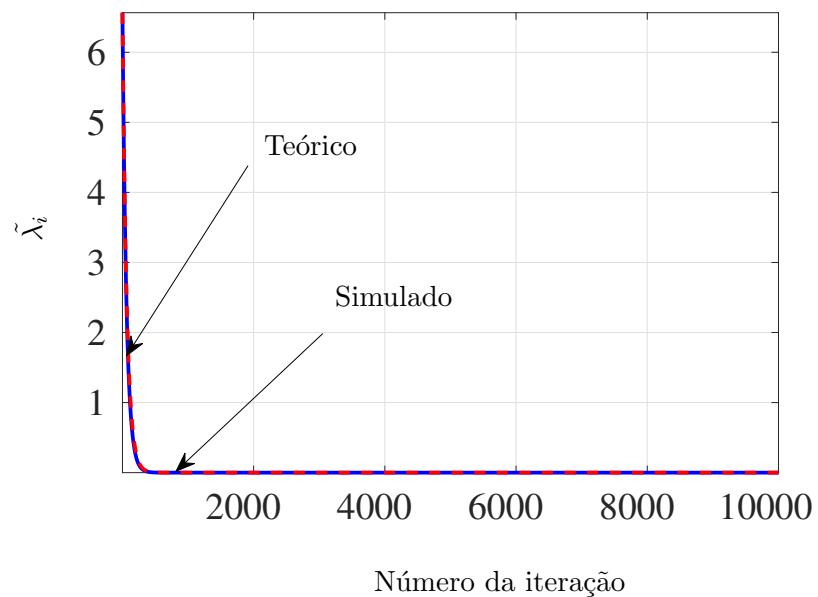


Figura VII.23: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

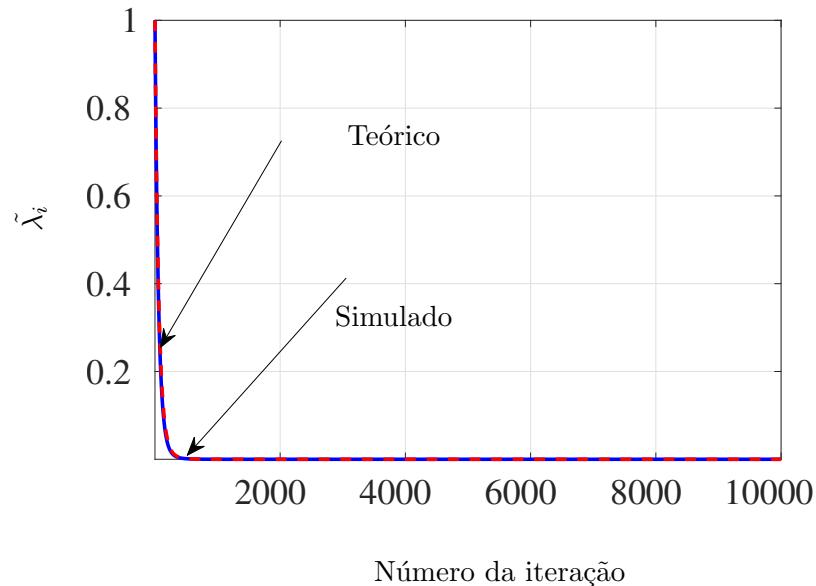


Figura VII.24: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

**Cenário 2** →  $N = 60$

$L = 5$

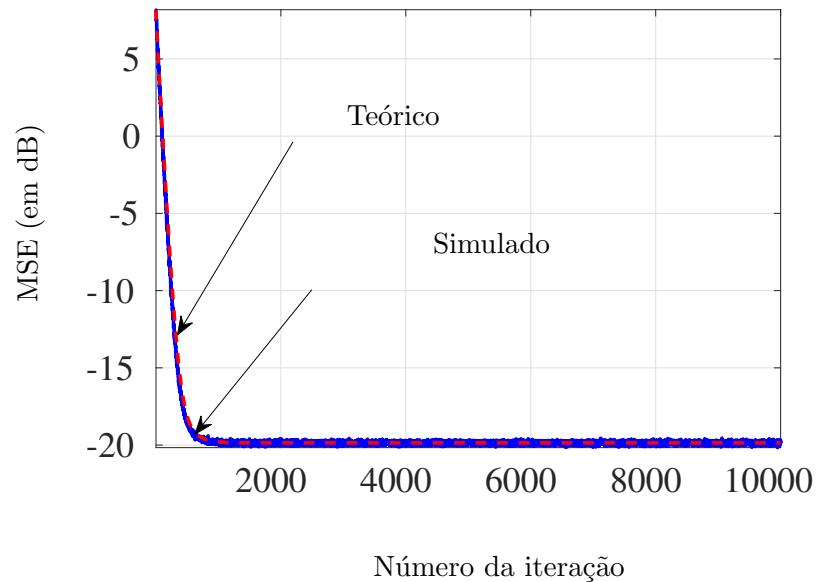


Figura VII.25: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

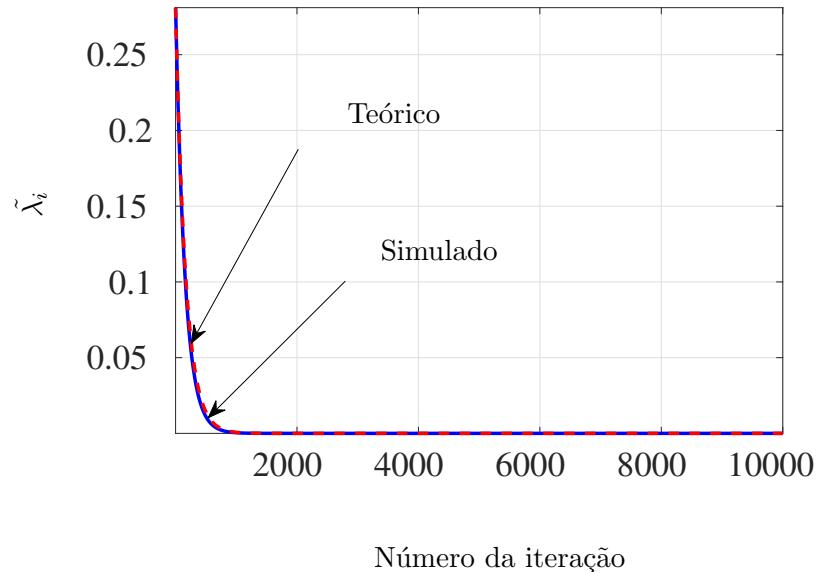


Figura VII.26: Evolução do  $\tilde{\lambda}_i(k)$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$ .

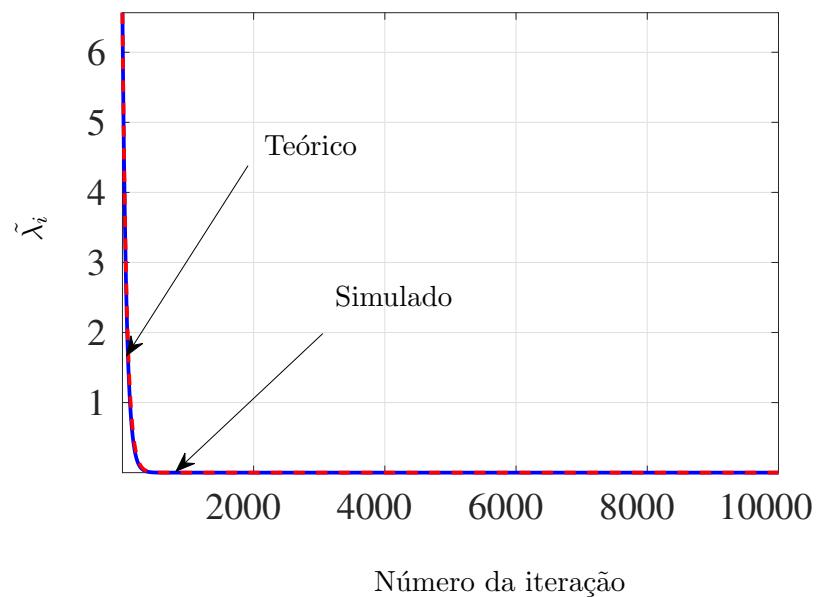


Figura VII.27: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

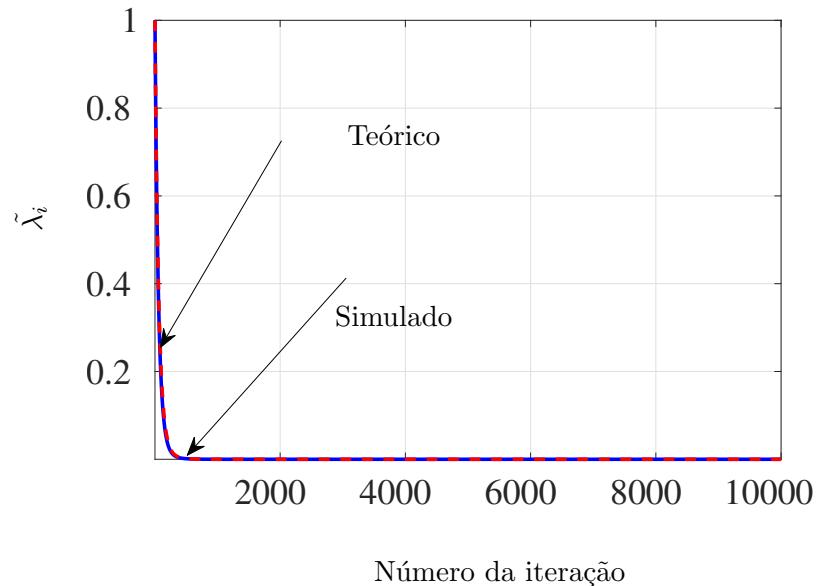


Figura VII.28: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

$L = 6$

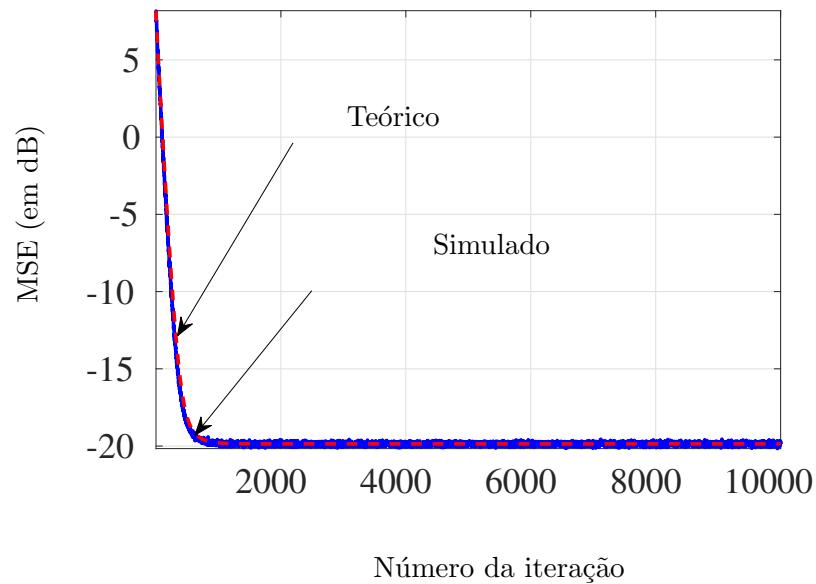


Figura VII.29: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

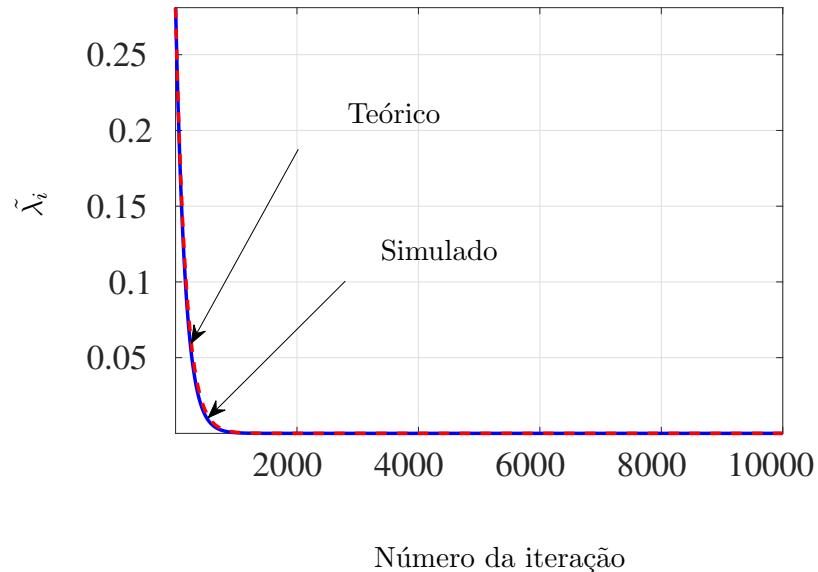


Figura VII.30: Evolução do  $\tilde{\lambda}_i(k)$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$ .

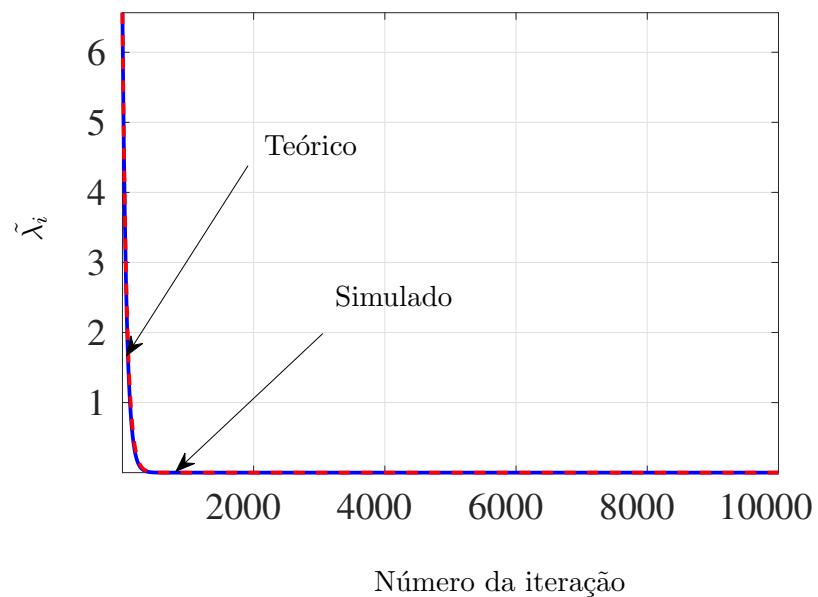


Figura VII.31: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

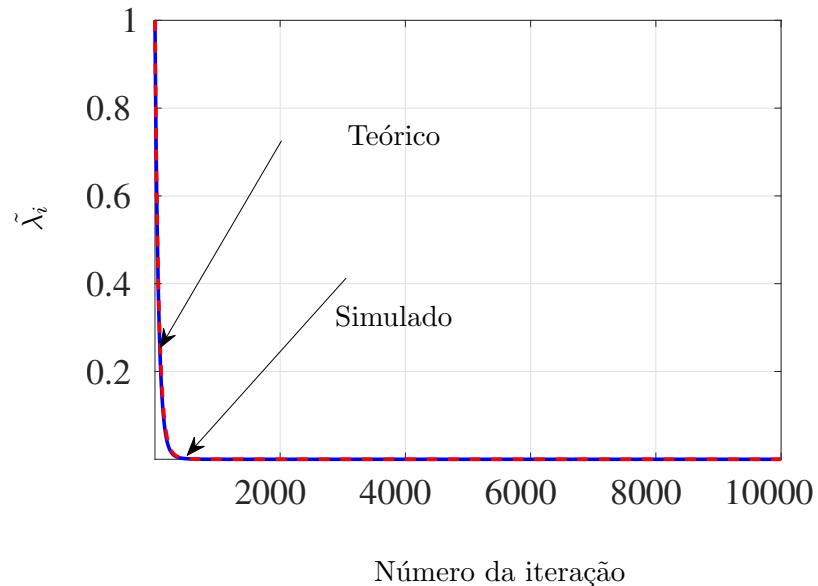


Figura VII.32: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

$L = 7$

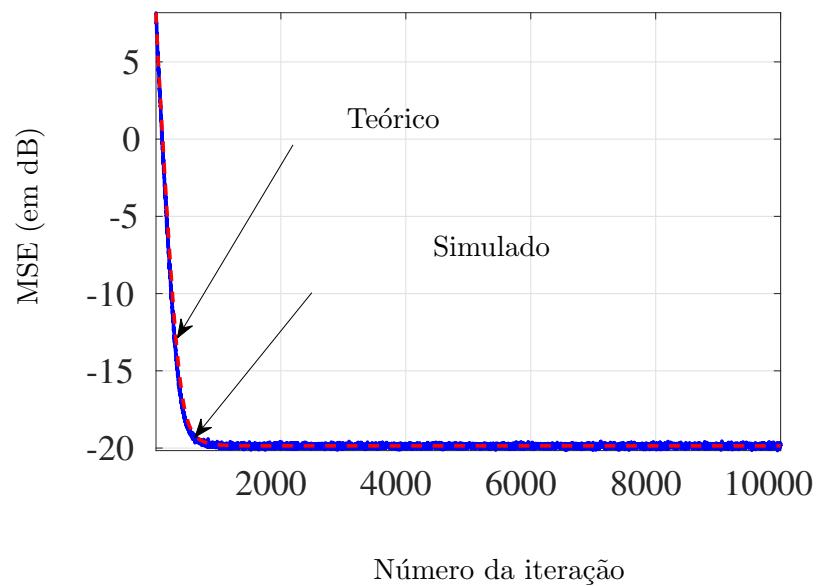


Figura VII.33: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

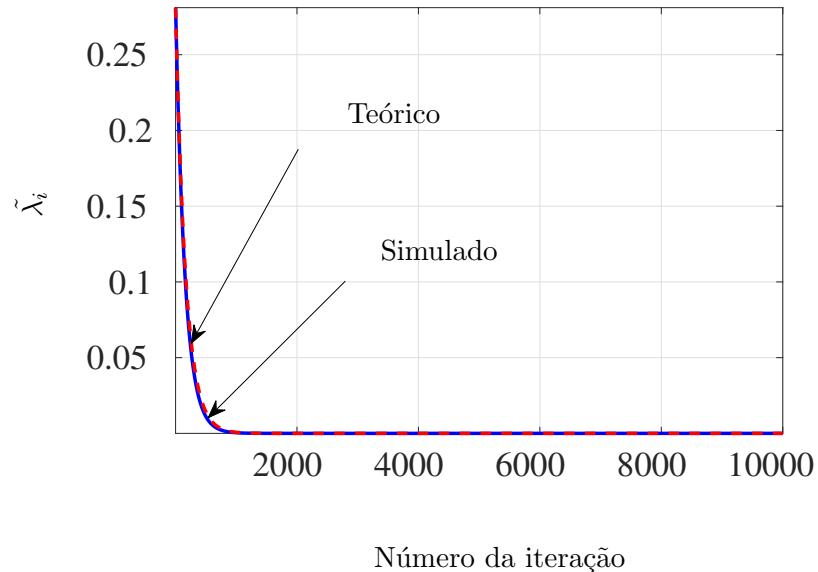


Figura VII.34: Evolução do  $\tilde{\lambda}_i(k)$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$ .

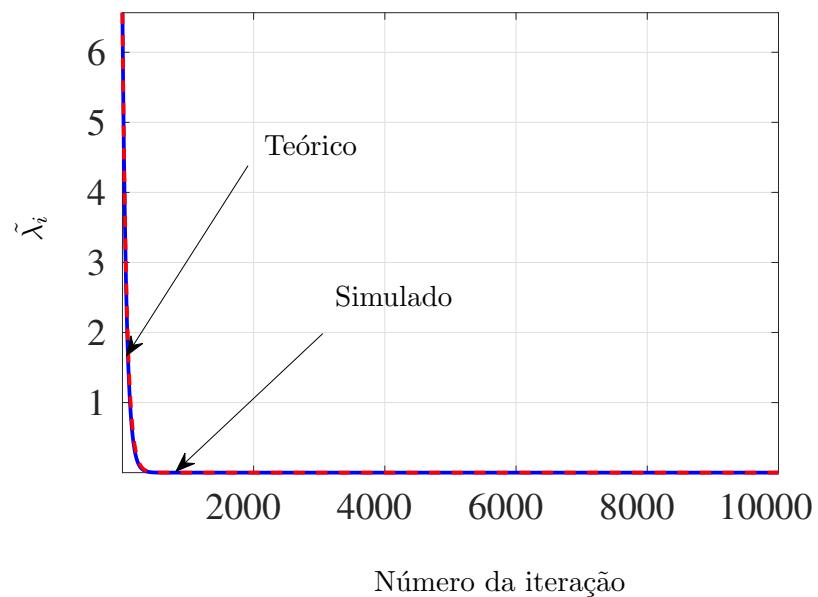


Figura VII.35: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

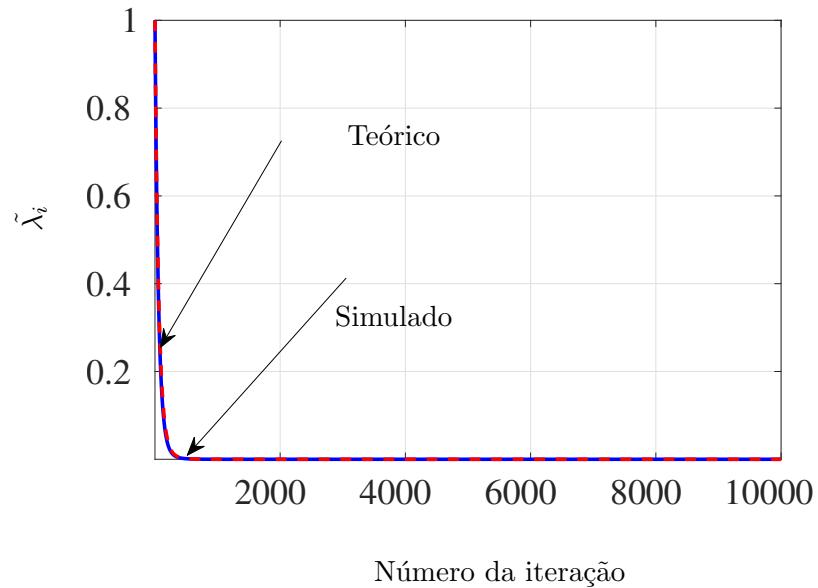


Figura VII.36: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

$L = 8$

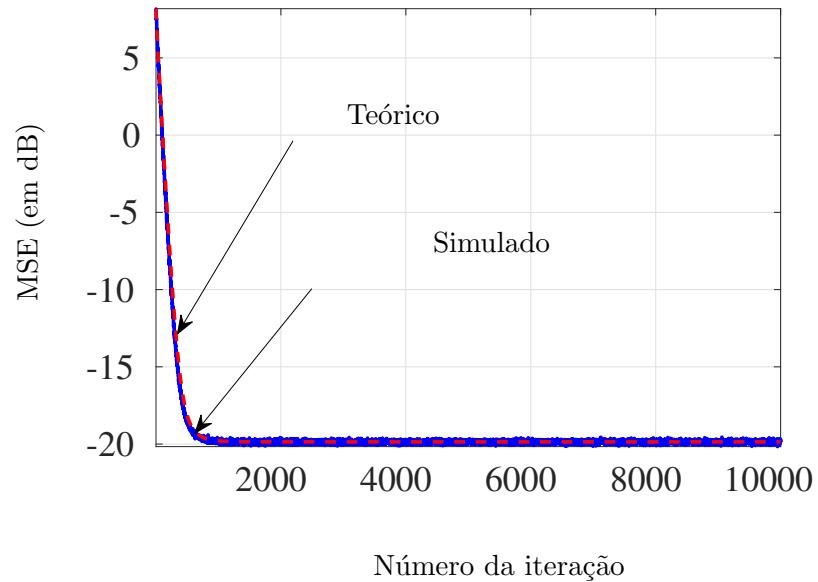


Figura VII.37: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

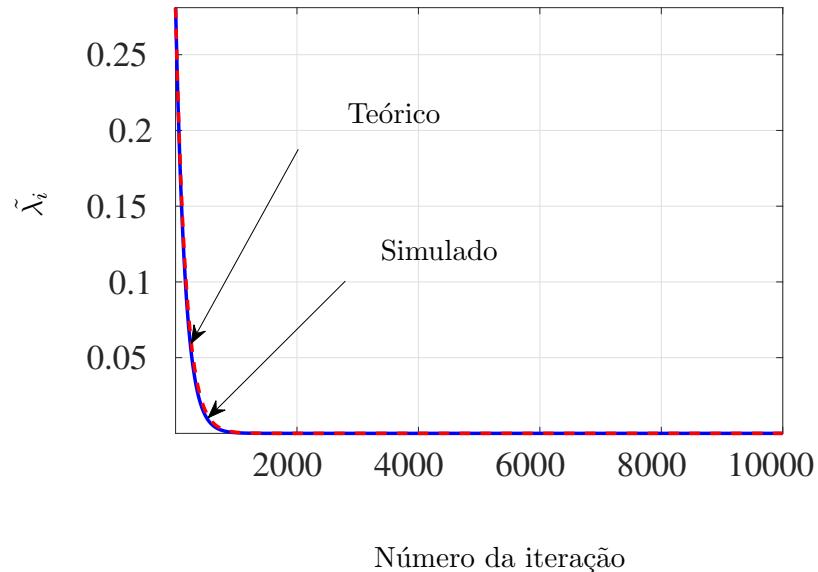


Figura VII.38: Evolução do  $\tilde{\lambda}_i(k)$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$ .

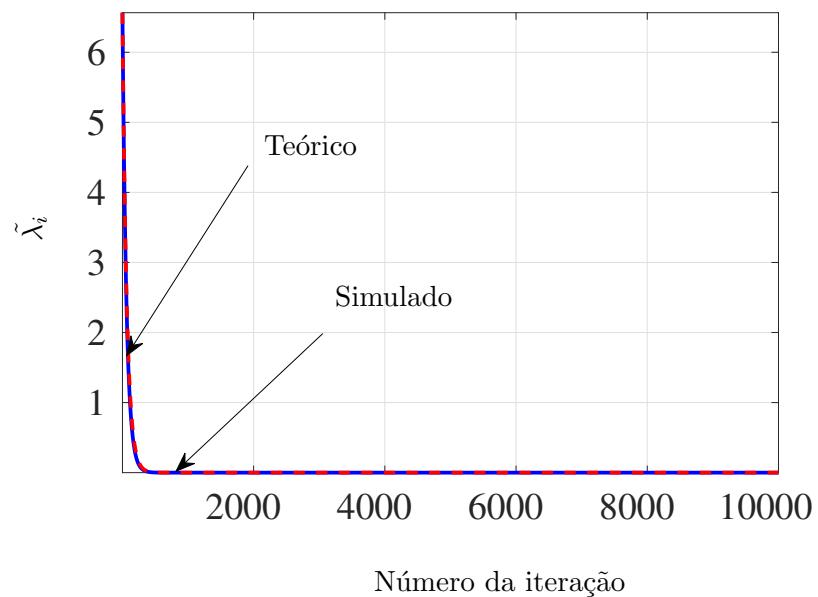


Figura VII.39: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

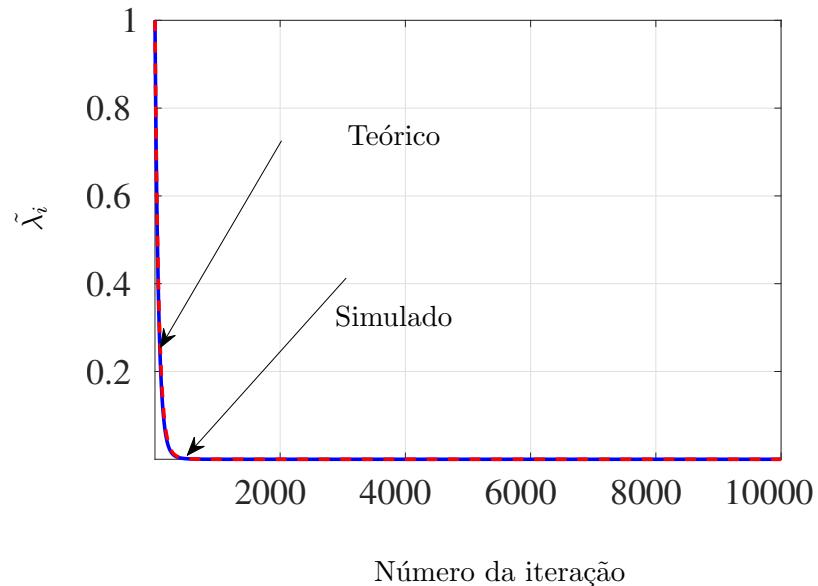


Figura VII.40: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

$L = 9$

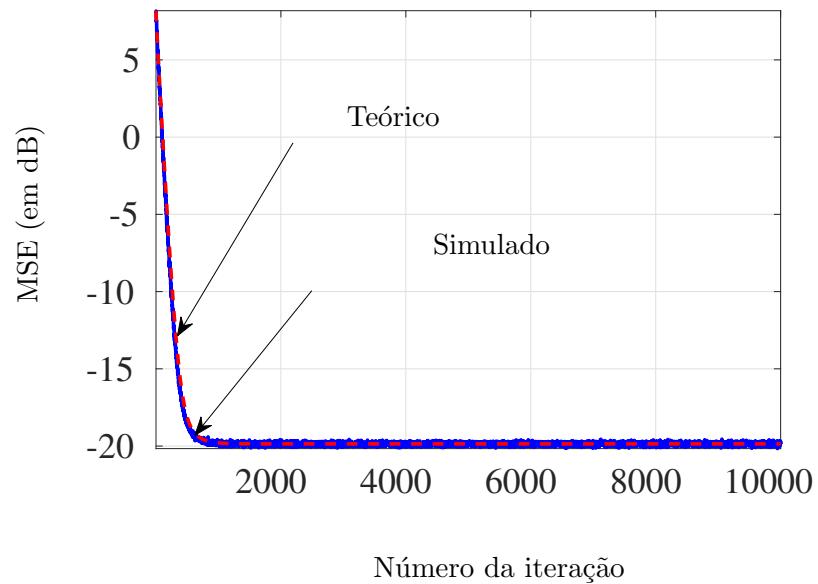


Figura VII.41: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

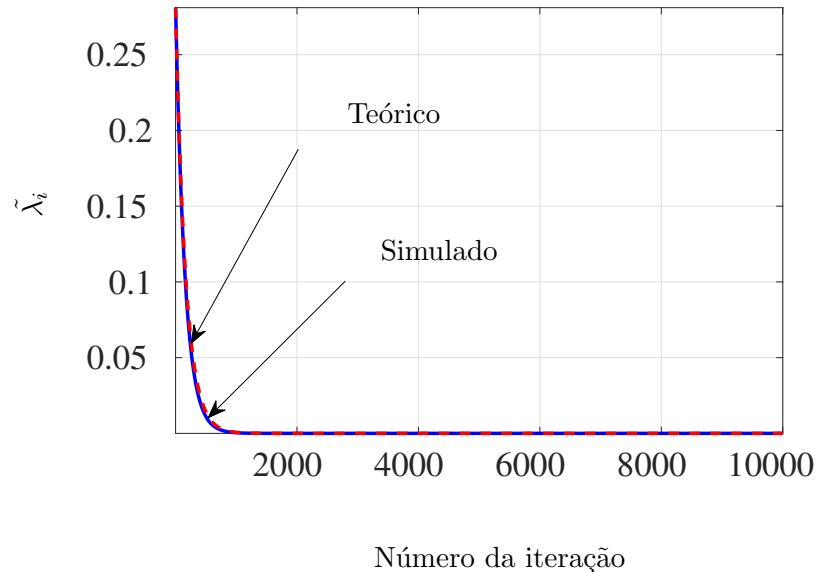


Figura VII.42: Evolução do  $\tilde{\lambda}_i(k)$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$ .

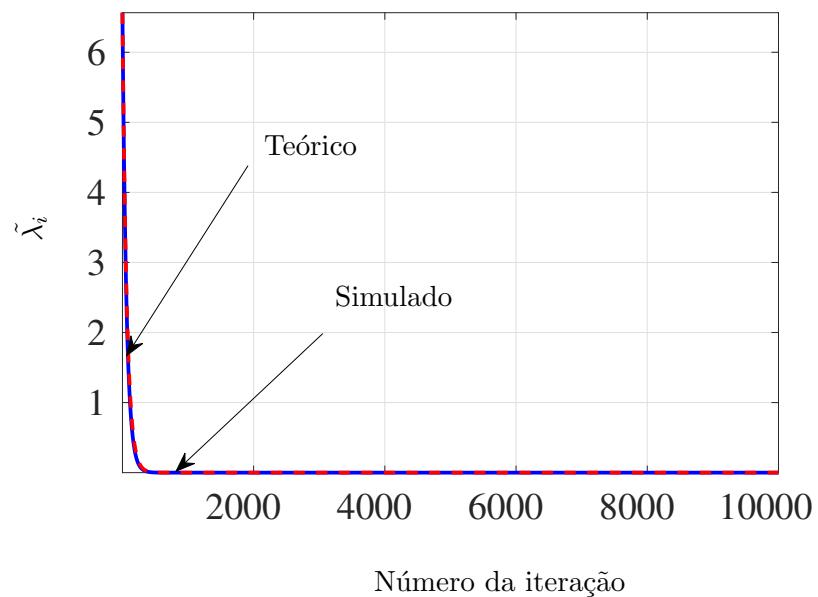


Figura VII.43: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

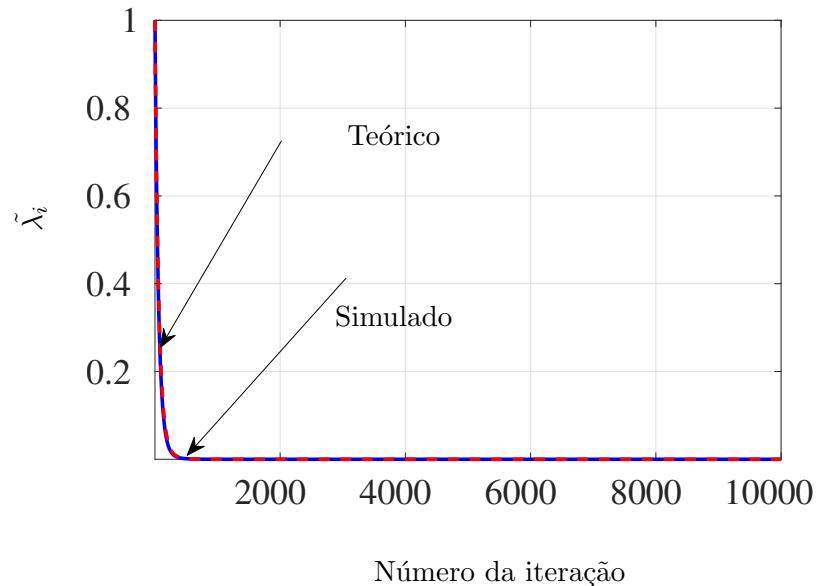


Figura VII.44: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

$L = 10$

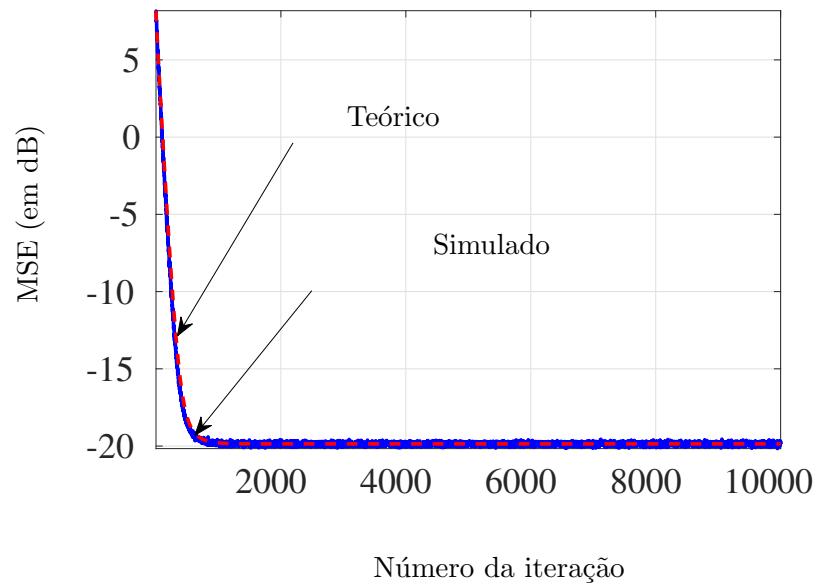


Figura VII.45: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

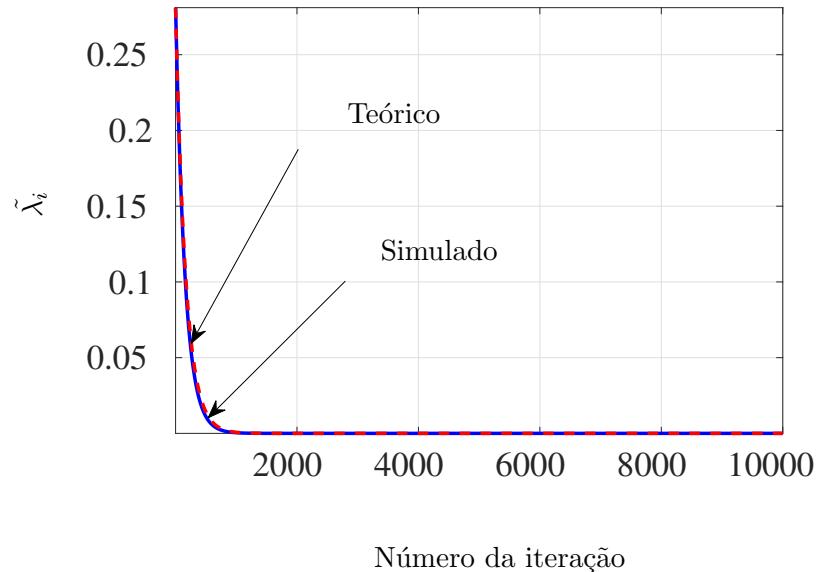


Figura VII.46: Evolução do  $\tilde{\lambda}_i(k)$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$ .

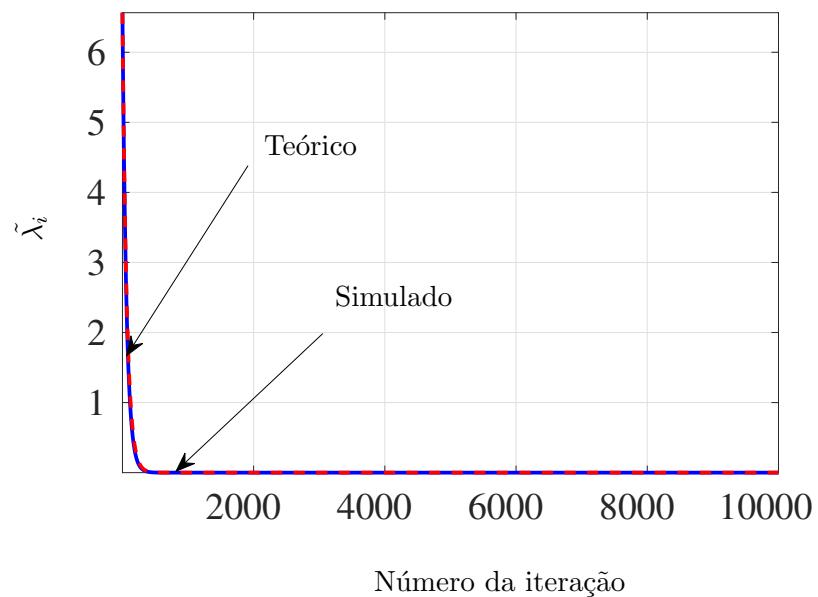


Figura VII.47: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

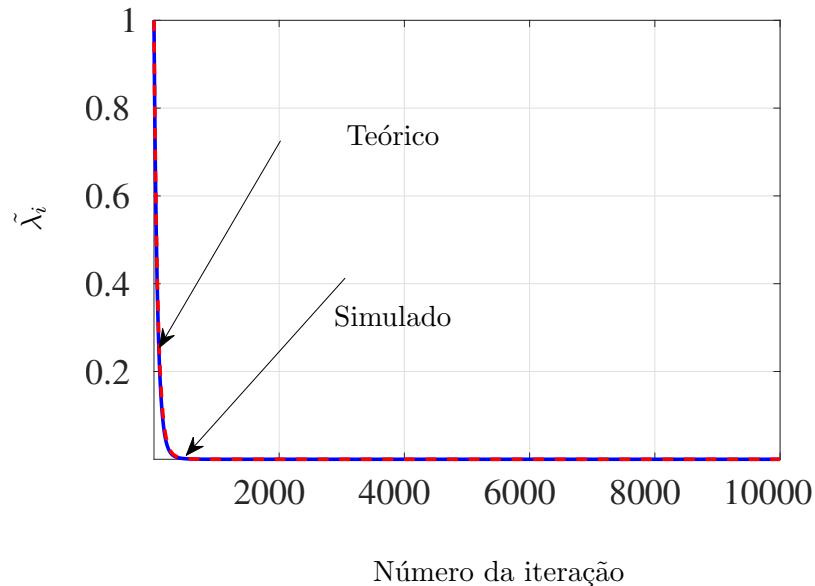


Figura VII.48: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BLMS com  $L = 5$  e  $N = 10$

## BNLMS

**Cenário 1**  $\rightarrow N = 10$

$L = 5$

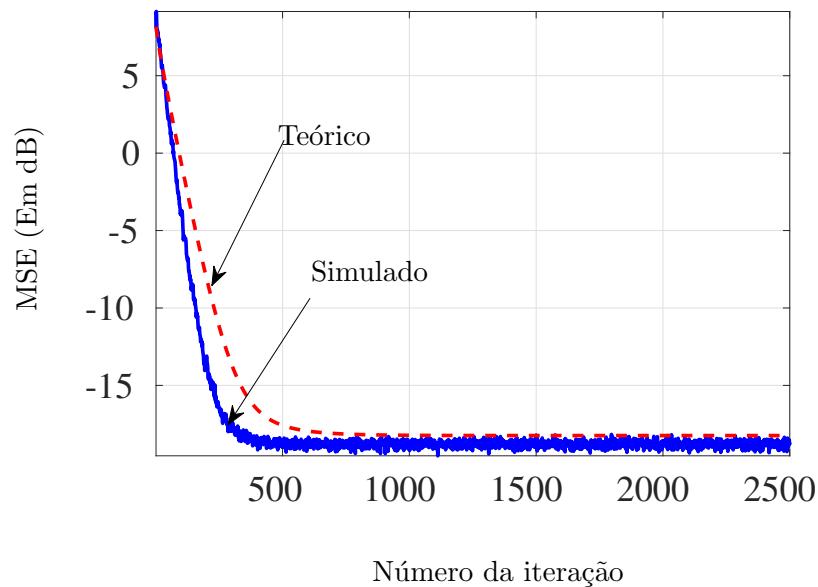


Figura VII.49: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

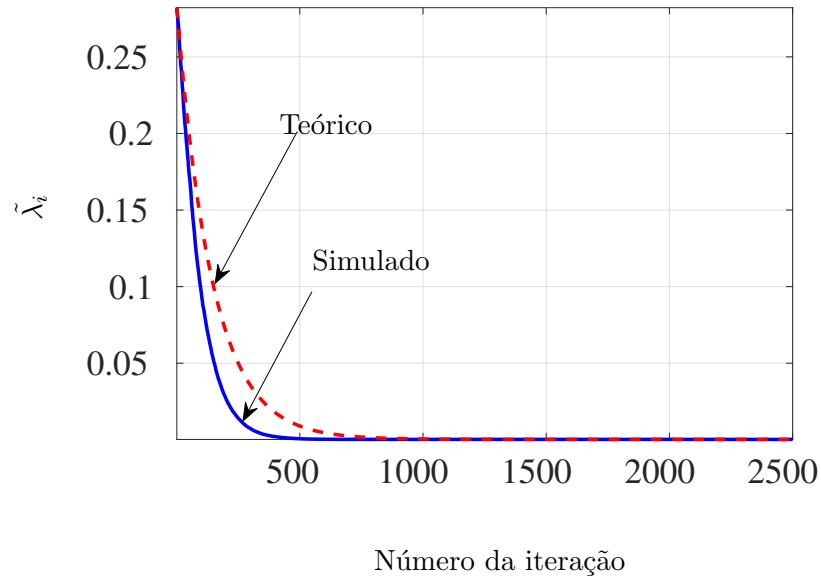


Figura VII.50: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

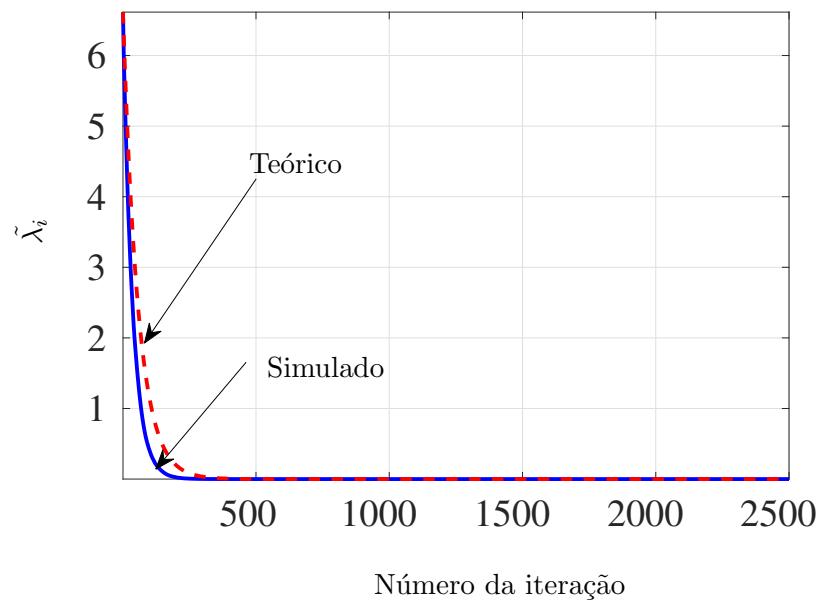


Figura VII.51: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

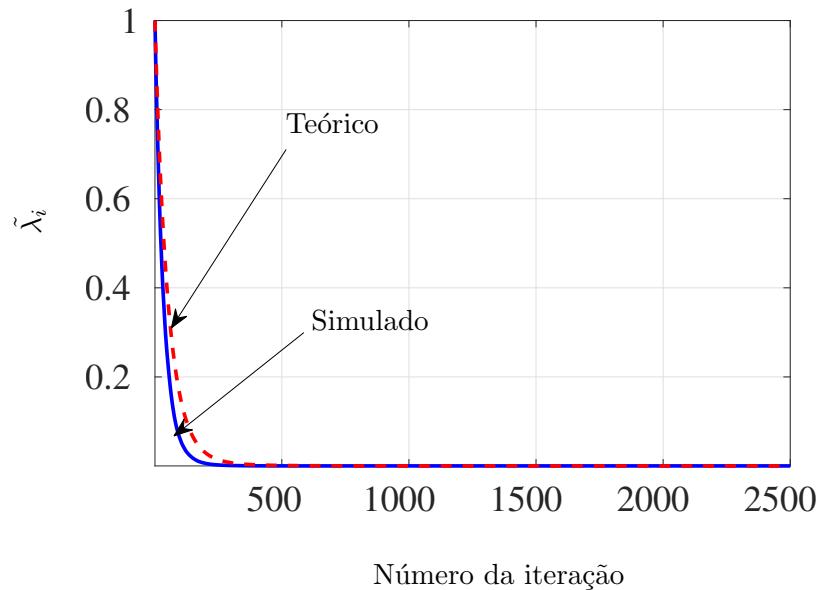


Figura VII.52: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

$L = 6$

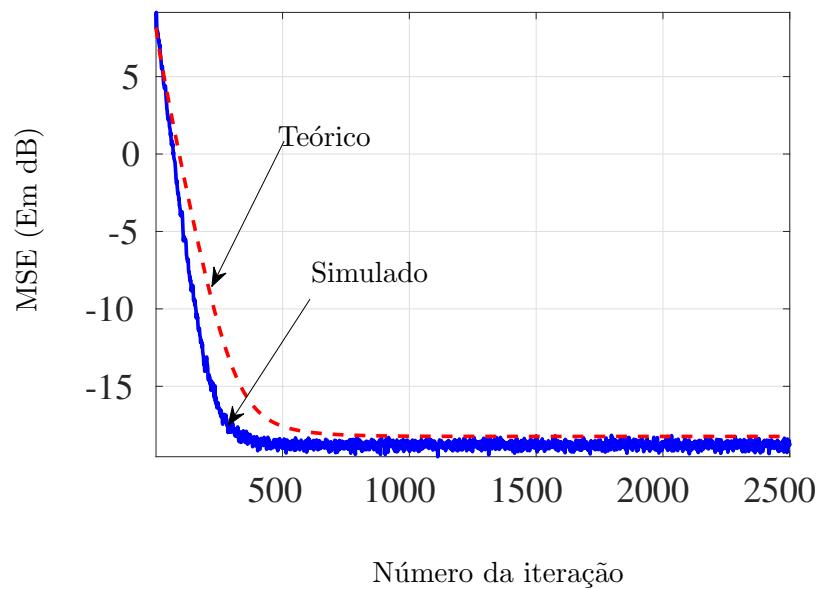


Figura VII.53: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

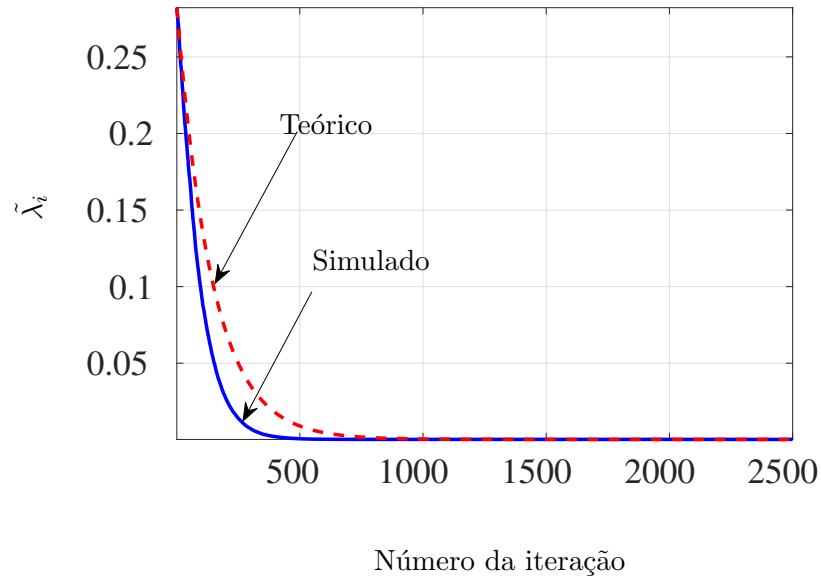


Figura VII.54: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

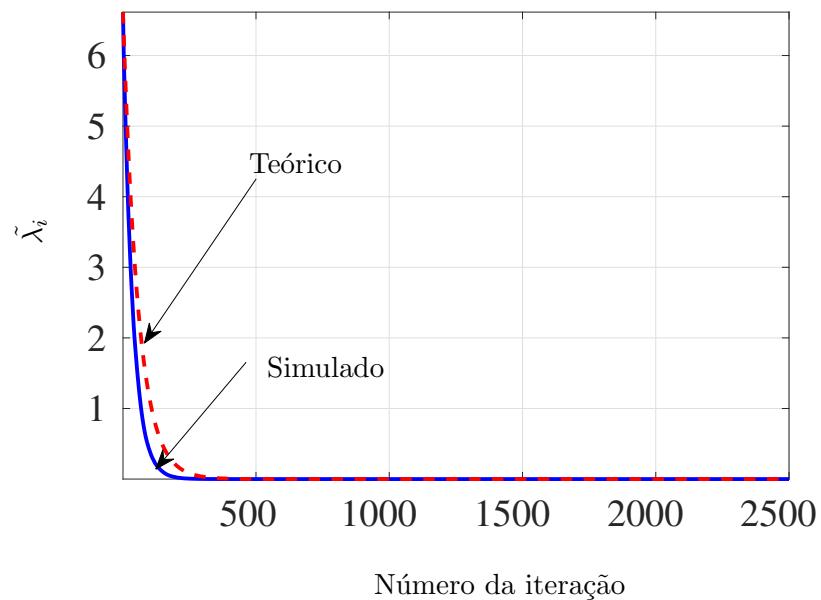


Figura VII.55: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

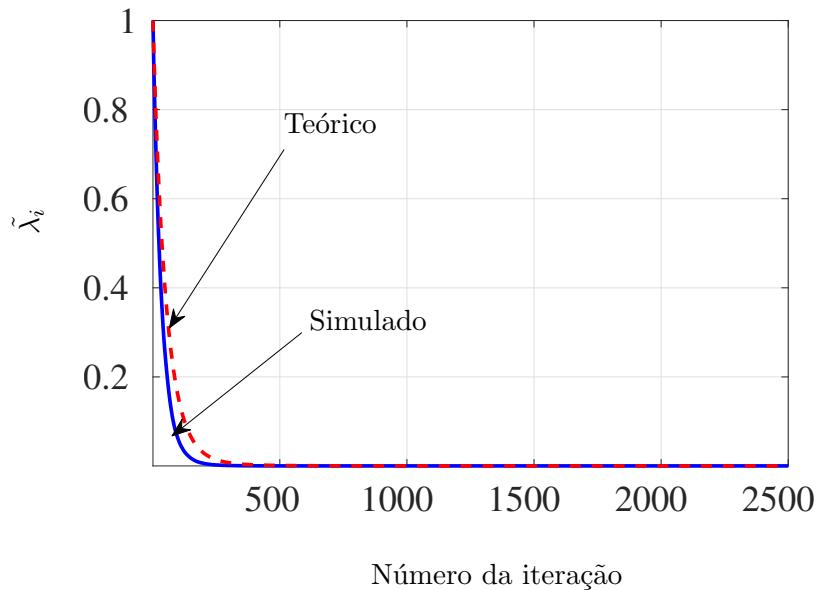


Figura VII.56: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

$L = 7$

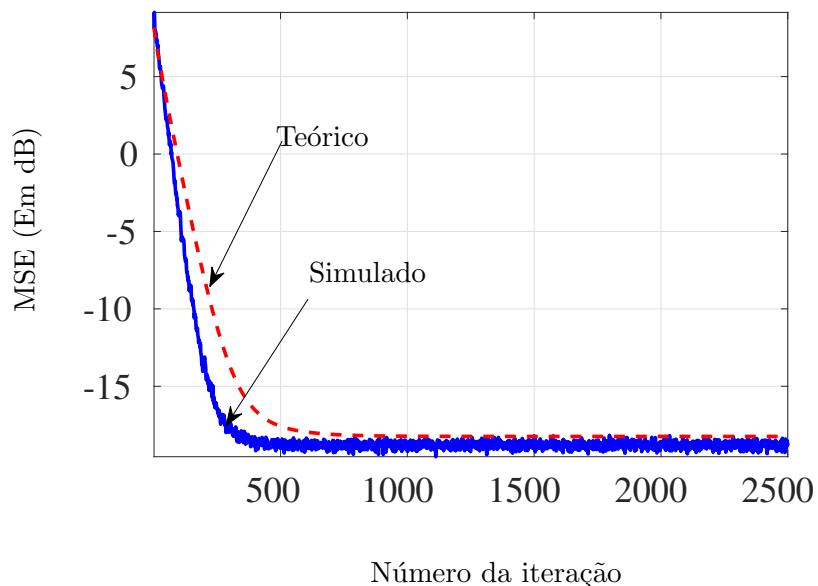


Figura VII.57: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

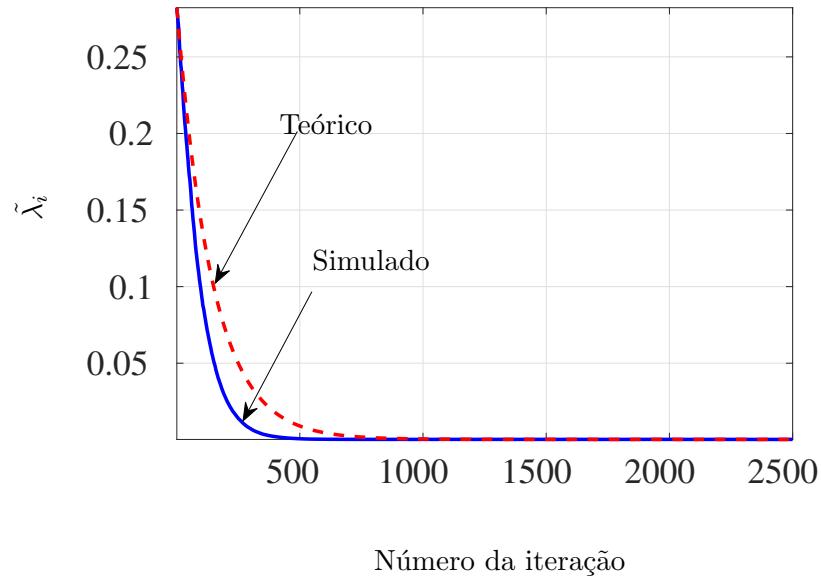


Figura VII.58: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

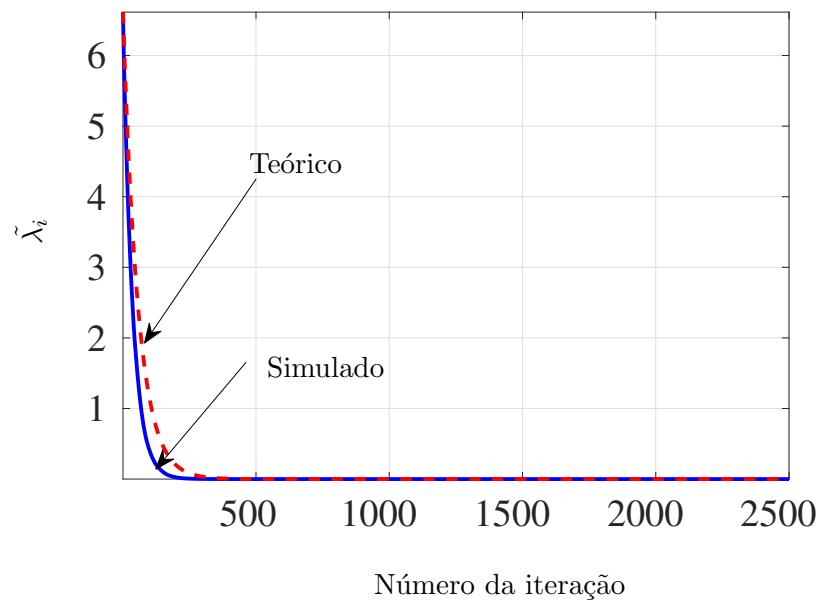


Figura VII.59: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

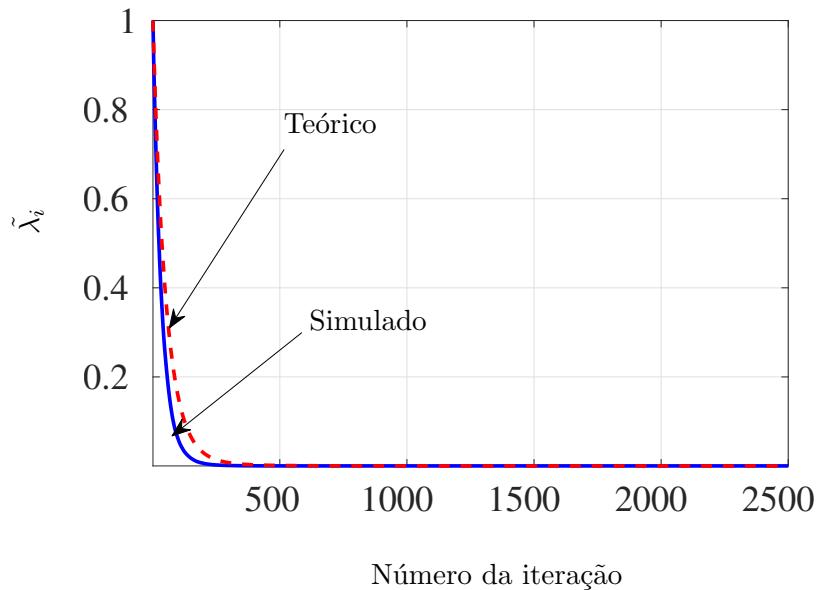


Figura VII.60: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

$L = 8$

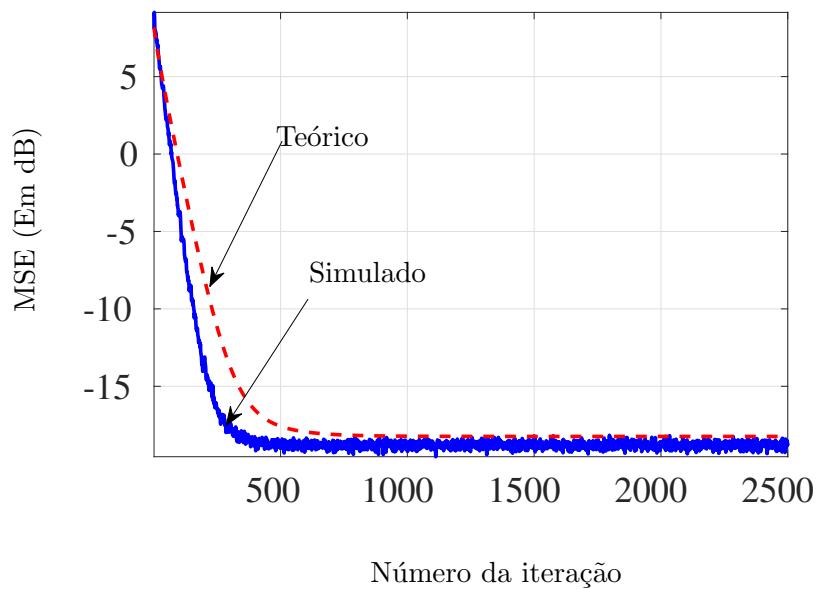


Figura VII.61: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

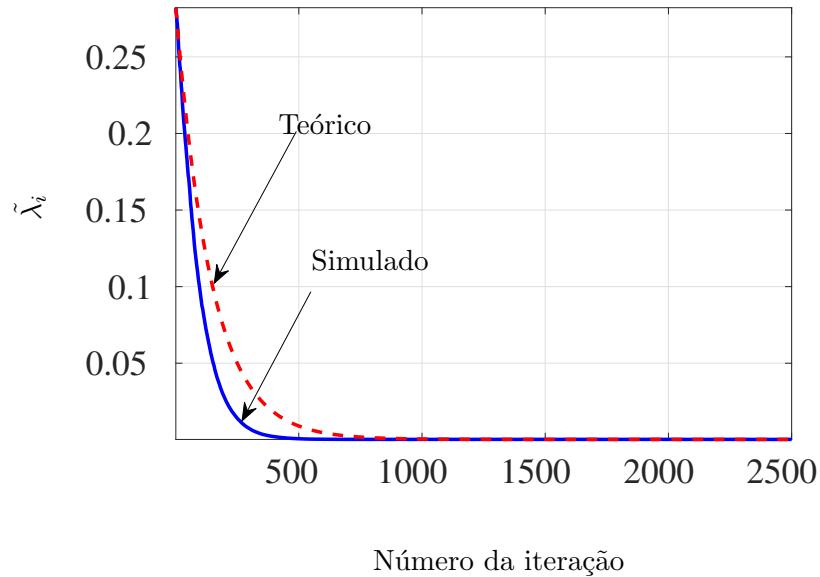


Figura VII.62: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

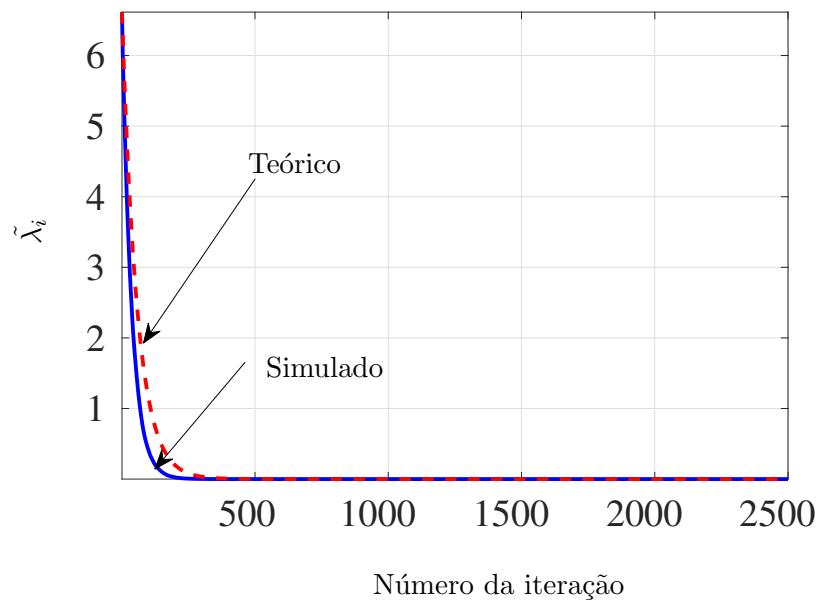


Figura VII.63: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

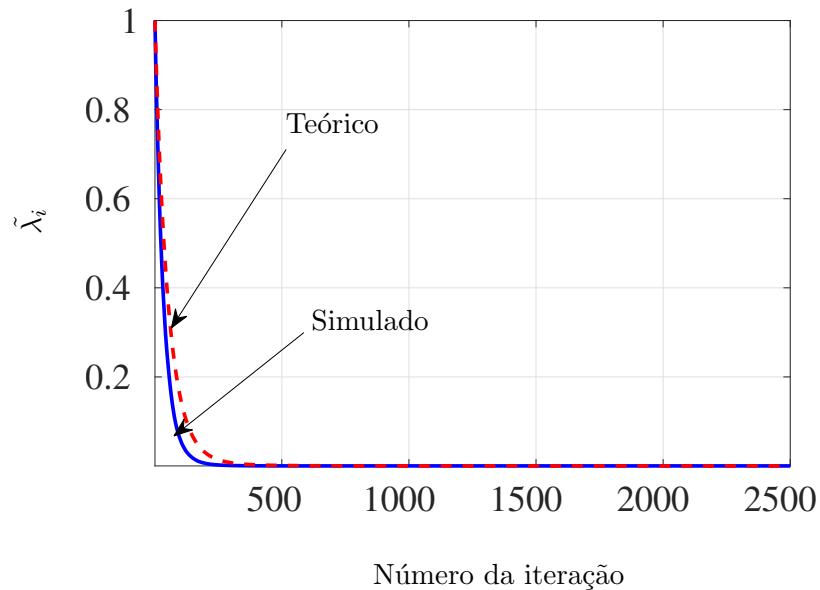


Figura VII.64: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

$L = 9$

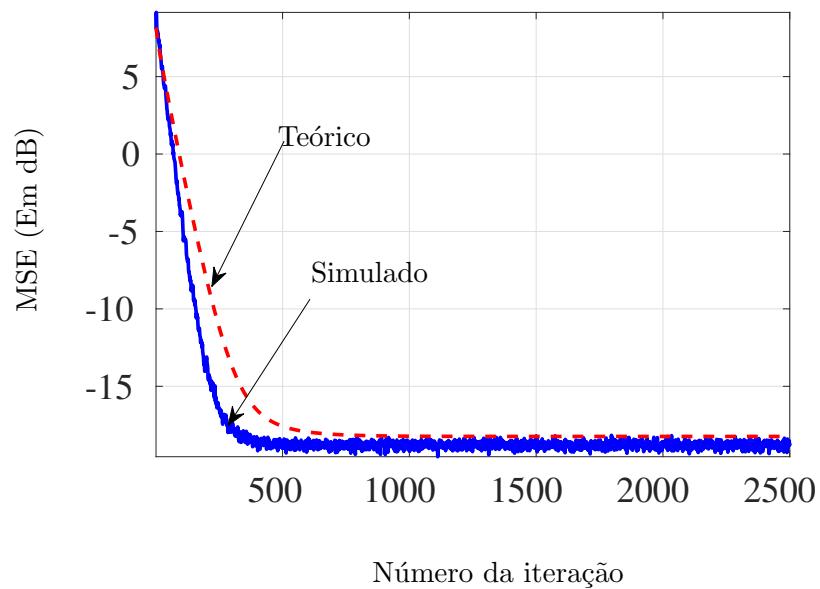


Figura VII.65: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

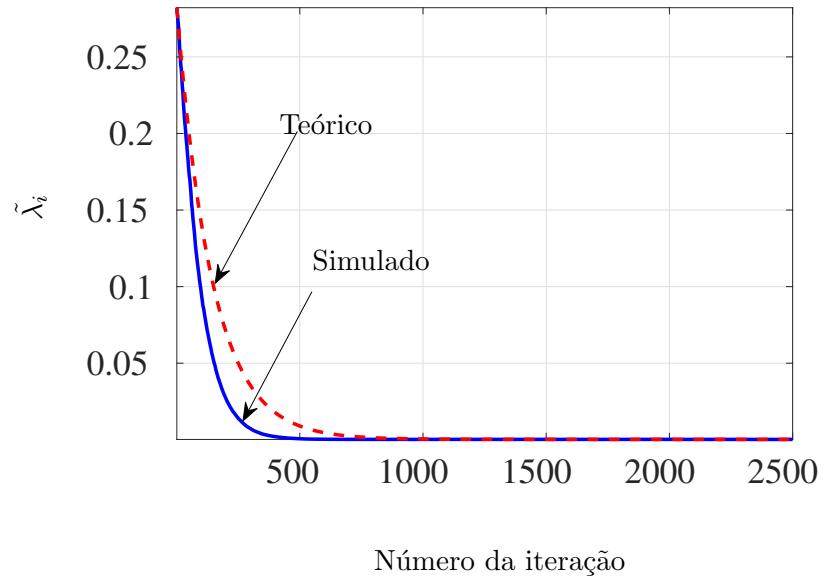


Figura VII.66: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

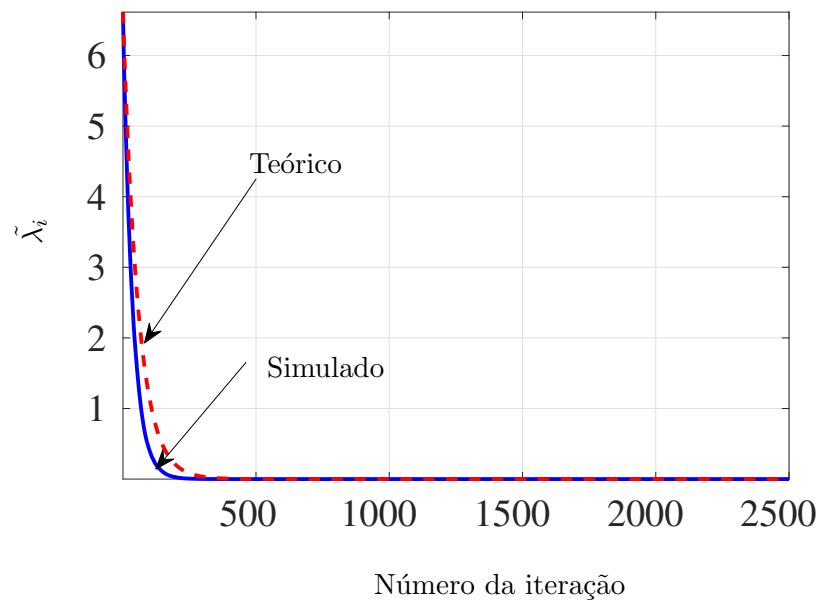


Figura VII.67: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

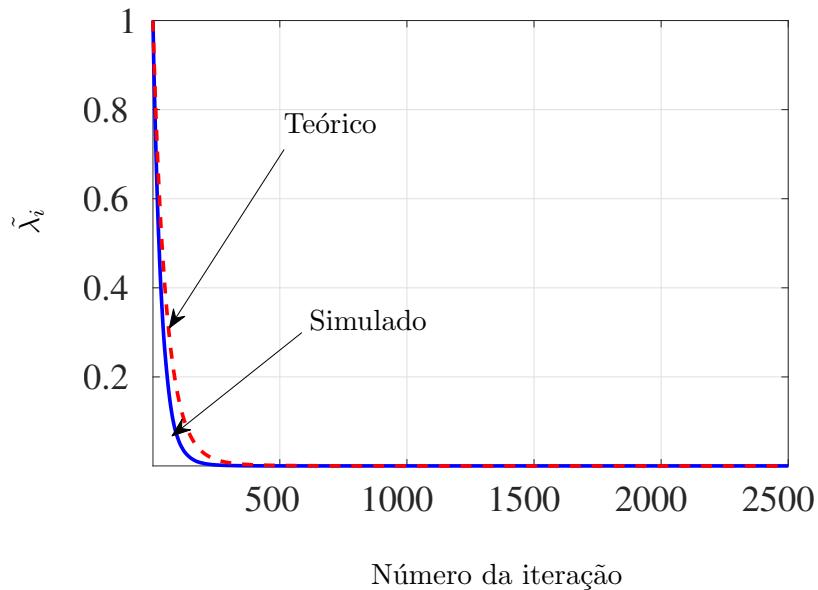


Figura VII.68: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

$L = 10$

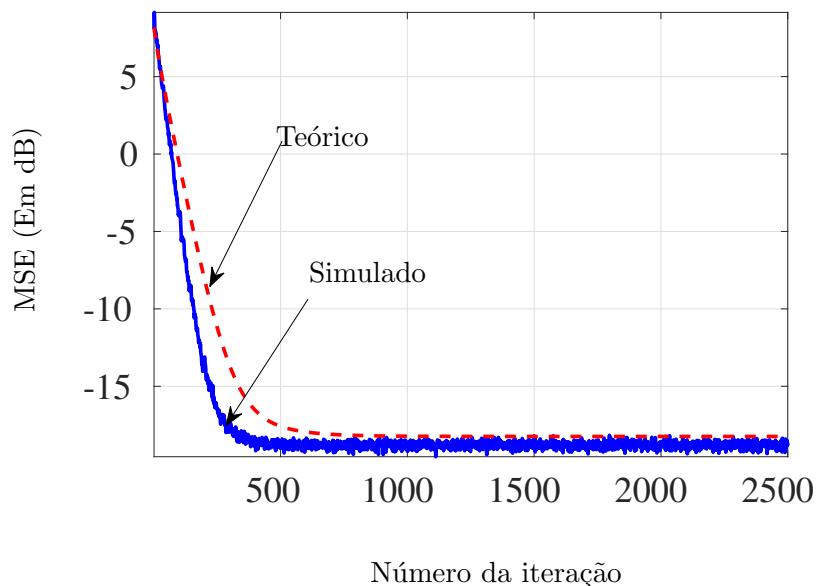


Figura VII.69: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

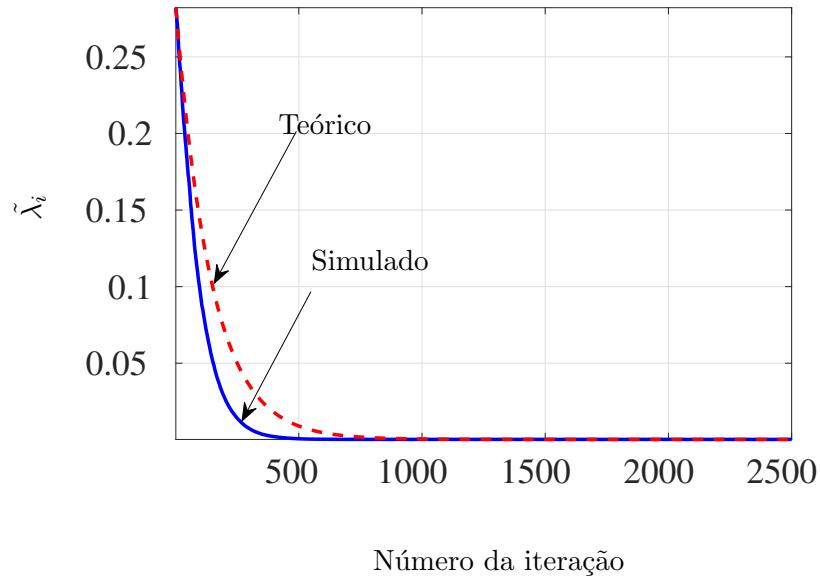


Figura VII.70: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

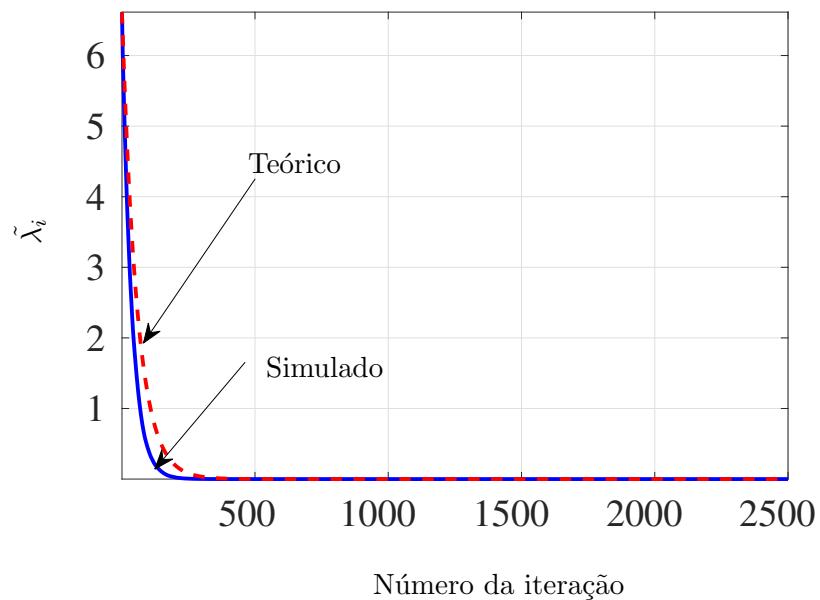


Figura VII.71: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

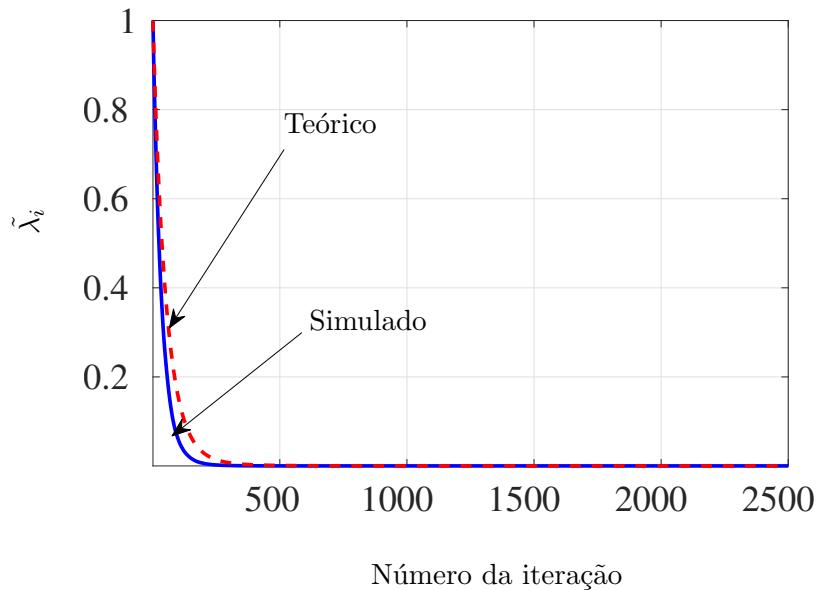


Figura VII.72: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

**Cenário 2** →  $N = 60$

$L = 5$

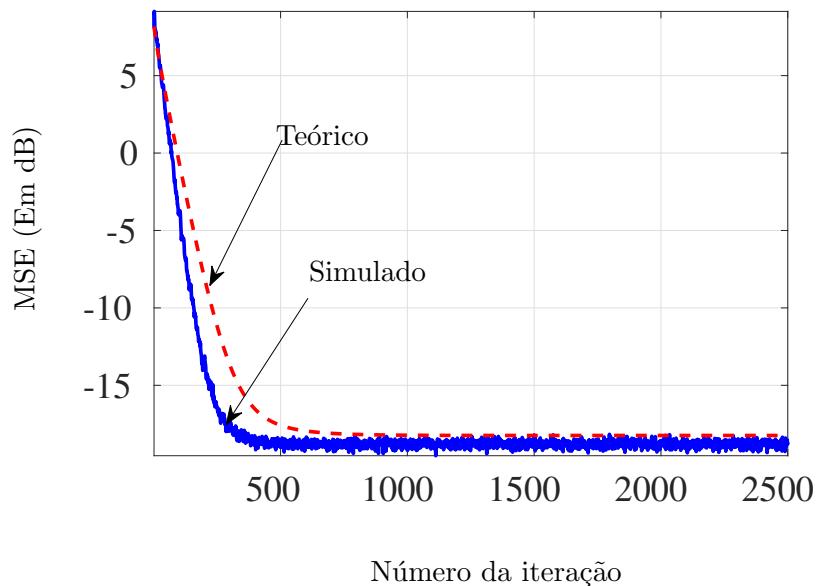


Figura VII.73: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

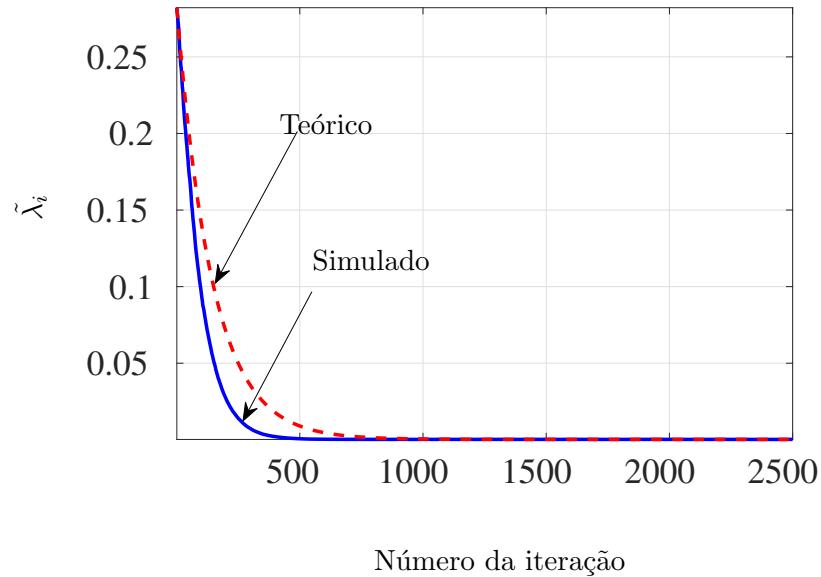


Figura VII.74: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

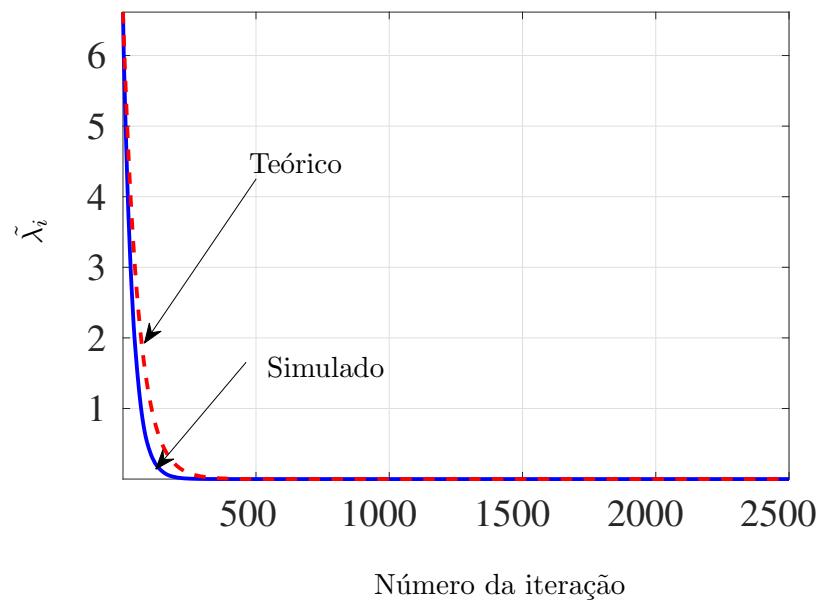


Figura VII.75: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

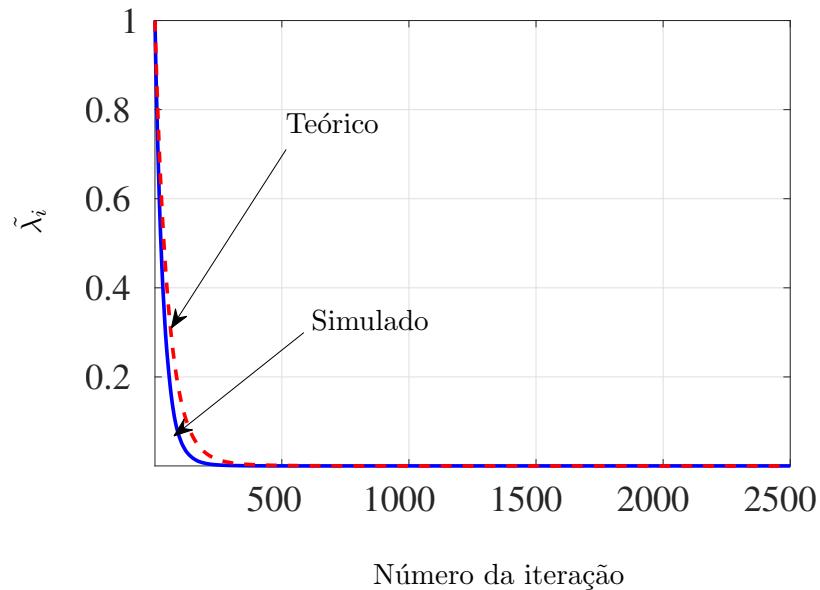


Figura VII.76: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

$L = 6$

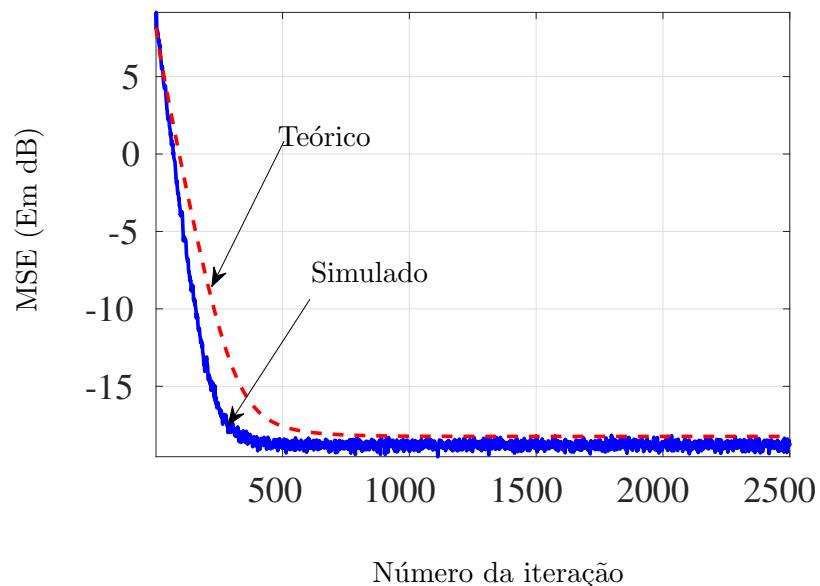


Figura VII.77: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

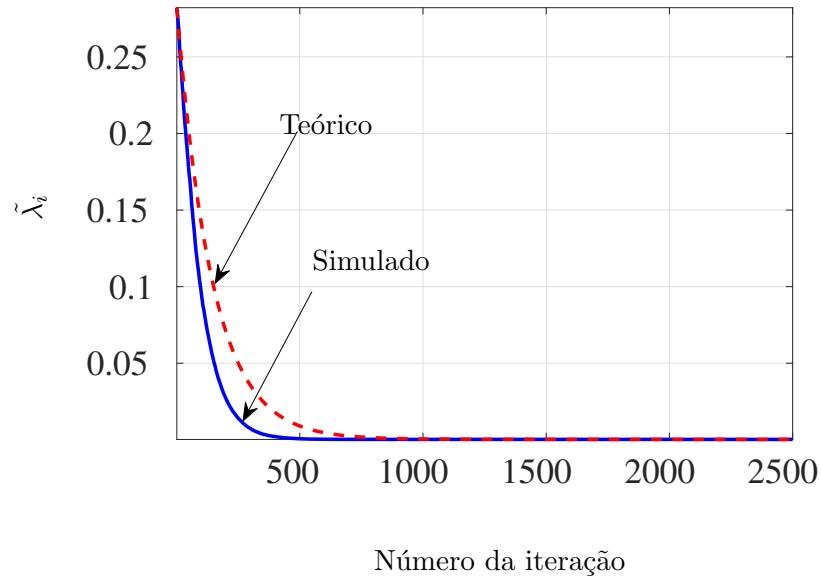


Figura VII.78: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

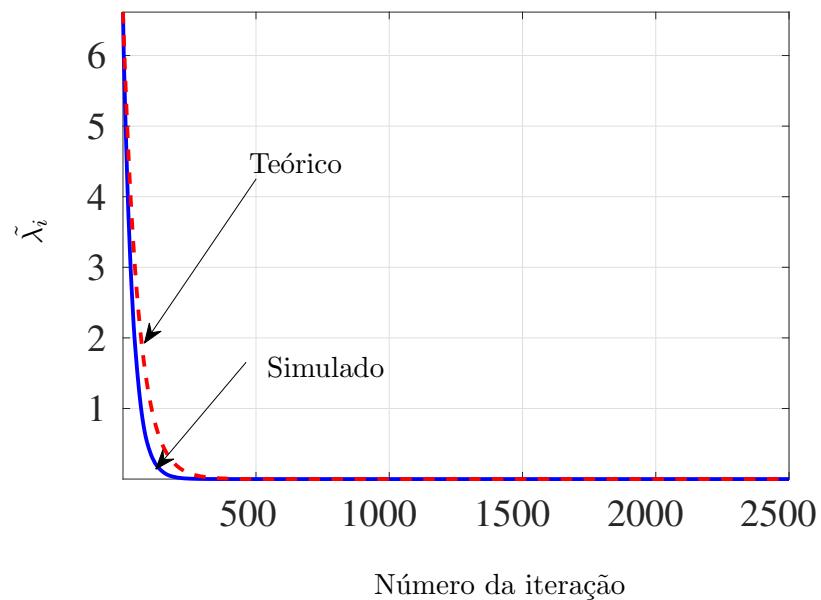


Figura VII.79: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

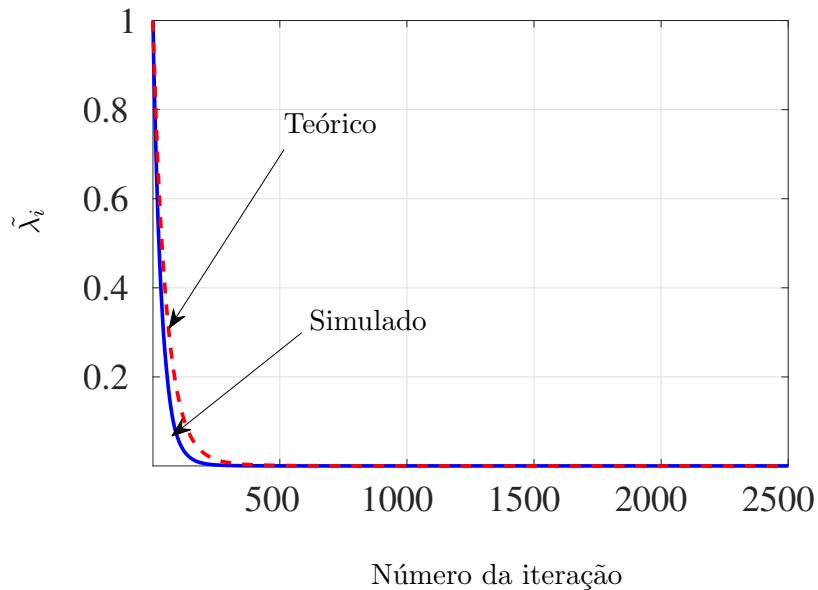


Figura VII.80: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

$L = 7$

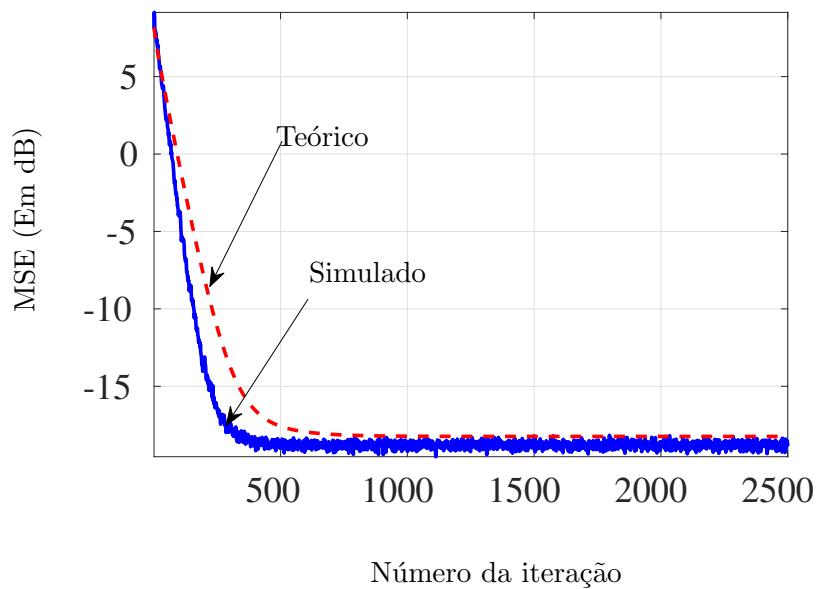


Figura VII.81: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

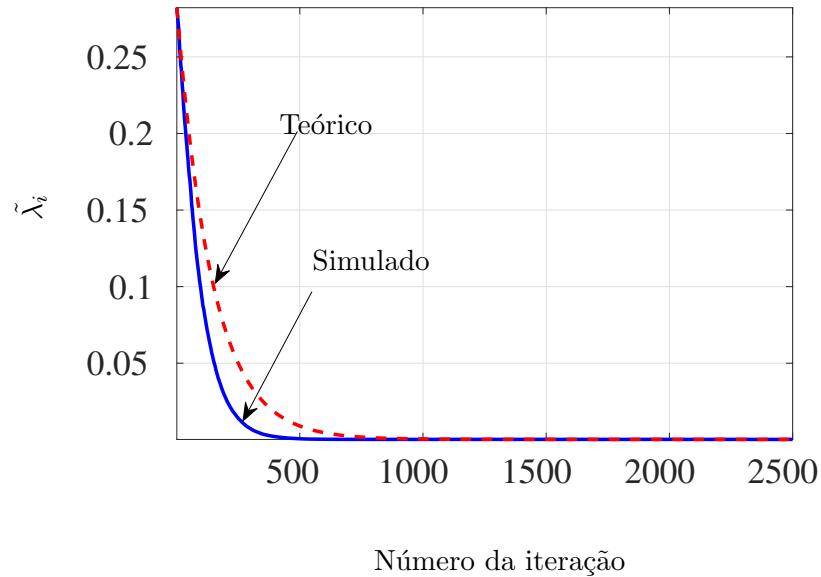


Figura VII.82: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

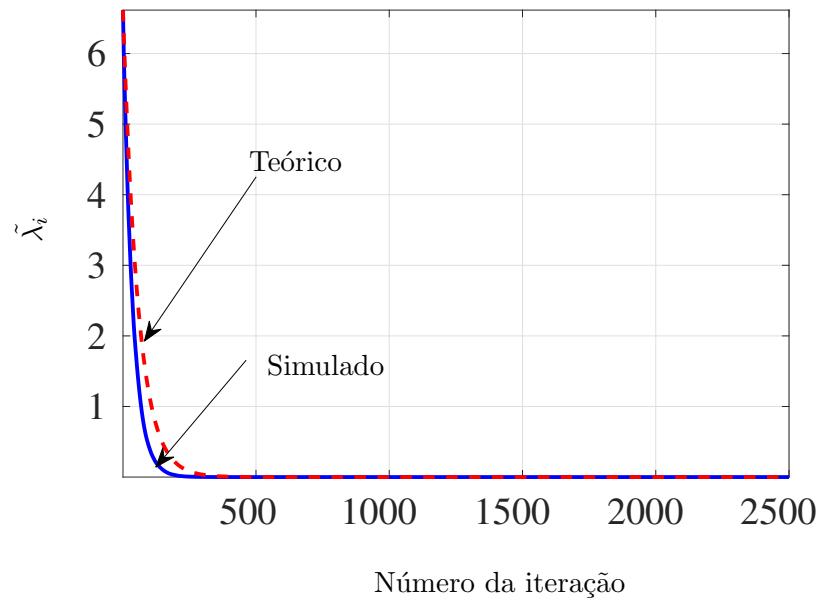


Figura VII.83: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

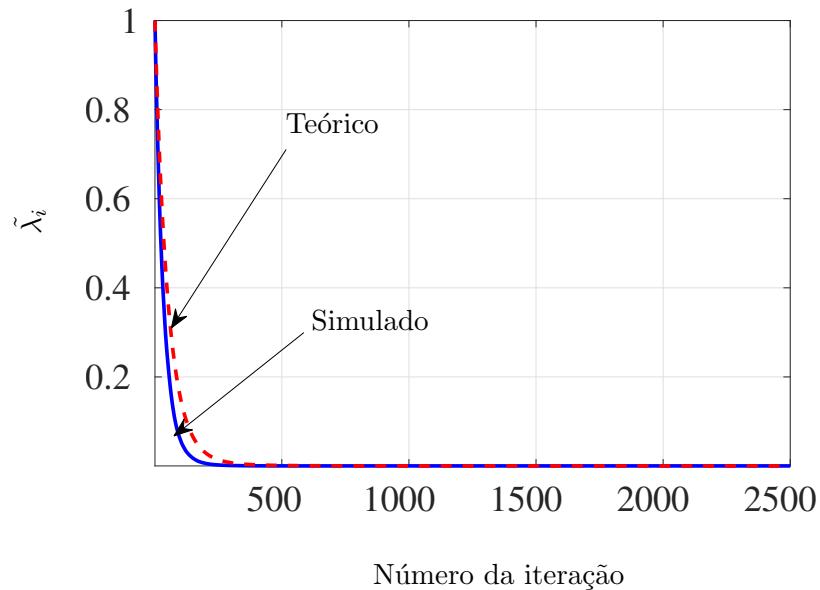


Figura VII.84: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

$L = 8$

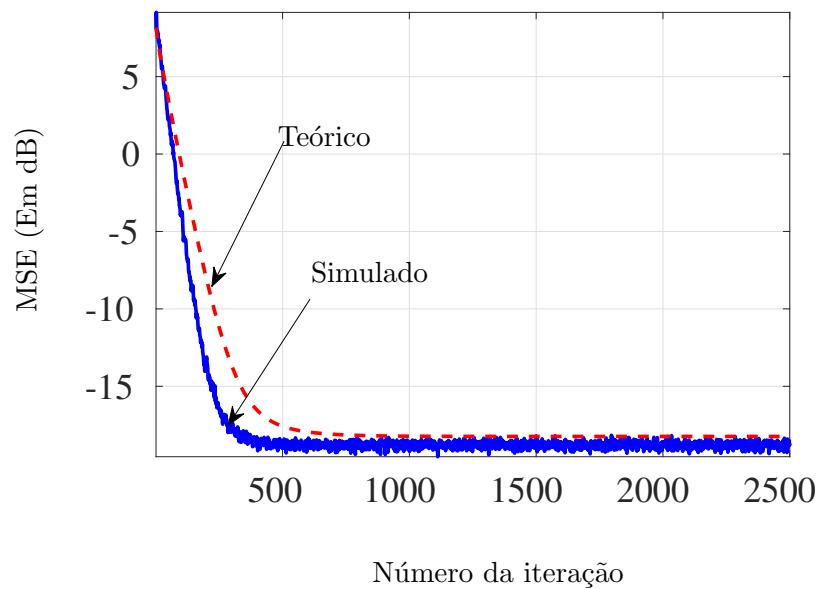


Figura VII.85: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

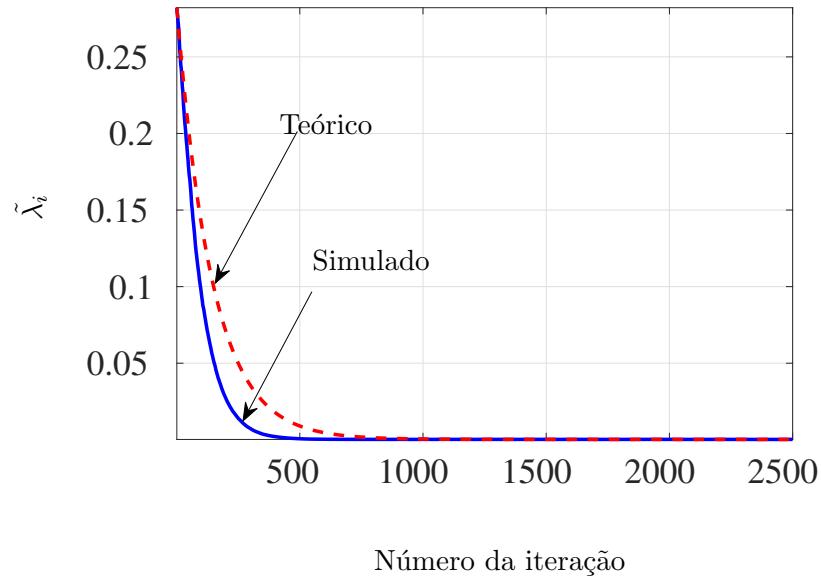


Figura VII.86: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

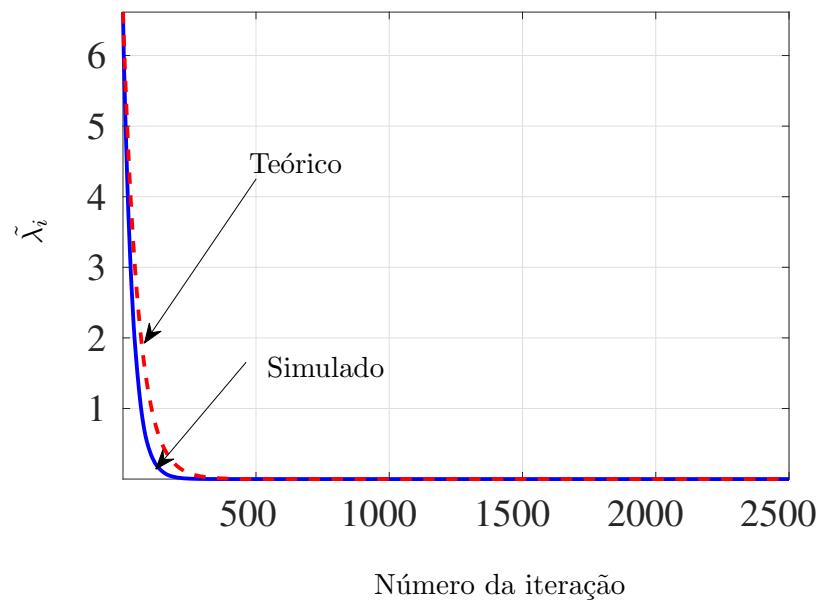


Figura VII.87: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

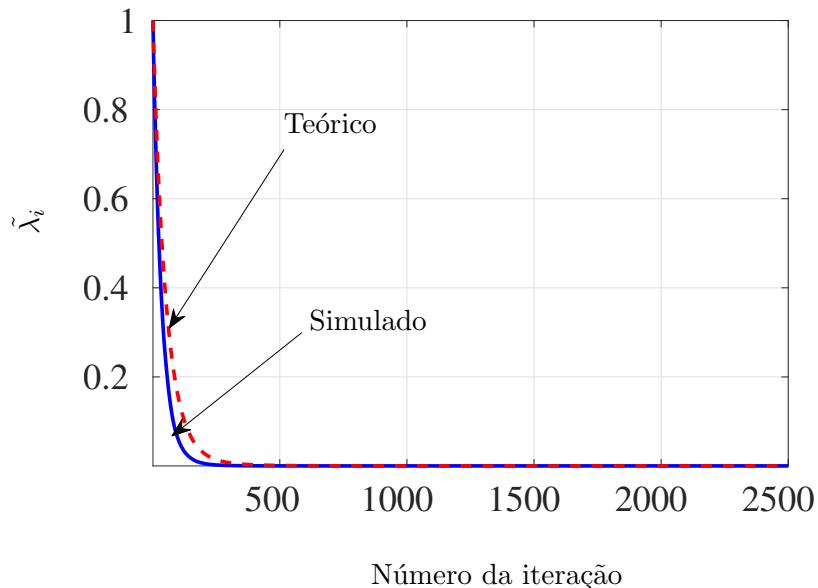


Figura VII.88: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

$L = 9$

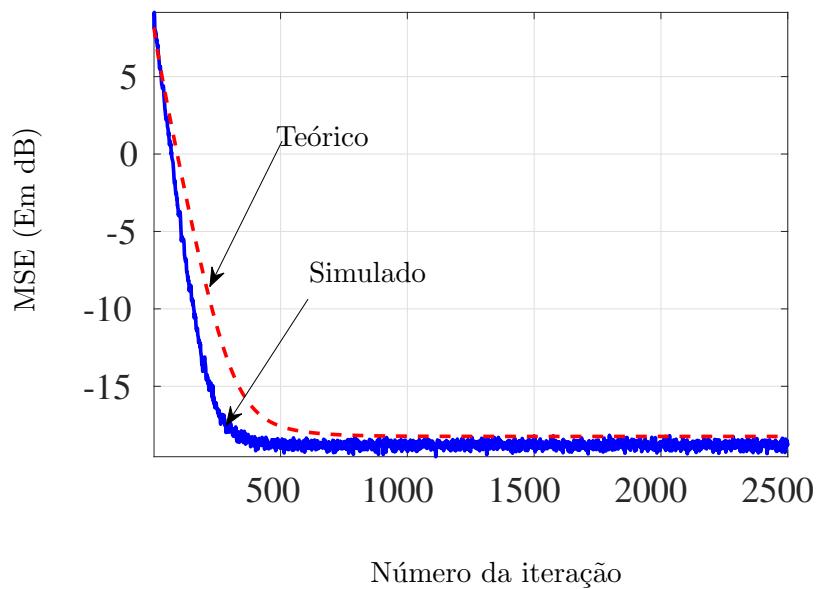


Figura VII.89: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

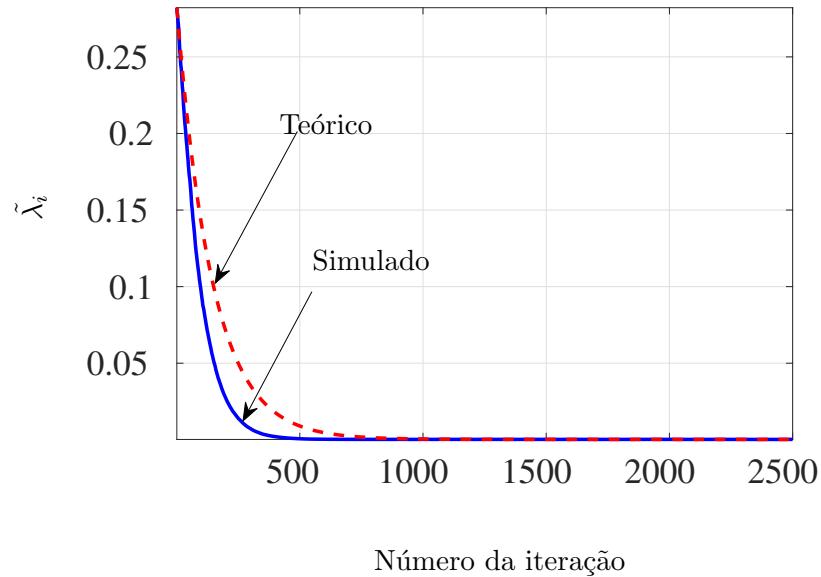


Figura VII.90: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

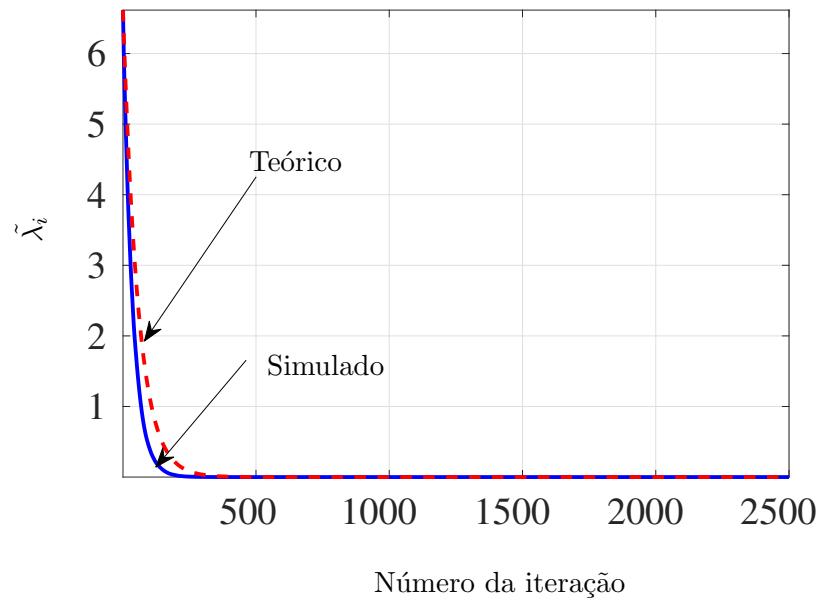


Figura VII.91: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

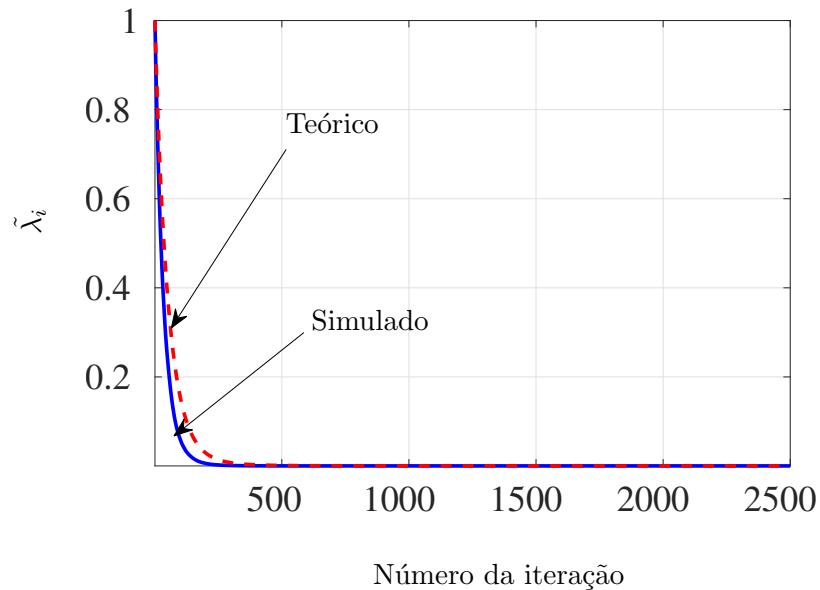


Figura VII.92: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

$L = 10$

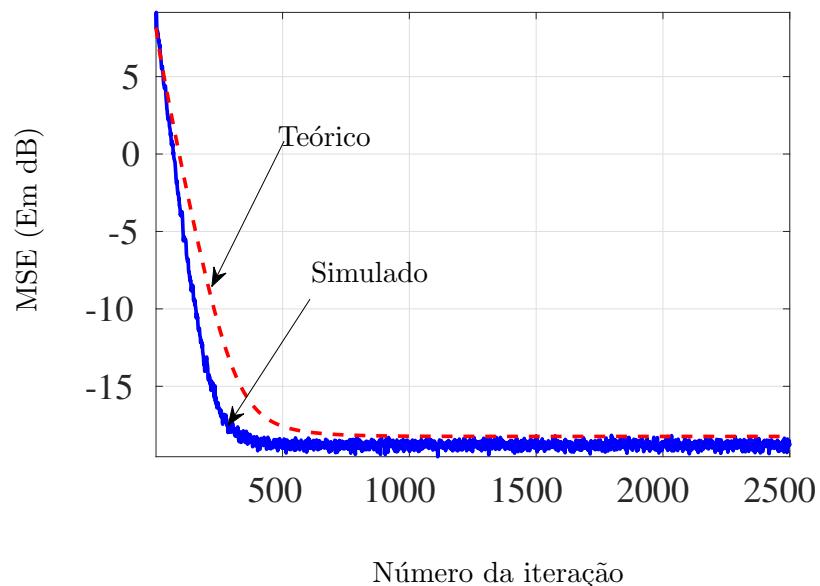


Figura VII.93: Evolução do MSE teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

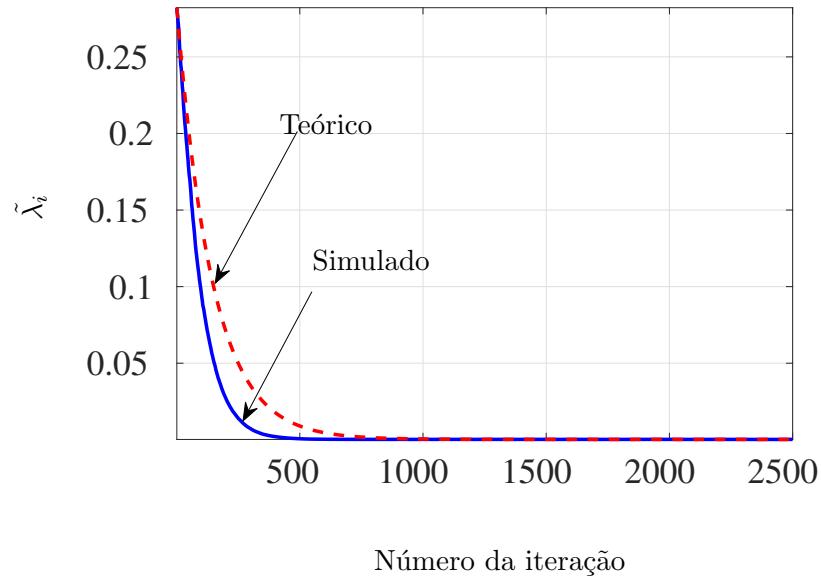


Figura VII.94: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 1$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

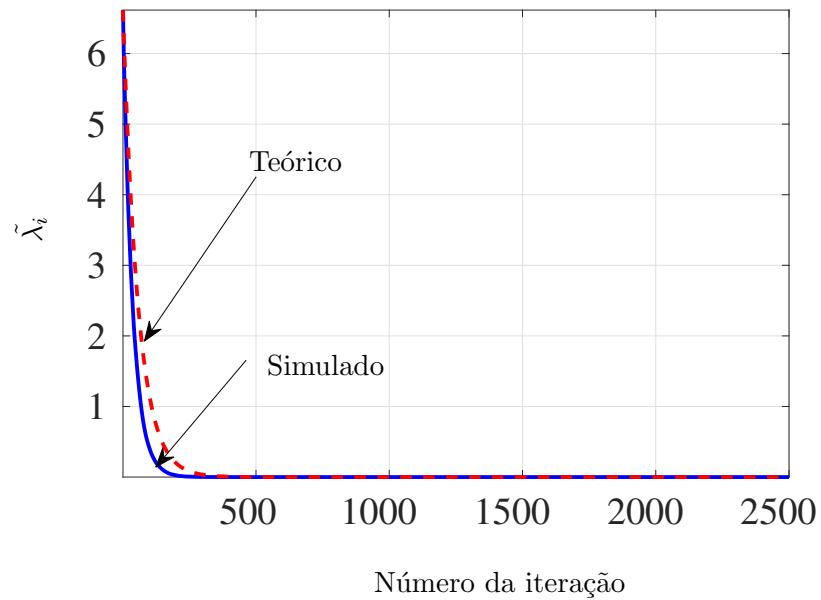


Figura VII.95: Evolução do  $\tilde{\lambda}_i$  teórico e simulado com  $i = 4$  ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

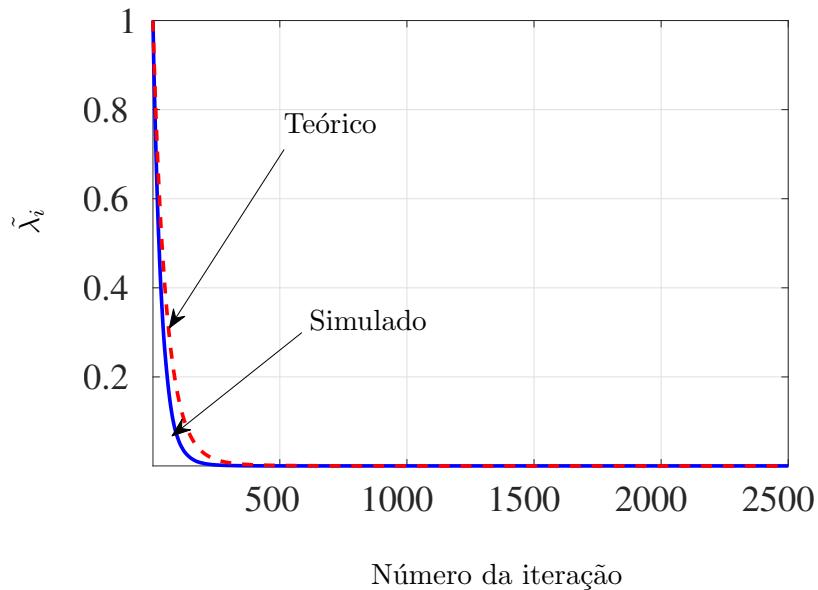


Figura VII.96: Evolução do  $\bar{\lambda}$  médio teórico e simulado ao longo das iterações do algoritmo BNLMS com  $L = 1$  e  $N = 10$

### Simulação 2 - Divergência em função de $\beta$

**BLMS**

**Cenário 1**  $\rightarrow N = 10$

$L = 5$

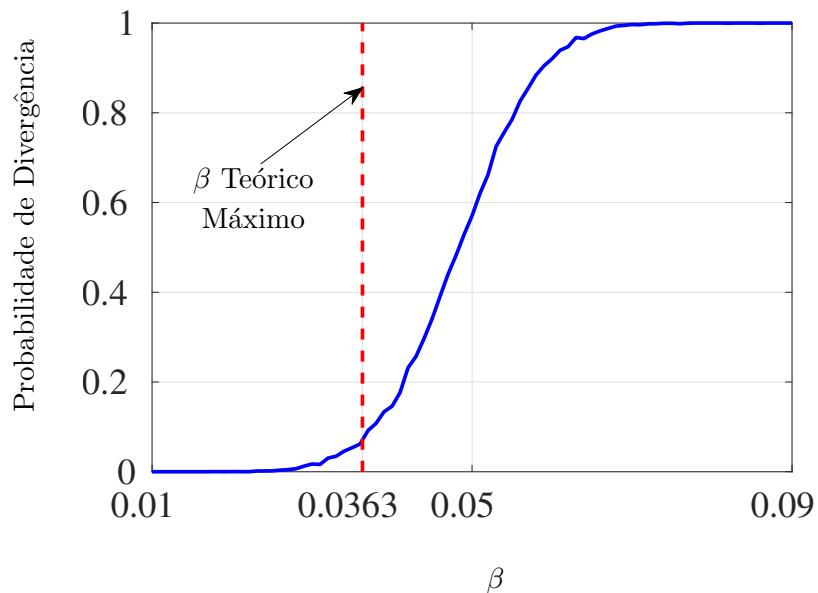


Figura VII.97: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 1$  e  $N = 10$ .

$L = 6$

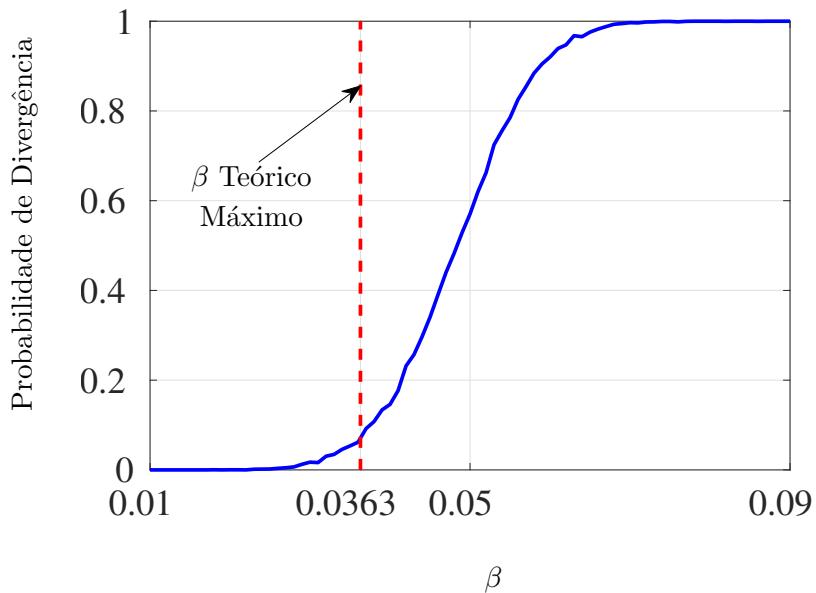


Figura VII.98: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 1$  e  $N = 10$ .

$L = 7$

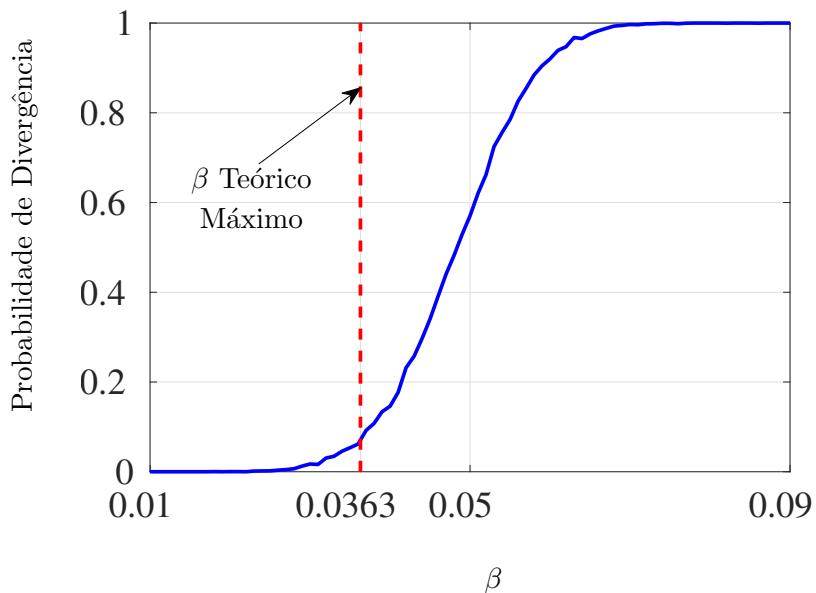


Figura VII.99: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 1$  e  $N = 10$ .

$L = 8$

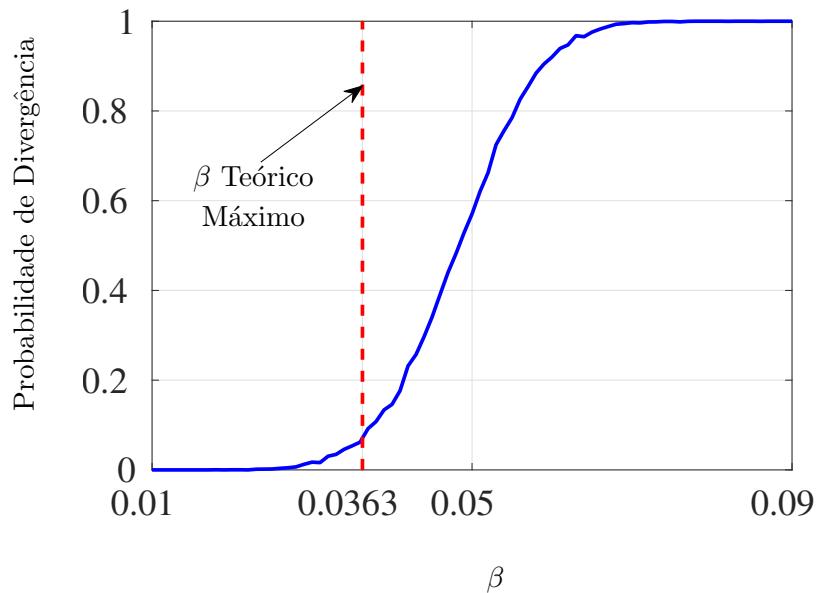


Figura VII.100: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 1$  e  $N = 10$ .

$L = 9$

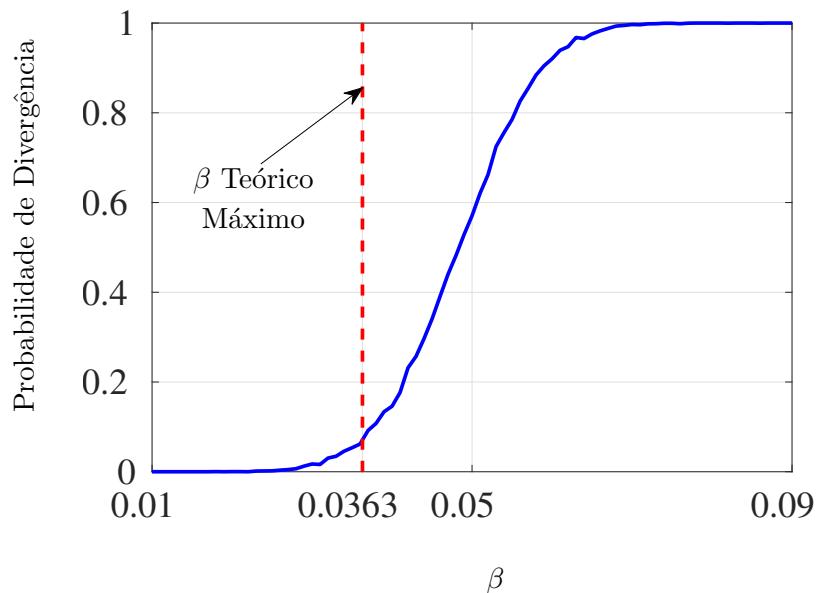


Figura VII.101: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 1$  e  $N = 10$ .

$L = 10$

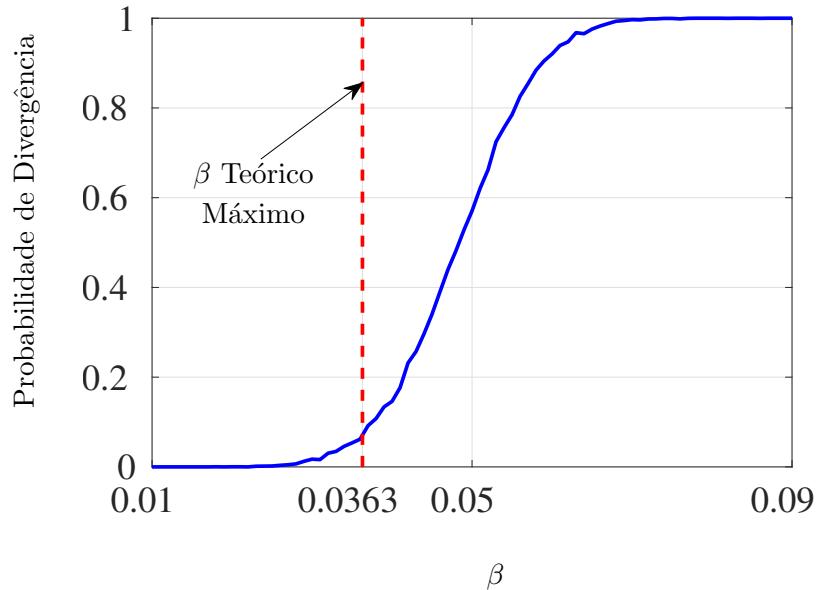


Figura VII.102: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 1$  e  $N = 10$ .

**Cenário 2** →  $N = 60$

$L = 5$

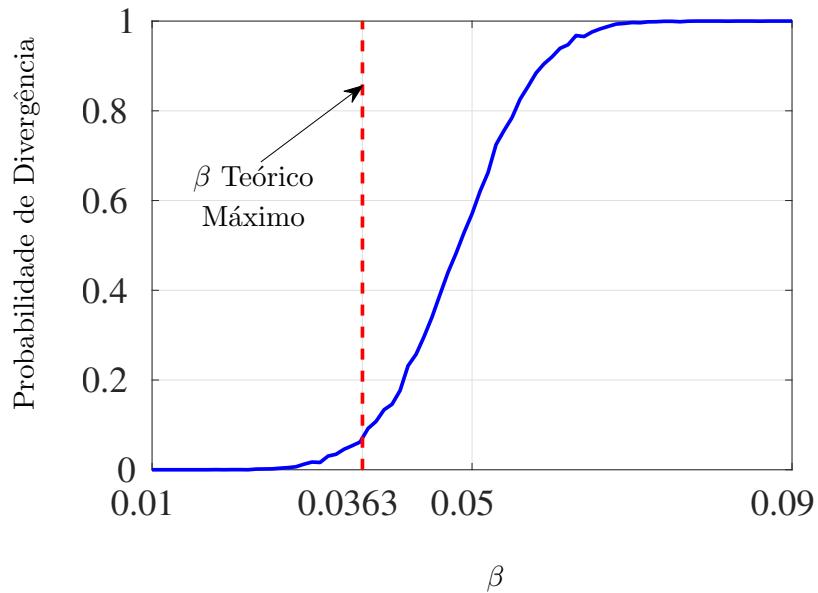


Figura VII.103: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 1$  e  $N = 10$ .

$L = 6$

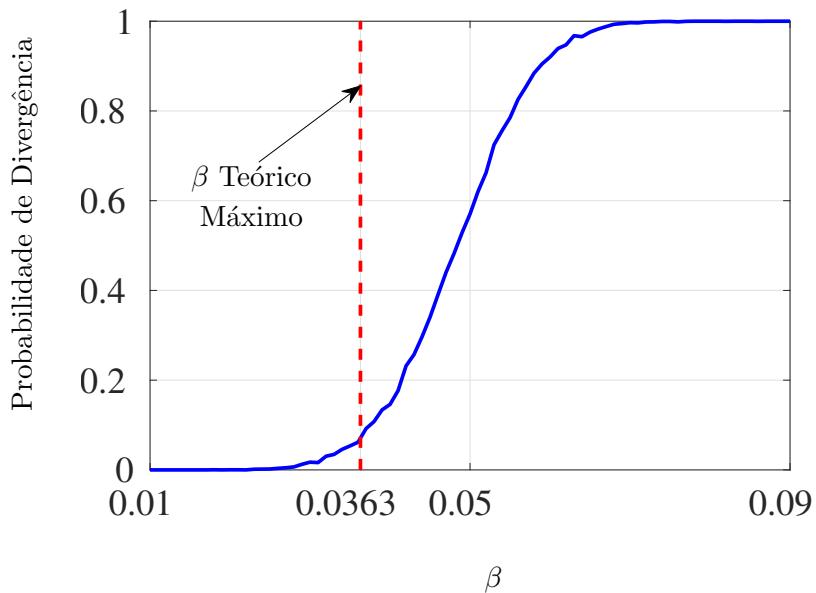


Figura VII.104: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 1$  e  $N = 10$ .

$L = 7$

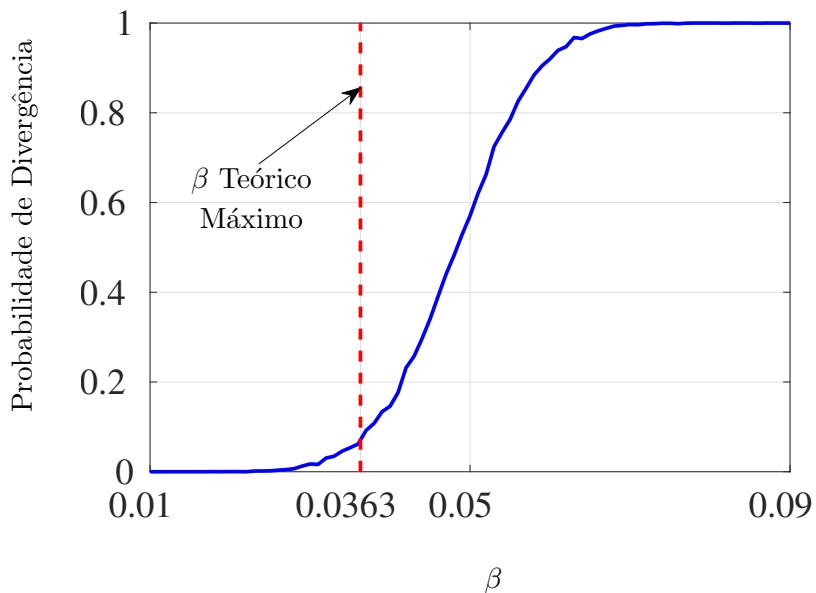


Figura VII.105: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 1$  e  $N = 10$ .

$L = 8$

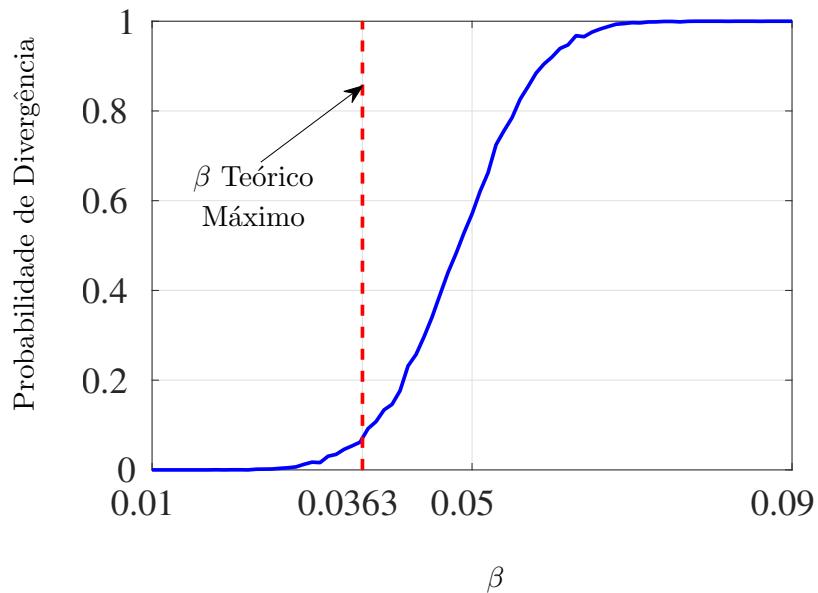


Figura VII.106: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 1$  e  $N = 10$ .

$L = 9$

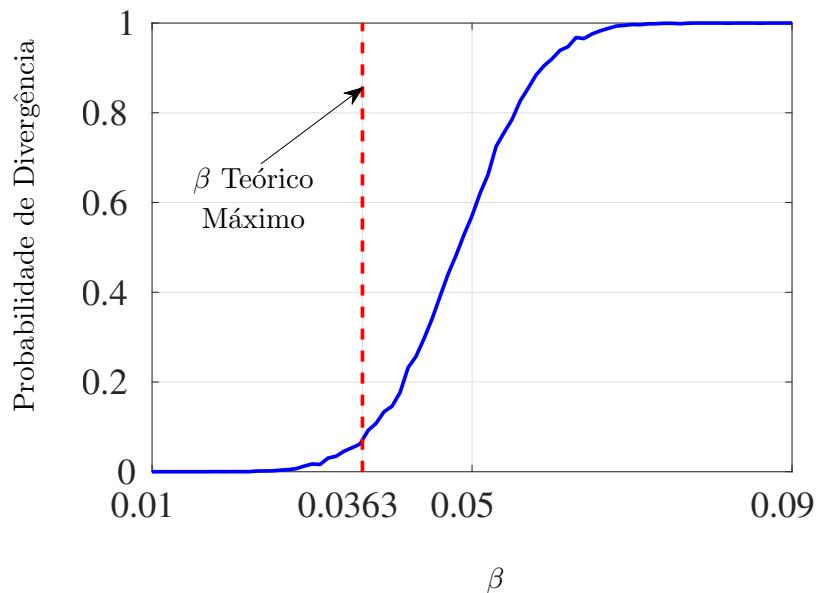


Figura VII.107: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 1$  e  $N = 10$ .

$L = 10$

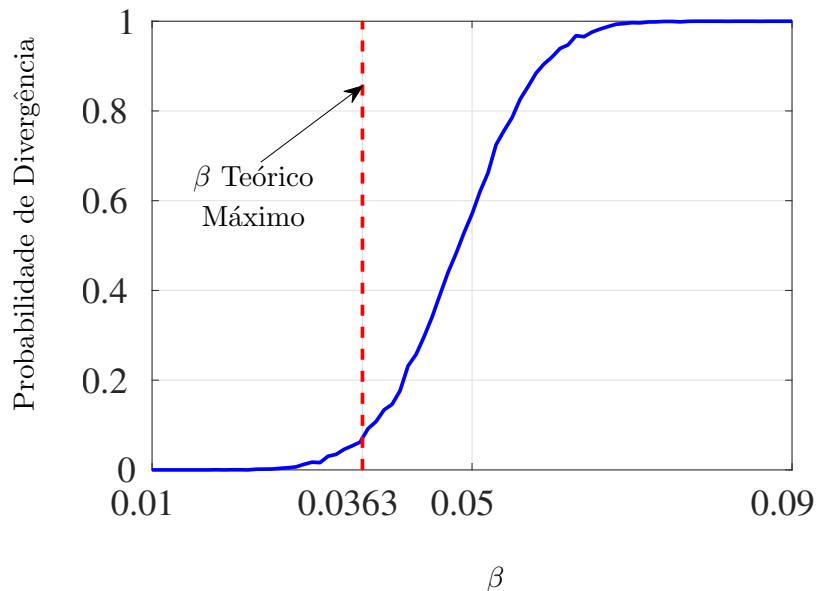


Figura VII.108: Curva de probabilidade de divergência simulada do algoritmo BLMS com  $L = 1$  e  $N = 10$ .

## BNLMS

**Cenário 1**  $\rightarrow N = 10$

$L = 5$

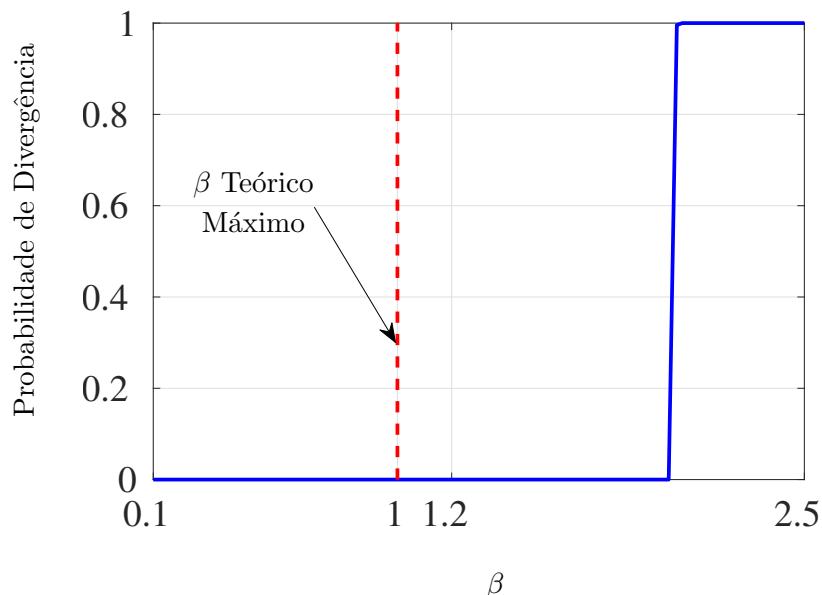


Figura VII.109: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 1$  e  $N = 10$ .

$L = 6$

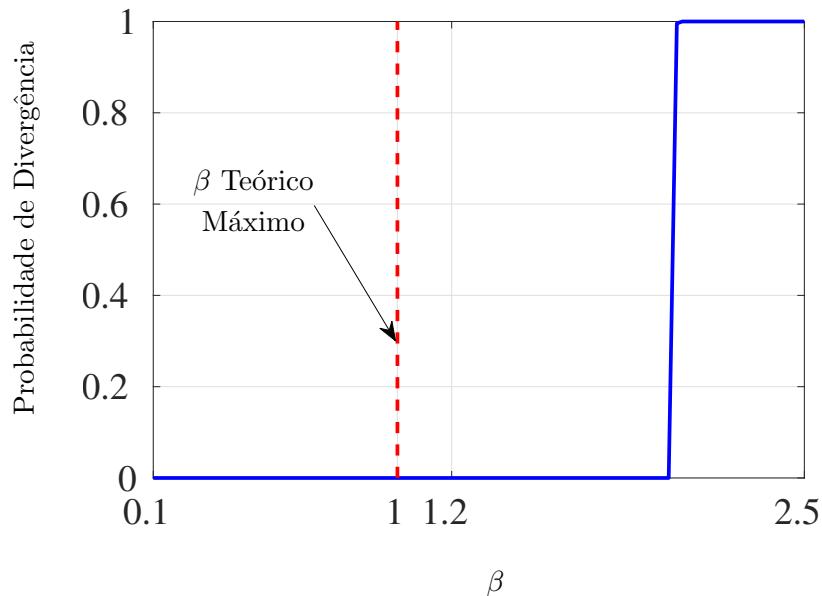


Figura VII.110: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 1$  e  $N = 10$ .

$L = 7$

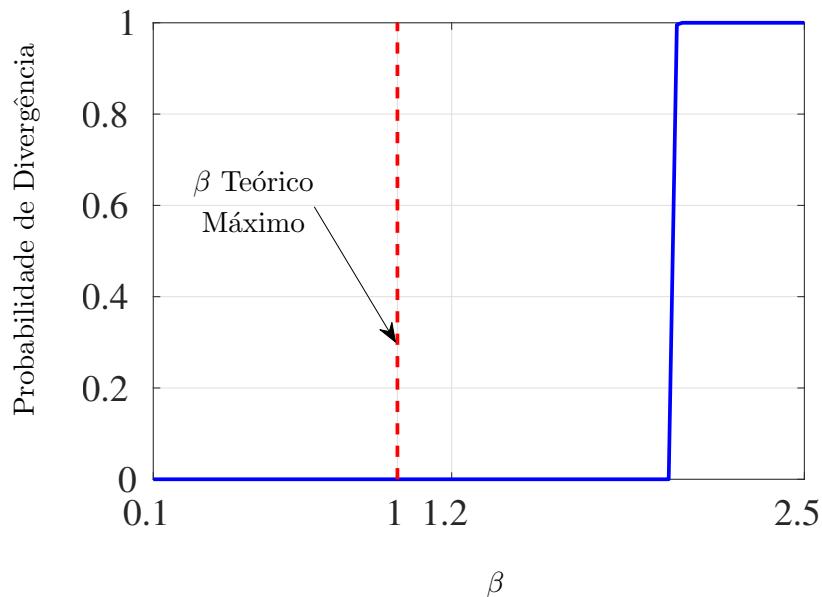


Figura VII.111: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 1$  e  $N = 10$ .

$L = 8$

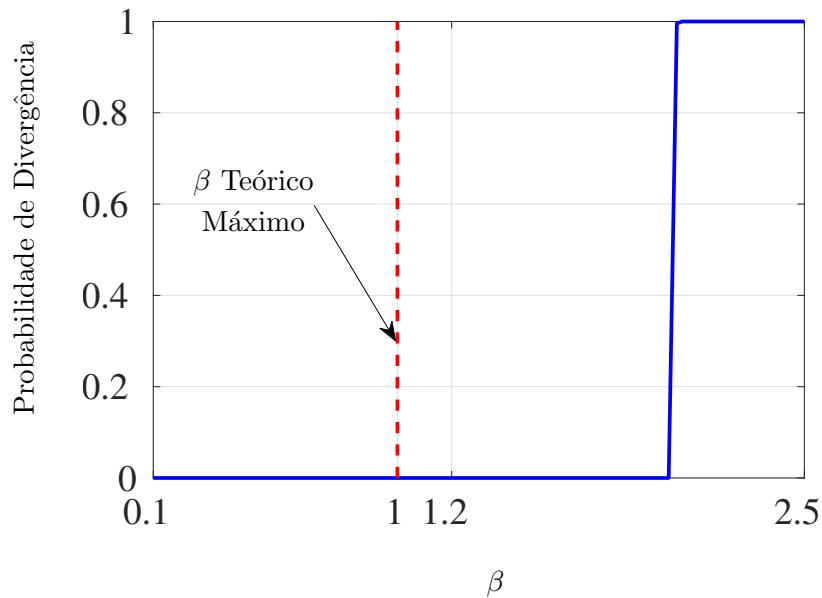


Figura VII.112: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 1$  e  $N = 10$ .

$L = 9$

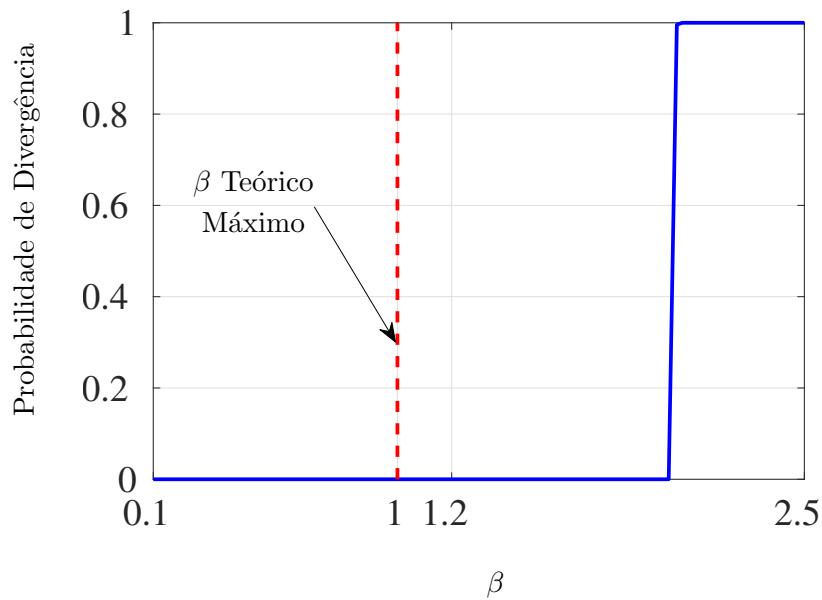


Figura VII.113: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 1$  e  $N = 10$ .

$L = 10$

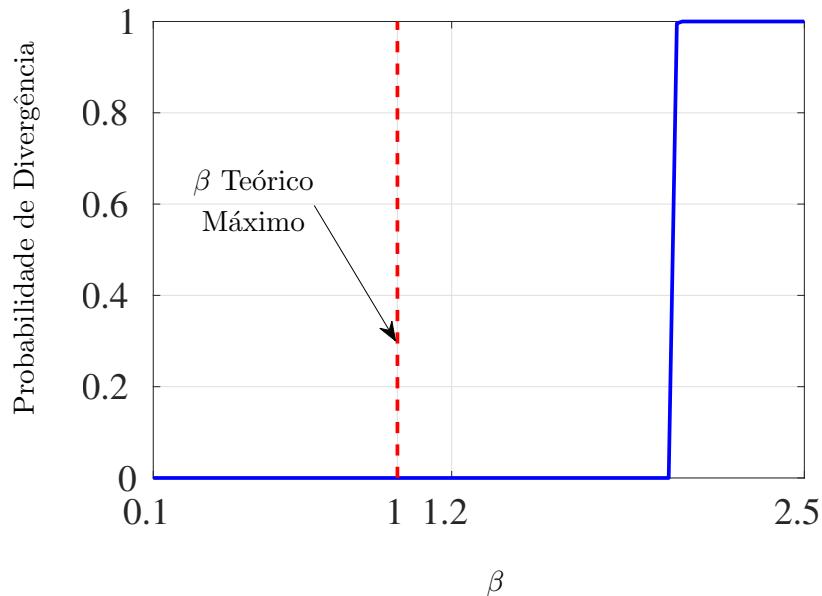


Figura VII.114: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 1$  e  $N = 10$ .

**Cenário 2** →  $N = 60$

$L = 3$

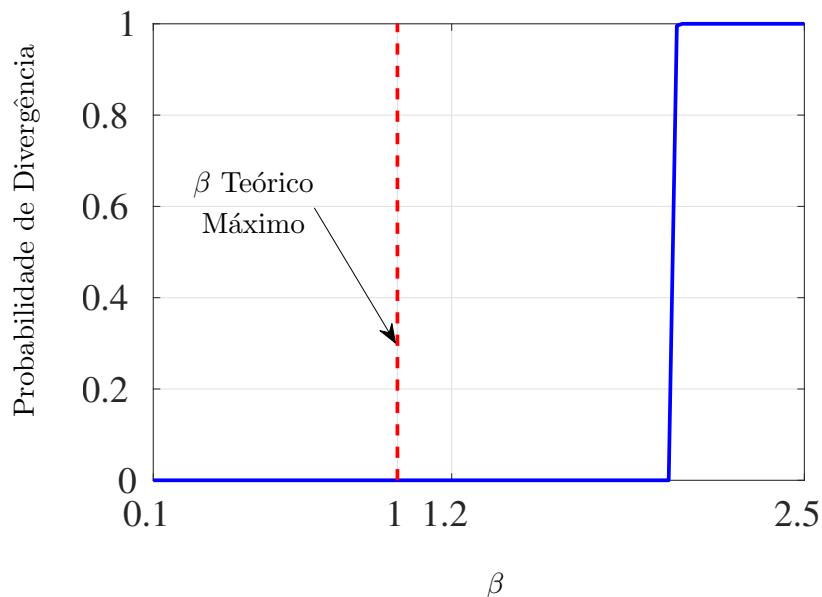


Figura VII.115: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 1$  e  $N = 10$ .

$L = 5$

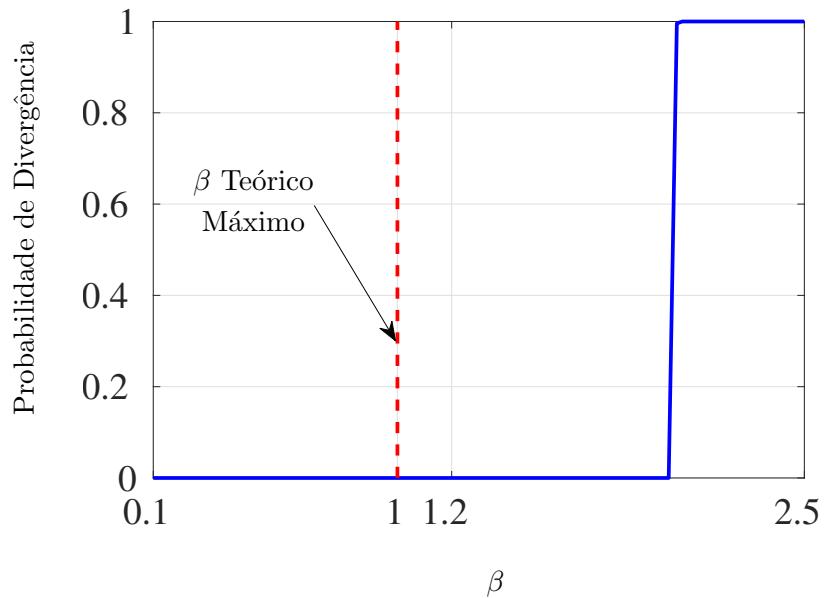


Figura VII.116: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 1$  e  $N = 10$ .

$L = 6$

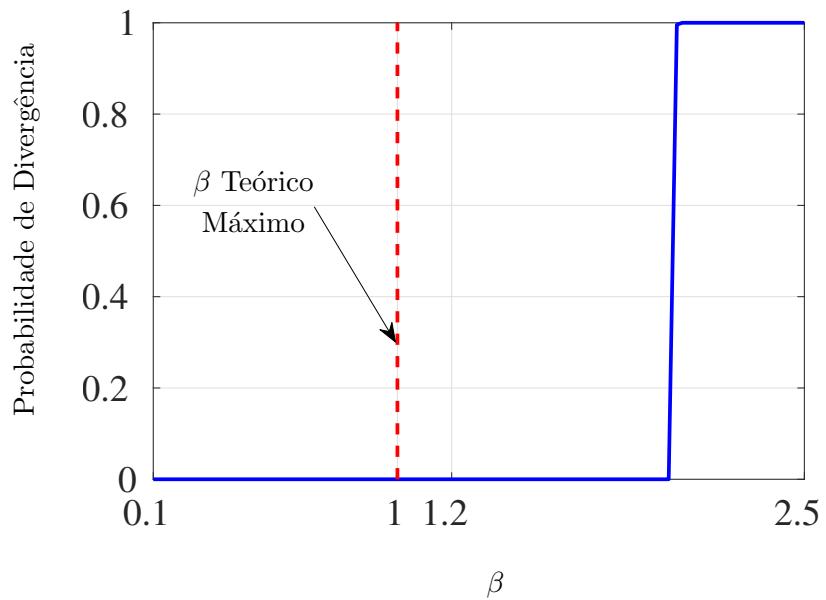


Figura VII.117: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 1$  e  $N = 10$ .

$L = 7$

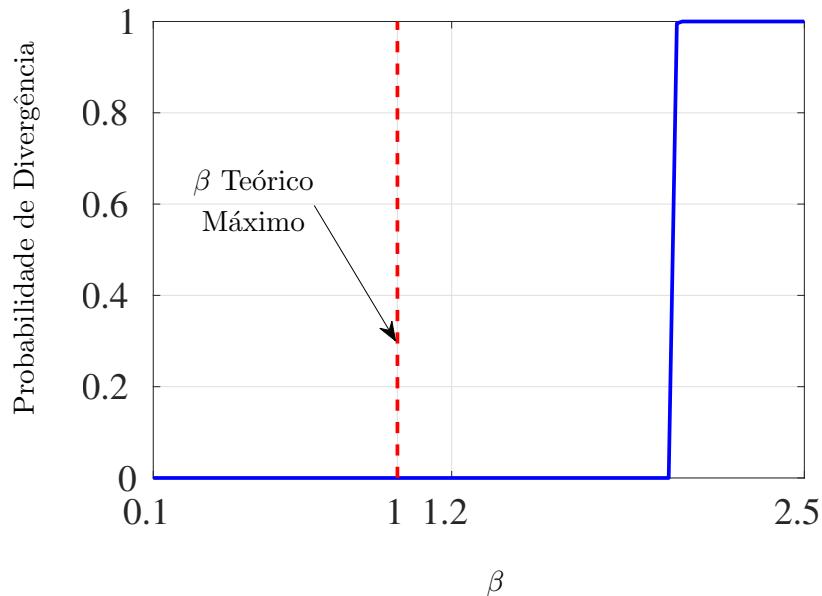


Figura VII.118: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 1$  e  $N = 10$ .

$L = 8$

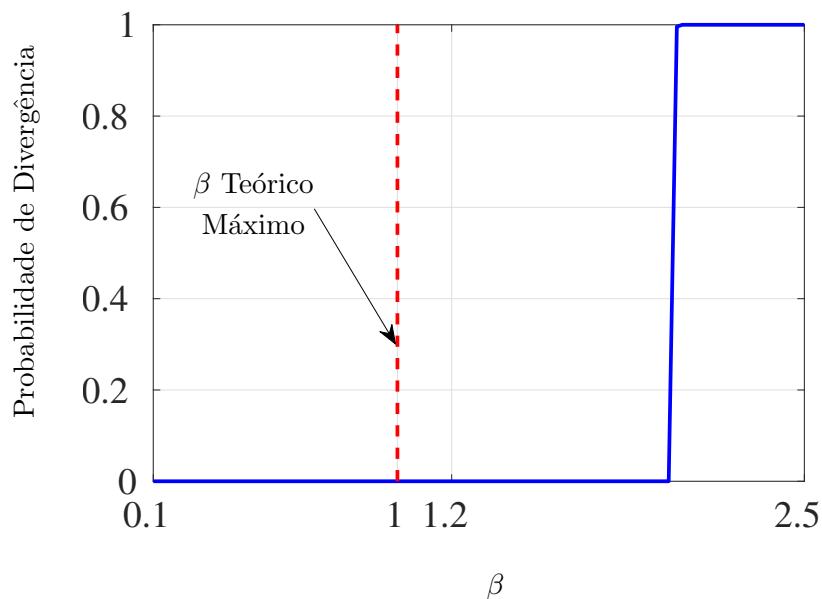


Figura VII.119: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 1$  e  $N = 10$ .

$L = 9$

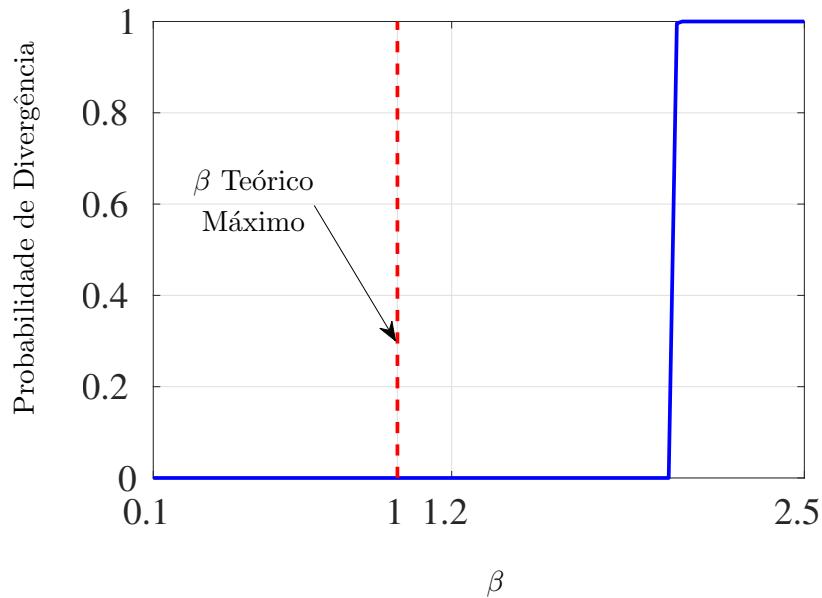


Figura VII.120: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 1$  e  $N = 10$ .

$L = 10$

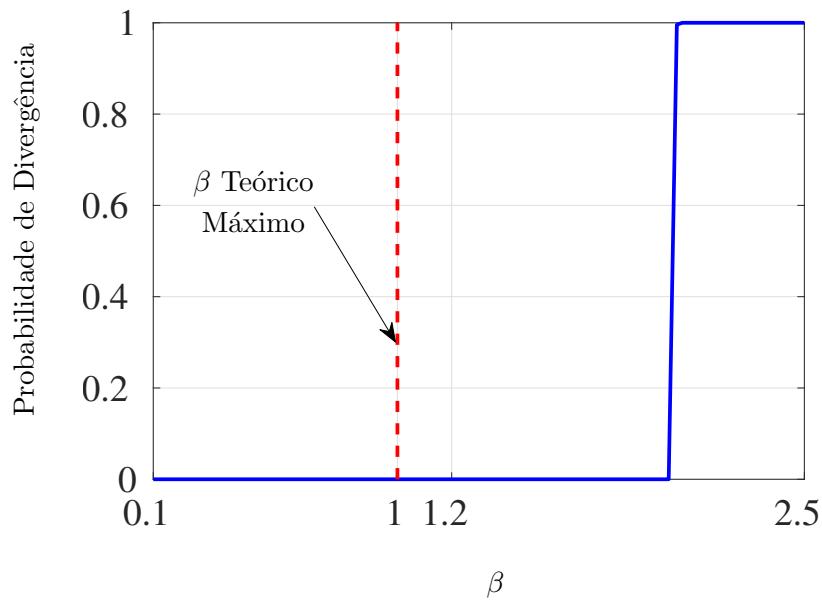


Figura VII.121: Curva de probabilidade de divergência simulada do algoritmo BNLMS com  $L = 1$  e  $N = 10$ .